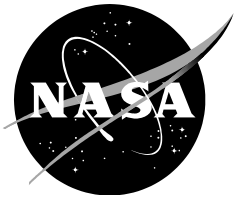
NASA/TP—2013–216508



Work Practice Simulation of Complex Human-Automation Systems in Safety Critical Situations:

The Brahms Generalized Überlingen Model

William J. Clancey
Ames Research Center, California
and Florida Institute for Human and Machine Cognition, Pensacola

Charlotte Linde, Chin Seah, Michael Shafto
Ames Research Center

May 2013

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

Access the NASA STI program home page at http://www.sti.nasa.gov

E-mail your question to help@sti.nasa.gov

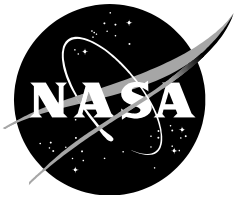
Fax your question to the NASA STI Information Desk at 443-757-5803

Phone the NASA STI Information Desk at 443-757-5802

Write to:

STI Information Desk
NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320

NASA/TP—2013–216508



Work Practice Simulation of Complex Human-Automation Systems in Safety Critical Situations:

The Brahms Generalized Überlingen Model

William J. Clancey
Ames Research Center, California
and Florida Institute for Human and Machine Cognition, Pensacola

Charlotte Linde, Chin Seah, Michael Shafto
Ames Research Center

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, CA 94035-1000

May 2013

Acknowledgments

This project was supported by the "Authority and Autonomy" task within the Aviation Safety Program (AvSP) of the System-Wide Safety and Assurance Technologies (SSAT) Project of NASA's Aeronautics Research Mission Directorate (ARMD). The work has been conducted at NASA Ames Research Center in the Intelligent Systems Division. Guillaume Brat, Neha Rungta, and Joseph Coughlan have helped us frame and focus the modeling and simulation effort. Ron van Hoof provided support for the Brahms simulation system.

The views expressed in this report are those of the authors and do not represent the views of NASA or the U.S. Government, nor do they necessarily correspond to the views of other researchers participating in the "Authority and Autonomy" task.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312
703-605-6000

This report is also available in electronic form at

http://ti.arc.nasa.gov/publications/

Table of Contents

Figures

Tables

1 Abstract

The transition from the current air traffic system to the next generation air traffic system will require the introduction of new automated systems, including transferring some functions from air traffic controllers to on-board automation. This report describes a new design verification and validation (V&V) methodology for assessing aviation safety. The approach involves a detailed computer simulation of work practices that includes people interacting with flight-critical systems. The research is part of an effort to develop new modeling and verification methodologies that can assess the safety of flight-critical systems, system configurations, and operational concepts.

The 2002 Überlingen mid-air collision was chosen for analysis and modeling because one of the main causes of the accident was one crew's response to a conflict between the instructions of the air traffic controller and the instructions of TCAS, an automated Traffic Alert and Collision Avoidance System on-board warning system. It thus furnishes an example of the problem of authority versus autonomy. It provides a starting point for exploring authority/autonomy conflict in the larger system of organization, tools, and practices in which the participants' moment-by-moment actions take place.

We have developed a general air traffic system model (*not* a specific simulation of Überlingen events), called the Brahms Generalized Überlingen Model (Brahms-GÜM). Brahms is a multi-agent simulation system that models people, tools, facilities/vehicles, and geography to simulate the current air transportation system as a collection of distributed, interactive subsystems (e.g., airports, air-traffic control towers and personnel, aircraft, automated flight systems and air-traffic tools, instruments, crew).

Brahms-GÜM can be configured in different ways, called *scenarios*, such that *anomalous events* that contributed to the Überlingen accident can be modeled as functioning according to requirements or in an anomalous condition, as occurred during the accident. Brahms-GÜM thus implicitly defines a class of scenarios, which include as an instance what occurred at Überlingen. Brahms-GÜM is a modeling framework enabling "what if" analysis of alternative work system configurations and thus facilitating design of alternative operations concepts. It enables subsequent adaption (reusing simulation components) for modeling and simulating NextGen scenarios.

This project demonstrates that BRAHMS provides the capacity to model the complexity of air transportation *systems*, going beyond idealized and simple flights to include for example the interaction of pilots and ATCOs. The research shows clearly that verification and validation must include the entire work system, on the one hand to check that mechanisms exist to handle failures of communication and alerting subsystems and/or failures of people to notice, comprehend, or communicate problematic (unsafe) situations; but also to understand how people

must use their own judgment in relating fallible systems like TCAS to other sources of information and thus to evaluate how the unreliability of automation affects system safety. The simulation shows in particular that distributed agents (people and automated systems) acting without knowledge of each others' actions can create a complex, dynamic system whose interactive behavior is unexpected and is changing too quickly to comprehend and control.

2 Introduction and Project Summary

This research report describes a new design verification and validation (V&V) methodology for assessing aviation safety. The approach involves a detailed computer simulation of work practices that includes people interacting with flight-critical systems. The simulation model is general, enabling what-if analysis of alternative work system configurations and thus facilitating design of alternative operations concepts.

This research is part of the "Authority and Autonomy" task within the Aviation Safety Program (AvSP) of the System-Wide Safety and Assurance Technologies (SSAT) Project of NASA's Aeronautics Research Mission Directorate (ARMD). The research is intended to provide methods for evaluating early-in-design models of complex interactions in which there are "multiple, different, simultaneous, situation-dependent assignments of authority and autonomy among both humans and automation." This effort explicitly includes organizational aspects of a work system: "what roles, functions, tasks, and activities are assigned to what actor in the organization?" (SSAT 2011). This project can be viewed as an experiment to evaluate the use of a particular, well-established work practice modeling tool, Brahms, with respect to the objectives of A&A research. This report argues that the experiment has been successful, leading to both valuable conclusions and suggestions for further research (Chapter 12.8).

Brahms is a multi-agent simulation system in which people, tools, facilities/vehicles, and geography are modeled explicitly (Clancey et al. 1998). In the Brahms modeling framework, the air transportation system is modeled as a collection of distributed, interactive subsystems (e.g., airports, air-traffic control towers and personnel, aircraft, automated flight systems and air-traffic tools, instruments, crew). Each subsystem, whether a person, such as an air traffic controller, or a tool, such as the ATCC[1] radar, is modeled independently with properties and contextual behaviors. The simulation then plays out the interactions among these separately existing models of subsystems (colloquially, the model is "run" to produce a chronology of behaviors in time, with the result called "a simulation run"). In this framework as in everyday work, authority is most often manifest as a combination of task responsibilities (i.e., enacting authority) and decision-making behavior in the context of guidance from multiple sources (i.e. following authority).

[1] See Glossary for acronyms.

The 2002 Überlingen mid-air collision (BFU Report 2004) has been chosen for this experiment using Brahms because systems like the Traffic Alert and Collision Avoidance System (TCAS 2012) deliberately shift authority from the air-traffic controller to an automated system. Thus, the Überlingen accident is often taken as a clear example of the problem of authority versus autonomy. It provides a starting point for exploring authority/autonomy conflict in the larger system of organization, tools, and practices in which the participants' moment-by-moment actions take place.

Here is a summary of the accident based on (Maiden et al. 2006). The accident is analyzed in Chapter 4 of this report with related information in Appendices 16-21.

> The Überlingen accident was a midair collision between two aircraft—a Tupolev Tu-154M passenger jet travelling from Moscow to Barcelona and a Boeing 757-23APF cargo jet travelling from Bergamo to Brussels. TCAS onboard both planes issued first a warning and then instructions for a change of course for both planes: a "Resolution Advisory." Seven seconds before TCAS' command to the Tupelov to climb, the air traffic controller in charge of the sector issued a command to descend, which the crew obeyed. Since TCAS had issued a Resolution Advisory to the Boeing crew to descend, both planes were descending when they collided.
>
> The immediate cause of the accident was the Tupelov crew's decision to follow the ATC's instructions rather than TCAS, although the regulations for the use of TCAS state that in the case of such a conflict, TCAS must be followed.
>
> This conflict of authority happened because a potential separation infringement between the two planes was not noticed by the air traffic controller early enough to issue instructions to one of the two planes to change course. Such potential separation infringements are frequent occurrences; it is part of the normal work of air traffic control to notice and correct them.
>
> A set of complex systemic problems at the Zurich air traffic control station contributed to the accident. Although two controllers were supposed to be on duty, one of the two was absent on a rest break—a common and accepted practice during the lower workload portion of night shift. On this evening, a scheduled maintenance procedure was being carried out on the main radar system, which meant that the controller had to use a less capable backup system. The maintenance work also disconnected the phone system, which made it impossible for other air traffic control centers in the area to alert the Zurich controller to the problem.
>
> Finally, the controller's workload was increased by a late-arriving plane, an Airbus 320, landing in Friedrichshafen. This required his attention and his

physical presence at a different work station. It also caused him to spend considerable time attempting to contact the Friedrichshafen controller by using the disabled phone system, thus distracting him from the potential separation infringement of the two planes.

Brahms is suitable for modeling such a scenario because control responsibility among people and automated systems can be represented in a flexible manner. In particular, a given agent/system can have more than one role/responsibility at a given time, and these roles/responsibilities can be reassigned during operations in a situation-dependent manner. For example, we can simulate that when an air traffic controller (ATCO) goes on break, as occurred at Überlingen, another ATCO shifts to handling multiple workstations. Simulated pilots and ATCOs also have context-dependent behaviors for communicating, following directions, and interacting with automated systems.

In summary, this report describes an air transportation system simulation model represented in the Brahms multi-agent framework and designed to satisfy these requirements:

- Extend **formal human-system performance modeling** from the individual level (one user, one task, one display) to the level of **complex multi-agent teams** (a choreography of people and automated systems);
- **Incorporate human experts and software agents** (e.g., TCAS);
- **Enable realistic mixed-initiative scenarios** that entail reconfiguration of airspace and reassignment of roles and responsibilities among human and software agents;
- Be consistent with providing Brahms with **formal semantics to enable using sophisticated software modeling tools** (e.g., Java Pathfinder) to provide useful analyses early in the design process.

Together these will demonstrate that the BRAHMS framework provides the capacity to model the complexity of air transportation *systems*, going beyond idealized and simple flights to include for example the interaction of pilots and ATCOs.

A work practice simulation represents chronological, located behaviors of people and automated systems. In contrast with functional models, which represent abstractly what behaviors accomplish (i.e., functions), a *behavioral model* represents what people and systems do, called *activities*. Activities include monitoring (looking), moving, communicating, reading and writing, all of which require time and occur in particular places with particular people, tools, materials, documents, and so on. In terms of work, a function model characterizes what a person or system does (e.g., "determine the altitude"), and a behavioral model represents how the work is done (e.g., move to see the altitude display and perhaps push a button, then perceive the altitude number). Figure 2-1 shows most of objects, systems, and human roles represented in the Brahms simulation presented in this report (not

shown are details such as Flight Plan Host Computer that communicates with ATC printers that print out Flight Control Strips).

The simulation is based on a fine-grained analysis of the published events of the Überlingen collision, relating spatial and temporal interactions of: 1) information represented on displays and documents at the air traffic control center and in the cockpit, 2) what controller(s) and cockpit crew were individually doing and observing, 3) alerts provided by automated systems, 4) communications within the cockpit and with air traffic control, 4) control actions to change automation and aircraft flight systems, 5) human beliefs and reasoning throughout regarding responsibilities of individuals and automated systems, progress appraisal of assigned responsibilities, and resolution of conflicting information/directives.

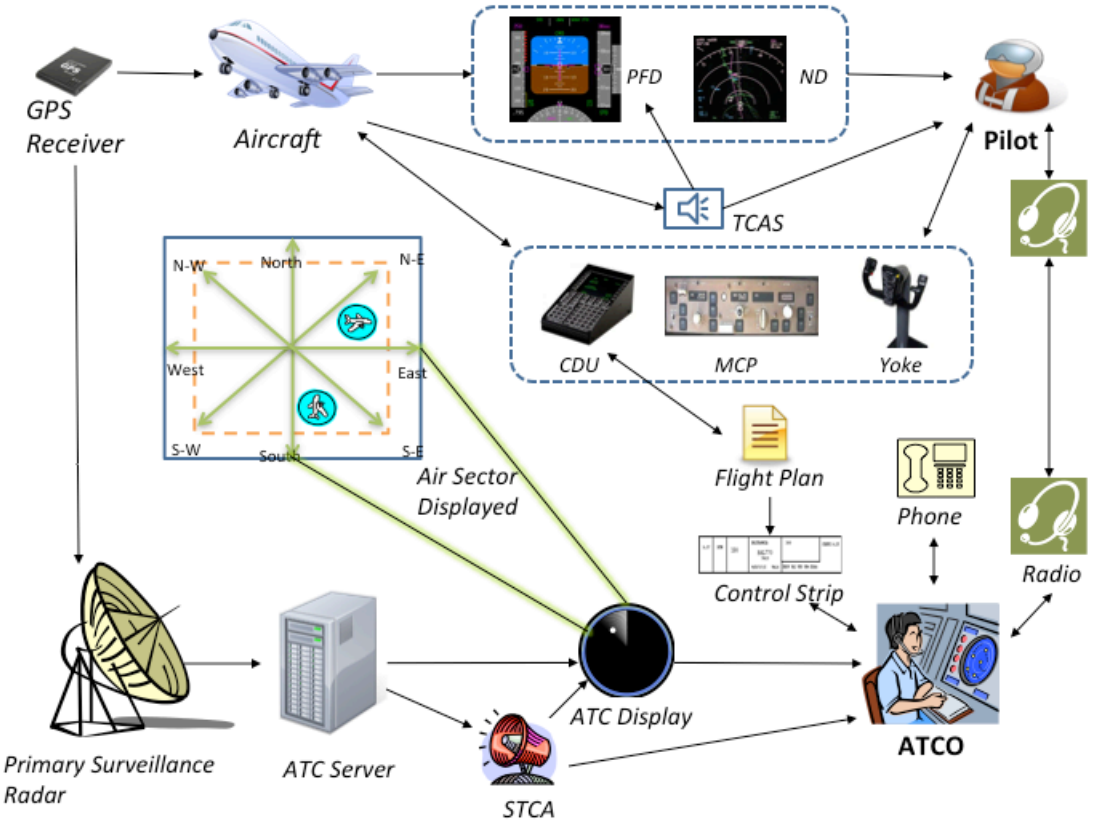


Figure 2-1: People and Systems Modeled and Simulated in Brahms-GÜM.

As mentioned above, the Überlingen case is of special interest because TCAS gave advice to one flight crew just seconds after they had already begun to follow a different directive from the Zurich air traffic controller. The "lessons learned" offered by the *BFU Investigation Report* stress the necessity of doing whatever TCAS instructs, but do not discuss the complexities involved in this advice. There are subtle psychological, social, and even physical coordination issues required by disengaging from an action in process that may make it difficult or impossible to follow this protocol. In particular, decision-making based on trust (Burnett et al.

2006) may be contextually bound to how people are mentally engaged in an already complex interaction with each other.

The analysis and model of the Überlingen collision makes the point that the issue of "authority" is as important as “autonomy” in designing automation for work systems. “Authority” may be defined by rules and protocols that the people and systems must follow, but in practice authority is a *relation* among actors, involving a mix of psychological, social, legal, and formal (mathematical and/or logical) interactions in a dynamic physical and temporal context. When aspects of the work system are missing or malfunctioning, interactions may be unpredictable, making an everyday complicated system into a complex system (Perrow 1999). During a complex human-automation interaction, as occurred at Überlingen, both people and automated systems are operating in an unknown and often unanticipated environment that they are creating for each other. A key objective of this project is to provide a means of formalizing and studying scenarios of interaction that might otherwise be unexpected, involving different configurations of human and system behavior, and thus potentially broaden the certification process beyond mathematical and logical relations of aircraft and automated systems to include human actions.

It is important to realize that the Brahms simulation model constructed in this research is not merely a replication of the Überlingen collision, that is, a single scenario of events. Rather the Brahms model created in this project consists of a generalization of all the subsystems (e.g., phones, radar, alert systems, aircraft, pilots, air-traffic controllers, ATCCs) that played a role in the Überlingen collision. We call it the *Brahms “Generalized Überlingen model”* (Brahms-GÜM). Rather than only representing the states and behaviors of these subsystems at the time of the collision, Brahms-GÜM represents the normal states and behaviors, but allows for them to be configured for each simulation “run” to characterize alternative behaviors, including absent, alternative, and dysfunctional or off-nominal forms (e.g., a pilot can follow TCAS or ignore it; the phones in an ATCC are not operating; a scheduled flight departs 15 minutes late).

In general, a Brahms model is configured by defining “initial facts” about the world, people, and subsystems, and “initial beliefs” and “group memberships” of people (conventionally, called the “initial parameters” of the model). Each of the many possible configurations of Brahms-GÜM parameters defines a *scenario*. Because of the variations in initial facts, beliefs, etc. and the probabilistic definitions of activity durations, each simulation run produces time-space-state interactions with potentially different outcomes. For example, in some configurations of the Brahms model, the Zurich ATCO notices the imminent collision and advises pilots before TCAS issues a traffic advisory. The combinations of all possible parameter settings define a *space of scenarios* that Brahms-GÜM should be able to validly simulate. What occurred at Überlingen is one scenario in that space.

In essence, the Brahms "Generalized Überlingen model" includes the proper practices and system functions that might have been present, as well as variations on practice and anomalous events that transpired during the Überlingen accident. The model development approach involved creating a series of complete (runnable) models that incrementally added off-nominal events and behaviors. This has enabled experimenting with arbitrary combinations of factors in a variety of scenarios (e.g., only one air traffic controller on duty, phone system not working, delayed flight requiring attention, degraded radar system).

As a starting point in creating the Brahms air transportation system, we adapted an existing functional model of how a pilot interacts with a flight automation system. We chose Pritchett's functional simulation, called "Work Model that Computes" (WMC, Pritchett & Feigh 2011) which was based on "cognitive work analysis," because it provided a ready-made framework detailing how different ways of configuring a flight management computer affected the aircraft and the pilot's complementary responsibilities. Adapting this simulation also enabled a direct comparison of cognitive work analysis to work practice analysis that is the theoretical basis of the Brahms activity framework. Specifically, this model development approach enables explicating from experience how a function model is converted into a work practice model. In particular, the Brahms-GÜM includes the perception, physical movements, and communications of the pilots as well as the ATCs, radar, telephones, radio, handoff protocols, TCAS, etc. The description of WMC, the Brahms-WMC model, and comparison appears in Appendix 17.

In summary, Brahms is useful for simulating complex human-automation interactions in safety-critical situations in the following ways:

- Shows how creating and experimenting with work practice models reveals interactions that are omitted, glossed over, or difficult to comprehensively describe in accident reports;
- Provides a principled way of determining where analysis requires psychological models, insofar as providing detailed behavioral models for all roles and activities becomes impractical;
- Provides a principled definition of "authority" and demonstrates how this is modeled and manifest in a multi-agent behavioral model;
- Reveals where formal methods are valuable, relative to systematic simulation of the parameter space (including the Monte Carlo method) and sensitivity analysis experiments.

Experimentation with Brahms-GÜM revealed that timing of events at the level of a few seconds made a substantial difference in the simulated outcomes. In particular, TCAS in 2002 was most vulnerable to an ATCO intervention with pilots a few seconds before it generates a resolution advisory, which is what happened at Überlingen. We had not encountered such sensitivity to timing and emergent interaction sequences in any of the prior Brahms models created over two decades.

This result is consistent with the claim that the degraded Überlingen work system was complex (Chapter 5) and provides evidence that the Brahms model appropriately represents and allows simulating a work system with complex human-automation interactions. The Brahms framework enables modeling the variability and dynamic implications of a work system that combines simultaneous agent activities and subsystem processes, and allows this model to be simulated in different configurations (scenarios) having contextual behaviors that interact in otherwise unpredictable ways.

We conclude that subtle issues of timing in human-automation interactions may arise when degraded or missing subsystems result in lack of information and inability to communicate, transforming a given configuration of flights that are routine in a normal work system to a situation too complex to handle. In particular, the events in the air traffic control center reveal how after people develop work practices in which they rely on automation (e.g., a collision warning alert), the absence of automation may cause the workload to increase and the evolving situations to become too cognitively complex to appropriately prioritize tasks or delegate responsibility.

A complementary research project, which is not presented in this report, aims to use model checking as a tool for developing, refining, and applying simulation models, in particular the Brahms simulation model developed here. The overall approach is to first focus on characteristics of work systems that we wish to model and understand, determine the strengths and weaknesses of the Brahms simulation framework in this regard, and subsequently determine how model-checking might enhance strengths and resolve some of the weaknesses.

We explicate how one might cast a work practice design simulation in terms of software engineering verification, emphasizing the challenges inherent in verifying a process model of a work system design that incorporates social and psychological scientific theories and assumptions about how people behave.

That is, the objective is not primarily a matter of "checking" the Brahms simulation, but using model checking to: 1) develop better/appropriate simulation models by indicating gaps, assumptions, lack of generality, or lack of flexibility for exploring some subspace of scenarios, 2) generate scenarios or, through formal analysis, provide scenario outcomes without running the model, and 3) construct a tool kit for scientifically understanding the behavior in human-automation systems and formulating principles for work system design. To this end, the objective of the present report is to provide an archival reference that documents the design and development of the Brahms-GÜM. Details about the analytic framework, challenges, and the refinement process are provided that may be useful for developing model checking tools that could facilitate the modeling process itself, as well as to be useful for using the model to discover properties about the work system, such as potential failures involving human-automation interaction.

Subsequent chapters in this report describe:

- The broader NextGen research program to which this project is designed to contribute (Chapter 3)
- The Überlingen collision facts, Normal Accident Theory analytic framework, and systemic failure analysis of the accident, emphasizing the nature of complexity (Chapters 4, 5,6)
- Further background about Brahms and work practice modeling with comparisons to other frameworks (Chapter 7)
- The development and structure of the Brahms Generalized Überlingen Model (Chapter 8), including details about modeling challenges and abstractions used (Chapter 9), and the methodology and rationale for refining and scoping the model to produce quantifiable analyses (Chapter 10).
- Discussion of authority and automation with respect to Brahms-GÜM (Chapter 11).
- Discussion of issues relevant to verification and validation of a work practice model and simulation—and why on the basis of the function and fallibility of TCAS, certifying this automated system requires a work practice simulation (Chapter 12)
- Conclusions and recommendations about using Brahms-GÜM for simulating human-automation systems with reference to the objectives of the Aviation Safety research program, lessons learned using Brahms, and prior recommendations from the National Academy of Sciences (Chapter 12.8).

Appendices provide details about the Überlingen accident and unexplained events (Appendices 16 - 18) ; the TCAS logic and protocol (Appendices 19 and 21); and Brahms-GÜM components, scenario configurations, simulation graphics, an annotated simulation run, and limitations (Appendices 22 - 28).

3 Background: NextGen Research Objectives & Requirements

This chapter briefly presents the air transportation system context and the research topic it motivates, followed by the requirements that have guided the definition and methods of the Brahms simulation effort.

3.1 NextGen ATS Problem and Approach

By 2025 US Air traffic is expected to double or triple, increasing density of flights with new aircraft classes and operational concepts, characterized as the Next Generation Air Transportation System ("NextGen ATS"; see FAA 2013, JPDO 2013). From one perspective, NextGen challenge might be described as "to keep collision risks low while increasing the occasions for collisions" (Perrow 1984, p. 158):

> [NextGen] proposes to transform America's air traffic control system from an aging ground-based system to a satellite-based system. GPS technology will be used to shorten routes, save time and fuel, reduce traffic delays, increase capacity, and permit controllers to monitor and manage aircraft with greater safety margins. Planes will be able to fly closer together, take more direct routes and avoid delays caused by airport "stacking" as planes wait for an open runway….
>
> Once implemented, NextGen will allow pilots and dispatchers to select their own direct flight path, rather than using a grid-like highway system. By 2020, aircraft are expected to be equipped to tell pilots exactly what their location is in relation to other aircraft, enabling planes to fly closer together safely. By providing more information to ground control and planes, planes are expected to land faster, navigate through weather better and reduce taxi times so flights and airports themselves can run more efficiently. The increased scope, volume and distribution of information is intended to help planes land faster, improve weather forecasts, automation and information sharing, as well as reduce taxi times. ("Next Generation Air Transportation System," Wikipedia, accessed 19 September 2012)

To manage risk within this growth regime, the Aviation Safety Program (AvSP) within NASA/ARMD seeks to "develop transformational methods, tools and techniques that advance safety assurance of complex, networked, distributed flight critical systems."[2] Referring in particular to the *Assurance for Flight Critical Systems* technical theme, this research has been described as:

> …the exploration and extension of mathematical approaches to systems engineering and safety analysis, based on formal methods usually associated with software engineering. The substantive issues being addressed span Aviation Safety and Airspace Systems, aiming to provide sophisticated model-based safety analyses of NextGen airspace control technologies being considered by the Joint Planning and Development Office (JPDO). [3]

[2] Sharon Graves, LaRC acting project lead, July 2010 overview slides.
[3] Michael Shafto, 17 Dec 2010 memo, Intelligent Systems Division, NASA Ames.

The particular technical theme of the task, *Authority and Autonomy*, explores methods

> …for extending formal human-system performance modeling from the individual level (one user, one task, one display) to the level of complex multi-agent teams incorporating human experts and software agents in realistic mixed-initiative scenarios. These scenarios may entail reconfiguration of airspace and reassignment of roles and responsibilities among human and software agents. The best examples of such scenarios in current-generation airspace concern Traffic-alert and Collision Avoidance System (TCAS) scenarios.
>
> The technical approach…is to adapt existing agent-based modeling systems (e.g., Brahms) and to provide them with formal semantics. Then sophisticated software modeling tools (e.g., Java Pathfinder [developed for software verification]) may be able to provide useful analyses early in the design process.

The SSAT Project Plan explains how the concepts of authority and autonomy arise in designing complex systems providing multiple functions that support many operating models, environments, and technologies:

> The ATS, especially with future NextGen concepts of operation, is a complex system involving dynamic interactions among multiple actors that are largely governed through formal assignment of roles and responsibilities. These assignments of authority and autonomy are made at the design level, but are executed at the operational level according to each actor's view of their roles and responsibilities. Operationally, the system continuously adjusts for shortcomings in the assignment of authority and autonomy, for shortcomings in the capacity of actors to perform their assigned roles and responsibilities, and to optimize various performance factors such as capacity, environmental impact, and safety. This suggests that system safety should be derived not only from a predictable execution of assigned roles and responsibilities but also from checks and balances to ensure that the system operates as designed in the face of failures, disturbances and degradations. The ability of the system to operate in off-nominal conditions as a result of the checks and balances extent in it provides resilience, a critical characteristic for system safety.

The objective of the A&A research area is to develop methods to ensure that flight-critical systems are free from safety concerns in the assignment of authority and autonomy, in terms of their comprehensiveness and lack of conflicts and ambiguities and in terms of their correspondence to system safety objectives including resilience. This research must account for context where capabilities may be degraded, for temporal effects during transition of authority and autonomy (including both transient and enduring problems), and for the dynamics of delegation involving both humans and automation.[4]

[4] This paragraph and following text are excerpted and adapted from the SSAT (2011).

In summary, this research focuses on developing **new modeling and verification methodologies that can assess the safety of flight-critical systems, system configurations and operational concepts.**

The research considers the air transportation system as a distributed, interactive system of systems with authority and autonomy assigned to both humans and automation at multiple levels. The approach is to develop modeling and V&V methods that can be applied to proposed concepts and configurations early in the development process to identify promising candidates as well as find design problems when they are easier to fix. This combination of modeling and V&V is intended to increase assurance of safety and motivate adoption of advanced automation and associated operations protocols.

3.2 Authority and Autonomy Research Theme

In one common formulation, the nature of A&A is characterized as an "allocation problem"—in which a work system consists of well-defined, bounded functional roles that satisfy the need for actors to have an unambiguous understanding of each other's actions and their consequences. The assumption is that authority bounds (limits) behavior in terms of ownership—who has authority in any situation—and how it may affect safety.

We provisionally adopt the definitions from an NASA Research Announcement:[5]

- *Authority* refers to having the right, or power, to exercise controls or issue air traffic commands that impact the position, velocity, and/or attitude of aircraft during operations.
- *Autonomy (or automation)* refers to a function or system that can operate independently of pilot or air traffic controller intervention.

Pilots and controllers in commercial airline operations may delegate their authority to automation for selected activities or functions (e.g. auto-land systems). The pilot remains responsible for monitoring the performance of the automation to assure it performs its intended function and to reclaim authority should it fail. This paradigm has worked well and has been demonstrated to be safe for many situations—due largely to rigorous V&V processes and well-defined and trained procedures. Nevertheless, in some situations the V&V process and/or operational procedure designs have failed and accidents have resulted. An example is the Überlingen mid-air collision in 2002 (BFU 2004), which grounds and focuses the analysis and modeling of this report.

[5] In this section we adopt and largely paraphrase NASA NRA Subtopic AFCS-1.4 (Authority and Autonomy): AMENDMENT No. 8 TO THE NASA RESEARCH ANNOUNCEMENT (NRA) ENTITLED "RESEARCH OPPORTUNITIES IN AERONAUTICS – 2011 (ROA- 2011)," NNH11ZEA001N, RELEASED August 26, 2011, pp. 19-25. In general the text of this announcement is incorporated and adapted in this section without further citation.

The future air transportation system, commonly referred to as "NextGen," anticipates a more fluid sharing of responsibility and authority, particularly with regard to flight path management. Four-dimensional trajectories or trajectory changes may be defined and executed by automated systems, ground-based "controllers," and/or pilots, collectively referred to as "agents." Further, *collaborative decision-making* (CDM) is promoted by NextGen, by which for example agents may negotiate flight trajectories. Furthermore, a variety of authority models are being proposed, ranging from the current model: "the pilot always has final authority" to the proposed, more controversial "automation can take over" mode of operation.

Furthermore, given the flexibility of allocation of authority and autonomy in NextGen, it makes sense to *expand the concept of safety from clear-cut safety conditions to the notion of resilience of a system composed of communicating organizations and agents*. *Safety resilience* is defined as the ability of a system to keep functioning safely in the presence of (possibly compounding) disturbances
This is especially of concern when human flexibility is confronted with the rigidity and failure of autonomous systems. Therefore, NextGen research focuses not only on nominal system behaviors but also on the structure and response of the ATS in off-nominal conditions.

This A&A research does not focus specifically on the design problem of determining the appropriate capabilities for each element of the system given its position within a broader distributed system context. Rather, it focuses on *methods for creating and evaluating early-in-design representations of systems* that include multiple, different, simultaneous, situation-dependent assignments of authority and autonomy among both humans and automation. This concept is broadly associated with analyzing the organizational aspects of a system: what roles, functions, tasks, and activities are assigned to what actor in the organization? Also, the interactional aspect requires systemic, "total system" analyses and models, relating in particular to the notion of distributed, simultaneous behaviors and events by different agents that may interact in complex, unanticipated ways (a pivotal characteristic of the Überlingen accident).

Previous V&V methods addressing human-automation problems have focused on scenarios restricted to a confined "operator-interface," such as a pilot and the plane's cockpit displays or a controller and traffic advisory displays. In contrast, V&V techniques applied to new concept of operations currently take a broad view, but often rely on crude models of people and automation. High-fidelity simulations (possibly with people in the loop) are expensive, time-consuming, and cover only some, usually highly simplified scenarios. This report provides an approach for bridging this gap through a "work practice" analysis and simulation.

3.3 Scenario Requirements

The project reported here using the Brahms simulation framework addresses requirements recommended for NASA AFCS research projects[6]:

1. Following the principle that good scientific research is grounded in real-world phenomena, we seek to develop and evaluate the applicability of V&V techniques for detecting A&A problems in realistic scenarios. Realistic scenarios have the following properties:

    - Sufficiently complex to model new aeronautics concepts and designs
    - Defined to expose problems associated with assignments of authority and function across the multi-agent (i.e., human and automation) design space.
    - Enable observing and measuring resiliency, the capability of a system to compensate for errors when they occur. That is, scenarios should be developed that enable the resiliency of a system to be observed and measured.

2. The project should help NASA understand how previous research on complex work environments, such as the design and evaluation of new indications/displays and alerting strategies, can be leveraged to perform V&V at the level of concept of operations.

3. Scenarios should have sufficient fidelity to study designs (i.e., operations concepts) at various levels of detail and to demonstrate or evaluate the applicability of other V&V methods and tools.

    - Scenarios should specify a set of bounding conditions, parameters, or assumptions that do not change during an operation. These include, for example: aircraft class, crew size, operating rules, and equipage.
    - Scenarios need not capture all operational details (e.g. the color and position of an indicator or button), details can be used to provide useful abstractions as to possible interactions.

These requirements are addressed by 1) the choice of the Überlingen collision (detailed in the next chapter), 2) the adaptation of an existing functional allocation simulation of human-automation interaction ("Work Model that Computes," Pritchett & Feigh 2011; Pritchett et al. 2011; see also Section 8.4), and 3) the use of Brahms to create a generalized model defining a well-defined space of scenarios whose parameters include the off-nominal factors that contributed to the collision (Chapter 8).

[6] Adapted from AMENDMENT No. 8 TO THE NASA RESEARCH ANNOUNCEMENT (NRA) ENTITLED "RESEARCH OPPORTUNITIES IN AERONAUTICS – 2011 (ROA- 2011)," NNH11ZEA001N, RELEASED August 26, 2011, pp. 19-25.

4 Überlingen Collision Overview

To provide sufficient background for the subsequent presentation and discussion of Normal Accident Theory (NAT), this chapter describes the Überlingen collision and why it was chosen for this project. NAT is an analytic framework applicable to complex human-automation systems; it is particularly useful for understanding how the local ATS became complex during the sequence of Überlingen events.

4.1 Choice of Überlingen Accident as Research Focus

The Überlingen accident, involving the TCAS air traffic advisory system, is often taken as a clear example of the problem of authority versus autonomy (A&A). It combines several well-known causes of errors: people and automated systems have different information about a situation and adopt different strategies; workload impairs performance or causes distraction, automation is not trusted, etc. (Riley et al. 1996). We have therefore chosen the Überlingen accident as a starting point for exploring the larger space of organization, roles, tools, procedures, and facilities in which air transportation work takes place:

- The Überlingen collision is a paradigmatic example of A&A conflicts. In particular, TCAS has ability to reconfigure the pilot-ATCO relationship, taking authority from the ATCO and telling the pilot what to do.
- The Überlingen collision was not an isolated event involving conflicts between TCAS and an ATCO:

  > About a year before the Bashkirian Airlines-DHL collision there had already been another incident involving confusion conflicting TCAS and ATCO commands. During the 2001 Japan Airlines mid-air incident, two Japanese airliners nearly collided with each other in Japanese skies. Both aircraft had received conflicting orders from the TCAS and ATC; one pilot followed the instructions of the TCAS while the other did not. Disaster was only averted because one of the pilots made evasive maneuvers based on a visual judgment…. As a consequence Japan called for measures to prevent similar incidents. However, the International Civil Aviation Organization (ICAO) did not take action until after the crash over Germany. In addition four near misses in Europe occurred before the German disaster, because one set of pilots obeyed the air traffic controllers while the other obeyed TCAS. *(Wikipedia, "Überlingen mid-air collision", accessed 19 September 2012).*

- The Überlingen collision was the basis for a significant revision to the TCAS algorithm (from version II 7.0 to II 7.1), requiring over a decade to formalize and deploy (see Appendix 19).
- The Überlingen collision proves that methods used for certifying TCAS II 7.0 did not properly consider human-automation interactions. In particular, the certification method treated TCAS as if it were flight system automation, that is, a system that automatically controls the flight of the aircraft. Instead, TCAS is a system that tells pilot how to maneuver the aircraft, an instruction that implicitly removes and/or overrides the air traffic controller's authority.

- Furthermore, the fallibility of TCAS means that understanding how this automated system affects aviation safety requires understanding how pilots integrate its advice with other sources of information (Section 12.8)

TCAS is an onboard aircraft system that uses radar transponder signals to operate independently of ground-based equipment to provide advice to the pilot about conflicting aircraft that are equipped with the same transponder/TCAS equipment.[7]

The history of TCAS dates at least to the late 1950s. Motivated by a number of mid-air collisions over three decades, the FAA initiated the TCAS program in 1981.[8] The system in use over Überlingen in 2002 was TCAS II 7.0, which had been installed by US carriers since 1994:

> TCAS II issues the following types of aural annunciations:
>
> - Traffic advisory (TA)
> - Resolution advisory (RA)
> - Clear of conflict
>
> When a TA is issued, pilots are instructed to initiate a visual search for the traffic causing the TA. If the traffic is visually acquired, pilots are instructed to maintain visual separation from the traffic… When an RA is issued, pilots are expected to respond immediately to the RA unless doing so would jeopardize the safe operation of the flight.

The separation timing, called TAU, provides the TA alert at about 48s and the RA at 35s prior to predicted collision; which corresponds precisely to the events over Überlingen. For reasons that are not documented, a secondary "increase descent/climb" RA was provided to both aircraft but at different times (p. 62).

4.2 Überlingen Scenario Narrative

The following is a summary of the Überlingen accident (Maiden et al. 2006); it is discussed in detail in subsequent sections; an annotated timeline appears in Appendix 17.

> On July 1 2002, a midair collision between a Tupolev Tu-154M passenger jet travelling from Moscow to Barcelona, and a Boeing 757-23APF cargo jet manned by two pilots, travelling from Bergamo to Brussels, occurred at 23:35 UTC over the town of Überlingen in southern Germany. The two flights were on a collision course. TCAS issued first a Traffic Advisory (TA) and then a Resolution Advisory (RA) for both planes. Just before TCAS' RA to the Tupelov to climb, the air traffic controller in charge of the sector issued a command to descend, which the crew obeyed. Since TCAS had issued a Resolution Advisory to the Boeing crew to descend and that they immediately followed, both planes were descending when they collided.

[7] For more detailed history and analysis see, Kuchar and Drumm (2007).

[8] "Traffic Alert/Collision Avoidance System," Aeronautics Learning Laboratory for Science, Technology, and Research, accessed 19 September 2012.

> The immediate cause of the accident, which represents the conflict between the authority of automated systems (TCAS) and people (crews and ATC), as well as their autonomy (freedom to act independently), was the Tupelov crews' decision to follow the ATC's instructions rather than TCAS, although the regulations for the use of TCAS state that in the case of such a conflict, it takes precedence.
>
> The potential for this conflict came about because a potential separation infringement between the two planes was not noticed by ATCO early enough to issue instructions to one of the two planes to change course. Such potential separation infringements are frequent occurrences; it is part of the normal work of air traffic control to notice and correct them.
>
> Leading to this was a set of complex systemic problems at the Zurich air traffic control station. Although two controls were supposed to be on duty, one of the two was resting in the lounge: a common and accepted practice during the lower workload portion of night shift. On this evening, a scheduled maintenance procedure was being carried out on the main radar system, which meant that the controller had to use a less capable air traffic tracking system. The maintenance work also disconnected the phone system, which made it impossible for other air traffic control centers in the area to alert the Zurich controller to the problem.
>
> Finally, the controller's workload was increased by a late arriving plane, an Airbus 320, landing in Friedrichshafen. This required his attention, compounded by the unavailability of the phones, distracting him from the potential separation infringement of the two planes.

4.3 Protocol for Pilot Interaction with TCAS and ATCO

The role of the ATCO relative to TCAS's advisories is at the heart of the work system design problem that this project and report investigates: Which verbal instruction should the pilot obey, the one uttered by the ATCO or by TCAS? By one account, the authority is clear, TCAS is always in control:

> This means that aircraft will at times have to manoeuver contrary to ATCO instructions or disregard ATCO instructions. In these cases, the controller is no longer responsible for separation of the aircraft involved in the RA until the conflict is terminated. (TCAS 2012)

However, the claim that "the controller is no longer responsible" is qualified:

> On the other hand, ATCO can potentially interfere with the pilot's response to RAs. If a conflicting ATCO instruction coincides with an RA, the pilot may assume that ATCO is fully aware of the situation and is providing the better resolution. But in reality ATCO is not aware of the RA until the RA is reported by the pilot. Once the RA is reported by the pilot, ATCO is required not to attempt to modify the flight path of the aircraft involved in the encounter. Hence, the pilot is expected to "follow the RA" but in practice this does not yet always happen. (TCAS 2012)

Pilot training emphasizes:

- Do not manoeuver in a direction opposite to that indicated by the RA because this may result in a collision.
- Inform the controller of the RA as soon as permitted by flight crew workload after responding to the RA. There is no requirement to make this notification prior to initiating the RA response.

Pritchett (2012) discusses situations in which it may be permissible for a pilot to disobey an RA, based on other information or experience that pilots may have about a particular situation in which a TCAS alert may occur. Other information may be provided by party-line communications, charts highlighting "normal" traffic flow, and "call outs" by the ATCO. Experience with the airspace, the aircraft, and of course TCAS advisories are also important. Overall, her analysis shows that pilot non-compliance with TCAS is not always an error.[9] Indeed, casting the problem of pilot compliance as an "authority" issue comes from viewing non-compliance as failure to obey, when in fact pilots know that TCAS is fallible (see also Section 12.8).

Commentary about the Überlingen collision has ranged over a variety of interpretations regarding the nature of a TCAS RA. Most notably, Frank Fischer, speaking for ANSA submitted a letter to Eurocontrol (ANSA+AirRadio, 2004, p. 67) prior to the publication of the BFU Report:

> As the term "RA - Resolution Advisory" implies, pilots are given an advice on how to resolve the indicated conflict. They are not being instructed. The term RA had been introduced in the respective ICAO procedures to match with the internationally agreed ACAS procedures … leaving it up to the pilot to take the last decision, if to follow the RA or to take another deconflicting course of action….
>
> Furthermore, air traffic regulations in the Russian Federation, in force at the time of the accident, forced pilots to give preference to instructions for evasive manoeuvres by ATCO before following a TCAS - RA.
>
> It is therefore unfortunate that the public is being mislead by such disseminated information, which often leads to prejudice and premature decision on who was guilty. The ICAO procedure explicitly excluded the obligation of pilots to obey a TCAS - RA: Also Eurocontrol´s ACAS training brochure as well as the Swiss and German AIPs only state that pilots "should" follow an RA, but not "shall", i.e. under all circumstances….

[9] In contrast with the analytic perspective presented in this project, the model of the "human as machine" applied by Pritchett follows the cognitivist approach of emphasizing the role of knowledge of an individual (the pilot), omitting the interactive aspects of perception and attention, including effects of the physical layout in getting information and using controls (Hutchins 2000).

As recently as late 2011, an FAA safety alert indicated that "FAA guidance permits non-compliance with an RA under certain circumstances."[10] The "Introduction to TCAS II v. 7.1" booklet provides this advice (FAA 2011, p. 39):

> TCAS does not alter or diminish the pilot's basic authority and responsibility to ensure safe flight. Since TCAS does not detect aircraft that are not transponder equipped or aircraft with a transponder failure, TCAS alone does not ensure safe separation in every case. Further, TCAS RAs may, in some cases, conflict with flight path requirements due to terrain, such as an obstacle-limited climb segment or an approach towards rising terrain. Since many approved instrument procedures and IFR clearances are predicated on avoiding high terrain or obstacles, it is particularly important that pilots maintain situational awareness and continue to use good judgment in following TCAS RAs. Maintain frequent outside visual scan, "see and avoid" vigilance, and continue to communicate as needed and as appropriate with ATC.

Therefore the distribution of potentially conflicting authority includes the pilot, not just TCAS versus ATCO. The pilot remains responsible for ensuring safe flight.

One role of a simulation model is to enable formalizing and studying different kinds of interactions, such as those described here, so better procedures and training can be provided to both pilots and controllers. The TCAS introduction booklet reviews operational experience that provides evidence that simulation-based training has been effective (p. 44). Training to handle failures in the logic producing incorrect or "nuisance RAs" is of particular importance (p. 45).

The Überlingen accident was one of the prime motivations for implementing TCAS II 7.1, which includes a "reversal" logic that had long been considered (from Wikipedia, "Überlingen mid-air collision", accessed 19 September 2012):

> Before this accident [Überlingen] a change proposal (CP 112) for the TCAS II system had been issued. This proposal would have created a "reversal" of the original warning - asking the DHL plane to climb and the Tupolev crew to descend. According to an analysis by Eurocontrol this would have avoided the collision if the DHL crew had followed the new instructions and the Tupolev had continued to descend.[citing BFU Report, 2004, p. 35]
>
> Additionally, an automatic downlink for the TCAS - which would have alerted the air traffic controller - had not been deployed worldwide at the time of the accident. [citing BFU Report, 2004, p. 50]

Worldwide deployment of TCAS II 7.1 was still in process in 2012, a decade after the Überlingen collision.

[10] Safety Alert for Operators 11010, 11/7/11
http://www.faa.gov/other_visit/aviation_industry/airline_operators/airline_safety/safo/all_safos/media/2011/SAFO11010.pdf

5 Analytic Framework: Normal Accident Theory

Normal Accident Theory (NAT, Perrow 1999) provides an especially appropriate framework for analyzing the causes of the Überlingen collision because of the systemic nature of the accident, including organizational factors, complex interactions among people with different roles and automation, and the tight coupling with short time dependencies within the local Air Traffic System (ATS) caused by non-operating equipment.

To begin, the collision is properly characterized as a "system accident." In NAT, a system consists of subsystems, composed of parts that are composed of units. Failures of units are "incidents," such as the loss of the telephone system at Zurich ATCC. A "component failure accident" is one that occurs below the system level, in which the sequential causative linkages were known and anticipated. A system accident such as Überlingen involves an "unanticipated interaction of multiple failures" (p. 70). The interaction of events is unanticipated because the units of the system are "tightly coupled" (defined subsequently); we say that the system's behavior (in a particular time period) is "highly interactive" (p. 11). The purpose of NAT is primarily to identify *interactive complexity* in a system's potential behavior and thus to understand the operational risks.

These risks and the resulting system accidents are normal because it is an "inherent property of the system to occasionally experience this interaction" (p. 6). As Perrow puts it simply, "nothing is perfect" (p. 356):

> [T]wo or more failures, none of them devastating in themselves in isolation, come together in unexpected ways and defeat the safety devices—the definition of a 'normal accident' or system accident. If the system is also tightly coupled, these failures can caused faster than any safety device or operator can cope with them, or they can be incomprehensible to those responsible for doing the coping. If the accident brings down a significant part of the system, and the system has catastrophic potential, we will have a catastrophe. That, in brief, is Normal Accident Theory. (p. 356-7)

Perrow defines a "catastrophe" as a system accident that "kills more than 100 people with one blow" (p. 357). Although the loss of lives at Überlingen fell somewhat short of this definition, the destruction of the two planes and subsequent murder of the Zurich ATCO would certainly count as a catastrophic accident.

Perrow emphasizes that the danger of such accidents is always inherent in an interactively complex system: "We need to have just the right combination of conditions in the response system and the surrounding environment for the catastrophic potential to be realized" (p. 357). In this respect, Perrow argues against "high reliability theory," which claims that system designs can guarantee safety. In particular, human-automation systems are intrinsically open systems—all possible states and behaviors cannot be known in advance—and thus interactions among Aircraft, environment, and human behavior cannot be completely predicted:

"We will always have accidents because of intrinsic characteristics of complex/coupled systems" (p. 369).

Perrow's conclusion that accidents are inevitable in a complex human-automation work system follows as well from the assumption that work practice (what people actually do) is not ultimately controllable. Trends and averages might be predictable, patterns will exist, as indeed the very notion of a "practice" suggests regularities and norms (expected ways of behaving). However, what any one individual does in a particular situation—that is over-constrained and time limited—could introduce an unexpected effect that interferes with what other people and/or an automated system are attempting to accomplish.

Perrow's conclusion is stark: "It follows that if systems have catastrophic potential, they should be abandoned, drastically scaled back, or drastically redesigned" (p. 369). The present project might be viewed in part as using a computer simulation to evaluate whether a system has catastrophic potential.

5.1 Relation of "Culture" to Accidents

Perrow cautions against blaming an accident on "culture" especially as a generic property of a system. He emphasizes instead that catastrophic potential arises more specifically from power differences among various actors in the system, which he exemplifies in his analysis of how the Union Carbide Corporation was responsible for the deaths of more than 4000 people in Bhopal, India, due to a toxic gas leak (p. 356-8).

Perrow accordingly criticizes Vaughan's (1996) analysis of Challenger accident because it minimizes the role of power and interests, characterizing NASA in 1986 as "a damaged organization that allowed unique production pressures to override safety concerns" (p. 379). By this perspective, the "culture" of NASA actually has multiple components, a managerial culture and an engineering culture; operations decisions are made by managers who view the engineers as instrumental to a mission, not as decision-making peers.

One might say from the perspective of a failure analysis that when power undermines technical judgment it is the organization that is "damaged." Shuttle operations during Challenger were already operating in a failure mode (managers exerting power over the engineers), even before the hardware component failures on launch occurred.

Perrow further cautions against Vaughn's "social construction of reality case" of blaming the bureaucracy for having created and sustained "a habit of normalizing deviations from safe procedures" (p. 380) One can say that acceptance of deviations as being normal happened leading to the Challenger accident, but "that interpretation minimizes the corruption of the safety culture" and effectively leads to ignoring "the extraordinary display of power that overcame the objections of the engineers who opposed the launch" (p. 381).

Crucially, the risk on launch day was "unprecedented" (as shown by Tufte's [2006] chart of the temperature compared to other launch days); the engineers "fumbled" in making their case and were coerced to "put on their managerial hats." "Upper management enforced" a particular "cultural script" (p. 380). That is, the "rules were deployed strategically"—the managers were not unthinking robots playing out cultural scripts (p. 380). What occurred was "the exercise of organizational power" (p. 380). The engineers were asked to abandon their engineering knowledge and judgment, and view the problem in terms of schedule and political risk. The same power manipulation between managers and engineers played out again during the Columbia accident (German et al. 2003).

Perrow concludes, "We miss a great deal when we substitute culture for power" (p. 380). For this reason, the stance adopted in this report is not to speak about a diffuse "culture" but to consider specifically roles, responsibility, and opportunity to act. We find that this perspective is perhaps applicable to understanding the Zurich ATCO's reaction to maintenance disruption of his work environment (see Section 5.3).

However, we do find Vaughan's concept of the "normalization of deviance" valuable and relevant to the overall analysis of the Überlingen accident. Normalization of deviance is a process by which a deviation from technical or procedural standards is seen in a series of cases not to cause a problem, and therefore becomes part of informally or even formally accepted action within a group. Warnings are misinterpreted as the historical context becomes a justification for its own continuation. In Vaughan's (1996) description:

> Behavior the work group first identified as a technical deviation was subsequently reinterpreted as within the norm for acceptable joint performance, then finally officially labeled an acceptable risk. They redefined evidence that deviated from an acceptable standard so that it *became* the standard. Once this first challenge to field joint integrity was resolved, management's definition of the seriousness of the problem *and* the method of responding to problems with the SRB joints to the next incident when signals of potential danger again challenged the prevailing construction of risk. Risk had to be renegotiated. The past – past problem definition, past method of responding to the problem – became part of the social context of decision making. (p 65.)

This process is explicitly described in the BFU report, in its discussion of "single man operations" in the air traffic control center, during periods of low traffic volume, and is later identified as one of the systemic causes of the accident:

> After the sectorisation work had started and the air traffic volume had decreased one of the controllers retired to rest in the lounge. Normally he would have returned to the control room early in the morning when air traffic increases, unless unusual circumstances would require his presence earlier. The spatial distance between the lounges and the control room prevents a quick alert of the second controller in

> conjunction with an immediate appearance. Thus the remaining radar controller had to assume the tasks of the radar planning controller (RP) and the radar executive controller (RE) and if necessary the tasks of the supervisor (DL) at the same time.
>
> Officially this procedure did not exist, but had been in practise at ACC Zurich for many years. This arrangement made the night shifts for the controllers more comfortable. This is a way of proceeding which does not provide any redundancy of human resources so that procedural errors, wrong distributions of attention or the omission of important actions may lead to hazardous situations as nobody is there to notice these mistakes and to take corrective actions. It follows that the breaks prescribed could not be taken. **Even though it was an unofficial procedure it was known to and tolerated by the management**. (BFU Report, p 75)

The concept of the normalization of deviance furnishes a precise identification of the process by which a risky unofficial procedure is known and tolerated by management. Rather than saying individuals engaged in "misconduct," Vaughn (2003) emphasizes that the accident was due to systemic, institutional, organizational, and political problems. Such problems are cultural because behaviors are reproduced by organizational structures and practices that are independent of the people involved. Hence, Vaughn correctly predicted in her analysis of Challenger that the same "organizational culture" would create a future accident—she went on to make significant contributions to the Columbia Accident Investigation Board.

The normalization of deviance in Zurich was manifest in how single man operations (SMOP) was allowed by skyguide,[11] the private company that provides air navigation services in Switzerland, during day operations, despite disapproval by regulatory authorities (p. 93). The rules of the procedure were incrementally ignored and reinterpreted to allow the practices in effect during the Überlingen accident.

In particular, the night shift was never approved for SMOP, but it was followed informally. SMOP required the supervisor (DL) to be present with one controller. Effectively during the night shift, the supervisor was replaced by a third ATCO (allowing one to sleep and the other to serve as DL)—an example of "deviance" accepted in practice. The deviation from original standards was taken to an extreme when the third ATCO was eliminated:

> The practice of rostering only two ATCOs had developed because of the personnel situation. The former system had scheduled three controllers for the night shift. It ensured that two controllers were always at their workstations and the third took a break. That this controller took a longer break during times of low traffic became unofficial practice. This practice was maintained as the night shift was reduced to two controllers. (BFU Report, p. 92)

[11] The company name, skyguide, is officially written in lower case.

On the night of the incident, the toleration of deviation went one step further: SMOP was only allowed if radar and alert systems were fully operational. At this point, practices had so degraded that none of the requirements for SMOP were met—it was not only not officially allowed, the normalization of deviance had permitted its rules to be entirely disregarded. In effect, the meaning of SMOP was now lost and the phrase was taken literally—only one person was monitoring the airspace.

5.2 Unseen and/or Unbelievable Interactions

NAT concerns systems in which "interactions are not only unexpected, but are incomprehensible for some critical period of time…saying he should have zigged instead of zagged is possible only after the fact" (p. 7). That is, an experienced person on the job is confronted with a particular work configuration (e.g., aircraft, resources) in a dynamic, time-sensitive sequence that the work system designers, trainers, and hence worker did not expect to occur, and the nature of the system interactions occur are too complex to comprehend on the spot in time to prevent an accident.

The Überlingen accident proved that the design of TCAS, introduced as a major safety device, makes possible unexpected and incomprehensible interactions. The accident illustrates that automation that was intended to increase redundancy does not necessarily or as a matter of course make a situation safer. According to Perrow, a safety device, "since it is often added after problems are recognized, too frequently creates unanticipated interactions with distant parts of the system that designers find it hard to anticipate" (p. 368).

In particular, it was not anticipated (or was ignored by designers, certifiers, managers, and trainers) that interventions of TCAS and the ATCO might overlap such that the ATCO would speak between the TCAS TA and RA, and his instruction to climb or descend would be followed; in which case following TCAS would then require reversing the aircraft's direction within seconds. In this respect, pilots might be confronted not only with an authority issue ("who is in charge?") but a physical (and perhaps psychophysiological) control issue—an alternative action has already begun, the timing might be too tight to reverse mentally and physically in maneuvering the aircraft. This report analyzes such timing and interaction issues in detail.

5.3 How Situations are Allowed to Become Complex

Situations can become more complex if people make decisions without appreciating interactions or constraints already in play (e.g., focusing on the late-arriving AEF flight without realizing the loss of STCA optical would require more careful monitoring of the wider sector) or existing processes that will be affected by an action (e.g., clearing DHL to fly at the same altitude as the BTC). People may make comfortable, familiar choices that reduce their range for future action and therefore increase the probability of an accident.

In general terms, Perrow gives an example of receiving an order from your supervisor (it could be a written directive or standing protocol), which later turns out to be ambiguous—should you do A or B? (p. 27) In particular, consider the supervisor in the Zurich ATCC who left early for the night and thus implicitly made the ATCOs responsible for supervisory overview of the center. Later, with one ATCO on break, the lone Zurich ATCO has to carry out multiple roles. He is soon confronted with unexpected maintenance work (he had not read the memo in the break room and it did not mention the loss of the STCA optical alert; BFU Report, p. 74, 89). What should he do? Suppose that action "A" would be to refuse permission for the maintenance work (as a supervisor might do) or call back the other ATCO to assist. Action "B" would be to carry on, to simply accept what the maintenance work entailed—after all, the work will be complete in a half hour or less.

Perrow suggests that these two choices are general ways of handling an unexpected situation:

- "A would be correct if something were terribly wrong or if the situation were quite unusual" (e.g., the ATCO might have concluded this if the maintenance team had told him what systems would be affected)
- "B would be correct if it were a situation that had occurred a few times before and was not all that serious" (e.g., perhaps the Zurich ATCO was familiar with maintenance work on Sunday evenings).

If B "has been used before, and it is easy to carry out…" (p. 27), ATCO proceeds and things happen as they should. This does not prove that B is a more correct action than A, yet it reinforces his decision and how he has conceptually framed the situation: "you are creating a world that is congruent with your interpretation, even though it is the wrong world. It may be too late before you find that out" (p. 28). That is, ATCO's interpretation now becomes part of his understanding of the current work context.

In this case, the Zurich ATCO has convinced himself that he can handle the traffic with the existing resources. He didn't realize that he had allowed the phones and optical STCA to be disabled, both of which would be essential for handling a late arriving flight when two other planes were on collision course. He did not reflect as a supervisor might; he did not ask questions to be sure he understood the potential effects on his equipment. He accepted that it was normal to have the supervisor and second ATCO absent and for maintenance to be done on a Sunday evening. Yet each change to the system he allowed was transforming the air traffic system that evening into a complex system, a configuration of controllers, flights, radar, TCAS, pilots, etc. that would have complex interactions as events unfolded just a few minutes later.

Woods (2005, p. 297) suggests that accidents exhibit a "classic drift toward failure… as production pressures and change erode the defenses that normally keep failure at a distance." At Überlingen, the gradual unknown loss of tools and disabling of taken-

for-granted methods and alerts, unexpectedly and apparently abruptly brought the ATCO to a complex situation: maintenance was viewed as normal rather than a process of introducing anomalies in the operational system; there was apparently no cross-check of the effects of maintenance (who was responsible for verifying that the "accepted" risks of the backup systems were in fact acceptable?); the background of prior success with SMOP led to organizational complacency that "hazards were not present" (cf. Woods on Columbia, p. 293); this ATCO's close-call the prior year was a clear warning of the accident yet to come, but the changes to the radar display led to the false belief that vulnerabilities of his operating alone had been resolved, with the systemic issues obviously poorly modeled in management's understanding of the risks.

The Zurich ATCO might have categorized the system and operations that night in July 2002 as *normal*, but the constraints that would eventually make human actions ineffective and force an accident were accumulating. Step by step, the ATCC allowed resilience to be removed from the system—the manager departs, the second ATCO goes on break, the maintenance begins. And the consequences of these events reveal more broadly the systemic failure of the organization—failures to learn from past mistakes, to follow staffing standards, to assign cross-checking roles, to manage safety critical situations.

5.4 Definition of "Complex Interaction"

Perrow's NAT is particularly valuable for defining the kinds of system interactions that will inevitably lead to accidents.

As indicated above, a *complex interaction* is one that is unintended or intended but unfamiliar (p. 77). For example, a TCAS RA reversing an ATCO instruction was perhaps unintended by the certifiers of the system's safety; this sequence was definitely unfamiliar to the Zurich ATCO. More generally, a "component can interact with one or more other components outside of the normal production sequence, either by design or not" (p. 78).

More simply, the interactions are said to be *complex* because they occur in an unexpected sequence (p. 78). That is, the *unfolding sequence of events* is complex; complexity is not a property per se of a system (p. 8), but of the relations among the events over time. If the interactions are linear, such that all causal relations in the system's behavior can be anticipated (pre-enumerated), by definition unexpected sequences of behavior will not occur. This means that complexity is not an abstract property of components, procedures, etc. in isolation, but rather characterizes how roles, tools, practices, environment, etc. interact to create unanticipated *sequences* of events—it is a property of the system's behavior, of what we often call "situations"—a dynamic configuration of objects, people, processes in action.

A situation is not a *state*, but a flow of events—"situation awareness" refers to a person's perceptions and conceptions in time (Clancey, 1999). Accordingly, a person's interpretation is temporal. The ATCO's conception of "what is happening

now" includes his experienced past, reinterpretations of the present, and anticipations of the future. This is what Perrow means by "creating a world that is congruent with your interpretation."

In particular, the Überlingen TCAS–Pilot–ATCO interactions became complex because of the timing of events—the flight paths of the planes on collision course, the delayed flight coming into Friedrichshafen, the irregular monitoring of the sector by the Zurich ATCO, the dysfunctional phones, the missing STCA Optical alert, the instruction to the BTC by TCAS, the instruction by TCAS to the DHL, and so on.

With one ATCO on break during the maintenance work, the system of controllers and their tools had lost redundancy—there were fewer eyes on the radar screens and less data and alerts displayed. The lone ATCO had to do more tasks and therefore work more quickly, with less support. The interactions among this lone ATC, the pilots onboard three aircraft, and TCAS—how their behaviors became part of the operating environment for each other—were now complex. More specifically, the system's interacting processes (people, aircraft, automated systems) were complex because they had become *tightly coupled*. As detailed subsequently, even the 12 second sweep delay of the radar was affecting what the sequence of events.

5.5 Tight Coupling

Perrow defines a *tightly coupled system* as having time-dependent processes that "cannot wait or stand by until attended to" (p. 92). There is "no slack or buffer or give" between processes, "what happens in one directly affects what happens in the other" (p. 90). In a tightly coupled system, parts of the system (people and automated subsystems) must behave in response to what other parts are doing—there is little or no flexibility. If A occurs then B must occur (and usually soon) to satisfy the operating requirements. For example, once the DHL aircraft was on a TCAS RA descent, it was necessary for the BTC pilots to *immediately reverse* the Zurich ATC's instruction and follow the command of the TCAS RA to climb.

A dynamic system (such as a configuration of flights, controllers, and automated subsystems) becomes tightly coupled when the course of acceptable actions and the time to act become more limited. As a system becomes tightly coupled, dependent interactions increase and time to respond appropriately (e.g., to avoid a safety violation) decreases. In air traffic control systems, tight coupling develops when redundancy is lost—in particular removing actors (people and/or subsystems) decreases flexibility as fewer actions become possible in a given time (e.g., less frequent monitoring and redirection of aircraft), thus increasing time-dependencies. The interactions in the system are then complex.

By contrast, loose coupling "allows certain parts of the system to express themselves according to their own logic or interests" (p. 92); loose coupling "allows recovery" (p. 160). "The sequences in tightly coupled systems are more invariant" (p. 93); if A occurs then B must follow.

It is important to understand that "tight coupling" in the context of the air transportation system pertains to a system comprised of controllers, flights, and automated subsystems *during operations*: it characterizes a developing situation, the actual or potential effects a configuration of subsystems and processes have on each other as events unfold.

If certain behaviors do not occur or occur at a different time, a system might remain loosely coupled. For example, if the Zurich ATCO had received an optical STCA alert, he might have intervened before the time of the TCAS TA, and the sequence of his speaking between a TCAS TA and RA would not have occurred. Instead, his intervention and the TCAS RA both occurred, and both instructions had to be processed by the BTC pilots. Or with another Zurich ATCO on duty, the paths of the DHL and BTC would very likely have been noticed and modified much earlier, eliminating interactions among ATC, TCAS, and pilots from the sequence of events that night.

The Überlingen accident involved at least three interacting processes that were impinging or encroaching on each other, such that what was happening in one process was now affecting another (a tight coupling):

1. Handoff of AEF delayed flight by Zurich ATCO (involving attempted communication with Friedrichshafen tower via the disabled phone system)
2. Zurich ATCO observing flight paths on radar and instructing BTC to descend
3. DHL pilots interacting with TCAS.

The analysis of a complex situation in terms of tight coupling reveals that temporal relations will be pivotal in modeling and simulating the Überlingen events and verifying system models. The analysis also suggests that the collision would not have occurred if TCAS didn't exist, which is one of several "what if" scenarios that Brahms-GÜM simulation experiments examine.

5.6 Cognitive Complexity

To summarize, what appear to be ordinary circumstances can change the configuration of the work system and environment causing a loss of redundancy in operation processes (how the work is done becomes less flexible). Interactions that emerge among people and systems become tightly coupled as choices and timing are constrained. The workload increases as additional tasks become necessary and/or are performed more frequently and quickly. Effort might need to be devoted to finding workarounds (e.g., alternative ways of communicating or getting information). More work might need to be done manually, which is slower and reduces attention to simultaneous responsibilities; situation awareness diminishes, critical tasks are not undertaken and events are not noticed (Dismukes et al. 2001). The work has become *cognitively complex*, such that the people are no longer managing events and become enmeshed in how the systems themselves are interacting. The probability increases of taking hasty actions infused with fear that make situations worse, causing or contributing to accidents.

Cognitive complexity is a relation between knowledge/skills (familiarity) and a situation comprised of time-dependent tasks and resources. In general, more work must be done more quickly with less information and assistance and fewer tools. As the system becomes tightly coupled, tasks shift from being a loosely unordered sequence of routine actions (redirecting aircraft) to a rapid sequence of finely reasoned actions that are necessary to avoid a catastrophe. When only one solution is possible (e.g., ATCO must advise BTC to climb rather than descend or not intervene at all), the lack of information and time will greatly increase the probability of doing the wrong thing.

During the Überlingen sequence of events the workload became more cognitively complex at each step: the second Zurich ATCO left requiring the lone ATCO to manage two workstations; the radar data was reduced, the phones disabled, and the optical STCA disabled—each reducing or preventing the flow of information in the work system; then the AEF arrived on scene, imposing a task that required full attention.

The presence of automation such as the optical STCA or another person's assistance would have reduced the cognitive complexity experienced by the Zurich ATCO. The workload was excessive because of lost resources; during a period of diminished situation awareness, there was too much to do simultaneously. Even the highest priority task, avoiding a collision, was not perceived until TCAS was engaged and—also unknown the Zurich ATC—was redirecting the aircraft for which ATCO was responsible.

Interactions are most commonly understood as physically causal, as for example two aircraft in a certain proximity will trigger a TCAS TA. Events may also be conceived by a person as requiring certain actions in a certain time frame, such that the relation of the events and attempted actions is *conceptual*, that is, a person's understanding. Thus as the AEF arrived late for landing at Friedrichshafen, the Zurich ATCO understood that he needed to carry out the normal handoff procedure by calling the Friedrichshafen airport control tower. The phones were down and he became fixated on calling, seeking an alternative number and spending too much time dealing with this situation. In fact, calling Friedrichshafen immediately was not his only option; in actuality, there was not a tight coupling between the plane's arrival at a certain time and ATCO's executing the handoff. The urgency and priority of the handoff, and even handling the landing by serving as handoff mediator, were a conceptual point of view that the Zurich ATCO adopted unnecessarily. In effect, he placed himself in a box of his own making.

The Zurich ATCO failed to exploit a safety feature within his repertoire (cf. p. 96, p. 161), namely to place the AEF in a holding pattern, releasing the handoff temporarily from his attention and thus buying time to find another solution. The tight coupling between the AEF's flight path and the ATC's phone-related actions was mental, a matter of personal judgment, than either a physical necessity or a

standard procedure imposed by his role. The problem was mostly in his mind. The system offered flexibility that was not exploited, illustrating how a failure might be cognitive, and the system would become in fact more tightly coupled from the resulting actions (e.g., the DHL and BTC aircraft were allowed to approach dangerously close, reducing his subsequent time to avoid a collision).

Backing up at bit we might try to explain the ATCO's fixation. The Zurich ATCO's focus on calling Friedrichshafen (which he attempted 3 times within 7 1/2 minutes, on the line 56 seconds total[12]), originated in the combination of the loss of regular and backup phones (for 11 ½ minutes, BFU Report, p. 17), a late-arriving plane, and working alone. Thus the loss of redundancy in the system to that point, when he discovered the bypass phones were down, presented overall an unfamiliar situation. The loss of redundant means of accomplishing the handoff could explain the Zurich ATC's fixation. The event of a late-arriving plane followed by discovering he was unable to communicate with the tower might have presented itself as one coupled event. The urgency to resolve the one problem (phone call) was compounded by urgency to resolve the other (handoff), raising the controller's anxiety, narrowing his attention further.

In effect, "handle arriving AEF" became identical in ATCO's mind with "make a phone call"—and it must be remembered that during these 7 ½ minutes when ATCO was on the phones for 56 seconds, BTC reported arrival in the Zurich sector, AEF called in twice pressuring ATCO to manage their arrival in Friedrichshafen, and two other flights (THA933 and MON5621) required handoff to other sectors.

The Zurich ATCO was perhaps also implicitly still relying on redundancy offered by the STCA optical alert, which unknown to him was not enabled —"safety devices contribute to complacency and inattention" (p. 153). The records do not indicate whether the Zurich ATCO ever relied on the STCA to call his attention to traffic that he was not otherwise monitoring. This is the kind of information an ethnographic study of ATCC work practices or a study of prior failure reports might reveal.

5.7 Resilience

Resilience of a work system concerns detection and recovery from problematic situations. By analogy to Hutchins' (1995) question, "How does the cockpit remember its speed?" we can ask, "How does the air traffic system detect and recover from problematic situations?" As throughout, in this analysis we are focusing on *interactive behaviors of the system*, which may involve any combination of human and automated actions.

Considering the resilience of the air traffic system leads us to consider how the work system is designed with checks and balances to distribute/share/activate attention

[12] The BFU Report states that seven calls were attempted (p. 7), but the ANSA transcript indicates only three. Based on timing, it is possible ATCO dialed multiple times during the second and third attempts.

and responsibility for action. The Überlingen work system became complex because of subsystems failing and/or being absent. The system was not resilient because *emerging problems were not detecting early enough to resolve easily* (leading to a cognitively complex situation for the ATCO) and *recovery from these problematic (unsafe) situations was mismanaged* (leading to the collision).

Errors are caused not only by individuals (lack of knowledge, "lapses" caused by workload/stress/fatigue), but also by the work system lacking checks and balances. Section 6.7 considers checks and balances that were missing or not executed properly during the Überlingen events by asking what could have happened differently.

5.8 Complexity and Coupling of Airways

Perrow (1984) analyzes six systems involving high-risk technologies: nuclear power, petrochemical plants, aircraft and airways, marine shipping, earthbound systems (e.g., dams), and "exotics" (space, weapons, DNA). His appraisal of aircraft/airways is ambiguous, suggesting that the ATS is relatively high in complexity but this complexity "will respond to [a] considerable extent, though not completely, by management and technological innovations" (p. 168).

On the one hand, he says the airways are "neither very tightly coupled nor complexly interactive" (p. 159). Yet he says the "airways system is high on interactive complexity and on tight coupling" (p. 168). In other words, he concludes that complexity is high, but not extreme—the ATS is not *very* tightly coupled. He says that with the introduction of transponder automation to reveal aircraft information on radars, "time constraints are still tight; the system is not loosely coupled, only moderately tightly coupled" (p. 160). This is because "delays are possible; aircraft are highly maneuverable and in three-dimensional space, so an airplane can be told to hold a pattern, to change course, slow down…" (p. 160). In other words, the system is resilient; pilots and controllers have time and space to adapt to manage separation.

In effect, Perrow is attempting to explain why the ATS is so safe as measured by loss of life. Reasoning backwards to fit the NAT model, it must be the case that the airways system is not complexly interactive and tightly coupled in general—reduction in complexity and coupling has resulted "in about as error-free a large system as we are likely to see in our society" (p. 168). He says, "if there ever was a safe system that was complex and coupled, it is this one" (p. 382).

Perrow emphasizes that reported near misses indicate the system's flexibility in recovery (compared to nuclear power plants) and "near misses reported to be under 100 feet are exceedingly rare and the proximity may be exaggerated" (p. 161).

He later says in the book's Afterword (published in 1999) that ATS safety may be resulting from the frequency of trials (flights) that has created a large database with rare events, which has promoted improved technology and training (p. 382). Perrow

also acknowledges minimizing the role of ATCs in his initial analysis, that 15 years later he believes that their role is "primarily…to pack as much traffic into airports as they could" (p. 383).

One way of understanding Perrow's appraisal of the ATS is that he is speaking in general, abstract terms about the complexity of the system at large, not specific situations that may occur within this system in a particular time and place. Thus the Überlingen ATS configuration on that Sunday evening in 2002 was atypical for the location and time—the system became complexly interactive and tightly coupled because of loss of operations redundancy and unusual flight circumstances.

Exemplifying this point, Perrow states that the SNA (John Wayne Airport, Orange County, CA) system configuration is "quite tightly coupled" (p. 152) because a small airport handles many private and commercial planes, making it complexly interactive. So in general (by this analysis) this airport has less operations flexibility and the traffic is more demanding for people and systems to manage safely than other airports. In simple terms, the everyday, routine work is more difficult.

In a cautionary conclusion relevant to NextGen, Perrow states (in the Afterword published in the 1999 reissue of the 1984 original edition): "The FAA is pressing for more automation in its system, thereby reducing the number of controllers extensively. Both of these…will lead to much tighter coupling—that is to less resources to recover from incidents" (p. 161). While TCAS exemplifies how automation can increase safety, the Überlingen collision also illustrates how the *absence* of routinely relied on automation (STCA Optical alert) and technology in general (the phones) can also increase the complexity of the system.

5.9 Theoretical Justification for Developing a Series of Models

In attempting to explain why the ATS is relatively safe, Perrow points out that one must consider the task at hand: "If one tried, it would be hard to make two aircraft collide" (p. 161). The Brahms-GÜM simulation of the Überlingen scenario demonstrates that point (see Chapter 10).

In general, evaluating a work system design requires knowing the context of particular work environments. As explained above, complexity of the work is a dynamic relation, it will vary as configurations of aircraft, weather, automated systems, workers playing different roles, and tools change. Complexity is not a property located in some part of the system, but about evolving operations from the perspective of agents (people and automated subsystems) with specific goals to affect the system's behavior.

Perrow's discussion of SNA (John Wayne Airport, Orange County, CA) illustrates that complexity varies with airport, time of day, season, weather, etc. It is in the combination of these specific factors that interactive complexity and tight coupling arise. Consequently "what if" simulation of different specific configurations is important for evaluating what kinds of interactions might arise that affect safety.

The effect of losing redundancy is of central concern in creating and experimenting with work practice simulations. Perrow's analysis reveals that redundancy for communicating information and taking action increases flexibility—choice and timing of actions, thus reducing interactions and coupling among them. The ability to cope with loss of redundancy is an aspect of the system's resilience.

Consequently, we concluded that the Brahms simulation should be designed not to replicate a single scenario, but rather the model should comprise a family or space of scenarios that enables systematically evaluating the effects of loss of redundancy. Evaluating safety properties such as "separation assurance" requires evaluating how air traffic control interactions and coupling are affected by loss of redundancy in the context of different, specific air traffic configurations (e.g., late arriving plane). The Brahms–GÜM has been designed and developed accordingly (described in Chapter 8). But first in the next chapters we explain in Chapter 6 more about the *content* of what needs to be modeled based on failure analyses of the collision, that is the objects, properties, and events relevant to the incident, and then in Chapter 7 we explain more about the *modeling framework* that is applied for representing this work system and its alternative configurations (scenarios).

6 Überlingen Collision: Systemic Failure Analysis

As explained in the previous chapter, following NAT there is no "root cause" or simple tree of events leading to the collision. Rather a sequence of events, some of which were causally related, reduced flexibility in the local ATS, causing interactions among people, aircraft, tools, and automated systems to become more tightly coupled, to the point that the system was out of control. The goals, states, and behaviors of the agents—pilots, controllers, and TCAS—became unknown to each other. As one commentator put it, "If Überlingen demonstrated anything, it's that nobody has to time to sort it out much less come up with a new plan and communicate it."[13] Synchronization required for safety was lost; their combined actions of maneuvering and guiding implicitly led the planes to collide.

This chapter reviews different analyses of the collision, providing a background reference for the subsequent presentation of the Brahms modeling framework and Brahms-GÜM simulation. The analyses include the official conclusions of the BFU Report, commentary on it by the ANSA AirRadio organization, and deviating positions published by the Kingdom of Bahrain[14], Switzerland, and Russian Federation. Three independent analyses are also presented by Aviation Knowledge, Aviation Safety Network, and an independent academic research report funded by Eurocontrol. The excerpts provided here illustrate the different levels of details (immediate and systemic causes), many decisions (e.g., directions given to the DHL flight) and intricate timings involved (e.g., a claim that the instruction to the BTC was already too late to ensure separation; the BFU Report [p. 109] states that separation was 7 nm during the last second of ATCO's first intervention at 21:34:56 hrs, which constitutes an infringement).

Subsequently, the factors presented in these reports are organized using the Human Factors Analysis and Classification of Causal Factors scheme (Section 6.5) and then re-presented using a form of causal tree analysis (Section 6.6), which is used to bring out alternative scenarios that might have occurred (Section 6.7). Finally, the interactions articulated in the analyses are considered from the perspective of the affect of timing of events causing, inhibiting, or requiring interactions among people and systems (Section 6.8).

Following the Normal Accident Theory framework and the specific analyses and commentary presented here, we conclude that the attempt to assign a single cause to the accident is fruitless, that the observed sequence of events occurred through complex, systemic interactions in the work system on that night, manifesting failures in organizational policies, staffing, supervision, air traffic control monitoring and collaboration in the center, pilot knowledge and training, and the design and certification of TCAS.

---

[13] PPRuNE Forums, http://www.pprune.org/archive/index.php/t-343376.html, accessed 26 Sept 2012, "ATCO Issues: TCAS or ATCO priority? Re. DHL 757 midair and TU-154"

[14] DHL International Aviation is based in the Kingdom of Bahrain.

6.1 Official Report by the German Federal Bureau of Aircraft Accidents

The primary source for developing Brahms-GÜM is the 2004 German Federal Bureau of Aircraft Accidents Investigation Report ("Bundesstelle für Flugunfalluntersuchung Report"). The BFU Report's conclusions (p. 110) are:

> The following immediate causes have been identified:
> - The imminent separation infringement was not noticed by ATCO in time. The instruction for the TU154M to descend was given at a time when the prescribed separation to the B757- 200 could not be ensured anymore.
> - The TU154M crew followed the ATCO instruction to descend and continued to do so even after TCAS advised them to climb. This maneuver was performed contrary to the generated TCAS RA.
>
> The following systemic causes have been identified:
> - The integration of ACAS/TCAS II into the system aviation was insufficient and did not correspond in all points with the system philosophy. The regulations concerning ACAS/TCAS published by ICAO and as a result the regulations of national aviation authorities, operational and procedural instructions of the TCAS manufacturer and the operators were not standardised, incomplete and partially contradictory.
> - Management and quality assurance of the air navigation service company did not ensure that during the night all open workstations were continuously staffed by controllers.
> - Management and quality assurance of the air navigation service company tolerated for years that during times of low traffic flow at night only one controller worked and the other one retired to rest.

6.1.1 Kingdom of Bahrain deviating position

The BFU report summarizes that "The Kingdom of Bahrain is of the opinion that the results of the Human Factors group shall have been made the sole basis for the analysis," and then proceeds to incorporate the deviating interpretations (BFU Report, Appendix 10, pp. 1-2):

> The second systemic cause should be expanded incorporating the findings from the HF Group report on the failure to assess the risks on the particular night, mitigate against them by manning both positions the whole night, briefing all staff appropriately, delegating responsibilities and effective training. Training does not necessarily mean TRM/CCC Training, but rather ensuring that the ATCOs understand and practice (simulate) operations in "radar fall-back mode". This should have been an essential element of their emergency/refresher training.
>
> The third systemic cause should also be expanded. How could management possibly tolerate a single controller working at night at 'low' traffic level, when such operation did not conform to SMOP's criteria? It also raises a question on how does one define 'low' traffic – three aircrafts on 01 July 2002 demanded a great deal of attention even notwithstanding the temporary radar and telephone shortcomings?

6.1.2 ANSA AirRadio commentary

The International Advisory Group Air Navigation Services (ANSA) and Aeronautical Radio & Air Traffic Control Advisors (AirRadio) (ANSA+AIRADIO 2004, p. 36) analysis is included here because it is quite different from the BFU Report's conclusions. Rather than focusing on the BTC crew response to TCAS in responding the imminent collision, this analysis focuses on the ATCO's route approvals that caused the planes to be put on a collision path (italic emphasis added):

> The development of the conflict situation between DHX 611 and BTC 2937 and the resulting collision could have easily been avoided, if the air traffic controller had properly *identified flight BTC 2937 and in application of vertical separation let it climb to flight level 370*, or alternatively hereto *had DHX 611 climb to flight level 350 only* instead of 360. This easy solution was however not chosen, since he apparently did not recognize the given situation or his *other distracting duties (arrival to Friedrichshafen) demanded too much of him*.
>
> On the other hand *it would have been more than advisable to issue a corresponding traffic information to flight DHX 611 at the time of issuance of the descent clearance for BTC 2937 to flight level 350*, or even better, *to turn flight DHX 611 away from its course and have it climb further, since it was previously being identified by himself.*
>
> ***For this commentator the air traffic controller acted in gross negligence and demonstrated lacking professional competence.*** [emphasis in original].

6.1.3 Russian Federation deviating position

In an appendix of the BFU Report (Appendix 10, p. 2), the Russian Federation provided a deviation opinion that like the ANSA AirRadio Commentary criticizes the Zurich ATCO's instruction to the BTC pilots, claiming that the traffic information he provided was incorrect and also that the DHL crew had sufficient information to avoid the conflict (presuming that they had ignored the TCAS RA):

- The TU154M crew followed the ATCO instruction to descend and continued to do so even after TCAS advised them to climb. This maneuver was performed contradictory to the generated TCAS RA.
- The crew was unable to follow TCAS RA as by that time they were at 35 500 feet and the controller informed them about conflicting traffic above, at FL 360.
- The false ATCO's information on the direction towards the conflicting traffic (2 o'clock instead of actual 10 o'clock) and contradictory ATCO and TCAS instructions did not contribute to the correct decision of the crew as well.
- The B757-200 crew who were at the same frequency and heard three ATCO instructions to descend, as well as the readback of the TU154M crew about leaving FL 360, had a real possibility to avoid collision.

6.1.4 Switzerland deviating position

The deviating position filed by Switzerland focuses on the descent through FL 350 by the TU154M following the ACC Zurich instruction as the cause of the accident (BFU Report, Appendix 10, p. 2):

3.1 Findings
Accident:

- When the TU154M, contrary to the instruction of the ATC, was descending through flight level 350, the airplane's rate of descent was approximately 1900 ft/min.

ACAS/TCAS:

- The simulation and the analysis of the alert sequence showed that the initial RA's would have ensured a safe vertical separation of both airplanes if both crews had followed the instructions accurately.

3.2 Causes (3. immediate cause)

- When reaching flight level 350, the rate of descent of the TU154M was still approximately 1900 ft/min. Subsequently the crew of the TU154M descended below the flight level assigned by the air traffic control unit.

6.2 "Aviation Knowledge" Analysis

The web site Aviation Knowledge[15] suggests that fault mainly lies with the Zurich ATCC skyguide management, following the systemic causes cited in the BFU Report, adding the fact that four skyguide middle managers were prosecuted for negligent homicide.

The details of this analysis are provided as reference for the discussion about the Brahms model, which includes most of the facts stated here. Readers interested in understanding the circumstances of the collision will find this summary of value; others can skip ahead.

Background information

The Tu-154M was a charter, being operated by Bashkirian Airlines as Flight 2937 and en route to Barcelona from Moscow, while the 757-200PF was a DHL freighter, Flight 611, flying from Bergamo, Italy, to Brussels. At the time of the accident, the aircraft were in controlled airspace being guided by the private Swiss air traffic control company, skyguide.

- **Single controller**
  There were two skyguide controllers on duty, but only one was working, while the other was on break for a significant period of time.
- **Disabled conflict detection system**
  Procedures required a conflict detection system to be operational while only one controller is working, however due to maintenance taking place at the time, it was disabled.
- **Downgraded radar**
  Furthermore, the maintenance work also meant that the radar systems were operating in a downgraded mode, which was less responsive and accurate [it did not automatically identify the aircraft and range scale bar introduced in response to May 2001 separation violation by this same ATCO was unavailable, BFU Report, p. 82].

[15] http://aviationknowledge.wikidot.com/asi:bashkirian-airlines-flight-2937-dhl-flight-611:mid-air-c

- **Disabled phone line**
  The phone line was also disabled, meaning the controller wasted a significant amount of time trying to contact a local German ATCO unit about another aircraft.
- **Unoperational backup phone system**
  Unfortunately, the backup phone system was not operational either, due to a software failure that was not detected even when tests were run on the ATCO system three months previously.
- **Two discrete frequencies**
  A final consideration was that the controller was working two different radar screens with discrete frequencies. This meant that he had divided attention, and could only deal with one screen and frequency at a time.
- **Collision course established**
  At 21:21:50 UTC, the Boeing's crew requested a climb from FL320 to FL 360. Eight minutes later, after this had been confirmed by ATCO and performed by the crew, the two aircraft were on a collision course at the same altitude.
- **Initial TCAS advisory**
  About five minutes after this, at 21:34:42, and 50 seconds before the collision, both crews received an initial TCAS traffic advisory: “traffic, traffic”, warning of the possible collision with the other aircraft. A few seconds later, the controller instructed the Tupolev's crew to descend expeditiously to avoid the Boeing, which the crew complied with but did not acknowledge.
- **Attempted sector controller warning**
  A German upper sector controller noticed the potential collision but could not warn the skyguide controller due to the disabled phone lines.
- **TCAS RA**
  Fourteen seconds after the initial TCAS traffic advisory, the Boeing's TCAS issued a resolution advisory to descend: “descend, descend”, and the Tupolev's TCAS issued a resolution advisory to climb: “climb, climb”. The Boeing's crew began to descend, while the Tupolev's crew also continued to descend, following ATC's instructions and ignoring TCAS. Shortly before the collision, the Boeing and Tupolev's crews received an “increase descent” and “increase climb” command respectively [but instruction to Tupolev crew was delayed by 11 seconds].
- **Collision**
  One second before the collision, both crews conducted drastic evasive maneuvers but it was too late to avoid the collision.

Considerations

- **Single controller duties**
  This procedure was met with great controversy when it was implemented a year before the collision, and resulted in many protests from the controllers' union. It was seen as an unsafe practice due to the lack of supervision or assistance in safety-critical situations.

- **Disabled systems**
  Skyguide's management should not have permitted maintenance to disable these key systems while there was only one controller working, and as a result of these errors, four skyguide middle managers were prosecuted for negligent homicide.
- **Controller's workload**
  Even though traffic was light, the working conditions would have put extra strain on the controller and greatly lowered the possibility of him preventing the collision.

Conclusion
This accident is an excellent example of an organisational accident, as while it does appear that the controller should be held responsible for the event, the decisions made by skyguide management made such an event almost inevitable.

6.3 Aviation Safety Network Analysis

An analysis by the Aviation Safety Network[16] assigns systemic fault to inadequate regulations in the form of operations procedures for pilots' interactions with TCAS, as well as repeating the BFU Reports criticism of skyguide management:

- The integration of ACAS/TCAS II into the system aviation was insufficient and did not correspond in all points with the system philosophy.
- The regulations concerning ACAS/TCAS published by ICAO and as a result the regulations of national aviation authorities, operational and procedural instructions of the TCAS manufacturer and the operators were not standardised, incomplete and partially contradictory.

6.4 Review of the BFU's Überlingen Accident Report

Johnson (2004b) provides a comprehensive independent analysis of the contents and presentation of the BFU investigation report, emphasizing how ATCO actions related to perceptual cues.

Johnson depicts events using a "simplified form of Events and Causal Factors [ECF] analysis, initially pioneered by the US Department of Energy" (p. 11); see Figure 6-1.

[16] http://aviation-safety.net/database/record.php?id=20020701-0

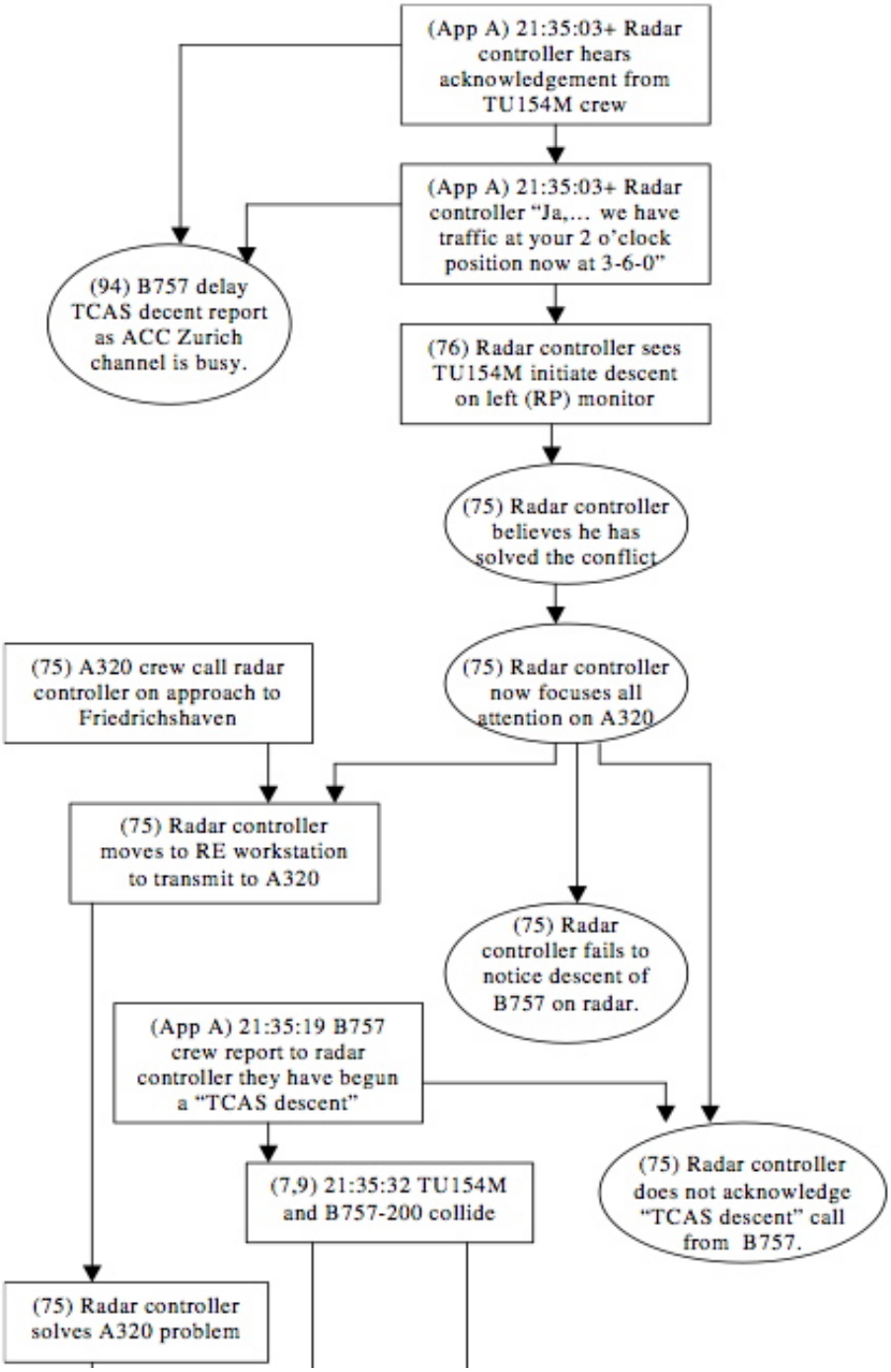


Figure 6-1: Events and Causal Factors (ECF) analysis (Johnson, 2004b, p. 18).

The remark in this diagram "Radar controller now focuses all attention on A320" illustrates the difficulty of reconstructing events. From the perspective of the other analyses, saying that he "focuses all attention" is arguable. In particular, the claim that "radar controller believes he has resolved the conflict" (which Johnson depicts as causing his shift back to AEF) is based on the claim "radar controller sees TU154M initiate descent on left (RP) monitor" (a statement made by ATCO during

the debrief, BFU Report, p. 85). An alternative assumption about the ATCO's perception and belief is that he shifted to respond to AEF because the "radar controller believes he must respond immediately to an aircraft approaching an airport." In short, claims about what the ATCO perceives or believes are difficult to justify aside from saying he is *reactive*, shifting back and forth between the workstations according to what comes to his attention. Also, a person may have multiple, simultaneous conceptual justifications for an action.

Rather than shifting "all attention" one might interpret the actions as managing two simultaneous activities according to what information allowed and timing required. Immediately after instructing the BTC to expedite descent, AEF 1135 called in again on the other workstation, so ATCO moved to respond, "ah okay, turn left heading 240, intercept ILS 24, descend four thousand feet." Indeed, the AEF 1135 is landing at precisely the time of the collision and called ATCO four times during the last minute before the collision. ATCO's lack of follow up to confirm that he had resolved the BTC/DHL separation might suggest a lack of proper prioritization of tasks. But his actions reflect as well the layout of the work stations —that a single controller on duty must move back and forth—and that to watch for the radar display to update to confirm the relative locations of the aircraft (details are reconstructed in Section 6.8.6). Hence he responded to the AEF flight immediately—the "shift in focus" reflects doing something urgent rather than just staring at the display for a few seconds. Unfortunately, it is just at this moment that the DHL calls in, "…six hundred..äh TCAS-descent," a remark that apparently ATCO did not hear.

Consistent with this example about of the difficulty of explaining what the ATCO is doing and why, Johnson notes the inadequacy of the BFU Report's Appendix 2 and 3 chronology and diagram of events for reconstructing events what occurred in the Zurich ATCC. His observations explain why developing the Brahms simulation required a great deal of investigation and inference about the events leading up to the accident:

> Appendix 2 does not go back far enough to consider the dissemination of information about the planning and maintenance work on ATCO systems (recommendation 01/2003) nor does it address the events that led to recommendation 11/2004 on the need to provide backup telecommunications systems in the event of a main telecommunications system failure. Secondly, the timelines in Appendix 2 is entitled 'Events in both Cockpits' hence the focus is not on the circumstances surrounding the controller's actions. (p. 7)
>
> The key point is that these timelines provide an entry point for any analysis and must be supplemented by additional techniques if they are to yield more detailed insights… (p. 8)

As we discovered in constructing the Brahms simulation, Johnson mentions that the timing of the STCA acoustic alert and why it wasn't heard are not described or depicted together anywhere in the document; one must piece together remarks on different pages:

> On page 44 of the BFU report it is stated that: "At 21:35:00 hrs the MV computer of ACC Zurich generated an acoustic STCA message which was addressed to the workstation RE SUED. It was not heard by any of the staff members present in the control room" (BFU, page 44 [sic; page 42]).
>
> Page 77 extends this analysis by revealing that the several seconds before the STCA the controller was already aware of the potential conflict and was taking action, which he believed would resolve the situation. At 21:34:49 and again at 21:35:03 he issued instructions for the TU154M to expedite a descent. It is important to note that this excerpt does not mention the STCA audible alarm even though the controller's preoccupation with issuing descent instructions provide a cogent explanation for the failure to hear this alarm….
>
> Again on page 91 [sic; page 89], a similar set of observations is made. The aural STCA alert was issued at 21:35:00, it was not heard by anyone. This paragraph does mention that the Controller had reacted to 'resolve the conflict' when the alarm was issued. (p. 9)

Of particular importance is how the Zurich ATCO finally realized the collision danger causing him to instruct the BTC pilots to descend. This topic is never mentioned in the investigation report, but is essential for understanding how people and systems interacted, particularly to evaluate resilience of backup processes and redundant sources of information:

> The construction of the more detailed timeline helped to identify the importance of the STCA and the timing of the controller's initial response to the conflict. It raises a number of questions that are not fully analyzed in the BFU report. For instance, *it is unclear when precisely the controller became aware of the potential conflict.* Similarly, it is difficult to determine *what might have made him aware of the potential conflict.* For instance, Appendix 2 of the BFU report indicates that the TCAS alerts were generated in the two planes at 21:34:42. It does not record any communications that alerted the controller to the conflict and that might then have triggered his instruction to the TU154M to 'expedite' the descent at 21:34:49. It seems too much of a coincidence that the controller responded within seven seconds of the TCAS warning and so *he may have been alerted by overhearing radio communications [of the Russians communicating among themselves?].* However, this is not explicitly stated in the BFU report. If he had been alerted by other systems or observations then this might add further insight to the report. (p. 9)

Johnson's analysis of the importance of STCA alerts highlights that the BFU Report focuses mostly on TCAS interaction with pilots and omits how to improve ways that automation interacts with ATCOs:

> Irrespective of the mechanisms by which the Controller was alerted to the conflict, our analysis has clearly shown the importance of the STCA system in this accident. *The importance of this 'safety net' is not reflected in the existing BFU recommendations.* The previous quote from paragraph 91 [page 89] of the report clearly shows that *even if the STCA warning had been heard and acted upon then it is unlikely that the collision would have been avoided [but if the Optical STCA were*

operational it would have generated a visual warning sooner, which would have been sufficient if heeded immediately].

It should be recalled that Recommendations 07/2004, 16/2004, 14/2004 all deal with informing pilots about the operational strengths and weaknesses of TCAS/ACAS. Our analysis has shown that *similar recommendations ought to be made so that controllers are aware of the role that STCA played in this accident*.

It is also important to emphasize the diverse ways in which the STCA was undermined by circumstance. *An STCA alert was generated at the Karlsuhe center at 21:33:24 but could not be communicated* because of problems with the SWI-02 telecommunications system. The visual STCA alert at ACC Zurich that would have been presented some two minutes before the aural alert was disabled as a result of the upgrade activities.

One insight into the Überlingen collision is that current STCA systems provide a final safety net. They do not guarantee that controllers will be able to respond in time to avert an accident and hence, *any use other than as a 'safety net' of last resort should be avoided. (p. 10)*

These remarks underscore the importance of a "total systems" model and simulation of the systems and events, including both the ATCC and the cockpits, particularly so one can understand what people perceived and hence better understand their behavior.

The BFU report makes an acceptable effort to discuss organizational factors, but is deficient in mentioning cognitive and perceptual cues, merely reciting Zurich ATCO actions without explaining them. Johnson concludes:

> **Additional Recommendation 7:** A subsequent analysis of the accident should be conducted to identify the cognitive and perceptual cues that helped the controller to identify the potential conflict. (p. 21)

With respect to organizational factors, Johnson argues that the way the Zurich ATCC management handled the temporary repair work is a systemic cause and is more fundamental than the "single controller" issue because it placed that controller in an untenable situation:

> The Chief controller briefed his two colleagues about the work at the start of the shift but did not tell them of the written instructions, mentioned above, nor about the additional staff. In consequence, **a single controller was placed in a situation where they believed they were responsible for the tasks associated with radar planning, radar execution, shift supervisor and systems manager at a time when profound changes were being made to the technical infrastructure.**
>
> The BFU argue that the safety culture and safety management practices of the ATM service provide should have ensured minimum manning levels. However, it can be argued that overstaffing of control room environments can lead to complacency, boredom and fatigue that are themselves error inducing factors during quiet intervals in safety-critical tasks.
>
> Hence, the ECF analysis … again reinforces the observation that it **is not the under-manning itself that is the root cause of the problem. The accident was caused**

> **by a combination of the under-manning *and a failure to recognise the risks* associated with the profound system changes and lack of normal system support** as a consequence of the SYCO flight plan processing system upgrade.

That is, the ATCC chief controller failed to carry out his responsibility because he did not perform actions that were well within his authority. Specifically, the supervisor could have instructed the two ATCOs to remain in the center until the maintenance work was completed. In particular, the following sequence of events is striking:

> 21:00:00Z maintenance work begins to reconfigure the system
>
> 21:15:00Z (approx.) the second controller left (BFU Report, p. 42)
>
> 21:23:00Z the remaining ATCO agreed for the phone system to be reconfigured, requiring use of the bypass system. (p. 17)
>
> *21:35:43Z first attempt to call Friedrichshafen*
>
> *21:29:25Z second attempt to call*
>
> *21:32:50Z third attempt to call*
>
> 21:34:37Z the main telephone system was operational again, though ATCO did not know (p. 17)

The collision occurred at 21:35:32. Therefore, *the second controller would have needed to delay his rest break for less than 20 minutes to allow the maintenance to be complete*, and it is likely the collision would have been averted.

Stemming from the above and related analyses, Johnson adds this recommendation:

> Any risk based assessment of the impact of large scale maintenance and upgrade activities should consider a range of plausible worst case scenarios especially where there may be common causes of 'failure'. *In this case it was important to consider the combined effects of the loss of telecommunications as well as radar and flight plan correlation facilities rather than considering the consequences of each system loss in isolation*. (p. 17)

The sequence of scenarios approach we have adopted in designing and developing the Brahms–GÜM, in which scenarios configure the subsystems in different ways, is consistent with this recommendation.

Regarding the affect of the distraction on monitoring the sector, Johnson notes that "The outcome of this 'distraction' or division of attention [responding to the A320] was that the controller failed to observe the radar trace of the B757's descent in response to the previous TCAS advisory" (p. 18). See Section 6.8.6 for a detailed analysis of the radar state during the crucial sweep prior to the collision.

In view of the Zurich ATC's failure also to hear the (arguably late) DHL radio call about their "TCAS descent," Johnson provides a communication recommendation for TCAS incidents—improved automation is insufficient:

> ***Additional Recommendation 8:*** *Further thought should be given to the verbal protocols governing the exchange of information between controllers and the crews of*

all aircraft involved in a TCAS incident. Whenever possible channels of communication should be kept clear until all the parties involved have confirmed their immediate response to the warnings. The BFU recommendation 08/2004 that RA's be downlinked to ATCO does not remove the need for such a verbal protocol given that even the revised ICAO guidelines offer crews discretion in the response to an advisory if they feel that to follow the TCAS alert would endanger safety. (p. 21-22)

6.5 Human Factors Analysis and Classification of Causal Factors

Analyses of the Überlingen accident suggest a variety of systemic and immediate causes. An obvious systemic cause of the accident is the missing air traffic controller in Zurich. Yet without the late arriving AEF flight the Zurich ATCO would have been monitoring the sector more normally and even with the lost equipment during the maintenance, would have been able to avert the collision. Similarly, a single controller could have handled the AEF flight and been done with that responsibility 10 minutes before the collision (about 21:25:43) if the backup phones had functioned. Or if the STCA visible alert had functioned, other combinations of problems might have been tolerated. With so many factors interacting over short time intervals, a simulation that breaks the scenario into independent factors and examines how they interact would therefore be useful. To do this, it would be helpful to have a systematic method for organizing the known factors and determining the completeness of the failure analysis.

The "Human Factors Analysis and Classification System" (HFACS) provides a way of organizing the causal factors in an accident (Shappell and Wiegmann 2000). HFACS was developed to provide structure and detail to the four levels of human failure presented by Reason (1990; Shappell and Wiegmann 2000, p. 2):

1) Organizational influences
2) Unsafe supervision
3) Preconditions for unsafe acts
4) Unsafe acts of operators

A hierarchical outline of events and causes provides a way to identify and organize the factors that a simulation might include. (See Section 7.2.3 for a critical analysis of the conventional sequential ordering of these levels.)

The following four subsections correspond to the four levels of HFACS with Überlingen events and anomalies listed.[17]

6.5.1 HFACS Level 1: Unsafe Acts

1) *Errors – unintentional behaviors*

[17] This particular HFACS outline and the descriptions are adapted from Wikipedia—http://en.wikipedia.org/wiki/Human_Factors_Analysis_and_Classification_System (Accessed 2 October 2012).

a) **Skill-Based Errors:** operator's execution of a routine, highly practiced task relating to procedure, training or proficiency results in an unsafe a situation (e.g., fail to prioritize attention, checklist error, negative habit).
   i) Failure of Zurich ATCO to monitor larger airspace
   ii) Failure of Zurich ATCO to put AEF flight into holding pattern

b) **Decision Errors:** behaviors or actions of the operators proceed as intended yet the chosen plan proves inadequate to achieve the desired end-state and results in an unsafe situation (e.g. exceeded ability, rule-based error, inappropriate procedure).
   i) Approval by Zurich ATCO of DHL altitude change to FL360
   ii) Failure by Zurich ATCO to use the opportunity of the BTC handoff (when he acknowledged its arrival in his sector) to resolve the eventual DHL intersection, especially given the previous clearance of DHL FL360 and control strip indicating BTC FL350 after TRA VOR (BFU Report, p. 75)
   iii) Instructing BTC pilots to descend without knowing or confirming path of DHL (given TCAS algorithm, ATCO should have known that both planes were within seconds of TCAS RA that might overrule him)

c) **Perceptual Errors:** an operator's sensory input is degraded and a decision is made based upon faulty information.
   i) Failure by Zurich ATCO to hear STCA audible alarm
   ii) Failure by Zurich ATCO to notice DHL and BTC flights on A RE (ARFA sector) radar display at least two minutes earlier than they were detected at S RE (left) workstation

2) *Violations – willful disregard of the rules and regulations*
   a) **Routine Violations:** a habitual action on the part of the operator and tolerated by the governing authority.
      i) Allowing second Zurich ATCO to sleep during Sunday night shift

   b) **Exceptional Violations:** an isolated departure from authority, neither typical of the individual nor condoned by management.
      i) NONE

6.5.2 HFACS Level 2: Preconditions for Unsafe Acts

1) *Environmental Factors – factors that affect practices, conditions and actions of individual and result in human error or an unsafe situation*
   a) **Physical Environment:** the operational setting (e.g., weather, altitude, terrain) and the ambient environment (e.g., heat, vibration, lighting, toxins).
      i) NONE

   b) **Technological Environment:** design and automation issues including the design of equipment and controls, display/interface characteristics, checklist layouts, task factors and automation.

i) Handling AEF flight and larger Zurich region required using and therefore moving between two workstations with radios on different frequencies
ii) Degraded radar data on display
iii) Missing STCA Optical alert
iv) Phone system including backup disabled (technical defect in bypass system, BFU Report, p. 17)
v) TCAS design provided no information to ATCO about pilot interventions
vi) TCAS design/certification did not account for ATCO intervention between TA and RA
vii) Busy radio prevented DHL pilot from reporting TCAS intervention
viii) DHL flight level data at time of Zurich ATCO's intervention was incorrect because of standard delay in radar sweep

2) *Condition of Operators – factors that affect practices, conditions or actions of individuals and result in human error or an unsafe situation*
   a) **Adverse Mental State:** mental conditions that affect performance (e.g., stress, mental fatigue, motivation).
      i) Stress from attempt to follow handoff procedure for late-arriving AEF flight without phones
      ii) Stress from repeated (four) calls by AEF flight when ATCO was attempting to monitor and deal with sector responsibilities
   b) **Adverse Physiological State:** medical or physiological conditions that affect performance (e.g. medical illness, physical fatigue, hypoxia).
      i) NONE reported
   c) **Physical/Mental Limitation:** operator lacks the physical or mental capabilities to cope with a situation, and this affects performance (e.g. visual limitations, insufficient reaction time).
      i) Inability to be at two workstations at the same time
      ii) Apparent inability to monitor A RE (ARFA Sector) radar display more broadly to detect DHL and BTC routes while tracking AEF flight and making repeated phone calls at the same workstation.

3) *Personnel Factors – crew resource management and personal readiness factors that affect practices, conditions or actions of individuals, and result in human error or an unsafe situation*
   a) **Crew Resource Management:** communication, coordination, planning, and teamwork issues.
      i) Allowing second ATCO to go off duty even though maintenance had begun and were scheduled to be completed within a half hour (a fact they might have known if they asked or had read the briefing in the break room).
      ii) Failure to include in maintenance briefing instructions for ATCOs the effects on the individual workstations and telephones at ATCC Zurich (BFU Report, p. 38)

iii) Failure by staff member from the ATCC management (who "had been instructed to act as a coordinator between controllers and technicians," BFU Report, p. 39) to engage with Zurich ATCO

b) **Personal Readiness:** off-duty activities required to perform optimally on the job such as adhering to crew rest requirements, alcohol restrictions, and other off-duty mandates.
   i) NONE reported

6.5.3 HFACS Level 3: Unsafe Supervision

Supervision Factors

a) **Inadequate Supervision:** The role of any supervisor is to provide their staff with the opportunity to succeed, and they must provide guidance, training, leadership, oversight, or incentives to ensure the task is performed safely and efficiently.
   i) Failure of Zurich supervisor to properly oversee and prepare ATCOs for maintenance process, including requiring both to be present
      (1) Failure to ensure that ATCC management coordinated between controllers and technicians (BFU Report, p. 39)
      (2) Failure to inform ATCOs about system manager's (SYMA) availability. who could have informed ATCO about phone alternative (BFU Report, p. 75)

b) **Plan Inappropriate Operation:** operations that can be acceptable and different during emergencies, but unacceptable during normal operation (e.g., risk management, crew pairing, operational tempo).
   i) Zurich supervisor could have told both ATCOs to remain on duty during the maintenance process or asked other ATCC personnel (e.g., SYMA) to assist ATCO in atypical manner (BFU Report, p. 39; 86)

c) **Fail to Correct Known Problem:** Refers to those instances when deficiencies are known to the supervisor, yet are allowed to continue unabated (e.g. report unsafe tendencies, initiate corrective action, correct a safety hazard).
   i) Failure to recognize and resolve systemic issue during similar circumstances the prior year (the same Zurich ATCO working alone allowed separation infringement in a crossing maneuver; the error was brought to his attention by an STCA Optical alert; BFU Report, p. 82)

d) **Supervisory Violation:** Refers to those instances when existing rules and regulations are willfully disregarded by supervisors (e.g. enforcement of rules and regulations, authorized unnecessary hazard, inadequate documentation).
   i) Reduction of ATCC workforce to one ATCO during Sunday evenings allowed by skyguide (BFU Report, p. 92)

6.5.4 HFACS Level 4: Organizational Influences

Organizational Factors

a) **Resource Management:** organizational-level decision-making regarding the allocation and maintenance of organizational assets (e.g. human resources, monetary/budget resources, equipment/facility recourse).
   ii) Skyguide reduction of workforce in ATCC (BFU Report, p. 92)

b) **Organizational Climate:** working atmosphere within the organization (e.g. structure, policies, culture).
   i) Overly rigid roles in ATCC, apparently without enabling informal assistance in time of emergencies

c) **Operational Process:** organizational decisions and rules that govern the everyday activities within an organization (e.g. operations, procedures, oversight).
   i) Failure by skyguide to require Zurich supervisor to manage maintenance process
   ii) Implementation by skyguide of Single Man Operation Procedure (SMOP) during day operations, despite regulatory objections, which was unofficially adopted for the night shift when a supervisor would assist by monitoring the traffic, and then continued when the supervisor position was eliminated and the staff was reduced to two ATCOs (BFU Report, p. 92).

6.6 Causal Tree Analysis

The HFCAS analysis is useful for eliciting factors that affect an accident, particularly for relating on the spot errors to broader contextual causes in design and operations. However, creating a simulation model requires understanding the multiple, ongoing processes and how they causally affect each other. The following outline was constructed for identifying what roles, equipment/automation, and events might be included in the simulation. This analysis begins to define the different states of systems and what people knew and did. Explanations of what did not occur are not necessary to replicate the accident, but insofar as they represent nominal behaviors or "best practices," they are included for generality in the simulation, broadening the space of scenarios the model can simulate.

In this informal outline, items (indented) indicate "immediate" causes contributing to the parent item, that is, they play a direct physical and/or temporal role in subsequent events. For example, the DHL aircraft descended because the pilots followed the TCAS instruction; they did not immediately communicate this action to the Zurich ATC; and they apparently did not hear or interpret the ATCO's urgent instruction to the BTC to also descend as relating to the TCAS alert or their trajectory. Contributing factors were that the DHL pilot not flying (PNF) had been out of the cockpit when the TA sounded (BFU Report, pp. 92-93), and when they attempted to communicate with ATCO the radio frequency was busy.

- Collision
    - DHLX 611 descended
        - TCAS instruction to descend was followed by the crew
        - Crew had no interaction with ATCO during this time
            - Crew did not notice how ATCO's instruction to BTC 2397 to descend related to their situation
                - PNF (co-pilot) had been away from his seat and on return was busy handling TCAS TA
        - Crew unable to communicate with ATCO when they attempted
            - Zurich ATCO radio was busy

    - BTC 2397 descended
        - BTC 2397 TCAS RA ("climb") was generated after ATCO instruction to descend
        - Crew did not view subsequent TCAS instruction as overriding ATCO.
        - ATCO repeatedly instructed BTC 2397 to descend
            - ATCO neglected control of BTC and DHL; he neither noticed the descent of the B757 nor did he hear their radio message reporting a TCAS descent.
                - ATCO distracted by AEF 1135 requiring using a different workstation
                    - Phone system not operative so unable to transfer the flight to Friedrichshafen
                        - ATCO not informed about the presence of a SYMA who would have been able to suggest an alternative
                    - Failed to put AEF into holding pattern
                    - Failed to monitor ongoing flights in sector
                - ATCO unable to safely execute all of the tasks required by two ATCO positions during this period
                    - Second ATCO was absent from his position
                    - Unaware of maintenance and its implications
                        - Supervisor did not brief ATCOs
                        - ATCOs didn't read the memo
                        - Memo did not mention loss of STCA Optical alert
                    - Maintenance caused unintended loss of backup ("bypass") phones
                    - ATCO didn't review equipment status after maintenance began
                        - ATCO could have asked SYMA to review system status
            - ATCO didn't know DHLX 611 was following TCAS instruction to descend

        - DHL unable to radio because ATCO was talking to BTC
            - Design of radio system prevented three-way conversations
    - ATCO was not alerted of the impending collision risk because the optical STCA was not available
        - DHL FL information ("now at 360") was incorrect because radar image was delayed

- ATCO inadvertently put BTC and DHL on collision path
    - After previously allowing DHL to change altitude to FL 360, did not notice when BTC first reported a course change to Zurich that put both planes at the same altitude and approaching at 64 NM
    - Did not use the handoff opportunity to instruct BTC to descend to FL 350, which the control strip showed after Trasadingen VOR

- The Karlsruhe ATCO did not avert the collision
    - The faulty phone lines prevented the Karlsruhe ATCO from contacting the Zurich ATCO about the impending collision.
    - The Karlsruhe ATCO followed a protocol that forbid contacting the DHL or BTC pilots directly on an emergency frequency (apparently even to prevent a collision), without coordinating first with the responsible controller.[18]

In summary, referring to this outline, a simulation of the Überlingen events should include the roles (e.g., Zurich ATC, pilots) and all of the systems named in this outline (TCAS, phones, radar, etc.). Off-nominal states must be modeled. Nominal behaviors that are prevented from occurring should also be simulated (e.g., DHL pilots attempt to notify the ATCO about TCAS RA), such that if conditions are different in a particular simulation run (scenario), a different sequence of events might occur. Sections 8.6 explains this "sequence of scenarios" approach and how it is implemented in a Brahms model.

[18] "The UAC Karlsruhe has a possibility of selecting the international emergency frequency 121.50 MHz…. Prior to initiating activities outside their own area of responsibility, the controllers must, in accordance with the effective regulations, coordinate them with the responsible controller, who, however, was not reachable" (BFU Report, p. 44).
"The BFU is of the opinion that UAC Karlsruhe exhausted all possibilities to prevent the impending collision. The possibility of transmitting a warning to the aircraft in form of a blind transmission on the emergency frequency 121.50 MHz was not taken into consideration by good reason. This would have been contradictory to the regulations in force, would certainly have led to a confusion of all parties involved. It would not have prevented the collision with a very high probability particularly since it could not be clarified whether in one of the two airplanes this frequency had been selected at all" (BFU Report, p. 77)

6.7 What Could Have Happened—Alternative "What If" Scenarios

As indicated by the previous sections, most analyses of the Überlingen accident focus on contributing causes. This section illustrates how one can create a *nominal model* by turning around a failure analysis to consider what could have happened instead of the actual sequence of events. By seeking to understand what factors caused the actual events to unfold, insight can be gained about psychological factors in play at the time (e.g., focus of attention, people's beliefs) and logical-practical constraints (e.g., the effect of the radar sweep delay). The nature of timing and interactions among processes becomes clearer.

As a way of beginning a "what if" analysis, one might begin with the two "immediate causes cited by the BFU Report (p. 110):

1. The Zurich ATCO failed to monitor and maintain focus on the BTC and DHL flights in his sector.
2. The BTC crew failed to follow the TCAS resolution advisory.

Creating a model of what might have happened differently requires identifying alternative strategies (approaches) for handling the situation, assuming that the same people are involved (e.g., one Zurich ATCO), they have the same information available in the actual events (e.g., what is presented on the radar display), and the equipment available in Zurich ATCC is also off-nominal. Focusing on what people could have done differently, a "what if" analysis is presented here in outline form as a series of questions and examples of alternative behaviors. Each list of alternatives begins with a null hypothesis that the actual event/behavior was inevitable.

6.7.1 What could have happened differently in Zurich ATCC?

The BFU Report comments why the Zurich ATCO didn't act differently:

> Generally it would have been possible for the controller to safely handle the traffic consisting of three airplanes at the time of the accident. The controller came to the same conclusion and did not ask for support from his colleague in the lounge. This decision was probably based on his experience regarding a smooth course of operation and did not take into consideration possible problems, such as the failure of the telephone system.
>
> Once he realized the problem with the inoperative telephone system it was already too late to alert the colleague. The repeated attempt to phone Friedrichshafen about the arrival of the A320 diverted his attention longer than intended from the proactive traffic control of the two other airplanes. (p. 105)

These explanations aside, what are the logical variations possible in the ATCO's behavior—what could he have done differently that might have made a difference?

1. Nothing would make a difference (i.e., the situation is untenable):
    a. Workload was too much for one person; an alternative would have endangered other flights

b. Circumstances are too complex: Any alternative would cause a conflict/situation that is not permitted.

2. Strategize attention differently by reframing conceptually what he was doing:
    a. Realizing the extreme danger, stay focused on the DHL and BTC flights after detecting danger, putting everything else on hold, until he had confirmed definitively that separation was assured:
        i. would have required 10-20 seconds to confirm on radar and/or
        ii. would have entailed contacting DHL to determine their situation
    b. Realizing that he had multiple flights and levels to monitor, more quickly disengage from the attempt to call Friedrichshafen Tower:
        i. Noticing the collision danger sooner would have made a difference:

> 21:33:49Z—This is the latest time at which Zurich ATCO should have given BTC 2937 the instruction to descend to FL 350 - i.e. at least one minute before this instruction was actually given (BFU Report, p. 75).

        ii. However, the BFU Report claims that he was appropriately assigning higher priority to handling the AEF flight (A320):

> At first the ATCO assigned a high priority to the task of handling the A320 arriving late at Friedrichshafen, as evidenced in his preparation for the ARFA sector and attempt to coordinate with Friedrichshafen. This was in accordance with the requirements of the ATM Manual but it distracted from the task of evaluating and planning the upper air situation (p. 84).

3. Categorize the conflict situation differently and act accordingly:
    a. Very likely, by the time Zurich ATCO detected the conflict, the only course of action that would prevent collision was advising BTC to climb
        i. Was the suggestion to BTC to descend based on a recency effect because he more recently communicated with BTC than DHL about FL360 (at 21:30:28Z)?
    b. Recognizing his broader responsibilities, the Zurich ATCO could have put the AEF flight into a holding pattern so that he could properly monitor the other flights and get assistance.

4. Get assistance of other Zurich ATC:
    a. Realizing the multiple obstacles to getting and giving information, he could have asked the Zurich ATCC CA to get the resting controller as soon as he was told that maintenance would be occurring and requiring use of backup
    b. Ask ATCC to spend more time resolving the communication problem, which might have included the CA consulting the manual:

> When the ATCO could not contact Friedrichshafen on the Bypass system he asked one of the CAs to find out another phone number. When he had no success with this number he discussed the options of relaying the information via Munich or contacting the technicians, before settling for the option of asking the crew of the A320 to contact Friedrichshafen directly. The Emergency Manual listed the three phone systems available to ATM staff, but the ATCO was not aware of this, so did not consider the use of the mobile phone at DL's suite.(BFU Report, p. 83)

6.7.2 What could have happened differently in the BTC cockpit?

That is, what could the Russian crew have done differently?

1. Nothing would make a difference (the situation is untenable):
    a. They were already following the controller's instruction prior to TCAS RA; it was already too late to change the aircraft's direction. (This was not the case.)
    b. The Zurich ATCO did not know about TCAS RA and might have continued to repeat "Expedite" instructions that could not be ignored.

2. First officer sitting behind the commander on the left could have been more forceful about following TCAS:

    > 21:34:59Z--BTC 2937 the copilot stated: "It (TCAS) says (говорит): "climb". The PIC replied: "He (ATC) is guiding us down". The copilot's enquiring response: "descend?" (BFU Report, p. 8).

3. Russian crew could have followed TCAS, believing it was the final word and over-ruled anything instructed by a controller.

As noted in the previous section, the Karlsruhe controller might have acted differently by contacting the pilots of the DHL and/or BTC directly. But this intervention is so problematic it is not included in the Brahms simulation reported here. While physically possible, it is contrary to ATC rules and practice.

6.7.3 Implications of "What if" analysis for model design

The above exercise provides a set of requirements for how parts of the work system should be modeled so it has flexibility (can be reconfigured) for modeling different scenarios. For example, the alternative realities we would like to simulate include:

- DHL pilots attempt to radio Zurich ATCO about TCAS RA and aircraft altitude change immediately after following TCAS instruction.
- When BTC first reported to Zurich, ATCO notices that planes are at same altitude and approaching at 64 NM; ATCO uses the handoff opportunity to instruct BTC to descend to FL 350, which the control strip showed after Trasadingen VOR.
- Backup phones actually work during the maintenance period; the Zurich ATCO uses the backup phone to call Friedrichshafen.

- Zurich ATCO checks phones immediately after maintenance begins and detects a problem; maintenance procedure is immediately halted.
- Zurich ATCO puts AEF into holding pattern on first noticing all phones are out; thus he prioritizes monitoring the broader sector.

Each of these variations involves one or more behaviors of people and/or systems that did not occur at Überlingen that should be included in the generalized model. These might all be construed as "nominal" or "best practices."

Because of project staffing limitations, the model cannot contain at first everything that might have happened differently to avoid the accident. Emphasis is placed first on simulating the complex interactions that did occur and exploring the sensitivity of the Überlingen work system configuration to circumstantial variations of timing and probability represented in the model. Small changes, such as the ATCO's frequency of monitoring the broader sector, might be more interesting to explore at first than "adding back" tools and automation (e.g., STCA optical alert) that would have very likely prevented the collision. However, the process of creating a general model involves including all of the behavior variations listed above and more, and systematically verifying that different combinations interact properly (e.g., if the backup phone works, the Zurich ATCO uses it at the appropriate time) and validating that simulation outcomes are plausible.

6.8 Crucial Nature of Timing/Sequencing of Events

The Überlingen narrative clearly shows that timing played a critical role in the outcome. Therefore, analysis and simulation modeling must pay close attention to what events occurred "late" or interacted because they were simultaneous, as well as what preceding factors affected these timings.

Some activities, such as the scheduled maintenance, persisted throughout the critical period of the approaching DHL, BTC, and AEF aircraft. The phone system was not operating for 17 minutes, and this occurred when AEF 1135 needed to be handed over to the tower controller. We do not model or allow to vary when the maintenance occurs, but rather the model can be configured (initialized) to represent the implications of the maintenance. That is, anomalous configurations (e.g., STCA optical and non-functioning telephones) can be varied to define a scenario. This approach follows from not simulating the maintenance activity itself, for example to simulate how the engineers' actions caused the phones to be disabled. This is a general approach for constructing "dispositions" of objects and people in a Brahms model—discrete known states and/or behaviors can be modeled and configured rather simulating the history of events that caused those states/behaviors.

Many of the key events, such as what the Zurich ATCO perceived on a radar display at a given time, are not documented and must be reconstructed—these assumptions can have a large effect on the outcome (particularly whether and when the Zurich ATCO intervenes to avoid a collision). Also the duration of many activities is

circumstantial (e.g., time it takes Zurich ATCO to speak to assistant about phone number), and hence these will be probabilistic (random distribution over a min-max range); even a few seconds variance from one simulation run to the next (using the same scenario) could cause different outcomes.

We knew of course such variability could be a property of a Brahms simulation, but Brahms-GÜM is the first time we have encountered such critical temporal interactions. The variability itself makes Brahms-GÜM an excellent candidate for model checking to explore and formalize how events interact.

This section outlines kinds of temporal interactions and describes the most important aspects of the simulation in which timing of events needed to be analyzed, reconstructed in detail, and assumptions made in the model.

6.8.1 Temporal aspects of the scenario

The interactions and events leading up to the collision have a wide variety of temporal relations that are modeled in Brahms-GÜM:

- **Parallel** – processes occurring independently at the same time in their own space, e.g., the planes are flying with their own flight systems independently of each other
- **Simultaneous** – processes that occur at the same time but overstep each other in the same space, e.g., TCAS RA to "Climb!" overlaps last second of Zurich ATCO "descend – expedite" to Russians, and also overlaps their action to disengage autopilot.
- **Sequence** – a process with established ordered steps, e.g., Zurich ATCO observes flight exiting his sector, he calls them instructing to contact next ATCC with a given frequency; they acknowledge.
- **Periodicity** – the regular rhythm of a process, a pattern of state changes or variations in intensity, e.g., the generally quiet period on Sunday evening with few flights in the Zurich airspace and no planes landing locally, contrasted with Monday morning, which affected the staffing choice of single person operation at the time of the Überlingen accident.
- **Phases** – relatively prolonged system states/processes; the system exists in different phases of operation/behavior, well-defined for designed systems, e.g., TCAS phases: indicating on monitor another flight nearby prior to TA; phase after TA; phase after RA. Phases in human behavior correspond to Brahms "activities."
- **Temporary** – a process/role that is in effect for a particular period, e.g., "pilot flying" or Zurich ATCO being responsible for the entire airspace while his partner is on break.
- **Permanent** – a fixed process/role associated with a job, object, or setting, e.g., how TCAS operates; how Zurich ATCO must handle flights landing locally at night.

6.8.2 Effect of departure time on collision

Regarding ultimate causes of the accident, it is important to observe that if either or both flights had departed on time (BTC was 18 minutes late; DHL 6 minutes late), the planes would not have collided. For example, if the Russian flight had departed on time, it would have been about 164 miles closer to Barcelona when the DHL plane reached the point of intersection of their flight paths. Similarly, if the DHL flight had departed on time, it would have been about 53 miles closer to Brussels when the Russian plane reached the point of intersection.

Based on their field cruise speeds of 463 kt and 470 kt (or roughly 7 nm/min), a *one minute difference* on route for either of the planes would have prevented the collision.

6.8.3 Effect of timing on following ATCO vs. TCAS

Presumably the Zurich ATCO contacted the BTC because he noticed the DHL and BTC aircraft on the radar display (if STCA Optical alert had been functioning, it would have alerted him much sooner). However, we do not know which radar display he was observing (left or right workstation) and why he perceived the imminent collision at this moment. Both aircraft were visible on the radar displays of both workstations for some time prior to his recognition of the problem.

The timing of the ATCO perceiving the aircraft locations, judging that a separation violation will occur, and intervening by advising the pilot(s) is of course pivotal in preventing a collision. Given the simultaneous, independent operation of TCAS onboard the aircraft, the intervention could occur before or after a TA or RA. In general of course, no collision will occur if the ATCO notices a conflict, instructs an aircraft, and the pilots react promptly before a TCAS TA. Furthermore, if ATCO intervenes soon after a TA and before a TCAS RA, then whether pilots allow ATCO to overrule TCAS instruction could be irrelevant—the pilots could take action sufficiently before the RA such that the RA takes into account their current change in altitude (that is, if still required, the advice would be consistent with their respective current trajectories, e.g., BTC would be descending). Thus the timing of the intervention relative to the RA can be expected to affect the outcome, with this timing becoming more critical if the pilots allow the ATCO instruction to dominate (ignoring TCAS RA) as occurred at Überlingen.

The analysis of timing is actually a bit more complicated because for the pilots to follow TCAS RA might require reversing an action already underway, the situation in which the BTC pilots found themselves. The BFU Report does not show sensitivity to the possible difficulty of reconceiving an emergency situation and reversing action in just a few seconds.

The BFU Investigation Report lists two immediate causes of the accident, the ATCO's lack of situation awareness and the BTC pilots not obeying TCAS (p. 5):

> The TU154M crew followed the ATCO instruction to descend and continued to do so even after TCAS advised them to climb. This maneuver was performed contrary to the generated TCAS RA.

However, the timeline of events (BFU Report, Appendix 3) shows that the TU154M crew realized the collision danger by reading the TCAS display before the TCAS TA, so at least one pilot onboard (Left Officer Rear) was understood this system's existence and function. More importantly, ATCO began instructing them to descend 6 or 7 seconds before the TCAS RA to climb. Yet the official conclusion of the BFU Report ignores (or at least disguises) the order of events. It is not clear whether the TU154M crew were giving priority to the ATCO instruction because it came first or that they viewed the ATCO as having more authority, or indeed that it was psychologically (or even physically) difficult to disengage from the interaction among the ATCO, pilots, and aircraft control system that was already in process.

Notice the wording in the report that *the maneuver was contrary to TCAS*. But it is evident from the chronology that the maneuver occurred in the final second(s) of the ATC's instruction to descend and was within a second of the TCAS instructing to climb. The *decision* to descend in response to ATCO came before TCAS gave a contrary instruction. This is obvious in the timeline graphic (Figure 6-2)—notice the AP disengagement signal. In contrast with what appears in the BFU Report, a more fair statement would be:

> The TU154M crew **was already following** the ATCO instruction to descend **when** TCAS advised them to climb. This maneuver **continued** contrary to the generated TCAS RA.

It might be argued that the issue here is *not who or what is viewed as having authority*— as if the TU154M crew weighed these two options impartially—but that *people are more likely to persist in an ongoing interaction, rather than shift attention to a different agent giving direction in mid-course*, particularly in a life-threatening situation when time is of the essence.

Put another way, a committed and dangerous "activity in process" trumps further reasoning. Also, the constraints are not just mental processes occurring in a single mind. Although the BTC commander (CP R Front) is very obviously in control and making the key decision to ignore TCAS, the transcript reveals a group of interacting pilots with at least in principle the possibility of the commander being convinced to follow TCAS. As a team they fail to act on the first officer's observation, "It says 'climb'" and subsequent question, "Descend?" Hence group dynamics, which themselves involve authority, are of special interest here. To understand the outcome, we need to understand the Russian crew's individual beliefs about each other, their protocols, and especially their roles and authority structure. Indeed, we would need to know considerably more about the commander's and first officer's experience and history relative with TCAS (e.g., had they ever flown when TCAS

issued an RA? As a special instructor onboard, did the commander actually have no training in TCAS or had he been trained to trust ATCO?).

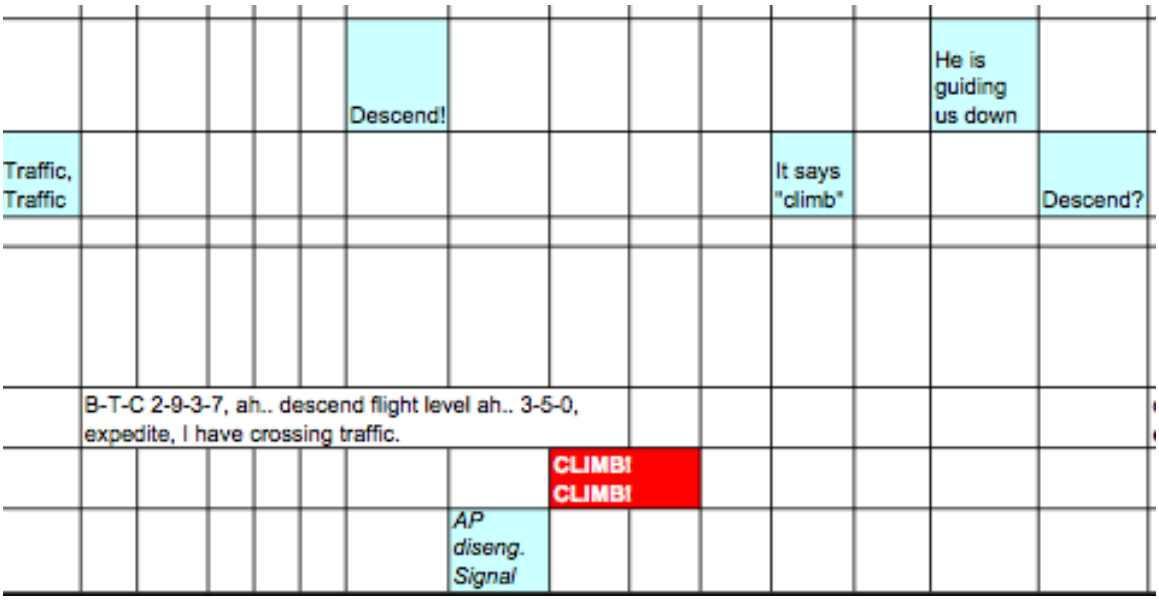


Figure 6-2: BFU Investigation Report Timeline (Appendix 3): ATCO instruction indicated by cursor occurs before TCAS RA (red). Blue squares represent Russian crew utterances and actions.

In short, although there is one "pilot flying" on the BTC, we start by assuming each person's actions are the result of personal knowledge and practices interacting with a dynamic environment, which included the TCAS display, other crew member statements and questions, the ATCO directive, and what was ultimately visible outside the cockpit. We may want to simulate what beliefs, perceptions, and practices caused specific changes in control. An open question is whether the BTC PF acted relatively autonomously and why the crew didn't more strenuously object.

The BFU Report provides convincing evidence that the BTC pilots were indeed uncertain what to do. The descent begins, is halted, the ATCO intervenes again saying "Expedite!" and the descent is continued. During this time TCAS has told DHL to "Increase Descend!" but does not intervene with the BTC to "Increase climb!" until 14 seconds later. This delay allowed the BTC crew insofar as they were weighing whether to follow ATCO to be swayed by his repeated, more urgent instruction to expedite descent after the TCAS RA, during which time TCAS remained silent. This lack of parallelism of TCAS interventions in the two cockpits is not commented on in the BFU Report.

Again, the BFU report's claim that the crew continue to descend after TCAS instructed them to climb in not strictly correct. The complexity of the interaction is articulated in the BFU Report itself (p. 10-11, with interpretations in bracketed italics from p. 72-73):

At 21:35:02 hrs, (six seconds after the RA " climb, climb") the PF pulled the control column. As a result, the rate of descent ceased to increase. The vertical acceleration rose from 0.75 g to 1.07 g. The engine thrust remained unchanged in conjunction with this control input….

[This could either be a delayed reaction of the PF to the RA "climb, climb" or a reaction to the exclamation of the copilot: "it says climb". It is the BFU's opinion that this action was taken to adjust the descent rate after a rapidly initiated descent. The analysed TCAS data shows a continued rate of descend of approximately 1 200 ft/min after stabilisation.]

At 21:35:03 hrs, the engine throttles were pulled back further.

The discussion between the crew members was interrupted at 21:35:03 hrs by the controller instructing the crew once again to expedite descend to FL 350 ("... descend level 350, expedite descend"). This instruction was immediately acknowledged by the PNF. The controller then informed the crew about other flight traffic at FL 360 in the 2 o'clock position ("...Ya, ... we have traffic at your 2 o'clock position now at 3-6-0") and the PIC asked: "Where is it?", the copilot answered: "Here on the left side!". At the time, the rate of descent was approximately 1 500 ft/min.

The voice of the flight navigator can be heard on the CVR saying:" It is going to pass beneath us!" while the controller was giving his last instruction.

[The conversation of the flight crew was interrupted by the controller who instructed them once again to expedite descent to FL 350. The second instruction of the controller had become necessary as the TU154M crew had not verbally replied to the first one. The disagreements have obviously prevented that the first instruction was (sic) acknowledged. The second instruction of the controller stopped the conversation of the crew about the RA. For the two pilots it was another confirmation of the decision to follow the ATC instruction, particularly since the other airplane was reported by the controller to be at FL 360.]

At 21:35:04 hrs the roll channel of the autopilot was switched off.

[The second instruction of the controller at 21:35:03 hrs coincided with the retraction of the thrust levers. It is the opinion of the BFU that this action was probably carried out by the instructor. The PF held the control column pulled for about two more seconds, before pushing the control column again in order to increase the rate of descent.]

At 21:35:05 hrs, the PF pushed the control column again and the rate of descent increased to more than 2 000 ft/min.

Five seconds later TCAS instructs the DHL to "Increase descent!" but does not correspondingly instruct the BTC to "Increase climb!" until 21:35:24.

One could complicate the situation in ways that make the importance of reasoning among the aircraft crew more salient. For example, if the Karlsruhe ATCO had intervened (contrary to regulations because he was not able to coordinate with the Zurich ATCO who was the responsible controller), the BFU Report argues that confusion might have resulted. If the Karlsruhe ATCO had given instructions contrary to TCAS, there is reason to believe the DHL pilots would have ignored him (so the outcome would be the same). In this case, he would have put the DHL pilots in the same situation as the Russians, having to decide whether to follow TCAS or the ATCO—though with two ATCOs saying the same thing, disregarding TCAS would perhaps have been easier. Or if the Karlsruhe ATCO gave instructions consistent with TCAS, the Russians would have received contrary instructions from the two ATCOs with no time to reason out who was correct—though delaying their descent further (as possibly occurred at 21:35:02) or going with the majority (Karlsruhe and TCAS) might have avoided the collision.

6.8.4 Relation of flight level and descent timing

The above analysis can be made more precise by referring to the altitudes of the two aircraft from the time descent began. To see the difficulty, notice the altitude discrepancies in Figure 6-3 given the three sources (TCAS flash memory data, BTC Flight Data Recorder, and calculations provided by IAC Moscow).

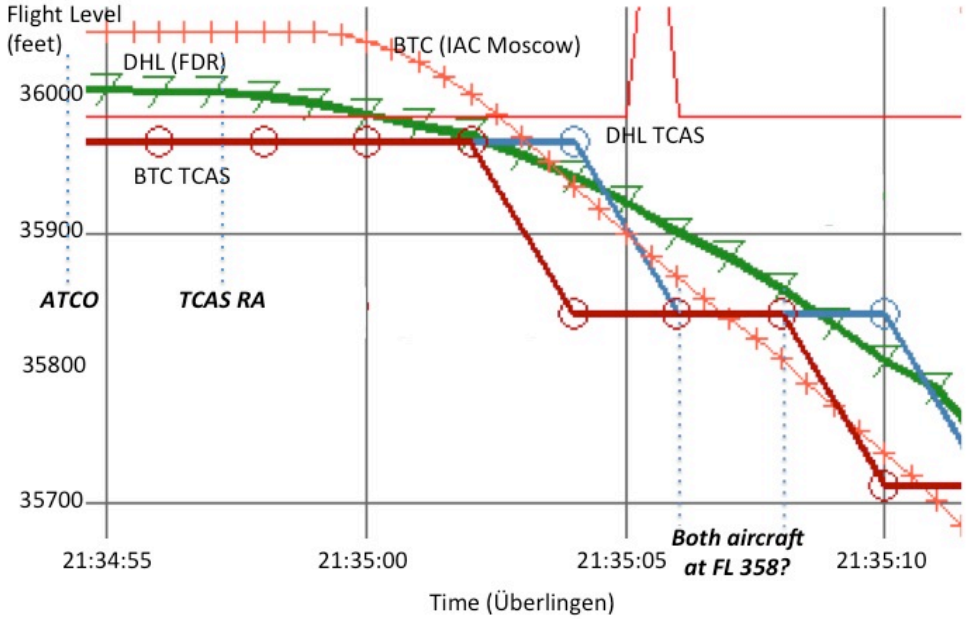


Figure 6-3: Flight levels of BTC and DHL aircraft at time of descent (from "TCAS- and FDR- parameters (extracts) of B757-200 and TU154M," BFU Report, Appendix 6).

The chart shows that the planes were not precisely at FL360 (i.e., 36000 feet) and TCAS calculations are discrete (snapshots at points in time). TCAS places the planes at the same flight level at 21:35:06, but other calculations show this occurred 2 or 3 seconds earlier. According to the calculated levels the aircraft DHL begins descent

at least two seconds before the BTC; but TCAS shows it was the opposite, BTC was two seconds ahead of the DHL. Because two aircraft 7 nm apart descending at the same rate from the same flight level at different times will not collide, when the descent begins and the varying rate from the expedite instructions of ATCO and TCAS are pivotal in determining whether a collision occurs.

Referring to the BTU Report timeline (Appendix 3), we can infer that the two crews and their aircraft responded similarly to instructions, but BTC had a two second head start because of TCAS's intervention. Both crews disengaged the autopilot (AP) 2 or 3 seconds after hearing the instruction to descend (BTC responding to ATCO and DHL to TCAS), and both aircraft reached FL358 eight seconds after the AP was disengaged. This strongly implies that the BTC crew lost little time considering or discussing what to do; the pilot flying responded as quickly on both aircraft. After BTC first officer (Left Rear) said, "It says climb!," at 21:35:01 instructor (CP Right Front) explained, "He is guiding us down" and the aircraft was already plainly on a descent path. (Though as presented in the previous section, the rate and possibly degree of commitment varied in the first 7-8 seconds until the ATCO commanded them to expedite and indirectly forced their acknowledgement.)

Table 6-1: When aircraft crew heard instruction, reacted, and FL358 was attained.

Flight	Crew heard "descend!"	AP disengaged	Aircraft dropped 200ft to FL358
BTC	21:34:53 (CP R Front says "descend!" a second later)	21:34:56	21:35:04
DHL	21:34:55-56	21:34:58	21:35:06

In a subsequent section, this same data is interpreted from the perspective of the ATCO in terms of what appears on the radar display.

6.8.5 What the control strips reveal about timing

The following analysis was inspired by a remark in an online blog in which air traffic controllers and pilots were commenting on the Überlingen accident:[19]

> Kontrolor :
> ultimately controller on duty cleared two planes on conflicting route to the same level.
>
> ATCO Watcher: Not quite correct my friend, he did not clear them together. The 757 was cleared off route direct by Geneva, and the Tu154 cleared off route direct by Munich, on the strip he had them 7 minutes apart…

Figure 6-4 shows the flight control strips, which we can use to understand the second remark. The first box shows the destinations of the two flights, Brussels

[19] http://www.pprune.org/archive/index.php/t-276578.html

(EBBR) and Barcelona (LEBL). The second box indicates the planned altitude when the aircraft arrives in the Zurich sector, with the ATCO's clearing the DHL for FL320 during the handoff and clearance to FL360 after dealing with six other flights and making his first call to Friedrichshafen. The third box indicates the planned altitude within the sector and that both request FL360. The last box indicates planned arrival times at waypoints before, within, and after the sector.

DHX611 LIME TGO B752	7524 0463 EBBR 465 48	260 320 360	360 R360	ABE KUD LOK 2120 2130 2135
BTC2937 UUDD NINTU T154	7520 4125 LEBL 470 44	360	350 R360	NEG TRA BEN 2136 2142 2151

Figure 6-4: DHL and BTC control strips (BFU Report, p. 36).

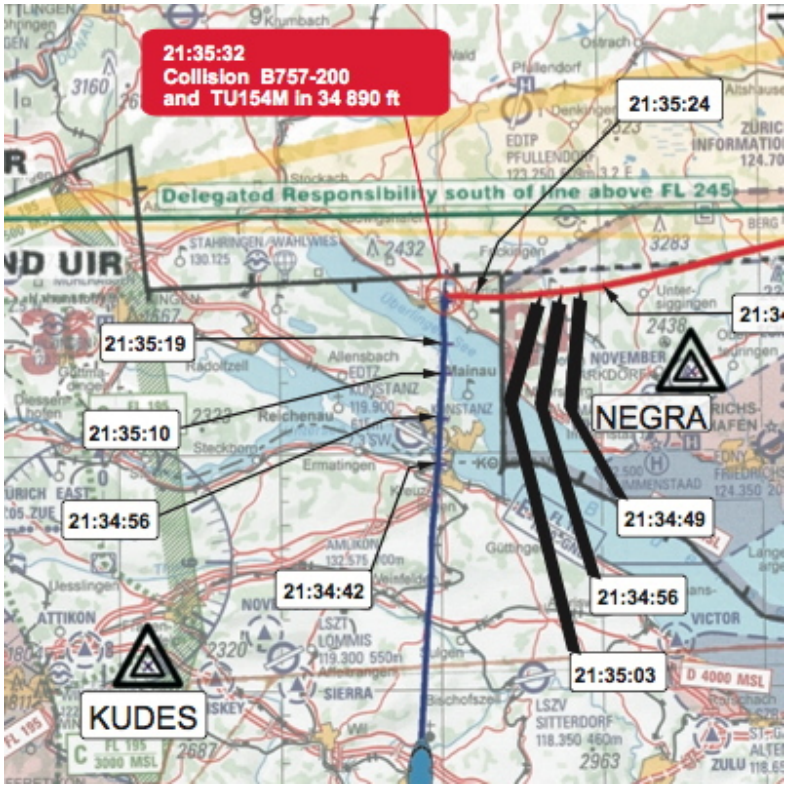


Figure 6-5: Aircraft positions (DHL blue, BTC red) and timings from BFU Report, Appendix 1, "Reconstruction of flight path according to radar data."

Table 6-2: Variance between control strip plan and actual timings for BTC and DHL flights.

	NEGRA (BTC)	KUDES (DHL)
Control Strip Time	2136	2130
Actual Time (approx.)	21:34+	21:33:30-
Variance from Expected	-2:00	+3:30

The actual position timings of the two aircraft relative to NEGRA and KUDES waypoints are summarized in Table 6-2, interpolating values given in the BFU Report (Figure 6-4).

The waypoint times for NEGRA and KUDES were originally 6 minutes apart (2136 – 2130). Because of unaccounted timing variations (perhaps from DHL altitude change to 360 and redirection "direct ABESI"), they arrived at NEGRA and KUDES respectively only about 30 seconds apart. In this respect, given that the planes are known to have arrived at the intersection point within seconds of each other, we can infer that the variance for their individual passing of NEGRA and KUDES is a reasonable measure of the planned separation, that is, DHL would have crossed the intersection point at least 5 1/2 minutes after the BTC (Table 6-2).

The ATCO Watcher in the blog says, "on the strip he had them 7 minutes apart," possibly referring to LOK and TRA, which are 7 minutes apart.

According to the ANSA commentary (p. 75), an ATCO would be considering and reading entry times from the control strips:

> Besides information on the call sign, aircraft type, speed, aerodromes of departure and destination and route of flight, they always contain the cleared flight level and the calculated time of entry into the respective control area (airspace sector).

It was reasonable for ATCO to clear the DHL to FL360 because the BTC was not yet in his sector. Subsequently not adjusting the BTC altitude during the handover might have been reasonable if the ATCO were making decisions on the basis of the control strips alone, as "ATCO Watcher" comments on the blog. However, "Both airplanes had been cleared for a direct approach to Tango VOR (B757-200) and Trasadingen VOR (TU154M) and thus the control strips did no longer correspond to the actual flight paths" (BFU Report, p. 75). Crucially, at the time of handoff, it would not make sense to be referring to the control strips alone, given that the aircraft were both visible at this time on the radar display before him. Consequently, ATCO's mental model appears to be based not on the control strips, but (as he indicated at the debriefing) that he did not believe the separation at the time of handoff (64 nm) to merit concern. His priority was to handle the late arriving AEF flight.

ATCO's distraction at the time of the BTC handoff is obvious:[20] Immediately after his second attempt to reach Friedrichshafen, AEF 1135 called in simultaneously; he put AEF on hold, asked BTC to repeat, they stated "level 360," ATCO gave the frequency, and AEF 1135 interrupted, told them to wait a second time, then proceeded to handle two other flights, when AEF interrupted again, at which point he focused his attention on them.

AEF's persistent interruptions coming during and after the BTC handoff provides strong evidence that the ATCO is not referring to the control strips or viewing projected trajectories proactively, but is rather focusing entirely on processing handoffs in a reactive manner.

6.8.6 Effect of radar sweep delay

A striking example of the interaction between automation, perception, situation awareness, and actions is revealed by the simple effect of the timing of the radar sweep to renew the display:

> … The B757-200 was already at FL 356 due to the descent initiated after the RA. However, the controller could not read the new flight level on the monitor, because the descent of the B757-200 was only to be seen after the radar image renewal at 21:35:24 hrs. With the preceding target image renewal at 21:35:12 hrs the FL 359 shown was still within tolerances. (BFU Report, p. 76)
>
> When the controller observed on the left monitor (RP) that the TU154M had initiated the descent he considered the problem solved and once again turned to the right monitor (RE). (p. 76)
>
> Even with the aural alert the ATCO would not have been able to recognise the situation was not evolving as he expected until further information was available. The TU154M was already complying with the descent instruction and the ATCO did not know the B757-200 had initiated an RA related descent. He would not have been able to recognise the B757-200 was descending until the screen update at 21:35:12 hrs or if he had heard the crew's TCAS descent call a few seconds later. (p. 89)

First, notice that the BFU report has contradictory claims about when the descent of the DHL would be visible: page 76 states 21:35:24 and page 89 states 21:35:12. He is telling BTC to expedite its descent from 21:35:03 to 21:35:17 (BFU Report Timeline, Appendix 2). Apparently the remark on p. 89 is incorrect; he would have had to wait 7 seconds (until 21:35:24) to confirm that the separation problem was resolved. If he had waited, he would have seen the DHL at FL352! Instead he returned to the right workstation to deal with AEF 1135, which had interrupted him during his urgent call to the BTC to expedite their descent.

Examining the available altitude and timeline data shows that the report's claims are contradictory and inconsistent—he could not have seen that the BTC was descending until four seconds after he completed the expedite instruction. Also,

[20] Refer to transcript of this period in Section 9.7 and annotated Brahms-GÜM log of events in Appendix 26.

there is no argument why ATCO gave priority to the AEF instead of calling the DHL and telling them to climb (and probably change course). Table 6-3 relates the display and actual altitudes with the radar updating process.

Table 6-3: Given and inferred flight levels, emphasizing DHL & BTC data visible to ATCO at 35:12 when he called BTC to expedite descent. "Radar Refresh" indicates when DHL aircraft data is refreshed on Zurich display (indicated by X). "Timeline" values are interpolated from BFU Report Appendix 1 map (Figure 6-5). "Chart" values are interpolated from the graph in BFU Report Appendix 6 (Figure 6-6).

	24:48	24:54	24:35:00	35:06	35:12	35:18	35:24
Radar Refresh	X		X		X		X
BTC Timeline	360	360	360	358	356	354	352?
BTC Chart	360	360	360	358	**357**	354	
BTC Display	360	360	360	360	**FL356?**	354	352?
DHL Timeline	360	360	360	358	356	354	352?
DHL Chart	360	360	360	359	**357**	355	
DHL Display	360	360	360	360	**FL359**	354	355?

Given that the scan (rotation of radar dish) requires 12 seconds per cycle, rotates clockwise, and the planes are about 90 degrees apart with the BTC coming from the east and the DHL from the south, then if the DHL signal is updated at 35:00, 35:12, 35:24, etc. as the BFU report states, then the BTC data would be updated approximately 3 seconds earlier at 34:57, 35:09, 35:21, etc.

At the 24:35:12 DHL refresh the radar probably showed the location of the BTC at 24:35:09 when it was nearing FL356 and the DHL at FL359. The BTC image would be updated at approximately 24:35:21 or 4 seconds after ATCO completed his expedite instruction, when he was already at the other workstation talking to AEF. Therefore, the claim that the ATCO "considered the problem solved" (p. 76) is not supported by what the BFU report implies was visible on the screen and ATCO's lack of communication with the DHL.

Once again, we see that the events are extremely time sensitive—to the point that radar data itself is insufficient in a collision situation given the 12 second refresh and differential time-delay depending on where an object is in the sky. We can conclude that the delay in radar refresh (i.e., lack of current aircraft location data) must be included as a cause of the collision. Actual data at 24:35:18—just one second after completing the BTC expedite instruction—would have shown both aircraft at FL354.

This analysis illustrates what is required to develop a work practice simulation and how it amounts to understanding the cognitive and physical-perceptual interactions that affect human behavior—and this understanding is necessary to understand

how a human-automation system behaves (or put another way, how people and their tools interacting cause air flight events). In this respect, the purpose of our analysis is analogous to Johnson's (2004b): "Our aim has been to go beyond the existing recommendations and extract any additional lessons that might be learned from this very unfortunate incident." (p. 20)

The deviating statements in the accident reports (Section 6.1), in themselves, show how current analysis methods lead to differing interpretations from same available facts and there is a need for better tools to assist analysis. Our analysis of timing and sequencing in this section shows that classifying events into causal categories (such as HFACS, Section 6.5) shows how a work practice analysis reveals interactions that are omitted, glossed over and difficult to describe in accident reports. The following section presents the Brahms work practice modeling framework we have applied for modeling and simulating the Überlingen work system and events.

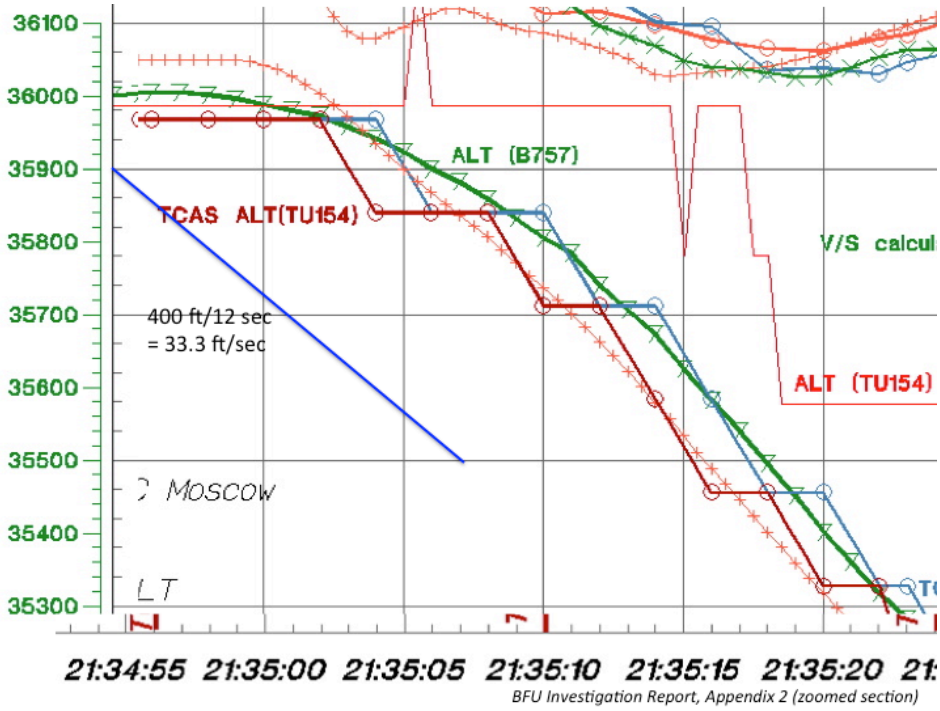


Figure 6-6: Larger excerpt from aircraft altitudes vs. time chart in BFU Report Appendix 6. Initial descent velocity (blue line) has been added for comparison.

7 Brahms Work Practice Modeling Overview

As previously described, the primary requirement of the research program is to develop methods for assessing flight-critical systems with respect to safety implications of the assignment of authority and autonomy. Assessment includes testing that the design is comprehensive, lacking conflicts or ambiguities, and resilient in the context of degradations of agent capability, delegation and or transition of A&A, and the dynamics of interacting roles. The simulation should facilitate what-if analysis of alternative system configurations and designing alternative operational concepts.

We have described in Chapters 1 and 3 the research problem and general properties of Brahms that make it suitable for the research objectives of this project. This chapter provides a general introduction to the Brahms modeling framework and its relevance to verification and validation of work systems and reviews the history of the Brahms tool and how it relates to other human-systems modeling methods. The next major section then describes the design of the Brahms Generalized Überlingen Model (Brahms-GÜM).

7.1 Introduction to Brahms Work Systems Modeling Framework

The Brahms simulation tool was originally developed in the early 1990s by NYNEX and the Institution for Research on Learning to complement business process modeling tools that were based on manufacturing (assembly line) models of work (Clancey et al. 1998). The description here is intended only as a broad introduction, not as a primer on the language. Numerous articles and online documentation describe the model in detail (e.g., Sierhuis et al. 2009).

Most work process modeling tools represent how tasks flow and are transformed; they are oriented around functions of people and automation, represented as input-output relationships: "Job X" flows from agent A (which modified it by function F) to agent B (which applies function G). Brahms turns task/functional models inside out by representing *how agents behave, that is what they do, rather than describing what their work accomplishes.* Behaviors of people and automation, represented as located, time-sensitive activities, that is, where they are, and what they are doing. For example, in Brahms we might model that a person is sitting in front of a computer display reading information and conveying it on the telephone; a functional, process-oriented model might simply model that the person "handoffs the flight to the control tower" without specifying where and how, or how the time required might vary because of the particulars of a given situation. In a simulation of activities, perception (detecting information of interest) and timing are interactive dependent variables.

In the case of people, activities are conceptual; activities represent a person's understanding of "what I am doing now," which is often a form of "who I am being now" (e.g., an air traffic controller working alone on the night shift") or "what I am responsible for now" (e.g., all of the responsibilities of the supervisor in addition to

the ATCO); thus Brahms models can be used to simulation a social world. Brahms simulations also represent what is happening in a simulated physical world (called the "geography") with respect to clock time, which may be simulated at different granularity (one simulation tick could simulate 5 seconds or 5 minutes, etc.).

7.1.1 Beliefs and behaviors of agents and objects

The simulation engine can be said to "run" the model, to create a chronology of behaviors of events. At each clock tick the Brahms simulation engine inspects the model to update the state of the world, which includes all of the agents and objects in the simulated world. Agents and objects have states (factual properties) that may change (e.g., where the ATCO is located; whether a telephone is operational). They usually have capabilities to model world (e.g., ATCO may believe that the phones are operational and then discover that he cannot make phone calls). Facts are "a god's-eye view of the world" and therefore constitute an "objective," true model; an agent's or object's models of the world can be incorrect (inconsistent with the facts) and hence constitute beliefs. (Treating an object's model of the world as being a set of "beliefs" is different from the use of the term "belief" in psychology, but is a convention in artificial intelligence where one might refer to a robot's beliefs.) Consequently, the instruments in the cockpit are modeled as being able to communicate "beliefs" about the state of aircraft. Similarly, what the ATCO reads on a radar display are the "beliefs" of the display. Agents and objects communicate with each other and may act to change their own state, beliefs, or other facts about the world.

For each modeled agent, the simulation engine determines the agent's new or modified beliefs and behaviors, which include inferences (via *thoughtframes* or consequences of *workframes*), *communications* with objects (e.g., reading a radar display) and other agents (e.g., talking); *movement* from one modeled location to another; and other *primitive actions* (i.e., non-decomposed activities that take time, such as changing the radio frequency in a cockpit). For each modeled object (e.g., aircraft), the simulation engine determines the object's state ("facts"; via *factframes*) and new or modified "beliefs" (i.e., representations about the world represented in the object, such as records in a database).

Some objects are not physical things in the world, but rather conceptual entities, called *conceptual classes* in the Brahms language. These represent processes, a complex of people, physical objects, and locations (e.g., flights), and institutional systems (e.g., airlines) that people know about and refer to when organizing their work activities.

Particular "instances" of a conceptual class are called *conceptual objects*. A particular flight (e.g., DHX611, a conceptual object) is operated by a particular airline and consists of a particular crew (a group) of pilots (agents) who file a particular flight plan document (an object), and so on. All of these agent and object instances have behaviors defined by workframes that are inherited from their group (for agents) or class (for objects). That is, behaviors are usually modeled at a

general level and all members of a group/class have the same capabilities (represented as activities, workframes, and thoughtframes).

Of course at any time during the simulation, agent and object behaviors, beliefs, and facts about them will vary depending on their initial beliefs/facts and the environment with which they are interacting. In particular, everyone working in a given ATCC might inherit some characteristics (e.g., knowing how to use the phones), but people playing different roles would inherit behaviors from different groups. The two ATCOs in Zurich ATCC are modeled as having identical activities with the same workframes and thoughtframes by virtue of belong to the same Brahms groups; but at a given time during the simulation one ATCO might be on break while the other is handling flights. That is, their active activities, locations, and facts about them can be different at a particular point in time.

As another example, the **PrimarySurveillanceRadar** (PSR) is an object class that is a simplified model of the PSR in the Air Traffic Control Radar Beacon System. Instances of this class, PSR objects, include particular radar systems located near Zurich and Karlsruhe. Each **PrimarySurveillanceRadar** object has attributes defined by the class (comments follow "//"):

```
AirSector airSector;      // description of sector configuration (conceptual object)
int range;                // range of radar in miles, e.g. 60 miles radius
double altitudeMin;       // floor of radar coverage, e.g. not below 200 feet
int rateOfDetection;      // renewal rate of radar
int displayUpdateRate; // renewal rate for display in seconds
```

When the simulation run begins, the PSR objects have "initial beliefs," for example the time period required for the PSR to sweep the sky is set to 12 seconds:

```
(current.rateOfDetection = 12)
```

Each **PrimarySurveillanceRadar** object monitors planes moving within its particular air space. When planes are detected, the information is sent to air traffic control computer servers associated with the PSR at the start of the simulation (part of the initial configuration that defines a scenario). The ATCC servers then send the information to radar screens in the ATCC. The PSR's behavior is modeled by the **Inform_Plane_Inside_Airspace** workframe:

```
workframe Inform_Plane_Inside_Airspace {
        priority: 20;     // priority determines order for applying the workframes
        variables:        // variables are specific objects or values referenced here
                foreach(Aircraft) plane;
                foreach(Flight) flight;
                foreach(string) flightNumber;
                forone(AirSpace) airSpace;

        detectables:      // these are world facts that the object can "perceive"
```

```
            detectable Plane_Location {
                detect((plane.location = unknown));
            }
            …       // there are similar detectables for longitude, heading, etc.

            detectable Planes_In_Airspace {
                detect((airSpace.numberOfPlanes = unknown));
            }

        // the workframe is applied when the following conditions match the radar
        object's beliefs

        when(knownval(current.airSpace = airSpace) and
            knownval(plane.altitude > current.altitudeMin) and
            …
            knownval(flight = plane.flight) and
            knownval(flightNumber = flight.flightNumber))

        // applying the workframe results in the object updating its model of the world
        (beliefs) and doing two "composite activities" (modeled by their own
        workframes), namely communicating the information to registered radar
        servers (informATCSystemPlaneInfo) and then updateRadarScanRate,
        described below

        do {
            conclude((airSpace planes plane));
            conclude((plane.flight = flight), fc:0);
            conclude((flight.flightNumber = flightNumber), fc:0);
            informATCSystemPlaneInfo(plane);
            updateRadarScanRate(plane, airSpace);
        }
    }//wf Inform_Plane_Inside_Airspace
```

The composite activity **updateRadarScanRate** defines how often the plane's information is updated within an air sector on the radar display. For example, suppose that there are three planes within an air sector and the radar scan rate (sweep) for the air sector is 12 seconds. Then each plane's information will be updated on average every four seconds (12 seconds divided by 3 planes). A more precise model would be based on the actual layout of the planes; this heuristic assumes they are evenly spaced (e.g., three planes are 120 degrees apart). If there is only 1 plane in the air sector, then the plane's information will get updated on the radar display every 12 seconds.

This example illustrates how the Brahms modeling framework enables modeling and hence facilitates designing work systems at varying levels of detail. For example, in an initial model a work group might be represented as a single person (agent); for example, Brahms-GÜM currently represents the aircraft crew by a single person, the pilot. By such an abstraction we would seek to replicate (most

importantly) how the aircraft is controlled—that is the overall effect of the crew's behavior—rather than how through the communications and actions of individuals observations, decisions, and controls/instruments were manipulated.

An object is typically modeled only in terms of the properties that a person (or an automated system such as a robot) can perceive (a "detectable") and/or modify. The radar display object could be simulated in greater detail to emulate when an update would actually occur given the aircraft's location in the sector. In particular, a computer model could be coupled to the radar display object to update its facts/beliefs according to a precise mathematical simulation.

A Brahms model is developed and tested incrementally by adding or refining agents, objects, and locations. Conceptually, every agent and object is an independent process: it simply behaves, interacting with its environment, which is also changing over time. This modeling and simulation flexibility facilitates verifying and validating proposed concepts and configurations early in the design process, as well as finding design problems in complex systems when they are probably easier to fix.

7.1.2 Relation of work and reasoning

In the human-centered perspective, the notion of *work* focuses on what people are doing, that is, their *activities* (how they conceive what they are doing in levels of abstraction ranging from broad, identity/role-oriented to specific, task-oriented terms), their *beliefs* about themselves and the world (including other agents), and *how they are interacting with the world* (what they are perceiving, saying/writing, physically manipulating, etc. and how they are moving).

Although Brahms incorporates representational constructs and processes from the methods of cognitive modeling (e.g., beliefs, inference [thoughtframes], and conditional actions [workframes]), *reasoning* is simulated as just another activity in Brahms: Thinking about something takes time, occurs in some conceptual and physical context, and often involves interacting with objects in the world in a back and forth process of manipulating, looking, and changing beliefs (as in using a radar display to get information about a flight). In a typical cognitive model, agents are modeled only in terms of beliefs and how "inputs" change beliefs (like a calculating machine). In Brahms beliefs form an important basis for action, but they are contained within a contextual model of a changing world and the agent's activities. Beliefs determine what workframes will apply and hence determine an agent's behavior and perceptions, but the organization and focus of the model is in terms of what the agent is doing, not only what he or she is thinking. Put another way, thoughtframes are used to model reasoning, but reasoning is not the driving engine of simulated behavior; rather reasoning serves at the periphery and to affect interpretations, choice of methods, and prioritization of activities.

7.1.3 The work system

In a Brahms model, the *system* being modeled is the entire work system, including:

- people, represented as agents and groups to which they belong
- facilities (buildings, rooms, offices, spaces in vehicles), represented as "areas" within a geography
- tools (e.g., radio, radar display/workstation, telephone, vehicles), represented as objects
- representational objects (e.g., a phone book, a control strip), represented as objects
- automated subsystems (e.g., TCAS), represented as object or agents if they are controlled by activities and internal model of the world (e.g., robots).

All of these are located in an abstracted geography represented as *areas* and *paths* (for simulating movement between locations). Thus the notion of "human-system interaction" in Brahms terms is more precisely a combination of interactions among behaviors of agents and objects in a work system.

As an example of how workframes model the interaction among an agent's beliefs, perception, and actions in a dynamic environment, consider how a pilot deploys the aircraft landing gear. A pilot uses the onboard landing control and then confirms that the landing gears are deployed while monitoring the aircraft's trajectory on Primary Flight Display. This is modeled in Brahms-GÜM as follows: a pilot (e.g., the DHL pilot) is a member of the **PilotGroup**, which has a composite activity, **manageAircraftEnergyConfiguration.** Before landing, the following workframe (whose name comes from the Pritchett et al. WMC simulation described in Appendix 17) becomes activated:

```
workframe Confirm_Gears_Changed {
        repeat: true; // this workframe is repeatedly activated, which models the
        pilot's continuing monitoring of the pfd until the gears are deployed
        variables:
                foreach(AircraftLandingControl) control;
                forone(PrimaryFlightDisplay) pfd;

        // when the action ("do" part) of this workframe is being applied, the pilot can
        detect the following fact in the world and thus form a corresponding belief

        detectables:
                detectable Plane_Gears {
                        detect((plane.gearsDeployed = unknown));
                }

        // for this workframe to apply, it must be the case that the plane has not yet
        touched down and is not stopped (i.e., flying, not preparing for take-off), the
        pilot is in the control.location of the cockpit (inclusion of the Primary Flight
        Display location is to facilitate future modeling of co-pilot; it is unnecessary in
```

```
        a model with one pilot), and the Aircraft Landing Control does not already
        indicate that the plane's landing gear is deployed

        when (knownval(plane.touchdown = false) and
                knownval(current.location = control.location) and
                knownval(current.location = pfd.location) and
                knownval(control.isGearsDown != plane.gearsDeployed) and
                knownval(plane.airSpeed > 0))
        do {
                readPrimaryFlightDisplay(pfd, plane, 1, 3);
        }
}//wf Confirm_Gears_Changed
```

7.1.4 Scenarios

A Brahms simulation model *configuration* consists of the modeled geography, agents, and objects as well as *initial facts and beliefs* of agents and objects. For example, the departure time for a flight might be an initial fact about that flight. One can modify the model for different departure times to define different *simulation runs*. That is, the same model may be run with different configurations to perform a what-if analysis.

Initial facts may include *work schedules*. For example, an air traffic controller might be working alone in the ATCC at a certain time in one configuration, but two controllers might be working together at different workstations in another configuration.

Initial beliefs of an agent might be broad preferences affecting behavior (e.g., "TCAS should overrule the ATCO"), thus initial beliefs can be used as switches to easily specify alternative configurations of interest.

Alternative Brahms model configurations are called *scenarios*. Thus for example, a scenario might be a variation of the Überlingen collision in which two aircraft have inter-route flight times that put them on an intersecting path over Überlingen, there is a late arriving flight for Friedrichshafen, and maintenance degrades the radar—but the telephones are operative. Or the phones might be inoperative but the STCA Optical alert functions normally.

Generally speaking, a Brahms model is designed by the model builder with sufficient flexibility to allow investigating scenarios of interest. The set of "causal factors" of interest (e.g., use of control strips when approving aircraft altitude changes, availability of telephones) constitute states of the world and behaviors that can be configured through initial facts and beliefs. The initial settings define a *space of scenarios*. Using Brahms to evaluate designs within this space, while using formal methods to help modelers understand its boundaries so they can refine the model to explore alternative scenarios, constitutes the main research objective of this project.

By configuring the simulation model to represent variations in the environment (e.g., ATCC tools, flights) and basic agent behaviors (e.g., following TCAS or the ATCO) we can simulate what happens in different scenarios. These "outcomes" (a chronology of system states and events) are actually predictions of how a particular set of events (here the Überlingen collision) might play out differently if different combinations of the known causal factors were present. Besides allowing a variety of scenarios, Brahms-GÜM is general in the sense that the components can be adapted and reconfigured for entirely different situations, such as a very different set of flights, revised systems (e.g., TCAS 7.1, which can reverse instructions to pilots), agents following different work practices (e.g., an air traffic controller who would have told AEF to call the tower after two failed phone attempts), etc. In this manner, the model that was developed for exploring variations of the Überlingen scenario can be adapted (edited and/or elaborated) to represent and assess different work system designs related to air transportation systems.

In summary Brahms-GÜM is not a simulation of a particular accident or restricted to understanding variations of the Überlingen scenario. The models of ATCC, ATCO, flights, aircraft, radar, etc. can be reconfigured and populated to simulate different air sectors and flight combinations. In effect, Brahms-GÜM can be viewed as providing a library of agent and object models that can be used and extended for different purposes in NextGen research.

7.1.5 Roles and responsibilities

Assignment of responsibility among people and automated systems is another form of flexibility of particular interest for AFCS research. Ideally, the Brahms model should be designed to enable flexibility by which a given agent/system can have more than one role/responsibility at a given time and these can change (be reassigned) during operations in a situation-dependent manner. More specifically, a person/system has more than one role/responsibility at the same time and different roles/responsibilities at different times during the simulation. During operations, people and automated systems behave independently, in parallel, enacting their different roles/responsibilities.

In a Brahms model roles/responsibilities are represented by group membership, which can be used to model defined roles (job position/functions) and activities. The Zurich ATCC on the evening of the accident was staffed by people carrying out the roles of the RE (Radar Executive), RP (Radar Planner) and CA (Controller Assistant). In the accident report, the controller is called an ATCO (Air Traffic Controller), who is carrying out the function of RP and RE simultaneously. To simulate how the Zurich ATCO is left alone to work two workstations, he is modeled as initially belonging to two groups. When the other ATCO goes on break, he communicates that he is leaving and reconfigures the A RE (right) radar display. The remaining ATCO then concludes that he is working at both workstations and that he is responsible for following the Friedrichshafen STAR procedures. Thus this ATCO will follow the behaviors required by two people working these two workstations, having to share his time and attention between them.

Here is the same description expressed in the Brahms language:

> To simulate how the Zurich ATCO
> (**Brahms Agent ATCO_Zurich_RP**)
> is left alone to work two workstations
> (**Brahms Areas WorkstationArea_Zurich_ARTCC_A_RE WorkstationArea_Zurich_ARTCC_S_RE**)
> he is modeled as initially belonging to two groups:
> (**Brahms Groups RadarPlannerGroup AirTrafficApproachControlGroup**).
>
> When the other ATCO
> (**ATCO_Zurich_AR_RE**, member of **AirTrafficApproachControlGroup** working at **WorkstationArea_Zurich_ARTCC_A_RE**)
> goes on break, he communicates that he is leaving and reconfigures the A RE radar display
> (**ATC_Display_Zurich_ARTCC_Arfa_Sector**)
> so that its **airSectors** property is **AirSector_Zurich_East_ARTCC**.
>
> The remaining ATCO (**ATCO_Zurich_RP**) then concludes that he is working at both workstations and that he is responsible for following the Friedrichshafen STAR procedures (i.e., believes that his **airportSTARs**, the procedures to follow, includes **STAR_EDNY_RWY_24**).

7.1.6 Tool and environment advantages

More broadly, the Brahms work practice modeling framework has several characteristics pertinent to air transportation systems simulation objectives:

- **Generality—**as explained above, the object-oriented design of Brahms models makes them amenable to reuse, such that one can develop a "simulation toolkit" for an application area. The models of aircraft, cockpits, radio, telephone, ATCCs, pilots, etc. developed in the Brahms-GÜM can be packaged for use by other research teams to develop ATS simulations.

- **Variable Detail and Simulation Coupling—**behaviors can be simulated coarsely, as in the example of the radar display above; object facts/beliefs can also be simulated by coupling the Brahms model to another simulation (or with appropriate APIs to an actual hardware/software system).

- **Analytic Metrics**—a *conceptual object* (e.g., a flight) can be tracked during a simulation such that statistics involving agents, activities, and time can be recorded. For example, it would be possible to record the percentage of time an ATCO is working on different workstations during a simulated period. Such data measure properties are of interest for evaluating a work system design (e.g., statistics about aircraft separation) and/or to justify further

model refinement or human-in-the-loop experiments with a proposed automation system.

- **Visualization**—the Brahms modeling environment includes the AgentViewer, a tool for displaying events in a simulation run on a timeline, such that interactions between selected objects and/or agents can be identified and analyzed. Collapsing the timeline allows visualizing aggregate patterns of interaction, such as recurrent agent/object communications (see Appendix 25 and Figure 22-4 in Appendix 22.5).

7.2 Relation of Brahms to Other Modeling Frameworks

A wide variety of ATS research relates to the present project, including approaches to studying and modeling work (e.g., cognitive task models), analyzing and formalizing system interactions (e.g., resilience engineering), and experimenting with prototype systems (e.g., human-in-the-loop airspace simulations). The present project is effectively an experiment with an established approach, work practice analysis and simulation, in particular to evaluate Brahms suitability for simulating complex human-automation systems in safety-critical situations.

This report details one year's activity on an effort that would naturally require at least three to five years to reach maturity. Because of the great amount of detail required to convey the analysis, simulation, and results, no effort is made here to comprehensively survey and compare the existing literature in air traffic systems, human factors, formal methods, or any of the related disciplines (e.g., psychology, organizational theory). The research and writing of such a report would merit a separately funded project.

Instead our investigation of related research has focused on the significant body of scientific publications that explicitly cite the Überlingen collision and have been valuable for understanding the circumstances of the collision and how to analyze it (Chapter 6). On the other hand, alternative modeling and simulation approaches can be helpful to articulate and contrast the methods and benefits of work practice simulation. Rather than providing a complete literature review, we mention some related work to highlight the theoretical and practical contributions of this project.

The explanation of work systems modeling in this chapter to this point has already highlighted how modeling and simulating activities as chronological, located behaviors in a simulated environment is fundamentally different from task-functional modeling, the modeling method that is most common in large-scale human-system models (e.g., see Air Force Office of Scientific Research Software and Systems research program). The most common multiagent simulation frameworks enable creating task-function models (or runtime programs) grounded in a logic formalism. These logic-based tools are directed at designing software agents whose behaviors are optimal with respect to formal definitions of rationality, information, economics, etc. These tools emphasize modeling and design of automated

processes, in contrast with the objective of Brahms to simulate *practice*, how people actually behave.

The following sections briefly review other Brahms simulations relevant to the present modeling effort, cognitive models of air traffic control, Reason's "Swiss cheese" accident model, the NextGenAA agent-based language, the approach of aviation safety problem analysis.

7.2.1 Hybrid Brahms Simulations

As illustrated by the incorporation of a model of TCAS in the Brahms-GÜM, the Brahms framework is designed to enable simulating automated systems and thus representing how people might behave when using or directed by automated systems. More generally, Brahms' framework enables incorporating other simulations or even actual software systems within a Brahms model:

- In the Brahms-OCAMS simulation (Clancey et al. 2008), a simulated backroom flight controller in Mission Control Center of Johnson Space Center interacts with Microsoft Office tools including Excel™ and Word™ that are integrated in the Brahms model through Microsoft Office APIs.
- An existing simulation of an air recycling system coupled to an automated air system control system was integrated with a Brahms model; the interface used by a simulated astronaut was modeled in Brahms with properties changed by the underlying automated system (called Brahms-CONFIG).
- In Brahms-VE a Brahms simulation of astronauts living and working in a Mars habitat was coupled to a virtual reality simulation of the habitat and astronauts, such that movements and gestures of the simulated astronauts were driven by the Brahms simulation and represented as a kind of dynamically constructed cartoon (Clancey et al. 2005).

Certainly the most relevant previous work is Brahms-FACET, in which Brahms was coupled to FACET to simulate how airlines might directly affect traffic flow management (Wolfe et al. 2008). The project explored alternative uses of the Brahms framework with respect to performance for a large-scale simulation with many flights. Brahms-FACET used Brahms to provide a simulation context for integrating FACET with algorithmic route selection methods, rather than modeling human reasoning and practices.

Given the focus on modeling flows in defined air traffic routes, the Brahms-FACET model did not simulate the work practices of pilots or air traffic controllers:

> The controllers themselves were modeled only as a constraint, i.e., the number of flights that could follow a particular air route…considered as a route capacity….[The model] assigned one controller per route and did not model sectors. (Wolfe et al 2008, p. 3)

The Traffic Management Unit was modeled as a single agent; modeling pilots was determined to be unnecessary given the focus on decision-making involved in planning by the TMUs.

The experience in developing the Brahms-FACET model exemplifies the flexibility of the Brahms language and importance of scoping the model's design to focus on what questions the model is intended to answer:

> Great care was taken to only represent the airspace components that were truly needed for agent decision making, and such components were represented in the most abstract (i.e., compact) form possible. In some cases, such as the identification of the demand-capacity imbalance, the processing was done in FACET (and supporting Java code) and *only the outcomes were represented in the corresponding agent's belief model.* (Wolfe et al. 2008, p. 6)

7.2.2 Cognitive models of air traffic control

Other work in ATS simulation has demonstrated the feasibility for formal models of human interaction with automation, but usually at the local, one-person–one-procedure–one-interface level (e.g., an ATCO or pilot(s)/crew, but not both interacting). Examples include (Callantine 2001, 2005a,b; Corker et al. 2000; Leuchter and Jürgensohn 2000; Pompanon and Raufaste 2009; Pritchett and Feigh 2011). A more complete study of related work would compare and contrast the methods of Degani, Rushby, Freed, Leveson, Feary, Sherry, Palmer, and others.

Cognitive simulations can be quite detailed in represent mental processes involving attention, planning, and inference. However, the notion of "activity" in these models corresponds only to mental activity that involves information (e.g., "a perceptual activity"):

> Agents transform their belief set by performing activities, in accordance with the theory that all salient operator activities in complex human-machine systems involve transforming or communicating contextual information (Callantine, 2005).

In contrast, in Brahms activities are what people do, such as moving to another workstation, looking for a control strip, saying something out loud in a room, and sleeping in a break room. Thinking about something is also an activity—it occurs in some setting, takes time, and often occurs while using representational objects (e.g., consulting a manual). The work of an ATCO involves more than transforming or communicating information and those that do, such as using a telephone require movement and take time. Indeed, it is through located interactions in the world over time that an ATCO discovers what information needs to be read, interpreted, recorded, communicated, etc. These caveats aside, it might be useful in future research to adapt and incorporate the Crew Activity Tracking System (CATS) activity model to create a more complete ATCO simulation than limited resources have allowed in developing Brahms-GÜM.

By focusing on the broader context, Brahms-GÜM demonstrates that the Brahms simulation tool has the potential to extend formal analysis of the ATS to a distributed set of agents, including people and automated systems, acting independently, cooperatively, or even competitively. This is of interest because it extends the state-of-the-art and is directly relevant to NextGen goals.

7.2.3 Reason's "Swiss cheese" accident model

Different "accident models" can serve as frameworks for analyzing and explaining the events in an accident. As detailed in the review of Normal Accident Theory (Chapter 5), this report applies a systemic perspective, which admits multiple, non-linear, dynamic "causes." Prevalent alternative perspectives adopted by analysts include sequential, epidemiological, and event-based (root-cause) frameworks (Shappell and Wiegmann 2000).

One particularly common metaphor that appears in aerospace accident reports is the "Swiss cheese" model that integrated and elaborated Reason's (1990) framework into the causal sequence analysis.

> It is generally accepted that like most accidents, those in aviation do not happen in isolation. Rather, they are often the result of a chain of events often culminating with the unsafe acts of aircrew. Indeed, from Heinrich's (Heinrich, Peterson, & Roos, 1980) axioms of industrial safety to Reason's (1990) "Swiss cheese" model of human error, a sequential theory of accident causation has been consistently embraced by most in the field of human error. Particularly useful in this regard has been Reason's (1990) description of active and latent failures within the context of his "Swiss cheese" model of human error. (Shappell and Wiegmann, p. 2)

However, it is somewhat unfair to blame (or credit) Reason (1990) for the sequential theory of accident causation. Reason depicted levels or *dimensions of analysis*—organizational influences, unsafe supervision, preconditions for unsafe acts, and unsafe acts of operators. Shappell and Wiegmann similarly emphasize the notion of analytic dimensions or perspectives in the HFACS framework (Section 6.5).[21] Although Reason depicted the dimensions as series of squares (p. 208, Figure 7.8), giving rise to the Swiss cheese metaphor, he appears to emphasize instead that these are "levels" or "layers" of analysis, emphasizing that causes were *systemic*, not all located in one set of players or setting and having a historic, cultural nature. The caption where he introduces the metaphor refers to "the chance of such a trajectory of opportunity finding loopholes in all of the defences at any one time" (p. 208), emphasizing the checks and balances that operate on many levels and across time in the organization, rather than a specific chain of events.

[21] Indeed, Wallace and Ross (2006) argue that Reason's model, which is based on deterministic causes, has the same fundamental weakness as Heinrich's (1959) "domino" theory: "there is one set root cause that triggers another and then another until the accident happens" (p. 26).

As Shappell and Wiegmann observe, Reason's conceptual framework is often represented as a sequence of events, implying that accident causes are lined up like dominos, one falling into the other. For example, in Figure 7-1, published in a June 2012 NASA e-book (Merlin et al. 2012), Reason's analytic levels are depicted and described as a "mishap chain." The arrow represents time and the holes suggest weakness or gaps in a single system level. This diagram confounds an analysis based on different systemic perspectives (occurring at different organizational and temporal scales) with an analysis of an accident as *temporally ordered causes*—a sequence of events.

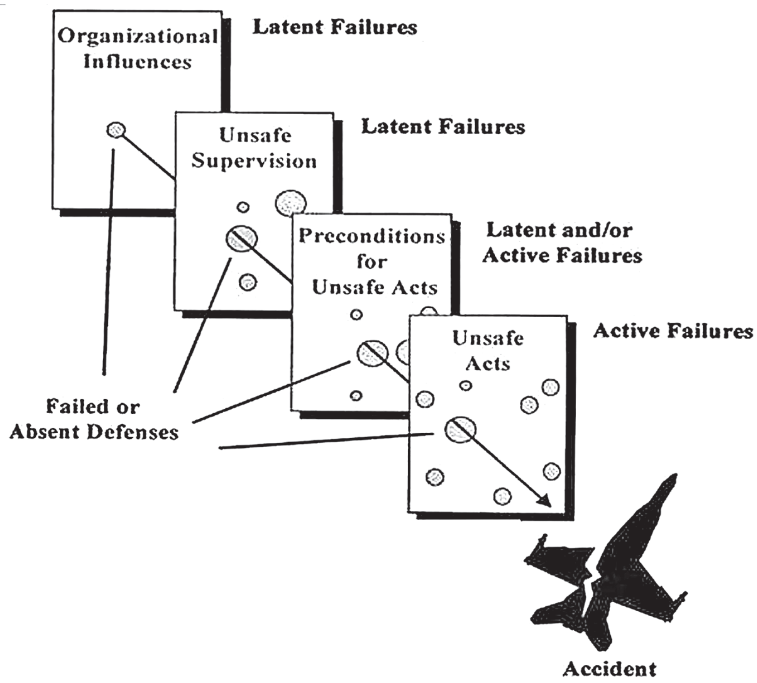


Figure 7-1: Reason's four dimensions portrayed as a "chain of events" leading to an accident (Merlin et al. 2012, Breaking the Mishap Chain, p. iv)

The implication of such diagrams is that if one event is missing, akin to moving a slice of cheese, so "the holes of vulnerability" (Merlin et al. 2012) do not line up in the same way, then the accident would not have occurred:

> Organizational safety researcher James Reason likens the layers of supervision and management in an organization to slices of cheese—specifically, Swiss cheese. The holes in each slice of the cheese represent areas of safety vulnerability in the context of operations. When the holes are small and out of alignment with one another, safe operations ensue. But when the holes in the various layers line up, the "accident arrow" is allowed to pass through, resulting in an accident. (p. 141)
>
> Such accidents and incidents rarely resulted from a single cause but were the outcome of a chain of events in which altering at least one element might have

> prevented disaster…even the most qualified individuals can become links in the mishap chain (pp. xiv-xv)

Pictures like this may be easy to remember and have some pedagogical value, but they may lead analysts astray by suggesting that an accident can be understood as a simple sequence of events. Of course "things happen" at certain times as an accident unfolds. Indeed, the narratives we construct in telling the story of an accident are almost always chronologically ordered—it is one of the defining properties of narrative.

The "narrative presupposition" is the assumption that the order of clauses in the narrative can be taken to mirror the order of events in some postulated real world.[22] Consequently, any presentation of events is likely to be interpreted as a narrative sequence. This can become a problem for understanding events in complex systems because a "timeline of events" is mostly irrelevant to understanding *complex interactions* occurring across dimensions (e.g., organizational vs. electromechanical) and among subsystems at the same level of analysis (e.g., two ATCCs and two cockpits).

In particular, the "holes" such as missing tools, automated processes, assistance from others, etc. may normally inter-operate, such that for example STCA Optical alert and phones are part of the ATCO's support system. Removing both is like introducing two "holes" in the same slice of cheese. Such interactions at the same level of analysis/description are fundamental in understanding how situations become complex. But the "mishap chain" metaphor suggests that we are most interested in "holes" that "link" organizations, supervision, and controller actions. In fact, the supervisory actions are occurring within the same work system as the controllers. What is really needed here—and that Reason's model emphasizes—is understanding the work system with respect to all of the players with different roles at other times and places, not just the "operators" on the line when an accident occurs. Talk about holes and slices doesn't offer much insight into such distributed practices, particularly those involving different authority and responsibility (e.g., the relation of the maintenance team to the Zurich ATCC).

The further categorization of anomalies as being "lost defenses" (Merlin, et al. 2012, p. xiii) again takes the metaphor of "accident barriers" too literally. Most tools and practices (e.g., control strips, telephones) are not "defenses" they are simply aspects of how the work is done. A missing alarm may be construed as a failed defense, but missing phones constitute a loss of workflow/coordination paths. Failure to do a task (e.g., perhaps ignoring AEF calls in the face of the DHL/BTC separation violation) is not in itself necessarily an "unsafe act."

[22] See Linde (1993, p 68) discussion of Labov's (1972) analysis of the discourse structure of narrative.

In reality, as we saw in the NAT discussion of Überlingen (Section 6.8.1), different processes are operating in parallel, simultaneously and coming into the interaction in unforeseen ways in the moments prior to the collision. System states may interrelate and become part of a network of causes. For example, it is possible that the ATCO might have intervened soon after the TCAS TA, such that TCAS would adapt its instruction to the BTC's maneuver to advise them to descend. By intervening seconds before the TCAS RA, ATCO's instruction might have come at the worst possible time relative to the operation of the TCAS algorithm, creating a situation for the pilots that required special training to react to correctly.

People will naturally construct a narrative that inherently linearizes a network of causes; this is a consequence of the narrative presupposition discussed above. The problem is that the choice of "events" to recount may be strongly governed by the necessity of story structure and telling a good story (the exigencies of rhetoric). Stories are good at highlighting heroes and villains, but insufficient and potentially misleading for expressing the interrelation of multiple, simultaneous causes. Within a narrative, one "chain" will tend to be highlighted over others. For example, by a retrospective analysis otherwise acceptable practices (e.g., not directing BTC or DHL to another altitude when they first contacted Zurich ATCO) may be viewed as "events" (failures to do a task) that contributed to an accident.

In some respects the "chain of events" diagrams codifies the perspective of hindsight bias by which the sequence of events appears incredible. Woods (2005, p. 306) explains how hindsight bias hinders accident analysis, such as in the Columbia Accident Investigation Board report (2003) which he was critiquing:

> Hindsight is not foresight. After an accident, we know all of the critical information and knowledge needed to understand what happened. But that knowledge is not available to the participants before the fact. In looking back we tend to oversimplify the situation the actual practitioners faced, and this tends to block our ability to see the deeper story behind the label human error.

Reconstruction after the fact leads to an oversimplification, which the Swiss cheese diagram exemplifies—failing to see the interaction of multiple factors, how normal practices make safety/production tradeoffs, and the mindset by which systematic vulnerabilities are dismissed.

Leveson (2004) and Qureshi (2007) provide excellent reviews and critiques of linear, "chain of events" models. Qureshi says:

> Sequential and epidemiological accident models are inadequate to capture the dynamics and nonlinear interactions between system components in complex socio-technical systems. New accident models, based on systems theory, classified as

> systemic accident models, endeavour to describe the characteristic performance on the level of the system as a whole, rather than on the level of specific cause-effect "mechanisms" or even epidemiological factors (Hollnagel 2004).
>
> A major difference between systemic accident models and sequential/ epidemiological accident models is that systemic accident models describe an accident process as a complex and interconnected network of events while the latter describes it as a simple cause-effect chain of events. Two notable systemic modelling approaches, Rasmussen's (1997) hierarchical socio-technical framework and Leveson's (2004) STAMP (Systems-Theoretic Accident Model and Processes) model, endeavour to model the dynamics of complex socio-technical systems.

Leveson' frames the systemic view this way: "The most effective models will go beyond assigning blame and instead help engineers to learn as much as possible about all the factors involved, including those related to social and organizational structures." She argues for "an accident model founded on basic systems theory concepts" (p. 1).

An analysis of a mid-air collision in Brazil by de Carvalho et al. (2009) applies Normal Accident Theory and the efficiency-thoroughness trade-off (ETTO) principle of Hollnagel (2004) to illustrate how ordinary system variations may not be perceived as signals or problems in the period preceding an accident, but rather are reinterpreted and chained together in hindsight:

> We note that the Brasilia controller only informed about the communication frequency. He simply did not mention any of the ''abnormal'' (at least in hindsight) indications they had on the radarscope. At this moment, he (or his supervisor) probably had already perceived the indication changes, but he did not think that these indications were important enough to be communicated to the fellow controller. *This brings the important question regarding how the system functions daily.* If the situations uncovered by this accident, like loss of radar contact, communications difficulties, and level variations due height-finding radar inaccuracy are frequent enough in a way that ''abnormal'' indications were being considered ''normal,'' then the ETTO principle and associated heuristics (these things always happen, it is not important to act now, the system is always changing symbols) function as an important factor for the construction of cognitive strategies. In this situation, we cannot attribute the cause of the accident to a chain of human errors. Doing so, we will be blind to address the real safety threats throughout the ATC system functioning. (pp. 338-339, *emphasis added*)

We also note that official accident investigation reports furnish the primary data for most accident histories. The main function of these reports is to establish the legal liabilities in particular accidents, not to provide all the data required for scientific analyses. We have already discussed a number of questions raised by the Brahms modeling process for which the BFU report does not furnish the data. Other analyses would doubtless find other omissions.

7.2.4 NextGenAA agent-based language

NextGenAA is a complementary NASA research project within the Authority and Autonomy theme. Comparing and contrasting the approaches helps elucidate the nature of agents and work system modeling in Brahms.

General objectives of the NextGenAA project (Bass et al. 2010a, b, 2011b) are similar to the Brahms AFCS project:

- To develop a unified agent modeling language which can be used by both agent based simulation and model checking to model autonomy
- To develop and demonstrate a computationally tractable approach that allows formal model checking techniques to verify the bounds of human behavior and accountability in a distributed human-automation system
- To develop and demonstrate an integrated tool-suite through real-world concepts under consideration for inclusion in Next Generation Air Transportation Systems via scenarios involving substantial human-automation interaction.

Table 7-1 indicates how constructs in the NextGenAA framework can be represented in the Brahms language. Brahms constructs and/or semantics that appear not to be included in the NextGenAA framework are highlighted in yellow (see Section 7.1 for overview of Brahms).

Table 7-1: Relation of NextGenAA agent framework to Brahms language constructs. Yellow highlight indicates model constructs in the Brahms language that are not distinguished in NextGenAA.

NextGenAA	BRAHMS
Actions (atomic)	Primitive Action (Behavior) Communication Action Movement Action
Values: Effects of actions (turning a knob)	*Changed Beliefs* represented in Conclude part of Workframe *Detectables:* Perceived facts in the world, probabilistically become beliefs.
Activities: Composition of actions with varied control (procedure vs. alternatives)	Composition of *workframes* (conditional actions) Semantics of activities: Conceptual, represent behaviors, correspond to agent's conception of identity/role

NextGenAA	BRAHMS
Communication	Synchronous communications are modeled by explicit handshake in the interaction. Agent can only communicate beliefs. Objects represent facts about the world as beliefs (e.g., a database). Changes in object state are modeled as facts. Objects can be *contained in* other objects (e.g., a photograph in a camera; a file in a directory)
Activity Abstraction	Composition of *groups* having activities consisting of workframes *Inheritance*: Subgroups and agents in groups inherit behaviors by membership.
Scenario: Agents, Systems, Environment	Scenario: Alternative *work systems* in alternative environment, consisting of *geographic layout, facilities, vehicles, tools, instruments, documents, etc. (world facts).*
Goal Annotation on Activities: *Safety Goals*: expressible as temporal logic *Mission Goals* (progress): Not verifiable, summary of purpose of action	<not included in Brahms language>

The focus of the NextGenAA is constructing models that can be verified using formal methods. Bolton et al. (2010) describe limitations of other task analysis languages that they sought to rectify:

> The power of these formal verification analyses is limited by the ability of the task analytic modeling notations to express normative human behavior. For example, CTT [9] and Field's task modeling notation [20] do not support all of the temporal and cardinal relationships between activities and actions (referred to as sub-acts below) of other task analytic modeling notations.

They then summarize how the logical relationships of sub-acts can be specified to control how activities or actions in a decomposition hierarchy execute. These are combinations of the *number of sub-acts* that must execute for the parent activity to finish (zero, one or more, all, exactly one) and their *temporal relations* (one at a time, (possibly) concurrently, in specified ordered, synchronously).

Applying a task analysis framework, the NextGenAA agent-based language models work systems from a functional perspective, as opposed to the behavioral-interactional descriptions in Brahms. As outlined in the table, Brahms models represent agents as having actions conditional on their beliefs; possible actions and initial beliefs are defined by group memberships. Agents move, perceive, reason, and change the independently modeled state of the world (consisting of simulated objects in modeled geography).

By abstracting the work system, a task analytic perspective does not model how perception, reasoning, action, and environment are related, in particular how agents and automated systems interact and may become co-dependent through circumstances. For example, in practice perceiving something often depends on being in a certain location (e.g., moving in a cockpit to read an instrument); communicating involves using tools (e.g., telephones) that are also located (requiring movements), and object behaviors need to be modeled to simulate insofar as they become the "environment" for agent activities. Similarly, reasoning often involves interacting with objects (e.g.,. control strips) and is carried out in an ongoing activity with circumstantial layout and temporal interactions (e.g., needing to move a chair to monitor two ATCC workstations). Often what you perceive affects what you want to know more about and hence direct both reasoning and action (e.g., to get, confirm, record information).

Because of the Brahms approach to modeling work systems as independently behaving objects and agents, the language and engine is based on an object-oriented, modular approach. Consequently, the agents, systems, objects (e.g., representations in the world), and the environment are all reconfigurable in a manner that minimizes dependencies. In particular, Brahms-GÜM includes models of aircraft subsystems including cockpit instruments (radio, FMS) and ATCC subsystems (e.g., radar) as well as what the people are thinking, perceiving, and doing.

This modular approach means that timing of interactions among agents and subsystems is usually emergent in the simulation. Although other frameworks (e.g., Pritchett's WMC) use discrete event simulation like Brahms, they may model the "next action" in a programmatic way (similar to Brahms primitive activities), rather than having the timing determined dynamically during the simulation by interactions among independently modeled people and systems in a simulated world.

It should be emphasized that Brahms was not designed for model checking, but emphasizes fidelity useful for detailed interactional design where temporal and spatial relations are most important, as occurs during the ATCO-TCAS-Pilot interactions during the Überlingen accident. The NextGenAA project aims to bring together task analysis and model checking in a manner informed by the needs of each other. Therefore, what constitutes an "agent" and an "action" is different in these two frameworks and should be assessed with respect to each project's goals. It may be determined for example that the fidelity provided by Brahms in modeling complex human-system interactions is too complex for present-day model-checking methods used in software engineering. Perhaps then the advantages of the work practice model might drive advances in model checking. The point of the Brahms AFCS project is to demonstrate applicability and advantages of an existing agent-based modeling framework, and then to determine the implications and best application of model-checking methods.

In summary, in this project we aim to use model checking as a tool for developing, refining, and applying simulation models—with the emphasis on using simulation as a means (a technique) for understanding and designing work systems involving complex human-automation interactions. We focus on characteristics of work systems that we wish to model and understand, determine the strengths and weaknesses of the Brahms simulation framework in this regard and subsequently how model-checking might enhance strengths and resolve some of the weaknesses. That is, the objective is not primarily a matter of "checking" the Brahms simulation, but using model checking to: 1) develop better/appropriate simulation models by indicating gaps, assumptions, lack of generality, or lack of flexibility for exploring some subspace of scenarios, 2) generate scenarios or through formal analysis provide scenario outcomes without running the model, and 3) construct a tool kit for scientifically understanding the behavior in human-automation systems and formulating principles for work system design. These points are taken up again in the discussion of future work, Chapter 12.8.

7.2.5 Aviation safety problem analysis

Improvements to automation after ATS failures (Kochenderfer 2012a) reveal that because of uncertainties in the real world (the nature of a system with complex interactions) the use of a deterministic model, particularly one that excludes human behavior, to evaluate and certify automation is insufficient because "pilots do not always behave as assumed by the logic" (p. 18). Consequently the "spectrum of responses" is not completely predictable, limiting the robustness of systems like TCAS, as demonstrated by the collision over Überlingen, explicitly mentioned by the Lincoln Laboratories report (p. 18).

The response of the automation designers, as exemplified by the history of TCAS, is usually to enhance "the surveillance and the advisory logic" (p. 19) of the automation. But the new monitoring and advising algorithms (e.g., ACAS X) continue to be designed and evaluated only within the often implicit assumptions of an artificially closed world. The TCAS design problem is described as "specifying an encounter model and using computational methods to find the logic that optimizes performance against a set of metrics" (Kochenderfer et al. 2012b, p. 29).

Such models and analyses fail to acknowledge that the *function* of systems like TCAS is to *convince* the pilot to change course. From the perspective of engineering, the matter is often couched as "TCAS has authority." But from the perspective of pilots in an uncertainly, highly risky situation, the situation is more like, "I know I'm supposed to obey TCAS, but do I believe its appraisal is correct and its recommended action is safe?" As recently as February 2011, the FAA reported that "unnecessary deviations from ATCO clearance continue to be an important factor in the effectiveness of TCAS," (FAA 2011, p. 44) though the frequency is decreasing through improvements to the logic (e.g., a horizontal miss distance filter, p. 29) and training.

In particular, pilots need to learn how to interpret and interact with TCAS in the context of prior ATCO clearances. As illustrated by how one member of the BTC crew reacted, the mere presence of the TCAS display adds information that is difficult to ignore: "Aircraft have also been observed making vertical or horizontal maneuvers based solely on the information shown on the traffic display, without visual acquisition by the flight crew and sometimes contrary to their existing ATCO clearance." In response to this problem, more information has been added to the display over time to guide the magnitude of altitude change (e.g., pitch guidance, p. 38), making the use of TCAS more complicated than just following a "climb" or "descend" instruction.

In summary, most analyses for certifying systems like TCAS are modeled from the perspective of the aircraft and automated subsystems, as if people are not present: "Different aircraft can have different views of a situation because of sensor limitations" (p. 26). The technological dimension of course must be incorporated, and the use of probabilistic evaluations is useful (Kochenderfer et al. 2012a). But the fact remains that air traffic controllers are deliberately and completely dropped from technical characterizations of TCAS's design—by omission this was equivalent to predicting that the events over Überlingen could never occur. Notice in particular that the BFU Report states as an immediate cause:

> The TU154M crew followed the ATC instruction to descend and continued to do so even after TCAS advised them to climb. This manoeuvre was performed contrary to the generated TCAS RA. (p. 5)

But the report offers no parallel cause:

> The Zurich ATCO contradicted the TCAS instruction to descend and continued to do so even after TCAS advised the pilots to climb. This intervention was performed contrary to the generated TCAS RA.

The report nowhere mentions that the lack of information provided to ATCO about the TCAS instruction was pivotal in causing the collision (though there is analysis about the delay in DHL informing ATCO that they were in TCAS descent, which is part of TCAS operations protocol). Suppose for example that ATCO had seen the RA alert on his radar display while he was speaking—he could have easily stopped what he was saying and immediately say, "No, follow TCAS, climb immediately! Climb!" Instead of a system involving ATCO, aircraft, TCAS, and pilots, the TCAS certification models only aircraft and TCAS.

The Brahms-GÜM project picks up where a purely technological analysis and redesign leaves off by creating a socio-technical simulation that combines models of systems (e.g., aircraft, TCAS, radar) with behavioral models of people who are perceiving, interpreting, manipulating, and ultimately guided by or ignoring such systems.

The SSAT R&T Portfolio Project Plan 1 October 2011 emphasizes the importance of reflective agents that, for example, can detect failed goals that they might track to a missing assignment or a conflict in assignments. As a first effort grounded in a collision scenario, the Brahms-GÜM model focuses more on what can go wrong than on the favored, dynamic capabilities of a reflective work system with checks and balances that adjusts to shortcomings and optimizes its performance. In many respects, the value of the Überlingen scenario is to highlight for analysts how much can go wrong at the same time, and thus as explained in the discussion of Normal Accident Theory (section 5.3), how a merely complicated system becomes complex and out of control. With this baseline, further research could then focus on adding back the positive "what if's" by which the work system might have compensated for deficiencies (Section 6.7.3).

7.3 Broader Project Objectives and Approach

To review, this project addresses the general objective of developing a tool and methodology that will be useful early in the design of automation and air traffic management work systems to facilitate adaptation of NextGen automation, without the cost and effort to develop high-fidelity operations lab simulation experiments. The approach of the Brahms-GÜM project is to use a work practice simulation of how people interact with each other and automated systems to provide a practical, relatively quick way to analyze the safety of proposed automation and changes to work practices.

Because a work practice simulation may be configured for a very large space of circumstantial factors (such as equipment failure, weather, air traffic, staffing, human interactions), finding limitations in the work system design as embodied in the simulation model requires more than running a large number of scenarios (in which each scenario defines the initial conditions for a simulation run). To guide and go beyond the scenario approach, the broader project in which we are engaged will use model checking methods that have been successful in software engineering.

Thus this project has three research problems:

1. Formulate Brahms ATS simulation whose human-systems interactions relate to A&A.

2. Define formal semantics that enable reformulating the Brahms model/trace (e.g., using FSM & temporal logic), so specifications can be proven/disproven formally (Rungta et al. 2013).

3. Apply/adapt existing model-checking tool to verify safety properties of the model with respect specifications.

A key objective of this report is to document how the Brahms-GÜM has been conceived, developed, and refined so the model-checking method and processes relate to the practice of building a work practice simulation. In some respects, such a tool would be related to explanation systems that help model builders understand a simulation by enumerating why certain events didn't occur (Clancey et al. 1986).

8 Method: Development and Structure of the Brahms "Generalized Überlingen Model"

This chapter describes the nature and advantages of a model that generalizes the Überlingen collision scenario, and the overall process by which the generalized model has been created by converting and refining an existing human-systems flight simulation.

8.1 GÜM Concept and Motivation

The overall strategy for developing the Generalized Überlingen Model was to create a series of complete Brahms models relating to the Überlingen scenario that incrementally add off-nominal events and behaviors:

- *Complete* means that each model version provides a simulation that runs through complete sequences of events in which planes depart, fly, and land at destinations.
- *Incremental* means that each model in the series introduces more of the people, systems, and interactions that occur in the air traffic control work system.
- *Relating to Überlingen* means that the final model we produce in this series is not a specific replication of the Überlingen accident (e.g., like a re-enacted play), but more general, such that one of the scenarios models the work system configuration at that time.
- *General* means that initial conditions in the Brahms model (facts and beliefs), which we call "scenarios," can be varied to produce different outcomes. The model is designed so any combination of initial conditions (as provided by the model's design) can be specified.
- *Off-nominal* refers to violations of air traffic regulations, as well as variations from standard practice in aspects of work that are not subject to regulations.

Producing a series of models has many advantages:

- To always have runnable Brahms simulations available for the formal analysis aspect of this project.
- Modeling off-nominal behavior on a foundation of idealized "correct" behavior, thus:
  - ***Disciplining the modeling process,*** so that we represent the causes of variations in terms of agent beliefs and characteristics and world facts, rather than formalizing variations as fixed, given ("hardwired") behaviors;
  - ***Providing an experimental workbench in which a space of scenarios can be generated by varying initial conditions*** (that are known by definition to be factors relevant to producing off-nominal behavior and/or safety violations);
  - ***Providing a Brahms library of reusable components, consisting of "most general" groups/agents/objects*** (e.g., the idealized air-traffic controller) that are parameterized by initial conditions, enabling

future models to be created more efficiently (through reuse) and effectively (by incorporating the same principled framework).

- ***To enable creating an increasingly more complex simulation*** that refines the combination of events (e.g., number of flights), forms and behaviors of automation systems, and representation of how agent beliefs affect behaviors.
- To provide the **initial formulation of a general design tool for NextGen**, not just an analytic tool or replication of Überlingen.

In short, instead of specifically modeling Überlingen's work system configuration and events, we start by creating a more general model. This generalized model includes the activities and equipment that are included in the Überlingen scenario, but we start by simulating the proper practices of ATCOs and pilots and properly functioning equipment. Alternative behavior preferences and system dysfunctions are modeled as initial conditions, variations thus constituting different configurations of the generalized model (i.e., different scenarios). So for example Brahms–GÜM enables running scenarios with telephones working, the BTC crew reversing course after TCAS RA, STCA optical working, two ATCOs instead of one, etc.

A Brahms model could potentially model "the ideal pilot" and "the ideal ATCO" etc. in great detail (e.g., Casner 2007, Chapter 7 "Human factors of commercial pilot automation" lists specific best practices; also p. 48, "how to be a responsible pilot"). For example one would include many circumstances in which it is necessary to " solicit information" from the flight management system (p. 147, 90ff). Then one could introduce particular pilot, ATCO, etc. agent models representing variations related to training, personal experience, and so on. One could also introduce complications of many sorts: false alarms, masked problems, manual operation (in)capabilities (p. 104). To do this, one would have to model in detail how systems such as FMC, VNAV, LNAV, FMA, etc. work to allow variations of flight situations. But because of limited time and resources and a practical focus on formulating and demonstrating the value of a framework related to formal methods, the Brahms–GÜM has been scoped to include only the components required to simulate aircraft flight, including interactions with ATCOs, and that played a role in Überlingen (e.g., telephones).

In other respects, the nature of a work practice model dictates that certain aspects of the work system must be included in some detail. One may think of creating a Brahms model as like writing a script for a play: you must specify the geographic setting and the places (e.g., airspaces, buildings, rooms) where the action occurs; you must include the key people and their "props," such as chairs, documents, and devices (e.g., telephones, computers); and you must model the behaviors of the players and devices—what they say and do, how the react to events around them (in a rule-like way). When all of the people and objects are set in motion, their interaction produces a sequence of events—planes fly, pilots report to ATCOs,

ATCOs give directions, the FMS reports the aircraft status, and of course TCAS monitors the planes to produce appropriate alerts.

Here is a summary of what is included in the Brahms–GÜM, constituting the main elements of what we call a "work system":

- People: Pilots, ATCOs, ATCC assistants
- Air Traffic Control Centers: people, geography, devices, activities
- Specific flights with relations among crew, flight number and route, aircraft
- Communication protocols: for using radios, for ATCC sector handoff, for informing about TCAS RA, etc.

One difference between a play and a Brahms model is that a Brahms model is always more general than a particular sequence of events. In effect, each simulation run produces something like a script of a play. The model is a kind of generator for different sequences of events.

A Brahms model is always more complete than a script of a play. A play's script indicates only that certain actors appear, say certain lines, and go off stage. In a work practice simulation each of the actors (agents) is always doing something (called activities). Details may be omitted of what is happening during an activity; for example, Brahms-GÜM may be configured so one ATCO is "in the 'break' activity" at a certain time, but the model does not represent what a break entails, only that it occurs in another location (i.e., it is a primitive activity in Brahms). This approach allows in principle multiple shifts and days to be simulated, where only the activities of interest are modeled in detail. Furthermore, like a play, there may be multiple "settings." But unlike a play, in Brahms models actors may be behaving on multiple "sets" at the same time. For example, in Brahms–GÜM agents are behaving in multiple aircraft and multiple ATCCs simultaneously.

The "objects" in a Brahms model have their own ongoing activities; for example every telephone, radar display, radio, etc. is always in some state, regardless of whether a person is interacting with it. Again, overall the events that transpire in the simulation "run" are produced by the interactions of all agents and objects behaving "independently," though of course they may be affecting each other.

Finally, in Brahms further generality is attained for creating alternative work systems and scenarios by modeling pilots, ATCOs, etc. and aircraft, flights, radios, sectors, etc. "generically" using the agent group and object class constructs (Section 7.1.1).

8.2 Notion of a Base Model: Origin of Scenario Definitions

The concept of the Brahms–GÜM developed over time from the initial notion of incrementally developing a Brahms simulation of the Überlingen work system and events. Here we elaborate more carefully the evolution of our reasoning in

developing the base model by formulating a *sequence of models*, and thus came to conceive of the Brahms simulation as being a generalization of the Zurich ATCC and aircraft/TCAS work system circa 2002, rather than only a model of particular events.

In effect, we interpreted the base model and incremental additions in a new way as we proceeded, such that building a series of models became not just a means for handling the complexity of the Überlingen collision, but more generally a method for experimenting with how the various known causative factors we were modeling affected each other. By ensuring that each added factor was configurable as part of initializing a Brahms simulation run, it was possible not only to accumulate factors and settings that replicated Überlingen's work system, but also to experiment with any combination of factors (e.g., the AEF1135 flight is late but two ATCOs are present).

The last model created, which includes all of the factors, is the Brahms-GÜM; each combination of how its various entities can be configured (e.g., telephones operate or not; BTC is one minute earlier in route) constitutes a scenario. Thus the original focus on creating a *sequence of models* shifted at the end of the development process to exploring experimentally a *space of scenarios*.

The design concept of a series of models begins with a *base model* that includes all of the entities (people, flights, equipment) operating nominally. For each subsequent model we change one major causative factor at a time, starting with the most frequent/likely off-nominal conditions. We call each such factor an "exception." Applying this principle to the actual circumstances of the Überlingen accident suggested the following loosely ordered sequence of models (for brevity model versions #3 and following are described by indicating only the exception added to the base model):

1. Base model—all behaviors and events are correct or normative (includes DHL611, BTC2937, AEF1135 flights and STCA & TCAS systems).
2. Base model with exception that second Zurich ATCO is napping in the lounge (this had been generally accepted practice, though it violated regulations).
3. AEF1135 is late (call in to Zurich ATCO distracts from monitoring the other flights)
4. DHL611 and BTC2397 depart late (specific timing places them on a collision course)
5. Russian crew trusts air traffic controller, thus ignoring TCAS directive (given training variances, this is probably more likely than maintenance that disables the standard radar and phone system).
6. An equipment upgrade in process in Zurich degrades the radar available to the Zurich ATCO (lack of optical STCA causes flight data display to be delayed)
7. An equipment upgrade in process disables the primary and backup phone systems (this prevents ATCO from calling Friedrichshafen)

Originally, the Brahms model resulting from this process (model #7, which includes all of the factors of #2-#6) was conceived as "the Überlingen simulation." The base model concept was only intended to provide a systematic, incremental process that would break up the complex construction effort into small pieces that could be independently defined, modeled, and tested. (The modeling sequence/plan was adjusted over time to allow for modeling some aspects in detail and omitting others.)

Inherent in the Brahms modeling framework is the notion that people and systems are simulated to a degree independently, in terms of their essential physical and behavioral properties, and thus that the resulting simulated events arise through the interactions of the parts, rather than being "hardwired" (fixed). That is, the models of the agents and objects are defined from the perspective of each agent/group and object/class, allowing the many-to-many combinations of states to emerge rather than being specified by the modeler. Thus for example, a radar display would simply behave according to its present settings and the aircraft in the vicinity, independently of what the air traffic controller was observing or doing, and the aircraft would fly independently of whether they were being tracked on radar.

From this perspective, our analysis and simulation begins by characterizing proper operation (e.g., how the phones are used in a handover process by the ATCO), then includes the particular configurations of the situation (e.g., how many ATCOs were present and their assignments; the flight paths of aircraft entering the sector). Finally, off-nominal conditions are simulated (e.g., the phones are not working; only one ATCO is present) and behaviors of the agents elaborated to simulate how they would normally cope with off-nominal conditions.

Insofar as creating a Brahms simulation is analogous to scripting and performing a play, the first step in creating the base model is actually to establish the geography model. The boundaries of the initial geography are established by the collection of events that model is designed to simulate. Thus we included all of the cities where the players (the aircraft and air traffic controller) would be located (e.g., Moscow, Bergamo, Zurich) and the "rooms" where people would be performing (e.g., aircraft cockpit, ATCC). Detail is determined by the purpose and focus of the model—so we modeled that the cockpit is part of the aircraft because we needed to simulate aircraft in flight, and we modeled instruments in the cockpit because we were going to simulate pilots interacting with instruments. Similarly, we modeled workstations and their location in an ATCC, but we didn't model specifically where the ATCC was located in a region because that was irrelevant to the interactions and events of interest.

In summary, the geography model, as well as all the modeled players and objects and their behaviors, are defined according to the processes and events that the simulation is designed to include. This is no different from any other scientific model, which defines certain aspects of the world and ignores everything else.

Just as a scientific model can be augmented by adding new parameters and relations, or a play might be elaborated by adding another player, scene, prop, or set, a Brahms model can be directly modified by changing the agents, their activities, objects, or the geography. For example, to add another flight one might add other cities to the geography (indicating relative distances to those already in the model), and literally copy and edit existing objects in the model to create another aircraft, crew, and flight definition, etc. Through the group inheritance process, all flights for example share the same definition structure, so adding a flight entails adding another "instance" of the class of flights and specifying its properties (e.g., origin and destination, waypoints).

8.3 Overview of Modeling Process

The following are the general steps we followed in creating the Brahms–GÜM by adapting an existing simulation, creating a sequence of models, and refining the simulation to fit known ATCO and pilot practices and the Überlingen variations.

1. **Adapt "Work Model that Computes" constructs to develop basic Brahms ATS simulation**

Pritchett (2011) developed a framework that extends "qualitative Cognitive Work Analysis (CWA) to a form suitable for computational simulation of multi-agent socio-technical systems." They chose a simple, problem-free flight from San Francisco to Los Angeles, modeling the landing at LAX. The extended CWA framework models the work domain as an abstraction graph of mean-ends relationships:

> *Functional Purposes* (representing mission goals of the system), *Priorities and Values* (representing principles or values that the system must follow or preserve), *Generalized Functions* (representing process descriptions entailed to achieve mission goals), *Physical Functions* (representing capabilities of agents and equipment), and *Physical Form* (representing physical characteristics of equipment).[23]

In related work, Ho and Burns (2003) applied the similar "Work Domain Analysis" abstraction hierarchy to modeling the function and operation of TCAS to "establish information requirements" and develop enhanced displays. The objective of this analysis was to relate TCAS's function to information it provides to the pilot, identifying additional information that might be useful (e.g., traffic path prediction). As applied here WDA focuses on the TCAS interface and does not actually model the pilot's work or reasoning.

Pritchett's project sought to convert the related CWA framework from a strictly functional hierarchy into a "work model that computes" by incorporating a *cognitive task analysis model* that identifies "the worker's states of knowledge and how they

[23] Pritchett, "Work Modeling Paper 8.1.2011" p. 2.

are processed" (p. 3). The overall framework incorporates as well an *operating strategy model* by representing different "control modes"; it also incorporates a *social organization model* by representing distribution of tasks.

Specifically, the project converted a static CWA model into a simulation that models *pilot-flight deck function allocation*, involving three flying modes (autopilot, manual, and hybrid). The resulting simulation is called "the work model that computes" (WMC).

Representing the WMC constructs in Brahms provided a way to bootstrap a simulation of Überlingen events by virtue of incorporating airspace systems and processes essential to any model of flight; the Brahms WMC model and modeling process are described in Appendix 17.

The conversion of WMC to Brahms also provides a direct contrast of functional and behavioral simulation approaches (Section 8.4). The use of the Brahms framework also demonstrates the advantages of using a structured agent-world modeling framework (in contrast with C++) for experimentation with alternative scenarios and reuse of the ATS model components. Furthermore, the design of the Brahms engine was hypothesized to be directly amenable to creating a model checker.

2. Elaborate Brahms WMC to create work system model with distributed spatial-temporal interactions

The WMC model does not represent a *complex* multi-agent team; it is instead a baseline simplification of actual ATS flight systems, probably the simplest case—one person on the ground (air traffic controller, ATCO), one in the air (pilot), and one automation system onboard (FMS), with no conflicts. The focus in WMC is on modeling function allocations to simulate emergent interactions such as workload; our concern in Brahms–GÜM is on simulating the *interactions* among people and automated systems work system, to show how interacting processes over time and space produce emergent overall system properties (e.g., a safety violation). Hence much more detail is required in simulating the people and systems (including for example aircraft radios and the communication protocols).

3. Develop Brahms-GÜM incrementally as series of models

With the framework and plan in place, the Überlingen scenario was gradually modeled in Brahms; for details see Section 8.6.

4. Sensitivity Analysis and Experimentation

After the initial modeling effort to incorporate all of the relevant entities, normal behaviors, and exceptions, we shifted to analyzing traces (simulation logs) to understand the timing sensitivity of events—what minor adjustments to presumed durations of "primitive activities" would change the outcome? Are the duration ranges plausible? Experiments were undertaken to force certain event orderings and thus verify that interactions occurred as anticipated and further understand sensitivity of outcome to minor variations (e.g., how soon must the ATCO intervene

after the TCAS TA and before the RA for the descent instruction to the Russian BTC to be effective?). The transcripts and available documentation were repeatedly reanalyzed to extract data about the operation of TCAS (e.g., why doesn't the BTC receive an expedite instruction at the same time as the DHL?), how the ATCO and the pilots behaved (e.g., delay time from instruction to aircraft response) and why people behaved in a certain way (e.g., why did ATCO repeat call to BTC to descend?). See Chapters 9 and 10 for detailed discussion.

5. Refinements to Model and/or Brahms Engine for Formal Semantics

In parallel with the Brahms–GÜM effort, other researchers were investigating the use of formal semantics of the Brahms language in applying formal methods for analyzing Brahms models and simulation outcomes (Rungta et al. 2013). For example, it was discovered that workframes were ordered based on when the workframes became "available" (conditions satisfied), rather than being random. Given that the intention of the Brahms framework was for ordering available WFs to be arbitrary (which is in accord with the formal language definition), the Brahms engine was modified. Application of model checking to Brahms–GÜM is mostly future work (Chapter 12.8).

The following sections explain the challenges and processes for creating a work practice simulation from the Brahms model converted from WMC, how Brahms WMC was modified incrementally to create Brahms-GÜM, and then the initial experimentation with ten Brahms-GÜM scenarios varying key factors causing the accident.

8.4 Elaborating Brahms WMC Model to Work Practice Simulation

Developing Brahms–GÜM from the Brahms WMC model (detailed in Appendix 17) required adding geographic locations and facilities, agents and their activities, many subsystems (e.g., radio, radar), and of course multiple aircraft and crew. The transcript of the Überlingen accident and BFU Report provided the basic information required, supplemented by independent analyses (Chapter 6). The process of creating a work system simulation based on the Überlingen accident from the WMC functional-allocation model accomplishes the following:

- Demonstrates flexibility and adaptability of an object-oriented framework like Brahms
- Elucidates WMC strengths and limitations for simulating human behavior compared to a work practice model
- Provides a structured analysis and model of the Überlingen accident's anomalous conditions to understand systemic interactions of nominal behaviors, equipment dysfunctions, and human errors.

In the first step of creating the Brahms WMC model, the WMC formulation (Kim 2011; Pritchett et al. 2011) is effectively converted to a multi-agent simulation, such that the agents are independently behaving processes. "Functional allocation" (FA) is an approach to work system design, involving functional decomposition and

mapping of functions to people and systems (for example, see Kim 2011). FA allows for different modes (i.e., mappings can change during operations), different strategies for accomplishing a function, and abstract levels of functions in which responsibility may be joint or distributed. FA emphasizes logical requirements and capabilities of people and systems, that is, it is a rational analysis that explains all behavior as procedure-following or logically derived from inferences.

Creating Brahms WMC involved converting the WMC functional descriptions that characterize essentially the *procedure* in flying the plane (in terms of steps and what each accomplished) to an *activity-behavior model* that characterizes how each step is accomplished by the pilot through interactions with flight management computer, the radio, and the air traffic controllers. (While as well, the air traffic controller's work context and behaviors are affecting the pilots and flight of the aircraft.)

In contrast, the WMC model focuses on general agent decision making and capacities:

> Rather than attempting to model the work of each agent as descriptions of their activities embedded within the agent models, the framework uses agents as a means to further allocate and regulate the decision and temporal actions described outside the agents in the work model.

In effect the simulation of human behavior in WMC involves having a generic agent process interpret a separately described functional model of the work. This approach enables a theoretical analysis, as WMC demonstrates in characterizing workload under different modes of operation. Such a functional analysis is particularly useful as a method for evaluating a work system design with respect to resources, timing, safety, and other constraints.

In Brahms, rather than being identical, agents with the same roles may have different beliefs, preferences, and behaviors, all of which constitute individual practices. Agents inherit beliefs and behaviors (activities) from (possibly multiple) groups, so variations need not be reprogrammed for all agents individually. If identical "work and mission goals" are desirable in the model, all of the agents can inherit beliefs that affect their behavior strategically. Using a hierarchical, object-oriented structure as in Brahms, one has the advantage of uniformity across the system and variability when it is desired.

The WMC structure makes an important simplification by making decision making ("deliberation" or inferential reasoning) the driver of every action. Brahms is based on the perspective of situated cognition (e.g., Clancey 1997, 2002), making the activity structure of the groups to which the agent belongs the organizing construct of the simulation. We are thus able to model work practices (how people behave) in dynamic contexts (not pre-defined or necessarily anticipated by the modeler).

Furthermore, this is accomplished in a manner that provides an easily modifiable and extendable model developed from a library of components.

Details about the advantage of the Brahms framework for modeling multitasking and shifts of attention are explained in subsequent subsections.

8.5 Additions to Brahms WMC Model to Create Brahms-GÜM

Our first step in converting the Brahms WMC model to simulate the Überlingen accident was to add and configure components to model an idealized scenario based on the Überlingen flights without any complications: two flights (FL2397 from Moscow to Barcelona and FL611 from Bergamo, Italy to Brussels), two ATCOs on duty in Zurich, and all ATCC equipment functional. The planes depart on schedule, and this alone averts the collision. Creating just this "simple" model required the following entities to be added to Brahms WMC:

a. Agents:
    i. Pilots in each aircraft
    ii. ATCOs (Zurich, Karlsruhe)
b. Groups:
    i. Air traffic controllers and "communicators" (agents who engage in protocol-based verbal exchanges)
    ii. Flight crew
    iii. Pilots and DHL Pilots
c. Geography:
    i. Airports (Moscow, Bergamo, Barcelona, Brussels) and runways
    ii. Air space, air routes, waypoints
    iii. Air traffic control centers (Zurich, Karlsruhe) and work station areas
    iv. Aircraft areas: cockpit, cabin
d. Objects:
    i. Aircraft: DHL, BTC (flights are conceptual objects associated with these)
    ii. Flight Management Computer (FMC) with Cruise & Standard Terminal Arrival Route (STAR) modes for DHL & BTC
e. Activities:
    i. Flight
        1. Take-off Phase:
            a. Clock in ATCC announcing time for departure
            b. ATCO communicates departure approval
            c. FMC guides with Standard Instrument Departure
            d. Pilot activities & communications
        2. Cruise Phase:
            a. FMC flying in auto-pilot mode using flight plan
            b. Pilot activities & communications
        3. Landing Phase:
            a. Pilot activities & communications

ii. ATCOs handoff & accept flights

Creating this model soon revealed that getting the planes to collide at all and for this to occur near Überlingen required interpolating data from the BFU Report to determine both the exact route and flight times between waypoints (such as ATCO clearing DHL for a "direct ABESI" path). The waypoints (intermediate locations on route) and timing of the BTC flight from Moscow were not available in the BFU Report, and could only be estimated from maps and by analogy with current commercial flights.

With the skeletal model of aircraft in flight in place, we turned to modeling the proper ATCO-pilot handoff communication protocol, which required modeling radios and how they were operated. Modeling the radar, STCA, and TCAS were all substantial projects themselves, followed by modeling ATCO monitoring a single and then multiple workstations, and so on.

The following general plan was followed for developing the Brahms-GÜM:

1. Complete interactions among Pilot, Flight Systems, and Aircraft for climb and cruise with European geography for single-plane DHL flight plan (i.e., adapt Brahms WMC model).
2. Add BTC flight, flight plan (two versions: on-time and delayed with collision) and geography — this is independent of ATCO actions, to confirm that simulation reproduces collision with flight paths actually flown.
3. Add Radar Systems and Displays with ATCOs, located in ATCCs, monitoring when flights are entering and exiting each European flight sector in flight plans.
4. Complete handover interactions between Pilot and ATCOs for each flight phases.
5. Two ATCOs in Zurich (Radar Planner and ARFA Radar Executive) assigned to two workstations (RE has nothing to do under these conditions).
6. Add TCAS with capability to detect separation violations, generate Traffic Advisory (TA) and Resolution Advisory (RA)
    a. DHL and BTC are delayed (i.e., on collision course, which tests TCAS)
    b. Pilots follow TCAS instructions
    c. ATCO might intervene prior to alert, depending on information from radar displays.
7. Third plane, the AEF flight, arrives late, requiring ATCO communications and handoff to Friedrichshafen control tower:
    a. Handled by ATCO in Zurich at right workstation (ARFA RE sector) and not left RP sector workstation.
    b. Phone communications for flight handovers
    c. Methods used by ATCO when phone contact doesn't work:
        i. Ask CA to get another number: requires about 3 minutes for CA to return

ii. After that number fails, discuss with CA other options about 30 seconds
iii. When not busy handling other flights, try second number again
iv. When plane is within a certain (configurable) distance of airport, method of last resort is to call pilots on radio and ask them to contact the tower directly

8. STCA optical subsystem added to ATCO workstations; ATCO responds to alert by advising Pilot to change flight level based on next flight segment of flight plan.
9. Reduce ATCC staff to one Zurich ATCO (this model begins the sequence of variations from the nominal situation)
    a. Requires ATCO managing workload and communications (e.g., associate workstation with flight frequency) from both RE and RP workstations
    b. Priority is given to the late-arriving AEF flight over monitoring the airspace
10. Handover interactions when one Zurich ATCO goes on break, requiring RE workstation to be configured for ARFA sector.

The sequence is based on simulating normal operations before introducing exceptions. TCAS was added relatively early to be sure that independent of ATCO advice, it would operate correctly. Two ATCOs were modeled before reducing to one, so the two positions were well formulated and reduction could be modeled by simply having one ATCO adopt behaviors of both roles (Brahms groups).

Over time, as described in Section 8.2, it was recognized that the model needed to be systematically structured to enable independently reconfiguring the modeled equipment and automation to simulate the entire space of scenarios the particular factorization of the work system in terms of people and systems allows. That is, all of the combinations of initial facts and beliefs (scenarios) must produce logically correct behaviors and interactions in the simulation. Some of the details required and complications are described in Chapter 9.

8.6 Defining the Sequence of Test Scenarios

The concept of a sequence of scenarios created by configuring Brahms-GÜM (the final model in the development sequence described in the preceding section) began with the idea that we would simulate the Überlingen situation first, then vary factors experimentally to identify key factors and/or interactions. These variations were based on the primary causal factors ("exceptions"):

- Überlingen as it occurred (1 ATCO, no phones, no STCA Optical, AEF arriving, DHL and BTC on collision path)
- Two Zurich ATCOs were present (second controller does not go on break)
- One Zurich ATCO but phones operational
- One Zurich ATCO but without delayed AEF flight

- One Zurich ATCO but AEF flight is treated differently (flight is put on hold, tower is contacted sooner, or landing AEF flight is treated as lower priority) when phone problem is discovered so that monitoring other flights is not impaired or delayed
- One Zurich ATCO but STCA Optical operational

Also, our plan was that if a collision still occurred with two air traffic controllers, we would change ("add back") exceptions to determine whether they made a difference. Our initial hypothesis was that either two Zurich ATCOs would suffice to avoid the collision or one Zurich ATCO with any of the key exceptions resolved.

The list above was then formalized and systematized (about six months into model development) by reversing the sequence to test Überlingen last, starting with a fully nominal configuration and adding exceptions until the "fully degraded" Überlingen scenario was replicated (Table 8-1; Table 24-1). This enabled testing more directly the hypothesis that all or most of the anomalies were required for the collision to occur and would reveal without single or simple combinations of exceptions could alone cause the collision. This approach would also reveal whether the "Swiss cheese" perspective (Section 7.2.3), which postulated that all exceptions were required for the collision to occur, was valid for the Überlingen situation.

In the table, "Y" and "N" indicate whether a characteristic is present or not. Thus the *exceptions* are defined as number of Zurich ATCOs is one, AEF Flight present (Y), BTC Pilots follow TCAS (N), STCA Optical absent (N), and phones absent (N). To enable verifying that the planes would collide, TCAS can also be disabled (N).

The sequence (rows in this table) is designed to sequentially and cumulatively add anomalies (e.g., compare scenarios 2A , B, and C). The scenario labeling scheme refers to the number of Zurich ATCOs; thus all scenario labels starting with 2 have two ATCOs. This table was extended as simulation outcomes suggested additional scenario combinations to test (Chapter 10).

The test sequence had the important side effect of making explicit a modeling design constraint that had been implicit in the original notion of "base model versions," namely that each aspect (such as the behavior of the radar system) had to be independent of the others. In particular, it had to be possible to run a simulation having a different work system configuration without rewriting any aspect of the model. In Brahms terms, the model had to be properly parameterized by "initial beliefs" of agents and "initial facts" about agents and objects. Thus, a scenario can be defined by changing certain "declarations" in Brahms-GÜM rather than changing the model of how people behave or systems operate in general (i.e., activities, workframes, thoughtframes, etc.). Appendix 24 indicates the actual model configurations corresponding to the ten scenarios in the test sequence table.

Table 8-1: Ten Test Scenarios Definitions and Predicted Outcomes, Given BTC and DHL on Collision Course

Scenario Description	Prediction: Collision Occurs?[24]	# Zurich ATCO	TCAS	BTC Pilots Follow TCAS	Radar & STCA	AEF Flight	Phones
0) Null	YES	0	N	—	—	—	—
2A) Normal	NO	2	Y	Y	Y	Y	Y
2B) Normal w/o Phones	No, expect 1st Zurich ATCO to advise BTC early enough	2	Y	Y	Y	Y	N[25]
2C) Phones out & Radar degraded, but TCAS rules	Is BTC advised in time so TCAS not activated? If not, does BTC reversal occur in time?	2	Y	Y[26]	N	Y	N
2D) ... but Zurich ATCO rules	Yes? BTC advised early enough?	2	Y	N	N	Y	N
1A) Normal-SMOP	No? BTC advised early enough, so TCAS not activated?	1	Y	Y	Y	Y	Y
1B) SMOP w/o Phones	No? STCA alert compensates for distraction?	1	Y	Y	Y	Y	N
1C) SMOP w/o Radar	No? Attends to radar?	1	Y	Y	N	Y	Y
1D) Actual, but TCAS Followed	Yes? Zurich ATCO advises too late; BTC reversal ineffective?	1	Y	Y	N	Y	N
1E) Überlingen	YES	1	Y	N	N	Y	N

Given that the model is always scoped, such that behaviors are limited, some aspects of the model had to be reconstructed from the initial design to achieve the desired configurability. For example, the modeler needed to know whether the DHL pilot would always follow TCAS, given that this variation was not specified in the table. This question clarified the general principle that the pilots should be modeled as clones (agents who all inherit behaviors from the same group), except for behaviors that needed to be varied, such as response to TCAS. It must be possible in principle to easily configure and run scenario arbitrary variations (e.g., DHL ignores TCAS and BTC follows TCAS, reversing course after Zurich ATCO instruction). There should be nothing inherent in the model that builds in behaviors of DHL and BTC pilots in any way (especially because we were not modeling how the aircraft crews behaved among themselves). Consequently, the behaviors required for "following TCAS" including reversing after an ATCO instruction are modeled at the pilot level. ATCOs are similarly clones; but also have contingent behaviors (e.g., potentially advising BTC to climb or advising DHL instead of BTC depending on circumstances).

[24] Prediction refers our expectation of outcome in simulation run for given scenario. Questions indicate that outcome is not obvious given alternative sequences of events that might occur.

[25] When phones don't operate, ATCO asks for assistance.

[26] "BTC following TCAS" implies pilots will ignore ATCO if TCAS advises first or pilots will reverse course if ATCO advises first and TCAS gives contrary instruction.

The next chapter details the main challenges discovered when creating or testing the ten scenarios. The subsequent chapter picks up the story of simulation outcomes and necessary refinements to increase fidelity of Brahms-GÜM (generally details and timings) for accuracy and validity.

9 Method: Modeling Challenges and Abstractions

Given the modeling objectives and work practice perspective, the Überlingen accident transcript provided by ANSA and BFU Report timeline provides most of the essential behaviors of key players.[27] The BFU Report text provides many additional details and analysis required for constructing the model; it was apparently produced by researchers sensitive to organizational and psychological human factors.

However, constructing Brahms–GÜM was complicated and made difficult for several quite different reasons:

- *Missing information about the work system*
  - Basic information about Überlingen work systems is not mentioned in published accident reports and analyses (e.g., what region is visible on radar display configured for ARFA sector?).
  - Modeling flights and routes/waypoints requires a great deal of data that is not readily available (e.g., routes of the flights Zurich ATCO directs before and after BTC enters his sector, including AEF 1135).
  - Normally a Brahms model is constructed from observational data of an ongoing work practice. Without such ethnographic data about air traffic work practices at Überlingen in 2002, we do not know when Zurich ATCOs typically detected and acted on separation infringement; to what extent they relied on STCA or equivalent alerts and specifically what happens when alerts are not present; and so on. (See Section 12.6.4 for further discussion.)
- *Limited modeling resources*
  - Simulating each of the complicated subsystems would normally require a modeler (e.g., radar server, TCAS) to create the simulation more quickly; simulating even simplified versions reduced time available for the single modeler to test configurations (drawing out the modeling process puts more demands on the research team to remember and track details).
  - Simulating crew interactions with participants having different knowledge and training particularly about TCAS (as in the BTC cockpit and in contrast with the DHL crew) would itself be a year-long effort.
- *Complexity of interactive details*
  - Simulating ATCO visual perception in reading and interpreting radar display and similarly simulating pilot's line of sight and interpretation of "intruder" aircraft is not appropriate for Brahms work practice simulation (but could be readily integrated with the model).

[27] The significant events of interest are represented in the BFU Report chart in *Report_02_AX001-1-2_%C3%9Cberlingen_Appendix_1-3.pdf* on pages 2 and 3.

In some cases, an aspect of the work system that raised substantial research problems on its own could be ignored. For example, the BFU Investigation determined that the STCA audible alarm was apparently sounded at 21:35 directed to the RE (right workstation), but was unable to determine and provides no conjectures why no one in the Zurich ATCC heard it (BFU Report, p. 42). We do not even know for example whether a radio call from a pilot could mask the sound of the alarm or where it is located in the ATCC relative to the workstations. Given that the research issue for ATCC work systems design is why the alarm was not heard, the existence of the alarm and its state could be ignored for our purposes. Further, the alarm's sounding at 6.5 nm make it irrelevant to the analysis because ATCO had just completed instructing the BTC pilots at that point.

As the list above shows, scoping the Brahms-GÜM modeling effort to fit the time and resources available required both omitting and simplifying aspects of the work system. In particular the "base model" (refer to description in Appendix 23) is a greatly simplified version of the actual work system at Überlingen in 2002:

- The aircraft crews consist only of a pilot who does all the work, rather than pilot flying (PF) and not flying (PNF) roles; pilot's activity model is limited to manual-only mode (see Chapter 17) for takeoff, cruise, and landing phases, handoff and instruction-following protocol with ATCOs, and TCAS response.
- Except for ATCO–pilot handoff and ATCO–CA regarding phones, communications and other interactions between people are not modeled. These include:
    - BTC cockpit crew arguments about interpretation of TCAS alert and what action to take
    - ATCO and engineers' interaction at time of maintenance
    - ATCOs and supervisor interaction prior to his departure
- Additional flights controlled by Zurich during the simulated period are omitted: EXS6497, THA933, TAR4575, NMB286, BVR305, EZS935, CRX256, LTU7791, SRR6073, MON6521, PGT505. These are treated instead in the aggregate as a periodic ATCO activity of "handling other flights"; implications are discussed in Section 9.7.
- Several arguably reasonable ATCO behaviors that did not occur during the Überlingen accident and might have prevented the collision are not modeled, for example:
    - Putting BTC at a different flight level on initial arrival in Zurich sector (noticing it is at the same altitude as DHL)
    - Not approving DHL FL360 given that control strip indicates BTC is at FL360
    - Putting late-arriving AEF into holding pattern to be sure larger sector is being monitored properly[28]

[28] The BFU Report refers to this flight only as "a late-arriving Airbus 320" whose schedule was not known to the two ATCOs. The ANSA transcript identifies the flight as "AEF1135 Aero Lloyd (Aero

    - SYMA or ATCO testing phones when maintenance work is reported
    - Supervisor (DL) informing ATCOs not to go on break until maintenance completes
    - Karlsruhe ATCO contacting BTC/DHL on emergency frequency despite not having approval from Zurich ATCO via phone
- Simulations of all automated systems are greatly simplified: radar, onboard flight management system, TCAS, phones (operating or not throughout a simulation run). Simulated flights travel over VOR waypoints rather than nearby.[29]

In summary, the version of Brahms-GÜM reported here does not replicate all of the pilot and ATCO actions that are apparent in the accident transcript, nor does it include the "obvious" behaviors that might have occurred to handle distractions and/or detect and avoid the collision. Instead, the approach for this project has been to create a flexible comprehensive model with all of the key players and systems involved in Überlingen to be sufficiently complex to merit model-checking research and sufficiently general to allow future extensions without significantly re-formalizing the existing model of object and agent behaviors.

Certain processes crucial for simulating the accident could not ignored, but because of the missing information and/or lack of resources could not be modeled in detail without making broad assumptions or simplifications. These crucial aspects and how they are modeled are discussed in subsequent sections: the radar display and how it is read; ATCO intervention with BTC; TCAS; crew's interpretation of TCAS/ATCO interventions; and timing of ATCO routine activities not explicitly modeled (especially other flights).

Most of these issues came to our attention while analyzing the simulation logs after the "base model" development sequence was completed and the Überlingen scenario itself was run. We expected to learn about the important behaviors that needed to be included and/or modeled more accurately only after experimenting with Brahms-GÜM scenarios and analyzing the simulation logs. However, it was somewhat surprising that timing of events at the level of a few seconds made such a difference in the simulation outcomes. We had not encountered such sensitivity to timing and emergent interaction sequences in any of the prior Brahms models created over two decades. This result is consistent with the claim that the degraded Überlingen work system was complex (Chapter 5) and provides evidence that Brahms-GÜM appropriately represents and allows simulating a work system with complex human-automation interactions.

Lloyd Flugreisen)" (p. 57). Neither report indicates the flight path of the aircraft, but it certainly was at a much lower altitude than the DHL and BTC flights.

[29] It was assumed that extensions of Brahms-GÜM would integrate existing simulations of TCAS and flight systems or even actual operating hardware/software systems with appropriate APIs into the simulation; e.g., Clancey and Lowry 2012; Clancey et al. 2012).

For convenience, we collect here the most important modeling challenges, including how we abstracted complicated behaviors and subsystems. The exposition is loosely ordered compositionally, relating systems to individual behaviors to interactions about multiple systems. This list mixes aspects known to be complicated when initially constructing the model (e.g., the workstation radar display) with those understood better by analyzing interactions that occurred during simulation runs (e.g., priorities of ATCO's activities).

The following chapter describes in more detail the process of running the model and the analyses (including further research) that revealed what refinements were necessary or desirable. Together these chapters reveal how subtle issues of timing in human-automation interactions may arise when degraded or missing subsystems result in lack of information and inability to communicate, transforming a given configuration of flights that are routine in a normal work system to a situation too complex to handle.

9.1 ATCC Workstation Radar Display

Modeling the radar display involves calculating data about the planes and transmitting it to the display, representing a changeable sector (geographic region) on the display, and updating the display periodically.

As summarized in Figure 9-1, the model of the Primary Surveillance Radar (a Brahms object) examines the airspace within an Air Sector and communicates the basic flight data to the ATC Server (object), which in turn communicates it to all of the ATC Displays handled by this server.

The ATC Display has properties indicating what region of the airspace should be represented on the screen. The region is defined in terms of its origin x-y coordinate and range from the origin. When a simulation run begins, the Min x-y coordinate (farthest South-West point from origin) and Max x-y coordinate (farthest North-East point) are calculated, given the range. If an aircraft is within this square and the flight level is within the altitude range of the air sector, then the ATC Display has a property indicating that the aircraft is present (visible).[30]

We are uncertain what the Zurich workstation was displaying at any moment because we do not know how the sweep of the radar related to the positions of the planes on the screen—data for a plane might be current or up to 12 seconds in the past. As a simplification, the radar model defines the update interval to be 12 seconds divided by the number of planes in the monitored air space. For example, if only DHL is in the air space, Zurich Radar reports the DHL plane's data to the Zurich ATCO server every 12 seconds. After Zurich Radar detects the AEF flight in its monitored air space (so there are now two planes in the displayed airspace), it

[30] Determining distance of the aircraft from the ATC Display origin would have required a Java calculation, which was avoided whenever possible so as to keep as much of the simulation within the Brahms syntax for future model-checking research.

reports the DHL plane's data to the Zurich ATCO server in the first 6 seconds, then 6 seconds later reports the AEF plane's data. When Zurich Radar also detects the BTC flight in its monitored air space, it updates the data for the three planes sequentially 4, 8, and 12 seconds during each 12-second sweep.

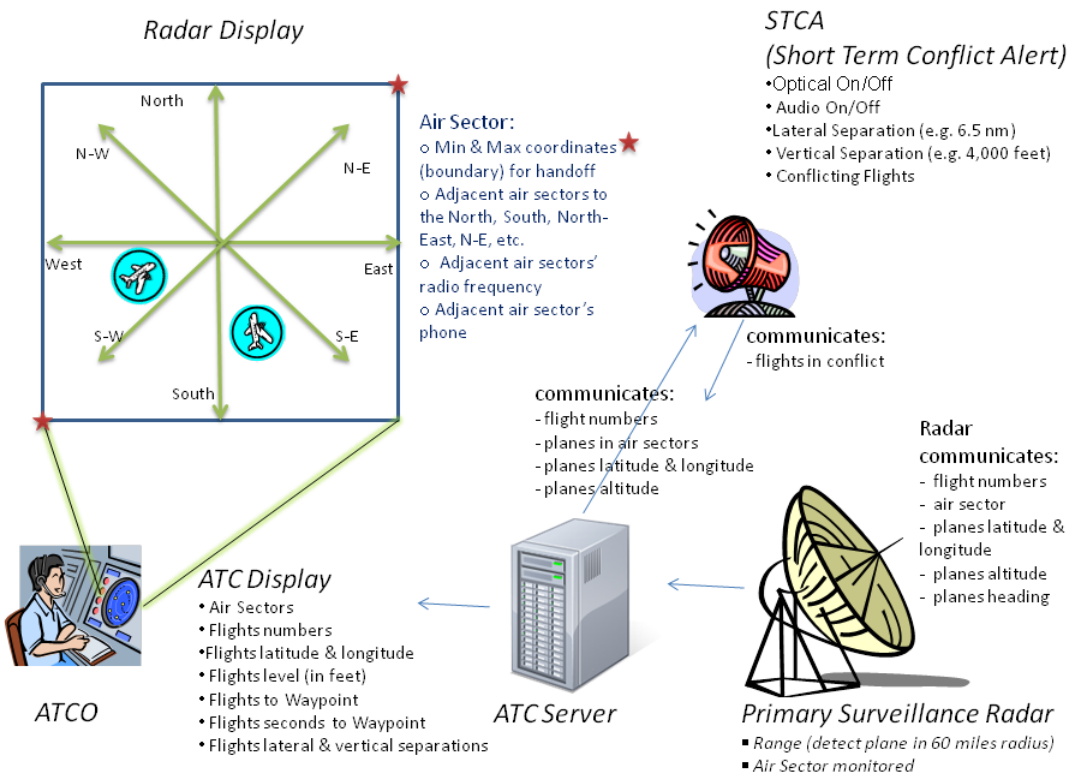


Figure 9-1: Communication of Air Sector data among PSR, Server, STCA, Display, and ATCO

Each of the two workstations in Zurich ATCC were modeled to represent the East/South sectors. When the second ATCO goes on a break, the right workstation is reconfigured to display only the ARFA Sector. Because the ARFA Sector's geographical boundaries are an irregular polygon, the coordinates were determined combining information from the BFU Report Appendix 3 diagram, BFU Investigation Report 1872 ("Areas of Responsibility"), and Google Earth. Figure 9-2 shows a simplified approximation of the ARFA sector outlined in black, superimposed on the BFU Report Appendix 1 map. The green internal section is the ARFA FL95-, which is not relevant to these operations.

The approaching BTC (red) and DHL (blue) aircraft are shown at approximately 30 nm separation, when the STCA optical alert would have been visible on the radar display if it had been operational.

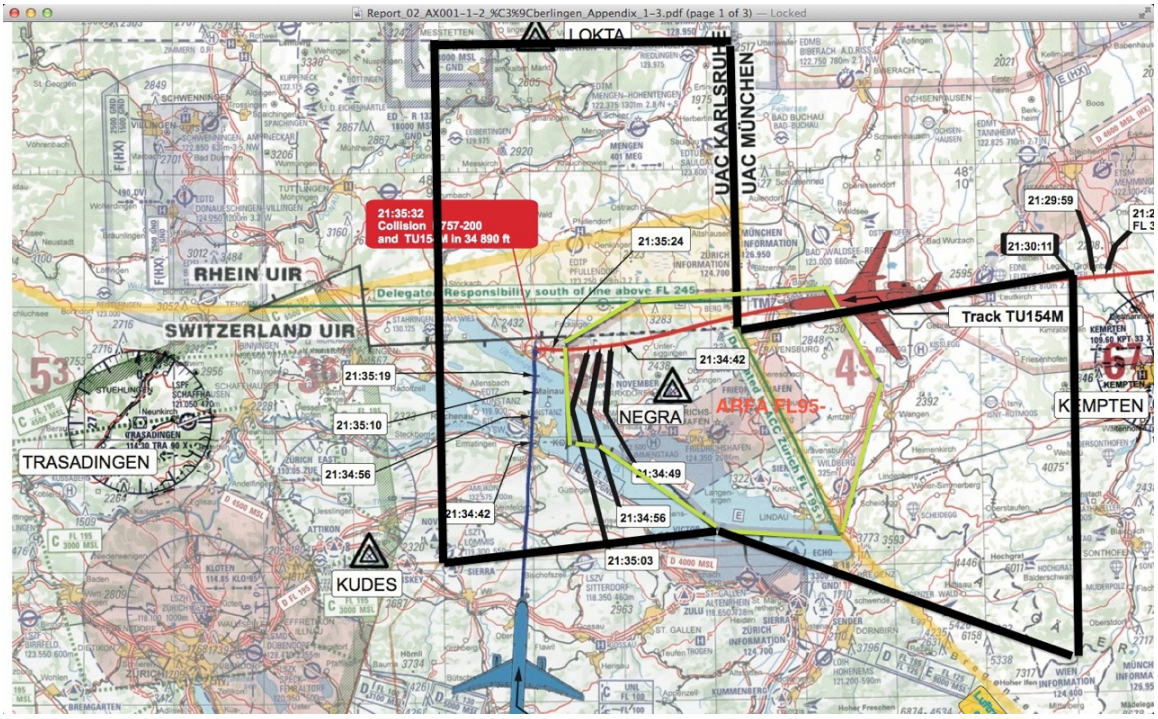


Figure 9-2: Approximate boundary (in black) of ARFA Sector monitored by Zurich ATCC on A RE (right) workstation

In Brahms-GÜM, the ARFA Sector is modeled simply as a square whose east boundary corresponds to where the BFU Report indicates that the BTC became visible on the display; the southern boundary is estimated as just south of Lake Constance, which fits approximately where and when the DHL was reported as becoming visible on the display.[31]

9.2 ATCO Reading Radar Display and Detecting Separation Infringement

Visual processing is an important topic in simulating human performance, particularly for air traffic control (Freed 1998). Brahms-GÜM does not represent how a person might visually interpret the radar display (e.g., perceiving geometries of flight paths), but rather models looking and reading the display as if it were text, consisting of data about the aircraft. In Brahms terms, the Radar Display object (shown in Figure 9-1 above) has "beliefs" about this data that are communicated to the ATCO when he monitors the display. The ATCO engages in a "communicate act" in which he "asks" the display for certain parameters.

The definition of the communicate activity **monitorPlane** shows the values ATCO receives from the radar display:

[31] Initial testing of Brahms-GÜM with the Überlingen work system configuration revealed that the initial ARFA Sector Radar model was a bit too far west and north, given times of BTC and DHL appearance on A RE radar display noted in BFU Report in Appendix 3. This reduced the time the simulated ATCO might observe both planes together while working at right workstation to assist AEF late-arriving flight. After the simulated ATCO was made less conservative in intervening, moving the boundary had no effect.

```
communicate monitorPlane(string dispStr, int minDur, int maxDur, Aircraft plane, Flight flight,
AirTrafficControlDisplay atcDisplay) {
                display: dispStr;
                random: true;
                min_duration: minDur;
                max_duration: maxDur;
                with: atcDisplay;
                about: receive(plane.location = unknown),
                      receive(plane.waypoint = unknown),
                      receive(plane.timeToWaypoint = unknown),
                      receive(plane.heading = unknown),
                      receive(plane.flight = unknown),
                      receive(plane.latitude = unknown),
                      receive(plane.longitude = unknown),
                      receive(plane.altitude = unknown),
                      receive(flight.flightNumber = unknown),
                      receive(flight.airSector = unknown),
                      receive(flight.flightInBoundary = unknown),
                      receive(flight.flightLateralSeparation = unknown),
                      receive(flight.flightClosest = unknown),
                      receive(flight.isFlightClosestCrossing = unknown),
                      receive(flight.flightVerticalSeparation = unknown);
          }//com monitorPlane
```

To model the geometric perception of detecting separation distance and path crossings, the ATC Radar Display object makes the essential calculations and conveys these to the ATCO (flight attributes shown in bold above). In particular, the radar display computes where flight paths are crossing, which flight is closest, lateral and vertical distances between flights. The attributes in bold are only communicated by the display if STCA Optical is operating.

This example illustrates that work practice modeling in Brahms is not concerned with layout of the radar display or how the ATCO is actually scanning the display. The emphasis instead is on what information can be determined from the display and most importantly when the ATCO monitors the display and the duration of that activity.

If the STCA Optical is operating, then the Radar Display will receive a belief from the STCA object that corresponds to the STCA Optical alert. The belief ("flightInBoundary") indicates for a given flight the closest aircraft that violates the STCA Optical minimum separations. Specifically, for a given flight, another aircraft is modeled as being flightInBoundary 120s before the aircraft's approach is less than 6.5 NM lateral separation and 1500 ft vertical separation (BFU Report, p. 37). For example, referring to the above workframe: with respect to flightNumber DHL 3611, if isFlightClosestCrossing is true for the flightClosest (e.g., BTC2937), flightLateralSeparation = 30 NM, and flightVerticalSeparation < 1500 ft, then flightInBoundary will be true for the closest flight (BTC2937, i.e., DHL611.flightInBoundary = BTC2397).

Knowing flightInBoundary is true, ATCO will intervene, based on the following remarks in the BFU Report:

> If the optical STCA had been available it would have resulted in an alert about 2.5 minutes before the collision and almost 2 minutes before the ATCO started his descent instruction to the TU154M. It would have been visible at both the RE and RP radar screen and would have drawn the ATCO's attention to the situation developing in the upper airspace. He would have had ample time to issue instructions to avoid a separation infringement. In this case, TCAS would not have become active. (p. 88-89)

If STCA Optical is not operating, ATCO can receive the beliefs about the closest flight, but flightInBoundary will be unknown (no belief will be received from the AirTrafficControlDisplay object). Instead, the need for intervention can be inferred by ATCO by a thoughtframe:

```
thoughtframe Flight_Conflict_Loss_Of_Separation {
variables:
        foreach(Flight) flight;
        foreach(Flight) otherFlight;
        foreach(AirTrafficControlDisplay) atcDisplay;
when(   known(flight.sectorFrequency ) and
        knownval(flight.isFlightClosestCrossing = true) and
        knownval(flight.flightClosest = otherFlight) and
        known(otherFlight.sectorFrequency ) and
        knownval(current.location = atcDisplay.location) and
        knownval(atcDisplay.minLateralSeparation > flight.flightLateralSeparation) and
        knownval(atcDisplay.minVerticalSeparation > flight.flightVerticalSeparation))
do {
        conclude((flight.flightInBoundary = otherFlight));
}
}//tf Flight_Conflict_Loss_Of_Separation
```

In this thoughtframe, a Brahms-GÜM parameter (minLateralSeparation) models the ATCO's threshold for the separation requiring intervention (i.e., ATCO acts if minLateralSeparation < flightLateralSeparation). A similar parameter models the vertical separation tolerance (set to 1500 feet). These parameters are among the initial conditions that define a scenario and can be modified prior to any simulation run.

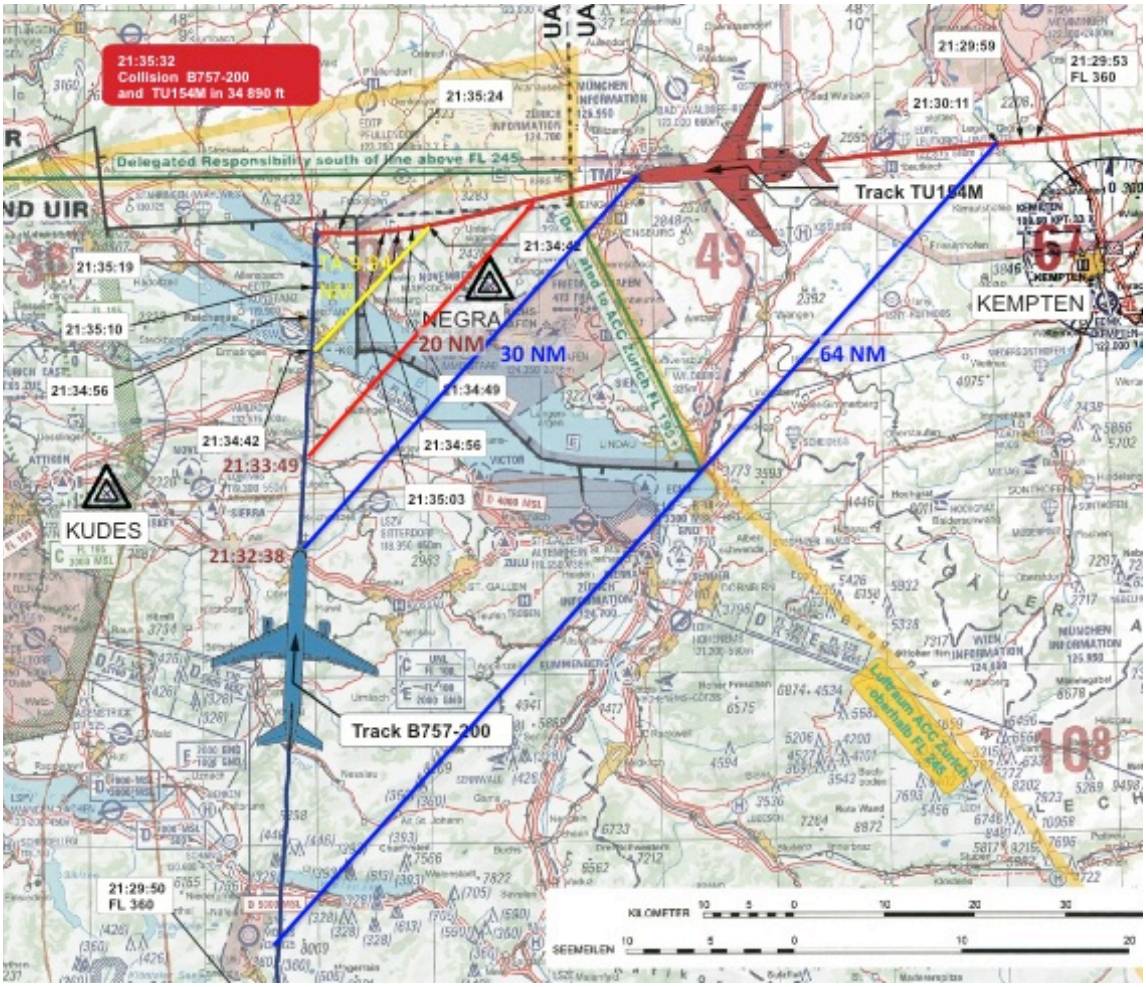


Figure 9-3: Separation distances between DHL (blue) and BTC (red) aircraft at times of BTC handoff to Zurich (64 nm); approximate point where BTC became visible in ARFA Sector radar (30 nm, 21:32:38); recommended last point at which Zurich ATCO should have acted (red, 20 nm at 21:33:49); and where TCAS TA is generated (yellow, 9.94 NM).[32]

Without ethnographic data, we inferred minLateralSeparation from the particulars of the Überlingen events. Originally the value of minLateralSeparation was set to 65 nm given commentary that ATCO might have acted at the time of BTC handoff (at 21:30:11 when separation was 64 nm, BFU Report p. 75). However, the Zurich ATCO testified that the did not believe "the imminent approach to be crucial" at that time (BFU Report, p. 105), which suggests that his minLateralSeparation is substantially less than 64 nm.[33]

[32] The timing of last recommended point for intervention, which is one minute earlier than what actually occurred, is based on a normal rate of descent of 1000 ft/minute to provide 1000 ft of separation (FL350) when the planes would be approximately 7 nm apart (BFU Report, p. 75).

[33] Note that ATCO's remark cited here is inconsistent with the claim elsewhere in the BFU report that "the controller did not notice that the B757-200 had just reached the same flight level and that both airplanes were approaching each other at right angles" (p. 75). Not noticing a fact and not believing it to be crucial are quite different.

Simulation runs confirmed that with minLateralSeparation set to 65 nm, intervention occurs at the time of the BTC handoff and the collision is avoided. Subsequently, minLateralSeparation was reset to 25 nm to better model the Zurich ATCO's behavior at Überlingen.

To understand this tradeoff between 65 nm and 25 nm, consider when and where the key events occurred within the ARFA Sector (Figure 9-3). If for example the STCA Optical had been operational, then an alert would have been provided when separation was less than 31 nm.[34] This alert would have been visible on the radar display when both planes were first displayed in the ARFA Sector because they were 30 nm apart. Using this graphic we can calculate roughly the separation distance at which an ATCO can be expected to take action to redirect flights to avoid separation infringement.

In particular, if the ATCO is attending to this radar display— actively monitoring the aircraft for issues—and his "sensitivity" for taking action (minLateralSeparation) is between 30 nm and 20 nm (the recommended last time to intervene given the aircraft trajectories and velocities), he will catch the imminent collision well before the TCAS TA occurs at about 10 nm. Therefore 25 nm separation was deemed to be a good estimate of when an ATCO would conclude "I need to act on this now" (in contrast with "this is something I'll need to handle eventually").

Furthermore, given that both planes were visible on both workstations during this interval (BFU Report, p. 76) when ATCO was preoccupied by the AEF flight (from the AEF's third call at 21:32:15 until he completes the handoff at 21:34:38), and the separation was less than would urgently require action on his part (i.e., less than 20 nm), we can conclude that *ATCO was not actively monitoring either display for separation issues during this 2 ¼ minute period*.

In summary, Brahms-GÜM represents ATCO as "apprehending" aircraft separation when monitoring the radar display, modeling perception of the STCA Optical alert and immediately acting upon it, if it is operational. The same conclusion can be made through inference when STCA is not operational, but ATCO allows for less separation (25 nm). In both cases, ATCO must be actively monitoring the radar display. Other activities such as handoff for a landing aircraft and responding to calls from aircraft have higher priority, so when that workload increases, less time will be devoted to scanning the radar display for separation issues—which is what occurred in the Überlingen situation.[35]

[34] "Separation" of course takes into account both horizontal and vertical distances. Because the DHL and BTC flights are both at FL360, their separation is their horizontal (lateral) distance apart, represented by the slanted lines on the map graphic.

[35] By modeling the STCA Optical alert as "detectable," its presence could be simulated as becoming known to the ATCO without his scanning the display and cause other activities to be interrupted. However, a unrealized shortcoming in the Brahms language makes the detectable location-independent, which was not desirable (see discussion of language limitations Section 28.4).

As is typical in a Brahms model of work practice, Brahms–GÜM does not represent the design of the interface or the searching, reasoning, and time required to get data from the radar display. Such a simulation could be coupled to Brahms if that level of detail were deemed relevant to understanding and simulating the broader aspects of roles, facilities, and human-systems interaction. In Brahms–GÜM the time is modeled in terms of the periodicity and duration range of activities that involve the radar display (Section 9.7).

9.3 ATCO Intervention Instruction for Separation Infringement

About 11 seconds after resolving the AEF landing situation (by advising the pilot to contact Friedrichshafen directly), ATCO has shifted to the other workstation and is urgently advising BTC to descend. ATCO might have seen the imminent infringement within the ARFA sector (right workstation) display, or more likely shifted his attention to monitoring the broader region on the left workstation's radar display. The aircraft are visible on both displays at that time.

Why the Zurich ATCO chose to instruct the BTC pilots and not DHL and why he told them to descend rather than to climb are not addressed in BFU Investigation Report. The choice might have been based on presence of other aircraft in the sector, the altitude (36,000 ft), confidence in the BTC's location, or a psychological "recency" effect from having last communicated with BTC flight (five minutes prior) rather than the DHL flight (eight minutes prior).

Another likely cause of ATCOs decision is that the control strip for the BTC flight indicated a route that involved lowering to FL350 after FL360 (and hence the critique that this directive should have been issued at the time of the initial interaction with the BTC flight). In Brahms-GÜM the ATCO intervenes with the aircraft at the higher latitude. Without relevant accident or general ethnographic data, we chose to model ATCO's selection by a plausible heuristic, namely that the ATCO might have been reading the radar display from top to bottom, and hence saw the BTC which was further north first. The intervention instruction itself is then based on comparing the flight level for the "next route" to the "current route" on the control strip. If nextRoute.flightLevel <= route.flightLevel, ATCO advises the pilot to descend; otherwise he advises to climb.

The following is one of several related workframes that are part of the activity of resolving a detected separation infringement. The action ("do") effectively determines what the pilot will be told, communicates with the pilot (radioFlightInfo), simulates a "talk delay" of between 2 and 4 seconds and then continuing to monitor the planes.

```
workframe Request_Descend_Next_Flight_Level {
variables:
        forone(FlightProgressStrip) strip;
        forone(FlightPlan) plan;
```

```
        forone(FlightSegment) route;
        forone(FlightSegment) nextRoute;
        forone(double) newFL;
        forone(PilotGroup) pilot;
        forone(AirTrafficControlRadio) radio;
when(   knownval(current.commReason != "flightLevel") and
        knownval(route = flight.route) and          // current flight segment or route
        knownval(plan.flight = flight) and
        knownval(current.location = strip.location) and
        knownval(strip.flightPlan = plan) and
        knownval(strip routes nextRoute) and
        knownval(nextRoute != route) and
        knownval(nextRoute.flightLevel < route.flightLevel) and
        knownval(newFL = route.flightLevel - current.heightSeparation) and
        knownval(pilot.flight = flight) and
        knownval(radio = pilot.commMedium))
do {
        conclude((route.flightLevel = newFL), fc:0);
        conclude((current.flight = flight), fc:0);
        conclude((current.commReceiver = pilot), fc:0);
        conclude((current.commReason = "flightLevel"), fc:0);
        conclude((current.commPerformative = "REQUEST"), fc:0);
        radioFlightInfo(radio, flight, true);          // give new route level
        talkOnRadio(radio, 2, 4, true);
        monitorPlanesInConflict(plane, flight);
}
}//wf Request_Descend_Next_Flight_Level
```

The direction advice given by the Zurich ATCO at Überlingen is not related in an obvious way to which aircraft data is more accurate, given how the radar sweep shows the flights shifting on each rotation. If the BTC data is more current (which the ATCO might be assuming by instructing them to descend), then because DHL is about 90 degrees apart coming from the south, its data will be updated about 4 seconds later. But if this was the ATCO's reasoning, he might also have questioned the accuracy of the DHL flight level displayed (FL360) and waited for it to be updated (see Section 6.8.6 for further discussion).

9.4 TCAS Alerts

TCAS's algorithm for generating advisories is based on the basic formula:

Time = Distance / Speed

Applied to two aircraft on an intersecting route:

Time to Collision (TAU) = (current separation /closing velocity) x 3600

where TAU is in seconds, current separation is "slant range" in nm, and closing velocity of two planes is expressed as nm/hr. Slant range accounts for different

aircraft orientations in 3d space. *TAU thus corresponds to the time until the closest point of approach (CPA).*

To scope the simulation effort, we decided to model TCAS accurately only for two planes flying level at the same altitude (that is, in the same plane). The closing velocity is then given by the law of cosines:

TAU = (current separation in nm / √ (V1*2 + V2*2 - 2 x V1 x V2 x cosine A)) x 3600

where the units of V1 and V2 are nm/hr and A is the angle of intersection.

Brahms-GÜM uses TAU = 48 sec for TA alert and 35 sec for RA alert, the published values for cruise altitudes. Thus, Brahms-GÜM will accurately model when TA and RA are given relative to the collision point (CPA) and closing velocity using the TAU calculation and thresholds. The locations (and hence lateral separation) and clock time will vary from Überlingen events because the simulated speeds of the aircraft are averages given in the BFU Report (actual speeds before TCAS TA occurred are unknown). The relative timing between TA and RA of 13 seconds in the simulations is the same as in the Überlingen sequence of events.

In particular, with average cruise velocities using the formula above with A = 92 degrees, TAU = 48 sec at 8.95 nm lateral separation compared to 9.94 nm that occurred at Überlingen (modeled speed for DHL is 470 kt versus 516 kt actual and for the BTC 463 kt vs. 499 kt; BFU Report, p. 72). The TCAS TA will therefore occur 1 nm closer than Überlingen, which is about 8 seconds, well within the expected accuracy of Brahms-GÜM in replicating the Überlingen scenario. What matters is that the causal relations and relative timings are accurate (e.g., when TCAS TA occurs relative to aircraft closing velocity). As in the real world, how much time is spent on AEF and the other flights will make the difference in whether ATCO acts before or after the TCAS TA, which is the crucial variable affecting the outcome.

The simulated BTC and DHL flights are not exactly at same altitude, but can vary from FL360 according to the initialization of the model (particular scenario definition). Consequently, the TCAS RA advice will vary in different scenarios, sometimes advising DHL to climb and BTC to descend.

9.5 Scenarios Without TCAS

When reconfiguring the model for the different scenario configurations, we discovered that certain simplifications made some behaviors dependent on each other. In effect, reconfiguring the model for different scenarios tested the model's generality and led to improving it. Most typically, a more principled representation was required to allow components and/or behaviors to be configured independently.

For example, well into the modeling process we determined that from a certain perspective TCAS's RA intervention was a cause of the collision—if the DHL flight

had not been advised by TCAS to descend and continued to fly at FL360, then the Zurich ATCO's instruction to the BTC to descend would likely have averted the collision. This hypothesis suggested that for testing the model we add the scenario of TCAS being absent (Section 8.6).

The existence of TCAS is modeled by an initial fact; for example object TCAS_BTC2937, an instance of the TrafficCollisionAvoidanceSystem class, has initial fact (current.inAircraft = Tupolev154_BTC2937). If this fact is omitted from a scenario configuration, then TCAS does not exist in that simulated aircraft.

However, a difficulty arose because the calculations for determining whether a plane has collided with another plane are performed within the TCAS object; removing TCAS resulted in the simulated plane not "knowing" that it had collided, so it would keep flying its planned route.

This design stems from a limitation in Brahms—objects cannot perform calculations when a "move" activity is occurring. In particular, an aircraft object cannot perform calculations while the aircraft is simulated as flying between waypoints, which is always the case during a flight. Thus, it was convenient to have TCAS, which was calculating and detecting separation, communicate the collision event to the aircraft (so that it could appropriately remove itself from the sky).

Therefore, to model the "TCAS doesn't exist" scenario, we added a binary attribute TCAS.isBroadcastOn. When false, TCAS doesn't sound alerts/advisories to pilots or sends information to Navigation Displays, which is equivalent to its absence in that scenario. (Having a redundant object perform the same calculation as TCAS for the purpose of detecting a collision would be more elegant, but would only add to the modeling and testing required, without offering any benefit to the project.)

9.6 Aircraft Crew's Interpretation of TCAS

The BFU Report transcript clearly indicates that the BTC First Officer sitting at the left rear seat was attending to the TCAS display and understood the indication of an approaching aircraft on a potential collision path, as well as the meaning of the subsequent TA and RA:

> For the time between about 21:33:00 hrs and 21:34:41 hrs the CVR recorded crew discussions concerning an airplane approaching from the left which was displayed on the vertical speed indicator (VSI/TRA) which is part of the TCAS. All flight crew members with the exception of the flight engineer were involved in these discussions. These recordings suggest that the crew strived to localize the other airplane as to its position and its flight level. At 21:34:36 hrs, the commander stated: "Here it is in sight", and two seconds later: "Look here, it indicates zero". During the time from 21:34:25 hrs to 21:34:55 hrs, the airplane turned at a bank angle of approximately 10° from a magnetic heading (MH) of 254° to 264°.
>
> At 21:34:42 hrs, TCAS generated a TA ("traffic, traffic"). The CVR recorded that both the PIC and the copilot called out "traffic, traffic." (BFU Report, p. 8)

In the absence of the intervention by the Zurich ATCO to descend, we do not know for sure what the BTC pilot (the commander) would have done. Given their probable detection of the DHL and turn prior to the TCAS TA—and keeping in mind the statistic that 24% of pilots maneuvered opposite to TCAS instruction in one study (Kuchar and Drumm 2007)—it is possible that in the absence of ATCO intervention the BTC crew would have used their own judgment in modifying their course and altitude, regardless of what TCAS instructed.

Because the BFU Report associates pilots' properly following TCAS with their training experience, Brahms-GÜM models pilots that follow TCAS and disregard ATCO instructions as being members of the group of pilots who were trained in TCAS (PilotTCASTrainedGroup). The combination of following or disregarding ATCO and TCAS leads a variety of behaviors that needed to be simulated.

Table 9-1 shows the possible cases that can occur in Brahms-GÜM, where the events of interest are instructions from ATCO or TCAS, and "follows" means that the pilot executes the instruction. For example, if BTC does not follow TCAS, and ATCO speaks before RA occurs (3rd row), then the ATCO instruction is followed and subsequent RA is ignored. By this design, any pilot on any flight in the Brahms simulation can be modeled as following TCAS or not; these combinations define different scenarios that were tested when developing Brahms-GÜM.

Table 9-1: Meaning of "BTC Pilot follows TCAS" (member of PilotTCASTrainedGroup).

BTC Behavior	First Event/Response	Second Event/Response
BTC follows TCAS	ATCO/follows	RA/follows, reversing if necessary
	RA/follows	ATCO/ignores
BTC does not follow TCAS	ATCO/follows	RA/ignores
	RA/ignores	ATCO/follows

9.7 ATCO Workload and Interaction Durations

ATCO's priorities are simulated by workframe priorities: the activities of handling planes that are landing or imminently colliding have the highest priority, carrying out sector handoff and calls from pilots have intermediate priority, and monitoring the radar display has the lowest priority. On examining the ANSA transcript, we realized that omitting the other flights that were requiring handoff and making requests (omitted because of the limited modeling resources) resulted in ATCO have more time to monitor and hence detect the DHL and BTC flights at the S RE (left) workstation.

We can see this workload clearly in the ANSA transcript (BFU Report, pp. 60-61) excerpt below. When the excerpt begins ATCO is at S RE (left workstation) working on phones for approximately 30 seconds. There is a cascade of interruptions,

requiring ATCO to shift between the workstations four times in about 2 ½ minutes, as he handles BTC, AEF, and two other flights (transcript lines begin <receiver> <station calling> <time>):

AEF 1135 calls in, heard on right workstation, apparently having been approved by Karlsruhe to begin descent to Friedrichshafen—
A RE 1135 21:30:07 "Zürich grüezi", äh... AEF äh...1135, descending flight level 80

BTC interrupts, heard on left workstation tuned to different frequency—
S RE 2937 :11 Zurich, good evening, BTC2937

ATCO shifts to right workstation to put off the AEF call so he can handle BTC; AEF flight has not yet appeared on the ARFA Sector (right) radar display—
1135 A RE :18 AEF1135, roger, äh... call you back

ATCO didn't hear BTC's interruption; he shifts to left workstation and asks for repeat; BTC reports back and they carry out handoff—
2937 S RE :26 station calling say again, please

S RE 2937 :28 äh, Zurich, good evening, BTC2937, level 360

ATCO approved DHL to FL360 less than 3 minutes earlier; he either doesn't realize BTC is on potential collision path or decides to handle it later because AEF is waiting and interrupts him—
2397 S RE :33 BTC2937, squawk äh.. 7520

Speaking over ATCO on the frequency broadcast by the right workstation, AEF interrupts; it still is not visible on the ARFA sector radar—
A RE 1135 :34 and AEF äh..1135 is inbound the final approach fix for ILS runway 24

ATCO shifts to right workstation and responds to AEF and defers the pilot a second time—
1135 A RE :44 "ja" expect so, call you back shortly

ATCO shifts back to left workstation to monitor larger airspace (possibly shortly before AEF appears on ARFA radar at 21:30:52); he handoffs two flights during the next minute—
933 S RE 31:15 THA933, contact now Munich 132 decimal 140, good-bye

S RE 933 :20 132 140, THA933, good-bye

5621 S RE :26 MON5621, contact Reims 133 decimal 830, bye-bye

S RE 5621 :32 133 830, MON5621

After a pause, AEF calls in again on right workstation frequency requesting permission to descend for the Friedrichshafen approach—
A RE 1135 :32:15 and AEF äh...1135, request lower

ATCO shifts to right workstation and responds to AEF, which he can now see on radar there—
1135 A RE :19 AEF1135, descend flight level 70

All of this is the prelude to the collision, with ATCO's concern with the phones being down preventing a proper handoff to the Friedrichshafen tower throughout. In reviewing this period, we determined that excluding the other flights had a significant effect on the simulated ATCO's workload, and thus this aspect of the Überlingen scenario (the arrival of other flights requiring attention) should be modeled more accurately.

If we simply count up time handling other flights that appear in ANSA transcript in the period prior to BTC handover, we find that every minute he is handling an average of 1.8 flights requiring 28 sec/flight (including radar interpretation time). All of these conversations occur at the S RE Workstation (left).

Table 9-2: Durations of Communication Activities in ANSA Transcript

Total time period (sec)	616
# flight conversations	18
Duration of conversations	497
Average time/flight (sec)	28
Periodicity (flights/min)	1.8
Touch time (sec)/minute	48

To accurately model the frequency and duration of these conversations, we added a workframe "Handling Other flights" that repeats every minute and has a primitive activity requiring 48 seconds every minute (48 sec = 1.8 * 28). Combined with DHL and BTC arrivals, this should fit the mostly non-stop work that evident in the ANSA transcript. Priority of this WF lies between the high priority activities of handling aircraft separation/landing and the lower priority of scanning the radar display.

In effect, this WF is a placeholder for the missing 13 flights in the scenario configuration. If we included these flights in the model, then the simulated ATCO would be busy handling them. Again, this simplification was part of the scoping decisions made for implementing the base model listed at the start of this chapter.

To model the talking time and multiple back and forth remarks (e.g., confirming an instruction) in the ATCO-pilot conversations, duration for communicating a new flight level (relevant to AEF, BTC, DHL) was set to range between 20 to 30 seconds, in accord with the durations in the transcript. Other handoff interactions are modeled at 15 to 20 seconds including listening and talking time.

Strikingly, ATCO asks AEF to call back twice, so he can process handoffs and aircraft calling in. Following the pattern we observe in the transcript, if the simulated ATCO can't see an aircraft calling in on the radar when he has other flights to handle, he asks the calling pilot to call back. A simulated pilot told to call back does so in 30 seconds.

The simulated ATCO can hear calls broadcast from a radio monitor at a workstation other than where he is sitting. For example, when ATCO is at the left workstation and AEF calls in on radio broadcast by right workstation, he can hear the AEF call and moves to the right workstation to respond. However, unlike what occurred at Überlingen, the ATCO always comprehends what is being said on the other radio, and does not need to ask the caller to repeat the call, even though he might have been talking to someone else when the other call occurred. It would have been possible to simulate this more precisely, so ATCO would not receive communicated beliefs when communicating with someone else, but the effort was not deemed worthwhile.

Finally, the attempts to make a call to Friedrichshafen are modeled at 30 seconds each, which is in addition to time spent interacting with the ATCC CA about the Friedrichshafen control tower's phone number.

As a side effect of having modeled as an aggregate "handle other tasks" activity rather than individually detected and handled flights, the ATCO is simulated as being busy doing this activity for 48 seconds as explained above, during which time he is not actually monitoring the radar display. In particular, he will not detect that any aircraft are closer than 25 nm. Simulation runs showed that the ATCO was so busy "handling other flights" he rarely noticed when BTC and DHL were in conflict at the same time. This problem illustrates indirectly why modeling all objects and processes that might interact is important and what happens when events and actions that might be modeled more properly as composite activities are aggregated into primitives such as "handling other flights."

The method for dealing with this aggregation problem in a Brahms model is to use the "detectable" construct to cause the "handling other flights" activity to be interrupted when aircraft are closer than 25 nm. In effect, to fix one shortcut, we needed to introduce another. In particular, in Brahms-GÜM an aircraft separation fact is asserted as a property of the S RE workstation area when a TCAS TA is triggered. Therefore, if ATCO is in the S RE display location handling other flights (or in any activity for which this is a detectable), he detects the aircraft separation issue and acts upon it (radioing a pilot to climb or descend). If ATCO is the ARFA workstation area, he will not detect that the separation fact. In short, when ATCO works at S RE (left) workstation between time of TCAS TA and RA, he will detect the imminent collision, fitting the behavior of the Zurich ATCO during the events at Überlingen. Put another way, the model is based on the assumption and replicates the behavior that any ATCO working at the radar display would see the imminent collision when the planes are so close as to require a TCAS TA.

9.8 Summary of Design Principles for Modeling Simplifications

To review, the initial "base model" (Section 8.2) was conceived to simulate nominal configurations and behaviors, omitting all of the anomalies thought to have contributed to the Überlingen collision. The base model is a gross simplification that abstracts the actual structure and behaviors of aircraft, crew, automation, etc.

considerable scoping of the base model with only one modeler. Thus the base model, as a first step in simulating complex scenarios, is itself a framework, a kind of simulation sketch, of the basic work system of pilots interacting with an aircraft subsystems and ATCOs interacting with ATCC subsystems. Aircraft are assumed to follow a strict flight plan with a constant cruise speed. Interactions between pilots and air traffic controllers are limited to handoffs and collision intervention. The simulated TCAS is precise only for aircraft flying at the same altitude.

The modeling process was then conceived as incrementally adding to the base model to model absent subsystems and other variations from ideal behavior in a sequence of increasingly complex scenarios such that Überlingen work system could be simulated and the events related to the published transcript and analyses (Section 8.5).

In effect, the research shifted to verifying that the various components interacted properly by testing them in different configurations of the given model components (Section 8.6). This scenario sequence was defined to verify that each aspect was modeled with sufficient fidelity and completeness to simulate the Überlingen collision—with of course the primary interest of revealing what might have occurred if different work system aspects were present or not (e.g., if the phones worked properly). In this respect, each test (new scenario or simulation run of a scenario) constituted an experiment in using the Brahms simulation to test a hypothesis about the role of a factor in the collision. The following chapter describes the results of these experiments and how the model was further refined as interactions and timings became salient and suggested ways to improve the fidelity of the model to better simulate how ATCOs and pilots behaved.

10 Results: Refinement and Analysis of the Generalized Überlingen Model

The previous chapters detailed how Brahms-GÜM was simplified and refined to model the Überlingen work system configuration, focusing on broad work practices (e.g., reading the radar display) and timings of activities (e.g., duration of a typical ATCO-pilot radio conversation).

This chapter begins with an overview of how scenarios are configured, run, and documented (Section 10.1) and details the analysis and refinement process following initial formulation of Brahms-GÜM (Section 10.2).

In the context of this project, "results" relate to the initial question of suitability of the Brahms framework for modeling and simulating complex human-automation systems. The main discussion therefore details the process of establishing Brahms' suitability through a sequence that demonstrates *completeness, generality, and accuracy* of Brahms-GÜM (Section 10.3).

We then demonstrate how the model is *useful* by refining it to answer interesting questions about key factors that, when timings are varied, can change the simulation outcome. In particular, we show (Section 10.4) by defining scenario initial conditions and controlling probabilistic variability of event timings and durations one can establish bounds on when an unpredictable, non-routine event can disrupt normal procedures and in this case lead to a separation infringement.

The chapter ends with a summary of the pivotal events and configurations in the Überlingen scenario, as clarified by the simulation experiments (Section 10.5).

10.1 Logging and Charting Simulation Outcomes

As described, different scenarios are simulated by reconfiguring the initial facts, beliefs, and relations ("parameter settings") of Brahms-GÜM that define aspects of the work system contributing to the collision (Section 8.6). Recall that Brahms-GÜM without such reconfiguration is designed as a "normative" model in which both Zurich ATCOs are present and all subsystems are working properly. (However, in all the scenarios we have configured, the DHL and BTC flights are defined as having intersecting routes.) Generally, just a few behaviors and anomalies are modified to configure Brahms-GÜM for each scenario; these were documented manually and stored with a copy of the edited model and log file of scenario events. Appendix 24 indicates the actual model revisions that define the scenarios.

Key events that occur during the simulation are logged chronologically in a file that constitutes a readable trace of the interactions among the ATCO, pilots, and automated systems. The log is defined by "print" actions in workframes (see annotated example Appendix 26). Logged events include:

- ATCO-pilot interaction regarding handoffs, route changes, including flight

levels and instruction
- Flight strip activities (printing, moving, reading)
- Separation violation events detected by TCAS, including TAU value; TCAS actions; collision or resolution events
- Radar display monitoring activity and flight information read
- Separation violation detected by ATCO when monitoring radar, including flights
- STCA optical or aural alerts, including separation detected
- Agent movements (e.g., ATCO shifting between workstations)
- Aircraft movements, including departure, entering and exiting sectors, waypoint arrival, landing, collision
- Aircraft control changes (e.g., autopilot disengaged)
- Radio frequency tuning and calls, including communicated beliefs
- Phone calls that fail to complete

A spreadsheet was then created to compare the multiple runs of the Überlingen scenario by transcribing data from the chronological logs. This data includes:

- Simulation Run Number
- Whether a collision was avoided
- Brief explanation of events
- Times when the following occurred:
    - BTC Handover to Zurich
    - ATCO Advises BTC; Second intervention (if any)
    - TCAS TA, RA, Second RA (if any)
    - DHL & BTC Auto-Pilot Off
    - AEF calls ZATC first time
    - ATCO informs AEF to call EDNY tower
    - DHL & BTC Cross Paths
- Intervention Advice Sense (Climb/Descend)
    - TCAS RA
    - ATCO
- TCAS TA & RA Range Separation (nm) and Vertical Separation (ft)
- Which flight ATCO advises to avoid separation violation
- ATCO intervenes Before/ During/ After TCAS TA?

The model is also instrumented to record in the file the BTC and DHL aircraft altitudes from the time of the TCAS RA to when their paths cross to facilitate graphing their trajectories and thus making visible whether the aircraft converge or diverge. Another key chart shows for each run relates 1) when the AEF flight contacted ACTO, 2) when ATCO notices the separation infringement, and 3) the occurrences of the TCAS TA and RA. As the analysis proceeded, this chart became the reference point for critiquing and comparing the simulation runs, facilitating detection of patterns such as implausible timings and missing outcomes.

10.2 Summary of Analysis and Refinement Process

To review, the process of experimenting with scenarios and refining Brahms-GÜM proceeded by creating Brahms-GÜM incrementally, adding aspects of the ATS work system relevant to the Überlingen accident. We then tested the model on ten scenarios (Table 8-1) each of which "added back" one putative causal factor (e.g., what if the phones were working?).

At this point, we then added the scenario in which TCAS is disabled, producing the outcomes summarized in Table 10-1.

Moving now from the sequence of building the model to experimenting with reconfigurations, we reformulated the analysis by defining the space of 48 configurations/scenarios implicitly defined by all combinations of "primary causal factors" that the model design allows to be varied by editing initial facts and beliefs.[36] We decided not to run the other 38 valid combinations, pending further analysis of the "Überlingen scenario" (Scenario 1F), that is, in which the work system is configured as it was at the time of the accident. In fact, analyzing and improving Brahms-GÜM by repeatedly running the Überlingen scenario then occupied the remainder of the research effort presented in this report. In effect, we shifted from studying variation in outcomes caused by reconfiguring the work system (different scenarios) to studying variation in outcomes caused by probabilistic events in a single scenario.

Put another way, we realized that there was no point in experimenting with scenario variations until we had verified and validated the Überlingen scenario itself, for which we had substantial quantitative data. Correspondingly, our modeling focus shifted from questions like "what if the phones had been working?" (which predictably prevented the collision from occurring) to the more general question, "Are the range of timings of ATCO actions and aircraft location/velocities generated by the simulation in accord with the events at Überlingen?" In effect we were moving from modeling timings and frequencies of individual behaviors (Chapter 9) to verifying emergent durations and relations of events (e.g., duration between BTC arriving in sector and ATCO advising pilots to descend).

Focusing on verifying timings between different events, resulting from interactions of aircraft arrivals, ATCO, pilots, and TCAS, led us to shift from the final outcome (whether a collision occurs) to the detection of the impending separation violation and what timings of prior events most critically affected detection (e.g., when AEF calls Zurich ATCC). In grasping this overall picture, we were better able to say what aspects of the work system should be emphasized to avoid a separation violation.

---

[36] Referring to Table 10-1, six binary factors yields $2^6 = 64$ combinations, but "BTC pilots follow TCAS" is not meaningful if TCAS is disabled, which gives 48 valid combinations plus the null case, which omits the ATCOs and TCAS (first row of Table 10-1). Of course, other combinations are possible using Brahms-GÜM such as "DHL pilots don't follow TCAS" and an infinity of flight routes and schedules.

Table 10-1: Simulation Outcomes of Test Scenarios, Given BTC and DHL on Collision Course.

Scenario Description	Collision?	# Zurich ATCO	TCAS	BTC Pilots Follow TCAS	Radar & STCA	AEF Flight	Phones
0) Null	YES—no ATC or TCAS intervention; proves flight paths are on collision course and timing leads to intersection near Überlingen	0	N	—	—	—	—
2A[37]) Normal	NO—ATCO observes in radar "loss of separation" between DHL and BTC before a TCAS TA is triggered	2	Y	Y	Y	Y	Y
2B) Normal w/o Phones	NO—Zurich ATCO advices BTC to descend early enough to avoid collision.	2	Y	Y	Y	Y	N[38]
2C) Phones out & Radar degraded, but TCAS rules	NO—BTC advised in time so TCAS not activated	2	Y	Y[39]	N	Y	N
2D) … but Zurich ATCO rules	NO—BTC is advised early enough.	2	Y	N	N	Y	N
1A) Normal-SMOP	NO—Zurich ATCO advises BTC; TCAS announces "Traffic! Traffic!" but RA not activated	1	Y	Y	Y	Y	Y
1B) SMOP w/o Phones	NO—STCA alert compensates for distraction of AEF & phone not working.	1	Y	Y	Y	Y	N
1C) SMOP w/o Radar	NO—BTC pilot already started following TCAS advisory before Zurich ATCO gives descent instructions.	1	Y	Y	N	Y	Y
1D) Actual, but TCAS Followed	NO—BTC pilot already started following TCAS advisory before Zurich ATCO gives descent instructions.	1	Y	Y	N	Y	N
1E) Actual, but TCAS not enabled[40]	YES—Zurich ATCO advises BTC pilot to descend too late while DHL remains on course.	1	N	—	N	Y	N
1F) Überlingen	Depends on when Zurich ATCO intervenes relative to TCAS	1	Y	N	N	Y	N

Correspondingly, we recognized more clearly what actions and timings were not significant and could be fixed in the model, similar to the original decision to only

[37] Numbering indicates number of air traffic controllers in Zurich; analysis ignores potential Karlsruhe contribution.

[38] When phones don't operate, ATCO asks for assistance

[39] BTC following TCAS implies will ignore ATCO if TCAS advises first or will reverse course if Zurich ATCO advises first

[40] Überlingen situation without TCAS. Note: When BTC continues flight (ignores or does not receive advice from Zurich ATCO or TCAS) but DHL reacts to RA and descends at slow rate of 20 feet/sec instead of 30 ft/s, collision occurs. In different runs of 1F, there is no RA so DHL remains on course while Zurich ATCO tells BTC to descend, which avoids collision.

experiment with scenarios in which DHL and BTC were on a collision course. In particular, we recognized that the pilots' manual control of the aircraft to avoid the collision (after being alerted by TCAS or the ATCO) was not the focus of our study and could simplified by making this behavior deterministic. In this respect, we were segmenting the simulation, separating all events that occurred prior to the pilot's detection of the separation violation from those that occurred after.

Subsequent sections describe this refinement process chronologically, detailing some of the particular observations in simulations runs and changes made to the model.

10.3 First Phase: Verifying and Refining Probabilistic Interactions

After initial conversations about Brahms-GÜM with research colleagues focusing on model checking, we realized that variability inherent in different runs of a given scenario (caused by probabilistic durations of primitive activities; Appendix 27) must be documented and understood. Our experiments with different scenarios (Table 10-1) established that Brahms-GÜM could in principle be reconfigured as we intended—that is, the different components interacted without error producing plausible outcomes. But we could not trust or interpret the results further without understanding why a collision is avoided or not in the simulation runs, that is, did the interactions and timings make sense or were spurious events occurring that affected the outcome?

We then began a more systematic process of verifying and validating the model relative by comparing outcomes to the events recorded in the BFU Report. Accordingly, we generated one hundred simulation runs of the Überlingen scenario (1F) and discovered that nineteen resulted in a collision. Again, in these simulation runs, the model, including all initial conditions defining the scenario, are held fixed; the variations are caused by probabilistic durations of primitive activities (other probabilistic features are possible, including particularly uncertainty of an action, but are not used in Brahms-GÜM). In some of these runs no collision will occur because of variances in departure or cruise flight times (i.e., the aircraft arrive at the crossing point of their routes at different times).

We analyzed the first ten runs of the one hundred in which the aircraft were on a collision course, focusing on the interactions of the independently operating processes (radar, TCAS, ATCO). We determined that whether or not a collision occurred was particularly dependent on the timing of ATCO's intervention with the BTC flight relative to the TCAS intervention with DHL. Furthermore, the timing of the late-arriving AEF flight determined when and whether ATCO noticed the separation violation. In particular in the ten simulation runs of the Überlingen scenario at hand, we observed two of the three possible times when ATCO might intervene:

- *Before TA*—No collision occurs: ATCO intervention before TA results in TA not occurring or subsequent RA takes BTC's descent into account and advises

DHL to climb; however, if ATCO intervention is shortly before TA, altitude change might not be sufficient yet to be taken into account by TCAS, but it is sufficient to avoid collision.

- *After RA*—If ATCO intervention occurs sufficiently after RA, no collision occurs because BTC pilots ignore TCAS RA, and DHL has already descended sufficiently so although both aircraft are descending, they cross at different altitudes.

Only the Überlingen case itself (ATCO intervention between TA and RA) is missing from these ten, which is not too surprising because the period between the TA and RA is only about 13 seconds with the given aircraft trajectories and velocities.

Several of the timings in the ten analyzed simulation runs were suspect. For example, when BTC flew level and DHL was advised to descend, how could a collision occur 18 seconds later? This prompted more detailed analysis comparing the simulation logs to the BFU Report details and ANSA transcript. We analyzed outcomes with respect to when ATCO intervenes, why and why not, and especially the sensitivity of the timing of intervention between the TCAS TA and RA relative to a collision occurring.

We discovered validity shortcomings in the model with respect to duration of events and behaviors involving TCAS alerts, ATCO-pilot conversations, radar display of the ARFA sector, and ATCO's attempted phone calls and sensitivity to potential separation infringement. Often we needed to change both how a subsystem was modeled (e.g., the radar display) and how ATCO interacted with it (including activity of monitoring, perception of information, and reaction to information). The main factors we discovered:

- *ATCO too sensitive to separation:* The simulated ATCO was detecting separation violation when the actual Zurich ATCO should have, at the time of BTC handover, well before TCAS TA, and about two minutes before the intervention actually occurred at Überlingen. Analysis surfaced minor issues in the model of the ARFA sector shape and coordinates (Section 9.1). But more importantly, the simulated ATCO's separation threshold for intervening, which becomes pivotal when the STCA optical alert is not functioning, was far too conservative relative to the Zurich ATCO's testimony (detailed in Section 9.2). In particular, the Zurich ATCO testified that certain observed distances were too large to merit action at that time (relative perhaps to priority of other tasks).
- *ATCO not observant enough.* The converse of ATCO intervening too soon was not noticing the aircraft on the radar display when they were already 10 nm or less apart (i.e., between the TCAS TA and RA). In this version ATCO was too busy handling the other flights. Given that he was looking at the radar display during this activity, he might have noticed the separation problem, so

we included this as a "detectable" for the workframes that involved monitoring the display.

Again, we use a detectable to simulate perception. In effect, visual relation of an impending collision is generated as a Brahms "fact" by the TCAS model (essentially "a separation problem exists with aircraft X and Y"). The ATCO detects this fact when he is located at and reading information ("communicating with") a radar display after the TCAS TA occurs.[41]

- *TCAS model too coarse:* The initial TCAS model, only intended as a placeholder for purposes of the base model, as not sufficiently accurate. In particular, the simplified TCAS model caused the TA to be given with a separation distance of 2.4 nm greater (about 15 sec earlier) than what occurred at Überlingen. We determined that the TAU must be calculated accurately with respect to the intersection angle of the aircraft. Comparisons to Überlingen were complicated because the speed of the planes varies and the simulation is based on averages (detailed in Section 9.4).
- *ATCO tasks and timings inaccurate:* The activity durations for attempting phone calls (to Friedrichshafen) and routine ATCO-pilot radio calls were much too short compared to averages in the Überlingen transcript (2 - 4 seconds instead of 20 – 30 seconds). Similarly, Zurich ATCO's activity of handling other planes takes so much time this workload must be modeled as well (detailed in Section 9.7).

Notice that this analysis concerns verifying the accuracy of the model relative to the specific work practice of the Zurich ATCO (e.g., the intervention threshold) as well what we assume to be widely shared, routine practices (e.g., replicating the duration of ATCO's activities). This mixing of arguably non-optimal and acceptable/expected behaviors is characteristic of a *work practice simulation*. The modeling effort straddles between replicating a particular work and style of interactions occurring at Überlingen—a matter of verifying by comparing the simulation to a given design—and modeling the general, regulated practices applicable in other situations—a matter of validating by comparing the simulation to generalizable patterns. In contrast, a typical *process model* would model agents as following regulations and official procedures. Brahms-GÜM could be used in that manner to test normative behaviors, but as the present project illustrates, what is especially of interest for understanding resilience of a work system is what happens when procedures are not followed and/or systems are not operating properly.

[41] This is an example of how behaviors can be "programmed" in Brahms when details are unknown or unimportant relative to the objective of the modeling effort. We have TCAS assert the "visual fact" simply for modeling efficiency; one could just as well have the radar display make the calculation (and consequently simulate other kinds of visual facts that ATCO could detect relative to aircraft locations and trajectories). The detectable could also be probabilistic (say 50% certainty of noticing) to simulate different degrees of attention.

In summary, to this point the model building process involved what are in retrospect standard modeling stages of scoping the model, creating a complete simulation, verifying for accuracy, and generalizing the model:

- *Analyze Work Setting and Scope Model:*
  - Analyze documents to determine causal factors of Überlingen accident; determine the model components (e.g., radar display) and players (e.g., CA) required to simulate these events, building on the Brahms-WMC model.
- *Develop Complete Model:*
  - Develop a general model relative to the Überlingen work system (Brahms-GÜM), defined in terms of a sequence of configurations (scenarios) in which each of the key components can be configured as dysfunctional/absent or functioning normally (e.g., STCA Optical alert).
- *Test Generality*
  - Run and analyze ten scenarios (Table 8-1 and Table 10-1) that combine anomalous behaviors and events in simple ways. Confirm that the model produces meaningful chronology of behaviors and outcomes for each scenario (e.g., the aircraft fly their planned routes, handoffs occur, TCAS operates, ATCO monitors radar).
    These variations establish generality of the model's logic; that is, coordination of work flow occurs (e.g., every arriving flight is announced and handoff occurs), regardless of the particular state of objects and agents and timing or sequencing of events. Put another way, objects and agents act "autonomously" to carry out their work in any simulated context.[42]
- *Test Accuracy:*
  - Verify and validate the simulation outcomes with respect to interactions occurring among ATCO-TCAS-Pilots as measured by separation and timing documented in BFU Report and ANSA Transcript.
  - Simulate the Überlingen scenario (termed "1F") 100 times to determine timing and causal sensitivity of probabilistic interactions.

[42] Note that some scheduled activities are configurable as initial Brahms facts and beliefs (e.g., flight schedule, when second ATCO goes on break). But most dysfunctions are binary (e.g., phones are modeled as working or not). The model could be designed to define timings as initial facts (e.g., when phones begin working again), relate timings to contextual events (e.g., when phones begin working again relative to when maintenance begins), or develop model so timings are emergent (e.g., simulation the maintenance work in some detail). However, notice that this effort would not establish the model's completeness and generality (e.g., ATCO completes the phone call to the tower and hence doesn't need to ask the CA for an alternate number) and would probably not clarify our understanding of what happened at Überlingen. Nevertheless, such modeling such timing variations might be useful if using the simulation to thoroughly explore unexpected interactions.

    - Refine Brahms-GÜM components (radar, TCAS, ATCO and pilot behavior) to improve accuracy and generality, based on analysis of the first ten of the 1F scenario runs.
- *Control Variability:*
    - Verify and compare relative timings and duration of key events in ten Überlingen runs in order to determine how variations in timings affect agent behavior (e.g., affect of time of AEF arrival on whether separation infringement was detected).
    - Determine circumstantial variability that is confounding the analysis by multiplying possible outcomes without adding information (e.g., relative altitudes of BTC and DHL at time of ATCO intervention).
    - Refine the simulation to fit variability consistent with the known facts and work practices (e.g., simplify ATCO choice of aircraft to advise so result is predictable and fixed for given flight paths, as well as being consistent with the scenario of interest).
    - Control confounding variations in aggregate behaviors (e.g., limit range of arrival times in Zurich sector by restricting possible departure times and removing time of flight variations between waypoints; fix velocity curves used to control plane flight when pilots are responding to TCAS/ATCO).

Verification and refinements for accuracy were interleaved with improving the generality of the model. As the information the simulation could provide became more evident, we also fixed behaviors (making them less general) to scope the simulation and analysis effort. Putting this another way, having succeeded in simulating the complex human-automation work system present at the time of the Überlingen accident, the simulation system was itself now complex—it produced many more outcomes than occurred at Überlingen (e.g., ATCO advising BTC to climb because it arrive at higher altitude than DHL) and these outcomes had unpredictable timings and interactions, which made analysis and understanding difficult. The trick is then to control or limit the space of the variations produced by the simulation, without losing generality (e.g., the set of scenarios or model configurations possible is not changed).

In particular, we determined that it would be advantageous for ATCO and TCAS to give the same advice as in each simulation run, replicating the Überlingen events. Prior to this time, in some simulation runs ATCO would contact the DHL or instruct the BTC to climb instead of to descend. Accordingly, we adjusted the variability affecting the aircraft flights, so on entering the Zurich sector BTC would be above the DHL aircraft, thus ensuring that TCAS would instruct BTC to climb and DHL to descend (retaining generality of the ATCO model). We modified the heuristic used by ATCO for selecting the aircraft to advise, so that he would select the aircraft with the higher latitude (matching the BTC flight). The direction advised (to climb or descend) is determined by the aircraft's control strip; BTC's plan indicated FL350, suggesting advice to descend (Section 9.3).

Notice that generality of Brahms-GÜM is retained because behaviors were not fixed by "hard-wire" programming, such as by writing an ATCO workframe, "Contact the BTC flight." WFs and TFs never reference specific flights or scenario-specific attributes. Instead, an abstraction of the states and behaviors of the given scenario is determined that is general and plausible; and because it fits the present situation it will affect the outcome.

More generally, we were figuring out how to model and simulate events that were known or believed with high confidence to arise from a cumulative effect whose outcome was *probabilistic* (e.g., relative altitudes of the aircraft at the time of the separation infringement), events that were also apparently *chaotic* (e.g., the variations in velocity of the two aircraft during descent), and events for which a causal story could be constructed, but might be *arbitrary* (e.g., ATCO's choice of DHL vs. BTC). We had begun wanting Brahms-GÜM to be very general, as the name attests. For example, the simulation allows for other flights to be included in initial conditions, for the BTC and DHL flights in particular to have different flight plans, and so on. But in focusing in the details of the interactions among different flights, ATCO's actions, TCAS, and the pilots' actions, some of this generality in the model (e.g., having ATCO follow a sophisticated set of rules for choosing which flight to advise) led to variances in the simulation runs that did not inform or clarify the temporal and causal effects of pivotal events.

So for example just as it provided little information to run a simulation in which the phone system is operating normally or the second ATCO is on duty (Table 8-1), we were not learning anything by having BTC arrive at an altitude about 200 feet below the DHL (such that TCAS advises DHL to climb and ATCO still advises BTC to descend). Hence we adjusted the model to reduce variability in flight paths.

This leads to an important realization about the modeling process: *Controlling unnecessary variability (relative to our interest and modeling purposes) makes the results of the simulation runs more predictable and hence reduces the complexity of the analysis required.*

Nevertheless, generality is not sacrificed and is evident in the simulation runs. For example, if during a simulation ATCO advises BTC to descend sufficiently soon, the TCAS RA will advise DHL to climb; and similarly, if ATCO intervenes sufficiently after the TCAS RA, he will detect that DHL is descending (which by then might also be reported by DHL) and advise BTC to climb.

We reduce the space of outcomes that occur in multiple simulation runs of the same scenario by defining the initial conditions to fit the scenario of interest and limiting variability in the modeled behaviors of the people (as well as the behaviors of the automation if the simulation were being used to redesign TCAS for example).

Limiting variability of modeled behaviors is accomplished in part by *reducing variability of the circumstances* in which that behavior occurs (e.g., location and timing of the aircraft) as well as by *making simplifying assumptions* about choices people were making, while still modeling activities in a general way (e.g., perhaps ATCO scans the radar display from top to bottom, so he sees the BTC flight first and consequently chooses to call them). Such simplification is merited when we have no evidence or strong theory to explain why people did what they did during the Überlingen events. Going forward in creating similar Brahms models, we would perhaps be more careful in using Occam's Razor to generalize from the known behaviors we are simulating without making the model more complicated than the available data suggests or that the modeling purposes require.

This is the first Brahms model we have created in which the variability of primitive activity durations (e.g., the time duration a flight waits after nominal departure time to get clearance for takeoff) can significantly change what occurs in the simulation. This sensitivity to timing, leading to unpredictable sequencing and durations, is entirely consistent with the claim that the work system at the time of the accident was complex (Section 5.9). Some of the complexity—the many ways in which the events at Überlingen could have occurred differently (such as ATCO contacting DHL instead of BTC)—can be "damped down" when the matter at hand is to model and understand better what occurred with given initial conditions (e.g., when BTC and DHL arrive in the Zurich sector). Scoping what needed to be more accurate and what could be fixed by making reasonable assumptions and simplifications became even more important during the next phase, as we shifted to using the simulation to provide metrics about specific causal interactions.

10.4 Second Phase: Defining Questions and Scoping Simulation Variability

During this phase of modeling, we shifted from "making the model" to "using the model." We started to study the simulation itself as a system and improve it in interesting ways. Having improved the model so that ATCO's activities and decisions as well as TCAS's alerts were simulated more accurately, we examined again when and why a collision was averted or not, focusing on the temporal sensitivity and variability of the interactions among ATCO, TCAS, and the pilots.

We started to realized what questions the simulation might answer, such as "Given that the arrival of the AEF flight is disrupting the ATCO's monitoring of the larger airspace (e.g., if it arrives sufficiently late, no collision occurs), what is the period (relative to the BTC and DHL flights paths) when AEF's arrival can cause collision? During this period, does a collision always occur or are there variations of how the AEF handoff occurs, such that sometimes the separation infringement is averted?"

We emphasize that the model was not designed to answer such questions, rather we were focusing on what factors to include (Section 6.6) and how to simulate complicated behaviors (Chapter 9). It never occurred to us until a year into the project that the Brahms-GÜM simulation might provide new information about sensitivity of event times and durations. However, once this interest developed, our

analysis indicated ways in which to bound the simulation (what to include and with what fidelity) to provide quantified answers to these questions.

Refinements for improving accuracy continued when errors were found, but *what fidelity was desirable* (and hence where it was discovered to be missing) became less a matter of replicating Überlingen events to controlling variability that was unnecessarily complicating our analysis. This process began in the first phase, but took on a different character as we studied the simulation itself to understand what affected whether and when ATCO detected the impending collision. Consequently, how pilots reacted to ATCO/TCAS and how the trajectories of the aircraft varied represented a different part of the simulation in which the model could be simplified to facilitate analyzing what came before.

In particular, we began to appreciate better the shape of the aircraft trajectories and their sensitivity to the pilots' reaction time, and the effect of TCAS's and ATCO's instructions to expedite descent:

- On comparing the simulation log to the BFU Report Appendix 3 timeline, we realized for the first time that the BTC pilot during the Überlingen accident reacted to the ATCO immediately, before acknowledging or informing ATCO about his actions (and this was the reason for ATCO's subsequent "expedite descent" instruction 7 seconds after the initial intervention).
- Despite the Auto-Pilot being disengaged while the ATCO was speaking, the BTC isn't shown in the BFU Report timeline as having dropped 200 feet until 7 seconds after it AP disengaged.
- The altitude data in the BFU Report timeline enables determining the descent rate in response to TCAS and ATCO instructions (e.g., average rate of descent for both aircraft is 31 ft/sec, with BTC descent starting about 2 seconds sooner than DHL because of timing of ATCO instruction).
- Modeling of some interactions that didn't always occur, involving combinations of behaviors, needed to be verified (e.g., if a pilot who is already descending/climbing because of ATCO intervention is later directed by TCAS RA to follow the same advice, the pilot should continue descent/climb and inform ATCO about the TCAS advisory).

In this set of ten simulation runs (again eliminating those in which the aircraft are not on collision course), ATCO intervened three times between the TCAS TA and RA. When ATCO notices separation problem depends on when ATCO completes handoff of AEF flight to Friedrichshafen (by informing the pilot to contact the tower directly). In order to simulate how the separation problem is not noticed until immediately after the AEF handoff is completed, the simulated ATCO's activity of handling a landing flight has priority over general monitoring of the radar, and the methods require a great deal of time, effectively fixating on this task at the expense of everything else.

However, another combination of events occurred in one of the simulation runs, in which monitoring occurs during the activity of handling the AEF flight. In particular, after the third failed phone call attempt, ATCO requests the CA to get an alternate phone number.[43] While waiting for CA, ATCO moves to S RE workstation to handle other flights, which includes the activity of monitoring the airspace. In particular, ATCO might move to the S RE workstation to respond to BTC's arrival. As he is reading the BTC flight control strip, he might also monitor other flights. When monitoring, he might detect that separation of BTC and DHL flights is less than 25 nm/1500 ft and intervene. Also as described in the previous section, if the TCAS TA has occurred, he will detect the visual fact of an impending collision, leading him to intervene (with a priority that interrupts other activities).[44]

We do not know what the Zurich ATCO was actually doing while waiting for the CA or how much time he spent talking with her after her return. In the transcript there is a period of 48 seconds in which nothing occurs between his third call attempt ending and his call to AEF to tell the pilots to contact the control tower themselves. As the simulation shows, if he looked at the radar display he would see the obvious separation violation.

It is also possible for the AEF flight to arrive much later (a range of at least 5 minutes is possible because of probabilistic variations in pilot activities that affect flight times). Because he is not preoccupied by the AEF flight, ATCO might then notice the separation problem prior to the TCAS TA. Subsequently, he becomes busy monitoring the conflict and this delays the AEF handoff. In particular, the priority of handling a conflict will prevent ATCO from noticing that the AEF flight is getting close to the TOD point, which would trigger telling the pilot to contact the tower directly.

Analysis of these ten runs also suggested that rate of descent of each plane was crucial. On examining the BFU Report data we observed that the planes were chasing each other down, with DHL starting about 4 seconds after BTC and catching up by accelerating its rate of descent after the TCAS "Increase Descent!" instruction. In the last 4 seconds DHL accelerates to -71 ft/sec.

A few critical points became salient in studying when descent begins relative to the instruction given to the pilots:

---

[43] The modeled landing handoff procedure involves three attempts to call the tower, asking CA for alternate phone number, using the alternative number after CA returns (not mentioned in ANSA transcript); giving up and asking pilot to call tower directly. Also, the ATCO will skip to the last step if the aircraft gets close to the point (Top-of-Descent; TOD) where descent to final approach altitude for landing would normally begin.

[44] Priorities of relevant activities in this version of Brahms-GÜM: Deciding on conflict resolution (climb/descent) for intervention (priority = 100); respond to pilot call-in arrival in sector (60); respond to pilot report about TCAS climb/descent (50); handling landing flight (40); handling "other flights" (10); monitoring radar display (5). See Appendix 23 for a list of all workframe priorities.

1) "AP disengage" shown in the BFU Report timeline is a key moment in the sequence, it proves that the pilots are responding to an instruction by taking manual control of the aircraft.
2) ATCO's initial instruction to BTC to descend requires 7 seconds; TCAS's instruction requires 2 seconds. During ATCO's utterance a BTC crew member repeats "Descend!"; AP is disengaged before ATCO finishes speaking.[45]
3) For both BTC and DHL, the AP is disengaged in the next second after the pilot hears the descend instruction (according to timeline in BFU Report, Appendix 3).
4) Altitude and descent rates (given by tables in BFU Report, pp. 57 and 59) indicate that the BTC aircraft changes altitude about 4 seconds after AP disengage and DHL changes about 6 seconds. We used 5 seconds in the model, consistent with assumption made in the TCAS algorithm (p. 69).

The rate of descent tables also reveal that the vertical rate of descent is not constant but changes for both aircraft:

1) According to the BFU Report tables, the BTC aircraft vertical rate of descent between 21:34:40 – 21:34:58 is about -1.6 ft/sec, roughly 30 ft drop. (The table shows the altitude as the same during these 18 seconds [35968 ft] because data resolution is 128 ft.) Also, aircraft climbs at 21:34:50. Given that AP is engaged and in prior period the vertical velocity indicates a drift up, this appears to be an AP compensation to get it back to a set altitude, not a response by BTC pilots to TCAS TA.
2) Tables indicate that aircraft both start at 35968 ft and descend to 35840 ft within 2 seconds of each other. DHL is two seconds behind because ATCO has intervened with BTC just prior to TCAS RA.
3) Descent rate curves of the two aircraft are similar, reaching about -30 ft/sec (-1800 ft/min, which is consistent with recommended -1500 to -2000 ft/min, p. 50). However, DHL catches up with BTC perhaps due to TCAS "Increase descent!" at 21:35:12, increasing descent to about -40 ft/sec within about 4 seconds (-2400 ft/min, somewhat less than -2500 ft/min recommended for responding TCAS "increase" RA, p. 50). BTC is reaching -39 ft/second at same time, perhaps in response to ATCO "Expedite descent!" instruction at 21:35:04.
4) In final four seconds before collision, DHL increases descent rate to -71 ft/seconds; BTC has increased to -39 ft/sec about 14 sec after ATCO Expedite, but that drops to about -30 ft/sec in final ten seconds, so they are holding back.
5) Both crews appear to recognize that the preferred (or inevitable) passing is for the DHL to be below the BTC aircraft. BTC NAV Mid remarks at

[45] A Brahms agent cannot carry out another action while "listening" in a communication activity. Therefore, the duration of the ATCO intervention utterances is shortened in the model accordingly to match when the AP disengage action occurred (i.e., to allow the pilot to react at the same time relative to the beginning of the utterance). This limitation is discussed in Appendix 28.2.

35:14 “It is going below us”; BTC FE R spots DHL on the left 7 seconds later. BFU Report graphic in Appendix 7 shows the DHL below at time of collision.

Having now understood the complexity of the trajectories, and not having any data to properly relate the varying rates of descent to pilot actions, let alone their beliefs, we determined that modeling the varying rates of descent lies outside the scope of the present Brahms-GÜM effort. Modeling descent properly would involve modeling the pilots’ perceptions, inferences, and actions in coordinating the instruction to expedite descent while projecting the point of intersection and working within the dynamics of the aircraft.

In summary, this second phase analysis of Überlingen simulation runs revealed that descent of the two aircraft begins within 2 seconds of each other, and the varying descent accounts for the collision. The BFU Report states, “Once the TU154M and B757-200 had initiated the descent the outcome was left to chance” (p. 85), meaning that it is too late for the ATCO to influence events, and whether a collision occurs depends on how the pilots zigzag through the airspace.

Until this point in developing Brahms-GÜM, we had always been most interested in examining a simulation run to know whether a collision occurred. But now it became apparent that what is most important with respect to the purpose of this project is not whether a collision occurs—which is highly dependent on pilot maneuvers to avert collision—but on the occurrence of a separation violation and the causal circumstances. In short, at this point in creating the Brahms-GÜM our attention shifted from attempting to replicate the collision, to focusing on ATCO’s actions in the few minutes prior to the separation infringement (which is marked by the TCAS RA, in the final second of ATCO’s instruction to the BTC).

Having decided that we had no interest in what occurred after the pilots disengaged the AP, we abandoned our model of the pilot’s control of the aircraft during an emergency descent, why ATCO intervenes a second time, and the pilot’s adjustment of velocity in response to “expedite” instruction. Instead, we incorporated the descent tables from the BFU Report, such that the two simulated aircraft replicate the exact changes in velocity that occurred at Überlingen. To allow for uncertainty in aircraft locations, we defined a collision as occurring if the aircraft are within 100 feet of each other at the crossing point. This guarantees that a collision will occur in the simulation if TCAS instructs DHL to descend and ATCO instructs BTC similarly within a few seconds of the relative times these events occurred at Überlingen. (As shown below, the instructions may also lead to divergence, as well as BTC flying level while DHL climbs or descends.)

At this time, as we became more sensitive to variations of a few seconds in the simulated events, we discovered a programming error in the TCAS model’s TAU calculation that was causing the advisories to occur too late. We also realized—on seeing the simulated TCAS issue “conflict cleared” for planes that were within 100

feet vertically and still on collision course—that the initial base model of TCAS released the advisory when TAU values no longer being below minimum thresholds. However, TAU is not used for determining when the conflict is cleared, specifically because the Vertical TAU can become quite large if the vertical changes in velocity are not constant.

The correct algorithm for "Weakening Advisories" is fairly complex, with a few special cases (FAA 2011, p. 33). The simplest case involves one plane climbing and the other descending, called a "Positive Corrective RA" (FAA 2011, p. 32):

> During an RA, if the CAS logic determines that the response to a Positive RA has provided ALIM feet of vertical separation prior to CPA (i.e. the aircraft have become safely separated in altitude while not yet safely separated in range) before CPA, the initial RA will be weakened to either a Do Not Descend RA (after an initial Climb RA) or a Do Not Climb RA (after an initial Descend RA). This is done to minimize the displacement from the TCAS aircraft's original altitude.
>
> In Version 7.0 and later, after ALIM [Altitude Limit, difference in altitude between the two aircraft] feet of separation has been achieved, the resulting Do Not Descend or Do Not Climb RA is designated as corrective. In Version 7.0, the RA is annunciated as "Adjust Vertical Speed, Adjust."

For altitudes between 20 and 40 thousand feet, ALIM is 600 feet, with a minimum of 300 feet required at the lowest altitudes. Rather than modeling the multiple directives used by TCAS including "level off," which are intended to avoid more displacement than necessary or the pilot reversing too steeply, the modeled TCAS only issues "Clear of Conflict." The pilot returns to the previously assigned altitude, informing ATCO (or acknowledging ATCO's instruction at that point). Of course, TCAS announces clearance of conflict after the closest point of approach (CPA).

10.5 Third Phase: Answering Questions from Multiple Simulation Runs

At this point we had refined the simulation for accuracy and sufficient fidelity to enable relating key events leading up to the *intervention* by ATCO and TCAS. We contrast this perspective with the alternative scoping, "events leading up to the *collision*," which includes pilot conversations and control of the aircraft. In this section we review ten simulation runs of the Überlingen scenario (again eliminating a few runs in which the aircraft are not on collision course) and show what information can be learned from the emergent interactions that occur. The basic qualitative data appears in Table 10-2, indicating whether the planes collide; a brief explanation of what occurred; the direction of ATCO and TCAS advice; and when the ATCO intervened relative to the time of the TCAS TA and RA.

Table 10-2: Outcomes of ten simulation runs of Überlingen scenario. Bold indicates greatest potential for collision (ATCO intervenes between TA and RA; both aircraft descending)

Run #	Collide?	Explanation	ATCO-BTC	TCAS RA-DHL	ATCO relative TA/RA
1	No	TCAS detects BTC plane descending due to ATCO; so advises DHL to Climb.	Descend	Climb	Before
2	No	TCAS detects BTC plane descending due to ATCO, so advises DHL to Climb.	Descend	Climb	Before
3	No	TCAS detects BTC plane descending due to ATCO, so advises DHL to Climb. AEF flight arrives very late after TCAS TA.	Descend	Climb	Before
4	No	DHL TCAS Descend; BTC above. Planes crossed > 100 ft vertical separation	**Descend**	**Descend**	**During**
5	**YES**	DHL TCAS Descend; BTC above. BTC AP turned off at DHL RA. Planes crossed < 20 ft vertical separation	**Descend**	**Descend**	**During**
6	No	DHL TCAS Descend; BTC above. ATCO later than RA, so BTC level. Planes crossed > 600 ft vertical separation	**Descend**	**Descend**	After
7	No	DHL TCAS Descend; BTC above. DHL AP turned off 2 seconds before BTC. Planes crossed > 100 ft vertical separation	**Descend**	**Descend**	**During**
8	**YES**	DHL TCAS Descend; BTC above. BTC AP turned off at RA. Planes crossed < 50 feet vertical separation	**Descend**	**Descend**	**During**
9	No	DHL TCAS Descend; BTC above. Planes crossed > 200 ft vertical separation	**Descend**	**Descend**	**During**
10	No	DHL TCAS Descend; BTC above. ATCO later than RA, so BTC level. Planes crossed > 600 ft vertical separation	**Descend**	**Descend**	After

As explained in previous sections, the model had been "tuned" so BTC flying is almost always slightly below DHL near the time of the interventions.[46] Nevertheless, interesting variations are immediately apparent:

1. When ATCO intervenes before TCAS TA, but planes have not separated sufficiently, TCAS will take BTC's descent into account, advising DHL to climb (runs 1, 2, 3).
2. When ATCO intervenes between TA and RA (as at Überlingen; runs 4, 5, 7, 8, 9), outcome depends on timing with 2/5 runs resulting in a collision (defined as vertical separation less than 100 feet, which is within 128 ft

[46] DHL is still climbing to FL360 when it arrives in the sector in the south, while BTC has been at FL360. Variability in the auto-pilot model can result in the DHL aircraft being below BTC at the time of the intervention, in which case TCAS would both advise DHL to climb. ATCO always advises BTC to descend following the control strip plan, unless pilot calls in reporting an RA. These runs are also using the fixed descent tables from the BFU Report. Again, our focus is on events leading up to intervention; for our purpose, the paths after intervention are only of dramatic interest.

precision of BFU Report values; runs 5, 8); TCAS issues "expedite" when vertical separation is improving quickly enough (runs 4, 5, 7, 8)—ATCO issues "expedite" to BTC in all ten runs because BTC pilots have not acknowledged his instruction.

3. When ATCO intervenes about 10 seconds after TCAS RA—which BTC pilots ignore (or might be imagined as discussing for a long time)—BTC continues flying level while DHL descends, so they miss each other, separated by more than 600 ft at the crossing point (runs 6, 10). In other runs, we have also observed that ATCO intervenes so late, he actually takes the pilots' report about TCAS RA instructions into account.

Table 10-3 provides the timing data from the point of TCAS and ATCO resolution advisories from the ten simulation runs of the Überlingen scenario:

- Separation at the time of TCAS advisories (average TA = 9.43 nm and RA = 6.96 nm) fits Überlingen values closely (TA = 9.94 nm and RA = 7.11 nm); differences are caused by aircraft speed estimates used by the simulation and simplification of the TCAS model.[47]
- Vertical separation of aircraft at the time of TCAS TA and RA is less than 10 ft if ATCO doesn't intervene before TA, fitting BFU Report which shows them as being at the same altitude.
- Period between DHL AP off to evidence of response is 13 - 10 sec; Überlingen is about 12 sec.
- Autopilot (AP) is switched off (Table 10-4) for DHL 1 – 3 sec after TCAS RA (average 2 sec; same as Überlingen), for BTC 8 – 11 sec after ATCO begins intervention (average 9 sec; Überlingen is 7 sec).[48]
- Cross-path time is when paths intersect, with either a collision or one aircraft flying above the other. Two collisions (runs 5, 8) occur 34 sec and 35 sec after TCAS RA; Überlingen was 36 sec.

[47] Because of a limitation in the Brahms engine, requiring all object activities to take time, the current model configuration results in the BTC and DHL TAs not being simultaneous, but up to 5 seconds apart (6 runs; in 4 runs it is 1 second). Because simulated pilots do not respond to the TCAS TA, this discrepancy has no effect on the outcomes.

[48] The two seconds discrepancy between these simulation runs and the timing at Überlingen is an artifact of how conversation between people (ATCO and pilot) using the radio is modeled compared to detecting the communication from an object (TCAS). The radio interaction— Pilot -> Aircraft Radio -> ATCC Radio -> ATCO—requires at least 7 seconds. Pilot response then involves reading auto-pilot setting on MCP (min = 1 sec, max = 2 sec) and disengaging auto-pilot (min = 1 sec, max = 2 sec). Finding ways to modify this timing could result in different collision outcomes, but again our focus at this point in the research is on events leading up to the intervention, not what occurs afterwards.

Table 10-3: Separation and Timing of TCAS and ATCO Intervention

Run	TA Time	TA Range Separation (nm)	TA Vertical Separation (feet)	ATCO Advises BTC	RA Time	RA Range Separation (nm)	RA Vertical Separation (feet)
1	21:33:54	9.75	297.00	21:33:07	21:34:09	7.02	454.20
2	21:33:53	9.44	133.00	21:33:24	21:34:06	7.09	270.19
3	21:33:27	9.45	377.00	21:32:32	21:33:42	6.85	534.20
4	21:33:47	9.68	7.00	21:33:59	21:34:05	6.77	7.00
5	21:34:02	8.48	9.00	21:34:04	21:34:12	6.94	9.00
6	21:34:04	9.40	8.00	21:34:31	21:34:19	7.02	8.00
7	21:33:23	9.69	9.00	21:33:37	21:33:41	6.94	9.00
8	21:33:34	9.58	7.00	21:33:43	21:33:51	6.90	7.00
9	21:33:16	9.61	3.00	21:33:21	21:33:34	7.03	3.00
10	21:33:30	9.24	5.00	21:34:00	21:33:44	7.00	5.00

Table 10-4: Timing of Key Events in Überlingen Simulations

Run	BTC Handoff	AEF first calls ZATC	ATCO informs AEF call EDNY tower	DHL Auto - Pilot Off	BTC Auto - Pilot Off	Cross-Path Time	Interval AEF arrives before TA	ATCO relative TA/RA
1	21:32:37	21:32:19	21:35:25	21:34:12	21:33:15	21:34:49	0:01:35	Before
2	21:32:46	21:32:30	21:35:46	21:34:09	21:33:32	21:34:44	0:01:23	Before
3	21:31:15	21:33:32	21:35:39	21:33:45	21:32:40	21:34:16	after	Before
4	21:31:53	21:32:46	21:36:56	21:34:07	21:34:10	21:34:37	0:01:01	**During**
5	21:31:46	21:32:28	21:35:43	21:34:14	21:34:12	21:34:46	0:01:34	**During**
6	21:32:13	21:33:06	21:35:41	21:34:20	21:34:41	21:34:53	0:00:58	After
7	21:31:26	21:32:45	21:36:06	21:33:43	21:33:45	21:34:13	0:00:38	**During**
8	21:31:28	21:32:09	21:35:16	21:33:53	21:33:51	21:34:26	0:01:25	**During**
9	21:32:45	21:32:29	21:36:10	21:33:37	21:33:29	21:34:07	0:00:47	**During**
10	21:32:17	21:31:59	21:35:02	21:33:47	21:34:09	21:34:18	0:01:31	After

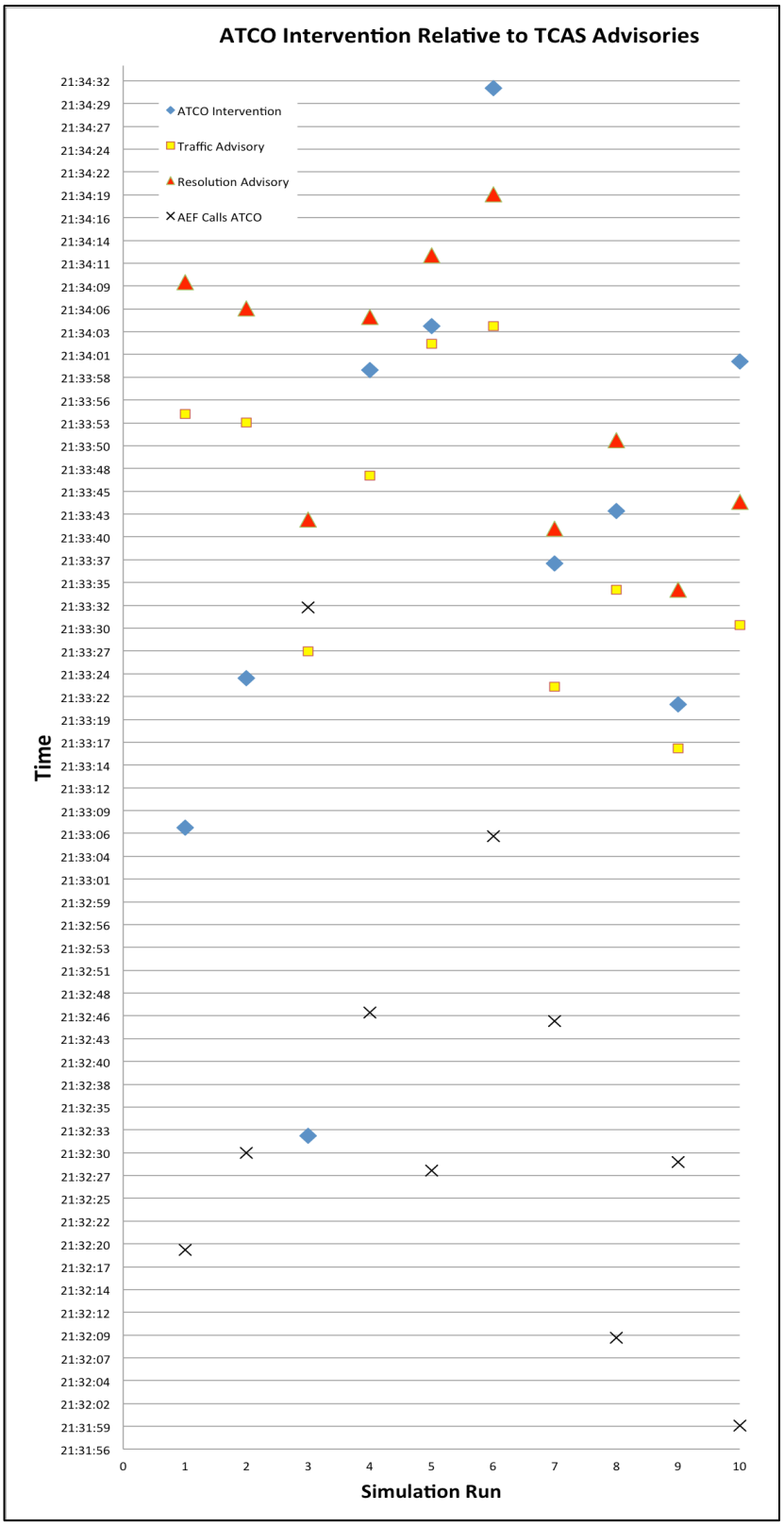


Figure 10-1: Key Events in Ten Runs of Überlingen Scenario

Together with the use of the descent tables for specifying aircraft maneuvers, the data from these tables shows that we have replicated quite well the visible response of the pilots and aircraft from the point of the TCAS RA and ATCO initial intervention.

However, as explained we view this part of the simulation as only having some "dramatic" interest because our primary interest is not how the pilots control the aircraft after an intervention, but the events and interactions affecting when ATCO intervenes. Table 10-4 provides timing of three known important events: When BTC arrives in the sector, when AEF 1135 first contacts ATCO[49], and when ATCO informs AEF to contact the tower directly. Figure 10-1 shows graphically the relation of AEF's first call to the times of TCAS TA, RA, and ATCO intervention.

The results show that when Brahms-GÜM is configured for the Überlingen scenario that in one case AEF's first call is not until after the TCAS TA (run 3; note location of "X" in Figure 10-1). Consequently ATCO advises the BTC to descend almost a minute before the TA, thus representing a normal, non-disrupted interaction.[50]

Otherwise, in the other nine runs, AEF first calls about 1.5 minutes or less before the TCAS TA; during the Überlingen events the interval was over 4.5 minutes, about which 2.5 minutes appears to be dedicated to this flight exclusively. In the simulation the multiple phone calls are completed sooner, rather than the third call occurring when AEF contacts ATCO the third time. Consequently, in some cases even though the handoff is not complete, the ATCO has an opportunity to monitor the radar display, such that the collision is averted (runs 1, 2; Figure 10-2).[51]

We also observe that if AEF contacts ATCO one minute or less before the TCAS TA, then ATCO will not intervene until after TA (runs 4, 5, 7, 8, 9) and possibly after the RA itself (runs 6, 10). Any outcome at that point is then possible.

When ATCO intervenes in the period between the TA and RA (runs 4, 5, 7, 8, 9), collision is possible, as at Überlingen. That is, ATCO has to intervene before TA advising BTC descent for BTC to respond sufficiently for TCAS to advise DHL to climb. In runs 4 and 7 (Figure 10-4) collision is narrowly averted because BTC

[49] As explained previously (e.g., see Appendix 17), during the Überlingen events, ATCO made his first phone call to Friedrichshafen more than four minutes before AEF contacted him for the landing handoff. Given that many other flights are handled during this period and the separation problem doesn't require handling until after the call, it seems reasonable to view AEF's first call as marking when ATCO became distracted from other tasks.

[50] The simulated ATCO considers flights closer than 25 nm and vertical separation of less than 1,500 feet to be within each other's boundary, hence requiring a change in flight path.

[51] An air traffic controller who reviewed this case with us suggests that verifying once that a phone is not working should be sufficient, and the aircraft should have been told to contact the tower directly then. The Zurich ATCO devoted 25 seconds to this third call (21:32:50-21:33:15), the second time trying the bypass number, followed by 48 seconds discussing with the CA "the options of relaying the information via Munich or contacting the technicians" (BFU Report, p. 83)—during this 1 min 13 seconds the separation problem should have been detected and acted upon.

begins descent 4 and 5 seconds after the TCAS RA, which is sufficient for a narrow miss (just over 100 feet). In run 9 the BTC descent begins 5 seconds before the RA, hence the aircraft miss by more than 200 feet). Runs 5 and 8 (Figure 10-3) lead to collision because the TCAS RA and BTC AP disengage occur at the same time, as happened at Überlingen. Because the model uses the Überlingen descent tables to control the BTC and DHL aircraft during the emergency descent, simulation matches the paths of the aircraft at Überlingen guaranteeing a collision (within defined range of error). In both cases, TCAS didn't instruct DHL to climb because BTC was above DHL at that time and of course hadn't begun its descent.

When ATCO intervenes after the RA (runs 6, 10; Figure 10-5), the simulated BTC pilots ignore the RA advice and continue level flight, which itself averts the collision—even though ATCO advises BTC to descend (which implies not considering that DHL is below them). We of course do not know what the BTC pilots would have done if ATCO hadn't intervened. With more than one pilot interpreting TCAS correctly, it appears likely the BTC would have climbed. Also, the PF's pulling back on the control column six seconds after the TCAS RA and thus slowing descent (BFU Report, p. 8) indicates that they were weighing which instruction to follow—the ATCO's second call to expedite descent then tipped the balance.

The final AEF handoff (directing the pilots to contact the tower) always occurs in the simulation after the TCAS RA; at Überlingen it occurred prior to the TA. However, during the process of debugging the model other simulation runs produced different sequences, but were discarded because a Brahms engine bug prevented running the simulation to completion. These runs included examples in which the AEF handoff was completed prior to the TA and runs in which AEF reaches time of descent (TOD; when ATCO must shift strategy for trying phones to telling AEF to contact tower directly), just prior to TA or RA. It should also be possible for a sequence to occur in which the DHL pilot radios to Zurich about following the TCAS RA instruction, but ATCO might not hear the DHL pilot (or not immediately respond) because he is busy giving final directions to the AEF flight at the ARFA workstation.

These variations suggest that we could run experiments perhaps using model checking methods to explore more systematically what happens if we adjust the AEF flight time and how the ATCO handles the phone calls. We expect that a more observant ATCO will detect the separation problem in time, and an ATCO persisting (apparently fixated on the handoff problem) will not.

As it stands, the simulation results are interesting because they show the value of not fixing the model so rigorously according to what happened at Überlingen, even for what we have characterized as the "Überlingen scenario." By allowing variability of AEF flight arrival time for example, we discovered that collision can occur even if the AEF arrived later than it did at Überlingen.

To avoid getting lost in these details, it is important to remember that from a wider systemic perspective the separation violation didn't only occur at Überlingen

because of the arrival time of the AEF flight. Rather the skyguide company had tolerated a deviant form of SMOP during night operations: consequently nobody was carrying out the role of the supervisor (DL) in the ATCC. Nobody was responsible for the system, particularly during the maintenance process. Otherwise ATCO would have been informed that STCA Optical alert was not functioning and that the backup phones had been disabled.

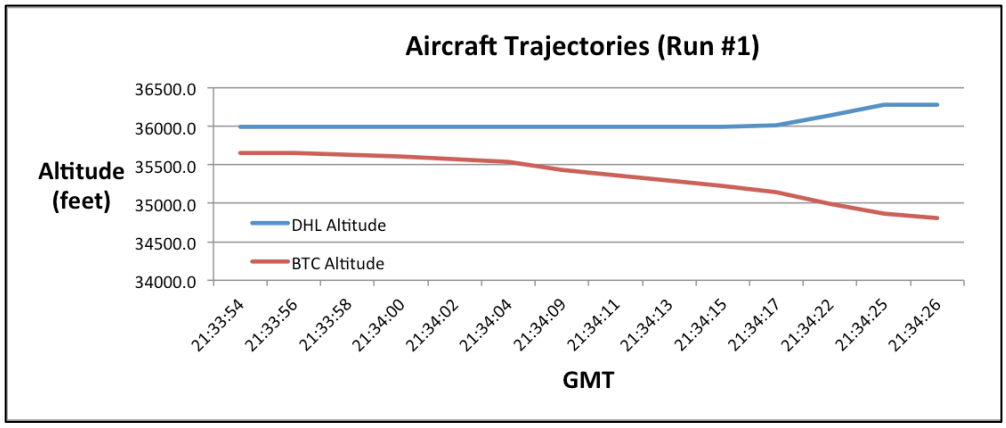


Figure 10-2: Simulation Run #1— TCAS detects BTC descending from earlier ATCO intervention and advises DHL to climb.

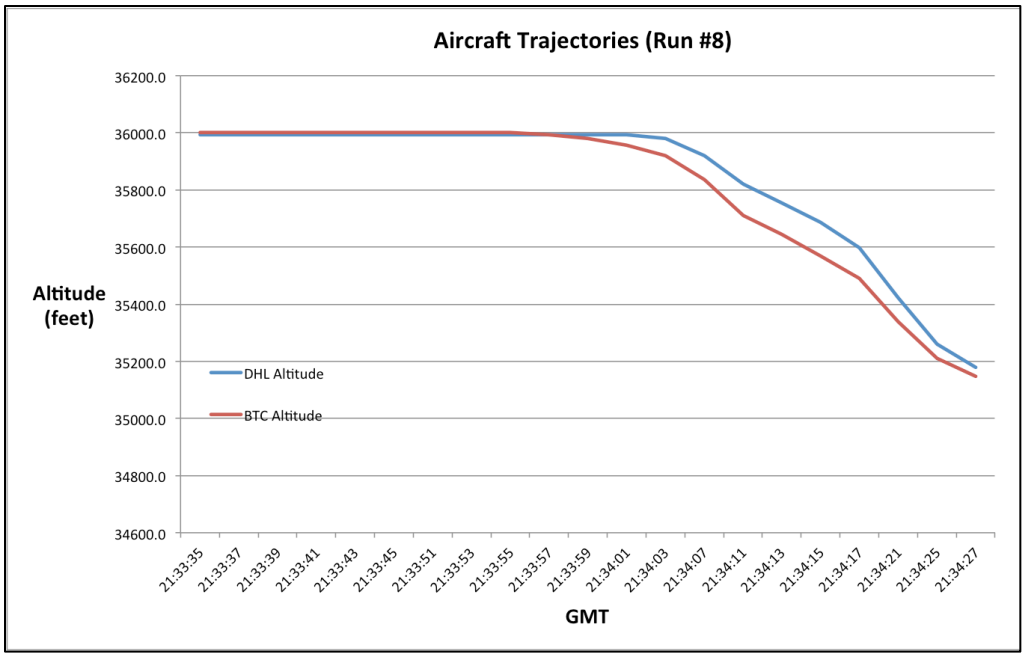


Figure 10-3: Simulation Run #8—Similar to Überlingen: TCAS advises DHL descend; BTC is above. ATCO advises descent before RA; BTC autopilot disengages at time of RA. Planes crossed < 50 feet vertical separation (collision).

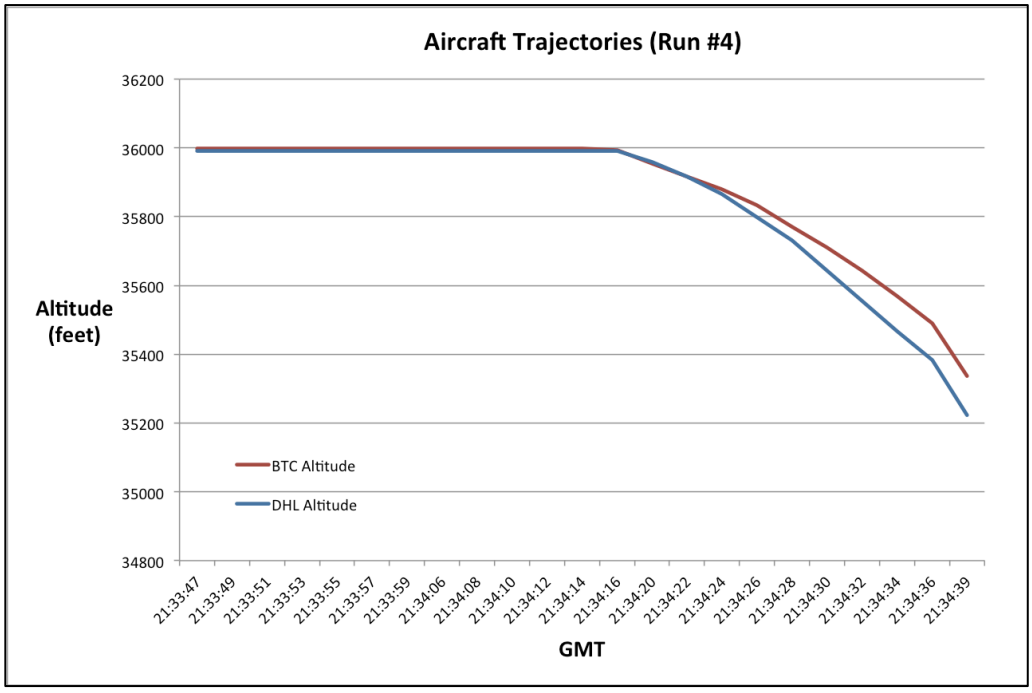


Figure 10-4: Simulation Run #4—TCAS RA advised DHL descend; BTC is above. Planes crossed > 100 ft vertical separation.

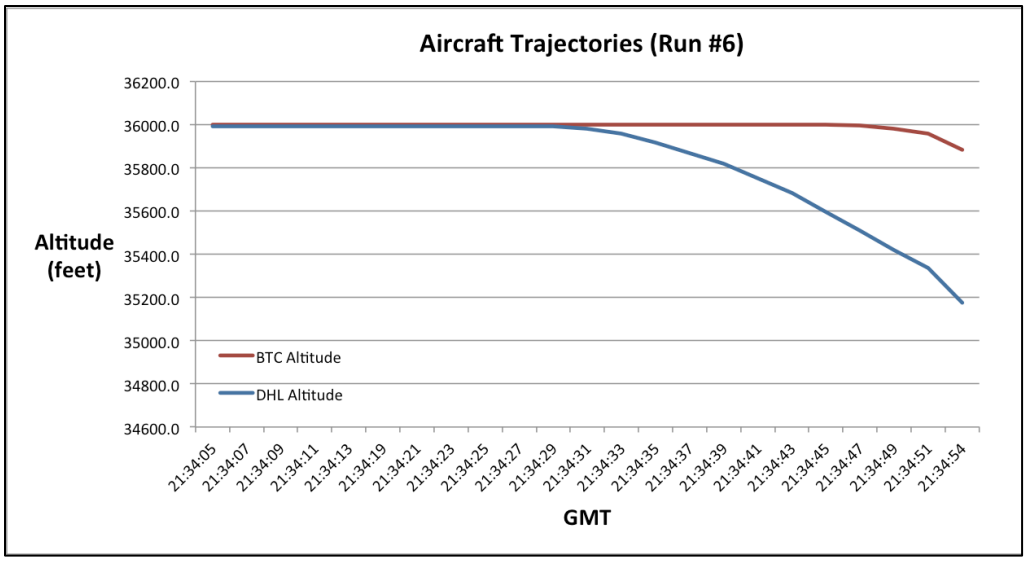


Figure 10-5: Simulation Run #6—TCAS advises DHL to descend; BTC is above. BTC ignores RA; ATCO then intervenes referring to control strip to advise descent. Planes crossed > 600 ft vertical separation.

11 Discussion: "Authority and Automation" Research Theme

An important aspect of the "Authority and Autonomy" research theme is understanding the nature of authority in a mixed-initiative system of people and automated systems.

Most human factors analyses focus on automation of part of the pilot's role, particularly pilots' understanding of the automation mode. The TCAS issue central to the Überlingen case has a somewhat different character: TCAS does not control the plane and therefore is not a form of *flight systems automation* in the usual sense. Rather TCAS instructs the pilot how to fly the plane. The issue of *authority* arises for TCAS when contrary instructions are given by another source, as in the case of Überlingen between an ATCO and other officers in the cockpit.

11.1 Understanding "Authority"

In Section 3.2 we cited two definitions from the research announcement related to this research:

- *Authority* refers to having the right, or power, to exercise controls or issue air traffic commands that impact the position, velocity, and/or attitude of aircraft during operations.
- *Autonomy (or automation)* refers to a function or system that can operate independently of pilot or air traffic controller intervention.

In this chapter we elaborate and critically examine this definition of authority with respect to the issues that people in the aviation system face in working with other people and autonomous systems and within overlapping structures of authority (e.g., airlines, air traffic control, federal and international regulations).

11.1.1 Two senses of "authority"

In both ordinary conversation and within the social sciences "authority" has two distinguishable senses. The first sense of authority refers to the combination of *the right to perform a given action and the ability or power to perform it*. This is the sense used in the NASA NRA definition; we might call this "authorization." The second sense of authority refers to a particular class of actions: *the act of instructing, requesting or ordering another actor to perform a given action*. This is a second-order form of authority by which one agent has the right to delegate or command another agent to carry out an action. For simplicity, we will call this the authority to "command and control."

This distinction can be clarified by a simple example. Imagine a closed door with the sign: "Entrance by authorized personnel only." Actor A may legitimately enter through the door because they have the authority to do so. With regard to the door, actor B may have a different kind of authority: the right and ability to instruct A to enter. Thus A has the right to enter; B has the jurisdiction to command and control A's actions. In everyday life, A might be a child and B a parent, or A a citizen and B a

policeman. (Other distinctions are possible considering the granting of a resource, such as a key that enables entering the door. Another resource, such as a completed form, may then give A the authority to request a key from an officer C who has the authority to physically provide keys. Such types and stages for giving authority are common in computer systems.)

It is important to distinguish between the *authority to act* and *the authority to command or control that action*, because they pose different questions for actors in situations of conflicting authority.

The authority to act is involved when an actor must ask: "Do I have the authority to do X?' The authority to command and control is involved when actor A must ask: "Does B have the authority to tell me to do X?" As we see in the Überlingen case and other aviation accidents and incidents, both kinds of authority may be involved, requiring actors to make both kinds of determination: "Am I permitted to do X?" and "May B tell me to do X?" For the BTC pilots, this ambiguity took the form of what authority a pilot has when two other agents (ATCO and TCAS) are both believed to have authority to command a pilot what to do.

11.1.2 Legitimate authority

The classic work on the authority to command and control, the authority to order someone else to do something, is that of the sociologist Max Weber. Weber raises the question of what the basis is for legitimate authority: authority that is recognized by the parties involved to be morally correct or proper, as opposed to brute force. Weber's research focused on political and religious authority, but is relevant to the aviation context, in raising the question that faces any actor within a system of where authority resides in any given situation.

Weber's discussion distinguishes three basic types of legitimate authority: rational-legal, traditional, and charismatic (Weber 1999).

Rational-legal authority depends on formal rules, and structured positions. Thus, an air traffic controller has legal authority, based on their position within the formal structure of the FAA, their training and certification for that position, and the air traffic control structure's rules of procedure. In this case, the air traffic controller has *authority to instruct*, but it is still within the power of the flight crew not to obey an air traffic controller's command. Indeed, there are situations in which disobeying air traffic control is the correct and legal decision on the part of the flight crew (e.g., see ANSA+AIRRADIO 2004, which refers to ATCO and TCAS as providing "assistance" and "advice," "recommendations").

The second type of authority is *traditional authority,* which may depend on established customs. A political example is primogeniture: on the death of a king, the authority moves to his oldest son. This type of authority is not relevant in the aviation safety context.

Finally, there is *charismatic authority*, which depends on the particular characteristics of an individual, as judged by those who choose follow the authority of that individual. Weber's key examples are religious charisma. However, even in a technical environment, there are those who are considered to have authority not on the basis of their formal position alone, but on the basis of their personality, demeanor, and experience. For example, we might credit a particular researcher as being an "authority in the field" and hence "having power to influence thought and opinion, intellectual influence" (Merriam-Webster 2013). Such an person has social standing because of past deeds and/or presence with respect to the standards and values of the group.

In the Mayday video of the Überlingen incident[52], one of the Russian pilots interviewed argues that naturally one would obey a passionate urgent human voice, rather than a flat, dispassionate mechanical instruction. This argument implies a kind of charismatic authority based on human passion or intensity.

11.1.3 Authority as a relation/contract

It is important to understand that authority can not be understood solely as a property of a single actor: it is an interactive relation established by the actions of social actors. Both of the two types of authority involve actions within a social system.

In particular, a right to perform an action exists within a social system, insofar as the right is granted and the actions can be judged by others having appropriate authority (which is also socially granted and judged). Thus, although a pilot may have the physical capability to control an aircraft to climb or descend to a certain flight level, without instructions from an air traffic controller, the pilot does not have the right to do so.

At issue in research on "authority and automation" is whether, how, and under what circumstances automation has authority. A given automation system may have capabilities to carry out goals through a combination of sensors/instruments and effectors/controllers. Putting people under the control of today's automation is problematic because of the both the limitations of today's technology and the standing of systems relative to human society.

Put another way, today's automation cannot engage in relations of responsibility with people in the sense that people do with each other. The most obvious limitation is technological: Most automation, like TCAS, operates within a fixed ontology for modeling the world, provided by the human designers/programmers. System behavior may be adaptable and flexible within the ontology, but it cannot contribute to the group's ongoing, value-based reconceptualization of the meaning of terms/parameters and their functional effect in operating procedures (e.g., see

[52] Cineflix: *Mayday* (http://www.cineflixproductions.com/shows/28-Mayday)

Clancey 1997). A system like TCAS cannot recognize that an event has occurred that lies outside its specification (e.g., ATCO intervening between the TA and RA), and also decide on-the-fly how to respond (e.g., attempt to convince the pilots that its advice is better because it has more complete information).

We might imagine that someday automated systems may have the capability to conceive and judge, such as what we ascribe to the Zurich ATCO when we ask why he didn't inquire about and test the effects of the maintenance. Automated systems might have human abilities to invent and articulate opportunities for improving practice through variations within the regulations (improvisations) or formulate conventionally allowed violations (workarounds). Then the notion of *responsibility* would shift from the issue of functional capability to the social domain. Could an automated system be a *participant* in an activity if it is not subject to the same laws and can be penalized? Would society agree to develop a community in which automated systems that have interests and emotions are treated reciprocally as persons, and so provide them a social identity?

Such questions highlight the fact that regardless of how future technology and society might develop, systems like TCAS are in a different class entirely. Today's automation lacks the ability to constructively contribute to work practices and thus has no ability—through lack of *interests, values, and emotions*—to have an equal social relation with people (of course, even among people the issue of "equal standing under the law" has been a matter of debate in matters of illegal immigration, marriage, etc.).

Simply put, TCAS and similar automation cannot be held *socially responsible* for their actions in the same sense that we hold a pilot or ATCO responsible. In this respect, we may say that TCAS interfered with ATCO's authority, abrogating his ability to carry out his responsibilities. But it is highly problematic to say that TCAS insofar as it is not a member of society—a social actor—has the *authority* to remove ATCO's rights. Rather TCAS's actions override the ATCO, who may be presumed to have himself abdicated his authority by failing to act sooner: "TCAS as an autonomous on-board warning device…only serves the purpose to avoid collisions between aircraft by obedience to the collision warnings, when air traffic control has failed to fulfill its duties" (ANSA+AIRADIO 2004, p. 76).

The question for the pilots of "who has authority, TCAS or ATCO?" reduces perhaps to something less philosophically complex than "Who is in control of my actions?" and is better stated instead as, "Where is the threat and what action should I take?" Because it is universally granted that the pilot is ultimately responsible for safety, the practical problem a pilot faces is not reasoning about authority, per se, but interpreting and acting on the information provided by different sources. He must do this cognizant of the operational context in which another pilot in an unseen aircraft may be following TCAS's instruction.

11.1.4 Multiple regimes of authority

We have discussed that authority is not a property of an actor, but rather a relation of actions among social actors. Even this statement, though, is a simplification of the issue of authority in the aviation system.

It is easy to think of authority as organized in a single hierarchical structure: within the cockpit, the captain, first officer and navigation officer (as in the case of the BTC Tupelov crew). However this structure is itself embedded within a larger authority structure of the airline, the local aviation authority, etc. A good example of an authority structure is provided by Rasmussen and Svedung's socio-technical model of systems operations (Figure 11-1).

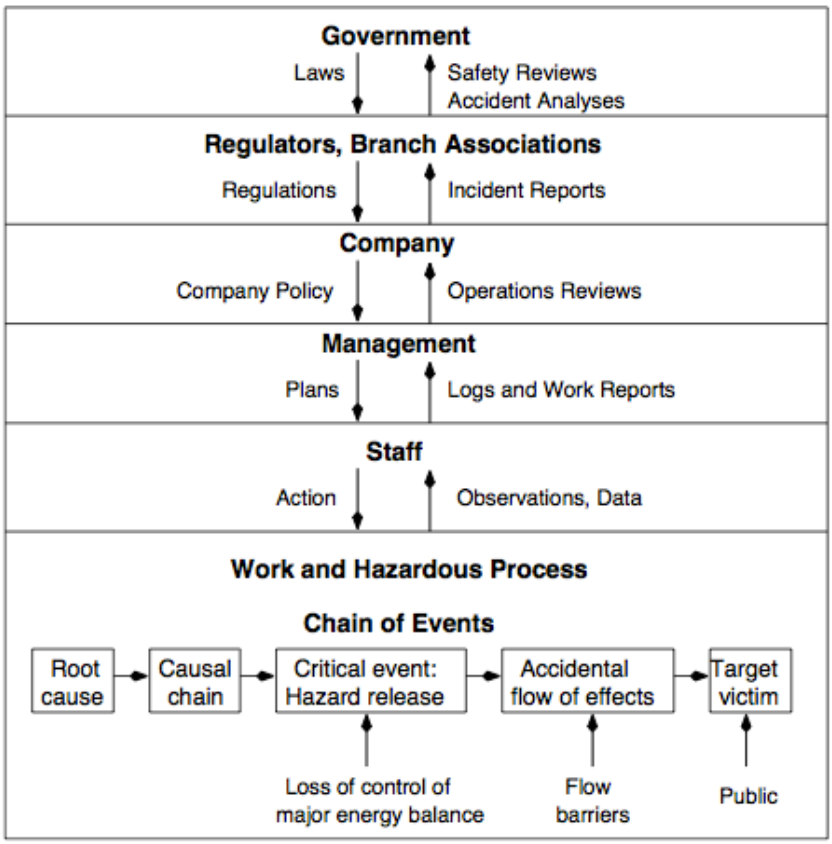


Figure 11-1: Rasmussen and Svedung's socio-technical model of system operations (Leveson, 2004, Figure 2, p. 10)

This model is helpful because it broadens the issue of authority from the immediate context of the cockpit to the larger system in which it operates. However, it still focuses on a single system in which authority is managed.

If we consider authority issues in the Überlingen case, we see multiple authority regimes, which overlap in complex ways (Figure 11-2).

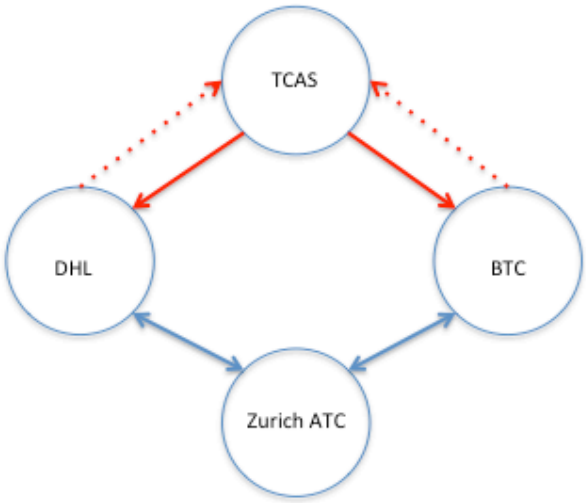


Figure 11-2: Direct authority relations in Überlingen Accident.

The immediate actors in the accident are the crews of the two aircraft, DHL and BTC. They are each in communication with Zurich Air Traffic Control, though not directly with each other. Although the crews cannot communicate, their aircraft's TCAS systems are continuously exchanging flight data.

The authority structures in the two cockpits are fundamentally clear, although there is an ambiguity in the authority relation in the Tupelov cockpit between the captain and the training captain concerning who is in command of the aircraft.

The following excerpts from the BFU Report offer several analyses of the authority relations among the BTC crew:

> The normal crew structure [of the BTC flight] had been altered by the inclusion of the Instructor, but the crew had restructured and operated in accordance with the normal procedures. The Instructor was in a familiar role as PIC and PNF, while the Commander in the left seat would have been less familiar in his role as PF without the responsibilities of being PIC. The duties of the navigator and flight engineer remained unchanged, while the copilot had no assigned duties.
>
> As the conflict situation developed the PNF was responsible for handling radio communications and was to acknowledge the ATC descent instruction. He was initially diverted from this task as he clarified his decision (as PIC) to the flight deck crew to follow the ATC instructed descent. His response to the second ATC descent instruction was immediate.
>
> The PNF was also expected to monitor and support the PF in the execution of any flight manoeuvres, including the ATC instructed descent. When the TCAS event started, his duties expanded by the need for a visual search for the conflicting traffic. It is probable that he at least monitored the PF as the descent was initiated, but he then trained his attention on the visual search. He did not advise the PF that they were cleared level of FL350. At this time the PNF's attention was concentrated on the visual search, and was probably centred in the wrong sector. At 21:35:12 hrs

the ATC controller passed the erroneous information that the conflict traffic was in the TU154M's 2 o'clock position. This increased the confusion of the situation, and the distress of the crew. The crew was probably affected in their capacity to perform their tasks and distracted from the visual contact with the B757-200 in their 10 o'clock position.

Nine seconds before the collision the PIC asked "where is it?" and the copilot replied "here, on the left", implying that the PIC had concentrated his search in the sector suggested by ATC, while the copilot had maintained visual contact with the B757-200 on their left.

About two seconds prior to the collision, the PF pulled the control column hard back, probably in response to visual contact with the now rapidly expanding B757-200. The time available did not allow the avoidance manoeuvre to take effect.

Hierarchy in the cockpit

Except for the inclusion of the Instructor, the flight deck crew was familiar with each other's behaviour and position within the team. The usual flight deck hierarchy would have seen the Commander as PIC and the copilot next in the chain of command, with the navigator holding a more authoritative position than the flight engineer. For this flight it was prescribed according to NPP (comparable to Operations Manual) that the instructor be the PIC. The copilot was relegated to an uncertain and undefined position, as he had no assigned responsibilities within the crew. (BFU Report, p. 98 -99)

According to documentation (approval of flights) provided by the operator the pilot sitting on the left was the commander (PIC). According to documentation (instruction for the conduct of flights) provided by the Aviation Ministry the instructor sitting on the right was the pilot in command (PIC).

According to the regulations of the aircraft operator, Barcelona airport was classified as an aerodrome in mountainous terrain. Each pilot flying to this destination had to make at least two flights with an instructor. For the commander (under supervision), this was the second flight to Barcelona. He was sitting in the front of the cockpit on the left while the instructor was sitting on the right. The instructor was – in the opinion of the BFU - the pilot in command. (BFU Report, p. 11)

The crews of both aircraft are subject to the regulations of their respective airlines in the form of company policies, training, etc. They are also subject to the authority of Eurocontrol, the European Union's air traffic control agency, while they are in European airspace. Because both airlines fly in international airspaces in addition to the European Union, the airlines, and the crews of the two flights are at other times subject to the authority of other air traffic control agencies as well (Figure 11-3). Although Eurocontrol has jurisdiction for BTC's flight in the Zurich sector, the primary authority for Bashkirian Airlines is the State Civil Aviation Authority of the Russian Federation.

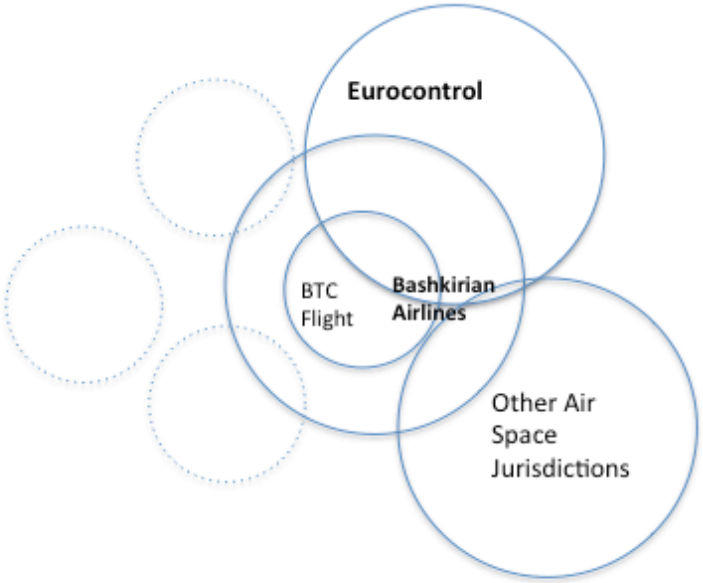


Figure 11-3: Authority Relations for BTC crew with Bashkirian Airlines, Eurocontrol, and other air space jurisdictions; DHL has analogous relations.

These overlapping authority relations are directly relevant to the Überlingen accident because at this time, TCAS was not mandated for aircraft flying within the airspace of the Russian Federation. Therefore the DHL and Bashkirian Airlines crews had different training in TCAS. The DHL crew had simulator training; the BTC crew had only written training. Bashkirian Airlines mainly flew within the airspace of the Russian Federation, which at that time did not require TCAS within its own airspace.

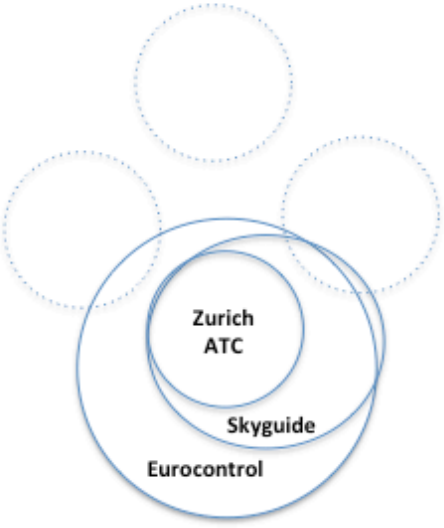


Figure 11-4: Authority Relations for Zurich Air Traffic Control Center

Once we move from the two aircraft to the ATCC, the issue becomes yet more complicated. The Zurich Air Traffic Control Center operates within a different though overlapping authority structure, subject to a different control hierarchy and different rules that those governing aircraft. Zurich Air Traffic Control Center is

subject to the regulatory authority of the European Union's Eurocontrol. It is also subject to the regulations and policies of its parent company, skyguide. Skyguide is also subject to regulations of the Swiss government, which we have not shown here.

A final actor is TCAS. Much research is required to determine how the organizational authority structure influenced the design, certification, and its implementation, including how it was incorporated in the airspace and air traffic control regulations.

We could add other authority regimes to this diagram; for example the national divisions of airspace into civil and military sectors led to a rather narrow corridor of civil airspace in this region of Germany and Switzerland. Nevertheless, this diagram of immediate actors is sufficiently complex to show that authority is not a simple matter, even at the sharp end of an accident, and becomes even more complex as our perspective is broadened to understand the socio-historical and socio-technological context of the Überlingen work system on that July evening in 2002.

11.2 Authority in the Context of Human-Automation Systems

The Authority and Autonomy research theme considers authority specifically in the context of human-automation systems. Although an automated function may be well-defined, the complement of tasks left to the human manager(s)—*the effect on work practice—may be ambiguous and conflicted,* as the Überlingen incident reveals.

TCAS's function involves formulating and issuing an instruction to pilots that will avoid a collision if acted upon strictly and quickly enough. Insofar as we cannot say that a pilot must always, inevitably, with no exception execute what TCAS instructs—*which every regulation concerning TCAS is careful not to say*—the effect on the pilot's situation is neither well-defined nor well understood. Instead, the pilot's response is left general ("insure the safety of the aircraft") and his actions left open to judgment on the spot—following TCAS is contingent (see discussion in Section 4.3). In fact the pilot's job has been made more complicated because now he must relate what TCAS says to the remarks of other crew members, a possible ATCO intervention, and his own judgment.

We may say that the function of TCAS is simply to advise—it has no authority to command the pilot insofar as he is not required to obey. Nevertheless, the pilot is obligated to take this advice into account, weighing the possibility of false alerts (Kochenderfer et al. 2012a, b; Kuchar and Drumm 2007) and considering how the pilot on the other aircraft might be responding. It is interesting in this respect to compare the "Climb, Climb!" instruction to the "Pull up!" advice issued for a stall. Unlike the stall advice, the effectiveness of the TCAS RA depends on the actions of pilots in another aircraft.

In related work, So Young Kim[53] describes the "technology-centered perspective" that specifies automation capabilities in terms of "function allocations," making the point that automation can not truly have final authority. Citing a wide range of studies, Kim summarizes the problems that have arisen in flight system automation:

- The crew's assigned functions are scattered across the flight deck and do not necessarily work to their strengths.
- People are assigned to monitoring automation, despite consistent findings that they are ineffective at this task.
- The technology-centered perspective allocates functions to the automation based on its capabilities.
- The complementary structure of the work to be performed by people is inefficient and incoherent, which may make their role ambiguous.

With respect to "authority," So Young recommends that this concept be interpreted and designed with respect to responsibility, meaning accountability:

> [W]hereas "authority" is generally used to describe who is given the resources to perform a function in operational sense, "responsibility" is used to identify who will be held accountable in an organizational and legal sense for the outcome.

But unless automation can be "proven to provide safety in all foreseeable operating conditions," people must remain responsible for the outcome of what automation actions. This requires continuous monitoring of automated systems, which if impractical requires trusting the automation to work autonomously.

If the person "who is held responsible does not have the resources and capability to act with authority," then there will be a mismatch between authority and automation. The information seeking and monitoring workload of the person will be substantially increased. Furthermore, it will be desirable to give people "the capabilities and the resources to judge and intervene to override automation's functions if necessary," characterized as the "responsibility-authority double-bind" (Woods, 1985). Kim stresses that this problem originates in designing automation by allocating functions to automation without regard for the capabilities of people. Hutchins (2000) adopts the total systems design perspective that "the flight deck as a whole should be viewed as a cognitive system," (p. 54). His design method focuses on patterns of information flow, creating redundant (cross-checking) processing among the crew (p. 72).

This is an important point: We will be performing an important service if we keep focused on how the various NextGen automated systems interact with one another and with the people who must manage them. A similar point is made in a recent

[53] Kim, So Young 2011. MODEL-BASED METRICS OF HUMAN-AUTOMATION FUNCTION ALLOCATION IN COMPLEX WORK ENVIRONMENTS, Georgia Institute of Technology Dissertation. August 2011. pp. 26 – 28.

Department of Defense study (Defense Science Board 2012) that characterizing and analyzing autonomy as a property of a system, such as in "levels of autonomy," is "counter-productive because…[it] focuses too much attention on the computer rather than on the *collaboration* between the computer and its operator/supervisor….." To design the total work system, this reports we should design autonomy for collaboration, not for operating without human interaction.

Pilot/ATCO problems with TCAS highlight what happens when designers neglect the *distributed interactions* of people and automated systems. By analogy with human factors critiques about designers who neglect the system interfaces, we might call this "interaction negligence."

11.3 Summary of Authority Aspects of Überlingen Scenario

In the sequence of events in the actual Überlingen accident, there were there no acts of exerting authority inappropriately—for example, the Karlsruhe ATCO didn't violate the fundamental air traffic control rule of reaching outside his airspace to contact the DHL or BTC flights directly—but many failures to exert authority (acts of omission).

We have argued in this chapter that in fact TCAS and the Zurich ATCO did not have overlapping authority, but rather the Zurich ATCO did not properly carry out his responsibility to safely maintain aircraft separation, an act of omission. We suggested that philosophically the question is not "who is in charge of what I do?" but more pragmatically, "Which trajectory is more likely to be correct?" and that might reduce to "Which advice do I *trust*?" One might argue that the BTC crew followed the ATCO's advice because he spoke first and emotionally, and that was more persuasive.

The BFU Report (p. 76) argues as well that the Russians might have developed a more complex mental model about trajectories of other aircraft:

> The *Russian Federation* states that the Russian pilots were unable to obey the TCAS advisory to climb; the advisory was given when they were already at 35500 feet while the controller wrongly stated there was conflicting traffic above them at 36000 feet. Also, the controller gave the wrong position of the DHL plane (2 o'clock instead of the actual 10 o'clock)... (TCAS 2012)

Perhaps the BTC crew rationally related the TCAS and the ATCO information, and they chose the most likely route to safety. Specifically, they might have concluded that TCAS was telling them to climb into traffic because ATCO had implied that an aircraft was above them. The combination of this information and ATCO's repetition of the command (which TCAS did not do), made the "descend" advice appear more believable.

Considering now the failures to exert authority, here is a summary of the factors we have discussed:

1. Zurich ATCO had authority as Supervisor (DL) during the night shift but did not behave accordingly; he did not exercise his authority by reflecting on the larger situation and realizing that the staffing was inadequate (BFU Report, p. 83).
2. The Zurich supervisor did not exert his authority appropriately to brief ATCOs about risks involved in the technical work, i.e., reviewing what might go wrong and what to do about it, such telling them, "Do not go on break until the work is done." (p. 88)
3. System Manager technicians (SYMAs) were trained and aware of the system degradation, but did not exert their implicit authority to remind and/or educate the ATCO, "Because the technical work had been planned long in advance they assumed they were not responsible and did not see any need either." (p. 89)
4. The two CIR CAs present at Zurich were not expected to reflect on the CIR situation and would not normally question the ATCO's authority to handle air traffic problems (p. 86).

The number of omissions raises the question of who is responsible for safety and avoiding errors. In practice at the Zurich ATCC at the time, the answer was "only the person on the sharp end" (p. 83, 87). For example, the BFU Report says "the CIR system design helps defend the controller from making errors at the 'sharp end', but also monitors for these errors, so they can be managed." (p. 87) That may be so, but ultimately some people must feel responsibility and adaptively speak and act for the system by being sensitive to and responding to unusual situations. For example, the SYMA might have taken it upon himself, seeing that the supervisor and an ATCO were absent and the ATCO moving a bit frantically back and forth between workstations, to verify that systems were functioning that were expected to be operational, such as the backup phones.

We draw a broader systemic conclusion not addressed by the BFU Report: narrow concepts of hierarchical job boundaries and responsibility are inadequate in a complex work system. People must be proactive to use their knowledge to help each other. Further it must be accepted that false positives will occur and people should not be penalized for speaking up (compare with Columbia Accident Investigation Board [2003] report about communications within the Johnson Space Center Mission Operations Directorate during Columbia mission). Somehow this must be balanced against the risks of intervening in time-pressured emergency situations (e.g., the Karlsruhe ATCO did not know what might be occurring on the aircraft or at the Zurich ATCC, reinforcing his understanding that intervention was inappropriate).

From a logical perspective, we may think of authority as a property of an agent: one agent may clearly have authority; there may be a clear transfer or change of authority; it may be unclear which agent has authority; or the answer to the question of authority may need to be refined in terms of some kind of type-structure, different agents having different types of authority. A major advantage of

a work practice analysis as we have performed for the Überlingen work system is that we may find our theoretical frameworks—however logically and dimensionally complete they may be—do not fit the world we are modeling.

To recapitulate what we have said in this chapter, Überlingen exemplifies that the "mantle of authority" notion, as something that can be transferred, doesn't necessarily fit real situations. Given that the pilot always has ultimate responsibility for safety, the issue of authority of ATCO versus TCAS is moot. The pilot has authority to do what is necessary: "TCAS does not alter or diminish the pilot's basic authority and responsibility to ensure safe flight" (FAA 2011, p. 39).

TCAS, like ATCO, is a source of information. Neither ultimately can be said to be in control of pilots' actions. The pilots control the aircraft.

"Authority" with respect to the TCAS might be better interpreted as meaning "providing the official source of information or advice," what is colloquially called "speaking with authority." But then the issue is not "transfer of authority" (as in transfer of control), but how the pilots weigh and interpret the information/advice.

12 Discussion: Verification and Validation of a Work Practice Simulation

In general, *verification* is a process of checking that a product, service, or designed system (e.g., a computer program) satisfies design requirements. In particular, verifying Brahms-GUM includes comparing work simulation behaviors to regulations, as well as evaluating the regulations with respect to yet more abstract specifications of function and value (e.g., safety, efficiency). In the context of work practice simulation, *validation* is a process of checking that the model correctly describes what occurs in an actual work system, that is, how people and system behave and interact.

Verification of Brahms-GÜM lies outside the scope of the effort reported here. Rather we take the first step of elucidating what it means to verify a work practice simulation. We provide here a framework for verifying Brahms-GÜM, or more generally, for verifying any work practice simulation in a framework similar to Brahms that models both *designed systems* (e.g., TCAS, radar display) and *people* (including specifically perception, conception of activities and methods, reasoning, movements, and communications).

By intent, a Brahms work practice simulation is a simulation of a *design*. Insofar as it includes models of how people behave it is a *scientific model*. And of course, the model's constructs, in combination with the simulation engine, constitute a *software program*. These three perspectives—a design, a scientific model, a program—can be used to define what aspects need to verified and the relevant methods.

In this chapter we first examine the relation of a software system to a design simulation and a scientific model. Then as important background for the discussion of verification, we clarify how regulations are represented in a work practice simulation. We next recapitulate the "total systems perspective," to highlight the presence of non-deterministic aspects in the model. Finally, we detail the relationships among work system requirements, design, model, and simulation behaviors (outcomes) and how these constrain and challenge verification.

In practice, validating Brahms-GÜM is a scientific process of finding evidence for the agent and object models, which is effectively a scientific activity. Validating object/subsystem simulations of systems with existing validating models, such as the TCAS, might be equated with verifying the Brahms-GÜM model of the subsystem with existing formal specifications. Validating agent thoughtframes could adapt methods applied in cognitive modeling (e.g., Kintsch, Miller, and Polson 1984). Validating agent workframes (e.g., of pilots and ATCO) entails evaluating whether the model properly describes how people behave. Such evidence begins by validating patterns of routine behavior, namely relating the conditionality, priority, and timing of workframes to methods people employ in the context of their activities. Collecting relevant evidence requires observing behavior and work system events over extended periods using ethnographic methods.

Accordingly, in this chapter we review the nature and importance of ethnography for creating and validating a work practice simulation. We follow this review by a case study of a shortcoming in the Brahms-MER simulation, which reveals the kinds of omissions and modeling errors that may make a difference in work system design, and hence reveal what validation of Brahms-GÜM might focus upon. Finally, we observe how because TCAS is fallible, certifying it requires a work practice simulation that models how people integrate its advice with other sources of information.

12.1 Comparing Software Programs, Design Simulations, and Scientific Models

The strategic approach of the broader research project that includes Brahms-GÜM is to apply model checking methods developed for software programing to verifying a Brahms simulation. Therefore, it is worth considering how a simulation of a work system *design* relates to a program, and then how a *simulation of a work system design*, as a model of an existing or proposed real world system, relates to a scientific model. These perspectives could help direct research for developing tools that facilitate creating and evaluating Brahms models.

On the one hand, a software program is a *system* with fixed, pre-defined inputs/outputs, and such a system can be modeled formally. In software engineering model checking involves creating a model of a system (the program), while Brahms-GÜM is by its very nature already a *model of a system*, the real world work system of people-systems-environment interactions. However, the Brahms model lacks formal semantics, and its behavior is determined by the Brahms simulation engine, a software program, whose correctness has not been formally established.

Furthermore, in the broader intent of this research, a Brahms ATS model would in general be based on a work system *design*, a possible work system configuration, formulated perhaps in documents describing equipment specifications and operations procedures. Because Brahms simulations can reveal emergent properties of a work system design, model checking methods applied to such Brahms models would be *checking emergent properties of a system design*, such as whether regulated system or human behaviors are violated. From the perspective of work system design, we would therefore locate model checking within the iterative process of transforming ATS requirements (including regulations, procedures, and protocols) to prototype work system designs, defining scenarios, and interpreting/analyzing behaviors, then working back to revising requirements and/or the design. This process is detailed in Section 12.5.

A Brahms simulation model may incorporate a software program used in the work system being modeled (e.g., TCAS). In this case, the work practice simulation in which the software program (or its simulation) is embedded can be used to verify that the program's design is consistent and complete with respect to the system as a whole, especially work practices. In particular, Brahms-GÜM shifts from certifying

TCAS only in terms of flight configurations, to include its functional role of affecting human behavior.

Finally, Brahms models are inherently simulations of cognition and action in people; hence model checking methods applied to Brahms models are *checking emergent properties of scientific models of socio-psychological processes.* Accordingly, it is important to understand and formalize how key events have probabilistic ordering and timing, and thus causally affect each other. What is the space of possible outcomes and what are the bounds of the domain under which the model is valid? Appendix 27 lists the probabilistic components in the Brahms language.

From a scientific perspective, particularly for cognitive scientists, the *mental models* people have about automation relative to their ongoing activities is especially important in simulating work practices. Of particular interest is how people monitor and understand system behavior within the requirements of an operational role, which determines what model of their environment is useful, how it is maintained, and what they should be doing. Brahms provides an excellent framework for modeling interactions of people playing different roles with different expertise in different settings. For example, in simulating the world of the pilot in the plane and the ATCO in the ATC, Brahms-GÜM simulation runs reveal how *multiple practices are coordinated in joint activity* (or not coordinated in the case of the Überlingen accident).

Brahms simulations by virtue of relating the processes in the environment to human activity show how perception occurs within activity and is thus affected by temporal and spatial relations. In particular, ATCO is engaged in "constraint-based problem solving"—an ongoing process of actively monitoring, interpreting, prioritizing, posing what-if questions, adjusting controls to stay within operational bounds, etc. With respect to mental models, model checking might emphasize how belief change occurs in Brahms simulations and how for example the work load or lack of communication affects situation awareness (e.g., failure to know the state of automation, such as that optical STCA is not operational).

In summary, Brahms simulations have a dual nature with both *engineering value* (as models of work system designs) and *scientific value* (as simulations of human-system interactions). For the purpose of ATS and the broader NextGen research program we would emphasize the engineering value: the model is the means for solving a problem, not the end in itself (in the sense that scientific models are products of a research project). In this respect, developing Brahms-GÜM might be better characterized as "doing analysis" rather than "modeling." We are using Brahms to analyze patterns of interaction among human and automated systems—patterns we can identify in today's automation (e.g., TCAS) and that will likely be present in the work systems of tomorrow.

In developing models like Brahms-GÜM we would over time create not only a library of components (e.g., aircraft, radio, radar display), but also a collection of

patterns of system configurations, extensible mini-scenarios based on real incidents and accidents (such as "missing alert system" or "authority usurped from agent without notification"), and that are applicable to the analysis of proposed future air transportation systems.

For those creating models like Brahms-GÜM the analytic aspect is always salient, an effect common to modeling human behavior in general. We believe we can correctly understand small snippets of behavior. We can approximate or simulate these in a variety of ways. But when the time-scale gets longer, the number of interacting agents grows, the choice-points proliferate; then we need to use the computer as a bookkeeping tool to work out the implications of our simple hypotheses as they are composed into a larger system. In part the problem is tracking or anticipating the interactions that might occur given the preponderance of systems having multiple states. The difficulty also relates to emergent relations in the larger system that we can't either imagine easily or simply diagram on paper (e.g., how the late arriving AEF affected Zurich ATCO's awareness of other flights, by virtue of the relation of radio frequencies to the location of radar displays and phones).

In short, creating Brahms-GÜM could be viewed as fundamentally an analytic method for understanding what happened at Überlingen, as we believe this report demonstrates. The ability to grasp complexity and insightfully prioritize design problems is the purpose of the analysis. For example, one could argue from the Brahms-GÜM simulation outcomes that it is extremely important that ATCO have a means of knowing that STCA optical is not working (effectively an alert that an alerting system has failed or is missing).

In developing a "what if" simulation, one can make predictions that certain work system configurations are risky, leading to unsafe situations. Such predictions are of value for the design process, as an engineering tool. The notion of accurate prediction is also not apt because it is unlikely that any initial model state (including the modeled knowledge and behaviors of people) could exactly fit future conditions that would occur in the real world. Predictions based on work practice models (interpretations of a Brahms simulation outcome) must always be characterized as "in circumstances like these…interactions like these might occur." When the simulation shows aircraft colliding or even repeated separation violations, then we know we have a problem worth investigating and can turn our attention to verifying and validating the model, and then trying alternative designs. Thus, the primary aim of the project is not accurate prediction of any particular work system configuration, but identifying general designs that merit improvement or further analysis.

12.2 Representing Regulations in a Work Practice Model

Perhaps one of the most obvious evaluative questions we can ask about a work system is whether it follows all established regulations and procedures. Consequently, it is important to understand how regulations and procedures are modeled in a work system simulation. A fundamental point about a model of work practice is that unless people in practice actually read regulations and procedures or

talk about them, they need not be explicit representations in the model. Most often, regulations and procedures will be implicit in the simulated behaviors—an observer (or monitoring process) must abstract what occurs in the simulation to verify that regulations and procedures are followed.

If a simulation works by have agents read (refer to) a representation of regulations and procedures to determine how to act—and thus the representations control the agent's behavior—the simulation is by definition not a work practice simulation, because that is not how people behave in the real world. Indeed, everyone knows that referring to a manual, written policies, etc. is not practical, and anyone doing so at every step would be unable to do their work. Within the Brahms framework, people are modeled as being mostly reactive, following well-practiced and accepted patterns of behavior, rather than "reasoning from first principles" (i.e., by which thoughtframes would deduce optimal behaviors in terms of their causal effects).

However, we can design computer programs that do operate "mechanically" (or compile their behaviors from such representations), so it is important to recognize how a work practice simulation is different and the implications for verification. We begin with a general review of how different mechanisms that may produce it what an observer may abstract as behavior that follows a pattern.

The behavior of a computer program may be regular by virtue of having an explicit set of rules, grammar, a procedure, and so on that generates the program's behavior or the interacting processes may implicitly produce what an observer describes as patterns. This distinction is well known in natural language—the vast majority of people speak "grammatically" without necessarily knowing or deliberately applying explicit rules of grammar. (Indeed, explicit grammatical rules would not produce anything close to what people actually say, hence Chomsky's distinction between competence and performance.)

In developing a computer program to simulate human behavior a modeler may adopt a grammar-based method or a more direct behavior-based method. In particular, in simulating ATCO for example, we might have formulated a library of procedures and protocols that ATCO must follow, and the simulated ATCO agent would generate every individual action by interpreting this library.

The Brahms framework allows for representing written manuals, online procedures, written "cheat sheets," and so on, but such constructs are included only insofar as we know people actually look at, read, and follow them as references. Indeed, we might have modeled pilot behavior in more detail by specifying how they do in fact refer to written checklists before takeoff, etc.

Simulating practice means representing what people actually do, emphasizing where the person looks, what is perceived (taken to be information), general movements, and when reasoning actually occurs. In general, if you examine a Brahms model of work practice you will not find a body of regulations or

procedures packaged and labeled as such. But you will find on examining the simulation behaviors that pilots and ATCOs appear to follow procedures and adhere to regulations.

As an example, consider a regulation such as, "When instructed by the ATCO, a pilot should change the radio frequency." This statement does not appear in Brahms-GÜM. However, the Pilot Group includes this top-level workframe:

```
workframe Tune_Radio {
    priority: 60;
    variables:
        forone(AircraftRadio) radio;
        forone(Flight) flight;
        foreach(double) freq;
        foreach(string) out;
    when(knownval(current.location = radio.location) and
        knownval(flight = current.flight) and
        knownval(flight.handoff = true) and          // pilot's flight is in handoff process
        knownval(radio.frequency != flight.sectorFrequency) and
        knownval(freq = flight.sectorFrequency))
    do {
        conclude((radio.frequency = freq));          // new belief about radio's frequency
        println_d("tuning radio to %1", freq, out);
        tuneRadio(out, radio);                       // change radio frequency (a fact)
        printlnWithSimTime(out);
        getCommTime();                               // call ATCC on new frequency
        conclude((current.commPerformative = "INFORM"));
        conclude((current.commReason = "flight"));
    }
}//wf Tune_Radio
```

The workframe describes a situated action: the pilot is located by the radio, which is not tuned to the next sector's frequency, and the flight is in handoff process. He tunes the radio (modeled as communicating with the radio) and then concludes that he must inform the ATCC about his flight arriving in the sector.

In summary, what might be expressed as a prescriptive rule for the pilot to change the radio frequency is expressed procedurally (e.g., the pilot changes the radio frequency after receiving the information, then contacts the ATCC). This "rule" is also modeled on multiple levels of abstraction (e.g., the communications are modeled in the Radio Communicator Group to which the Pilot Group belongs) and within the Brahms framework that provides methods for modeling interactions among objects and agents (e.g., tuning the radio frequency involves "communicating" with it).

As this example illustrates, regulations and procedures are not generally found as statements in the model, but rather the modeled behaviors (practices) embody and

respect such requirements. Indeed, one might draw an analogy of the relation between software requirements and the code itself as being like the relation between regulations/procedures and a model of work practice. In this respect, one could use a body of formalized regulations/procedures to verify that a Brahms simulation "fits specifications." For example, comparing Brahms-GÜM simulations to ATCC regulations, one would detect a violation when only one ATCO is on duty.

Presenting this another way, looking for regulations in a model of work practice is a category error. As this is described by Gilbert Ryle (1949), after a tour of a campus one might say, "I see the student union, the library, and all the classrooms, but where is the university?" Or we might say, "I see the teachers and the students and all the classes, but where are the principles of higher learning?"

Because of the emergent (unanticipated and difficult to predict) interactions that may occur, it is particularly important for an ATS work system simulation to be verified for adherence to safety properties, such as the separation between aircraft. As we have explained, the simulated ATCO does follow certain rules about when to handle a potential collision (Section 9.2), but circumstances involving a particular sequence of events and priorities of other activities may delay or prevent ATCO's actions. A separation violation occurred at Überlingen because of the failure to monitor the larger airspace while focusing on the AEF handoff; the same result occurs in some of the simulation runs of the Brahms-GÜM scenario.

Insofar judges in court disagree about the meaning of regulations, and different observers disagreed about to what extent ATCO or the BTC pilots were at fault at Überlingen, a work practice simulation can only be verified relative to an observer's interpretation of what the regulations mean. In a model checking process, this interpretation would correspond to a formal representation of the "semantics" of the regulation. That is, a model of the regulations will be checked against a model of the work system. As for the remarks made about accuracy of prediction, what is at issue in verification is not so much whether the work system design is "correct" (i.e., objectively, from any point of view), but whether problems can be detected and to characterize perhaps the space of scenarios in which the design is reliably safe.

In summary, it should now be apparent why regulations/procedures are generally not represented explicitly in a Brahms model. Regulations/procedures are *normative descriptions* of interactions among people, systems, and environment, that is, *abstractions* of what is supposed to happen. Therefore, they would not in general be contained in the model. Like the relation between specifications and a program, regulations/procedures are at a different level from the work system. The proper way to relate regulations and a work practice simulation is to use models of regulations outside the simulation to evaluate the system's behavior in different scenarios.

12.3 The Importance of Verifying the Total Work System

Recall that the technical approach being used in the broader research project is adapting an existing agent-based modeling system (Brahms) and using sophisticated software modeling tools to provide useful analyses early in the work system design process (Chapter 1). The analysis of work systems composed of people and automated subsystems whose interactions may become complex requires methods for verifying that safety properties are not violated.

As illustrated by the Überlingen accident, verification of automated systems should take into account interactions with other systems, people, and the environment. Understanding how the work system behaves entails considering communications, protocols, regulations and procedures, organizational roles and policies, geographic locations and facility/vehicle layout, controls/displays, automated controls and alerting systems, weather, and so on. The people and this context of resources and operational constraints constitute the "total work system," also referred to a *socio-technical system*.

The verification of TCAS (Kochenderfer et al. 2012a,b) has not heretofore been placed in a total system perspective, and instead considers mainly the mathematics of aircraft in flight. The analysis of Überlingen by Kuchar and Drumm (2007) states the pilots did not obey TCAS, without considering the ordering and timing of ATCO's intervention: "The Russians' choice to maneuver opposite to the RA defeated the coordination logic in TCAS. An advisory system like TCAS cannot prevent an accident if the pilots don't follow the system's advice" (p. 284). Viewed in terms of the automated system alone, they then ask, "Why didn't TCAS reverse the sense of the RAs when the situation continued to degrade?" rather than "Why didn't ATCO know that TCAS intervention was underway and his authority was usurped?" or "How difficult is it for pilots to reverse course under direction of a computer seconds after a person of authority has fervently told them what to do?"

A component like TCAS cannot be properly designed or validated in isolation: One must remain committed to the integrity of the work system, which includes the capabilities and socio-cognitive strategies of the people and the causal effects of their sequential and simultaneous interactions with technology. As an example, de Carvalho et al. (2009) provide a systemic analysis of how, even when all component systems are functioning normally, an accident can occur. Verifying and validating in terms of possible failures and errors alone is not sufficient—only a total system perspective will reveal "coincidences, unexpected links, and resonance" (p. 339) by which the operating conditions drift into unsafe states (p. 326).

Kuchar and Drumm (2007) acknowledge and provide data about pilot response (p. 287) and incorporate some aspect of pilot response in their model (p. 290), but the emphasis is on improving TCAS rather than improving the overall concept of operations from the pilot's perspective. A system with limited situation awareness may be made yet more complicated in order that it will know when to reverse its

previous command. As for the pilots, "this specific problem [what occurred at Überlingen] is being solved by improving pilot training to comply with RAs" (p. 294). But the same analysis concludes: "Pilot non-compliance to an RA may not necessarily compromise safety in a particular encounter," such as when visual-separation procedures are permitted (p. 287). The inherent contradictions in the system remain.

Verifying TCAS requires viewing its function from perspective of the total work system, this includes convincing pilots to manipulate the aircraft controls to effect a change of course. So even before the Überlingen accident, it would have been obvious from a total system perspective that the function of the TCAS system has two aspects, operating in different domains (Figure 12-1): 1) detecting planes and giving alerts with safe advice (the logical-functional domain), and 2) convincing the pilots to control the aircraft in a certain way (socio-technical domain). But this leaves out the ATCO, whose actions may change the context in which a pilot interprets TCAS. Consequently, the total system must include at least the ATCO responsible for the aircraft and his/her operational context. (See the discussion in Sections 11.2 and 11.3 concerning why we describe TCAS's role as inherently *convincing* pilots rather than ordering them to change course.)

Traditional human factors analyses of TCAS might focus on the display design, the loudness of the alert, the choice of voice, the phrasing of the instruction, and so on. These are relevant and important considerations. However, socio-technical systems analysis starts with the total system—what are people trying to accomplish? What are the functions of tools relative to the stakeholders' objectives and values? Who or what interacts in space and time to produce the behavior of the overall system? Specifically, what information about the air space might enable pilots to judge the right course of action so they might better weigh the advice from TCAS?

Similar observations have been made by Rushby (2011) that certification should be based on an "*argument* that certain *claims* about safety are justified by *evidence* about the system…. The argument must consider all possible circumstances of the system's operation, including those where faults afflict its own components, or its environment behaves in undesirable ways" (p. 211). Rushby explains that the compositional approach tends to be favored by computer scientists, but is not appropriate for flight certification:

> Computer scientists might wish for a more compositional (i.e., component-based) approach, but this is antithetical to current certification practices. Experience teaches that many hazardous situations arise through unanticipated interactions, often precipitated by faults, among supposedly separate systems. (p. 216)

All of this is another way of justifying the use of a work practice simulation—it is of value not only as a design tool for grasping complicated interactions in which many variables (including capabilities of automated systems and roles of people and

systems), but also as a means for verifying a work system design, exactly because it includes the relevant people and systems.

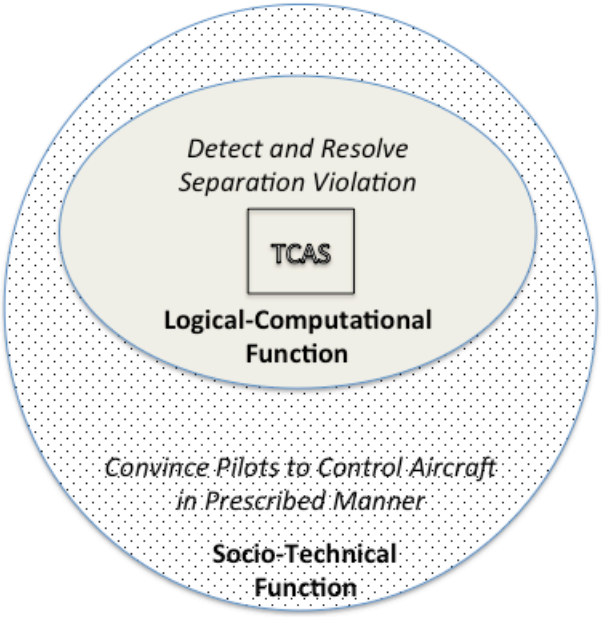


Figure 12-1: Total System perspective for verifying TCAS functionality

12.4 Relating Requirements, Design, Model, and Simulation Outcomes

Running simulation scenarios provides a way of verifying a design, such as by indicating gaps and unacceptable situations. However, this is a conceptually complex process, because the simulation has several aspects including the work system design and rationale and how these are represented, the model of the design, the program that runs the model (simulation engine), and the simulation outcomes (the behaviors of agents and objects).

Table 12-1 provides one way to grasp this complexity by analogy with software verification. In software engineering, formalized requirements guide and constrain the program's design and behaviors; verification by one approach involves creating a model of the program and relating it to requirements.

Table 12-1: Relation of program to work system simulation model

	Requirements (Success Criteria/ Constraints)	Design	Implementation/Code
Computer Program	Technical Requirements	Technical Design	Program
Work Practice Simulation	Goals and Regulations	Work system design (WSD)	Simulation model

In the context of work practice simulation, the system to be verified is already a model. In some respects the specifications, the design is represented in the model explicitly (e.g., groups, agents, geography, tools); in other respects the design is represented implicitly (e.g., regulations, as discussed in Section 12.2).

Table 12-2 elaborates the scientific perspective introduced in Section 12.1: the "system" is some part of the air transportation system in the world; the model to be verified comprises the model components, scenarios, and outcomes; and the specifications are the mix of ATS goals, policies, regulations, etc. that the system's states and behaviors are supposed to obey.

Table 12-2: Relation of System, Model, and Specification in Brahms-GÜM

Conceptual Level	Brahms-GÜM Aspect
SYSTEM	ATS work system
MODEL OF SYSTEM	ATS work system design formulated as Brahms model (static definitions of properties and behaviors of objects and agents and geographic/facility models)
	ATS scenarios (initial configurations of model components)
	Work system behaviors (simulated states/actions)
SPECIFICATION	Regulations, protocols, related ATS goals (e.g., safety requirements; critical states/thresholds)

The value of verifying the simulation model will depend on its validity. A model that omits key aspects of the real world system or improperly represents how people or objects behave may be verified with respect to ATS regulations and operational procedures, but be insufficient for certifying the design. Therefore using a work system simulation as a design tool requires at least the following distinctions:

- Verifying and validating the *simulation model* (including "model checking")
- Verifying the *work system design* (WSD) with respect to regulations, procedures, etc. by simulating the design to evaluate the system's performance (e.g., does this design satisfy safety properties)
- Verifying a *proposed or revised regulation*[54] in a simulation model driven by reality-based scenarios (e.g., actual input events and/or load statistics)—will a regulation be satisfied in practice if we design air traffic systems in certain ways (class of WSDs).

The regulations, work system design, and work system simulation have a triadic relation (Figure 12-2). All three are in some sense models: regulations prescribe properties of the work system; the design describes components and behaviors intended to satisfy the regulations; and the simulation model represents the design.

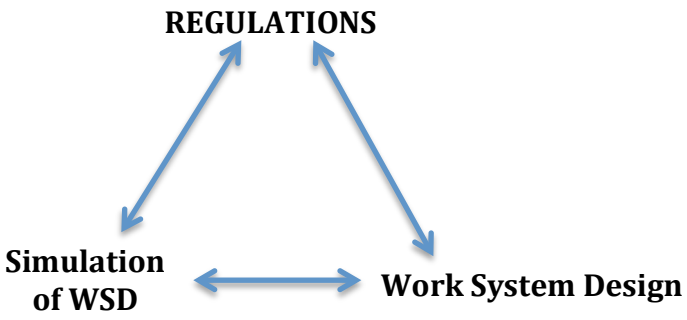


Figure 12-2: Triad of abstractions: Simulation Model, Work System Design, and Regulations

The triad shows possible flexibility in what is taken as given and what is being verified. Insofar as the work system includes new or revised components such as new automation, some of the regulations will be proposals about how these components are to interact with others. One might view the future work system as given and view the simulation as a means of determining whether the regulations are consistent and sufficient. The regulations themselves are being designed with respect to more abstract goals, which in ATS involve safety and efficiency:

Efficiency & Safety Goals ⇔ Regulations ⇔ Work System Design(s)

Consequently, in using a work system simulation for "verification" early in an ATS work system design process we might focus on different aspects:

1. **Check a WSD:** Given a candidate work system design, *does this WSD satisfy the regulations?*

[54] Hereafter in this chapter the term "regulations" refers to all formal and semiformal requirements imposed by organizations about work products and how the work is to be done such as policies, procedures, protocols, etc. that affect staffing, facilities, tools, behaviors of people and systems, etc.

2. **Check a regulation:** Does this regulation produce the desired system performance, with respect to efficiency & safety?
3. **Redesign a work system:** Can we "debug" a WSD to fix it relative to the regulations?
4. **Generalize WSD:** What range of WSDs *satisfy a regulation?* i.e., what variations are allowable?
5. **Revise regulations:** Can we modify a given regulation so it is satisfied by a class of WSDs? (e.g., relax a requirement)

In practice the work system design and regulations might be formulated together. We might then think of the requirements for the work system as consisting of *performance constraints* on work system behavior (e.g., aircraft separation distances) and *operations constraints* on work system design (e.g., requiring at least two ATCOs on duty), corresponding respectively to what we have called here the regulations and design.

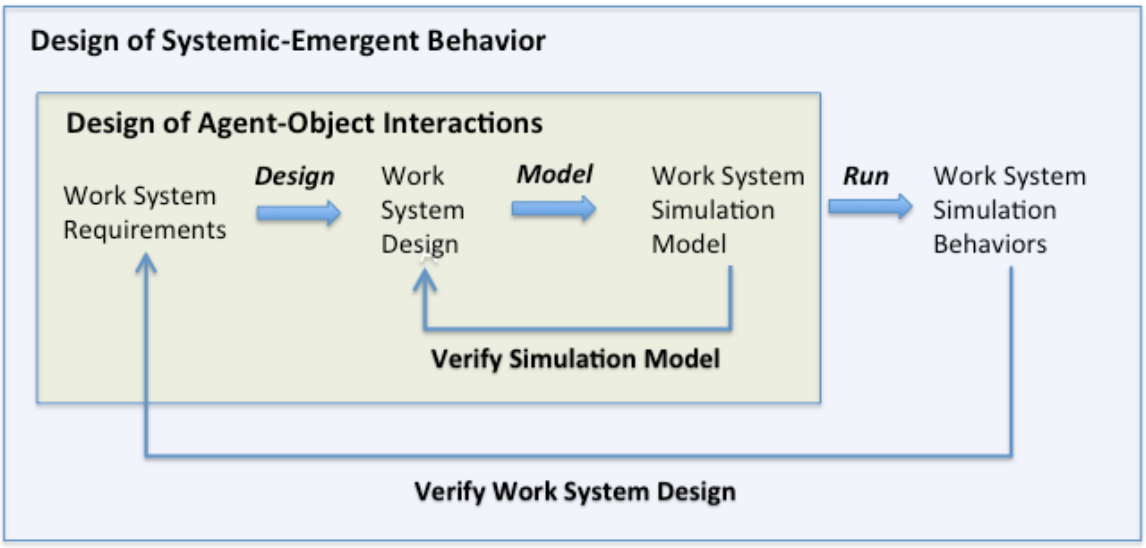


Figure 12-3: Double-Loop Investigation: Behaviors of verified work simulation model provide evidence for verification of work system design

In summary, we are interested in understanding how a WSD will behave, how it relates to requirements, and finally whether a proposed change to the regulations will function as desired. Because the work system design and requirements both serve to define the simulation, "verifying a simulation model" involves at least two levels of analysis. Figure 12-3 shows how these two aspects of the work system design process can be related from a verification perspective in what has been called "double-loop learning" (Argyris and Schön 1978).

Referring to the inner feedback loop, we can investigate whether the simulation model implements the work system design. And then given the simulation behaviors, we can investigate whether the work system's behaviors satisfy the

requirements (e.g., regulations, cost, efficiency, and other practical human-centered concerns) and hence the design is satisfactory. Put another way, behaviors of a verified work simulation model provide evidence for verifying the work system design.

A strict analogy to software engineering verification will be misleading because the requirements of a WSD are not necessarily fixed, but might need to be changed because they are impossible to satisfy for a given design, aspects of which may be fixed (e.g., legacy systems). This highlights why work practice simulation is useful for designing future work systems in which automation, people's practices, and even regulations might be as yet undetermined or modifiable. This is the valuable "double-loop" aspect of the project in which goal and values might be reframed or reordered.

12.5 Developing and Applying Work System Simulations Scientifically

So far we have considered verification from the perspective of software engineering that relates two formal systems (e.g., a set of requirements and a program or a model). Because it models work practices, a work system simulation model may also be viewed as being a scientific model of the activities, objects, beliefs etc. of that work system. In this section we consider in more detail how developing and using a work system simulation may be approached from a scientific perspective.

Work practice simulations may be considered as either descriptive or predictive models. We create and use them in ways analogous to scientific modeling in general, but the emphasis is on formulating and evaluating a work system design. Considerations for validating the simulation include both the model of current work practices as well as the more general theory of human behavior that is implicit in both agent models (e.g., how people act in certain situations) and in the framework/engine itself (e.g., how attention may be interrupted and resumed on the basis of felt priorities).

Practically speaking, the value of a work system simulation is not in predicting the future, because it is unlikely that a specific initial configuration (e.g., aircraft locations and flight paths) will occur or that the set of simulated, interacting agents would match the specific knowledge and habits of actual people. Instead, we view the work system simulation as a form of test bench, a virtual environment used to verify the completeness and consistency of the work system design and regulations (broadly defined).

The modeling process can be expanded to make clearer the iterations involved and role of analysts in evaluating the simulation results (Figure 12-4):

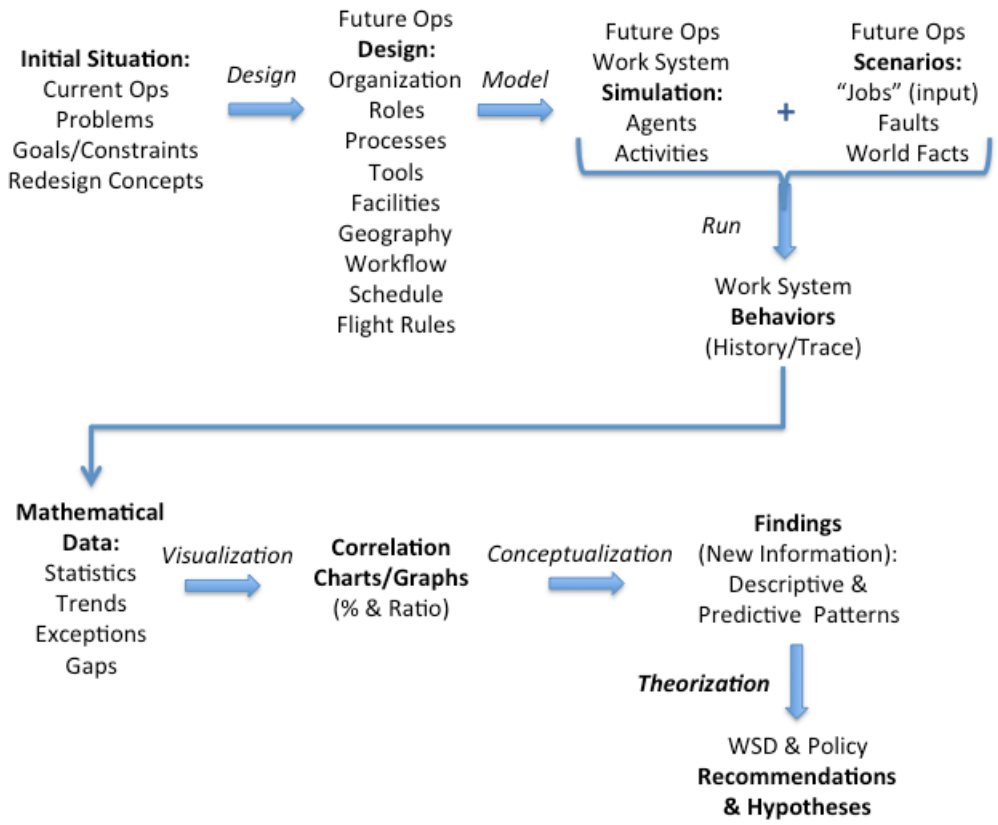


Figure 12-4: Workflow in creating and analyzing work system simulations as a scientific process.

1. **Design:** Formulating design of future operations, often by reference to current or historical operations
2. **Model:** Representing people, facilities, tools, operations, etc. as a work system simulation
3. **Simulate:** Running the simulation against scenarios (i.e., model configurations, including absent or "faulty" subsystems, as well as "work" or "jobs" that flow through the system, e.g., flights in the ATS, customer orders in an office setting)
4. **Visualize:** Creating visual representations for reflecting on patterns in work system behaviors over time (logged in a file as a "history" or "trace" of time-stamped events)
5. **Conceptualize:** Describing patterns (e.g., trends) of interest, indicating surprises, evidence for and against hypotheses, etc.
6. **Theorize:** Articulating implications for future operations, including revised design and regulations.

This workflow is based on our experience with the OCA mirroring simulation (Figure 12-5; Clancey et al. 2008), but incorporates general methods of experimental data analysis (for another example, see Clancey and Lowry 2012). The

process of creating charts and conceiving new relations ("findings") is of course illustrated by the refinement of Brahms-GÜM described in Chapter 10.

As the charts from the Brahms-OCAMS simulation illustrate (Figure 12-5), multiple simulations and/or scenarios may be compared quantitatively in the analytic phase. Here metrics generated from a simulation model of future operations are compared to the metrics—running over the same scenarios—produced by the current operations model. The future OCA operations model was created by extracting part of an agent model of a Mission Control Center backroom flight officer and identifying those same activities as being executed by a software agent.[55] If funding and time had permitted, we would have validated the current operations metrics by observational data from Johnson Space Center Mission Control, lending credence to the predicted metrics generated by the future operations simulation.

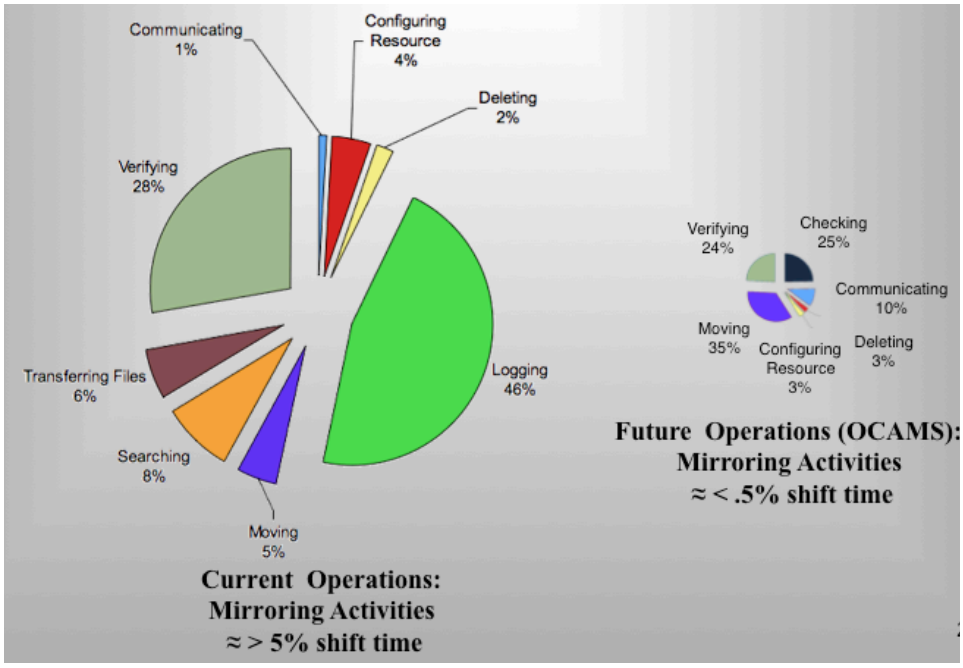


Figure 12-5: Visualization of statistics generated from "current ops" simulation compared to "future ops" simulation, showing the activities and percentage time devoted to the "mirroring" task. Automation would reduce the effort required for the "mirroring" task from 5% to .5% of total shift time (8 hours).

This part of the analysis must be performed by human analysts. Analytic tools such as Excel are useful for collecting and graphing simulation statistics. In the case of the OCAMS example, the "conceptualization phase" noted in Figure 12-4 appraised the

[55] Workframes were edited to allow for the task being automated. Thus the software agent in the future operations simulation was a prototype of the agent that was later incorporated in the deployed OCAMS system. This illustrates the *simulation-to-implementation methodology* (Clancey et al. 2008).

difference between the current and future operations simulations. People must be involved because interpretations of the simulation outcomes involve value judgments, such as how large a quantitative improvement is sufficient to justify implementation of the future operations design. This appraisal requires background knowledge about the organizational objectives and values (e.g., what kinds of investments are acceptable?). A business-economic analysis is usually required, such as a payback analysis (time to recoup costs) with a cost/benefit tradeoff (e.g., effect of denying other investment opportunities). Such analysis may justify improving the simulation in certain areas, such as increasing fidelity, in order to increase confidence in the data that it generates.

In summary, verifying and validating a work system simulation is not an entirely automatic or automatable process. Rather, people must play a role in determining whether a work system simulation output is valid and useful. This appraisal operates on several levels including verification of the model (inner feedback loop, Figure 12-3), reformulating regulations (including policies, procedures, etc.; the outer feedback loop), and also revisiting the effectiveness of the simulation model, including the kinds of metrics, visualization, and analysis with respect to the purpose of the modeling enterprise (e.g., increasing acceptance of proposed automation).

The first phase of modeling and analysis (top of Figure 12-4) may involve formal verification methods. The second phase can be aided by analytic software, but people will do the work of reflecting on the simulation results and generating further experiments. In the broader research project of which Brahms-GÜM is a part, examines how formal methods can be used to facilitate this analytic process as well.

The entire work system design modeling and simulation process is a form of inquiry, which means that most aspects are tentative and subject to revisions:

- Proposed requirements (regulations)
- Proposed work system design
- Scenarios of work system configurations (what might occur)
- Predicted work system behaviors
- Causal analysis of work system and regulation incompatibilities
- Recommendations for redesign and/or regulation revision

The investigation is experimental, involving modeling with repeated reinterpretation of observations and documents, variation and analysis of alternative models, and experimentation with alternative designs and regulations. As illustrated by the Brahms-GÜM effort (Section 6.8, Chapters 8, 9, 10), the modeling process involves discovering new events and relations of importance, realizing aspects of the model that require more detail, and throughout reframing how the model can be formulated in a useful way, where even the potential value of

the model will be affected by the kind of simulation that is practical and the results that emerge.

To this point we have emphasized the steps involved in analyzing simulation behaviors from a given simulation run. The other side of this process is the well-known problem of managing—generating and exploring—a large space of simulation runs. With respect to a given Brahms model, this space is defined by the initial model configurations (scenarios,) and the probabilistic elements in the model (Appendix 27). As can be expected, there are too many scenarios to run as dozens of variables are introduced. Model checking methods adopted from software engineering (e.g., Hunter et al. in preparation), could provide a means of systematically exploring this space. Accordingly the Brahms engine was modified to enable logging "branch points" (events affected by a probabilistic element) and tracking whether a belief or fact was created from an initial belief/fact (i.e., part of the scenario definition). Further consideration of this approach lies outside the scope of the presently defined Brahms-GÜM project and report.

Because a work practice model represents both "normal" and non-optimal behaviors, it is important that the modelers be aware of what variations are possible within regulations and what variations might occur because of workload, fatigue, or complex situations. As we have shown, in some respects important non-optimal behaviors (e.g., ATCO not monitoring the overall airspace) will emerge in Brahms simulations from the workframe priorities and circumstantial interactions. Other undesirable behaviors may occur in practice that the model fails to capture because of assumptions about agent capabilities (e.g., duration of an activity) or methods (e.g., how ATCO handles a dysfunctional phone system). In general, we are less concerned about how people might be more creative and capability in reality, than where the model makes socio-psychological claims that are not always true.

One way to anticipate such shortcomings, to help us critically analyze a model, is to refer to how simulations have failed in the past, such as in the example of the Crater simulation program that was misused for predicting damage to the Columbia shuttle tiles:

> Crater was validated using small pieces of foam and ice on single tiles. During the process of turning empirical data into a predictive equation, the limitations and contingencies of these initial data sets were lost. Furthermore, the process of computerization of Crater rendered the uncertainties inherent in the tool even more invisible, and the specific mode of computerization, a plug-in-the-numbers spreadsheet, gave a false sense of clarity and certainty to the results." (Brown 2006, p. 396)

We need tools for tracking limitations and contingencies in a complex simulation like Brahms-GÜM. For example primitive activity durations need to be documented indicating how they were determined, why and when they were modified from analysis of simulation runs, and represent in some way under what circumstances the assumed upper and/or lower duration bounds might be different in practice. We need to determine critical variables (e.g., when ATCO monitors the larger airspace)

and what affects them (e.g., prioritization given to aircraft in landing approach), how and when these determining events might occur (e.g., a Sunday evening), what complications might make the work difficult (e.g., other ATCO off duty), what automation could mitigate these complications (e.g., STCA Optical alert), what other complications might disable or defeat the automation (e.g., maintenance), and so on. The modeling and simulation process enables us to get a handle on such causal relations, and particularly how their ordering might change and influence the overall system. Model checking might enable then discovering combinations that are not expected and have not yet been discovered through scenarios that have been deliberately tested.

Finally, in our consideration of verification and validation of Brahms-GÜM, we might step out from the language of requirements, design, and so on, and broadly view the modeling and verification process from the perspective of the person or agency that is sponsoring and/or benefiting from the simulation. In the broader concept of this research, the "customers" for Brahms-GÜM are the airlines ("vendors") who will be investing and using proposed NextGen technology. In this respect, the goals of the work system design, simulation, and analytic process are to increase safety in operations; while in the context of the invention of new technologies, the goals of the entire work system simulation enterprise is to provide useful analyses early in the design process, and thus to increase confidence among stakeholders so they will adopt standards that implement the NextGen plan:

NextGen Confidence/Adoption Goals ⇔ Work System Simulation

That is, the use of work system simulation could be adopted as part of an overall methodology for NextGen design, evaluation and adoption. Increased confidence could result from a combination of factors including formalization of work system design as a simulation model; verification of the regulations, design, and model through a combination of scenario experiments, model checking, and analysis of performance metrics. Put another way, confidence rests on *believing the simulation output* (involving verification and validation), which can be enhanced by participating in the model-building and evaluation process. Consequently, the tools in the model-building environment, such as the Brahms AgentViewer, could be essential for making the model accessible and the output understandable to pilots, ATCOs, air traffic control center managers, airline managers, and so on. Thus, in considering methods for verifying and validating a work system simulation, we need to take into the people and activities that will provide a context for creating, analyzing, and interpreting a simulation with respect to NextGen ATS.

In summary, so far in this chapter we have discussed a broad range of issues involved in "verifying" Brahms-GÜM. The simulation effort is one means of verifying a work system design, which may be accomplished by a combination of experimentation with scenarios and formal methods that relate the space of alternative simulated behaviors to regulations (e.g., safety properties). The Brahms language and engine itself implicitly embodies a socio-psychological theory of

human activity, including aspects of perception, reasoning, and physical interactions with objects in the world. The methodology for creating and using such a simulation might occur within the context of activities involving a variety of stakeholders (e.g., pilots, airlines, FAA) to facilitate NextGen technology development and certification, along with new regulations and work processes.

12.6 Use of Ethnography in Modeling

The scientific modeling perspective shows that verification and validation of a work practice simulation are both essential. In this section we discuss an empirical method for model development and validation using ethnography.

Ethnography is a method of observation and analysis of social groups and activities. Ethnographic methods were developed within the discipline of anthropology, but are now used in a variety of academic and applied fields, including sociology, social psychology, industrial design, user interface design, and human factors research. Ethnographic methods are combined and related to describe and analyze participants' activities at a variety of time scales, including the methods of participant observation; video, audio and still photography recording and analysis; self-diaries; informal on-the-spot interviews; formal oral histories; and collection of workplace artifacts, especially documents and drawings.

Ethnographic observation is a necessary part of understanding work practice, that is to learn and understand how work actually gets done. Ethnographic observation typically forms a part of the Brahms modeling method, including learning about the actors, their formal and informal relationships, activities, communications, documents, procedures, tools of the trade, the work setting, and so on. Ethnography has been applied previously in ethnographic contexts to "reveal patterns, raise questions, and enhance our understanding of systems in ways not easily available with other methods" (Mindell and Mirmalek 2011; Hutchins, 1995, 2000).

Validating some aspects of the Brahms-GÜM simulation model requires ethnographic observation (or some automated means of recording data). This includes questions such as when and for what period ATCOs monitor radar displays, at what distance ATCOs typically detect potential separation violations, and the effect of alerts such as the STCA Optical system on ATCO behavior. As illustrated throughout this report, documents alone are not sufficient for constructing a work practice model, particularly in matters related to aviation safety where it is well known that pilot and ATCO mishaps are not uniformly recorded.

12.6.1 People's descriptions of their jobs

It might appear that the best way to discover how work is done is to ask the workers. The problem with interviews is that people are not able to describe their work accurately and in detail. What they often provide an abstract description of how things are supposed to work under "normal" circumstances, perhaps a textbook or classroom procedure they were taught. These are idealized statements that bear an unknown relation to the real work. For example, one commonly hears

the complaint, "I've had so much busywork this week that I haven't been able to get any real work done." This statement implies a categorization of activities into "busywork" and "real work," raising questions about what is actually getting accomplished and why other activities are not occurring. Some of the busywork might be making the real work possible, such as gathering information, assembling and preparing tools, getting feedback, etc. Thus one must understand the objectives and functions of the workplace and relate these idealizations to what occurs and why on any given day.

12.6.2 Formal representations of a work system

A work system may be represented by a wide range of formal methods, including legal regulations, organizational policies and procedures, flow charts, task analyses, project timelines, work breakdown structure charts, org charts, and so on. Such representations are useful in practice for managing schedules, budgets, personnel, etc.. However, such artifacts represent activities at a level of abstraction that is too general and idealized to be sufficient in modeling work practice. Instead, an ethnographer may begin by gathering such descriptions and use them as a starting place for observing workplace interactions and interviewing participants. Most obviously, if the BFU Report on relied only on skyguide's written regulations, analysts would not have known about the practice of reduced the number of ATCOs managing the Zurich sector on Sunday evenings.

12.6.3 Use of ethnography in prior Brahms models

Ethnography has played a central role for developing almost all Brahms models over the past two decades:

- Initial formulation of the Brahms language by simulating NYNEX work practices in telecommunications provisioning (order processing and installing point-to-point circuits for corporate customers in Manhattan; Clancey et al. 1998)
- Patient-caregiver interactions in a medical clinic at Kaiser-Permanente in Pasadena, CA (Clancey et al. 1998a; exists only as a sketch)
- A day-in-the-life of six people living in the Flashline Mars Arctic Research Station on Devon Island, Canada (FMARS, Clancey et al. 2005)
- Science operations in preparing programs for the Mars Exploration Rover (Seah et al. 2005)
- Space Shuttle ascent operations at JSC Mission Control Center (called MODATA, Sierhuis et al. 2007)
- File management operations between JSC MCC ground support and the ISS (OCAMS, Clancey et al. 2008).

In most cases the ethnographic study including on-site observations over several weeks and interviews. For Brahms-GÜM we visited an ATCC for one day and arranged several meetings with a few local air traffic controllers to provide some feedback about our analysis of Überlingen events. Ideally, we would have engaged in extensive observation and delayed modeling. However, the project timing and resources (one modeler), the lack of access to the people and places involved and

the time that has passed since the Überlingen accident, and the changes in equipment, plus the many reports about the accident suggested as a whole that we develop a model from the extensive material available online. Although not ideal, we believe this "ethnography at a distance" approach has been successful for our purposes.

12.6.4 "Ethnography at a distance"

Modeling the Überlingen accident in Brahms requires the kind of information about participants and their activities, locations, equipment, communications, timings, etc. that would in normal circumstances be obtained from sustained and careful personal observation of ongoing work activities, the standard ethnographic methods. However, in the case of an aviation accident, we obviously do not, and can not, have direct access to all of this information. Rather we must use what Diane Vaughan (1996), in her discussion of the research process on the Challenger launch decision, describes as "historical ethnography": the use of official documents, reports, interviews, and artifacts. Note that these are the central methods for historical analysis of recent events.

Within anthropology this kind of research, also called ethnography at a distance, is a known practice, exactly in conditions where direct observation is impossible. The classic study is Ruth Benedict's *The Chrysanthemum and the Sword: Patterns of Japanese Culture* (Benedict 1946). This book is based on research conducted on Japanese culture during the Second World War, commissioned by the Office of War Information and the Office of Strategic Studies. Benedict had never traveled to Japan, and of course was unable to do so during the war. She used written materials, movies, Japanese government documents, and interviews with Japanese nationals living in the United States: a data set very similar the one we have used to study aviation accidents.

Both these cases argue that ethnography at a distance can produce valuable results. However, the usefulness of ethnography at a distance depends on the nature of the probable cause of the accident as related to the available data. For example, if we had an accident that was affected by what readings were visible in a smoke filled cabin, we would need video data of the actual accident to be absolutely sure. We could use a similar cabin to set up an experiment, but if we did not know the density of the smoke and the physical efforts the crew made to get readings, we could only speculate about events.

12.6.5 Available data for Überlingen accident

Here we review the data that has been available for the study of the Überlingen accident, and indeed most aviation accidents.

Direct information about the situation

Direct information about the events. These are available as part of the accident investigation, in this case by the German Federal Bureau of Aircraft Accidents

Investigation: *Bundesstelle für Flugunfalluntersuchung*, or BFU. Direct information includes:

- Transcripts (and translations) of conversations within cockpits, and between pilots and Air Traffic Controllers, from cockpit recorders (black boxes) and audio recording of Air Traffic Control interactions.
- Recordings of aircraft instruments, depending on what recorders survive and are recovered.
- Reports on the remains of the aircraft, depending on what remains survive and are recovered.
- Information about the equipment and capacities of the aircraft involved in the accident
- Filed information about the flight plans of the aircraft.
- Information about training and history of specific flight crew members and air traffic controllers.
- Training manuals and regulations of the airlines and air traffic control facilities involved in the accident.

Indirect information about the situation

- The analysis and conclusion sections of the official accident report.
- Interviews with surviving participants (not available in the case of Überlingen, except for employees of the Air Traffic Control Center in Zurich managed, who could describe general conditions, though not the specific circumstances).
- Interviews with pilots and air traffic controllers, who can assist with domain knowledge, and describe what would be normal procedures and responses in the circumstances of the accident. In our investigation of the Überlingen accident, we have had available to us NASA subject matter experts in piloting and air traffic control, whose guidance has been extremely important in helping us understand aviation language, procedures, what responses are normal, what responses are unusual, etc..
- Observation of other Air Traffic Control Centers, and Air Traffic Control Simulators at NASA. We have had the opportunity to visit the Oakland Air Traffic Control Center, as well as air traffic control simulations at NASA Ames Research Center. These observations allow us to understand better the work of air traffic control in terms of the physical plant, the equipment, the pace of work, and the ways in which controllers communicate.
- Analyses, including: the conclusions of official accident reports, dissenting opinions to accident reports, academic and professional analyses of accident data and reports. It must be remembered that each of these forms of analysis has its own bias. Most particularly, official investigation reports attempt to analyze the data to determine who is to blame. While these reports can be quite wide ranging, their primary purpose is to provide testimony for the legal process that follows an accident. In practical terms, this means that many details that would be important for a work practice model are not included, since they have no legal consequences.

As well as these fairly standards forms of data for historical ethnography, in the case of Überlingen (and a number of other aviation accidents) TV dramatizations provide additional valuable though somewhat problematic source of information.

There are several dramatizations of the Überlingen accident: one produced by the BBC and one by the Canadian Broadcasting Company. These dramatizations are extremely useful in helping one to visualize what was happening. But they must be used carefully, because the details of speech, physical locations, and equipment are not necessarily accurate to the level required by a BRAHMS model. Also they necessarily contain interpretations of the data that must be understood as someone's interpretations, as opposed to fact. For example, the CBC dramatization edits out other flights being handled by the air traffic controller, in order to concentrate on the story of the two soon-to-collide planes and the late arriving plane coming in for a landing. At the same time, the narration and music of the dramatization suggests that the controller's workload is extremely high, an interpretation that our subject matter experts in air traffic control dispute.

The point is that these dramatizations are extremely valuable, providing a rapid way for the investigator to get up to speed on what happened, and making the official transcript much more comprehensible. At the same time, it is necessary to remember that these dramatizations are an interpretation leaning in the direction of a story deliberately dramatized to be entertaining or at least to keep the viewer's interest. They must always be checked against official transcripts and other sources.

12.6.6 Missing information important for refining and validating Brahms-GÜM

Clearly there is a great deal of information, interpretation, analysis and speculation available about the Überlingen accident. Nevertheless, further information would be valuable to increase the precision of the model that might be determined with ethnographic observation, on the spot interviews with personnel, and perhaps records that are not publicly available. The following are some illustrative examples of missing information that would improve the simulation model and/or validate our assumptions.

Information about the Zurich ATCC

- Exact dimensions of the distance between the two workstations being operated by ATCO and the minimum time it would take for ATCO to move between one station and another. When ATCO notices the impending collision is affected by the cumulative time for his many moves back and forth.
- Typical radar monitoring sequence of looking and manipulating used at the time in Zurich, providing an information flow analysis (Hutchins 2000) and data about when separation issues were typically detected and handled.
- Information about the shift handover practice. For example, was there a shift change binder ("read-and-initial" binder) containing information about activities on the upcoming shift, as is standard in US control rooms? We

would like to know what ATCO and other ATCC staff actually knew about the upcoming maintenance activities.

- Relation of SYMA to ATCO's activities. Observation would tell us more about the role of the SYMA (Systems Manager) and how this position interacted with ATCOs. One of our questions has been why the ATCO on duty did not call SYMA in to help as he discovered that his equipment was degraded and his workload was rising or why the SYMA, if responsible for systems administration, was not proactive in ensuring that the maintenance did not disrupt operations. The essence of a "generalized model" is to represent best practices and not just absences, omission, and malfunctions (Section 8.1); our understanding what other people might have done in similar circumstances is limited to what is available in published information.
- Past history of maintenance operations including frequency, effects, issues. For example, had the phone system ever been disabled in previous maintenance efforts?

Information about the flights

- Details about the AEF flight that distracted the ATCO. The BFU Report refers to this flight only as "a late-arriving Airbus 320" whose schedule was not known to the two ATCOs. The ANSA transcript identifies the flight as "AEF1135 Aero Lloyd (Aero Lloyd Flugreisen)" (p. 57). Whether the plane was arriving from the north or south could affect ATCO's perception of the other aircraft on the radar display. We do not know whether there was some reason for urgency for the ATCO to handle this flight personally, rather than deciding sooner to instruct the pilot to contact the Friedrichshafen tower directly.
- Frequency of charter and late-arriving flights like the BTC and AEF on a Sunday evening.
- Transcript of the BTC cockpit, full 30 minutes if possible, to better understand the crew dynamics.

12.7 Case Study: Brahms MER Validity Failure

None of the sponsors of projects in which Brahms has been used to date have provided funding or resources to enable validating the model. Instead, project managers supporting the efforts deemed that the insights gained from the experience in developing the model to be sufficient for the purposes at hand (the NYNEX and MODAT models) or the models were themselves experimental efforts to evaluate the capability of the modeling framework, as in the case of FMARS and Brahms-GÜM.

The OCA "Current Operations" simulation (Section 12.5) was essentially validated by the work group's judgment that the metrics (e.g., time devoted to particular tasks) were plausible. However, no effort was devoted to analyzing and quantifying video data that could have validated the durations for primitive activities in the model. The "Future Operations" simulation itself was used to validate the OCAMS

software agent design, however after the OCAMS system was deployed, there was no effort to compare what actually occurred (e.g., the time required to interact with OCAMS) to the predictions made by the simulation.

In the deploying OCAMS by converting the simulation agents to runtime agents, we did encounter difficulties because of simplifying assumptions about how computer systems were used. A related issue had arisen in developing Brahms-MER (Seah et al. 2005) in which a difficulty arose during operations training sessions that we had failed to discover when simulating the same operations. We drew two conclusions from the experience: 1) a simulation should explicitly model and validate how people provide input to software tools, what output they specifically "read" from the program, and what they do with this information; 2) the purpose of a simulation should be well defined, preferably to answer design questions and/or evaluate a particular tool or process (rather than being an unfocused "model of work practices" in some setting). In particular, the Brahms-MER simulation failed to represent that some data was being transformed manually (between the SAP and MAPGEN programs) and the time required. During the MER training the manual work was found to take so long that it justified developing an intermediate tool (the "Constraint Editor"). The Brahms-MER model represented the SAP output and MAPGEN input, but the model of MAPGEN was incorrect, assuming it was in the form provided by SAP.

Relating prior experience to Brahms-GÜM, we recognize in particular that the work required to use a radar display and interpret the data has been greatly abstracted—only the final practical beliefs about flight information are modeled and they are directly accessible without requiring intermediate steps of instrument manipulation, data lookup, correlation, calculation, logical inference, etc. In effect, the workload is not simulated at the grainsize that could properly account for the physical and perceptual work required (and this is common to most task/functional models as well). In this respect, one must be cautious in concluding what should have been "obvious" to the ATCO at certain points in time. It is precisely here that ethnographic observation to validate the model would be important.

In conclusion, validating Brahms-GÜM and similar models would be important in using the simulation results for work system design or forensic analyses. This project has focused on demonstrating the value of the Brahms framework for modeling and simulating complex interactions of people and automated systems relevant to aviation safety. Carrying the work further to use Brahms-GÜM in improving TCAS or similar systems for example would necessitate validating the simulated objects (e.g., radar display) and how they are used, as well as the protocols and all of the conditional sensitivities (e.g., aircraft separation at which ATCO notices and intervenes) and timings (e.g., how often and for how long ATCO monitors flights) related to work practices. Related validation is required that would observe how pilots notice, interpret, and respond to TCAS, stressing their use of other information and their communications in the cockpit.

12.8 Research Issues in Specifying and Verifying TCAS Itself

To this point, we have focused on the broad problem of verifying and validating a work practice simulation. However, our discussion would not be complete without commenting on the subproblem of verifying and validating the automated systems contained within the work system. In particular, the difficulty of specifying the TCAS algorithm and verifying the program highlights why a work practice simulation is essential.

Simply put, if TCAS were infallible, then pilots would know to always do what TCAS advises and to always ignore other sources of information, including the ATCO. The issue of authority would be trivial: TCAS would always overrule ATCO and the pilots' own judgment. One would simply need to do what TCAS says and making the air transportation system safe would reduce to installing TCAS and proper training.

The fundamental problem with TCAS is not just that it is demonstrably fallible (e.g., Kuchar and Drumm 2007; EUROCONTROL 2009), but the program itself has become difficult to understand and modify. A Lincoln Labs report (Kochenderfer et al. 2012b p. 28) states:

> The evolutionary development process of the collision avoidance logic has resulted in complex pseudocode with many heuristic rules and parameter settings whose justification has been lost over the years. Unfortunately, due to the complexity of the system, correcting issues without introducing new vulnerabilities is very difficult and costly. Next-generation procedures and new sensor systems will require reengineering much of the logic and tuning many parameters.

Kochenderfer et al. (2012b) advocate a principled, model-based approach to generating the TCAS logic:

> Recent work…has pursued a new model-based optimization approach to developing logic that has the potential to shorten the development cycle, improve maintainability, and enhance safety with fewer nuisance alerts. This new approach involves using computers to directly optimize the logic based on encounter models of traffic and performance metrics...
>
> The computer-generated logic will still have to undergo rigorous safety analysis that may result in modifications to the model or metrics. However, the development cycle will be shortened because the logic does not require manual revision (p.29).

This report shows that, even without considering interactions among pilots, ATCOs, and TCAS as we have done in Brahms-GÜM, specifying and verifying the TCAS algorithm itself is still a difficult research problem.

This critical evaluation of TCAS—that parts of the program's operation cannot be validated because justifications have been lost—strongly supports the view that TCAS advisories today should be interpreted by pilots as relevant information, but not indisputable commands. Furthermore, the fallibility of TCAS itself strongly

justifies a work system simulation that incorporates the judgment of pilots and ATCOs because in practice they must relate multiple sources of information to TCAS to decide what to do when an RA occurs. This fact provides another perspective on the argument articulated above that certifying TCAS requires a work practice simulation—TCAS not only has the function of affecting what people do, what they do must involve relating multiple sources of information because they know that the RA could be unnecessary or wrong. Hence understanding the *effect* of TCAS on safety requires understanding the total work system.

TCAS's limitations also suggests that one way of improving safety of the total work system is to expose the program's vulnerabilities to pilots. In particular the probability of advice being incorrect in a particular situation (e.g., using the probabilistic estimates Kochenderfer et al. derive from historical data) could be presented to the pilot. This approach would be in line with the 1970s expert systems explanation perspective that a fallible system should provide some means for people to evaluate its advice, that is, for its logic to be transparent.

13 Conclusions and Future Research Recommendations

This chapter considers what we have learned of relevance to NextGen in developing Brahms-GUM and provides recommendations for future research. We begin by reviewing the objectives and method for using Brahms to simulate a complex human-automation work system, emphasizing the generality of Brahms-GUM.

We then relate and evaluate Brahms with respect to the questions and topics that have framed research in the "Authority and Autonomy" task within NASA's Aviation Safety Program. We critically review how Brahms-GÜM could be improved to simulate 1) how different agent "ontologies" or ways of categorizing the world interact, 2) mental models, and 3) emotional-physiological modes influencing work behavior. We present methodological lessons learned in developing a work system simulation for air transportation systems using frameworks like Brahms.

Turning finally to aviation safety lessons learned and recommendations, we review TCAS training issues raised by the BFU Report and comment on the relevance of Brahms-GÜM to the recommendations by the Panel on Human Factors in Air Traffic Control Automation (Wickens, et al. 1998) concerning the effect and role of automated systems in the airspace system.

13.1 Summary of Objectives and Method

In this project we have investigated a "benchmark" scenario drawn from current air transportation systems, in which pilots and controllers interact with TCAS and other automation systems. The Überlingen accident was specifically chosen to understand interactions when authority changes in controlling flight paths. The accident also has many anomalies contributing to the collision, which can be simulated in different combinations to cover a wide variety of pilot-automation and controller-automation interactions.

We have developed a general ATS model (*not* just a simulation of Überlingen events), called the Brahms Generalized Überlingen Model (Brahms-GÜM). Brahms-GÜM can be configured in different ways such that *anomalous events* that contributed to the Überlingen accident can be modeled as functioning according to requirements (or following operating procedures) or in an anomalous condition, as occurred during the accident. For example, telephones in the ATCC can be simulated in Brahms-GÜM as operating normally or disabled as they were during the accident.

The configurable events in Brahms-GÜM include causative factors such as *human actions* (e.g., an ATCO absent from the ATCC) and *preferences* (e.g., at what point lateral separation requires action), *otherwise everyday occurrences* (e.g., a delayed flight departure time), and *automation parameters* (e.g., the refresh delay on the ATCC radar display).

The simulation model is designed such that these factors are specified as initial conditions when the model is simulated (see Appendix 24 for configuration

examples). Every configuration of factors, together with fixed components and behaviors in the model, constitutes a *scenario*.

To be clear, Brahms-GÜM provides several kinds of generality:

- Initial model conditions can be configured:
  - to include other flights on different flight paths and their scheduled departure times
  - to specify whether and when the second ATCO goes on break
  - to specify whether pilots obey the TCAS RA
  - to specify sensitivity of ATCO to separation infringement
- Agents can be cloned using copy and paste methods to add more ATCOs, for example, with other workstations, telephones, etc.; or to add more pilots on other flights.
- Geographic areas can be cloned using copy and paste methods to add more sectors, ATCCs, planes with cockpits, etc.
- Models of people and automation can be independently modified to change conditions for behaviors, durations for primitive actions, probability of detectables and inferences; methods (WFs) of activities can be added/deleted or reprioritized independently.
- New models of automation can be added independently; e.g., one could change what appears on a radar display by adding a new kind of alert, without necessarily modifying when agents monitor or how they interpret the "beliefs" represented by the radar.

Brahms-GÜM is designed in a general way to enable "what if" experimentation over a wide range of scenarios, as well as subsequent adaption (reusing simulation components) for modeling and simulating NextGen scenarios. We emphasize that the Brahms-GÜM is not per se a specific model of the Überlingen accident, but rather *a kind of "superset" model that implicitly defines a class of scenarios*, which include as an instance what occurred at Überlingen.

One can create variations of Überlingen directly by modifying the other flights handled by the Zurich ATCO in frequency and duration of work required (when AEF 1135 arrives in the airspace), setting any of the anomalous conditions, etc. But Brahms-GÜM provides a yet more general test bench with reusable components: One could simulate any other air sectors, flights, and extend it to experiment with new forms of NextGen onboard or ground automation. In effect, Brahms-GÜM is a framework for modeling and simulating human-automation interactions among aircraft, flight systems, crews, ATCOs, and systems in ATCCs.

This report has illustrated the analytic method for creating this general framework by generalizing the work system configuration and events that occurred during a specific incident. Our success in simulating a variety of related scenarios (e.g., Table 8-1) demonstrates the extent to which Brahms, a multiagent work practice modeling framework, facilitates this generalization. In particular, the complex

space-time interactions of people and systems—which provide an environment for each other—motivates a behavior-based "total systems" perspective for understanding the variety and effects of interactions among people and automation. We contrasted this approach with a functional-allocation model that might be employed synergistically in a formal verification process.

The process of creating Brahms-GÜM has clarified: 1) *the nature of complex work systems*, particularly cognitive complexity in operations (e.g., how a complicated but routine situation becomes complex and out of control) and 2) *emergent effects in a complex system*, particularly temporal relations, and why these relations are difficult to predict or even comprehend (and what statistics and visualizations facilitate understanding and perhaps predicting them). The process of creating a model focusing on human-systems interactions—and particularly understanding the emergent effects—guided our search for and interpretation of historical data, with many experiences of re-looking and seeing new "facts" in the transcripts and charts that we had been studying for months.

13.2 Conclusions about Using Brahms for Aviation Safety Simulations

In this section we relate and evaluate Brahms-GÜM with respect to the questions and topics that have framed research in the "Authority and Autonomy" task within NASA's Aviation Safety Program. These comments may be viewed as a high-level comprehensive appraisal of what the Brahms-GÜM effort—a work practice simulation of complex human-automation systems in safety critical situations—has accomplished with respect to the approach and goals of this research program.

- *How and to what extent can complex scenario simulations be used to verify and validate new work system designs to a sufficient confidence level that we can begin to introduce them (e.g., they adequately represent the actual environment)?*

Brahms-GÜM demonstrates that we have the capability to develop complex scenario simulations that can be used to verify and validate new work system designs. Demonstrating that such simulations are sufficiently accurate and valid to use them with confidence for introducing new automation systems requires a great deal of further research, including simulation of crew interactions, more and higher-fidelity models of onboard and ground ATCC systems, and extensive validation from ethnographic and human-in-the-loop performance studies.

- *Does this simulation approach provide appropriate metrics for evaluating safety and performance properties relevant to regulatory processes and policies?*

Brahms-GÜM can be used to generate a variety of relevant metrics. A multi-year research effort is merited to develop methods for instrumenting the Brahms framework itself to generate and assist analyzing metrics and to relate such metrics to regulatory processes and policies.

- *Consequently, how can current V&V processes and tools be revised or extended to refine and amplify simulations of complex scenarios?*

The application of V&V research to Brahms-GÜM is a related but separate project. However, our results and analysis suggest that V&V processes and tools might focus on methods for efficiently defining and focusing the space of modeled variables (e.g., simulating flights entering ATCO's sector). Another research project might specifically characterize tools that would assist the refinement steps in developing Brahms-GÜM (Chapter 10).

- *What is the efficacy and feasibility of the work practice simulation approach for designing and evaluating the effects of display, alerting, and procedures on safety?*

Because of the total systems approach allowing variable model fidelity of any human role or subsystem, Brahms-GÜM shows that developing work practice simulations is feasible and usefully informative for designing and evaluating automation. In particular, the Überlingen scenario illustrates that a work practice simulation is particular appropriate because unlike many human factors models in aeronautics, it includes and relates the behaviors of the ATCC and the cockpit, as well as automated systems they rely on for safety. TCAS exemplifies how a system can usurp a person's authority, and the simulation shows how failure for this change in authority to be communicated immediately to ATCO may result in a disruption of TCAS' actions and the designers' assumptions.

In short, Brahms-GÜM research suggests that a work practice simulation is especially well suited to designing and evaluating the effects of display, alerting, and procedures on safety. The framework is compatible with future efforts to model formally-checkable strategies for task performance, management of communications, and maintenance of situation awareness under conditions where authority is re-assigned (or managed) across human and automated agents in a variety of ways, including dynamic changes. Therefore, such total system simulations could be of considerable value within a comprehensive V&V process for NextGen A&A problems.

- *Can we measure the time to deal with perturbations and the risk of a problem occurring?*

We have shown in some detail (Section 10.5) how Brahms-GÜM enables modeling how distractions (e.g., late-arriving flight, dysfunctional telephone system) can perturb the routine process of monitoring flights and lead to separation violation. Further, the intervention of ATCO before and after a TCAS TA influences whether an RA is required and if so the particular advisories generated for each aircraft. Varying the model of any aspect (e.g., time devoted to handling phone problem; prioritization of activities such as flight handoff and routine scanning of the radar display), enables generating metrics on sensitivity of effects and frequency of

different sequential behaviors that result (e.g., how soon before TCAS RA intervention can be effective rather than being disruptive).

- *Could such a simulation help design methods for quantifiably detecting that a system is starting to fail during operations?*

A simulation like Brahms-GÜM can generate metrics like those mentioned above to detect "intolerable" configurations, such as determining that the current workload exceeds what ATCO can manage, requiring a change in strategy (e.g., instructing AEF to contact the tower directly immediately) and/or reconfiguration of the flight paths (e.g., detecting that BTC has arrived in the sector at the same altitude as was provided to the crossing DHL flight). In effect, one way of characterizing "starting to fail" is in terms of accumulating conflicts that need to be resolved and/or routine operations that are consuming more time than usual and thus delaying required actions.

Much of the research direction in A&A focuses on law-like properties, such as official regulations, including separation violation. Focusing as well on cognitive complexity, we could for example define properties such as "ATCO should spend X% of time monitoring larger air space" and "ATCO should not have more than two pending tasks." Such thresholds might provide advance warning of undesirable behaviors:

  - failures to carry out certain activities according to expected schedule/intensity etc. in normal practice, e.g., not monitoring larger airspace;
  - becoming reactive, flitting about, juggling several things at once, allowing quality of work to degrade, e.g., ATCO allowing AEF to dominate his attention; not monitoring DHL and BTC after the intervention;
  - not realizing that certain events or information could have or would have consequences downstream; more generally, not thinking ahead, a kind of tunnel vision and action, e.g., clearing DHL to FL360 and then not thinking through the implications of BTC's reported altitude.

Such properties could be interpreted as indicating that the work system is transitioning from being complicated to being complex. In effect we could instrument the Brahms engine to provide a form of prognostic indication of increased risk of system failure.

- *What are the advantages of a unified agent architecture like Brahms versus simulations that allow a variety of representations?*

The Brahms framework enables flexibly integrating the simulated world of Brahms agents and objects with other simulations as well as operational software (through an API). This enables adapting and refining simulations efficiently and pragmatically, promoting reuse of model components, with all of the forms of

generality outlined above. The inherent modularity of the agent-oriented and object-oriented architecture facilitates creating simulations incrementally and making local modifications experimentally (e.g., Table 10-2).

The key concepts provided by Brahms that make it a unified architecture are: 1) parallel simulation of processes that may interact at different levels (e.g., ATCO's intervention interacts with TCAS through simulation of verbal commands), 2) uniform use of message passing for communications among people and systems (with domain-specific message structures as required), 3) simulation of located, chronological behaviors of people and systems (i.e., modeling activities within an abstracted geographic model of facilities, vehicles, landscapes, etc.), 4) simulating human perception through "detectables" (Clancey et al. 1998b), 5) simulating "broadcast" (verbal out-louds) and movement as behaviors that are conditional on agent beliefs (model of world), 6) simulating inference as occurring within an activity.

- *How does "scenario-based thinking" relate to analytic/verbal arguments, e.g., establishing a different kind of evidence that a system is safe?*

With respect to scenario-based thinking, Brahms enables simulating a work system in a general way, such that the simulation can be configured to model a large number of scenarios. Simulating scenarios promotes *systems thinking* in both analysis and in modeling. A Brahms simulation of pilot activities in extended flight (over multiple sectors) requires modeling air traffic controllers, and this requires modeling the tools both parties are using to inform and carry out their communications and to execute the requests they might make of each other (e.g., pilot request to change altitude). A Brahms simulation is to an analytic/verbal argument as a play is to reviewer's synopsis and critique. As representations, the play and the review are of course complementary. With respect to aviation safety, the Brahms simulation provides a test bench for generating metrics, for doing what-if experiments, for studying causal-temporal relations in a systematic manner. In effect, the value of a scenario-based simulation relative to analytic/verbal arguments is similar to the value of having a model for scientific and engineering over having just lists of facts, assumptions, and inferences. The model can be used to suggest hypotheses that leads to more data collection, refined theories, and new experiments. Scenario-based thinking in Brahms has the potential of making work system design a scientific enterprise.

- *What shortcomings were uncovered by unexpected research results, suggesting requirements for further research?*

Appendix 28 discusses some limitations in the Brahms language; the following sections describe how models might be improved and the methodological lessons learned.

13.3 How Brahms-GÜM Could be Improved for Simulating Complex Human-Automation Systems

The timing sensitivity and variety of behavioral sequences (Table 10-3) that result from running a single scenario (Überlingen, 1F in Table 10-1) demonstrates that the Brahms framework enables modeling the variability and dynamic implications of a work system that combines simultaneous agent activities and subsystem processes. Further, the model is general to the extent that it can run in different configurations (scenarios) in which the simulated actions of people and automated system interact and influence each other in unexpected ways.

In the following sections we elaborate on the advantages of Brahms for simulating complex human-automation systems, and correspondingly how Brahms-GÜM might be enhanced to exploit the Brahms framework.

13.3.1 Modeling different agent and object ontologies

Pilots and ATCOs have different ways of modeling flight activity. That is, they have different ways of characterizing (categorizing) the objects, their properties, and processes of the world, which constitutes an *ontology*. For example, they track progress of flights relative to different "progress landmarks." Generally, speaking pilots think and talk in terms of distance—"where am I within my arrival and approach path?"; ATCOs think and talk in terms of time—"When will X occur? Is there time for Y to do X?" Put another way, the mental models of ATCOs, pilots, and automation will differ in describing "the same" events, tasks, goals, etc.

Brahms enables variation of mental models because the entities and attributes that make up each group or agent's beliefs can be modeled independently. In principle, all of the agents and objects can "think in different languages." Of course, any communications will be in terms of a shared ontology, and any belief gained by a Brahms detectable in terms of the ontology of the world model, the so-called objective "facts." An agent or object can translate from one frame of reference to another in inferences (TFs).

Effectively the ATCOs are managing a large airspace with many planes as well as the runways. They are locating planes in space, focusing on separation and allocation of places. The pilot is on a trajectory and is instructed by ATCO about when to arrive at the waypoints. The pilot flies with a single-minded, one-directional orientation, geodetically aligned for efficiency—the shortest path from A to B is preferred. The ATCO is managing *flows*, streams of planes passing through the same airspace and airport regions.

The pilot locates a single point (the plane) in 3-d space, constrained by altitude, pitch, roll, and speed. The ATCO locates an entire population of planes in 3-d space (e.g., the ARFA Sector, Figure 9-2), all in motion, but attempts to sustain a dynamic equilibrium: planes arriving and landing at a certain pace and so many planes in a given flight path, separated by fixed amounts. So despite the entire system being in constant motion, the dynamic properties are bounded and routine.

We can experience the ATCOs perspective by observing the stream of planes at night coming into an airport. We can routinely see several planes, maybe even five or more, progressively distant on the same flight path, descending to the airport, spaced more or less equally apart.

Knowing the ATCO's ontology enables us to better understand and model his work. For example, how much time does the ATCO have available for advance notice, given the frequency and duration of interactions, etc.? What does the incoming stream look like? How do events bunch or backlog? How are these influenced by the circumstantial additions and subtractions of speed on route from wind and rerouting? In particular, future research using Brahms models could examine the practicality of viewing the modern airline pilot as a "manager" (as prescribed by Casner 2007)—what ontology characterizes the activities of monitoring, controlling, and communicating?

13.3.2 Modeling the "mental models" of agents and objects

The notion of ontologies highlights a key opportunity we have in modeling work practice within a multiagent system: We can simulate the mental models of people who have different frames of reference in different work settings (with different tools, responsibilities, support structures, etc.). This is close to the bedrock of the theoretical perspective of "situated cognition" (e.g., see Clancey 1997)—ontologies vary with specialization of the agent and there is no single, right way of characterizing reality. The "right model" depends on purposes and more specifically, the distinctions useful for planning, choosing, and communicating actions.

In effect, articulating how group ontologies differ is part of modeling their culture. How are the different practices of the pilots and ATCOs manifest in their language? In what activities using what tools does their language for modeling the planes and airspace region differ? To what extent do the tools impose a role-specific way of modeling the world? To what extent have pilot-ATCO voice communications established a common ground that will be modified or disturbed by the ATC–plane Data comm link?[56] Does the new practice using text displays, printouts, and a button acknowledgement establish new "boundary objects" for coordinating their work functions? Does not repeating an instruction verbally as confirmation affect the pilots' comprehending, remembering, etc. the flight directive?

A secondary theoretical point is that people's ontologies are fluid, open-textured (i.e., not precisely defined in mathematical-logical terms) because they are references to *conceptualizations*, which are relational, coordinating ways of perceiving and acting. This leads to the fundamental corollary that automation works within fixed ontologies (the language of its mental model), whose

[56] FAA information about Data Communications being deployed for NextGen: http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/atc_comms_services/datacomm/

interpretation and action implications may be adaptable, but only within another fixed ontology of permissible variation.

These are crucial issues to model, not least because they fit very well how Brahms work system simulations make social-psychological processes visible. In particular, the different attentional foci of pilots and controllers are not visible to a casual observer. This critical aspect of current operations could well be ignored or minimized in designing NextGen automation systems. Brahms simulations provide a way for engineers to talk about and model how expertise, attention, and knowledge interact in operations.

Brahms-GÜM can be augmented to represent agent and object ontologies, thus better modeling the work required to adapt a work practice to the data or services provided by a tool. Such a model would also reveal the differing interpretations pilots, ATCOs, and automated systems infer from their interactions with each other. With respect to the social character of work practices, research could focus on how human-systems relations parallel or different from teamwork among people, and how these relations make interaction easier or more problematic.

In particular, models of mental models must include people's understanding of the goals and methods of automation, as well as perhaps the assumptions design of automated systems makes about the goals and methods of people who are affected by it. Model checking could then include detecting discrepancies between mental models of agents and object and their actual states.

One could also model how people and systems *reflectively* model their own goals and methods *as part of their practice* to develop coherent models of *progress appraisal* (Feltovich et al. 2008). Such models would relate to roles, responsibilities, and ongoing status to characterize "What I am doing now." "How well is this task proceeding?" "What are the wider, long-term implications of this decision?" If the Zurich ATCO during the Überlingen accident were constructing such a mental model of progress, he might have noticed that fussing over the telephones was distracting from his primary role in ensuring safety of all the flights in his sector. Mental progress appraisal models of other agents/objects, could include for example, "is X functioning well doing his/its task now?" With such a mental model, representing an aspect of situation awareness, the Zurich ATCO might have cared to ask and verify what effects the maintenance work was having on the tools he relied upon for carrying out his responsibilities.

13.3.3 Modeling emotional-physiological modes

Creating a model of interactive human-automation behavior raises psychological questions about emotion and fatigue, exemplified by our analysis of why the Zurich ATCO shifted from handling the BTC/DHL separation to responding to AEF's final call, rather than confirming that the potential collision was resolved (Section 6.8.6).

Emotional aspects may be relevant, such as the relief of closure: ATCO may have conceived the problem as communicating with BTC, so their descent confirmation gave an experience of closure and ability to respond to AEF's call at that moment. The effect of fatigue on multitasking is also of course well known: ATCO's ongoing focus and frustration in getting closure in the AEF landing handoff was perhaps pulling him back towards the other radar scope, the activity in which had he become fixated over five minutes.

The Brahms framework does not have direct, built-in methods for simulating emotional-physiological modes of feeling/attitude, fatigue, stress, attention fixation, etc., which may be cumulative effects of workload. An approach for simulating physical properties of a person, which was used in the Brahms model of "a day in the life" of a Mars research station (Clancey et al. 2005), is to model the body of a person as an associated object with physiological attributes (e.g., need for eating, sleep, using a toilet) whose values are a function of time. These attributes could also be set by workframes that model emotional or fatigue effects of doing an activity (e.g., a function of the time required to accomplish a task). This object could then "communicate" beliefs to the agent that model feeling tired, focused on a task, etc., which would be included in WF conditions and hence influence what and how activities are carried out.

Another way to proceed might be to use a Brahms simulation to characterize the work and context in which an agent is operating, and represent these states and situations in other suitable modeling frameworks. Such a model could then be coupled to the Brahms simulation through the "body as object" approach. The Brahms language could also be extended; for example detectables might be dynamically modified by an associated model of fatigue (e.g., decrease the probability of noticing a fact in the world).

The Brahms engine could also make accessible to the "body model" data about the ongoing number of interrupted WFs, the frequency of interruption by higher-priority activities, the duration for carrying out an activity compared to a normal/expected duration, etc. These factors would represent workload that the body model could incorporate into the model of fatigue/stress, excitement/interest, displeasure/anxiety, etc.

13.4 Methodological Lessons Learned in Simulating Work Systems and Scenarios

As mentioned, Brahms-GÜM has been the most complex Brahms model we have constructed to date, in terms of the number of independently operating kinds of objects (e.g., aircraft, FMS, TCAS, radar) and agent roles (pilot, ATCO, CA). We sought to develop a system useful for model checking research within one year, which could not have been done without the single modeler's extensive experience using Brahms over a decade. The models of people and subsystems were necessarily scoped in detail and fidelity, as summarized at the beginning of Chapter 9. In particular, we had originally intended to model intra-crew roles and interactions to

simulate the BTC interactions regarding following TCAS versus the ATCO. It would be worthwhile to model also what the "DL identity" of ATCO would entail (BFU, p. 83) and how the ATCC might more properly adjusting roles to accommodate the maintenance situation (BFU, p. 84).

Developing the model incrementally resulted in being continuously surprised about timing effects especially, requiring improved fidelity in models of both subsystems (e.g., radar display sweep), aircraft flight paths (e.g., response to TCAS), the air sector (e.g., bounds of ARFA), pilots (e.g., acknowledging ATCO/TCAS after taking action), and ATCO (e.g., handling dysfunctional phones).

In managing such a technical project one would generally prefer to know and to have documented what needs to be in the model early on. Of course, the amount of detail would have appeared daunting, and in practice appreciating what needed to be included required many months of experience in critiquing the simulation output and research reading to grasp such detail. Nevertheless, in approaching such a model in the future, we will develop a technical design in the same manner we document realtime Brahms agent software systems. In particular, knowing that a one-second clock tick is required, we are now better able to anticipate the corresponding degree of fidelity required for agent and object behaviors. This said, the project schedule must allow for late discoveries and significant changes to the model, as we have documented in Chapter 10—nearly a third of the year was devoted to "refining" the model to produce sufficiently complete and coherent simulations.

Our prior experience suggested Brahms-GÜM would benefit from having three modelers at least, focusing on ATCC, aircraft/flights, and the cockpit. Research, design documentation, review, testing could have proceeded more systematically and efficiently with more personnel. In particular, the modeling process would have benefited greatly—as well as future research using Brahms-GÜM—by disciplined documentation of the source of every WF, what assumptions and simplifications were made, and the history of modifications with justifications (e.g., interaction effects). Such records were not necessary when developing other Brahms models, but the complexity of this simulation made it difficult at times to remember or to reflect on how the model was being designed (e.g., modeling reading the radar as analogous to communicating with a radio was inadequate to model ATCO's decision of which aircraft to advise to descend).

In using or adapting Brahms-GÜM for developing new automation or significantly changed work practices, our experience from Brahms-MER (Section 12.7) shows that detailed data-flow and use-case diagrams are essential for verifying workflow connectivity early in design of the simulation.

In short, a lesson learned from our experience is that, when shifting from a research project like that reported here to a development context, modeling complex real-world systems should follow standard software engineering practices; consequently

more personnel and time will be required than might be apparent from the accomplishments of the Brahms-GÜM project. This entails writing and maintaining model requirements specification and technical design documents, such as describing in detail how automated components like TCAS are modeled, including significant simplifications/workarounds and omissions. Such a model is too complex for a research manager to review in detail or sufficiently frequently.

It is particularly important that modelers not arbitrarily decide how to model processes without discussing the design. Such consideration and coordination takes time that must be factored into the project schedule. At least semi-formal designs created and improved during a relatively long analysis period is required. Issues discovered by later research reading (e.g., why ATCO believed DHL was at the proper altitude [FL360]; why ATCO repeated the intervention call to BTC) might have arisen during the model specification phase. Similarly, the failure to detect in Brahms-MER the workflow gap in MER operations workflow (Section 12.7) might have been transparent if model components and communications had been carefully specified prior to coding the model. Such analysis and documentation requires continuously updated, easily indexed descriptions of every modeled entity in terms of properties, behaviors, caveats, etc.

Speaking for the other point of view, one could also argue that the modularity of Brahms and a modeler's facility with the language enable frequent and sufficiently rapid modifications that one can use the model itself as a sketch pad. This approach has the well-known advantage of establishing a broad framework and "painting in" details as required. The team might experience might ongoing surprise and be repeatedly revising the same sections of the model, particularly of human behavior in complicated situations (e.g., ATCOs handling of AEF, other flights, monitoring, equipment dysfunctions). This is not surprising or a problem if the overall configuration is stable, as was the case in generalizing from the Überlingen events. Furthermore, insofar as Brahms-GÜM now serves as an adaptable framework with reusable components, future research efforts using the simulation for NextGen design and certification might be proceed in a more conventional, planned, and predictable way.

13.5 TCAS Training Issues and Cognitive Complexity

In this section we provide a brief summary of how training relates to Überlingen accident and how training is discussed in BFU Report. We consider how the perspective of cognitive complexity informs training methods relevant to automated systems like TCAS.

13.5.1 Training for pilots

The most obvious training issue raised by the Überlingen accident is the question of pilots' TCAS training, specifically the difference between the training of the DHL crew and the BTC crew.

The BTC crew, who violated the rule to obey TCAS rather than the controller in case of a conflict, had not had simulator training on TCAS. Their training was solely classroom based. The DHL crew had had simulator training. This makes it easy to conclude that presence or absence of simulator training was the deciding factor in the correctness of the crews' actions. However, the DHL crew did not face a decision about whom to obey; they had only the command from TCAS. Consequently, we cannot conclude what they would have done if like the BTC crew they had been advised by ATCO first.

In 2002, the Russian Federation did not require TCAS to be installed for flights within the Federation's airspace. Only planes and crews flying in airspace requiring TCAS, such as the European Union, needed to be TCAS capable. This explains the relative scarcity or absence of TCAS simulators, and hence simulator training within the Russian Federation. Data from simulator training on a variety of permutations of the Überlingen scenario would be useful to determine whether the order of commands received has an effect on whether crew members can make a rapid reversal in an emergency situation. In particular, Brahms-GÜM might be incorporated in a human-in-the-loop simulation, in which people serve as ATCOs at workstations with simulated Brahms pilots (or vice versa).

13.5.2 Training for ATCO and ATCC management

The BFU Report makes the observation that the ATCO on duty had not received training as a DL (supervisor) and thus lacked higher level skills of human factors and Team Resource Management:

> When the ATCO released his colleague to take an extended rest break he assumed responsibility for the duties of both the RE and RP, and some of the duties of the DL. He was assisted by one CA. He was neither aware of the potential support available from the ATCO and SYMA assigned to the team of technicians, nor the technical expert assigned to him.
>
> Although the ATCO had not received a training as a DL he was aware of these duties. However, he was more familiar with his role at the heart of the operational ATM team and was not so familiar with the subtle differences of supervising the overall CIR ATM system; an advanced sociotechnical system. Normally when the ATCO had a situation requiring support he could call on the DL to assist. Efficient supervision of the system by the DL would ensure the ATCO was afforded the appropriate resources at the "sharp end" to best manage the air traffic situation. (BFU Report, p 83)

Lack of DL training is mentioned here as only one of many problems associated with SMOP for the night shift. The BFU report also notes that due to staffing shortages standard refresher training was not available as often as required, and that in any case, supervisory training was not provided for ATCOs:

- Several controllers at ACC Zurich were only insufficiently informed about operating the radar system in the fallback mode. Operation in this degraded

mode was not regularly trained, nor had suitable guidance documentation been developed. Although operating in the fallback mode is not overly problematic or inherently unsafe it must be understood how the system's defences are affected.

- The night shift ATCOs were expected to assume some functions of the DL but were not trained to handle the role.
- Refresher courses were scheduled every six months. Due to the tight personnel situation they were run on an annual basis only.
- There was no comprehensive training in emergency or unusual procedures in a suitable simulation device included in these courses. The company did not provide specific and suitably detailed material to the ATCOs for the handling of emergency or unusual situations. (BFU Report, pp. 91 – 92)

Given that the accident analysis reveals systemic flaws in the Zurich ATCC, training for both middle and upper management is essential, too. In trial of skyguide, the defendants were four middle managers, three technicians, and the air traffic controller; but upper management at skyguide was not indicted. Were the staff shortages at the Zurich ATCC also partly caused by business decisions by upper management?

13.5.3 Training cannot guarantee safety for cognitively complex systems

It is easy and common to assign to training the requirement of mitigating vulnerabilities in a work system design: make sure that people do the right thing. Insofar, as Perrow argues, it is impossible to give prescriptions that will insure that accidents will not happen for a system that may become cognitively complex during operation, this approach provides no guarantee of safety and may obscure or prevent taking seriously the flaws in the work system design.

Johnson (2004b, p. 32) makes this argument in his analysis for Eurocontrol of the BFU Report:

> There is a paradox in the BFU report. Not only does it criticise the risk assessment practices of the Air Traffic Service provider. It also uses a form of "20-20" hindsight to criticise the training of Air Traffic Control Officers on the basis of the problems that emerged during the Überlingen accident. There seems to be an assumption that because errors were made then training must have been to blame.

Johnson insists that the analysis must recognize that the ATC system had become cognitively complex and hence not resolvable by training:

> The focus of the first report in this contract has, however, been to focus more on the problems of risk assessment. Even if the relevant personnel had been better trained there is little evidence and few guarantees that they would have been able to cope with the demands that their operating environment placed on them during this accident.

In particular, Johnson suggests that the Überlingen collision provides evidence that the work system at the time of the accident was not safe because the configuration had changed (and continued changing) in ways unknown to the ATCO:

> The BFU argued that "although operating in fallback mode is not overly problematic or inherently unsafe it must be understood how the system's defences are understood" (BFU page 91). This statement is contentious. Certainly by the standards in other industries, such as nuclear power or military operations, it would be difficult to argue that the fallback mode is inherently 'safe' even if controllers have a complete understanding of the available defences. Recall also that these defences changed over time as the SWI-02 communications system was gradually brought back into service without the controller being informed.

People holding complex responsibilities might also be trained to recognize, as Don Norman notes, that a piece of technology, whether a computer screen or a soda machine, covered with sticky notes explaining how to use it can be assumed to be a bad piece of design. For a complicated task such as ATC, the very need for such intense training may be revealing the degree of risk involved in working with such tools.

Kathy Abbott's FAA study on operational use of flight path management systems illustrates how training for existing automation systems may be missing because not all emergency situations can be anticipated by the designers:

> … pilots have to deal with many failures for which there are no checklists, and for which there is no training of any kind, such as failures or malfunctions of air-data computers; computer or software failures; electrical failures; and uncommanded autopilot disconnects or pitch-up incidents for which the reason is unknown. Failure assessment is difficult, failure recovery is difficult, and the failure modes were not anticipated by the designers. (Learmont 2011)

Although the Überlingen accident provides an easy argument that crews should obey TCAS over ATCO instructions, the possibility of unanticipated situations suggests that TCAS's role must more generally be advisory. Rather than being an agent dictating action that must always be obeyed, TCAS is just an important source of information—implying extremely urgent need for attention—that the pilot must relate to the broader context and information.

The bottom line is that a judgment was and always is required in responding to TCAS—as the huge number of false positives attest: "false alarms may cause pilots to lose trust in the system and ignore alerts" (Kuchar and Drumm 2007, p. 277). Interpreting and responding to TCAS does not involve following a simple rule of doing what it says. Indeed, a 2005 study of 1725 TCAS RA "climb" events, cited by Kuchar and Drumm, showed astounding lack of compliance:

> Only 13% of pilot responses met the assumption used by TCAS: pilot responses within 5 seconds and achieving a 1500 ft/min vertical rate. In 63% of the cases, the

> pilots maneuvered in the proper direction but were not as aggressive or prompt as TCAS assumed. Pilots maneuvered in the opposite direction to the RA in 24% of the cases.

That so many pilots so frequently choose an opposite course of action suggests that in practice pilots have learned to use TCAS as an information source and have experience about when to ignore it. Accordingly, the BFU Report's lead systemic cause of the accident appears to be pointing in the right direction:

> The integration of ACAS/TCAS II into the system aviation was insufficient and did not correspond in all points with the system philosophy. The regulations concerning ACAS/TCAS published by ICAO and as a result the regulations of national aviation authorities, operations and procedural instructions of the TCAS manufacturer and the operators were not standardised, incomplete and partially contradictory.

What remains to be determined is how TCAS should be integrated into the aviation system and what regulations can be stated that are not incomplete and contradictory with respect to responding to TCAS.

13.6 Relation of Brahms-GÜM Methods and Results to Study Recommendations

To conclude this report, we comment on the relevance of Brahms-GÜM to the recommendations by the Panel on Human Factors in Air Traffic Control Automation (Wickens, et al. 1998, pp. 1-8). This National Academy of Sciences panel report focuses on the interaction of pilots and air traffic controllers, concerning particularly the design, effect, and role of automated systems in the airspace system.

This report is particularly germane to the use of work practice modeling for design and as a component of verification of work system designs including automated systems. In analyzing and simulating the Überlingen accident we have been looking back to the past, whereas in NextGen design we require methods for looking forward, that is a prognostic mode of analysis. The report specifically advocates modeling and simulating systems and interactions like those represented in Brahms–GÜM, such as "situations in which two or more coordinating agents receive information inputs that are incongruous or contradictory" and the effect on loss of situation awareness leading to separation violation from the use of higher-level decision and action selection systems. These and other factors that argue for Brahms' value for creating and evaluating work system designs are elaborated in each excerpted section that follows (titles of excerpts are from the report, pp. 1-8; emphasis added; comments follow each of the recommendations).

Levels of Automation

- The panel recommends that automation efforts focus on reliable, high-level automation applications for information acquisition, integration, and presentation and for aiding controller decision making in order to support all system functions. *Especially important in the near future is the development of decision aids for conflict resolution and maintaining separation.* These aids should be directed primarily toward ensuring

proper spacing between aircraft in preparation for the final stages of approach to landing and toward en route flight path efficiency improvement.

Our analysis of Überlingen events and results of the scenario simulation suggests that decision aids and alerting systems for maintaining separation and resolving conflicts are among the most important use of automation for air traffic controllers and pilots.

- The panel recommends implementation of high levels of automation of decision and action selection for system tasks involving relatively little uncertainty and risk. However, *for system tasks associated with greater uncertainty and risk, automation of decision and action selection should not proceed beyond the level of suggesting a preferred decision/action alternative*. Any consideration for automation at or above this level must be designed to prevent: loss of vigilance, loss of situation awareness, degradation of operational skills, and degradation of teamwork and communication. Such designs should also ensure the capabilities to overcome or counteract complacency, recover from failure, and provide a means of conflict resolution if loss of separation occurs.

This recommendation is consistent with viewing TCAS and corresponding alerts for ATCO as handling degraded monitoring and hence awareness of the current situation. TCAS 7.1's reversal mode could compensate for the kind of intervention at Überlingen that affected pilot response.

Recovery

- The panel recommends investing sufficient resources in studies of human response to low-probability emergencies; actively pursuing failure modes/fault tree analysis, particularly to *identify situations in which two or more coordinating agents receive information inputs that are incongruous or contradictory*; and involving human factors specialists in the development and testing of system recovery procedures.
- The panel recommends the development of models, for given designs and procedures, to examine the implications of recovery in a high-density, unstructured airspace created by increased capabilities of ground-based automation or free flight.
- *The panel recommends the development of airspace safety models that can predict the likelihood of midair collisions*, as a function of frequency and parameters of near-midair collisions and losses of separation, for varying standards of traffic separation. To do this, models should be developed that are *sensitive to loss of situation awareness and the possible degradation of skills that may result from moving controllers to progressively higher levels of automation of decision and action selection.*

Brahms-GÜM demonstrates how to model and simulate multiple coordinating agents with different, incongruous or contradictory information; it can be used to predict the likelihood of midair collisions as a function of many variables affecting situation awareness and causing loss of separation

- The panel recommends that air traffic control subject-matter experts collaborate with specialists in the behavioral sciences to *model individual and team responses to emergency situations and to populate the models with data to be collected in studies of human response time to low-probability emergencies.* Policy makers should be made

aware that choosing median response times to model these situations can have very different implications from those based on worst-case (longest) response times; these kinds of modeling choices must be carefully made and justified.
- The panel recommends that system functionality should be designed so that failure recovery will not depend on skills that are likely to degrade.

Brahms-GÜM could be used without further modification as a tool for focusing multidisciplinary modeling, data collection, and analysis of response times to low-probability emergencies.

Locus of Authority

- A ground-based scenario consistent with formulated plans of the Federal Aviation Administration can increase efficiency without radical changes in authority structure from the current system (e.g., the expanded national route program). The panel therefore recommends the *development and fielding of current and proposed automation tools for ground-based air traffic control,* following the guidelines specified in this report regarding the selection of levels of automation. We also recommend the *vigorous pursuit of projections of how various tools will operate in concert.*

The components of Brahms-GÜM could be directly used, adapted, and extended to simulate the operational processes and challenges of using a suite of ground-based ATC systems that operate in concert.

- Because free flight design concepts that assume a high level of airborne authority over control of aircraft flight paths have more uncertainties than design options involving ground-based authority with increased automation, the panel recommends extreme caution before existing levels of free flight are further expanded to greater levels of pilot authority for separation. Furthermore, we *recommend the conduct of extensive human-in-the-loop simulation studies and validation of human performance models before decisions are made regarding the further implementation of free flight*; this is needed to obtain reliable prediction of the safety implications of worst-case scenarios. We also recommend heavy reliance on scenario walk-throughs and focus group sessions with controllers, pilots, traffic managers, and airline dispatchers.

Brahms-GÜM could be coupled to other airspace simulation systems in human-in-the-loop simulation studies, using the methods developed for the realtime Brahms agent systems (Clancey et al. 2012). In particular, prototype automated systems or simulated systems (as Brahms-GÜM simulates TCAS) could be combined to understand the effects of increased pilot authority, as well as decreased authority by onboard flight advisory systems like TCAS that affect or impose the free-flight mode.

Introducing Automation

- The panel recommends that senior Federal Aviation Administration management should reexamine the results of the study by the Human Factors Subcommittee of the FAA's Research, Engineering, and Development Advisory Council, with a view toward implementing those recommendations that appear most likely to achieve more active,

continued, and effective involvement of both users and trained human factors practitioners in the development and implementation of advanced air traffic control systems. *All aspects of human-centered automation should be considered in fielding new automated systems.*

From the perspective of computational methods relevant to fielding new automated systems, Brahms-GÜM is a prime example of a human-centered computing technology that can facilitate and promote a holistic design and certification methodology.

- The Federal Aviation Administration should continue to support integrated product teams with well-trained human factors specialists assigned to the teams. *Both users and human factors specialists should be involved at the early stages to help define the functionality of the proposed automation system.* These specialists should be responsible to report to human factors management within the Federal Aviation Administration as well as to project managers.
- The Federal Aviation Administration should continue to work toward an infrastructure in which some human factors training is provided to personnel and program managers at all levels of the organization (and contract teams).
- The Federal Aviation Administration should ensure that *adequate funding is provided for needed human factors work at all stages of system development and field evaluations both before and after systems acquisition*.

Both air traffic controllers and pilots participated as time and funding allowed in developing Brahms-GÜM. With additional funding, much more could have been accomplished by collaborative design and evaluation, including using ethnographic methods for data collection and model validation (Section 12.6).

- *During the development of each automation function, system developers should consider possible interactions with other automation functions (under development or already existing), tools, and task requirements that form (or will form) the operational context into which the specific automation feature will be introduced.*

The Brahms-GÜM project has demonstrated that a multi-agent, activity-based modeling framework can simulate non-deterministic interactions of people and systems. In particular, the Brahms framework enables simulating how the actions of people and systems form the operational context for each other. Such a work system simulation can be useful at every stage in developing automation functions, but especially early in design.

Our experience analyzing and modeling the Überlingen accident underscores that verification and validation must be about the entire work system —and the properties of interest are that checks and balances are in place to handle failures of communication/alerting subsystems and/or failures of people to notice, comprehend, and/or communicate problematic (unsafe) situations.

Simulating how independently operating people, tools, and automated systems interact reveals how simple behaviors can combine to produce a complex system—an air transportation system in which the dynamic configurations of aircraft and computerized systems has become too difficult for pilots and air traffic controllers to perceive, understand, and safely control.

14 References

ANSA+AIRADIO 2004. "COMMENTARY on the Aircraft Accident of 1 / 2 July 2002 Mid-air Collision of Flights DHX611 (DHL) + BTC2937 (BAL) at Überlingen (Lake Constance) Germany," 24 January.

Argyris, C. and Schön, D. 1978. *Organizational Learning: A theory of action perspective*. Reading MA: Addison-Wesley.

Aviation Knowledge. 2011. "Bashkirian Airlines Flight 2937 & DHL Flight 611: mid-air collision – analysis."
Available: http://aviationknowledge.wikidot.com/asi:bashkirian-airlines-flight-2937-dhl-flight-611:mid-air-c

Bass, E., Feigh, K., Gunter, E., Rushby, J., Mansky, W. 2010a. "Core features of Agent-Based Language," NextGenAA Project Report, November 15.

Bass, E., Feigh, K., Gunter, E., Rushby, J., Kannan, S., Lee, G., Lee, D. W. 2010b. "Requirements for Analysis Toolset," NextGenAA Project Report, November 15.

Bass, E. J., Bolton, M. L., Feigh, K. M., Griffith, D., Gunter, E., Rushby, J., and Mansky, W. 2011a. Toward a Multi-method Approach to Formalizing Human-automation Interaction and Human-human Communications, *Conference on IEEE-Systems, Man, and Cybernetics*, October 9-12, 2011, Anchorage, Alaska.

Bass, E. J., Feigh, K. M., Gunter, E., and Rushby, J. 2011b. Formal Modeling and Analysis for Interactive Hybrid Systems, *4th International Workshop on Formal Methods for Interactive Systems (FMIS)*, June 21, 2011, Limerick, Ireland.

Benedict, Ruth, 1946. The Chrysanthemum and the Sword: Patterns of Japanese Culture, New York: Houghton Mifflin.

Bolton, M. L., Siminiceanu, R. I., and Bass, E. 2010, A Systematic Approach to Model Checking Human-Automation Interaction Using Task Analytic Models, *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*. 41(5), 961–976.

Brown, A. 2006. Accidents, Engineering, and History at NASA 1967-2003. In Dick & Launius, *Critical Issues in the History of Spaceflight* (Chapter 12), NASA SP 2006-4702.

BFU Report. 2004. Bundesstelle für Flugunfalluntersuchung *Investigation Report* German Federal Bureau of Aircraft Accidents Investigation. AX001-1-2/02, May.
Available at:
http://www.skybrary.aero/index.php/T154_/_B752,_enroute,_Uberlingen_Germany,_2002_(LOS_HF)

Burnett, C., Norman, T. J., and Sycara, K. 2006. Trust Decision-Making in Multi-Agent Systems, *Proc. Twenty-Second International Joint Conference on Artificial Intelligence*, 115-120.

Callantine, T. 2001. Agents for analysis and design of complex systems, *Proc. 2001 IEEE Intl. Conf. Systems, Man, and Cybernetics*, Tucson, pp. 567-573.

Callantine, T. 2005a. Performance evaluation of a computational model of en route air traffic control, *Proc. 13th Intl. Symp. Aviation Psychology*, Oklahoma City, OK, pp. 86-91.

Callantine, T. 2005b. Computational Modeling of Air Traffic Control: Terminal Area Case Study, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Kona, HI, 2249-2254.

de Carvalho, P. V. R. Gomes, J. O., Huber, G. J., Vidal, M. C. 2009. Normal people working in normal organizations with normal equipment: System safety and cognition in a mid-air collision, *Applied Ergonomics* 40, 325–340.

Casner, Stephen. 2007 [2001]. A Pilot's Guide to the Modern Airline Cockpit. Aviation Supplies & Academics, Inc.

Clancey, W. J., Richer, M., Wilkins, D. C., Barnhouse, S., Kapsner, C., Leserman, D., Macias, J., Merchant, A., and Rodolitz, N. 1986. GUIDON-DEBUG: The student as knowledge engineer. Stanford Knowledge Systems Laboratory Working paper No. 86-34.

Clancey, W. J. 1997. *Situated Cognition: On Human Knowledge and Computer Representations*. NY: Cambridge University Press.

Clancey, W. J. 1999. *Conceptual Coordination: How the Mind Orders Experience in Time*. Hillsdale, NJ: Lawrence Erlbaum.

Clancey, W. J. 2002. Simulating activities: Relating motives, deliberation, and attentive coordination. *Cognitive Systems Research*, 3(3) 471-499.

Clancey, W. J. 2005. Modeling the perceptual component of conceptual learning—a coordination perspective. In P. Gärdenfors and P. Johansson (Eds.), *Cognition, Education and Communication Technology*, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 109-146.

Clancey, W. J., Sachs, P., Sierhuis, M., and van Hoof, R. 1998a. Brahms: simulating practice for work systems design. *Int. J. Human-Computer Studies*, 49, 831-865.

Clancey, W. J., Torok, D., Sierhuis, M., van Hoof, R., and Sachs, P. 1998b. *Simulating work behavior*. US Patent No. US6216098. Available: http://www.google.com/patents/US6216098

Clancey, W.J, Sierhuis, M., Damer, B., Brodsky, B. 2005. Cognitive modeling of social behaviors. In R. Sun (Ed.), *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, pp. 151-184. New York: Cambridge University Press.

Clancey, W.J., Sierhuis, M., Seah, C., Buckley, C., Reynolds, F., Hall, T., Scott, M. 2008. Multi-agent simulation to implementation: A practical engineering methodology for designing space flight operations. In A. Artikis, G. O'Hare, K. Stathis, & G. Vouros (Eds.), *Engineering Societies in the Agents' World VIII*. Athens, Greece, October 2007. Lecture Notes in Computer Science Series, Volume 4870. Heidelberg Germany: Springer, pp. 108-123.

Clancey, W. J. and Lowry, M. 2012. Lunar Surface Systems Software Architecture Study: Interoperability. NASA/TP—2012–216040. Available: http://ti.arc.nasa.gov/publications/

Clancey, W. J., Nado, R., van Hoof, R., Sierhuis, M., Jones, G., Dvorak, D. 2012. Lunar Surface Systems Software Architecture Study: Open Architecture. NASA/TP—2012–216041. Available: http://ti.arc.nasa.gov/publications/

Columbia Accident Investigation Board. 2003. *CAIB Report, Volume 1*. NASA. Available: http://www.caib.us/news/report/volume1/default.html

Corker, K., Gore, B., Fleming, K., and Lane, J. 2000. Free flight and context of control: Experiments and modeling to determine the impact of distributed air-ground air traffic management on safety and procedures. *Proceedings of the 3rd USA/Europe Air Traffic Management Research and Development Seminar*, Naples, Italy.

Defense Science Board. 2012. Task Force Report: The Role of Autonomy in DoD Systems. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, D.C. July.

Dismukes, R. K., Loukopoulos, L., and Jobe K. K. 2001. The challenges of managing concurrent and deferred tasks, *In R. Jensen (Ed.), Proceedings of the 11th International Symposium on Aviation Psychology. Columbus, OH: Ohio State University*.

EUROCONTROL. 2009. ACAS Resolution Advisor (RA) Downlink Workshop Report. Berlin, November. http://www.eurocontrol.int/ra-downlink/public/standard_page/berlin_workshop_oct_09.html, or final report at http://www.eurocontrol.int/ra-downlink/gallery/content/public/library/berlin/ws_report(final).pdf

FAA. 2011. Introduction to TCAS II Version 7.1. US Department of Transportation, Federal Aviation Administration. February 28. Report No. HQ111358.

FAA. 2013. "NextGen," Federal Aviation Administration website. Available: http://www.faa.gov/nextgen/

Feltovich, P. J., Bradshaw, J. M., Clancey, W. J., Johnson, M., & Bunch, L. 2008. Progress Appraisal as a Challenging Element in Human and Machine Joint Activity. In A. Artikis, G. O'Hare, K. Stathis, & G. Vouros (Eds.), *Engineering Societies in the Agents' World VIII*. Athens, Greece, October 2007. Lecture Notes in Computer Science Series, Volume 4870. Heidelberg Germany: Springer, pp. 124-141.

Freed, M. A. 1998. *Simulating Human Performance in Complex, Dynamic Environments*, Computer Science PhD Dissertation, Northwestern University, Evanston, IL.

Gehman, H. W., et al. 2003. Columbia Accident Investigation Board Report, August. NASA. Available: http://www.caib.us/news/report/volume1/default.html

Heinrich, H. W. 1959. *Industrial Accident Prevention*. 4th Edition. New York: McGraw Hill.

Heinrich, H.W., Petersen, D., & Roos, N. (1980). *Industrial accident prevention: A safety management approach* (5th ed.). New York: McGraw-Hill.

Ho, D. and Burns, C. M. 2003. Ecological Interface Design in Aviation Domains: Work Domain Analysis of Automated. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 47(1) 119-123.

Hollnagel, E. 2004. *Barriers and Accident Prevention*. Ashgate, Aldershot, UK.

Hunter, J., Raimondi, F., Rungta, N., and Stocker, R. in preparation. A Synergistic and Extensible Framework for Multi-Agent System Verification, submitted to *12th International Conference on Autonomous Agents and Multiagent Systems*.

Hutchins, E. 1995. How a Cockpit Remembers Its Speeds. *Cognitive Science,* 19, 265-288.

Hutchins, E. 2000. The cognitive consequences of patterns of information flow. *Intellectica*, 30, pp. 53-74.

Johnson, C. W. 2004a. "Have We Learned Enough from Überlingen:The Challenges of Safety Improvement in European Air Traffic Management," Available: http://www.dcs.gla.ac.uk/~johnson/papers/EUROCONTROL_RD_Ueberlingen.pdf

Johnson, C. W. 2004b. "Final Report: RFQ/137/04 Review of the BFU's Überlingen Accident Report," Available: http://www.dcs.gla.ac.uk/~johnson/Eurocontrol/Ueberlingen/Ueberlingen_Final_Report.PDF

JPDO. 2013. "NextGen Overview," The Joint Planning and Development Office website. Available: http://www.jpdo.gov/About_Us.asp

Kim, So Young. 2011. *Model-Based Metrics of Human-Automation, Function Allocation in Complex Work Environments*. PhD Dissertation. Georgia Institute of Technology, August.

Kintsch, W., Miller, J. R. and Polson, P. 1984. *Method and Tactics in Cognitive Science*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kochenderfer, M. J., Holland, J. E., Chryssanthacopoulos, J. P. 2012a. Next Generation Airborne Collision Avoidance System, *Lincoln Laboratory Journal*, 19(1) 55-71.

Kochenderfer, M. J., Chryssanthacopoulos, J. P., Weibel, R. E. 2012b. A New Approach for Designing Safer Collision Avoidance Systems, *Air Traffic Control Quarterly*, 20 (1) 27-45.

Kuchar J. K. and Drumm A. C. 2007. The Traffic Alert and Collision Avoidance System, *Lincoln Laboratory Journal*, Volume 16, Issue 2, pp. 277-295.

Labov, William. 1972. "The Transformation of Experience in Narrative Syntax," *Language in the Inner City*. Philadelphia: University of Pennsylvania Press, pp. 354-396.

Ladkin, Peter B. 2004. "A comment on ACAS and Überlingen," Thu 04 Mar 2004 Available: http://www.cs.york.ac.uk/hise/safety-critical-archive/2004/0103.html

Learmont, D. 2011. "Industry sounds warning on airline pilot skills." *Flight Global*. 6 February. Available: http://www.flightglobal.com/news/articles/industry-sounds-warnings-on-airline-pilot-skills-352727/

Leuchter, S. & Jürgensohn, T. 2000. A tutoring system for air traffic control on the basis of a cognitive model. In J. L. Alty (Ed.), *Proceedings of the XVIII. European Annual Conference on Human Decision Making and Manual Control*. Loughborough: Group D, pp. 275–281. http://www.safety-critical.de/doc/eam.pdf

Leveson, N. 2004. A New Accident Model for Engineering Safer Systems. *Safety Science*, 42(4) 237- 270.

Linde, Charlotte. 1993. *Life Stories: The Creation of Coherence*. Oxford University Press.

Livadas, C., Lygeros, J., and Lynch, N. 2000. High-level modeling and analysis of the traffic alert and collision avoidance system (TCAS), *Proceedings of the IEEE*, vol. 88, pp. 926–948, July 2000.

Maiden, N., Kamdar, N., Bush, D. 2006. Analysing *i** System Models for Dependability Properties: The Uberlingen Accident, The *Twelfth Working Conference on Requirements Engineering: Foundation for Software Quality*. Luxembourg, June.

Merlin, P. W., Bendrick, G. A., and Holland, D. A. 2012. *Breaking the Mishap Chain: Human Factors Lessons Learned from Aerospace Accidents and Incidents in Research, Flight Test, and Development*, NASA eBook, Available: http://www.nasa.gov/connect/ebooks/break_mishap_chain_detail.html

Mindell, D. A. and Mirmalek, Z. L. 2011. An ethnographic approach to human-machine relationships in commercial aviation: Heads-up guidance and enhanced vision, *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* AIAA 2011-966, 4 - 7 January, Orlando, Florida.

Nunes, A. and Laursen, T. 2004. Identifying the factors that led to the Uberlingen mid-air collision: implications for overall system safety, *Proceedings of the 48th Annual Chapter Meeting of the Human Factors and Ergonomics Society*, September 20 - 24, 2004, New Orleans, LA, USA.

Qureshi, Zahid H. 2007. A Review of Accident Modelling Approaches for Complex Socio-Technical Systems. *Australian Computer Society, Inc. 12th Australian Workshop on Safety Related Programmable Systems*.

Perrow, C. 1999 (1984). *Normal Accidents: Living with High-Risk Technologies*. Princeton, NJ: Princeton University Press.

Pompanon, C., and Raufaste, E. 2009. The Intervention Trigger Model: Computational Modelling of Air Traffic Control. In N. Taatgen and H. van Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 2262-2267), Amsterdam, July 29-August 1.

Pritchett A. R. 2011. "Work modeling paper." Unpublished manuscript.

Pritchett A. R. and Feigh, K. M. 2011. Simulating first-principles models of situated human performance, *Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*. Miami Beach, FL: IEEE, pp. 144-151.

Pritchett, A. R., Kim, S. Y., Kannan, S., and Feigh, K. 2011. Simulating situated work, *Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*. Miami Beach, FL: IEEE, pp. 66-73.

Pritchett, A. R. 2012. The Pilot As An Expert Fallible Machine During Traffic Events, *HCI-Aero International Conference on HCI in Aeronautics*, EUROCONTROL, Brussels, September.

Rasmussen, J. (1997): Risk Management in a Dynamic Society: A Modelling Problem. *Safety Science,* 27(2/3):183-213.

Reason, J. 1990. *Human Error*. Cambridge University Press.

Riley, V., Garg, C., and Adams, J. 1996. Analyzing the Dynamics of a Next Generation Air Transportation Management System. Honeywell Technology Center, Final Report, NASA Contract NAS2-14288.

Rungta, Neha, Brat, G., Clancey, W. J., Linde, C, Raimondi, F., Seah, C., and Shafto, M. 2013. Aviation Safety: Modeling and Analyzing Complex Interactions between Humans and Automated Systems. *Proceedings of the 3rd International Conference on Application and Theory of Automation in Command and Control Systems*. Naples, Italy.

Rushby, J. 2011. New Challenges In Certification For Aircraft Software. *EMSOFT'11,* October 9–14, Taipei, Taiwan.

Ryle, G. 1949. *The concept of mind*. New York: Barnes & Noble.

Scott, A. C., Clancey, W., Davis, R., and Shortliffe, E.H. 1977. Explanation capabilities of knowledge-based production systems, *American Journal of Computational Linguistics*, microfiche 62. Also in B. Buchanan and E.H. Shortliffe (Eds.), *Rule-Based Expert Systems*, (Reading, MA: Addison-Wesley, 1984), pp. 338-362.

Seah, C., Sierhuis, M., and Clancey W. J. 2005. Multi-agent modeling and simulation approach for design and analysis of MER mission operations. *SIMCHI — Human-Computer Interface Advances For Modeling And Simulation*, January, pp. 73-78.

Shappell S. A. and Wiegmann, D. A. 2000. "The Human Factors Analysis and Classification System (HFACS)," Report Number DOT/FAA/AM-00/7 (Washington, DC: FAA, Office of Aviation Medicine)

Sierhuis, M., Diegelman, T. E., Seah, C., Shalin, V., Clancey, W. J., Selvin, A. M. 2007. Agent-based Simulation of Shuttle Mission Operations. *Agent-Directed Simulation, Spring Simulation Multiconference* (SpringSim), pp. 53-60. Norfolk, VA, March.

Sierhuis, M., Clancey, W. J., and van Hoof, R. J. J. 2009. Brahms: An Agent-Oriented Language for Work Practice Simulation and Multi-Agent Systems Development. In M. D. Rafael H. Bordini, Jürgen Dix, Amal El Fallah-Seghrouchni (ed.) *Multi-Agent Programming,* 2nd Edition: Springer.

Suchy, N. 2007. TCAS Reversal Logic Error. FAA Office of Technology Development. Presented at *Armed Forces Communications and Electronics Association (AFCEA) CNS/ATM Conference*, April 25.

Sutton, Oliver. 2012. "ACAS: flight path to catastrophe: one year after Uberlingen, we look at the importance of human factors, especially in high-stress situations. What can be done to further reduce the risk of mid-air collision?" *Interavia Business & Technology*. FindArticles.com. 05 Jan.

SSAT. 2011. *System-Wide Safety and Assurance Technologies (SSAT) R&T Portfolio Project Plan*. Aviation Safety Program (AvSP), Aeronautics Research Mission Directorate (ARMD), 1 October.

TCAS. 2012. "Traffic Collision Avoidance System," http://en.wikipedia.org/wiki/TCAS (accessed 23 July 2012).

The Observer. 2002. "Losing Control of the Skies," 6 July 2002, http://www.guardian.co.uk/business/2002/jul/07/theairlineindustry.observerfocus

The Telegraph. 2002. "Air crash warning system switched off," 3 July, http://www.telegraph.co.uk/news/1399137/Air-crash-warning-system-switched-off.html.

Tufte, E. 2006. *Beautiful Evidence*. Cheshire, CT: Graphics Press.

Vaughn, D. 1996. *The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA*. Chicago: University of Chicago Press.

Vaughn, D. 2003. David Karp Interviews Diane Vaughan Discussions of the Challenger and Columbia (edited by Jean Lovett), *Sociology Speaks—The Newsletter of the Sociology Department of Boston College*, issue 2002-2003, pp. 16-19.

Wallace, B. and Ross, A. 2006. *Beyond Human Error: Taxonomies and Safety Science*. Boca Rotan, FL: Taylor & Francis.

Weber, Max 1999. *Sociological Writings*. Edited by Wolf Heydebrand. New York: Continuum.

Wickens, C. D., Mavor, A. S., Parasuraman, R., and McGee, J. P., editors. 1998. *The Future of Air Traffic Control: Human Operators and Automation*. Panel on Human Factors in Air Traffic Control Automation, National Research Council. Available: http://www.nap.edu/catalog.php?record_id=6018

Wolfe, S., Sierhuis, M., and Jarvis, P. 2008. To BDI, or Not to BDI: Design Choices in an Agent-Based Traffic Flow Management Simulation, *Agent-Directed Simulation Symposium*, SpringSim Multi-Conference, Ottawa, ACM.

Woods, D. D. 2005. Creating Foresight: Lessons for Resilience from Columbia. In W. H. Starbuck, and M. Farjoun (Eds.), *Organization at the Limit: NASA and the Columbia Disaster*, pp. 289–308. Oxford: Blackwell.

15 Glossary

A&A	Authority and Autonomy
Activity	In Brahms, a located behavior of agent or object, which takes time. A Composite Activity consists of a prioritized set of workframes, such that the activity may involve a combination of perceiving and changing the state of objects, inferring new beliefs, communicating (asking or telling beliefs), and moving.
AFCS	Assurance for Flight-Critical Systems, task within the Aviation Safety Program (AvSP) of the System-Wide Safety and Assurance Technologies (SSAT) Project of NASA's Aeronautics Research Mission Directorate (ARMD)
Agent	In Brahms, a proactive, located process that models the world and carries out activities; may represent a person, a goal-oriented robotic system, or an "intelligent" software program
AirRadio	Aeronautical Radio & Air Traffic Control Advisors
ANSA	International Advisory Group Air Navigation Services
AP	Autopilot
ARMD	Aeronautics Research Mission Directorate

ATC	Air Traffic Control
ATCC	Air Traffic Control Center
ATCO	Air Traffic Controller. Acronym used in the European Union. In the United States, the FAA uses the term "controller".
ATS	Air Traffic System
Authority	having the right, or power, to exercise controls or issue air traffic commands that impact the position, velocity, and/or attitude of aircraft during operations.
Autonomy (or automation)	a function or system that can operate independently of pilot or air traffic controller intervention.
AvSP	Aviation Safety Program
Belief	In Brahms, a proposition about the world, constituting part of an agent's or object's model of the world
BFU	Bundesstelle für Flugunfalluntersuchung: German Federal Bureau of Aircraft Accident Investigation
Brahms	Multiagent representational framework for modeling and simulating work practices of people in modeled distributed geographic settings, interacting with other people and objects over time.
Brahms-GÜM	Generalized Überlingen Model represented in Brahms language
CA	Controller Assistant for the ATCO
CDU	Control Display Unit
CPA	Closest Point to Approach
Detectable	In BRAHMS, perceived fact in the world
DL	Supervisor at ACC Zurich (German: Dienstleiter)
Eurocontrol	European Organisation for the Safety of Air Navigation: Counterpart of the United States Federal Aviation Administration (FAA)
Fact	In Brahms, a proposition about an agent, object, or geography that from modeler's perspective is objectively true (contrast with belief)
Flight-critical System	Any system required to ensure the safe conduct of an aircraft flight. Includes air, ground, and space systems; and recognizes that human performance is central to flight-critical system performance.
FSM	Formal Semantic Methods
Geography	In Brahms, a model of the places (areas and subareas) of the simulated world (e.g., buildings, workstation area, cockpit) and paths for moving from place to place
Group	In Brahms, a class consisting of agents, representing any common set of activities and initial facts and beliefs (i.e., true when simulation run begins)
GÜM	Generalized Überlingen Model
HFACS	Human Factors Analysis and Classification System
IAC	Interstate Aviation Committee: executive body overseeing the use

	and management of civil aviation in the Russian Federation
ICAO	International Civil Aviation Organization
Life cycle	The series of phases of a system including conceptualization, design, development, maintenance, upgrading, and retirement
Multiagent System	Computer program of parallel processes consisting of agents and objects, generally distributed in a simulated or real world environment
NAT	Normal Accident Theory
ND	Navigation Display
NextGen ATS	Next Generation Air Transportation System
Object	In Brahms, any located entity having factual properties (e.g., size) and optionally behaviors and changing states (e.g., an aircraft)
PDF	Primary Flight Display
PSR	Primary Surveillance Radar
RA	Resolution Advisory: TCAS command to pilot
Sectorisation	Virtual division of airspace
skyguide	Private air traffic control service for Swiss airspace and adjoining airspace areas in Germany, Austria, France and Italy that have been delegated to its control. (Officially written in lower case.)
SMOP	Single Man Operations Plan
SSAT	System-Wide Safety and Assurance Technologies Program
STAR	Standard Terminal Arrival Route. STARs are procedures listing arrival waypoints, altitudes, speeds, etc. for each airport runway
STCA	Short-Term Collision Avoidance system (provides optical and audible alerts for ATCO)
SYCO	Flight Plan Processing System
SYMA	Systems Manager
TA	Traffic Advisory: TCAS warning to pilot
TAU	Time to Collision
Tau value	Number of seconds until lateral or vertical collision
TCAS	Traffic Collision Avoidance System (includes display and voice commands to pilots)
Thoughtframe (TF)	An inference rule that concludes new beliefs on the basis of existing beliefs
TSAFE	Terminal Tactical Separation Assured Flight Environment
UTC	Coordinated Universal Time. Equivalent to GMC: Greenwich Mean Time
V&V	Verification and Validation
Validation	Confirmation that proposed system requirements, and/ or operational systems, meet the expectations of the customer and other stakeholders, accomplishing the intended purpose in the intended environment(s), throughout the system's life cycle
Verification	Confirmation that proposed or operational systems comply with requirements throughout the system's life cycle

VOR	Very High Frequency Omni directional Radio Range: ground radio beacon
VSI	Vertical Speed Indicator
Waypoint	Set of coordinates for navigation, identifying a point in physical space, including latitude, longitude and altitude
WMC	Work Model that Computes
Work system	An interacting configuration of hardware, software, people, facilities, and procedures organized for some purpose, which is accomplished in its operations
Workframe (WF)	A situation-action rule that causes an agent to carry out activities when certain conditions based on agent beliefs are true; either "top-level" or part of a Composite Activity of the agent. (For objects these are called Factframes.)

16 Appendix: Key Events in Überlingen Collision

This is an excerpt from the BFU Report (pp. 108-109) listing the key events that contributed to the collision.

Air traffic control (ATC)

- Sectorisation work was carried out within ACC Zurich in order to re-arrange the control sectors in the night from 1 to 2 July 2002. During this time the radar system was operated in the "fallback mode" and the separation minimum had been increased from 5 to 7 NM. In doing so the MV9800 radar computer was not available to the controllers, therefore
    - no automatic correlation of the flight targets was possible and
    - the optical STCA was not displayed anymore.
- The direct phone connections with the adjacent ATCO units were not available to the controller of ACC Zurich during the time from 21:23 hrs until 21:34:37 hrs. An automatic change-over of incoming calls to the bypass system was not in existence. At 21:34:44 hrs the first of a total of four calls, three calls from UAC Karlsruhe and one call from Friedrichshafen, was registered. These calls had not been answered.
- There were written directives concerning the accomplishment of the work, however, they did not include explanations about the effects the work would have on the availability of technical equipment.
- The CoC did not know about the sectorisation work. An assessment to minimise risks did not take place.
- Besides the technicians three additional colleagues were present in the CIR.
    - One of the managers to support the ATCO
    - One SYMA
    - One controller to support the technicians The ATCO did not know about the tasks of these colleagues.
- The sectorisation work had not been coordinated with the adjacent ATCO units.
- According to the duty schedule, two controllers were responsible for the control of the en- tire airspace of ACC Zurich during the night shift. They had to assume the tasks of radar planning (RP), radar executive (RE) and to a limited extent also the functions of the super- visor (DL) and the system manager. Therefore, a continuous management of the different tasks was not ensured. An assessment to minimise risks during the night shift did not take place.
- The controllers were obliged to read the directives concerning the accomplishment of the system work. But they did not read them. The supervisor (DL) had merely given them general information about the work.
- Two assistants were at the disposal of the controllers to support them with routine and co- ordination tasks, however, they had no authorization to assume any traffic control functions.
- After the air traffic flow had decreased one controller retired to rest at about 21:15 hrs and approximately 10 minutes later one assistant retired to rest. Normally they would not return to the control room until early in the morning.

- It had been known to and tolerated by the management and the quality assurance of the air navigation service company for years that during the night at periods of low traffic flow only one controller performed all traffic control tasks whereas the other controller had a rest.
- Both controllers were qualified and licensed in accordance with the regulations in force.
- The controller remaining in the control room was examined after the accident for medicine, drugs and alcohol which produced negative result.
- At the time of the accident the controller had to control three airplanes:
    - the B757-200 in direct approach to Tango VOR at FL 360
    - the TU154M in direct approach to Trasadingen VOR at FL 360 and
    - a delayed Airbus A320 approaching Friedrichshafen.

  The Airbus was controlled on 119.920 MHz and the two other airplanes on 128.050 MHz. Therefore they could not hear each other which resulted in simultaneous transmissions. For all flights the control strips were available to the controller in time. From the control strips the impending conflict situation (B757-200 and TU154M) was only recognisable in combination with the radar display.
- The controller was solely responsible for the entire ATCO within ACC Zurich. For this he had to fill two adjacent workstations with different frequencies and worked with two radar monitors. In order to control flights in the upper airspace and the approach in the lower airspace to Friedrichshafen. Radar charts with different ranges were displayed on the monitors.
- The controller was not aware that in the fallback mode the optical STCA was not available. The system did not provide an automatic indication that the optical STCA was not available.
- During the last five minutes prior to the collision, the controller paid more attention to the Airbus A320 in approach to Friedrichshafen.
- The bypass telephone system had temporarily a technical defect so that the necessary co- ordination with Friedrichshafen could not take place by phone.
- At 21:33:24 hrs the radar controller of UAC Karlsruhe was alerted by his STCA of the conflict situation. His attempts to warn the controller of ACC Zurich by phone were not successful as a telephone connection could not be established.
- The controller did not notice the imminent separation infringement in time. He instructed the TU154M crew at 21:34:49 hrs (43 seconds prior to the collision) to descend to FL 350 which was too late to ensure the required separation to the B757-200. The phraseology used did not correspond with the urgency of the situation.
- At 21:34:56 hrs the prescribed separation of 7 NM was infringed.

ACAS/TCAS

- The TU154M crew followed the ATCO instruction immediately and initiated the descent.
- At 21:34:56 hrs (35 seconds prior to the collision) TCAS generated RAs in both aircraft simultaneously.

    - The B757-200 crew received an RA to descend. The copilot was not in his seat at the time. The PIC followed the RA and initiated the descent.
    - The TU154M crew had already initiated the descent when they received the RA to climb. The RA did not change the decision and the descent was continued. This decision did not take into account that very likely simultaneously with this RA the other air- plane involved would receive a complementary RA.
- The copilot of the TU154M questioned the continuation of the descent twice. But he could not gain anybodies [sic] ear. A comment that TCAS has priority over ATCO did not come from any of the crew members.
- The B757-200 crew reported 23 seconds after the RA the "TCAS descent" to ACC Zurich. The copilot hat taken his seat again at that time and the frequency was free.
- According to his statement the controller did not notice the message of the B757-200 crew. The first part of the message was incomprehensible due to the simultaneous transmission of both crewmembers. The second part coincided with a message at the adjacent work- station (RE) transmitted by the A320.
- At 21:35:00 hrs the MV9800 computer of ACC Zurich released an aural STCA warning to the workstation of the controller. This warning had not been noticed in the control room.
- Once the controller noticed that the TU154M had initiated the descent he again turned to the A320 whose crew had already called him twice. He did not continue to observe the developing situation.
- An automatic downlink, integrated in the TCAS equipment, carrying information about issued RA's to the respective ATCO units has not been introduced worldwide yet. It was determined that with the prescribed reports via radio delays and loss of information may occur.
- The ACAS/TCAS related international regulations and national procedures valid on the day of the accident were not sufficiently clear or incomplete and misleading and did not fully correspond to the system philosophy.

17 Appendix: Überlingen Timeline

One analysis of BFU Investigation Report commented about the inadequacy of the timeline (Johnson 2004b). Most obviously, there is no common scale for time—the width of columns, which represent 1 second each, varies according to the text the designer wanted to fit in each column's cells (Figure 17-1).

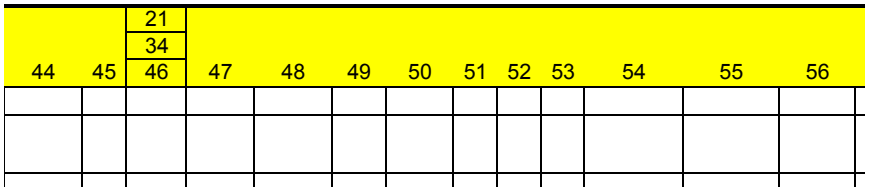


Figure 17-1: BFU Report Appendix 3 excerpt from "View of the Events" timeline. Each column represents 1 second but the columns have different widths.

Furthermore, the diagram is titled "the last minute" and represents the last 1 minute and 15 seconds, yet many important events leading to the accident occurred at least five minutes earlier (such as the attempts to call Friedrichshafen). The diagram is extremely helpful and represents a great deal of information, but omits ATCO's movements, so we must reconstruct what he saw and heard, an essential part of a work practice simulation.

Figure 17-2 represents our reconstruction of the seven minutes prior to the collision, emphasizing ATCO location when interacting with each of the flights. The chart is segmented into three sequential parts from top to bottom. Each segment indicates when ATCO was at right (top of segment) and left (bottom) workstations and when he interacted with certain flights there workstation. "Friedrichshafen" represents ATCO's attempts to call the tower on the telephone. Shading and numbers in cells indicate possible prioritization of simultaneous activities to explain what he was doing at each time. (1 is the active communication; 2, what will do next; 3, further delayed; 0, the communication will be handled immediately but caller told to wait.) Question mark indicates his location is uncertain. Gray shading designates two events that ATCO apparently did not perceive because he was at right workstation: 1) 21:32:38 DHL 611 appears on ARFA sector radar, 2) 23:35:20 DHL calls in "600 TCAS descent."

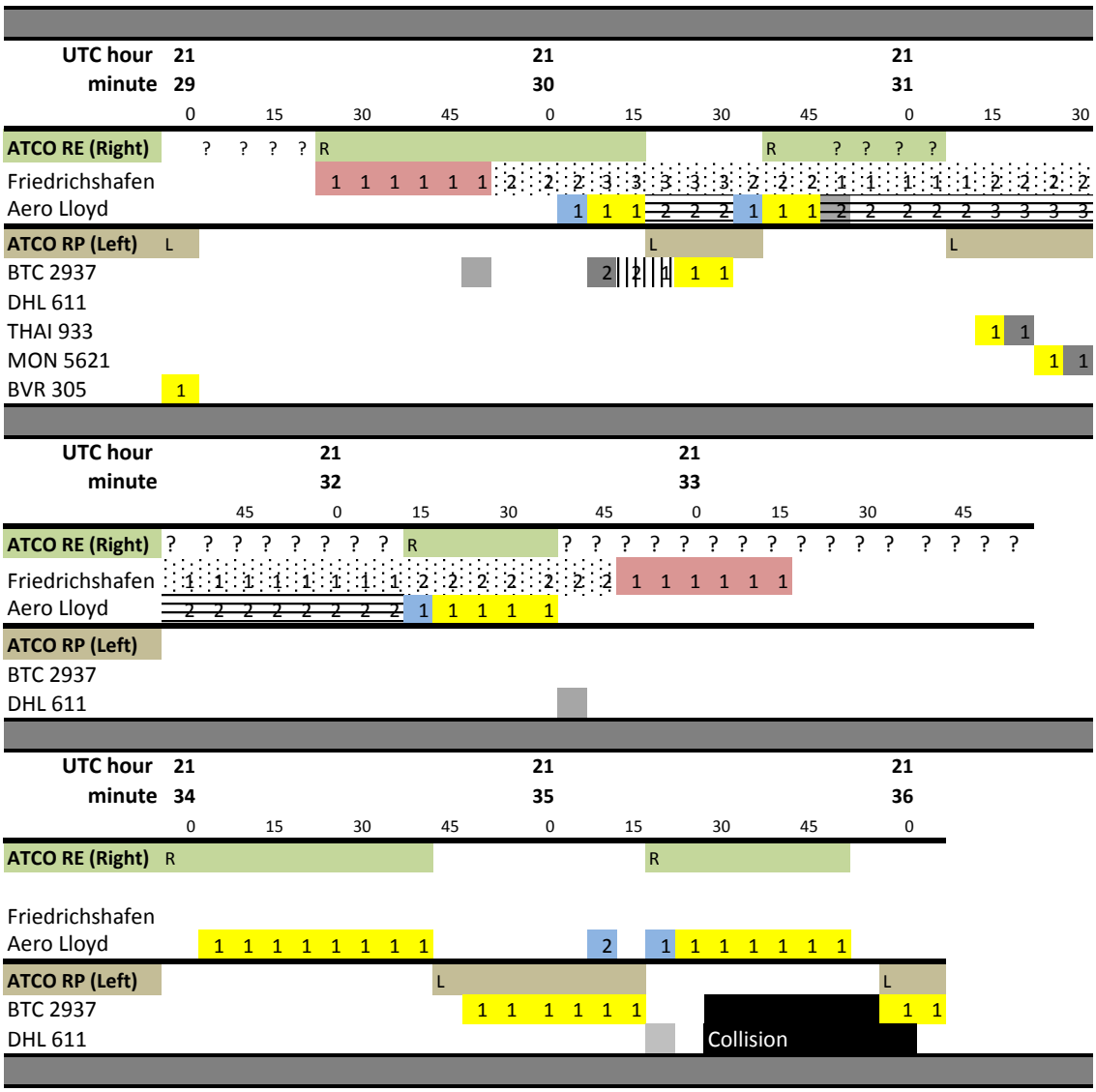


Figure 17-2: Timeline representing ATCO location when interacting with different flights during last seven minutes (see text).

This analysis would enable refining the ATCO model to development a prioritized "mental model" of active tasks (specifically here, handling flights in his sector). The current model uses WF priorities to effectively sort actions, such that handoff for an arriving flight has priority over monitoring the radar display, etc. (see Table 23-1 in Appendix 23). An alternative approach when an activity involves simultaneous equal-priority possible actions is for the agent to have an belief about "what is my top priority to do now," "what I need to do next" and "what will be handled as soon as practical," corresponding to the ranking 1-3 in the chart.

18 Appendix: Überlingen Unexplained Events and Behaviors

This appendix examines two key aspects of the accident that are mentioned in the BFU Report and Mayday video, but are not adequately documented or explained in reports, namely how the TCAS display influenced the BTC crew and why the Zurich ATCO did not notice that the DHL aircraft was descending.

- **Russian crew interaction prior and during TCAS TA/RA is complex and not correctly or completely presented by BFU timeline or the Mayday video (BFU Report, p. 8).**

    - BFU Investigation Report description and diagrams are not complete or consistent:
        - "At 21:34:42 hrs, TCAS generated a TA ("traffic, traffic"). The CVR recorded that both the PIC and the copilot called out "traffic, traffic" (p. 8)
        - "View of the Events" Appendix 3: shows Copilot calling out "traffic, traffic" at 21:34:48, but does not show PIC saying this.

    - BTC Crew is pre-occupied by discussing TCAS; they are not disregarding it; during this 1 min 41 sec (just prior to TA) they do not call into ATCO to inquire about the intruder; is this a common occurrence?

      For the time between about 21:33:00 hrs and 21:34:41 hrs the CVR recorded crew discussions concerning an airplane approaching from the left which was displayed on the vertical speed indicator (VSI/TRA) which is part of the TCAS. All flight crew members with the exception of the flight engineer were involved in these discussions." (BFU Report, p. 8)

    - Mayday video and BFU report imply that Russians received Blue TCAS but DHL did not
        - **Russian situation:**

      The TU154M crew noticed at 21:33:18 hrs for the first time on the TCAS display (VSI/TRA) another airplane approaching from the left. At this time the distance between the two airplanes was still approximately 27 NM. It is to be assumed that the display had been set to a range of 40 NM, as provided by the flight operations manual. There is no doubt that it was the B757-200 with which the TU154M later collided because there was no other airplane in this airspace at the time. (BFU Report, p. 70)

            - But the video shows the range is set at 16 NM when it is first depicted.
            - The copilot asked at 21:34:30 whether he should decrease the scale and the PIC replied "No."

      At 21:34:36 hrs - six seconds prior to the TA - the commander said that he had recognized the airplane at the same flight level. This finding is based on

the commander's statements: "Here visually" and two seconds later: "Here, it is showing us zero". This statement referred to the altitude difference indicated on the VSI/TRA display ("00" indication near the blue solid diamond). (BFU Report, p. 70)

- **DHL situation:** When Russian TCAS shows Blue TCAS (prior to TA), the video then says, "On the DHL plane, the crew is relaxed, they don't know they are on a collision course. TCAS hasn't sounded a warning yet…. The first officer goes to the washroom."

At 21:34:30 hrs the copilot handed over the control of the airplane to the PIC in order to go to the lavatory installed in a cubicle at the rear of the cockpit. At 21:34:31 hrs the PIC confirmed that he had taken over. (BFU Report, p. 7)

When the copilot as the PF at 21:34:30 hrs handed over the control of the airplane to the PIC in order to go to the lavatory, the distance between both airplanes was still 13.6 NM. The crew would have been able to recognise the other airplane on the VSI/TRA if a range of 16 NM was set. The setting, however, could not be determined. *It is probable that the crew had not noticed the other airplane at that time otherwise the copilot would not have left his seat.* (BFU Report, p. 68, emphasis added)

- The report justifies the DHL pilots not noticing the TCAS display:

It is not part of the prescribed and practised procedures to constantly observe the TCAS display on the respective instruments. According to the system philosophy, the attention of the crew is drawn to a potential conflict by the acoustical annunciation of a TA or an RA. (p. 68)

- Further the BFU report states that the Russians were wrong to be attending to the TCAS display prior to the TA:

Note: The BFU is of the opinion that the use of the TCAS display for the development of an own assessment of the air traffic situation and/or of a situation awareness is not compatible with the TCAS system philosophy or the system ATC. There are also no procedures in existence in order to use the system in such a way. (p. 99-100)

- The report implies that despite the TCAS "philosophy" and lack of procedures, the fact the TCAS display was visible and influenced the BTC crew's interpretation of ATCO's intervention:

The crew did recognise the other airplane on the VSI/TRA early and was able to identify it visually before the accident. That way they had become aware of a possible conflict which then seemed to be solved through the instruction of ACC Zurich. (p. 99)

- **Why doesn't the Zurich ATCO see that the DHL is also descending?**

- Why didn't he see that DHL was descending very close to BTC, if he was looking at the BTC datablock at 21:35:12 (BFU p. 89). Why doesn't he inform DHL about traffic? What was Zurich ATCO doing between 21:35:17 and 21:35:25?

- Zurich ATCO obviously he sees the DHL on his radar by 21:34:49 or he wouldn't have issued the urgent descend instruction to the Russians.
  - The DHL had been descending since 21:34:57, two seconds after the Russians.
  - It is claimed (BFU Report, p. 89) that "He would not have been able to recognise the B757-200 was descending until the screen update at 21:35:12 hrs."
  - He was at the left workstation, just finishing his remark to the Russians to expedite, and then listening to them respond from 21:35:13-17. So he was presumably staring at the screen exactly when the DHL (B757) update occurred at 21:35:12.
  - He repeated his command to the Russians, "expedite descent" at 21:35:07, so presumably he is still seeing both planes on a collision course; that is, he was seeing DHL on the radar display.
  - He confirmed the TU154M was descending based on his testimony (p. 85). This confirmation had to be at 21:35:12 because 1) he issued the expedite just prior, and the previous screen update occurred at 21:35:00 only 4 seconds after the AP disengage, too soon to show the descent.
  - *If he is able to confirm that the BTC is descending based on its data label (p. 85) why doesn't he also see that the DHL is descending, given that he sees both planes and they are near each other on the screen?*

- As Figure 18-1 shows, when ATCO was talking to BTC at left workstation, AEF calls in (heard on right); it is possible he detected this and wanted to respond to them; how could this be more important than confirming that collision was resolved (or did he surmise it was out of his control)?

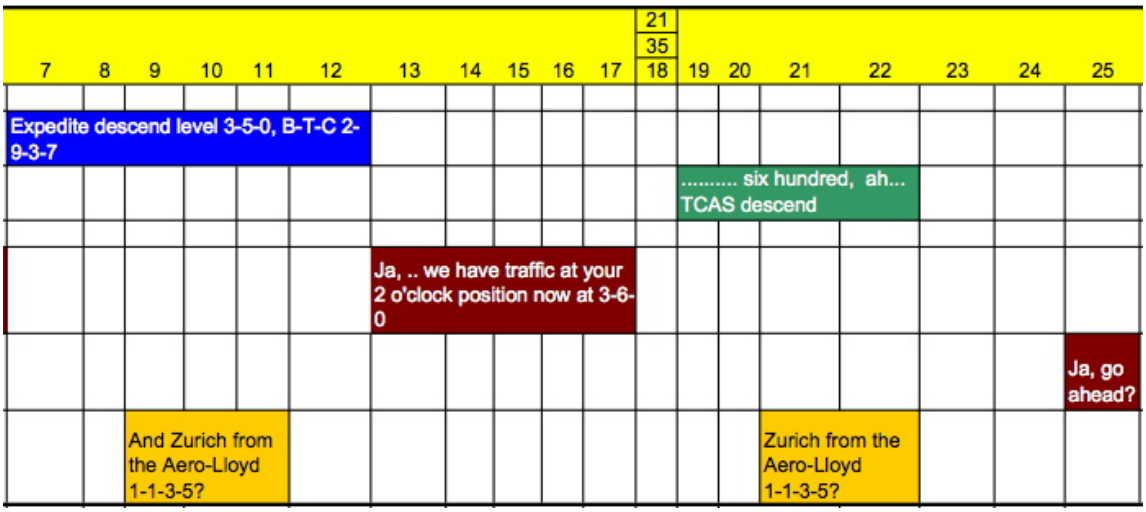


Figure 18-1: Excerpt of last minute timeline (BFU Report, Appendix 3).

19 Appendix: TCAS II Version 7.1 CP112E Reversal Logic

The DHL and BTC planes were both equipped with TCAS II 7.0 which included a form of "reversal logic" (BFU, p. 49).

> TCAS II, Version 7 is capable of generating a Reversal RA, i.e. a coordinated RA into a direction contrary to the initial RA. The Reversal is a way out, if during the avoidance manoeuvre an inversion of the original geometrical situation of the flight paths occurred. This situation will arise in particular if the crews respond contrary to the initial RA. A Reversal RA can be issued if the following conditions are fulfilled:
>
> - The calculated distance at the CPA must be greater in the new direction than in the initial direction and must be greater than 100 ft.
> - The altitude difference between the airplanes must have already exceeded 100 ft in the new direction.
> - A reversal may be generated not earlier than 9 s after the initial RA.
> - Up to the calculated moment of collision a period of at least 4 s must be left.
>
> These conditions within the algorithm for the calculation of the Reversal have been introduced in order to preclude frequent reversals of TCAS avoidance manoeuvres. This is necessary in order to maintain the trustworthiness of TCAS.
> During the descent of the B757-200 and the TU154M a Reversal RA was not generated, because conditions for an RA were not given.

The BFU Report does not specify which conditions for a reversal RA did not hold.

Kuchar and Drumm (2007, p. 284) provide a more complete explanation. They begin by citing a condition omitted from the BFU Report:

> First, a reversal will be triggered only by the aircraft with priority—that is, the aircraft with the lower Mode S address. If the aircraft has a higher Mode S address than the intruder, the RA sense will be reversed only when directed to do so by the priority aircraft through the data link.

In the configuration at Überlingen (Figure), the Russian aircraft had a lower Mode S address (effectively a serial number that distinguishes each aircraft with TCAS), so it was responsible for triggering a reversal. A climb-RA was in process and the internal TCAS "template predicted adequate separation between aircraft, at least until the final few seconds; therefore, TCAS did not issue an RA reversal." The TCAS algorithm did not take into account that the TU154M was not actually climbing. Yet even if the DHL aircraft had priority to generate the reversal, TCAS II 7.0 would not have generated a reversal because "both aircraft remained within 100 ft vertically of each other throughout the encounter" (p. 285).

The revised TCAS II 7.1 reversal algorithm (Change Proposal 112E) is specifically designed to detect one plane disregarding TCAS, in which case it instructs the complying pilot to reverse (and says nothing to the non-complying pilot; see figures below).

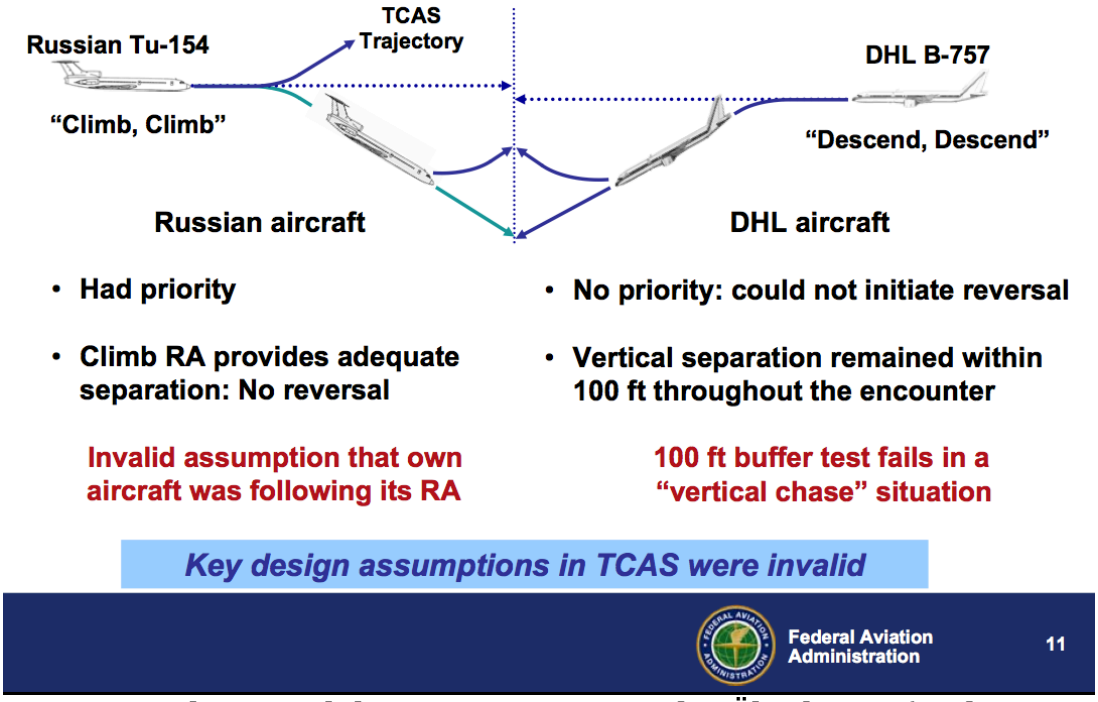


Figure 19-1. Why TCAS did not generate reversal at Überlingen (Suchy 2007, p. 11)

Apparently following Überlingen, there were a series of meetings, with a working committee report in 2008, and review process that completed in 2010 (to deal with what was understood to be an "urgent" design problem).[57]

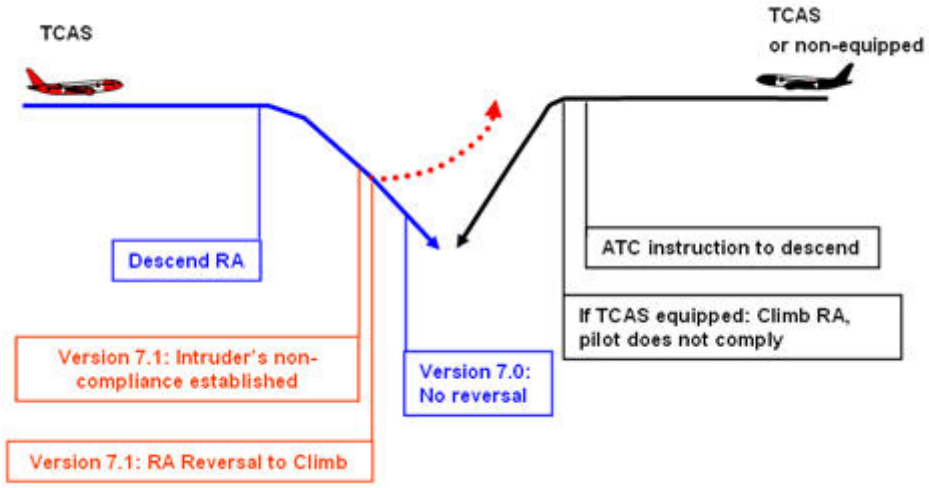
> Following a number of comments from airlines, EASA has proposed the following dates for the TCAS II version 7.1 mandate in European airspace: forward fit (new aircraft) 1 March 2012, retrofit (existing aircraft) 1 December 2015. These dates are proposed dates, subject to further regulatory processes and are not final until the Implementing Rule has been published.
>
> Although TCAS II is an aircraft system, it has been implemented to improve ATM safety. Studies conducted for EUROCONTROL, using recently recorded operational data, indicate that *currently* ***the probability of a mid-air collision in European airspace is 2.7 x 10-8 which equates to one in every 3 years.*** *When TCAS II version 7.1 is implemented that probability will reduce by a factor of 4.* (emphasis added)
>
> ….
>
> The issues with the reversal logic were resolved through a significant code change for TCAS II logic version 7.0. This change is known as Change Proposal 112E (CP112E).

[57] This information and following excerpts are from http://www.eurocontrol.int/msa/public/standard_page/ACAS_Upcoming_Changes.html

CP112E brings improvements to the reversal logic of TCAS II logic version 7.0 by detecting geometries close to that of the 2002 Überlingen mid-air collision, and by easing the triggering thresholds of reversal RAs in encounters in which the aircraft remain vertically within 100 ft of each other. The basic principle is to detect that two aircraft are climbing, or descending simultaneously. Two mechanisms are used to ensure that reversal RAs are triggered when necessary.

CP112E first adds a feature which monitors RA compliance. When it is detected after a certain period of time that an aircraft is not responding correctly to an RA, it circumvents the "100 ft box" rule, allowing reversal RAs for aircraft closer than 100 ft vertically.

CP112E also adds a prediction of the vertical separation at the closest point of approach, based on current vertical speeds, to detect the need for a reversal RA. Indeed, when this prediction shows that the aircraft are probably going to be closer than a predefined threshold, reversal RAs are considered as a valid option for aircraft closer than 100 ft vertically.



Reversal RAs are not triggered too early in an encounter, to leave time for the initial RAs to be efficient before reversing. In addition, reversal RAs are not triggered too close to the predicted closest point of approach, to avoid useless reversal RAs.

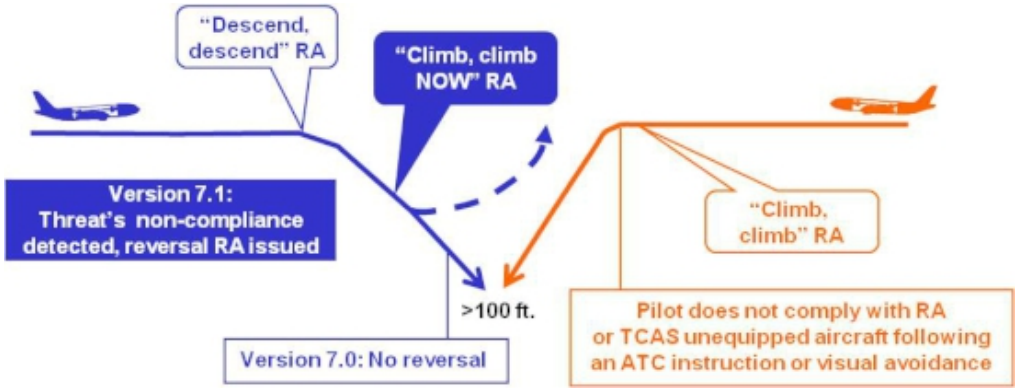
CP112E has been extensively validated in Europe by the EUROCONTROL-sponsored team composed of DSNA and Egis Avia experts, and by several organizations in the USA (MIT Lincoln Lab, MITRE, FAA, and Johns Hopkins University). This validation shows that CP112E significantly improves the safety brought by TCAS. It triggers reversal RAs in time in the geometries in which the issue with reversal RAs was identified. **In addition, side effects and performance deteriorations are minimal for CP112E and are considered acceptable compared to the collision risk with current versions of TCAS.**

A more precise characterization might be: **compared to the collision risk *given how people might interact* with current versions of TCAS.** The topic and analysis clearly focus on the automated system, rather than viewing the entire human-automation system as flying the plane and accounting for accidents.

Ladkin (2004) provides a helpful summary:

> The technical components of the ACAS system are purely advisory. They paint pretty pictures on a display and generate spoken sentences. They do not control the aircraft. The ACAS system depends for any effectiveness it may have exclusively on actions of the aircraft crew. The aircraft crew are thus an integral part of the ACAS system as system (not just the kit).

An apparently final or public version of the "Secondary Surveillance Radar Mode Select (SSR MODE S) and Airborne Collision Avoidance System (ACAS) Programme" (MAS) report has been posted online.[58] It does not mention Überlingen and provides a simpler diagram indicating the actual advisory text:



We note also these remarks from Eurocontrol's FAQs online page about ACAS:[59]

> TCAS (Traffic Alert and Collision Avoidance System) is a specific implementation of the ACAS (Airborne Collision voidance System) concept. TCAS II version 7.0 and 7.1 are currently the only available equipment that is fully compliant with the ACAS II Standards and Recommended Practices (SARPs).
>
> ACAS II provides "Resolution Advisories" (RA's) in the vertical sense (direction) telling the pilot how to regulate or adjust his vertical speed so as to avoid a collision.

and also this remark about the certification process:

> The change to the reversal logic has been evaluated using mathematical modelling….. John Lygeros, Carolos Livadas and Nancy Lynch have formally proved the correctness of the two-airplane TCAS avoidance algorithms [7]. (Ladkin 2004)

The history and development of TCAS suggest the following simulation and formal modeling challenges relevant to future research using Brahms-GÜM:

1. Using the Brahms simulation with the existing reversal algorithm

[58] http://www.eurocontrol.int/msa/public/standard_page/TCAS71.html
[59] http://www.eurocontrol.int/msa/public/faq/ACAS_FAQ.html - qa10

implemented, under what simple variations would it be enabled after all?
2. Using a Brahms simulation revised to incorporate this new algorithm, does the reversal algorithm now prevent collision? Note that Mitre/Honeywell testing could not have taken all of the factors of Überlingen into account.
3. Under what conditions will the new TCAS reversal algorithm fail?

20 Appendix: Proposed TCAS Resolution Advisory Downlink

Although it might appear from the Überlingen accident that it is essential for the sector ATCO to be automatically notified that TCAS has issued an RA (called "RA Downlink"), the issue is controversial and the FAA has no plans for implementing this.

A workshop was held in Berlin in October 2009 (EUROCONTROL 2009) [60] to review the pros and cons of the RA Downlink:

> Notification of ACAS II Resolution Advisories (RA) to controllers as they occur has been contemplated for many years. In Europe the Überlingen mid-air collision gave additional impetus for a number of organisations to implement what usually is referred to as RA Downlink. With the increasing operational use of Mode S, at least one enabling technology is readily available in a number of States. To avoid proliferation of concepts of use, it is now urgently needed to find common ground for use of RA Downlink in Europe.

A latency study predicted that in 95% of the cases, the ATCO would be informed of the RA using Mode S within 10 seconds, taking all technical and human factors into account.

A previous study (PASS) by EUROCONTROL reported:

> An analysis of pilots' TCAS RA reports to ATC was performed. About 50% of "Climb/Descend" RAs are reported, whilst reporting of other RAs is about 20%. Preliminary findings also show that the lack of report by crews receiving a coordinated "Climb/Descend" RA is often associated to a pilot report by the threat aircraft.
>
> Finally lack of report is sometimes associated to pilots not following the RA, while short duration RAs in general are often not reported.

The MIT Lincoln Laboratory analysis (EUROCONTROL 2009) cites the following:

[60] Presentations are posted:
http://www.eurocontrol.int/ra-downlink/public/standard_page/berlin_workshop_oct_09.html

- The frequency of RA encounters is significantly higher in the USA. This is in part explained by the higher traffic density and in part by the following points.
- The vast majority of RA encounters occur in uncontrolled airspace and reflect interactions between TCAS-equipped aircraft and VFR traffic or non-equipped business jets.
- Many of such interactions are caused by VFR traffic legally flying at 500 feet between IFR flight levels.

The FAA currently has no plans for RA Downlink display because many RAs do not require changes in flight path; pilot non-compliance with climb/descent RAs has the potential to cause confusion; and because the role of ATC in interactions between IFR and VFR traffic is unclear.

Fewer than 10% of RAs in the US are coordinated TCAS-TCAS between two aircraft; the rest are single aircraft advisories. Pilots do not comply with 38% of RA "climb" advisories. Most RAs involve only adjusting vertical speed and half involve VFR of general aviation "intruder."

The International Federation of Air Traffic Controllers Associations (IFATCA) is opposed to RA downlink.[61] They argue that any consideration of this capability must take into account:

- Clear and unambiguous legal responsibilities for controllers.
- No delay in downlink (e.g. due to antenna rotation).
- Displayed at the appropriate controller position(s).
- Must be fully compatible with ground safety nets (STCA, APW, etc.)
- Nuisance and false alerts must be kept to an absolute minimum.

Other presentations in 2009 similarly indicated that nuisance RAs would not help the ATCO and were a reason for not providing RA downlink and would add "another visual element" to screens that were already full of data. They are concerned that the RA downlink would be broadcast, rather than sent to the appropriate sector. Dehn et al.[62] provide a useful academic report that assesses "the operational affect of RA downlink."

In conclusion, key stakeholders believe that RA downlink could substantially disrupt ATCO work practices. Studies are required to:

- Design what is presented and how it is shown on the display (some say only "RA" not direction; others favor maximum info)
- Understand how the RA might interact with existing "safety nets" (e.g., STCA)

[61] Posted presentation: http://www.eurocontrol.int/ra-downlink/gallery/content/public/library/berlin/2-4 Philippe Domogala.pps
[62] http://reports.nlr.nl:8080/xmlui/bitstream/handle/10921/478/TP-2012-309.pdf?sequence=1

- Develop "global guidelines and procedures" for ATCOs regarding RA downlink
- Demonstrate that the RA downlink provides beneficial information, even with nuisance/false alerts.

21 Appendix: TCAS Protocol for ATCO and Pilot Decision Making

The material in this appendix is provided as background, excerpted from http://en.wikipedia.org/wiki/TCAS (accessed 23 July 2012), bold emphasis added.

TCAS II issues the following types of aural annunciations:

- Traffic advisory (TA)
- Resolution advisory (RA)
- Clear of conflict

When a TA is issued, pilots are instructed to initiate a visual search for the traffic causing the TA. If the traffic is visually acquired, pilots are instructed to maintain visual separation from the traffic. The pilot training programs also indicate that no horizontal maneuvers are to be made based solely on information shown on the traffic display. Slight adjustments in vertical speed while climbing or descending, or slight adjustments in airspeed while still complying with the ATCO clearance are acceptable.[4]

When an RA is issued, pilots are expected to respond immediately to the RA unless doing so would jeopardize the safe operation of the flight. This means that aircraft will at times have to manoeuver contrary to ATCO instructions or disregard ATCO instructions. In these cases, the controller is no longer responsible for separation of the aircraft involved in the RA until the conflict is terminated.

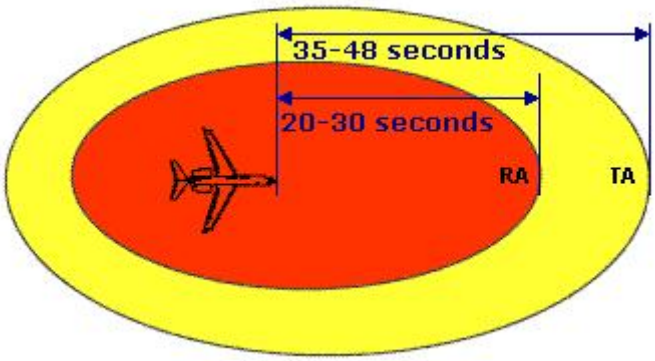
On the other hand, ATCO can potentially interfere with the pilot's response to RAs. **If a conflicting ATCO instruction coincides with an RA, the pilot may assume that ATCO is fully aware of the situation and is providing the better resolution. But in reality ATCO is not aware of the RA until the RA is reported by the pilot.** Once the RA is reported by the pilot, ATCO is required not to attempt to modify the flight path of the aircraft involved in the encounter. Hence, the pilot is expected to "follow the RA" but in practice this does not yet always happen.

Some States have implemented "RA downlink" which provides air traffic controllers with information about RAs posted in the cockpit obtained via Mode S radars. Currently, there are no ICAO provisions concerning the use of RA downlink by air traffic controllers.

The following points receive emphasis during pilot training:

- Do not manoeuver in a direction opposite to that indicated by the RA because this may result in a collision.
- Inform the controller of the RA as soon as permitted by flight crew workload after responding to the RA. There is no requirement to make this notification prior to initiating the RA response.

- Be alert for the removal of RAs or the weakening of RAs so that deviations from a cleared altitude are minimized.
- If possible, comply with the controller's clearance, e.g. turn to intercept an airway or localizer, at the same time as responding to an RA.
- When the RA event is completed, promptly return to the previous ATCO clearance or instruction or comply with a revised ATCO clearance or instruction.



TCAS II
After the Überlingen mid-air collision (July 1, 2002), studies have been made to improve TCAS II capabilities. Following extensive Eurocontrol input and pressure, a revised TCAS II Minimum Operational Performance Standards (MOPS) document has been jointly developed by RTCA (Special Committee SC-147) and EUROCAE….
TCAS II Version 7.1 will be able to issue RA reversals in coordinated encounters, in case one of the aircraft doesn't follow the original RA instructions (Change proposal CP112E). Other changes in this version are the replacement of the ambiguous *"Adjust Vertical Speed, Adjust"* RA with the *"Level off, Level off"* RA, to prevent improper response by the pilots (Change proposal CP115).; and the improved handling of corrective/preventive annunciation and removal of green arc display when a positive RA weakens solely due to an extreme low or high altitude condition (1000 feet AGL or below, or near the aircraft top ceiling) to prevent incorrect and possibly dangerous guidance to the pilot (Change proposal CP116).

Studies conducted for Eurocontrol, using recently recorded operational data, indicate that currently the probability of a mid-air collision in European airspace is 2.7×10^{-8} which equates to one in every 3 years. When TCAS II Version 7.1 is implemented, that probability will be reduced by a factor of 4.

The above article suggests that there is currently no solution to the problem of contrary ATCO and RA instructions occurring "simultaneously."

"**If a conflicting ATCO instruction coincides with an RA, the pilot may assume that ATCO is fully aware of the situation and is providing the better resolution. But in reality ATCO is not aware of the RA until the RA is reported by the pilot.**"

The first sentence implies that in the case of coincidental instructions, as for Überlingen, then the ATCO should be followed. But the second sentence implies

that, as for Überlingen, *the ATCO could not be fully aware of the situation until after the pilot reports the RA*—with no probably time for negotiation of the inconsistency. So the first sentence is logically incoherent—if the pilot hasn't reported the RA, the pilot can not assume that the ATCO is fully aware, and hence the pilot should obey the RA instead.

With the introduction of TCAS 7.1 new reversal logic, it is hoped that this issue will be resolved – it won't matter what this crew does; if they violate TCAS RA, then the other plane will be instructed to reverse.

22 Appendix: Brahms Reformulation of WMC model

This appendix describes the BRAHMS model and simulation created based on Kim's (2011) dissertation. Following Young and Pritchett's nomenclature, we refer to this as the WMC ("work model that computes") model.

The purpose of this Brahms modeling effort in the fall of 2011 was to develop fundamental constructs that would be useful in any Brahms air traffic simulation model, while learning how the representation of work practice in the "work model that computes" relates to the work practice formulation in Brahms. The WMC model was extremely helpful for providing an initial framework on which to develop a Brahms model of pilot-flight system interactions. The WMC model was created for a to illustrate a functional-allocation simulation based on cognitive task analysis, which is complementary to a work systems simulation. The comparisons that appear here are relative to the analytic perspective of work practice modeling and the purposes of the Brahms-GÜM project, and are not intended to be a critique of the WMC methods or results.

The WMC model emphasizes the interaction of the pilot and automation under three control regimes; the air traffic controller's work is only included to the extent required by the crew's actions. After creating the "manual only" model, our project efforts were redirected in December 2011 to the Überlingen scenario. Consequently, we did not simulate the hybrid and fully automated modes in Brahms. However, the model does include the pilot's interaction with automated flight control systems to both get information and control the aircraft, as described subsequently in the descent scenario narrative.

The greatest part of the effort in creating this Brahms model lies in the simulation of the aircraft flight representing the position and velocity vector in three dimensions at three second intervals. These calculations occur in a Java function, invoked by a Brahms workframe that updates the aircraft's status parameters (see figures). This appendix provides details about this part of the model because it was directly reused creating the Brahms generalized Überlingen model.

As background, the following is an outline of the WMC agent modeling framework:

Agent:	entity that performs an action
Action:	work performed by an agent at one instance in time
Resource:	a specific state of the environment
Environment:	collection of resources available for interaction with the agent
Decision actions:	process of selecting a course of action based on the environmental context
Temporal actions:	actions initiated by the agent. It obtains a specific resource from the environment and changes its value
Functions:	describes how something may be achieved (in the coding sense). It can call upon other functions or temporal actions.

Thus activity is modeled in terms of abstract functions and primitive (temporal) actions. The abstract functions of WMC are organized hierarchically in four levels (from most to least abstract):

Functional purpose:	Purposes for which the system was designed as well as the external constraints on its operation.
Abstract functions:	The criteria that the work system uses for measuring its progress towards the functional purposes.
General functions:	Basic functions that the system is designed to achieve in order to accomplish the higher functional purposes.
Temporal functions:	Collections of temporal actions

These abstraction levels were originally replicated in Brahms but have been subsequently reformulated to better conform to Brahms constructs so we could simulate multitasking of the ATCOs and pilots; see Appendix 22.6.

22.1 Descent Scenario Narrative

Here we describe the particular flight simulated in the WMC model; font formatting indicates *aircraft*, locations, and **agents** that are components of the Brahms model.

The scenario begins with a *United Airlines plane*, flight UA 888, traveling from San Francisco (SFO) to Los Angeles (LAX). At 400 nautical miles from "Top of Descent" (TOD) coordinates, which is the starting point for landing into Los Angeles airport, the **Captain** starts to review the "LAX RIIVR TWO Arrival" *STAR* (Standard Terminal Arrival Route) *procedure* for landing onto runway 25L of LAX using the *CDU* (Control Display Unit) in the cockpit.

Meanwhile, an **ATCO** (Air Traffic Controller) at LAX notices flight UA 888 entering controlled airspace being monitored on ATC's workstation. **ATCO** radios the plane to request for handoff from the regional air traffic control center. The **Captain** receives and acknowledges handoff from LAX **ATCC**. The **Captain** then requests clearance from LAX **ATCO** to start descent from cruise and fly to each waypoint as specified in the *STAR procedure*. LAX **ATCO** responds to give clearance to land.

The **Captain** starts the plane's approach to TOD and continuously monitors progress of plane to each waypoint using the *PFD* (Primary Flight Display) and *ND* (Navigation Display). Just before arriving at each waypoint, the Captain reviews and inputs the next waypoint's heading, airspeed and vertical speed into FMS (Flight Management System) using the *MCP* (Mode Control Panel). The plane tracks its *GPS* coordinates, as it changes speed and heading, and reports back its information to the *CDU*, *PFD* and *ND*.

As plane approaches the last waypoint before touching down onto the runway, the Captain quickly deploys and confirms the plane's *landing gears* are deployed. As the plane touch down onto the runway, the Captain deploys *flaps* and applies plane's *airbrakes* to slow down the plane on the runway.

Figure 22-1: Data Flow Among Brahms WMC Model Components

22.2 Brahms WMC Model Components

The following sections describe geography/locations, agents and objects modeled in BRAHMS to simulate the descent scenario. Refer also to Appendix 23 for complete list of Brahms-GÜM components to understand how model definitions of groups/classes and instances are stored and organized.

22.2.1 Geography models

An area in BRAHMS represents a geographical location and is used to create a geographical representation for use in the model. Areas, such as "LAX," are instances of Brahms Area Definitions, which are classes in this domain (e.g., cities, airports).

Table 22-1: Brahms WMC Los Angeles Airport & Airspace

Area	Description	Area Definition
LAX	Los Angeles Airport	Instance of an Airport within City
LAX_ARTCC	LAX Air Traffic Control Center	Instance of a Air Route Traffic Control Center building and located in LAX
LAX_RWY_25L	Runway 25L in LAX	Instance of Runway within Airport
LAX_ATC_Workstation	LAX ATCO workstation area	Instance of Workstation Area and located in LAX_ARTCC
LAX_ControlledAirSpace	Controlled air space above LAX	Instance of Controlled Air Space within World
TOD_Waypoint, GRAMM_Waypoint, RUSTT_Waypoint, HABSO_Waypoint, RIIVR_Waypoint, DECEL_Waypoint, LUVYN_Waypoint, KRAIN_Waypoint, FUELR_Waypoint, GAATE_Waypoint, HUNDA_Waypoint, LIMMA_Waypoint, RWY_25L_Waypoint	Waypoints in air space with associated GPS Data and Runway (if any).	Instance of Controlled Air Space within World

Table 22-2: Brahms WMC Plane Areas

Area	Description	Relationship to other Areas
UALPlaneArea	United Airlines Plane areas	Instance of Aircraft Areas
UALPlaneCockpitArea	Cockpit area within plane	Instance of UALPlaneArea
UALPlaneCabinArea	Passenger cabin area within plane	Instance of UALPlaneArea

22.2.2 Agent models

Agents have behaviors and beliefs which they can inherit from multiple groups. Thus the UA888_Captain is an agent belonging to the UALPilots group, which belongs to the Pilots group, etc. The UA888_Captain therefore inherits all of the behaviors of the UAPilots, Pilots, FlightCrew and RadioCommunicator groups.

Table 22-3: Brahms WMC Agents

Agent	Group Membership	Description	Location
UA888_Captain	UALPilots < Pilots < RadioCommunicator, FlightCrew	Captain of United Flight 888	UALPlaneCockpitArea
ATC_LAX	AirTrafficControllers < RadioCommunicator	LAX's Air Traffic Controller	LAX_ARTCC

22.2.3 Object models

Table 22-4: Brahms WMC Plane and Instrument Objects

Object	Class	Description	Location
UA888_Plane	Boeing747-400	United Flight 888 plane	LAX_ControlledAirSpace
UA888_Plane_Control_ Stick	Aircraft Control Column	Control yoke or stick of UA 888 plane	UALPlaneCockpitArea
UA888_Plane_Speed_ Brake_Control	Aircraft Speed Brake Control	Control to decrease speed of plane	UALPlaneCockpitArea
UA888_Plane_Landing _ Controls	Aircraft Landing Control	Control to deploy landing gear	UALPlaneCockpitArea
UA888_Plane_Radio	Aircraft Radio	Radio installed in aircraft	UALPlaneCockpitArea

Table 22-5: Brahms WMC Flight Management Systems (Objects)

Object	Class	Description	Location
UA888_Plane_PFD	Boeing Primary Flight Display	Displays the flight mode annunicator modes, altitude, air speed and heading of plane.	UALPlaneCockpitArea
UA888_Plane_ND	Boeing Navigation Display	Displays horizontal view of the area ahead of aircraft, its heading, and the waypoints defining the flight route.	UALPlaneCockpitArea
UA888_Plane_CDU	Control Display Unit	Displays that shows Flight Plans and to program waypoints and their restrictions.	UALPlaneCockpitArea
UA888_Plane_MCP	Mode Control Panel	A long narrow panel located centrally in front of the pilot, may be used to control Heading(HDG), Speed(SPD), Altitude(ALT), Vertical Speed(V/S), Vertical Navigation(VNAV) and Lateral Navigation(LNAV).	UALPlaneCockpitArea

Table 22-6: Brahms WMC Air Traffic Control Systems (Objects)

Object	Class	Description	Location
LAX_ATC_System	Air Traffic Control System	Computer used by Air Traffic Controller	LAX_ATC_Workstation
LAX_PSR	Primary Surveillance Radar	Radar monitoring planes in its air space.	LAX_ARTCC

Table 22-7: Brahms WMC Flight Procedures & Data (Objects)

Object	Class	Description	Location
STAR_RIIVR_TWO_Arrival	Standard Terminal Arrival Route	Published Flight Segments/routes (typically via VOR's and intersections) from each of these transitions to a point near a destination airport.	
LAX_DescentChecklist	Descent Checklist	Procedure composed of multiple steps ensuring the flight deck systems are configured properly, and the pilot and cabin crews are "briefed" for upcoming phases.	UALPlaneCockpit Area
TOD_FlightSegment, GRAMM_FlightSegment, RUSTT_FlightSegment, HABSO_FlightSegment, RIIVR_FlightSegment, DECEL_FlightSegment, LUVYN_FlightSegment, KRAIN_FlightSegment, FUELR_FlightSegment, GAATE_FlightSegment, HUNDA_FlightSegment, LIMMA_FlightSegment, RWY_25L_FlightSegment	Flight Segments	Published procedure for a flight segment that specifies waypoint, airspeed and altitude restriction, etc.	
TOD_gps, GRAMM_gps, RUSTT_gps, HABSO_gps, RIIVR_gps, DECEL_gps, LUVYN_gps, KRAIN_gps, FUELR_gps, GAATE_gps, HUNDA_gps, LIMMA_gps, RWY_25L_gps	GPS Data	Global positioning system coordinates – longitude, latitude & altitude.	

22.3 Conceptual Objects Model

A conceptual object in BRAHMS is used to model and track in the simulation entities that have properties that are conceptual. A flight is a conceptual object; its properties include scheduled departure and arrival times. Conceptual objects do not exist as discrete things, but are realized by a combination of people, objects, and events, such as an aircraft, who is actually onboard, and actual departure time, route, etc. Similarly, an airline's flight schedule is a conceptual object, representing a combination of flights, each of which is a conceptual object. By incorporating conceptual objects in the model, statistics can be generated during the simulation, such as "touch time" and "cycle time"; also object flows can be generated to depict a work process.

Table 22-8: Brahms WMC Conceptual Objects

Object	Class	Description
Manage_Aircraft_Energy, Manage_Aircraft_Systems, Manage_Communication, Manage_Lateral_Route, Manage_Stability_Of_Work_Environment, Manage_Trajectory	Generalized Function	Temporal functions that are grouped into generalized functions.

The following section describes the notion of "Generalized Functions" in the WMC model and how they relate to pilot actions and aircraft states.

22.4 Structure of Pilot's Activities, Workframes, Thoughtframes

For simplicity, we examine a single "function allocation" in the WMC model of the descent phase in "mostly manual" mode.

Table 22-9: "Mostly-manual" function allocation (FA4, teamwork actions red-coded; excerpt from Kim 2011, p. 86).

Temporal Function	*Pilot*	*Automation*
Control Vertical Speed	Dial Altitude Selector Dial VS Selector Push Alt Hold Switch Push FLCH Switch Push Vertical NAV Switch Push Vertical Speed Switch Monitor Green Arc	Update Pitch Control Evaluate Vertical Mode Evaluate Alt Restriction Mode Altitude Reminder

A temporal function "aggregates (temporal) actions according to inherently-coupled dynamics and purposes." In the Brahms model the temporal functions are activated by top-level workframes of the Pilot and the actual functional allocation (Pilot column above) correspond to Workframes in an associated composite activity (see excerpt Table 22-10).

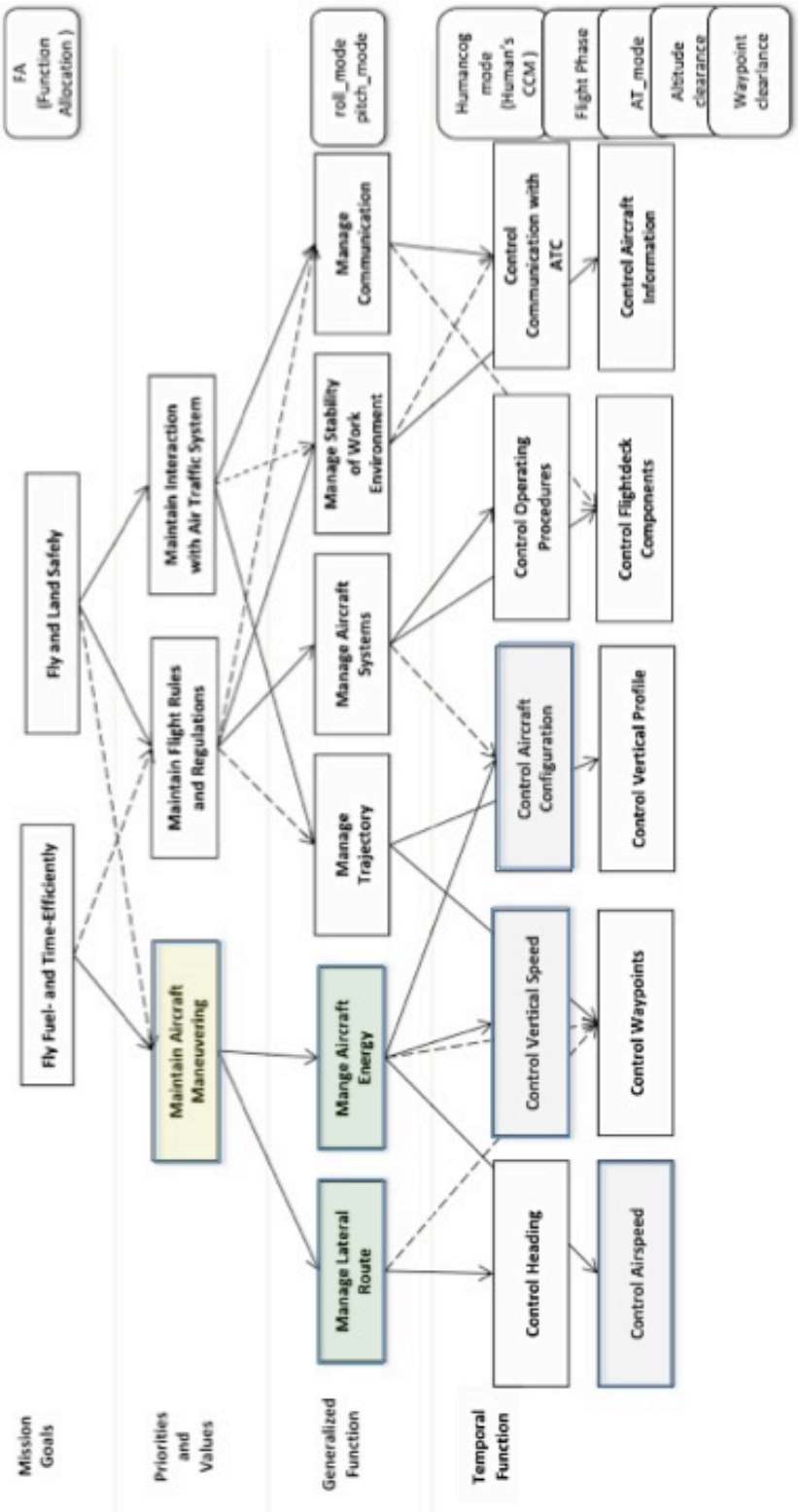


Figure 22-2: WMC "Arrival-Approach work model" hierarchy (adapted from Kim 2011, p. 81)

Table 22-10: Representation of WMC Structure in Brahms Thoughtframes, Workframes, and Activities

I. Pilot Top-Level Thoughtframes (pTF)	**II. Pilots Group Attribute generalizedFunction [a conceptual object], belief of Pilot set by pTF**	**III. Pilot Top Level Workframes (with Composite Activity, CA)**	**IV. Workframes in CA**
Priorities and Values	*Generalized Functions*	*Temporal Functions*	*Functions Allocated to Pilot*
Change_Vertical_Speed	Manage_Aircraft_Energy	Control Vertical Speed (controlVerticalSpeed)	Dial Altitude Selector Dial VS Selector Push Alt Hold Switch Push FLCH Switch Push Vertical NAV Switch Push Vertical Speed Switch Monitor Green Arc

These relations are visible in the WMC "Arrival-approach work model" (Figure 22-2). As noted above, this level of abstraction is represented by the top-level thoughtframes of the Pilot. A key notion in the WMC model is that the actual actions are determined by the Opportunistic, Tactical, and Strategic "cognitive control modes." These affect for example the degree of monitoring possible. We do not currently model these modes in the Brahms formulation because of the initial focus interactive behaviors—what people do in particular situations—rather than describing abstract states of mind corresponding to these kinds of behaviors. In effect cognitive control modes could be viewed a way of grouping and labeling workframes that represent different ways of behaving.

This outline summarizes how the model is structured (*examples in parentheses*), referring again to the table above:

- Pilot's top-level TFs (*Change_Vertical_Speed*) set generalizedFunction belief (*Manage_Aircraft_Energy*).
- Pilot's top-level WFs (*Control Vertical Speed*) model how Pilot specializes (focuses) the generalizedFunction depending on what flight parameters require attention, causing a composite activity to begin (*controlVerticalSpeed*).
- Composite Activity WFs (*Dial Altitude Selector, …*) determine what selector/switch needs to be adjusted, and do a primitive activity (*dialingSelector* or *pushingSwitch*).

The top-level TFs and associated generalized functions are provided by the next table.

Table 22-11: Pilot's top-level TFs and associated generalized functions

Pilot Top-level Thoughtframes	generalizedFunction belief	Pilot Top-level Workframes
Review_Operating_Procedures	Manage_Aircraft_Systems	Control_Operating_Procedures
Configure_Flight_Deck	Manage_Aircraft_Systems	Control_Flight_Deck_Components
Configure_Aircraft_Before_Touchdown	Manage_Aircraft_Systems	Control_Aircraft_Configuration
Configure_Aircraft_Touchdown	Manage_Aircraft_Systems	Control_Aircraft_Configuration
Respond_To_ATC	Manage_Communication	Control_Communication_With_ATC
Response_From_ATC	Manage_Communication	Control_Communication_With_ATC
Request_Waypoint_Clearance	Manage_Communication	Control_Communication_With_ATC
Verify_Aircraft_Information	Manage_Stability_Of_Work_Environment	Control_Aircraft_Information
Monitor_Plane_To_Waypoint	Manage_Trajectory	Control_Waypoints
Monitor_Plane_Vertical_Profile	Manage_Trajectory	Control_Vertical_Profile
Change_Airspeed	Manage_Aircraft_Energy	Control_Airspeed
Change_Vertical_Speed	Manage_Aircraft_Energy	Control_Vertical_Speed
Change_Heading	Manage_Lateral_Route	Control_Heading
Plane_Taxi_On_Runway	unknown	

The priorities assigned to Pilot's WFs are based on generalized functions:

100 = Manage Communications (with ATCO or pilots)
40 = Manage Stability Of Work Environment (e.g. verify route & waypoint info, etc.)
30 = Manage Lateral Route (e.g. change heading)
20 = Manage Aircraft Energy (e.g. change airspeed, vertical speed, deploy/retract flaps or gears, brakes, etc.)
10 = Manage Aircraft Systems (e.g. turn on/off auto-pilot, altitude alert, etc.)
0 = Manage Trajectory (e.g. monitoring travel to waypoint, rate of descent, etc.)

Communications interrupt all non-communication related workframes because they are given the highest priority in the model.

As described below (Section 22.6), this framework did not provide the flexibility required for modeling interruptions and shifts of attention as new information is broadcast or visible to the ATCOs and pilots. Details about this difficulty are explained after first describing how So Young and Pritchett et al.'s functional allocation framework is reformulated in terms of Brahms' behavioral, activity-based framework.

22.4.1 Example of pilot's behavior logic/flow

In the original Brahms WMC model, which is being documented in this appendix, the pilot's behaviors are determined by beliefs about his/her "generalizedFunction," an attribute of the Pilots group. For example,

Current.generalizedFunction = Manage_Aircraft_Energy

Here generalizedFunction is a conceptual class and Manage_Aircraft_Energy is an instance of this class (i.e., a conceptual object). By convention in Brahms, an attribute of a group/agent having a value that is an instance of a conceptual class is interpreted as being something the agent conceives about him/herself.

In this formulation, the pilot's top-level (always active) thoughtframes set the GeneralizedFunction (instance of Generalized Functions), thus modeling the pilot's conceiving what he should be doing now. These TFs are conditional on the Pilot's route (FlightSegment instances) and plane's airspeed, vertical speed, bearing/heading, etc. They roughly correspond to phases of activity during the flight. Initial and final stages occur only once (e.g., Configure Flight Deck); others represent repeating activities (e.g., Respond to ATC).

For example, the GeneralizedFunction Manage_Aircraft_Energy is set by the top-level Pilots group TF Change_Vertical_Speed.

```
/*
* Generalized Function - Manage Aircraft Energy
*/
thoughtframe Change_Vertical_Speed {
        display: "Change Vertical Speed";
        priority: 10; // higher priority
        variables:
                foreach(FlightSegment) route;
                forone(Aircraft) plane;
        when(   knownval(current.generalizedFunction = unknown) and
                knownval(current.route = route) and
                knownval(route.cleared = true) and
                knownval(route.verified = true) and
                knownval(current.inPlane = plane)  and
                knownval(plane.touchdown = false) and
                knownval(plane.verticalSpeed != route.verticalSpeed))
        do {
                conclude((current.generalizedFunction = Manage_Aircraft_Energy), fc:0);
        }
}//tf Change_Airspeed
```

The TF precondition above indicates if the pilot is not currently engaged in some flight management activity (generalizedFunction = unknown), the route is cleared (with ATC) and verified (see below), the plane is not already on the ground, and the plane's vertical speed is not the prescribed speed of the route, then Manage_Aircraft_Energy (i.e., control the vertical speed)."

```
workframe Control_Vertical_Speed {
        display: "Control Vertical Speed";
        variables:
                forone(Aircraft) plane;
                forone(FlightManagementSystem) fms;
```

```
            foreach(FlightSegment) route;
        detectables:  << SEE FOOTNOTE[63] >>
            detectable Respond_To_ATCO {
                detect((current.generalizedFunction = Manage_Communication))
                then impasse;
            }
            detectable Configure_Aircraft {
                detect((current.generalizedFunction = Manage_Aircraft_Systems), dc:0)
                then impasse;
                }
        when(   knownval(current.generalizedFunction = Manage_Aircraft_Energy) and
            knownval(plane = current.inPlane)  and
            knownval(current.location = fms.location) and
            knownval(route = current.route) and
            knownval(plane.verticalSpeed != route.verticalSpeed))
        do {
            controlVerticalSpeed (route);
                }
}//wf Control_Vertical_Speed
```

Controlling the vertical speed is a composite activity, having multiple workframes that represent different operations that might be required. In the manual model there are seven workframes:

Dial_Altitude_Selector
Dial_VS_Selector
Push_Alt_Hold_Switch
Push_FLCH_Switch
Push_Vertical_NAV_Switch
Push_Vertical_Speed_Switch
Monitor_Green_Arc

These are applied in sequence; the first two dial selectors and the second two push switches. These are primitive activities that only take time.

```
primitive_activity dialingSelector() {
        display: "Dialing Selector";
        random: true;
        min_duration: 1;
        max_duration: 5; }
//pac dialingSelector
```

[63] From the perspective of the Brahms framework, this WF is somewhat nonsensical—it should not be necessary for agents to detect their own beliefs. In this direct mapping of WMC into Brahms, problems were encountered in modeling how an agent returns from a prior activity after an interruption (which is normally handled by how the Brahms engine handles subsumption, interruption, and resumption of WFs). These detectables are a workaround so that after dealing with communications, the agent returns to the previous activity that had been "impassed" – otherwise the belief about the previous generalizedFunction would be lost. This awkwardness was resolved in Brahms–GÜM (Section 22.6).

The 5th and 6th WFs actually make the change that affects the aircraft. The workframe **Push_Vertical_Speed_Switch** communicates altitude change to Mode Control Panel (MCP) which communicates to plane which communicates to the Control Display Unit (CDU – treated here as being the automation system itself) that performs calculations for rate of descent/ascent (vertical speed) based on current airspeed; this is communicated back to Primary Flight Display (PFD), from which the pilot read the vertical speed originally.

Notice the ordering of the conclude and communication activity **pushVerticalSpeedSwitch** (where the pilot sets the MCP, which appears here as with: mcp). The conclude changes the pilot's belief about the vertical speed to the correct setting. The "send" action gives the mcp the pilot's belief about the route.verticalSpeed.[64]

```
workframe Push_Vertical_Speed_Switch {
        display: "Push Vertical NAV Switch";
        priority: 20;
        variables:
                forone(ModeControlPanel) mcp;
                forone(Aircraft) plane;
        when(   knownval(current.location = mcp.location) and
                knownval(current.inPlane = plane) and
                knownval(route.verticalSpeed != plane.verticalSpeed))
        do {
                conclude((route.verticalSpeed = plane.verticalSpeed), fc:0);
                pushVerticalSpeedSwitch (mcp, route);
        }
}//wf Push_Vertical_Speed_Switch

communicate pushVerticalSpeedSwitch(ModeControlPanel mcp, FlightSegment route) {
        display: "Pushing Vertical Speed Switch"
        random: true;
        min_duration: 1;
        max_duration: 4;
        with: mcp;
        about: send(route.verticalSpeed = unknown);
}//com pushVerticalSpeedSwitch
```

22.4.2 Pilot's communication with air traffic control

In the Brahms WMC model, all communications occur through the model constructs shown in the table.

Table 22-12: Pilot's thoughtframes, belief, and workframe related to communications

Pilot Top-level	generalizedFunction	Pilot Top-level Workframes

[64] The "unknown" in the Send is a Brahms convention/shorthand—the syntax requires some value here; whatever the Pilot believes will be conveyed.

Thoughtframes	belief	
Respond_To_ATC	Manage_Communication	Control_Communication_With_ATC
Response_From_ATC	Manage_Communication	Control_Communication_With_ATC
Request_Departure_Clearance	Manage_Communication	Control_Communication_With_ATC
Request_Waypoint_Clearance	Manage_Communication	Control_Communication_With_ATC

The WMC Generalized Function "Manage Communication" is here modeled by four TFs, as listed in the first column of the table above. Each of these TFs gives the pilot the belief that his generalizedFunction is Manage_Communication. Each TF has a screen to assure that the pilot is not already engaged in this generalized function, "current.generalizedFunction != Manage_Communication." The use of the "foreach" construct in the example below implies that this TF is applicable (i.e., to be potentially fired) for every ATCO and every topic.

```
thoughtframe Respond_To_ATCO {
        display: "Respond To ATC";
        variables:
                foreach(AirTrafficControllers) atc;
                foreach(string) topic;
        when(   knownval(current.generalizedFunction != Manage_Communication) and
                knownval(atc.flightCallSign = current.flightCallSign) and
                knownval(atc.performative = "REQUEST") and
                knownval(atc.subject = topic))
        do {
                conclude((current.generalizedFunction = Manage_Communication), fc:0);
                conclude((current.subject = topic), fc:0);
        }
}//tf Respond_To_ATC

thoughtframe Response_From_ATCO {
        -- same as Respond_To_ATCO except
        knownval(atc.performative = "AGREE")

thoughtframe Request_Departure_Clearance {
        variables:
                foreach(StandardInstrumentDeparture) sid;
                foreach(Aircraft) plane;
                foreach(FlightSegment) route;
        when(   knownval(current.generalizedFunction != Manage_Communication) and
                knownval(sid = current.airportSID) and
                knownval(current.inPlane = plane) and
                knownval(plane.location = sid.departureAirport) and
                knownval(current.route = sid.mapFlightSegments(1)) and
                knownval(route.cleared = false))
        do {
                conclude((current.generalizedFunction = Manage_Communication), fc:0);
                conclude((current.subject = "departure"), fc:0);
        }
}//tf Request_Departure_Clearance

thoughtframe Request_Waypoint_Clearance {
```

```
        display: "Request Waypoint Clearance";
        variables:
                foreach(AirTrafficControllers) atc;
                forone(Flight) flight;
                forone(FlightSegment) route;
        when(   knownval(current.generalizedFunction != Manage_Communication) and
                knownval(atc.flightCallSign = current.flightCallSign) and
                knownval(flight = current.flight) and
                knownval(flight.handoff = true) and
                knownval(route = current.route) and
                knownval(route.cleared = false))
        do {
                conclude((current.generalizedFunction = Manage_Communication), fc:0);
                conclude((current.subject = "clearance"), fc:0);
        }
}//tf Request_Waypoint_Clearance

thoughtframe Waypoints_Cleared {
        display: "Waypoints Cleared";
        variables:
                forone(StandardTerminalArrivalRoute) arrivalRoute;
                forone(Flight) flight;
                forone(AirTrafficControllers) atc;
                collectall(FlightSegment) routes;
        when(   knownval(flight = current.flight) and
                knownval(flight.cleared = true) and
                knownval(flight.clearedByATCO = atc) and
                knownval(routes.cleared = false))
        do {
                conclude((routes.cleared = true), fc:0);
                conclude((routes.clearanceByATCO = atc), fc:0);
        }
}//tf Waypoints_Cleared
```

When "current.generalizedFunction = Manage_Communication," (i.e., the pilot believes he is engaged in a communication—therefore he needs to say and/or do something), the following top-level WF of the pilot becomes available for application (corresponding to the agent conceiving "what I can be doing now"). This WF simply invokes a composite activity, controlCommunicationWithATC, to carry out the communication.

```
workframe Control_Communication_With_ATCO {
        variables:
                forone(Aircraft) plane;
                forone(AircraftRadio) radio;
                foreach(AirTrafficControllers) atc;
                foreach(string) topic;
                foreach(string) speechAct;
        when(   knownval(current.generalizedFunction = Manage_Communication) and
                knownval(plane = current.inPlane)  and
```

```
                knownval(radio.location = current.location) and
                knownval(atc.flightCallSign = current.flightCallSign) and
                knownval(topic = current.subject) and
                knownval(speechAct = current.performative))
        do {
                controlCommunicationWithATC(atc, radio);
        }
}//wf Control_Communication_With_ATC
```

The composite activity controlCommunicationWithATCO handles receiving, requesting, and responding to information. The WFs are:

Receive_Altitude_Clearance
Receive_ILS_Clearance
Receive_Waypoint_Clearance
Receive_Departure_Clearance
Respond_Handoff
Respond_Clearance
Request_Departure

These WFs were not completely modeled in Brahms WMC because project objectives suggested shifting to developing Brahms-GÜM rather than perfecting Brahms WMC; see the footnote.

The WFs set two primitive activities, "listen" (which takes a certain time) and "sendMessageViaRadio." The latter is a communication activity of AirTrafficCommunicators, a group to which pilots belong; it simply sends ("tells") a set of beliefs to the control center ("unknown" here allows transmitting whatever the pilot believes about the indicated attributes, which are concluded in the WFs that invoke sendMessageViaRadio—see below):

```
communicate sendMessageViaRadio(AircraftRadio radio, int minDuration, int maxDuration) {
        display: "Send Message Via Radio";
        random: true;
        min_duration: minDuration;
        max_duration: maxDuration;
        with: radio;
        about:  send(current.controlCenter = unknown),
                send(current.reciever = unknown),
                send(current.flightCallSign = unknown),
                send(current.subject = unknown),
                send(current.performative = unknown),
                send(current.route = unknown);
}//com sendMessageToPlane
```

There are also two TFs, Departure_Cleared & Flight_Plan_Routes_Cleared, that make corresponding assertions in response to information received (listed after the composite activity's WFs).

```
composite_activity controlCommunicationWithATC(AirTrafficControllers atc, AircraftRadio radio) {
        display: "Control Communication with ATC";
activities:
        primitive_activity listen(int minDuration, int maxDuration) {
                display: "Listening";
                random: true;
                min_duration: minDuration;
                max_duration: maxDuration;
        }
workframes:
workframe Receive_Altitude_Clearance {  SEE NOTE[65]
        display: "Receive Altitude Clearance";
        variables:
                forone(FlightSegment) route;
        when(   knownval(atc.performative = "AGREE") and
                knownval(atc.subject = "altitude_clearance") and
                knownval(atc.flightCallSign = current.flightCallSign) and
                knownval(route = current.route))
        do {
                listen(2, 3);
                retractBelief(atc, "performative");
                //controlVerticalSpeed(route);
                conclude((current.generalizedFunction = unknown), fc:0);
        }
}//wf Receive_Altitude_Clearance

workframe Receive_ILS_Clearance {
        same as Receive_Altitude_Clearance, except:  knownval(atc.subject = "ils_clearance")
        and has actions:
                conclude((flight.cleared = true), fc:0);
                conclude((flight.clearedByATCO = atc), fc:0);

workframe Receive_Waypoint_Clearance {
        same as Receive_ILS_Clearance, except:  knownval(atc.subject = "waypoint_clearance")

workframe Receive_Departure_Clearance {
        same as Receive_ILS_Clearance, except:  knownval(atc.subject = "departure")

workframe Respond_Handoff {
        display: "Respond Handoff";
        variables:
                forone(Flight) flight;
        when(   knownval(atc.performative = "REQUEST") and
                knownval(atc.subject = "handoff") and
                knownval(atc.flightCallSign = current.flightCallSign) and
```

[65] Receive_Altitude_Clearance & ILS_clearance not fully modeled and doesn't do anything. The pilot never requests altitude clearance and ATCO is not modeled to give clearance. The WF as written is clearance for particular flight segment/route rather than clearance for entire flight. Receive_Departure_Clearance in the Brahms model is not in Pritchett et al. WMC; it is also incomplete. Receive_Waypoint_Clearance comes from Pritchett WMC; the Brahms WMC model clears all waypoints for entire flight.

```
                knownval(flight = current.flight))
        do {
                conclude((current.controlCenter = atc.controlCenter), fc:0);
                conclude((current.subject = "handoff"), fc:0);
                conclude((current.performative = "AGREE"), fc:0);
                conclude((current.reciever = atc), fc:0);
                sendMessageViaRadio(radio, 1, 5);  // move on to request for clearance
                retractBelief(atc, "performative");
                listen(2, 3);
                conclude((flight.handoff = true), fc:0);
                conclude((current.generalizedFunction = unknown), fc:0);
        }
}//wf Respond_Handoff

workframe Request_Clearance[66] {
        display: "Request Clearance";
        variables:
                forone(FlightSegment) route;
                forone(Aircraft) plane;
        when(   knownval(plane = current.inPlane)  and
                knownval(route = current.route) and
                knownval(route.cleared = false) and
                knownval(atc.flightCallSign = current.flightCallSign) and
                knownval(current.subject = "clearance"))
        do {
                conclude((current.controlCenter = atc.controlCenter), fc:0);
                conclude((current.performative = "REQUEST"), fc:0);
                conclude((current.reciever = atc), fc:0);
                conclude((radio.inAircraft = plane), fc:0);
                sendMessageViaRadio(radio, 1, 5); // move on to request for clearance
                listen(1, 3);
                conclude((current.generalizedFunction = unknown), fc:0);
        }
}//wf Request_Clearance

workframe Request_Departure {
        same as Request_Clearance, except:  knownval(current.subject = "departure")
```

Thoughtframes:

```
thoughtframe Departure_Cleared {
        variables:
                foreach(StandardInstrumentDeparture) sid;
                foreach(int) order;
                foreach(FlightSegment) route;
        when(   knownval(current.airportSID = sid) and
                knownval(route = sid.mapFlightSegments(order)) and
                knownval(atc.performative = "AGREE") and
                knownval(atc.subject = "departure"))
        do {
                conclude((route.cleared = true));
```

[66] Reference to "flight" variable and precondition removed here because it was unnecessary.

```
        }
}//tf Departure_Cleared

thoughtframe Flight_Plan_Routes_Cleared {
        variables:
                foreach(FlightPlan) plan;
                foreach(FlightSegment) route;
                foreach(int) order;
        when(   knownval(plan = current.flightPlan) and
                knownval(route = plan.mapFlightSegments(order)) and
                knownval(atc.performative = "AGREE") and
                knownval(atc.subject = "departure"))
        do {
                conclude((route.cleared = true));
        }
}//tf Flight_Plan_Routes_Cleared
} //cac controlCommunicationWithATC
```

22.5 Simulation of Aircraft Flight

The following graphics show the flight path coordinates (longitude, latitude & altitude) as the plane travels from one waypoint to another. Figure 22-3 shows the path in Google Earth. Figure 22-4 is from the Brahms AgentViewer, configured to display the aircraft's and CDU's behaviors. The calculation of each waypoint (Figure 22-5) triggers the pilot model to set the correct values into cockpit computer CDU.

Here are the details: At the start of the timeline (Figure 22-4), UA 888 is in Workframe (wf) "Fly To Waypoint," which performs a Move activity (mv) to new waypoint, RIIVR_Waypoint. Simultaneously, Control Display Unit (CDU) is in Workframe "Track GPS to Waypoint," which executes a Java Activity that updates GPS coordinates beliefs in UA888_Plane_CDU_Tracking_GPS_5 every three seconds (set by an initial belief "current.gpsUpdateRate = 3") until UA 888 communicates it has reached that waypoint. When the Java Activity updates UA888_Plane_CDU_Tracking_GPS_5 coordinate beliefs, a Workframe in this object is triggered to communicate them to CDU. (This is being updated to assert facts instead, which will make the communication unnecessary.)

When UA 888 reaches the waypoint, CDU calculates distance to the next waypoint using updated GPS coordinate data. UA 888 then starts Workframe "Cruise at Waypoint" and CDU continues tracking changes in UA888_Plane_CDU_Tracking_GPS_5 coordinate beliefs in Workframe "Track_GPS_At_Waypoint." This serves as an intermediate state to represent arrival at the waypoint without further instructions being dialed in. The pilot communicates the new waypoint, airspeed, vertical speed, etc. to UA 888, which communicates these values to CDU. CDU then calculates travel time to new waypoint and communicates time to UA 888. Once UA 888 gets travel time to new waypoint, it starts Workframe "Fly To Waypoint" to move to new waypoint. Again CDU starts "Track GPS to Waypoint," which updates

UA888_Plane_CDU_Tracking_GPS_5 coordinate beliefs.

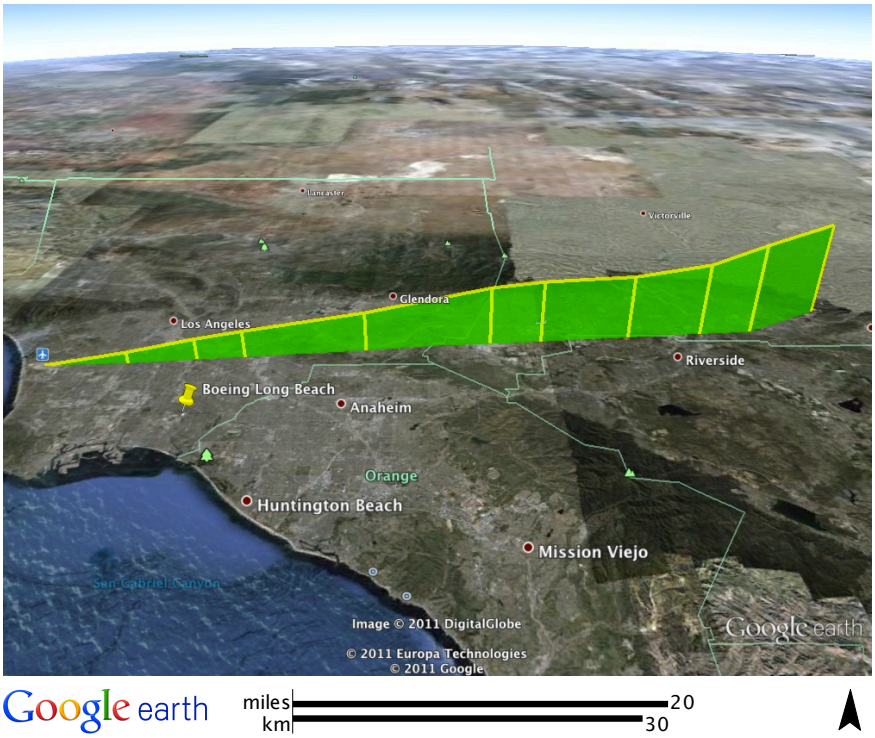


Figure 22-3: Graph of UA888 Flight Path from coordinates generated by WMC simulation

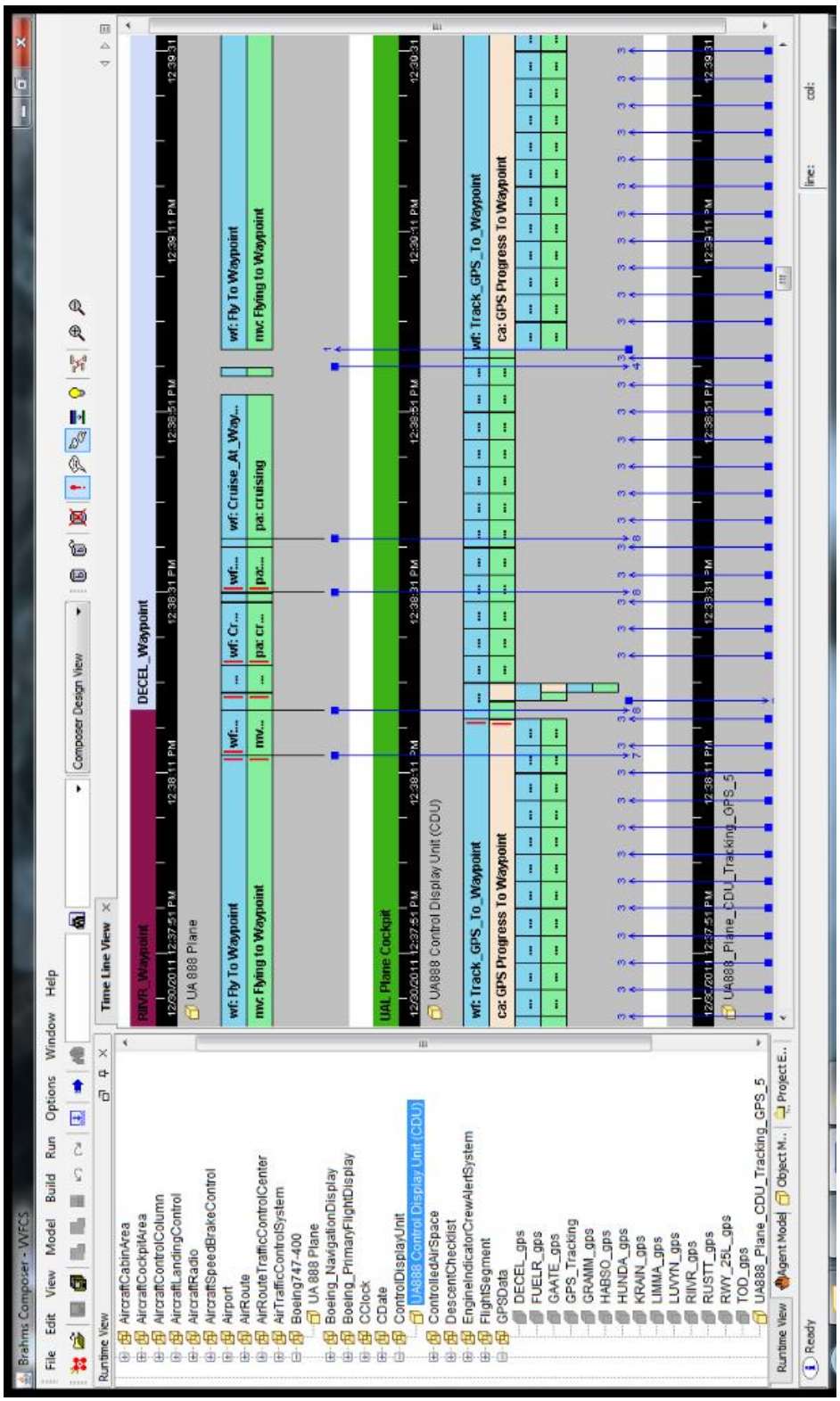


Figure 22-4: AgentViewer Display of Aircraft "Fly to Waypoint" Behavior (see text for details)

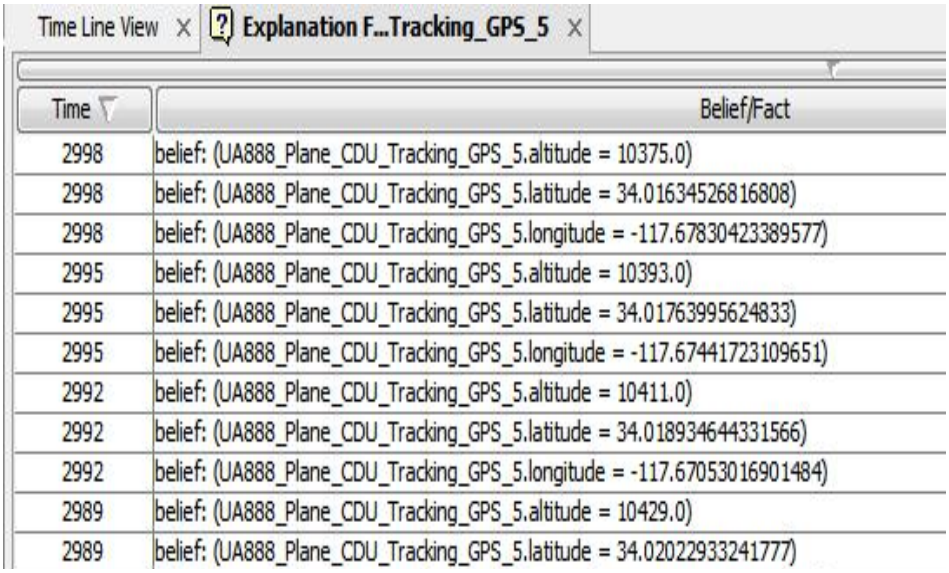


Figure 22-5: Model Propositions Indicating Aircraft Position, Updated at 3 sec Intervals

Also visible in Figure 22-4 are blue and yellow squares designating workframes and actions whose names have been omitted. The Track_GPS_At_Waypoint repeats with shorter durations as the plane cruises along at the waypoint. The unlabeled workframe invoked by the "GPS Progress To Waypoint" composite activity is Update_Next_Waypoint, which performs another composite activity, which includes the workframe Calculate_Distance_To_Next_Waypoint.

In Figure 22-4 the numbers at the end of the blue lines indicate the number of beliefs communicated. For example, UA888_Plane_CDU_Tracking_GPS_5 repeatedly communicates three beliefs, its longitude, latitude and altitude. One can click on the blue square at beginning of communication lines to see what beliefs are communicated.

Here are yet more details: The object UA888_Plane_CDU_Tracking_GPS_5 is dynamically created during the simulation run using the Brahms built-in create object activity (which is why the object has a strange name).

```
create_object createGPSData(GPSData gps) {
        max_duration: 1;
        action: new;
        source: GPSData;
        destination: gps;
        destination_name: Tracking_GPS;
}
```

The CDU receives a belief about the existence of this object, enabling the CDU to pass the object as an input parameter to the Java Activity that computes the next location of the aircraft.

As can be seen in the create-object activity, UA888_Plane_CDU_Tracking_GPS_5 is an instantiation of the **GPSData** class, which stores longitude, latitude and altitude values. This object and class are examples of software (computational) objects, to be contrasted with physical objects (e.g., the CDU) and conceptual objects (e.g., a scheduled flight). The "Flight Procedures and Data" table above list other objects in this class.

The GPS update rate is an initial belief because most of the CDU workframes are of type dataframe, which process data/beliefs of objects. As a general rule, propositions about computational objects are modeled as beliefs in Brahms because they represent world states (where world states are facts); database records and written expressions on a display or a paper form are treated similarly as being beliefs, meaning that they are assertions that are not necessarily true about the world.

In reality, a plane's Control Display Unit doesn't perform the computation to track the GPS. Our implementation is a workaround to handle the limitation that Brahms agents and objects cannot be engaged in more than one activity at a time. In particular, while the plane is performing a Move activity, it cannot simultaneously perform a Java Activity. Therefore some object within the plane must compute the location; for simplicity the CDU was used for this purpose.

The following are the workframes that appear in Figure 22-4.

```
In Aircraft class:
        workframe Fly_To_Waypoint {
            display: "Fly To Waypoint";
            type: dataframe;
            variables:
                foreach(Waypoint) wayPoint;
                foreach(int) travelTime;
            detectables:
                detectable Altitude {
                    detect((current.altitude = unknown));
                }
                detectable Latitude {
                    detect((current.latitude = unknown));
                }
                detectable Longitude {
                    detect((current.longitude = unknown));
                 }
            when(knownval(current.waypoint = wayPoint) and
                knownval(current.location != wayPoint) and
                knownval(current.timeToWaypoint > 0) and
```

```
            knownval(travelTime = current.timeToWaypoint))
        do {
            flyToWaypoint(wayPoint, travelTime);
            conclude((current.timeToWaypoint = 0), fc:0);
        }
    }//wf Fly_To_Waypoint

    workframe Cruise_At_Waypoint {
        type: dataframe;
        variables:
            foreach(Waypoint) waypoint;
        detectables:
            detectable Stop_Cruising {
                detect((current.waypoint != waypoint))
                then abort;
            }
            detectable Touchdown {
                detect((current.touchdown = true))
                then abort;
            }
            detectable Altitude {
                detect((current.altitude = unknown));
            }
            detectable Latitude {
                detect((current.latitude = unknown));
            }
            detectable Longitude {
                detect((current.longitude = unknown));
            }
        when(knownval(current.waypoint = waypoint) and
            knownval(current.location = waypoint))
        do {
            cruising();
        }
    }//wf Cruise_At_Waypoint
```

In ControlDisplayUnit class:

```
    workframe Track_GPS_To_Waypoint {
        type: dataframe;
        variables:
            forone(Aircraft) plane;
            foreach(FlightSegment) route;
            foreach(Waypoint) waypoint;
        when(knownval(current.inAircraft = plane) and
            knownval(plane.waypoint = waypoint) and
            knownval(route.toWaypoint = waypoint))
        do {
          conclude((plane.timeToWaypoint = route.timeToWaypoint), fc:0);
            timeToWaypoint(plane);
            gpsProgressToWaypoint(route);
        }
    }//wf Track_GPS_To_Waypoint
```

```
workframe Track_GPS_At_Waypoint {
        type: dataframe;
        repeat: true;
        variables:
                forone(Aircraft) plane;
                foreach(int) lateralSpeed;
                foreach(int) verticalRate;
                foreach(double) heading;
                forone(int) interval;
                forone(GPSData) coordinate;
        detectables:
                detectable Not_At_Waypoint {
                        detect((plane.location != plane.waypoint))
                        then abort;
                }
        when(knownval(current.inAircraft = plane) and
                knownval(plane.location = plane.waypoint) and
                knownval(lateralSpeed = plane.airSpeed) and
                knownval(verticalRate = plane.verticalSpeed) and
                knownval(heading = plane.bearing) and
                knownval(interval = current.gpsUpdateRate) and
                knownval(coordinate = current.gps) and
                knownval(coordinate.altitude > 0))
        do {
                getPosition(lateralSpeed, heading, verticalRate, interval, coordinate);
                getGPSCoordinates(coordinate);
                conclude((plane.altitude = coordinate.altitude));
                conclude((plane.latitude = coordinate.latitude));
                conclude((plane.longitude = coordinate.longitude));
        }
}//wf Track_GPS_At_Waypoint
```

22.6 Reformulation of Brahms WMC Model for Multitasking

In extending the pilot model of Brahms WMC we found that the structuring by "generalized function" copied from WMC, which posits that conscious inferences about goals drive behavior was inconsistent with Brahms framework, which emphasizes reactive, interactive behavior driven by ongoing conception of activities in context.

Rooting all behaviors in the pilot's thoughtframes (pTFs) conflicted with the essential structure of Brahms' framework, namely that behaviors of agents are driven by WFs that select Activities, which enables simulating multitasking. If WFs were controlled by TFs, then the TFs themselves had to be controlled—and this leads a goal-oriented, inference-based model. But if behavior is driven by decision making (inference by applying TFS), then the agent must always be reasoning about what he/she should be doing and juggling priorities and plans.

In Brahms, these "decisions" about what to do next are modeled by the conditions and priorities of the agents WFs. Most importantly the subsumption organization of

activities provides an implicit contextual structure (modeling a conceptualization of "what I'm doing now") by which the agent is always simultaneously doing multiple activities (within the subsumption path of WF-Activity-WF-…). (A process subsumption architecture is to be contrasted with a function invocation hierarchy in which only one function/procedure is "running" or "active" at one time.) The Brahms engine simulates how people shift their attention within an activity (or at a top-level for reactive behaviors such as responding to a nearby alarming event in the world) and thus replicates an ability for multitasking with an "interrupt and resume" mechanism. (For detailed explanations of this mechanism see Clancey et al. 1998b; Sierhuis et al. 2009). The important distinction between an agent's model of his/her goals and conceptual, often tacit motives is explained in Clancey (2002).

The activity-based subsumption architecture effectively turns the goal-oriented perspective inside out. Inferences are not viewed as driving all behavior in some timeless, isolated space independent of activity. Rather in the Brahms work practice framework, inferences occur within ongoing activities that provide the conceptual context for constraining what is of interest (what you are observing and thinking about), what tools are used (e.g., pen and paper, spreadsheets, computer models), and how judgments are made, represented in the world, and communicated.

In creating Brahms-GÜM model we needed to model multitasking of both the ATCOs and the pilots. To return "control" to the agent's ongoing activities, the pTFs asserting GeneralizedFunction pilot beliefs in Brahms WMC were eliminated. In Brahms-GÜM these beliefs are instead concluded by WFs with preconditions resembling the original pTFs. This enables Brahms interruption-and-resume mechanism, which operates on WFs, to enable the pilot agent to shift attention and activity focus as the dynamic context requires.

In summary, in Brahms-WMC a pilot agent's top-level TFs assert a GeneralizedFunction (GF) belief that trigger (are preconditions of) specific actions in WFs, that include Composite Activities. In the reformulation, rather than viewing GFs as beliefs about goals, they are beliefs about activities, and the GFs are modeled as activities (what the pilot does). Table 22-13 outlines this reformulation.

Table 22-13: Reformulation of GeneralizedFunction in Brahms-GÜM

Brahms-WMC CONSTRUCT	Brahms–GÜM CONSTRUCT(S)
Pilot Top-level **Thoughtframes** asserts:	Pilot Top-level **Workframes** that assert GF belief and invoke:
generalizedFunction belief (goal) that triggers:	**GF Activities**, which contain:
Pilot **Top-level Workframes**, which invoke:	**GF Workframes**, which invoke:
Pilot **CompositeActivities**	Pilot **CompositeActivities**

As can be seen, the analytic shift in the Brahms framework is to model behavior in terms of the agent's conception of "what I'm doing now" (WFs and Activities), which may include beliefs about what the agent is doing (GF belief). In a goal-oriented model by contrast, inference rules that assert goals are at the top and these beliefs control actions. This fits the theory that conceptualization organizing human attention and behavior is largely tacit, with conscious "talk" to oneself or others about goals and actions (aka "planning") being itself an activity. In this respect, planning doesn't sit apart from activity, invoking it, but is rather an interactive behavior of its own right (Clancey 2002).

In Brahms-GÜM the pilot agent still has a belief about the GF ("what I'm doing now"), but WFs control attention directly. A GF precondition for the workframes within the GF Activities is not necessary. The logic of the top-level TFs in Brahms-WMC (which was a direct mapping implied by the So Young & Pritchett et al. WMC model) is moved to WFs that directly control activities, fitting the "reactive" nature of a Brahms agent model.

23 Appendix: Brahms AFCS Überlingen Model Components

Brahms model constructs are organized into text files, which are written by the Brahms Composer, which provides a project explorer and smart editing interface. File names are arbitrary, but conventionally reflect the names of agents, concepts, areas, and objects in the Brahms model. The following listings are created from the model files, indicating the groups/agents, classes/objects, and geographic areas/subareas. The communications among agents and objects are diagramed in Figure 22-1 and Figure 9-1.

Brahms-GÜM consists of two parts, the generic model components (e.g., defining an airport) that are useful for any ATS simulation and the more specific model components that enable defining the space of scenarios that are variations of what occurred at Überlingen (e.g., the specific airports involved).

In the lists that follow, the itemized elements are file names that usually correspond to names in Brahms-GÜM. For example, "Airport" represents the file name "Airport.b," which contains the Airport area definition, specifying it as a subclass of the Country_City area and having two properties, a name and airport code:

```
areadef Airport extends Country_City {
        attributes:
          public string fullName;
          public string airportCode;
} // Airport
```

23.1 Generic Air Transportation System Model Components

These are Brahms component definitions useful for creating Brahms models for many purposes related to air transportation systems. These include agent groups, object classes, and general areas, as well as ATS objects that would be common to most ATS models (e.g., radar, radio). However, there is nothing dictated by the Brahms language that dictates what is in the "generic" model versus the "Überlingen scenarios" part; it is just a way of organizing the definitions and files. The Aircraft definitions (e.g., AircraftTupolev) might just have well been placed with the definitions defining airports involved in the Überlingen scenarios, but were deemed as more globally useful for creating future NextGen Brahms models than for example the definition of staffing in the Zurich ATCC.

23.1.1 Agent groups

- AirTrafficApproachControlGroup
- AirTrafficControllerAssistantGroup
- AirTrafficControllerGroup
- AirTrafficControllerOfficerGroup
- AirTrafficGroup
- AirTrafficTerminalControlGroup
- FlightAttendantGroup
- FlightCrew
- FlightServiceGroup

- PilotGroup
- PilotTCASTrainedGroup
- RadarPlannerGroup
- RadioCommunicatorGroup
- TelephoneCommGroup

23.1.2 Conceptual object classes

- Airline
- AirSector
- Flight
- FlightModeAnnunicator
- FlightStatus
- GeneralizedFunction
- WeatherEvent

23.1.3 Geography areas

- AircraftArea
- Airport
- AirSpace
- AirRouteTrafficControlCenter
- AirTrafficTowerControl
- ATCCWorkstationArea
- Country City
- Runway
- Waypoint

23.1.4 Object classes

- AfssComputer
- Aircraft
- AircraftAirbus
- AircraftBoeing
- AircraftControls
- AircraftFlightInstruments
- AircraftGPSReceiver
- AircraftRadio
- AircraftTupelov
- AirSectorOperatingProcedure
- AirTrafficControlDisplay
- AirTrafficControlRadio
- AirTrafficControlServer
- AirTrafficControlSystem
- AirTrafficProximityAnalysis
- AutoFlightSystem
- ControlDisplayUnit
- DescentChecklist
- EngineIndicatorCrewAlertSystem
- FlightManagementSystem
- FlightPlan
- FlightPlanHostComputer
- FlightProgressStrip
- FlightSegment
- FlightStandardOperatingProcedure

- GPSData
- ModeControlPanel
- NavigationDisplay
- PrimaryFlightDisplay
- PrimarySurveillanceRadar
- Printer
- Radio
- ShortTermConflictAlertSystem
- StandardInstrumentDeparture
- StandardTerminalArrivalRoute
- Telephone
- TrafficCollisionAvoidanceSystem
- WeatherReport

23.2 Model Components Required for GUM Scenarios

These are Brahms component definitions required for modeling what occurred in the Zurich ATCC airspace during the late evening of July 1, 2002. Such components would of course be useful for modeling and simulating an infinite variety of scenarios involving any of these countries, airports, airlines, controllers, etc.

23.2.1 Agent groups

- AirTrafficControllers Austria
- AirTrafficControllers Belarus
- AirTrafficControllers Belgium
- AirTrafficControllers France
- AirTrafficControllers Germany
- AirTrafficControllers Italy
- AirTrafficControllers Poland
- AirTrafficControllers Russia
- AirTrafficControllers Spain
- AirTrafficControllers Switzerland
- FlightCrew AEF1135
- FlightCrew BTC2937
- FlightCrew DHX611
- FlightCrew OTHER

23.2.2 Conceptual objects

- Airline AEF
- Airline BTC
- Airline DHL
- Airline Other
- Flight AEF1135
- Flight BTC2937
- Flight DHX611
- Flight OTHER

23.2.3 Geography areas

- Airport Austria
- Airport Belarus
- Airport Belgium
- Airport France
- Airport Germany

- Airport Italy
- Airport Moscow
- Airport Poland
- Airport Spain
- AirSpace Austria
- AirSpace Belarus
- AirSpace Belgium
- AirSpace Czech Republic
- AirSpace Europe
- AirSpace France
- AirSpace Germany
- AirSpace Italy
- AirSpace Poland
- AirSpace Russia
- AirSpace Spain
- AirSpace Switzerland
- ARTCC Barcelona
- ARTCC Karlsruhe
- ARTCC Marseille
- ARTCC Minsk
- ARTCC Moscow
- ARTCC Munich
- ARTCC Vienna
- ARTCC Warsaw
- ARTCC Zurich
- Austria
- Belarus
- Belgium
- Czech Republic
- Eurocontrol Headquarters
- France
- Germany
- Italy
- Poland
- Russia
- Spain
- Switzerland

23.2.4 Objects

- Airbus320 AEF1135
- ATC Systems Austria
- ATC Systems Belarus
- ATC Systems Belgium
- ATC Systems Cologne
- ATC Systems France
- ATC Systems Germany Frankfurt
- ATC Systems Germany Friedrichshafen
- ATC Systems Germany Karlsruhe
- ATC Systems Germany Munich
- ATC Systems Italy
- ATC Systems Poland
- ATC Systems Russia

- ATC Systems Spain
- ATC Systems Swiss Zurich
- Boeing747 Other
- Boeing757 DHX611
- DHX611 Checklist
- Eurocontrol AFSS Computer
- FlightPlan AEF1135
- FlightPlan BTC2937
- FlightPlan DHX611
- FlightPlan EDDF EDNY
- FlightPlan LIME EBBR
- FlightPlan UUDD LEBL
- StandardOperatingProcedures EBBR
- StandardOperatingProcedures EDDF
- StandardOperatingProcedures EDNY
- StandardOperatingProcedures LEBL
- StandardOperatingProcedures LIME
- StandardOperatingProcedures UUDD
- Tupolev154 BTC2937

23.2.5 Workframes and thoughtframes

The following tables are the workframes and thoughtframes of the Air Traffic Controllers group in Brahms-GÜM.

Table 23-1 Air Traffic Controller Agents Workframes Priorities

Workframe	Priority	Agents
Get ATC Display Info *(like air sectors, etc.)*	100	RP, ARFA RE, CA
Monitor Crossing Planes Closing In *(like closest flight, separations, etc.)*	80	RP, ARFA RE, CA
Get Plane Info *(like latitude, longitude, heading, etc.)*	70	RP, ARFA RE, CA
Read Flight Progress Strip	50	RP, ARFA RE, CA
Monitor Plane Latitude	5	RP, ARFA RE, CA
Monitor Plane Longitude	5	RP, ARFA RE, CA
Monitor Plane Altitude	5	RP, ARFA RE, CA
Monitor Flight In Sector	0	RP, ARFA RE, CA
Start My Break	0	RP, ARFA RE, CA
Check Flights Loss Of Separation *(check closest flight, etc.)*	100	RP, ARFA RE
Confirm Flight In Sector	60	RP, ARFA RE
Hold Off Flight Unknown Sector *(pilot radio-in but not seen flight in display yet)*	60	RP, ARFA RE
Search Flight In Display	60	RP, ARFA RE
Agree Flight Level *(pilot request for flight level change)*	60	RP, ARFA RE
Resolve Flight Conflict *(radio flight to climb/descend)*	60	RP, ARFA RE
Informed Flight TCAS Climb *(monitor flight in display)*	50	RP, ARFA RE
Informed Flight TCAS Descent *(monitor flight in display)*	50	RP, ARFA RE
Acknowledge Flight Conflict Resolved *(pilot inform TCAS alert off)*	50	RP, ARFA RE
Check Flight Before Request Handoff	20	RP, ARFA RE
Request Handoff *(ask pilot to tune radio to another ATC)*	10	RP, ARFA RE
Get Flight Level Confirmation *(pilot confirm climbing/descending)*	10	RP, ARFA RE

Workframe	Priority	Agents
Request ATCO Take Over *(in order to take a break)*	10	RP, ARFA RE
Take Over From ATCO *(so other ATCO can go on a break)*	10	RP, ARFA RE
Handover Briefing *(between ATCOs)*	10	RP, ARFA RE
Track Flight In Conflict	8	RP, ARFA RE
Handling Other Flights	10	RP
Refuse Flight Descent Not Cleared *(after pilot radio in request)*	50	ARFA RE
Request Alternate Phone Number *(after failed 3rd call attempt)*	45	ARFA RE
Coordinate Flight Approach Airport	40	ARFA RE
Coordinate Flight Approach Alternate	40	ARFA RE
Discuss Sector Contact Alternatives *(with CA after call failures)*	40	ARFA RE
Report Alternate Phone Status *(to CA success/fail about phone call)*	30	ARFA RE
Flight Approach Cleared For Handoff *(after contact on phone)*	30	ARFA RE
Monitor Flight Descent Altitude	5	ARFA RE
Monitor Flight Descent Latitude	5	ARFA RE
Understand SID *(Standard Instrument Descent Procedure)*	0	ARFA RE
Understand STAR *(Standard Terminal Arrival Procedure)*	0	ARFA RE
Check Departure Runway Status	50	EDNY ATC
Check Arrival Runway Status	50	EDNY ATC
Transmit Flight Plan	50	EDNY ATC
Give Flight Departure Clearance	20	EDNY ATC
Give Flight Arrival Clearance	20	EDNY ATC
Coordinate Flight Approach Airport *(over the phone with ARFA RE)*	10	EDNY ATC
Handover Flight Progress Strip	50	CA
Search Flight In Display	20	CA
Alternate Phone Contact Request *(go get phone number, etc.)*	10	CA
Alternate Phone Contact Succeeded	10	CA
Alternate Phone Contact Failure *(go discuss alternatives, etc.)*	10	CA

Table 23-2 Air Traffic Controller Agents Thoughtframes

Agents	Thoughtframes
EDNY ATC	Flight En Route After Takeoff, Flight Descent To Airport, Flight Approach Runway No Handoff, Flight Landing On Runway, Flight Landing No Handoff
ARFA RE	Alternate Airport Phone Contact, Flight Descent To Airport, Flight Approaching Runway, Flight Pass TOD to Runway, Handoff Flight Approaching Runway
RP, ARFA RE	Informed Flight Conflict Resolved, Handoff Flight Exit Air Sector, Handoff Flight Leaving Max Latitude, Handoff Flight Leaving Min Latitude, Handoff Flight Leaving Max Longitude, Handoff Flight Leaving Min Longitude, Handoff Flight Below Min Altitude, Handoff Flight Above Max Altitude, Interrupt Phone Call TCAS Climb, Interrupt Phone Call TCAS Descent, Receive Pilot Greeting In Sector, Receive Pilot Agree Handoff, Receive Flight Agree Handoff
RP, ARFA RE, CA	Flight Enter Air Sector, Current Flight Strip Waypoint, Ignore Unknown Area Radio Comm, Ignore Not Work Area Radio Comm

The following tables are the workframes and thoughtframes of the Pilots group of different flights. "Other" refers to the generic flight that represents ongoing traffic to simulate the Zurich ATCO's workload in the Überlingen scenario.

Table 23-3 Pilot Agents Workframes Priorities

Workframe	Priority	Agents
Situate *(1st time in plane cockpit)*	100	DHL 611, BTC 2937, Other
Pre Flight Preparation	100	DHL 611, BTC 2937, Other
Control Communication To ATC	100	DHL 611, BTC 2937, Other
Control Communication From ATC	100	DHL 611, BTC 2937, Other
Get TCAS Traffic Alert Info	100	DHL 611, BTC 2937, Other
Get TCAS Resolution Alert Info	100	DHL 611, BTC 2937, Other
Tune Radio	60	DHL 611, BTC 2937, Other
Control Operating Procedures *(checklists, briefings, etc.)*	50	DHL 611, BTC 2937, Other
Control Flight Deck Components *(auto-pilot, altitude settings, etc.)*	50	DHL 611, BTC 2937, Other
Control Aircraft Information *(verify routes, waypoints, etc.)*	40	DHL 611, BTC 2937, Other
Control Heading	30	DHL 611, BTC 2937, Other
Control Aircraft Configuration *(flaps, gears & speed brakes)*	25	DHL 611, BTC 2937, Other
Control Airspeed	20	DHL 611, BTC 2937, Other
Control Flight Level	20	DHL 611, BTC 2937, Other
Control Vertical Speed	20	DHL 611, BTC 2937, Other
Takeoff From Runway	20	DHL 611, BTC 2937, Other
Land On Runway	20	DHL 611, BTC 2937, Other
Traffic Alert *(from TCAS)*	20	DHL 611, BTC 2937, Other
Control Vertical Profile	0	DHL 611, BTC 2937, Other
Control Waypoints	0	DHL 611, BTC 2937, Other
Monitor Route *(before given clearance by ATCO)*	0	DHL 611, BTC 2937, Other
Follow Increase Climb Advise *(from TCAS)*	100	DHL 611
Follow Increase Descent Advise *(from TCAS)*	100	DHL 611
Follow Initial Resolution Advisory	90	DHL 611
Monitor Resolution Advisory	70	DHL 611
Clear Of Conflict *(from TCAS)*	70	DHL 611
Monitor Traffic Alert	10	DHL 611

Table 23-4 Pilot Agents Thoughtframes

Agents	Thoughtframes
DHL 611, BTC 2937, Other	Flight Phase Preflight, Flight Phase Takeoff, Flight Phase En Route, Flight Phase Approach, Flight Phase Landing, Next SID Route, Next STAR Route, Next Flight Plan Route, Plane Taxi On Runway, Inform Flight In Sector Again, ATC Message For Other Flight

Agents	Thoughtframes
DHL 611	Ignore ATCO Climb Request, Ignore ATCO Climb Faster Request, Ignore ATCO Descend Request, Ignore ATCO Descend Faster Request

24 Appendix: Scenario Configurations

These model configurations—initial conditions formalized as properties of agents and objects—correspond to the ten scenarios (Table 8-1; Table 10-1) intended to sequentially and cumulatively add anomalies to a base scenario that served to define a complete scenario and by this sequence to test its generality. The simulation time in every case was 18:15 GMT on 07/01/2002, corresponding to the scheduled BTC 2937 departure time from Moscow at 18:45 GMT.

24.1 Configurations of Brahms Agents

Brahms agents are configured for each scenario based on their Group memberships. Each Brahms Group defines behaviors for all agents within the group. A Brahms Group can inherit behaviors from other Brahms Groups. For example, the "Air Traffic Approach Control Group" defines activities for handling flights, and its group members inherit additional activities for handling flights approaching an airport, which are not inherited by the "Radar Planner Group." Similarly, the "Pilot TCAS Trained Group" has activities for pilots to follow TCAS advisories that are additional to the behaviors inherited from the "Pilot Group."

Table 24-1 Air Traffic Controllers and Pilots Configuration for Alternative Brahms-GÜM Scenarios

Scenario Description	Brahms Agents	Brahms Group Membership
2A) Normal	ATCO Zurich AR RE	AirTrafficApproachControlGroup
	ATCO Zurich RP	RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup
2B) Normal w/o Phones	ATCO Zurich AR RE	AirTrafficApproachControlGroup
	ATCO Zurich RP	RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup
2C) Phones out & Radar degraded, but TCAS rules	ATCO Zurich AR RE	AirTrafficApproachControlGroup
	ATCO Zurich RP	RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup
2D) … but ZATC rules	ATCO Zurich AR RE	AirTrafficApproachControlGroup
	ATCO Zurich RP	RadarPlannerGroup
	Pilot BTC2937	PilotGroup
1A) Normal-SMOP	ATCO Zurich RP	AirTrafficApproachControlGroup, RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup

Scenario Description	Brahms Agents	Brahms Group Membership
1B) SMOP w/o Phones	ATCO Zurich RP	AirTrafficApproachControlGroup, RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup
1C) SMOP w/o Radar	ATCO Zurich RP	AirTrafficApproachControlGroup, RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup
1D) Actual, but TCAS Followed	ATCO Zurich RP	AirTrafficApproachControlGroup, RadarPlannerGroup
	Pilot BTC2937	PilotTCASTrainedGroup
1E) Überlingen	ATCO Zurich RP	AirTrafficApproachControlGroup, RadarPlannerGroup
	Pilot BTC2937	PilotGroup

24.2 Configurations of Brahms Objects

Brahms Objects are configured for each scenario based on the values assigned to their attributes in the initial model configuration. For example, for the scenario when Radar is degraded, STCA Zurich will also be turned off, which is modeled by setting the attribute isOpticalOn to false. Similarly, for scenarios where Phones are unavailable or calls are unsuccessful, phoneStatus is set to "busy."

Table 24-2 Radar and Phones Configuration in Alternative Scenarios

Scenario Description	Brahms Objects	Attributes	Values
2A) Normal	STCA Zurich	isOpticalOn	true
	ATC Phone Friedrichshafen ATCT	phoneStatus	free
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	free
2B) Normal w/o Phones	STCA Zurich	isOpticalOn	true
	ATC Phone Friedrichshafen ATCT	phoneStatus	busy
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	busy
2C) Phones out & Radar degraded, but TCAS rules	STCA Zurich	isOpticalOn	false
	ATC Phone Friedrichshafen ATCT	phoneStatus	busy
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	busy
2D) … but ZATC rules	STCA Zurich	isOpticalOn	false
	ATC Phone Friedrichshafen ATCT	phoneStatus	busy
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	busy
1A) Normal-SMOP	STCA Zurich	isOpticalOn	true
	ATC Phone Friedrichshafen ATCT	phoneStatus	free
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	free
1B) SMOP w/o Phones	STCA Zurich	isOpticalOn	true

	ATC Phone Friedrichshafen ATCT	phoneStatus	busy
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	busy
1C) SMOP w/o Radar	STCA Zurich	isOpticalOn	false
	ATC Phone Friedrichshafen ATCT	phoneStatus	free
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	free
1D) Actual, but TCAS Followed	STCA Zurich	isOpticalOn	false
	ATC Phone Friedrichshafen ATCT	phoneStatus	busy
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	busy
1E) Überlingen	STCA Zurich	isOpticalOn	false
	ATC Phone Friedrichshafen ATCT	phoneStatus	busy
	ATC Phone2 Friedrichshafen ATCT	phoneStatus	busy

25 Appendix: Brahms Simulation Graphics of System Interactions

This appendix provides several examples of Brahms-GÜM simulations presented in the Brahms AgentViewer, which displays the chronology of activities and associated workframes for selected agents and objects.

25.1 Pilot-Aircraft Operations

Brahms-GÜM simulates how pilots control an aircraft by the activities of pulling/pushing the cockpit control column, flipping switches or dialing selectors on Mode Control Panel (MCP). Pilots get information about performance of an aircraft by the activities of looking at Primary Flight Display (PFD) and Navigation Display (ND). Figure 25-1 shows an example timeline of DHL pilot activities during initial take off from the Bergamo airport runway. The pilot pulls back the control column to get the plane to climb, monitors the plane's climbing altitude on PFD, retracts landing gear, and continues monitoring altitude until plane reaches a certain flight level above the airport.

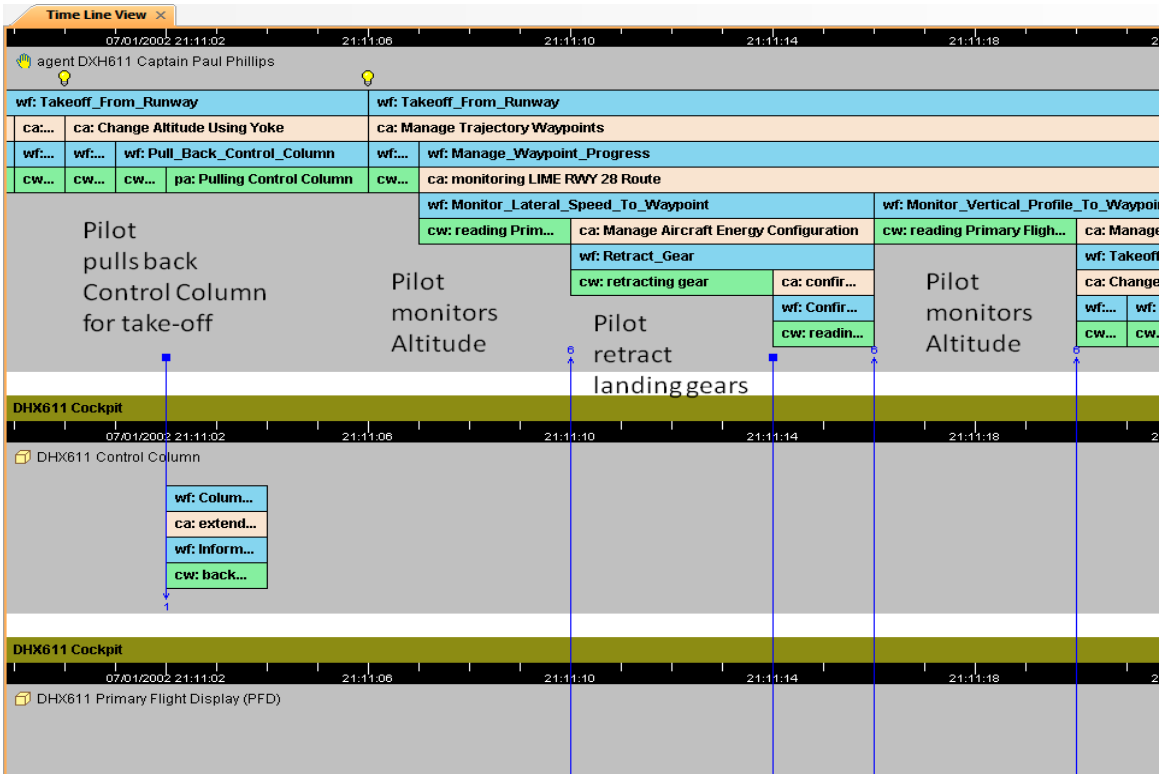
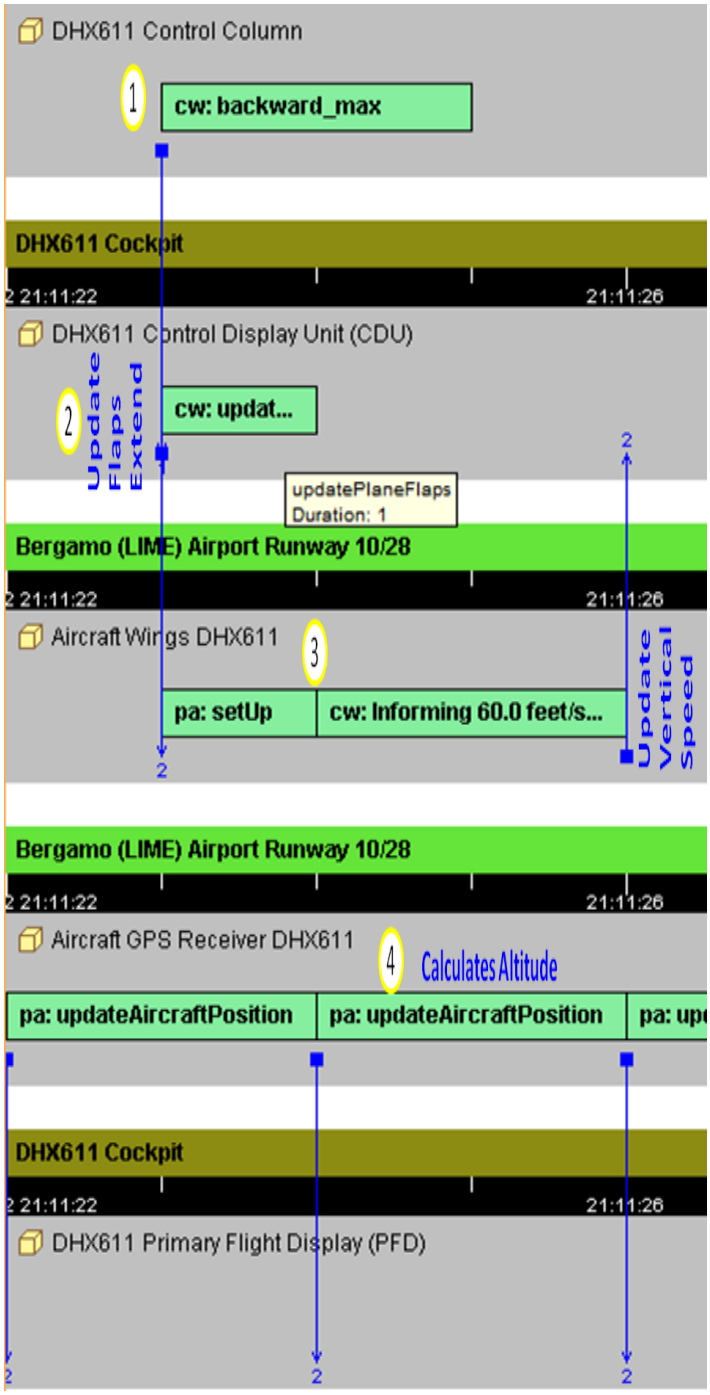


Figure 25-1: Pilot Takes-off from Bergamo Runway

Figure 25-2 provides a detailed view of what happens after the pilot pulls the control column in the DHL cockpit, which changes the vertical speed of the plane as it climbs above the runway. Data is communicated from Control Column to aircraft computer (Control Display Unit; CDU), which commands the aircraft wing flaps to extend. Extended flaps will accelerate vertical speed based on flaps climb/descend acceleration value (e.g., accelerate by 6 feet per second). The GPS receiver updates

longitude, latitude positions and altitude of plane based on both air and vertical speeds and sends plane's altitude and vertical speed to Primary Flight Display for the pilot to observe.



Blue lines indicate communications with numbers indicating how much data is being transferred.

1) *DHL pilot pulls back Control Column which causes Control Column to send info to CDU* —Column State = max backward.

2) *CDU converts that column state to extend flaps to maximum and informs Wings* – Flaps State = max extend, Rate of Display = 2 seconds (for how often to calculate acceleration of vertical speed).

3) *Wings sets up flaps to extended max & calculate next accelerated vertical speed* – vertical speed = 60 feet/second.

4) *GPS receiver **detects** vertical speed and calculates position of plane then sends info to PFD* – altitude = 1286 feet, vertical speed = 60 feet/second.

Figure 25-2: Detail View of Vertical Speed changes

Figure 25-3 shows that after the plane takes off, Captain Phillips changes the air speed from 150 kt to 350 kt by dialing the speed selector on the MCP and pushes the switch to "set". He then monitors air speed displayed in PFD to confirm that the plane is changing air speed. Captain Phillips then confirms the next waypoint, LIME NDB1, which is displayed in CDU; he directs the plane to move to this next waypoint.

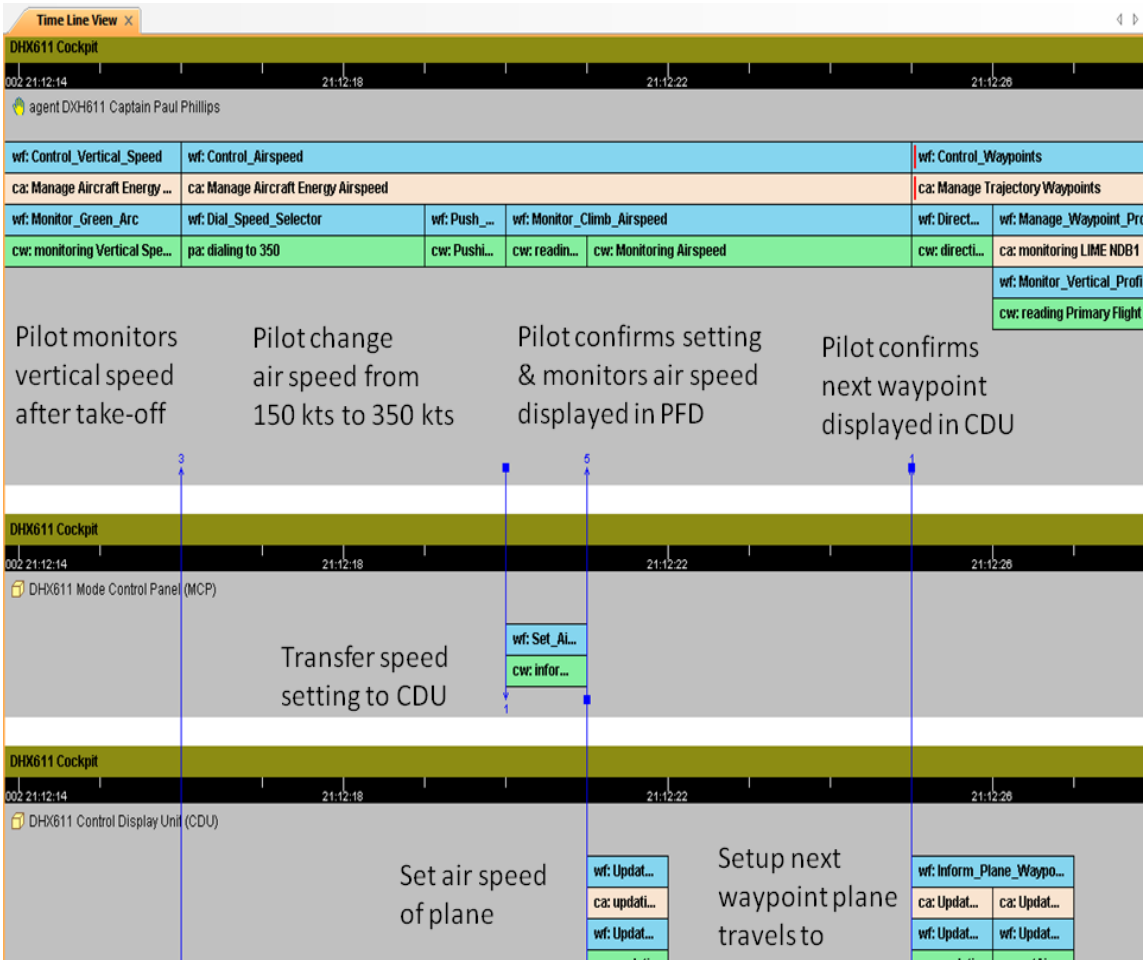


Figure 25-3: Pilot sets Air Speed to Next Waypoint

25.2 Radio Communications

All radios tuned to the same frequency receive the same communications in the Brahms-GÜM simulation. Figure 25-4shows communications between the BTC2937 Pilot/Cockpit and BTC plane's radio during handoff to another Air Traffic Controller. Communications only occur between the BTC pilot and ATC Zurich if the radios are operational and tuned to the same frequency. Also occurring in parallel are similar communications among BTC Radio, Munich ATC Radio, and Munich ATCO.

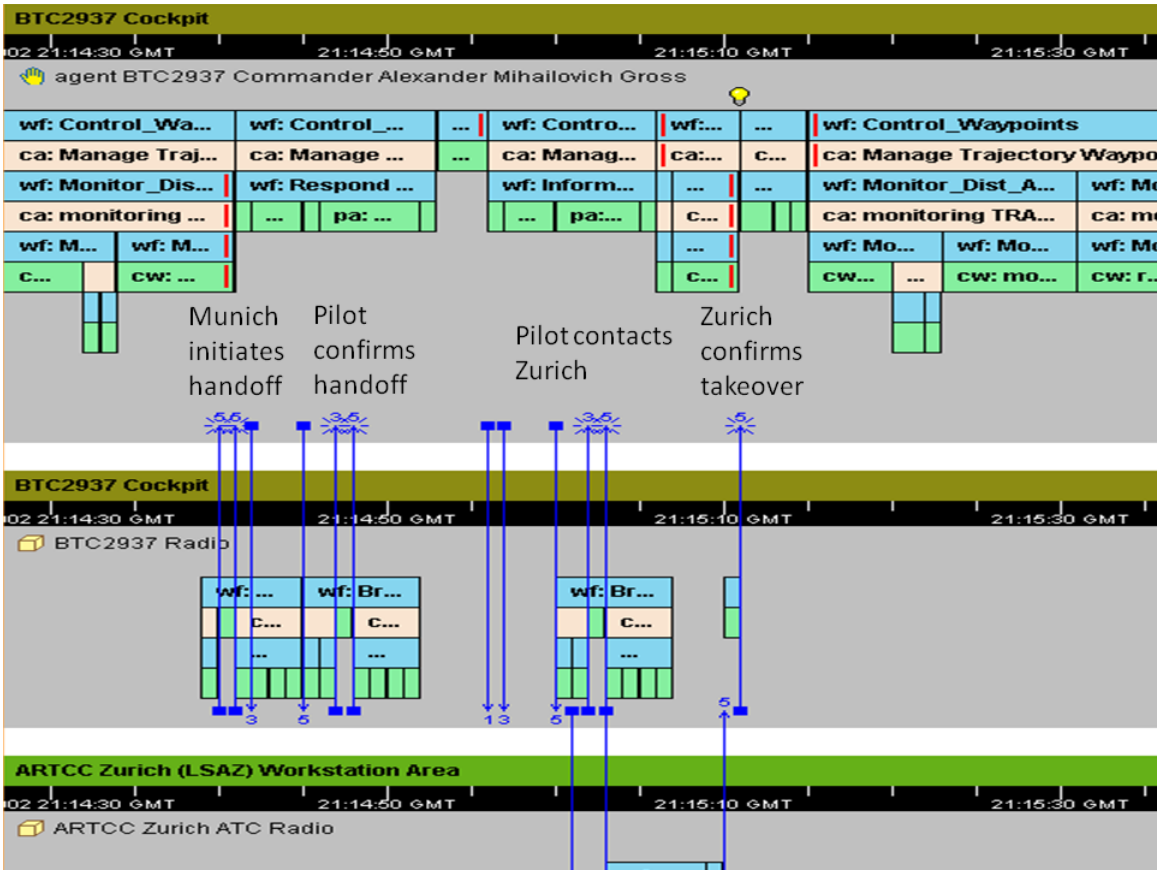
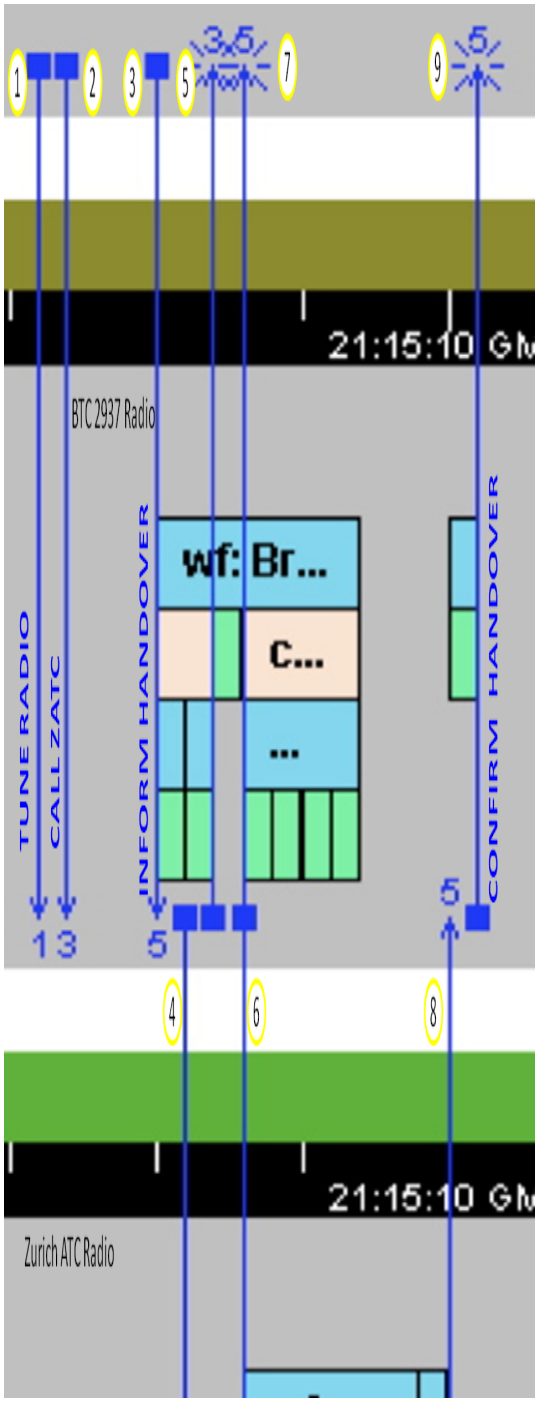


Figure 25-4: Munich Handoff Flight to Zurich

Figure 25-5 provides a detailed view of Pilot communications via the Radio during a particular handoff. The BTC Pilot tunes the radio to Zurich frequency, contacts Zurich ATCC to inform that his flight is entering the Zurich air sector, and receives acknowledgement from Zurich.



Blue lines indicate communications; numbers indicate how many attributes are being transferred

1, 2, & 3. BTC Pilot to his Radio (direct communication):
1) *BTC Pilot tunes radio to Zurich ATC* — frequency = 128.05
2) *BTC Pilot calls into Zurich ATC*— Pilot.flightNumber, Flight (BTC2937) & BTC2937.flightNumber
3) *BTC Pilot informs Zurich a/c is entering sector* —receiver (i.e., Zurich ATC), performative = **INFORM**, reason = flight, Medium = radio, commTime (for prioritization of responses)

4 & 6 BTC Radio transmits on frequency to Zurich radio
4) *BTC Radio relays to Zurich flight call* (i.e., relay #2)
6) *BTC Radio relays to Zurich* (i.e., relay #3)

5 & 7 BTC Radio broadcasts in cockpit (beliefs same as #2 & #3, a workaround so the co-pilot can know what Pilot told radio; see Section 28.3)

8. Zurich Radio transmits on frequency to BTC Radio:
Zurich ATC confirms handover—receiver (i.e., BTC Pilot, performative = **CONFIRM**, reason = flight, Medium = radio, commTime (for prioritization of responses)

9. BTC Radio broadcasts in cockpit (i.e., relay #8)

Figure 25-5: Detail View of Radio Communications

25.3 Radar Display and Monitoring

Primary Surveillance Radars (PSR) scan their associated air sectors for planes. An air sector is defined by its maximum and minimum latitude, longitude, and altitude. A plane's latitude, longitude, and altitude are updated by its GPS receiver as the plane flies toward a waypoint and the PSR checks to determine whether the plane's position is within its monitored air sector. Figure 25-6 shows the Tupolev plane, flight BTC 2937, flying toward the Trasadingen waypoint and its position being updated by its GPS Receiver. Zurich PSR scans its air sector and detects that the position of BTC 2937 is within its sector; so PSR informs Zurich Air Traffic Control (ATC) Server (not shown) that BTC 2937 has entered the Zurich air sector.

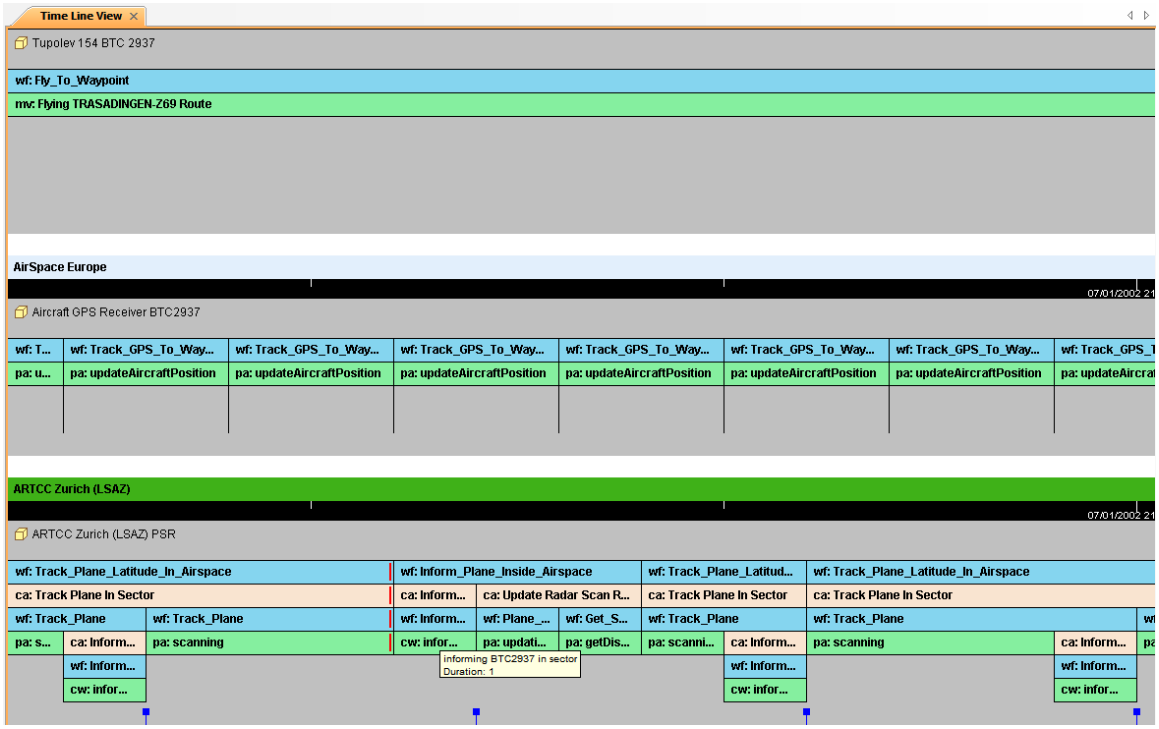
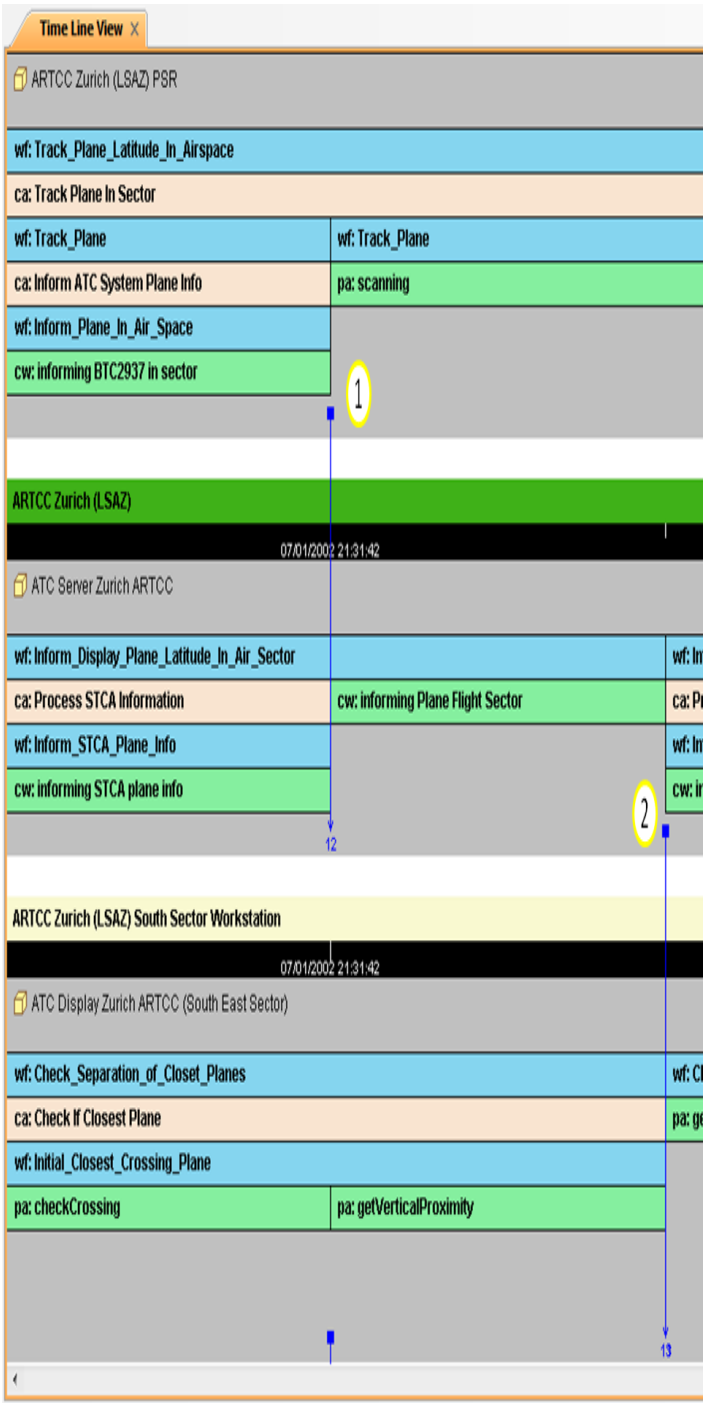


Figure 25-6: Zurich Radar Scans for Planes

Each Air Traffic Control (ATC) Display is set up with an ATC Server. ATC Displays may have smaller air sectors displayed; for example, Zurich's AFRA Display was displaying a smaller area within Zurich air space just above Friedrichshafen airport. Zurich ATC Server has information about this smaller "AFRA air sector" being displayed; so it only sends information about a plane to AFRA Display when a plane's position is within the AFRA air space.



Blue lines indicate communications; numbers indicate how many attributes are being transferred.

1) Zurich PSR detects BTC 2937 in its air space and sends plane info to Zurich ATC Server –*plane's geographic location, heading, bearing, longitude, latitude, altitude, airspeed, flight, flight number, waypoint, time to waypoint, and other planes within Zurich Air Sector.*

2) Zurich ATC Server sets BTC 2937's air sector to South-East sector and sends info to South-Sector Display - *plane's geographic location, heading, bearing, longitude, latitude, altitude, airspeed, flight, flight number, air sector, waypoint, time to waypoint, and other planes within South-East Sector.*

Figure 25-7: Detail View of Zurich Radar data to Display

Figure 25-7 shows that when BTC 2937 enters the Zurich air space, its identity and location information is transferred from the Zurich PSR to the ATC Server and then to the South-Sector Display. The plane's position is within the smaller South-East Sector defined for South Sector Display, located at South Sector Workstation; so the Server updates its record of BTC 2937 as being in this smaller air sector and sends the aircraft information to this display. When South-Sector Display receives the BTC 2937 information, it determines which plane is closest to BTC 2937 (lateral and vertical separations between planes) and whether the closest plane is on an intersecting flight path.

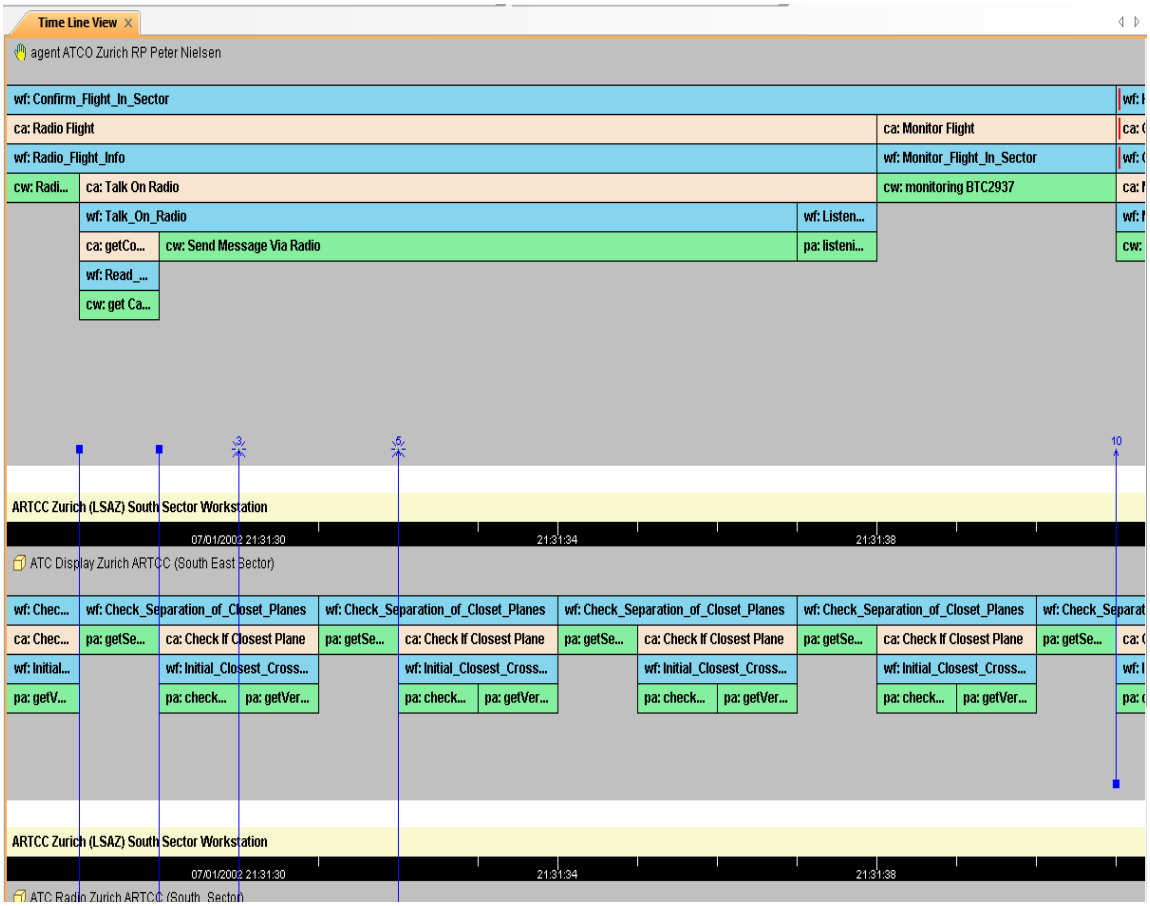


Figure 25-8: Zurich ATCO monitors BTC 2937

Figure 25-8 shows an example of an ATCO monitoring the radar display. Here the Zurich ATCO, Peter Neilson, responds to flight BTC 2937's call-in of having entered the Zurich air sector. Peter confirms that he sees the BTC 2937 displayed on his radar display and monitors the flight in the Radar display.

25.4 TCAS Operation

TCAS calculates range and vertical tau values, which are the number of seconds until lateral or vertical collision for a plane within same air sector. Based on TCAS rules and calculated tau values, a Traffic Alert ("Traffic! Traffic!") and/or Resolution Alert ("Climb! Climb!" or "Descend! Descend!") is broadcast to pilots. TCAS also sends

range separation, vertical separation, and heading of conflicting plane to the Navigation Display (ND), representing information that the real-world ATCO can detect when looking at the display.

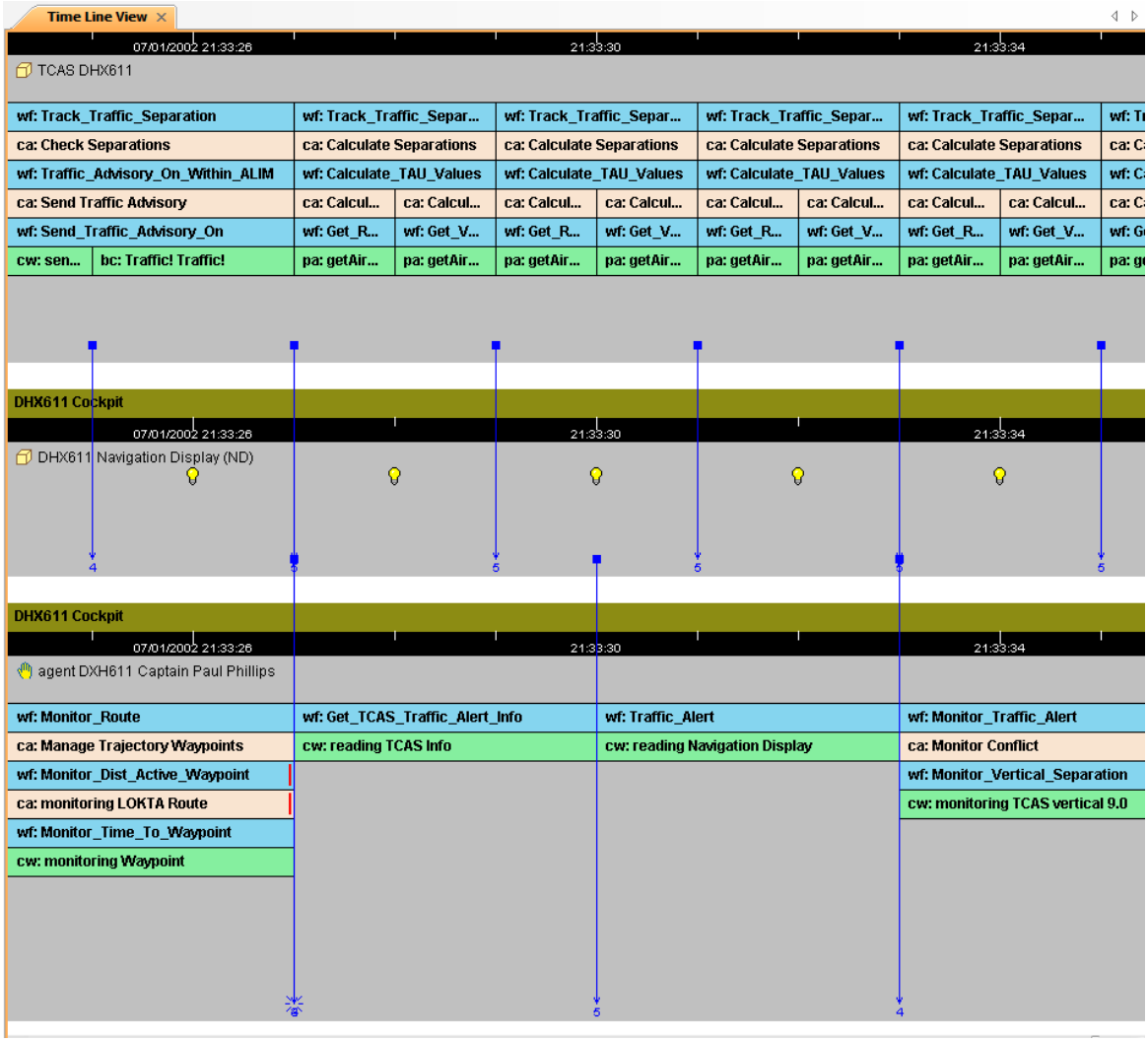


Figure 25-9: TCAS issues Traffic Alert

Figure 25-9 shows what happens when TCAS broadcasts “Traffic! Traffic!” in the DHL 611 cockpit. Captain Phillips immediately reads the plane range separation, vertical separation, and heading from ND. Captain Phillips also monitors vertical separation between his plane and BTC 2937 by monitoring ND.

Figure 25-10 shows what happens when TCAS broadcasts “Descend! Descend!” in the DHL 611 cockpit. Captain Phillips looks for the flight altitude to which to descend provided by TCAS and displayed in the PFD. Captain Phillips then pushes the Control Column forward to start plane descent.

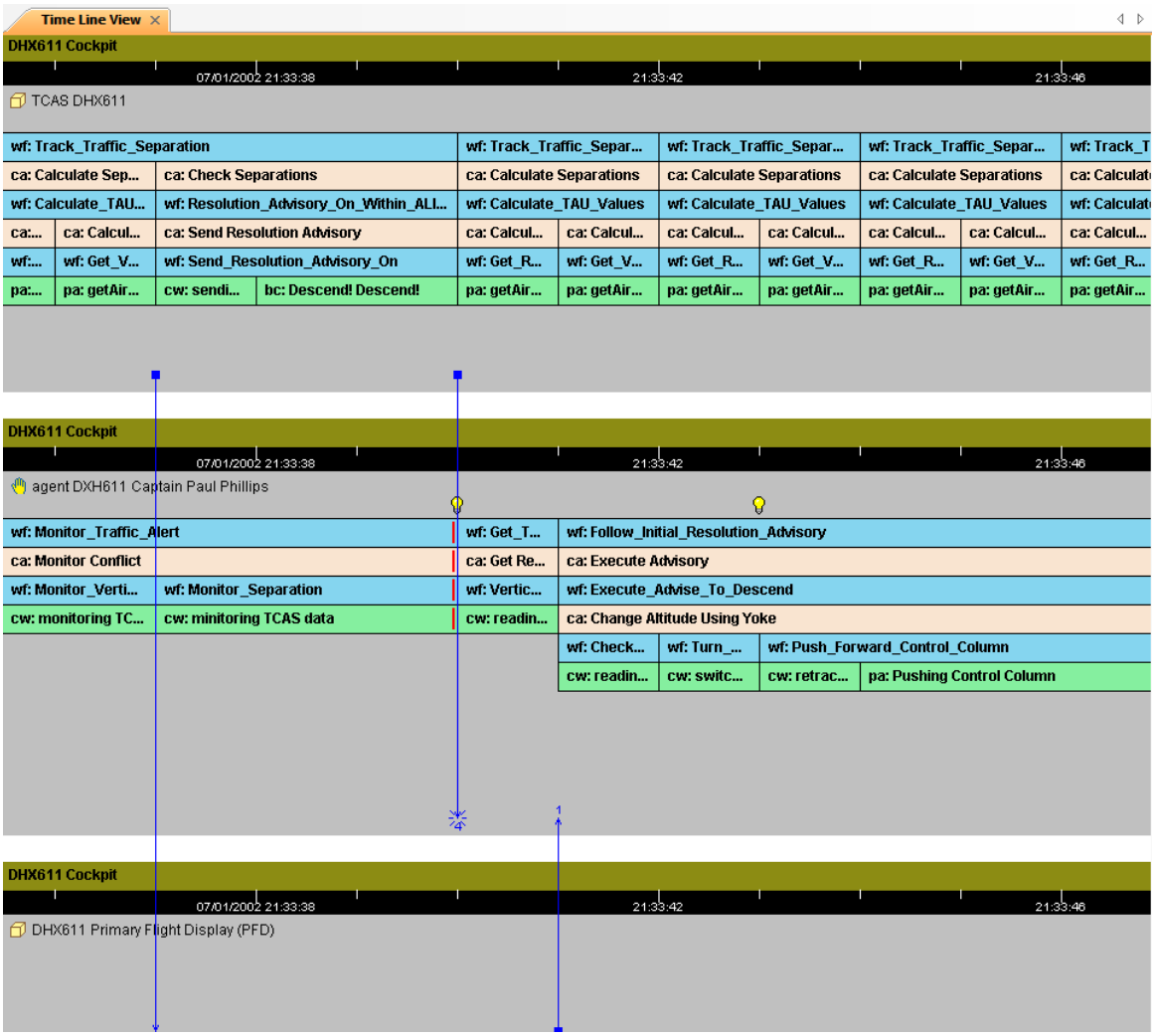


Figure 25-10: TCAS issues Descend Resolution Alert

Figure 25-11 shows the events in the Überlingen scenario how TCAS issues advice to DHL 611 to descend and gives BTC 2937 the inverse advice to climb to a higher altitude.

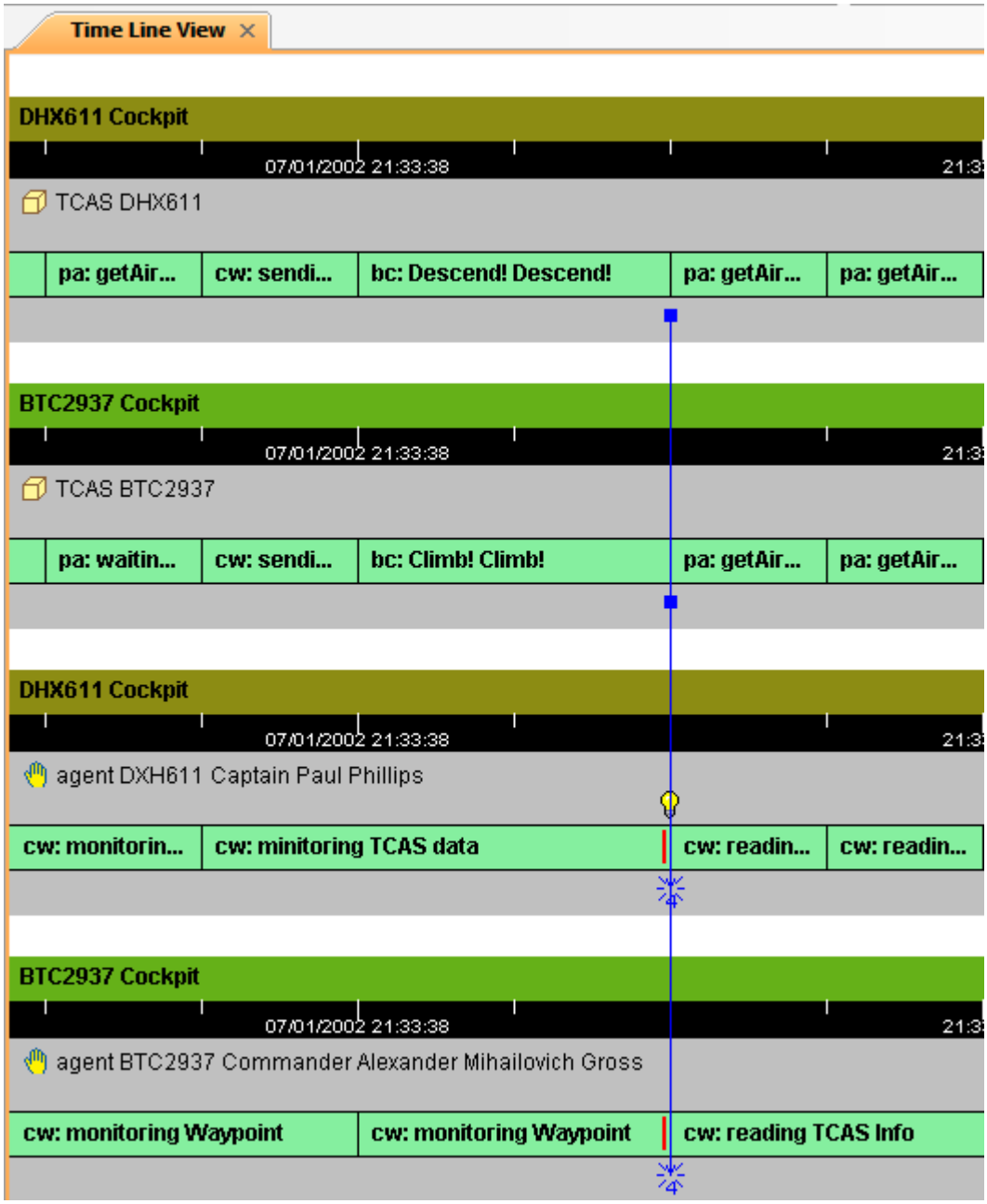


Figure 25-11: TCAS issues contrary Resolution Alerts

25.5 ATC-Pilot Communications

Figure 25-12 shows that after the DHL pilot, Captain Phillips, has completed handoff on entering the Zurich air sector, he requests a change of flight level from 26,000 feet to 36,000 feet and is given clearance by Zurich air traffic control.

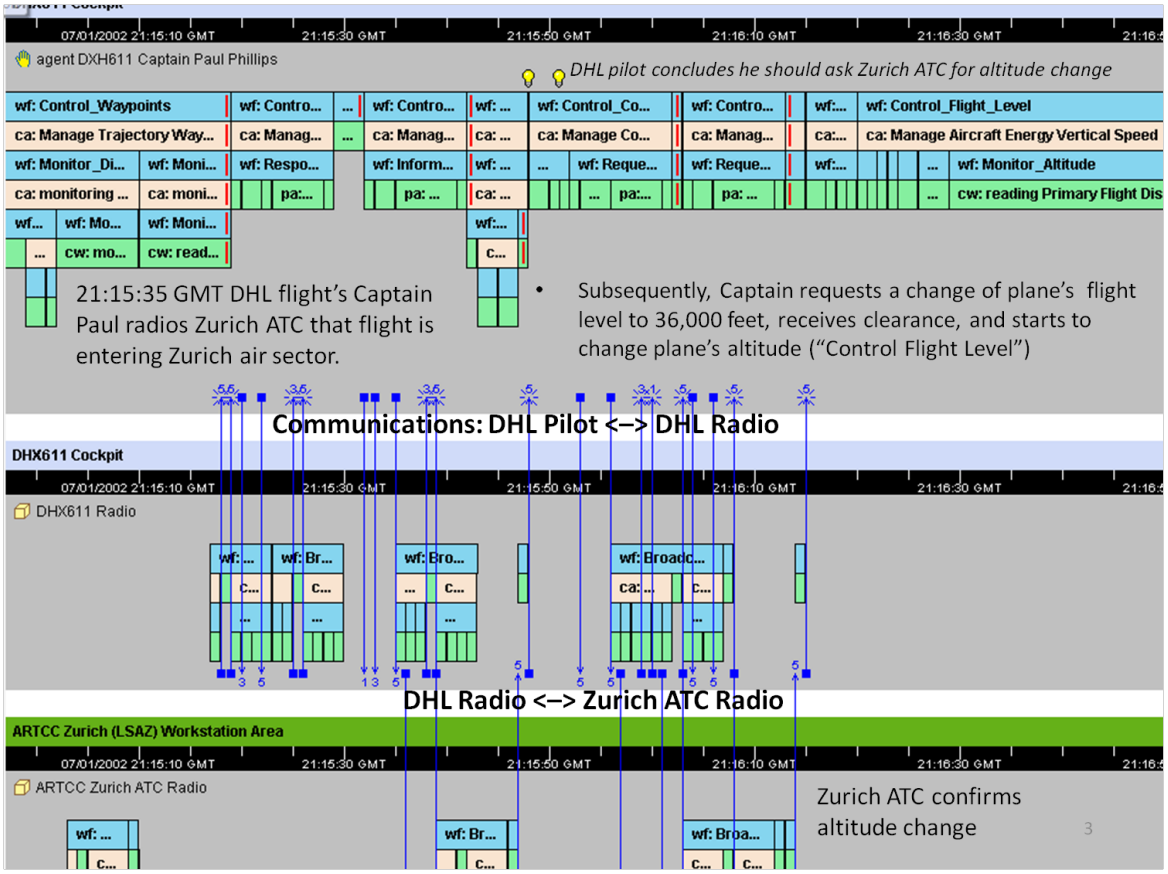


Figure 25-12: DHL Pilot Requests Flight Level Change

26 Appendix: Example Log of Brahms Simulation Run

This is an excerpt of a log file created during the simulation of Brahms-GÜM with the Überlingen scenario configuration (designated 1F in Table 10-1; refinements were made subsequently, as described in that chapter). The excerpt is annotated to indicate what general events are occurring.

The late-arriving AEF flight enters the air sector of the Friedrichshafen airport. ATCO in the control tower reads the flight strip, looks for the flight in the radar display, and learns about the route from the flight strip.

[ATC_Server_Friedrichshafen_ATCT] - 21:18:07 : Flight AEF1135 in Air Sector Friedrichshafen Airport

[ATCO_Friedrichshafen_Airport] - 21:18:10 : Wait for Flight Strip printout
[ATCO_Friedrichshafen_Airport] - 21:18:16 : Reading Flight Strip
[ATCO_Friedrichshafen_Airport] - 21:18:17 : Search for AEF1135 in display
[ATCO_Friedrichshafen_Airport] - 21:18:17 : Search for flight in display
[ATCO_Friedrichshafen_Airport] - 21:18:40 : Read AEF1135 Flight Strip route

DHL flight enters Zurich south air sector

[ATC_Server_Zurich_ARTCC] - 21:21:49 : Flight DXH611 in Air Sector Zurich South ARTCC

Zurich Controller Assistant takes DHL flight strip to ATCO who then examines the radar.

[Zurich CA]- 21:22:06 : Moving to ARTCC Zurich (LSAZ) South Workstation
[Zurich CA]- 21:22:08 : Handover DXH611 Flight Strip
[ATCO_Zurich_RP] - 21:22:08 : Reading Flight Strip
[ATCO_Zurich_RP] - 21:22:12 : Search for DHX611 in display
[ATCO_Zurich_RP] - 21:22:12 : Search for flight in display
[Zurich CA]- 21:22:12 : Moving to ARTCC Zurich (LSAZ) CA Workstation
[ATCO_Zurich_RP] - 21:22:28 : Search for flight in display

Meanwhile another (unnamed) flight enters the Zurich sector; ATCO detects the DHL and reads the information, then detects the other flight and interacts with it. This designates activity "handling other flights" that is modeling arrival and handoff without details that occur for BTC, DHL, and AEF flights, which are modeled individually. This "other flight" is modeled as something like a helicopter that shows up in the airspace in one location that ATCO interacts with every minute.

[ATC_Server_Zurich_ARTCC] - 21:22:38 : Flight OTHER in Air Sector Zurich East ARTCC
[ATCO_Zurich_RP] - 21:22:49 : Read DXH611 Flight Strip route
[ATCO_Zurich_RP] - 21:22:54 : monitoring DXH611
[B757] - 21:23:01 : Flight DXH611 Flying LOKTA Route
[B757] - 21:23:01 : Flight DXH611 Heading North
[B757] - 21:23:01 : AirSpeed: 463, Vertical Speed: 37
[B757] - 21:23:01 : Altitude: 16295.0

[ATCO_Zurich_RP] - 21:23:23 : monitoring OTHER
[ATCO_Zurich_RP] - 21:23:26 : Radio Flight OTHER REQUEST chat
[ATCO_Zurich_RP] - 21:23:54 : monitoring DXH611

BTC flight strip is similarly printed, conveyed, and printed. ATCO handles several other flights.
[ATC_Server_Zurich_ARTCC] - 21:24:10 : Flight BTC2937 in Air Sector Zurich ARTCC
[ATCO_Zurich_RP] - 21:24:11 : monitoring OTHER
[ATCO_Zurich_RP] - 21:24:13 : Radio Flight OTHER REQUEST chat
[Zurich CA]- 21:24:14 : Wait for Flight Strip printout
[Zurich CA]- 21:24:22 : Reading Flight Strip
[Zurich CA]- 21:24:24 : Search for BTC2937 in display
[Zurich CA]- 21:24:24 : Search for flight in display
[ATCO_Zurich_RP] - 21:24:36 : monitoring OTHER
[Zurich CA]- 21:24:40 : Search for flight in display
[ATCO_Zurich_RP] - 21:24:56 : monitoring OTHER
[Zurich CA]- 21:24:57 : Search for flight in display
[ATCO_Zurich_RP] - 21:25:12 : monitoring OTHER
[Zurich CA]- 21:25:15 : Read BTC2937 Flight Strip route
[ATCO_Zurich_RP] - 21:25:16 : Radio Flight OTHER REQUEST chat
[Zurich CA]- 21:25:17 : Read BTC2937 Flight Strip route
[ATCO_Zurich_RP] - 21:25:44 : monitoring OTHER

DHL exists Bergamo Airport sector and receives information about handoff to Zurich.
[ATC_Display_Bergamo_Airport] - 21:25:51 : Flight DXH611 exits displayed sector
[ATCO_Bergamo_Airport] - 21:25:52 : Radio Flight DXH611 REQUEST handoff
[A320] - 21:25:54 : Flight AEF1135 Flying TURKHEIM Route
[A320] - 21:25:54 : Flight AEF1135 Heading South-West
[A320] - 21:25:54 : AirSpeed: 320, Vertical Speed: 0
[A320] - 21:25:54 : Altitude: 9997.0
[Pilot_DHX611] - 21:26:01 : Flight DXH611 AGREE handoff
[ATCO_Zurich_RP] - 21:26:05 : monitoring DXH611

BTC now appears in Karlsruhe radar sector. ATCO handles another flight; DHL pilot tunes to Zurich sector, informs ATCO of arrival, and ATCO confirms handoff. CA provides AEF flight strip.
[ATC_Server_Karlsruhe_ARTCC] - 21:26:10 : Flight BTC2937 in Air Sector Karlsruhe ARTCC
[ATCO_Zurich_RP] - 21:26:13 : monitoring OTHER
[Pilot_DHX611] - 21:26:16 : tuning radio to 128.05
[ATCO_Zurich_RP] - 21:26:17 : Radio Flight OTHER REQUEST chat
[Pilot_DHX611] - 21:26:18 : Flight DXH611 INFORM flight

[ATCO_Zurich_RP] - 21:26:39 : Radio Flight DXH611 CONFIRM flight
[Zurich CA]- 21:26:42 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[Zurich CA]- 21:26:46 : Handover AEF1135 Flight Strip
[Zurich CA]- 21:26:48 : Moving to ARTCC Zurich (LSAZ) CA Workstation

ATCO sees DHL in display; DHL requests FL360; ATCO goes to ARFA (right workstation) to read AEF flight strip, then returns to left workstation to responds to DHL, confirms change, and marks the flight strip accordingly. Aircraft-to-ATCC radio communication takes 3 seconds before beliefs get transferred, thus ATCO didn't "hear" request to change flight level before noticing AEF flight control strip at ARFA workstation area (placed by the CA).
[ATCO_Zurich_RP] - 21:26:58 : monitoring DXH611
[Pilot_DHX611] - 21:27:01 : Flight DXH611 REQUEST flightLevel
[Pilot_DHX611] - 21:27:01 : Flight Level 36000.0
[ATCO_Zurich_RP] - 21:27:02 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[ATCO_Zurich_RP] - 21:27:05 : Reading Flight Strip
[ATCO_Zurich_RP] - 21:27:08 : Search for AEF1135 in display
[ATCO_Zurich_RP] - 21:27:08 : Search for flight in display
[ATCO_Zurich_RP] - 21:27:10 : Moving to ARTCC Zurich (LSAZ) South Workstation
[ATCO_Zurich_RP] - 21:27:12 : Radio Flight DXH611 AGREE flightLevel
[B757] - 21:27:27 : Vertical Speed: 0
[B757] - 21:27:27 : Altitude: 26063.0
[ATCO_Zurich_RP] - 21:27:30 : Update DXH611 LOKTA Route on flight strip

Returning to ARFA workstation, ATCO reads AEF flight strip, recognizes that the route requires his coordination with control tower, and attempts handoff by phone
[ATCO_Zurich_RP] - 21:27:31 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[ATCO_Zurich_RP] - 21:27:32 : Read AEF1135 Flight Strip route
[ATCO_Zurich_RP] - 21:27:42 : Pick-up & Dialing Phone

This is the workframe for the phone attempt…

```
// group AirTrafficApproachControlGroup
workframe Coordinate_Flight_Approach_Airport {
        priority: 40;
        variables:          // omitted here for brevity
        when(knownval(current.radioUsed = unknown) and
                knownval(current.phoneUsed = unknown) and
                knownval(flight.status = FlightStatus_Descent) and
                knownval(flight.cleared = false) and
                knownval(flight.handoff = false) and
                knownval(flight.handoffCompletedOn = unknown) and // handoff has not occurred
                knownval(star.runway = flight.runwayArrival) and
                knownval(airportSector = star.airportAirSpace) and
                knownval(myPhone.location = current.location) and
                knownval(airportPhone = airportSector.phoneNumber) and
                knownval(current.numberRetriesLeft > 0) and          // initial belief allows three attempts
```

```
            knownval(attempts = current.numberRetriesLeft - 1))
        do {
            conclude((current.generalizedFunction = Manage_Communication), fc:0);
            conclude((current.commReason = "approach handoff"), fc:0);
            conclude((current.commPerformative = "REQUEST"), fc:0);
            coordinateFlightApproach(flight, airportSector, myPhone, airportPhone);
            conclude((current.numberRetriesLeft = attempts), fc:0);
        }
}//wf Coordinate_Flight_Approach
```

Meanwhile BTC flight enters the Zurich sector; CA provides the flight strip. ATCO's call is unsuccessful; he hangs up and returns to left workstation to read the BTC flight strip.

[ATC_Server_Zurich_ARTCC] - 21:27:55 : Flight BTC2937 in Air Sector Zurich East ARTCC
[ATCO_Zurich_RP] - 21:28:03 : Hang-up Phone
[ATCO_Zurich_RP] - 21:28:09 : Pick-up & Dialing Phone
[Zurich CA]- 21:28:11 : Moving to ARTCC Zurich (LSAZ) South Workstation
[Zurich CA]- 21:28:15 : Handover BTC2937 Flight Strip
[Zurich CA]- 21:28:18 : Moving to ARTCC Zurich (LSAZ) CA Workstation
[ATCO_Zurich_RP] - 21:28:28 : Hang-up Phone

ATCO's gives priority to reading flight strip over trying the phone call again.

[ATCO_Zurich_RP] - 21:28:29 : Moving to ARTCC Zurich (LSAZ) South Workstation
[ATCO_Zurich_RP] - 21:28:31 : Reading Flight Strip
[ATCO_Zurich_RP] - 21:28:33 : Search for BTC2937 in display
[ATCO_Zurich_RP] - 21:28:33 : Search for flight in display
[ATCO_Zurich_RP] - 21:29:02 : Read BTC2937 Flight Strip route

AEF flight progresses to next segment. ATCO attempts to call Friedrichshafen again, fails, and handles another flight.

[A320] - 21:29:06 : Flight AEF1135 Flying KEMPTEN Route
[A320] - 21:29:06 : Flight AEF1135 Heading South-South-West
[A320] - 21:29:06 : AirSpeed: 320, Vertical Speed: -26
[A320] - 21:29:06 : Altitude: 9841.0
[ATCO_Zurich_RP] - 21:29:17 : Pick-up & Dialing Phone

The Karlsruhe ATCO provides information to AEF for handoff to Zurich. Zurich ATCO's call fails again; he handles another flight.

[ATCO K]- 21:29:32 : Radio Flight AEF1135 REQUEST handoff
[ATCO_Zurich_RP] - 21:29:33 : Hang-up Phone
[ATCO_Zurich_RP] - 21:29:36 : monitoring OTHER
[ATCO_Zurich_RP] - 21:29:39 : Radio Flight OTHER REQUEST chat
[Pilot_AEF1135] - 21:29:44 : Flight AEF1135 AGREE handoff

AEF flight arrives in Zurich ARFA sector; AEF informs ATCO of arrival, who tells him to wait then confirms handoff after flight appears in ARFA radar display.

[ATC_Server_Zurich_ARTCC] - 21:29:52 : Flight AEF1135 in Air Sector Zurich ARFA ARTCC
[Pilot_AEF1135] - 21:29:58 : tuning radio to 119.92
[Pilot_AEF1135] - 21:30:00 : Flight AEF1135 INFORM flight
[ATCO_Zurich_RP] - 21:30:08 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[ATCO_Zurich_RP] - 21:30:10 : Radio Flight AEF1135 INFORM hold_off
[ATCO_Zurich_RP] - 21:30:31 : Search for flight in display
[ATCO_Zurich_RP] - 21:30:55 : Radio Flight AEF1135 CONFIRM flight
[ATCO_Zurich_RP] - 21:31:20 : monitoring AEF1135
[ATCO_Munich_ARTCC] - 21:31:23 : Radio Flight BTC2937 REQUEST handoff

ATCO moves out of work area where radar display is located to higher-level area to tell (Brahms "broadcast," an outloud utterance) CA that alternate airport phone number is needed.
[ATCO_Zurich_RP] - 21:31:23 : Moving out of work area

BTC exits Munich sector
[ATC_Display_Munich_ARTCC] - 21:31:27 : Flight BTC2937 exits displayed sector

[A320] - 21:31:27 : Flight AEF1135 Flying TOD Friedrichshafen Route
[A320] - 21:31:27 : Flight AEF1135 Heading West-South-West
[A320] - 21:31:27 : AirSpeed: 300, Vertical Speed: -26
[A320] - 21:31:27 : Altitude: 7768.0

After three phone attempts, ATCO requests an alternate number from the CA.
[ATCO_Zurich_RP] - 21:31:28 : REQUEST alternate_phone
[ATCO_Zurich_RP] - 21:31:30 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[ATCO_Zurich_RP] - 21:31:32 : Moving to ARTCC Zurich (LSAZ) South Workstation
[Zurich CA]- 21:31:33 : Moving to ARTCC Zurich (LSAZ) Manager Workstation

ATCO handles another flight while BTC receives information for Zurich handoff, informs Zurich ATCO, who confirms.
[ATCO_Zurich_RP] - 21:31:34 : monitoring OTHER
[Pilot_BTC2937] - 21:31:36 : Flight BTC2937 AGREE handoff
[Zurich CA]- 21:31:38 : Moving to ARTCC Zurich (LSAZ) South Workstation
[ATCO_Zurich_RP] - 21:31:38 : Radio Flight OTHER REQUEST chat
[Zurich CA]- 21:31:41 : PROPOSE alternate_phone
[Pilot_BTC2937] - 21:31:51 : tuning radio to 128.05
[Zurich CA]- 21:31:51 : Moving to ARTCC Zurich (LSAZ) CA Workstation
[Pilot_BTC2937] - 21:31:53 : Flight BTC2937 INFORM flight

[ATCO_Zurich_RP] - 21:32:03 : Radio Flight BTC2937 CONFIRM flight
[B757] - 21:32:12 : Vertical Speed: 0
[B757] - 21:32:12 : Altitude: 36041.0
[ATCO_Zurich_RP] - 21:32:24 : monitoring BTC2937

ATCO attempts to use the alternate number provided by the CA and fails. DHL arrives in Zurich and Karlsruhe radar display sectors. ATCO moves back to handle AEF flight, then returns to left workstation to handle another flight. Karlsruhe receives the DHL flight strip.
[ATCO_Zurich_RP] - 21:32:33 : Pick-up & Dialing Phone
[ATC_Server_Zurich_ARTCC] - 21:32:41 : Flight DXH611 in Air Sector Zurich East ARTCC
[ATCO_Zurich_RP] - 21:32:51 : Hang-up Phone
[ATC_Server_Karlsruhe_ARTCC] - 21:32:55 : Flight DXH611 in Air Sector Karlsruhe ARTCC
[ATCO_Zurich_RP] - 21:32:55 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[ATCO_Zurich_RP] - 21:32:57 : monitoring AEF1135
[ATCO_Zurich_RP] - 21:32:59 : Moving to ARTCC Zurich (LSAZ) South Workstation
[ATCO_Zurich_RP] - 21:33:01 : monitoring OTHER
[ATCO_Zurich_RP] - 21:33:04 : Radio Flight OTHER REQUEST chat
[ATCO Karlsruhe]- 21:33:05 : Wait for Flight Strip printout
[ATCO Karlsruhe]- 21:33:07 : Search for DXH611 in display
[ATCO Karlsruhe]- 21:33:07 : Reading Flight Strip
[ATCO Karlsruhe]- 21:33:09 : Read DHX611 Flight Strip route

Following simulates that ATCO could apprehend separation problem if he monitors the left workstation radar display. However, he is first busy handling another flight handoff…
[ATCC Z Display SE]- 21:33:10 : BTC - DHL lateral separation: 26.46
[ATCC Z Display SE]- 21:33:18 : BTC - DHL lateral separation: 25.27
[ATCO_Zurich_RP] - 21:33:25 : monitoring OTHER
[ATCO_Zurich_RP] - 21:33:27 : Radio Flight OTHER REQUEST chat
[ATC_Display_Munich_ARTCC] - 21:33:42 : Flight AEF1135 exits displayed sector
[ATCC Z Display SE]- 21:33:45 : BTC - DHL lateral separation: 19.36

With handoff of AEF to Friedrichshafen still most urgent, ATCO does not detect the separation issue, but rather moves to ARFA workstation and suggests that AEF simply contact the control tower (thus skipping protocol step of ATCO informing tower first). AEF agrees and proceeds.
[ATCO_Zurich_RP] - 21:33:49 : Moving to ARTCC Zurich (LSAZ) ARFA Workstation
[ATCO_Zurich_RP] - 21:33:54 : Radio Flight AEF1135 REQUEST handoff

[Pilot_AEF1135] - 21:33:56 : Flight AEF1135 REQUEST arrival
[Pilot_AEF1135] - 21:34:08 : Flight AEF1135 AGREE handoff
[ATCC Z Display SE]- 21:34:09 : BTC - DHL lateral separation: 15.25

The higher priority handoff complete, ATCO returns to left workstation and monitors the radar, finally detecting the loss of separation between the DHL and BTC (now less than 15 nm).

[ATCO_Zurich_RP] - 21:34:20 : Moving to ARTCC Zurich (LSAZ) South Workstation
[Pilot_AEF1135] - 21:34:22 : tuning radio to 124.35
[ATCO_Zurich_RP] - 21:34:22 : monitoring BTC2937
[Pilot_AEF1135] - 21:34:24 : Flight AEF1135 INFORM flight
[ATCO_Zurich_RP] - 21:34:26 : See Loss of Separation between DXH611 and BTC2937

This is the workframe for checking loss of separation (logic is discussed in Section 9.2)…

```
// group AirTrafficControllerOfficerGroup
workframe Check_Flights_Loss_Of_Separation {
        priority: 100;
        variables:          // omitted for brevity
        when(knownval(current.radioUsed = unknown) and
                knownval(current.phoneUsed = unknown) and
                knownval(plane.flight = flight) and
                known(flight.sectorFrequency ) and
                unknown(flight.flightInBoundary ) and
                knownval(flight.isFlightClosestCrossing = true) and
                knownval(flight.flightClosest = otherFlight) and
                knownval(otherPlane.flight = otherFlight) and
                known(otherFlight.sectorFrequency ) and
                knownval(flightNum = flight.flightNumber) and
                knownval(otherFlightNum = otherFlight.flightNumber) and
                knownval(atcDisplayLocation = atcDisplay.location) and
                knownval(atcDisplay airSectors flight.airSector) and
                knownval(atcDisplay.minLateralSeparation > flight.flightLateralSeparation) and
                knownval(atcDisplay.minVerticalSeparation > flight.flightVerticalSeparation))
        do {
                println WithSimTime_ss("Check Loss of Separation between %1 and %2", flightNum,
otherFlightNum);
                monitorPlanesInConflict(plane, flight, atcDisplayLocation);
        }
}//wf Check_Flights_Loss_Of_Separation
```

In this simulation run, choice of flight to inform was random; ATCO chose to contact DHL and based on flight route radios for them to descend 1500 feet.

[ATCO_Zurich_RP] - 21:34:26 : Reading DXH611 flight strip
[ATCO_Zurich_RP] - 21:34:28 : Request DXH611 Descend to next route flight level

Workframe that determines instruction to descend based on control strip information

(logic is discussed in Section 9.2)…

```
// group AirTrafficControllerOfficerGroup, composite_activity resolvePlanesInConflict
workframe Request_Descend_Next_Flight_Level {
	variables:
		forone(FlightProgressStrip) strip;
		forone(BaseAreaDef) stripLoc;
		forone(FlightPlan) plan;
		forone(FlightSegment) route;
		forone(FlightSegment) nextRoute;
		forone(double) fl;
		forone(double) newFL;
	when(knownval(current.commReason != "descend") and
		knownval(current.commReason != "descend_faster") and
		knownval(current.commReason != "climb") and
		knownval(current.commReason != "climb_faster") and
		knownval(pilot.commReason != "tcas_descent") and
		knownval(route = flight.route) and // current flight segement or route
		knownval(plan.flight = flight) and
		knownval(stripLoc = strip.location) and
		knownval(strip.flightPlan = plan) and
		knownval(strip routes nextRoute) and
		knownval(nextRoute != route) and
		knownval(nextRoute.flightLevel < route.flightLevel) and
		knownval(fl = route.flightLevel) and
		knownval(newFL = fl - current.heightSeparation))
	do {
		printlnWithSimTime_s("Request %1 Descend to next route flight level", flightNumber);
		conclude((route.flightLevel = newFL), fc:0);
		conclude((current.flight = flight), fc:0);
		conclude((current.commReceiver = pilot), fc:0);
		conclude((current.commReason = "descend"), fc:0);
		conclude((pilot.commReason = "descend"), fc:0); // to remember advise to pilot
		conclude((current.commPerformative = "REQUEST"), fc:0);
		radioFlight(radio, flight, true, 1, 2, true); // give new route level
		updateFlightProgressStrip(flight);
	}
}//wf Request_Descend_Next_Flight_Level
```

[ATCO_Zurich_RP] - 21:34:28 : Radio Flight DXH611 REQUEST descend
[ATCO_Zurich_RP] - 21:34:28 : Flight Level 34500.0

Friedrichshafen confirms the handoff with AEF. TCAS alerts traffic in both BTC and DHL cockpits with 47 seconds to collision (vertical TAU is zero because the planes are essentially at the same altitude).

[ATCO_Friedrichshafen_Airport] - 21:34:32 : Radio Flight AEF1135 CONFIRM flight

[TCAS_BTC2937] - 21:34:33 : Traffic! Traffic!
[TCAS_BTC2937] - 21:34:33 : TCAS TA! Range Tau: 47, Vertical Tau: 0
[TCAS_DHX611] - 21:34:33 : Traffic! Traffic!
[TCAS_DHX611] - 21:34:33 : TCAS TA! Range Tau: 47, Vertical Tau: 0
[TCAS_BTC2937] - 21:34:33 : Separations Lateral: 10.63 nm Vertical : -2.0 ft
[TCAS_DHX611] - 21:34:33 : Separations Lateral: 10.63 nm Vertical : 2.0 ft
[TCAS_BTC2937] - 21:34:37 : TCAS TA! Range Tau: 41, Vertical Tau: 0

[TCAS_DHX611] - 21:34:37 : TCAS TA! Range Tau: 41, Vertical Tau: 0
[ATCC Z Display SE]- 21:34:37 : BTC - DHL lateral separation: 9.5
[TCAS_BTC2937] - 21:34:41 : TCAS TA! Range Tau: 38, Vertical Tau: 0
[TCAS_DHX611] - 21:34:41 : TCAS TA! Range Tau: 38, Vertical Tau: 0

In this particular model version the DHL pilots take 19 seconds to respond to ATCO, subsequently modified to 5 seconds. AEF makes approach for landing.

[Pilot_DHX611] - 21:34:47 : Auto-Pilot Off
[A320] - 21:34:47 : Flight AEF1135 Flying Friedrichshafen NDB Route
[A320] - 21:34:47 : Flight AEF1135 Heading West-South-West
[A320] - 21:34:47 : AirSpeed: 300, Vertical Speed: 0
[A320] - 21:34:47 : Altitude: 6871.0
[Pilot_DHX611] - 21:34:47 : Push Control Column to Descend

With TAU now at 35 seconds, TCAS now informs BTC to climb and DHL to descend.

[TCAS_BTC2937] - 21:34:49 : Climb! Climb!
[TCAS_BTC2937] - 21:34:49 : TCAS RA! Range Tau: 35, Vertical Tau: 0
[TCAS_BTC2937] - 21:34:49 : Separations Lateral: 8.38 nm Vertical : 1.0 ft
[TCAS_BTC2937] - 21:34:50 : TCAS RA! Range Tau: 35, Vertical Tau: 0
[TCAS_DHX611] - 21:34:50 : Descend! Descend!
[TCAS_DHX611] - 21:34:50 : TCAS RA! Range Tau: 35, Vertical Tau: 0
[TCAS_DHX611] - 21:34:50 : Separations Lateral: 8.38 nm Vertical : -1.0 ft
[TCAS_DHX611] - 21:34:51 : TCAS RA! Range Tau: 35, Vertical Tau: 0

DHL pilots recognize they are in TCAS descent (which they already started under ATCO direction) and inform ATCO. BTC reaches -31 ft/sec vertical change in velocity; AEF requests arrival at airport; ATCO is monitoring the two flights on collision course ("tracking" indicates he is determining whether to issue expedite, which is unnecessary here).

[Pilot_DHX611] - 21:34:53 : Execute TCAS Advisory to Descend
[Pilot_DHX611] - 21:34:54 : Flight DXH611 INFORM TCAS_descent
[TCAS_BTC2937] - 21:34:54 : TCAS RA! Range Tau: 27, Vertical Tau: 0
[ATCO_Friedrichshafen_Airport] - 21:34:54 : monitoring AEF1135
[TCAS_DHX611] - 21:34:55 : TCAS RA! Range Tau: 24, Vertical Tau: 0
[B757] - 21:34:56 : Vertical Speed: -31
[B757] - 21:34:56 : Altitude: 35999.0
[Pilot_AEF1135] - 21:34:57 : Flight AEF1135 REQUEST arrival
[ATCO_Zurich_RP] - 21:34:57 : monitoring BTC2937 in conflict
[TCAS_BTC2937] - 21:34:58 : TCAS RA! Range Tau: 21, Vertical Tau: 0
[TCAS_DHX611] - 21:34:59 : TCAS RA! Range Tau: 21, Vertical Tau: 0
[ATCO_Zurich_RP] - 21:35:01 : tracking DXH611 in conflict
[TCAS_BTC2937] - 21:35:02 : TCAS RA! Range Tau: 19, Vertical Tau: 3
[TCAS_DHX611] - 21:35:03 : TCAS RA! Range Tau: 19, Vertical Tau: 6
[ATCC Z Display SE]- 21:35:03 : BTC - DHL lateral separation: 5.01
[TCAS_BTC2937] - 21:35:06 : TCAS RA! Range Tau: 16, Vertical Tau: 9
[TCAS_DHX611] - 21:35:07 : TCAS RA! Range Tau: 13, Vertical Tau: 9
[ATCO_Friedrichshafen_Airport] - 21:35:07 : Radio Flight AEF1135 AGREE

arrival

[TCAS_BTC2937] - 21:35:10 : TCAS RA! Range Tau: 10, Vertical Tau: 12
[TCAS_DHX611] - 21:35:11 : TCAS RA! Range Tau: 10, Vertical Tau: 12
[TCAS_BTC2937] - 21:35:14 : TCAS RA! Range Tau: 7, Vertical Tau: 15
[TCAS_DHX611] - 21:35:15 : TCAS RA! Range Tau: 7, Vertical Tau: 18
[TCAS_BTC2937] - 21:35:18 : TCAS RA! Range Tau: 5, Vertical Tau: 21

[ATCO_Zurich_RP] - 21:35:18 : tracking BTC2937 in conflict

[TCAS_DHX611] - 21:35:19 : TCAS RA! Range Tau: 2, Vertical Tau: 21
[TCAS_BTC2937] - 21:35:22 : TCAS RA! Range Tau: 0, Vertical Tau: 24
[TCAS_DHX611] - 21:35:23 : TCAS RA! Range Tau: 0, Vertical Tau: 24

When range (lateral) TAU drops to zero and vertical TAU is greater than zero, the planes have flown past each other, as indicated by these descriptions.

[T154]- 21:35:25 : BTC 2937 West of Uberlingen!
[B757] - 21:35:25 : DHL 611 North of Uberlingen!
[TCAS_BTC2937] - 21:35:26 : TCAS RA! Range Tau: 0, Vertical Tau: 27
[TCAS_DHX611] - 21:35:27 : TCAS RA! Range Tau: 0, Vertical Tau: 30

[ATCC Z Display SE]- 21:35:27 : BTC - DHL lateral separation: 0.52

[TCAS_BTC2937] - 21:35:30 : TCAS RA! Range Tau: 0, Vertical Tau: 33
[TCAS_DHX611] - 21:35:31 : TCAS RA! Range Tau: 0, Vertical Tau: 33

[ATCO_Zurich_RP] - 21:35:35 : monitoring DXH611 in conflict

[TCAS_BTC2937] - 21:35:36 : TCAS TA! Range Tau: 0, Vertical Tau: 36

[ATCO_Zurich_RP] - 21:35:36 : monitoring BTC2937 in conflict

[TCAS_DHX611] - 21:35:37 : TCAS TA! Range Tau: 0, Vertical Tau: 36

ATCO handles another flight; DHL informs ATCO that the conflict had cleared.

[ATCO_Zurich_RP] - 21:35:37 : monitoring OTHER

[TCAS_BTC2937] - 21:35:40 : TCAS TA! Range Tau: 5, Vertical Tau: 42
[B757] - 21:35:40 : Vertical Speed: 0
[B757] - 21:35:40 : Altitude: 34697.0

[Pilot_DHX611] - 21:35:41 : Flight DXH611 INFORM conflict_cleared

[ATCO_Zurich_RP] - 21:35:41 : Radio Flight OTHER REQUEST chat

[ATCO_Zurich_RP] - 21:36:00 : Radio Flight DXH611 CONFIRM conflict_cleared

27 Appendix: Brahms Probabilistic Constructs Affecting Simulation Variability

Brahms models include probabilistic events such that running a given model multiple times may produce different outcomes. Variances are possible in durations and beliefs, which affects both decisions made by agents and circumstantial temporal-spatial interactions (e.g., if an agent isn't at a location at a particular time, he might not detect an alarm). Table 27-1 lists the Brahms model constructs that may have probabilistic values. Following sections enumerate the variables in Brahms-GÜM.

Table 27-1: Probabilistic variability in action durations and beliefs in Brahms models

Primitive Activity Duration	The **duration of the activity** can be defined to be a fixed amount of time or a random amount of time. For a fixed time, the random facet is set to false and the max-duration attribute is set (seconds). To define a random amount of time, the random facet is set to true, the min-duration and the max-duration facets are set (seconds).
Detectable When	Detectable may be checked after every fact/belief change or at a specified **percent completion time of the workframe** or activity, varying from 0% (start) to 100% (end) completion.
Detectable Probability (WF action, Composite Activity, & End-Condition to Complete or Abort Composite Activity)	The fact Detect-Certainty (DC) is an unsigned number [0, 100], representing the **probability that the fact(s) will be detected** with a normal distribution. The default, if the detect-certainty is not specified, is 100.
Consequence Probability	Belief-Certainty (BC) and Fact-Certainty (FC) are an unsigned integer [0, 100] representing **probability (normal distribution) that the proposition will be concluded**.
Workframe priority <= Composite/Primitive Activity Priority	Activities can be assigned a priority. The priorities of activities in a workframe are used to define the priority of a workframe. The workframe will get the priority of the activity with the highest priority defined in the workframe. (Used conventionally to order a composite activity into an ordered process of WFs.)
Thoughtframe priority	When multiple TFs are available to be fired at the same time, the one with the highest priority will fire first.

The following tables show variability in durations between minimums and maximums of activities for Air Traffic Controllers and Pilots.

Table 27-2 Air Traffic Controller and Pilot Activities

Activity Type	Activity Name	Min Duration (sec)	Max Duration (sec)	Description
primitive	Talk	1	3	Talking between ATCOs or pilots
communicate	sendCommunication	1	3	ATCO to ATCO or pilot to pilot communications
communicate	sendCommunication ViaDevice	3	6	ATCO communicates using a device like a phone. ATCO to pilot using radio.
primitive	listenToRadio	1,6	2,12	Listening to radio like waiting for pilot response, etc.
move	moveOverToRadio	2	4	Move to where radio is located.
communicate	sendMessageVia Radio	2	4	ATCO or pilot communicates using radio.
communicate	sendFlightCommunic ation	3	5	Communicate flight information.

Table 27-3 Air Traffic Controller Activities

Activity Type	Activity Name	Min Duration (sec)	Max Duration (sec)	Description
communicate	readScheduleClock	1	3	Read time from clock
move	mv_ToLocation	1,3	3,8	Move to location like workstation area, etc.
primitive	Situate	2	5	Get familiar at a new work location
primitive	pickUpPhone	1	3, 5	ATCO picks up phone to make a call.
primitive	putDownPhone	2	4	When hanging up phone at end of conversation, other phone is busy, etc.
communicate	listenToPhone	10	30, 300	Listening to phone. Get phone status = Free, Busy, etc. Interrupted (& ends) when phone is answered.
primitive	listenToRadio	1,6	2,12	Listening to radio like waiting for pilot response, etc.
move	moveOverToRadio	2	4	Move to where radio is located.
communicate	sendMessageVia Radio	2	4	ATCO communicates using radio. Within composite activity talkOnRadio()
communicate	sendFlightPlan	1	5	Send (File) flight plan to Automated Flight Service Station (AFSS) Computer

communicate	readPlanDetatils	1	5	Read flight plan details like flight number, routes, etc.
communicate	readFlightDetails	1	5	Read flight details like from airport, destination, etc.
communicate	readRouteDetails	1	3	Read route waypoints, flight levels, etc.
communicate	sendFlight Communication	3	5	ATCO communicates flight info. Workframes Request Approach Handoff, Agree Takeover On Phone, Refuse Takeover On Phone
communicate	checkRadio	1	3	Check Radio is set to frequency.
get	getFlightProgress Strip	1	3	Take flight progress strip before moving it
put	putFlightProgress Strip	1	3	Put flight progress strip in area after moving it
move	moveWorkstation Area	1, 3	3, 6	Move to workstation area where radar display, radio, etc. are located.
move	moveAirTraffic ControlArea	2	5	Move out of workstation area which are sub-areas of air traffic control area
communicate	monitorPlane	2, 3	4, 5	Read plane information from radar screens. Workframe monitorFlight
communicate	monitorSector	1, 3	3, 5	Read what planes are in sectors displayed. Workframes Search Flights In Display, Monitor For Flight Conflict, etc.
communicate	getPlaneInfo	5	10	Read plane location coordinates, altitude, etc. information from radar display. In composite activity getPlaneFlightInfo.
communicate	getFlightInfo	2	4	Read flight number, closest flight, etc. information from radar display. In composite activity getPlaneFlightInfo
primitive	Read	1	10	Reading
communicate	configureATCDisplay	1	3	Change ATC Display properties like turn on audio sound, etc.
communicate	readFlightDetails	1	5	Read flight information, like flight number, etc. from flight strip. In composite activity readFlightProgressStrip
communicate	readRouteDetails	1	5	Read route information, like waypoints, flight levels, etc. from flight strip. In composite activity readFlightProgressStrip.
communicate	readFlightSegment Info	1	3	Read flight segment information. Similar to readRouteDetails but different information about route being read.

communicate	checkFlightInAir Sector	2	5	Check flight in air sector displayed on radar display
communicate	giveATCDisplay Briefing	3	10	Briefing ATCO taking over during handover to ATCO
communicate	sendAirSectorInfo	3	5	Radio next air sector radio frequency to pilot during air sector handoff
communicate	sendFlightInfo	1, 2, 5	2,4,6	Radio flight number. In composite activity radioFlight
communicate	sendFlightRouteInfo	1, 2, 5	2,4,6	Radio flight level for route. In composite activity radioFlight
communicate	updateFlightRoute	1	3	Update route flight level in flight strip. In composite_activity updateFlightProgressStrip
communicate	monitorFlightInfo	2	5	Quick monitoring of flight in radar display - shorter duration than monitorPlane
communicate	monitorPlaneIn Conflict	1	3	Read conflicting planes' location coordinates, altitudes, etc. information from radar display. Workframe Monitor Plane In Conflict
broadcast	requestSector AlternatePhone	2	5	ATCO asks out loud for alternate contact phone number. In composite activity requestAlternateAirSectorPhone
communicate	reportAlternate Phone	2	5	Report if call using alternate phone number succeeded or failed.
primitive	lookForWorkArea	1	3	Look for where other ATCOs are located.
communicate	informFlightStrip	1	5	Inform ATCO flight strip is available and at workstation area
communicate	discussAlternate SectorContact	10	20	Discuss about alternate contacts in other Air Traffic Control Centers
communicate	readAirSectorContact Info	2	5	Assistant reads alternate phone number contact from document
communicate	proposeAlternate PhoneNumber	7	10	Assistant proposes ATCO use alternate contact phone number
communicate	giveFlightDetails	5	15	ATCO communicates approaching flight's number, etc. information. In composite activity coordinateFlightApproach
communicate	givePlaneDetails	5	15	ATCO communicates approaching flight's plane location, etc. information. In composite activity coordinateFlightApproach
communicate	readGPSInfo	1	3	Read GPS information of waypoint
communicate	readWaypointInfo	1	3	Read Waypoint information of SID or STAR

communicate	readSID	5	60	Read Standard Instrument Departure (procedure for take-off)
communicate	readSTAR	5	60	Read Standard Terminal Arrival Procedure
primitive	checkRunwayStatus	1	3	Check whether runway is available for take-off or landing

Table 27-4 Pilot Activities

Activity Type	Activity Name	Min Duration (sec)	Max Duration (sec)	Description
primitive	talk	1	3	Talking.
communicate	sendCommunication	1	3	Pilot to Pilot communications.
communicate	sendCommunication ViaDevice	3	6	Pilot communicates using a device like radio.
primitive	listenToRadio	1,6	2,12	Listening to radio like waiting for ATCO response, etc.
move	moveOverToRadio	2	4	Move to where radio is located.
communicate	sendMessageVia Radio	2	4	Pilot communicates using radio. Within composite activity talkOnRadio().
primitive	dialingSelector	1	5	Dialing selectors on Mode Control Panel (MCP), changing radio frequency, etc.
primitive	pushingSwitch	1	2	Pushing a switch on Mode Control Panel (MCP)
primitive	pushingControl Column	3	5	Pushing Control Column (a.k.a. Yoke) to get plane to descend.
primitive	pullingControl Column	3	5	Pulling Control Column (a.k.a. Yoke) to get plane to climb.
communicate	configureFlaps	1	2	Pilot pushing on cockpit control column to extend or retract plane's flaps to climb or descend.
communicate	informAboutFlight	1	2	Pilot communicates flight information, like flight number, etc. using Radio
communicate	readPrimaryFlight Display	1, 3, 5, 30	3, 6, 10, 60	Read air speed, vertical speed & altitude displayed in Primary Flight Display (PFD)
communicate	pushVerticalNAV Switch	1	2	Pushing Vertical Navigation switch on Mode Control Panel. Send altitude value
communicate	pushVerticalSpeed Switch	1	4	Pushing Vertical Speed switch on Mode Control Panel. Send vertical speed value
communicate	pushSpeedSwitch	1	2	Pushing Air Speed switch on Model Control Panel. Send air speed value

communicate	readMCPSettings	1	2, 3	Read Mode Control Panel settings like bearing, auto-pilot on/off, etc.
communicate	readNavigation Display	2, 3	5, 8	Read time & distance to waypoint in Navigation Display (ND). Durations depends on gpsUpdateRate of AircraftGPSReceiver.
communicate	readTCASInfo	3	4	Read TCAS info displayed in Navigation Display
communicate	readRadioFrequency	1	2	Read frequency setting of radio
communicate	switchAutoPilot	1	2	Switch auto-pilot on or off on Mode Control Panel.
communicate	tuneRadio	1	2	Tune radio to frequency
communicate	getAltitudeInfo	4	9	Read altitude info from Primary Flight Display (PFD). Durations depends on gpsUpdateRate of AircraftGPSReceiver.
communicate	readMCPVertical Speed	1	3	Read Mode Control Panel setting for vertical speed.
communicate	readMCPAltitude	1	3	Read Mode Control Panel setting for flight level.
communicate	directToWaypoint	1	5	Set Control Display Unit to fly to waypoint
communicate	monitorWaypoint	3	5	Read time & distance to waypoint in Navigation Display (ND). Durations depends on gpsUpdateRate of AircraftGPSReceiver.
communicate	skipWaypoint	3	5	Update (program) flight plan in Control Display Unit (computer) to skip next waypoint.
communicate	requestDeparture Clearance	1	2	Radio ATCO for permission to depart airport. Performed with composite activity talkOnRadio()
communicate	requestArrival Clearance	1	2	Radio ATCO for permission to land on airport runway. Performed with composite activity talkOnRadio()
communicate	requestFlightLevel Clearance	1	2	Radio ATCO for permission to change flight level/altitude. Perform with composite activity talkOnRadio()
communicate	acknowledgeHandoff	1	2	Radio ATCO to agree to handoff. Perform with composite activity talkOnRadio()
communicate	pushHeadingSelector	1	2	Pushing a selector on Mode Control Panel (MCP)
communicate	monitorHeading	3	5	Read plane heading from Primary Flight Display (PFD).
communicate	monitorVertical Speed	1	5	Read plane vertical speed from Primary Flight Display (PFD).

communicate	monitorAirspeed	1	5	Read plane airspeed from Primary Flight Display (PFD).
communicate	changeGear	3	5	Pilot setting gears down for landing or up for takeoff.
communicate	speedBrake	3	5	Pilot applying speed brakes after landing plane to decrease plane's speed on runway.
communicate	getRouteInfo	3	5	Pilot reading flight segment/route info.
communicate	readWaypointGPS	1	3	Get GPS of Waypoint
communicate	readGPSInfo	1	3	Get GPS coordinate latitude, longitude, altitude, etc.
primitive	briefing	5	10	Briefing other pilots in cockpit. Workframe Perform Takeoff Briefing, Perform Approach Briefing
communicate	readWaypointInfo	1	3	Read waypoint info from STAR.
communicate	readFlightSegment Info	1	3	Read flight segment/route info from Control Display Unit (CDU)
communicate	reviewChecklist	300	420	Review checklist for departure or landing.
communicate	turnOffAlert	1	2	Pilot turning off alerts via Control Display Unit (CDU).
communicate	readFlightPlanDetails	5	8	Get flight plan info like SID, STAR, flight segments, etc.
communicate	readFlightDetails	15	20	Get flight details like arrival & departure time, airports, etc.
communicate	readFlightPlanRoute	5	8	Get flight segment/route info like to & from waypoints, etc.
communicate	readSTAR	15	20	Read Standard Terminal Arrival procedure (STAR)
communicate	readSID	15	20	Read Standard Instrument Departure procedure (SID)
communicate	updateRouteInCDU	3	5	Update (program) flight segment/route info in Control Display Unit (Computer).
communicate	monitorTCASInfo	1, 3	3, 8	Read TCAS info about nearest flight, etc. from Navigation Display. Durations depends on gpsUpdateRate of AircraftGPSReceiver.
communicate	readVerticalRate Advice	1	2	Read TCAS advice to climb or descend from Navigation Display
communicate	informATCAdvisory	1	2	Inform ATCO on radio (talkOnRadio) about TCAS advisory to climb or descend

27.1 Modifications to Brahms Engine for Compatibility with Automated Verification Methods

We made several modifications to the Brahms engine to enable managing the variability inherent in Brahms simulations, both for testing and refining the model as well as for applying model checking methods.

The nature of Brahms variables is that a single model configuration ("scenario") can produce different outcomes each time it is simulated. Further, the interactions of modeled people and systems in Brahms-GÜM are sensitive to cumulative timing effects (e.g., each flight handoff requires a certain number of seconds, which is variable, and the number of flights to be handled is also variable). This variability will generate different behavior sequences ("chains of events") over multiple runs.

It therefore became useful to allow the model builder to define a "seed" for the random number generator, such that every run with a given model would be identical. This of course facilitated understanding and debugging the simulation. This seed value is used by the Brahms virtual machine to generate the randomized duration for activities and to determine the certainty value for belief, fact, and detectables, as indicated in Table 27-1. The absence of this mechanism from Brahms heretofore demonstrates how the complexity and precision of Brahms-GÜM exceeds any simulation we had previously constructed. In particular, we had usually used a discrete event cycle (clock tick) of 3 or 5 seconds, whereas we use 1 second in Brahms-GÜM.

One complication for model checking is that Brahms model builders have heretofore relied on textual order of WFs in definition files to determine the ordering in which WFs for a given activity are applied, a default of the Brahms engine. However, to make such semantics explicit, priorities are now assigned when order is important. Also, the engine now randomizes the selection of available workframes, thoughtframes, and interrupted workframes having the same priority.

28 Appendix: Limitations of Brahms Framework

The Brahms-GÜM simulation exercised the Brahms framework to a greater extent than any previous model, combining every method we had previously developed, such as how to simulate reading a display, radio/phone conversations, moving in an office, and broadcast ("out loud") communications. Simulating agent activities in a moving vehicle such as an automobile had been anticipated in the design of the language, but had never been attempted, let alone simulating flying an aircraft by manual and autopilot controls on a planned route. We had modeled displays but nothing as complicated as simulating radar of objects in motion. Not surprisingly a few limitations in the language were discovered, which are described in this appendix.

28.1 Perceiving Broadcast While Moving

The built-in "Move" activity is treated like any primitive activity in a Brahms model, which means that the agent cannot do anything else while the activity is being performed. In particular a *broadcast* while the agent is moving will not be detected by the moving agent because an agent in motion was not represented as being in any location at all. For example when moving say from the left to the right workstation, the agent would previously not be in any location at all, rather than being simulated as being in the ATCC. This may seem like an obvious shortcoming in the design, but it has not previously caused a problem. In the Überlingen scenario the limitation prevented the ATCO from detecting radio calls while moving from the workstation area to speak with the CA.

To remedy this undesired effect, we modified the Brahms engine so when an agent is moving from one location to another, the agent is resident in the parent location while moving (i.e., the more general location shared by both the start and end locations, if one exists). A side benefit is that a moving agent can now also detect other agents who reside in the transit (more general) area, and other agents will detect arrival (and departure) of the moving agent.

28.2 Simulating Acting During a Communication

Agents can only do one activity at a time, that is, their actual primitive actions are serial, not parallel. So although they can now detect broadcast communications while moving, they cannot read something on a wall while walking by or carry out a conversation, or even write something down while they are listening to a person or the radio.

An obvious example of parallel actions appears in the BFU Report timeline chart (Appendix 3), which shows that the BTC crew understood and were acting on the ATCO's command before he finished speaking. He spoke for 8 seconds. A BTC crew member says "Descend!" at 34:54, which is 5 sec into ATCO's utterance. The last part of the phrase was "expedite, I have crossing traffic," which requires a bit less than 2 seconds to say quickly. Therefore, to model the relation between the start of the ATCO's intervention and AP disengage it was necessary to shorten the ATCO's

modeled speech by a few seconds, with the AP disengage occurring in the second immediately following.

28.3 Simulating an Object "Hearing" a Broadcast Communication

An oversight in the design of the Brahms language is that a modeled object (e.g., a radio) can broadcast communications that will be "heard" by agents who are in the geographic area, but it cannot receive broadcast communications. This is an obvious shortcoming for modeling a microphone for example. Consequently, in the simulation of radio communications in the cockpit (Figure 25-5), it is necessary for the simulated pilot to communicate directly with radio, and others in the cockpit will not hear what he is saying. In Brahms-GÜM the radio broadcasts back the messages it receives.

Giving objects the ability to receive broadcast messages requires some consideration. Many more objects are commonly present in a geographic area than agents, so the engine would have to be smarter about how broadcast communications are processed. For example, objects might be given an attribute to indicate that they are capable of receiving audible messages.

28.4 Simulating "Monitoring" a Display

In Brahms-GÜM we simulated ATCO "reading" the radar display as we have always modeled a person reading documents. First, the representations (text or graphics) printed on a document or displayed on a screen are modeled as "beliefs"—they are propositions about the world, and hence (unlike Brahms "facts") are not necessarily true. Thus to simulate the activity of monitoring the display we model ATCO's "reading" activity as a Brahms communication act that "asks" the display to transfer its beliefs about, for example, the aircraft designations, altitudes, etc. This information is modeled as beliefs of the radar display object, that is, as propositions, as opposed to being represented as geometric lines or character marks. This approach is well-known in cognitive science, though of course it finesses the work involved in looking and interpreting the marks on a screen (Clancey 2005)

This approach has been sufficient in modeling documents such as job orders in a business office, checklists on Apollo, or written procedures onboard the ISS. But a radar display is dynamic and almost always perceived selectively (e.g., scanning different screen areas clockwise in sequence). Unlike a document or typical computer screen, the radar screen is also updating while you are looking at it. Also, the information is obviously not all text, graphics must be examined and conceptually related (e.g., conceiving that a flight is "close" to a sector boundary).

However, having the radar display simply communicated all of the information relevant to a particular flight (including separation issues), make it easier for the simulated ATCO to know important facts. It might be preferable in future research using Brahms to simulate more of the effort required to look and relate parts of the display. One way to do this is to represent the *graphic relations* as propositions (e.g., the physical location of objects on the screen), that is, representing the information

as facts about the display. It would then be possible to model the agent's looking activity in a fine-grained way using detectables, which operate on facts in the world (i.e., the distinction between someone presenting their view of the world vs. your observing the world directly yourself). The result would still be that the agent has an uncertain, incomplete model of (beliefs about) the world, and the partial character would reflect better how the ATCO was looking at the display. This is relevant in particular to Überlingen: We believe the right workstation's display—where the Zurich ATCO was sitting while speaking to AEF 1135 and apparently following the AEF on the screen—showed the BTC and DHL on an intersecting course less than 20 nm apart, but he did not perceive that information in the display.

Ideally, the radar system to model calculations could be modeled as belief and what is *displayed* on screen would be facts (at the level of simple text and partially abstracted graphics). These facts would be created in the simulation as the screen is refreshed. The model would then have "world facts" about the actual location of the aircraft, velocity, etc.; the radar system's "beliefs" about these facts (partial and not current because of the delay of the radar sweep); facts about what is presented on the display at any time; and ATCO's beliefs about the location of the aircraft, etc. by virtue of detecting and reasoning about what is presented on the display. This example illustrates the redundancy and variation of models of the world that occur in actuality.

However, a complication in Brahms prevents implementing this fact–>belief–>fact–>belief flow directly—detectables are not specific to areas. So if for example, the simulated ATCO sitting at the left workstation detecting information about flights would be able to detect information about flights displayed on the right workstation—or anywhere in the ATCC. This is an issue in the Brahms language/engine not encountered previously because: 1) detectables were not intended for simulating perception of documents or displays, and 2) detectables were intended for simulating perception of facts about things in the world (e.g.., a tool on a table, a disconnected wire, the setting of a switch), 3) detectables are associated with workframes, part of activities, which are always located (occurring in some modeled geographic area).

The analysis clarified that the design of the language/engine was implicitly based on the semantics that detectables are used to model perception of visual facts and communications to model auditory perception (including broadcasts such as an alarm).

The analysis also revealed that the semantics of detectables (as implemented in the Brahms engine) was less constrained than we realized—facts are detectable from anywhere in the world. When detection involves facts about objects that in the same location as the agent, there is no problem—the model is designed so the agent must move to the object's location before the activity with detectables is performed (e.g., Brahms-GÜM models the ATCO moving from one workstation to another to read the display and control strips, speak on a certain frequency, or use the telephone). The

complication in Brahms-GÜM is that the flight is not actually in the ATCC, so detecting facts about it is not possible; ATCO can only detect facts about what's on the screen, which are representations about the flight.

Of course, we do not what to represent squiggles on the display but propositions about flights associated with different areas on the display. Given that Brahms was never designed for simulating perceptual work, relegating this to a coupled simulation would appear to be the best approach.

28.5 Object Actions Require at Least One Clock Tick

In other Brahms simulations, we have associated on clock tick of the engine with three seconds of real time. For the Überlingen scenario we used one second per tick because of the actions that needed to occur the second following another event (e.g., disengaging AP in response to TCAS). However, this means every action-reaction or two sequential actions in general requires two seconds. Thus, for example a radio call requires (at least) a second to speak into the microphone (i.e., communicating with the radio), a second for the speaker's radio to communicate to the listener's radio, and another second for the listener's radio to communicate to the listener—three seconds for what is practically a speed-of-light transaction. Consequently, in the Überlingen scenario if the ATCO takes five seconds to utter the emergency course of action, the BTC pilots cannot respond until at least eight seconds after he begins speaking, which is a second or two too long. Another example of the "one tick per action" restriction occurs within the TCAS model itself, in which in a TCAS object may be busy updating its internal model of aircraft relationships when it should be issuing an advisory. This is manifested in simulation runs in which there is a five second delay between the DHL and BTC TCAS traffic advisories.

We can work around the TCAS issue by reconfiguring the model, but determined it not to be necessary for present purposes because the simulated Brahms-GÜM pilots do not act on a TA. The issue with the radios, however, is a problem that can only be solve by modifying how objects are simulated by the Brahms engine, allowing for actions to take less than the period of the clock tick, and thus processing multiple cycles of object actions within a clock tick.

Notice that eliminating the radios would seem obvious, but violates the "total system model" principle. For example, we could not model the implications of a pilot entering the wrong frequency in the radio (one of the causes of the crash of the plane flown by JF Kennedy Jr.).

28.6 Modeling an Activity's Constraints, Goals, and Distributed Responsibilities

It would be advantageous to document and provide metadata about activities, such as the goals and responsibilities being satisfied (Clancey 2002), constraints affecting priorities, and so on. To do this, we would represent an activity as an *conceptual object* within Brahms. Just as a "flight" is a conceptual entity, an ATCO or pilot would have a mental model representing the activities of a flight. Alternatively, one could

extend the Brahms language to represent this conceptual information within the activity definition proper, that is, allow an activity to have additional properties besides WFs, TFs, etc. One could also represent constraints that constitute specifications, such as properties that should be satisfied after the activity completes, which would be useful for model verification.

28.7 Providing an "Explanation System" for the Simulation

Aside from the changes to the language and engine, it would be helpful for modelers and subject matter experts studying simulated scenarios to implement an explanation system like that developed for consultation systems in the 1970s (Scott et al. 1977). Rather than studying logs and the AgentViewer, one could ask natural language questions about the simulation run. The following are illustrative general questions that would make it easier to understand, critique, and improve the simulation and hence the corresponding work system design:

- What does agent X believe about {agent, group, object} Y?
- Which agents currently do/don't believe P?
- Why does/doesn't agent X believe P?
- Did agent X do activity Y? When? Why?
- Why didn't agent X do activity Y?
- What activities were pending/interrupted for agent X during time period T?
- If agent X believed P at time T, how would that have influenced his subsequent behavior?

A program can answer such questions by analyzing a simulation trace and interpreting the logic of the model with respect to the operation of the engine (Clancey et al. 1986).