Distributed Intelligence
for Ground/Space Systems

N95- 23685

Mads Aarup &
Klaus Heje Munch
CRI Space
Denmark
ph: +45 4582 2100
fax: +45 4581 3217
maa@nov.cri.dk
khm@spd.cri.dk

Joachim Fuchs
ESA/ESTEC/WGS
The Netherlands
ph +31 1719 8 5296
fax: +31 1719 8 5419
joachim@wgs.estec.esa.nl

Ralf Hartmann
Dornier
Germany
ph: +49 7545 8 9374
fax: +49 7545 8 4411
hartma@spacediv.dofn.de

Tim Baud
Cray Systems
United Kingdom
ph: +44 272 277854
fax: +44 272 290917
baud@craysys.co.uk

INTRODUCTION

DI is short for Distributed Intelligence for Ground/Space Systems and the DI Study is one in a series of ESA projects concerned with the development of new concepts and architectures for future autonomous spacecraft systems. The kick-off of DI was in January 1994 and the planned duration is three years. The total budget is 600,000 ESA Accounting Units corresponding to approximately $720,000.

Problem Definition

The background of DI is the desire to design future ground/space systems with a higher degree of autonomy than seen in today's missions. The aim of introducing autonomy in spacecraft systems is to:

- lift the role of the spacecraft operators from routine work and basic trouble-shooting to supervision,
- ease access to and increase availability of spacecraft resources,
- carry out basic mission planning for users,
- enable missions which have not yet been feasible due to eg. propagation delays, insufficient ground station coverage etc,
- possibly reduce mission cost.

Project Description

The study serves to identify the feasibility of using state-of-the-art technologies in the area of planning, scheduling, fault detection using model-based diagnosis and knowledge processing to obtain a higher level of autonomy in ground/space systems.

A demonstration of these technologies will be developed in the form of a prototype to run in a laboratory environment for the purpose of evaluating future ground/space system designs, and to experiment with the distribution of functionalities of the autonomous architecture between the ground and space segment. DI will use the ERS-1 earth observation mission as the reference mission for the study.

Consortium

The DI Study is carried out for the System Simulation Section of ESA's Technology Center ESTEC by a consortium, led by CRI, and backed by Cray Systems and Dornier.

CRI has a background in the development of ground control systems, planning/scheduling and simulation, combined with spacecraft operations support in the area of flight dynamics. CRI has applied knowledge-based techniques for ESA/ESTEC and ESA/ESOC to mission planning, flight operations, and failure detection, diagnosis and repair. CRI is head of an industrial Consortium developing the Ørsted Scientific Micro Satellite, with direct responsibility for AIV and mission planning, space and ground segment and operations. Ørsted will be launched by a Delta Launcher early 1996.

Cray Systems has developed simulators for most ESA missions, including ERS-1. Also, Cray has substantial experience in the development of control centers and mission planning. Cray has been a main player in the development of the ERS-1 Control Center, and has designed and implemented the operational ERS-1 mission planning system for ESA's Operations Center ESOC.

Dornier was prime contractor for the ERS-1 industrial consortium, and has played a lead role in numerous other spacecrafts, providing solid spacecraft and ground system engineering experience. Dornier offers extensive experience in the development of flight operations plans, in addition to knowledge-based planning.

REFERENCE MISSION

A suitable reference mission for verification of a distributed knowledge-based ground/space architecture providing autonomy should involve a complex spacecraft in an orbit that is either partly without ground contact or so distant that significant delays are inevitable. A natural choice is to select ERS-1 as the reference mission since:

- ERS-1 is equipped with several scientific instruments with many operational constraints, implying very complex mission planning,
- ERS-1 is in a low polar orbit causing it to be out of ground contact during prolonged periods of time,
- operational experience has been gained, making it possible to qualify advantages of autonomy and AI.

Furthermore, the ERS-1 systems engineering expertise and the ERS-1 simulator is available in the DI consortium.

APPROACH

The DI study is divided into two phases. In phase I, we have taken the rather provocative liberty to simply consider the ground and space segment as one combined system. This allows focusing on the essential user requirements on the overall system and on the interaction of the various modules of the system. In the phase I mock-up, the following software will be reused:

- The goal-oriented planning module of Dornier's TINA planner,
- The Optimum-AIV scheduling kernel that CRI previously extended with ERS-1-like subsystem models for the GMPT prototype,
- Cray Systems' operational ERS-1 simulator (for simulating all aspects of the spacecraft behavior),

Furthermore, several ideas from the faults diagnosis and constraints generation module of CRI's EOA (Expert Operator's Associate) may be re-used for the fault diagnosis and repair part of the mock-up.

In phase II, the focus will be concentrated on the distribution aspects of the ground and space segments taking into account issues of distributed artificial intelligence. The development of the distributed phase II prototype will further improve the integrated software tools of the phase I mock-up enabling the evaluation and demonstration of benefits.

ARCHITECTURE

The phase I architecture is based on a hierarchical, object oriented approach providing basis for re-use of existing software modules and ease of final distribution of functionality between the ground and the space segment in phase II.

An overview of the architecture is shown in Figure 1.

Selected data/knowledge structures and modules shown in the architecture are briefly described in the following.

Data/Knowledge Structures

<u>User Requests</u> describe either experiments or spacecraft maintenance operations, and

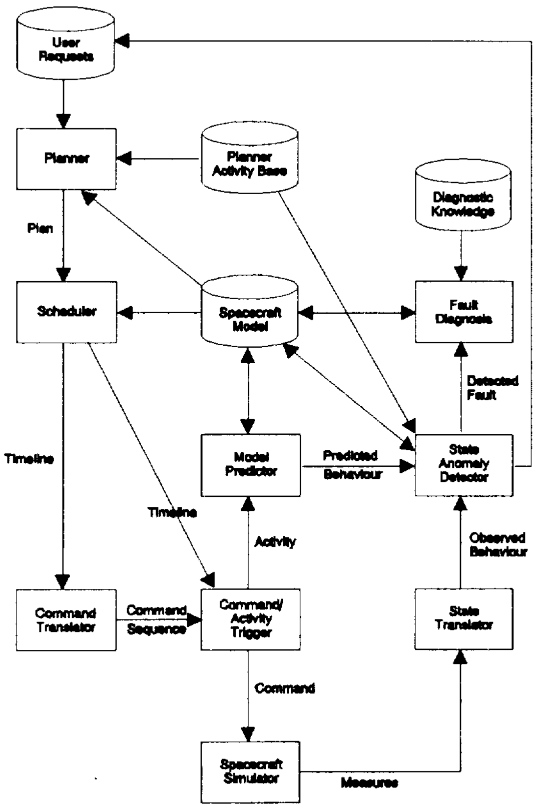

Figure 1: Functional Architecture

are defined by a number of attributes e.g. instrument to use, execution time, orbit position, priority, etc. The formulation of a user request does not require knowledge of the low-level activities necessary to accomplish the request.

<u>Planner Activity Base</u> contains definitions of low level activities to be used for achieving user requests. An activity is defined by:

- preconditions necessary to start the activity,
- resources necessary to carry out the activity (used during scheduling), and
- changes which the activity applies compared to its initial state, e.g. concerning resource availabilities or auxiliary constraints.

<u>Spacecraft Model</u> contains various types of information about the spacecraft used for:

- the prediction of spacecraft behavior,
- the comparison between predicted and observed behavior of the spacecraft (and thereby the fault detection), and
- the diagnosis of a detected fault, e.g. an unexpected component state change or a change of available resources.

The model includes static knowledge about the structure and behavior of the spacecraft and its subsystems, and dynamic knowledge about the current state of the spacecraft. The static knowledge facilitates the reasoning about behavior of the spacecraft as a response to activities, and the generation of diagnosis hypotheses on defective components based on discrepancies in predicted and observed behavior. The dynamic knowledge which is maintained by the model predictor includes such information as resource availabilities (electrical power, data storage capacity, etc.), and descriptions of all anomalies identified by the fault diagnosis module. The model is an abstraction of the spacecraft and the corresponding spacecraft model used in the ERS-1 simulator. It will consist of a subset of the real spacecraft such that it is self-contained with little or no reliance on un-modelled functions. Furthermore, the reasoning about the behavior for the spacecraft will be on the level of activities/predicted behavior rather than the lower command/measures level of the spacecraft simulator.

<u>Diagnostic Knowledge</u> contains an abstraction of relevant experience from satellite designers, manufacturers and operators used for diagnosing faults. This knowledge, expressed as a number of heuristics, can be used either for postulating a priori diagnosis hypotheses or for focussing a systematic model-based diagnosis.

Modules

<u>Planner</u> defines a plan for achieving a number of user requests, i.e. selects and arranges a number of low-level activities

defined in the planner activity base such that the execution of the activities will achieve the requests. The planner must take into account the actual state of the spacecraft model. Replanning is invoked if either the user requests are changed or the spacecraft model is updated as a result of fault diagnosis. The planning process is goal-driven based on backward chaining with backtracking.

Scheduler produces a timeline of the activities generated by the planner. The timeline defines the starting time and duration of all activities. The scheduler is initiated each time a new plan has been generated or some resource availability has changed due to a failure. It interfaces the spacecraft model for retrieving constraints used in the scheduling process, e.g.:

- resource constraints on requests made by the activities,
- temporal constraints on predefined fuzzy times due to orbit position or target visibility and to the duration of activities,
- system state constraints on configuration and platform maintenance.

Model Predictor generates expected behavior of the spacecraft based on the spacecraft model as a response to commands. The model predictor applies forward chaining for reasoning about the behavior. It updates the changing states and modes of the subsystems in the model.

State Anomaly Detector (or fault detector) identifies faults based on:

- the observed behavior being an abstraction of the measures derived from the spacecraft simulator,
- the predicted behavior derived from the spacecraft model by the model predictor,
- the definition of activities in the Planner Activity Base for verifying post-conditions associated to activities,
- constraints defined in the spacecraft model some of which depend on the actual state of the spacecraft subsystems.

The fault detection enables the autonomous system to detect such faults as:

- hardware or software errors where the predicted behavior of the spacecraft is inconsistent with the observed behavior,
- errors where the current state of the spacecraft is inconsistent with verification parameters or constraints defined in the model, e.g. due to a wrong time-tag in a manually up-linked command sequence.

Having detected a fault, the fault detection triggers the fault diagnosis module.

Fault Diagnosis generates hypotheses explaining a detected fault. The most important method to be applied for fault diagnosis is model-based diagnosis using the spacecraft model for generating hypotheses about abnormal subsystems or components explaining the fault.

The result of the fault diagnosis is an update of the spacecraft model in case the analysis derived an anomaly, e.g. that a spacecraft status or constraint have changed in an unforeseen manner or that a spacecraft resource has changed in an unexpected way. In the former situation, the fault diagnosis module reinvokes the planner as such problems require an update of the logical sequence of activities to be carried out for recovery. In the latter situation, the scheduler is reinvoked for recovery.

CONCLUSION

The current status as of June 1994 is that a Draft User Requirements Document for the phase I prototype has been produced and the ERS-1 mission demonstration scenarios have been described. The prototype mock-up development has just begun with a clarification of the general MMI strategy.