2001 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

# THE VIRTUAL TEST BED PROJECT[1,2,3]

Luis C. Rabelo, Ph.D.
NASA Fellow
lrabelo@mail.ucf.edu, lrabelo@alum.mit.edu
4000 Central Florida Boulevard
Industrial Engineering and Management Systems (ENG 2, Room 312)
University of Central Florida
Orlando, Florida 32816

## ABSTRACT

This is a report of my activities as a NASA Fellow during the summer of 2002 at the NASA Kennedy Space Center (KSC). The core of these activities is the assigned project: the Virtual Test Bed (VTB) from the Spaceport Engineering and Technology Directorate. The VTB Project has its foundations in the NASA Ames Research Center (ARC) Intelligent Launch & Range Operations program. The objective of the VTB project is to develop a new and unique collaborative computing environment where simulation models can be hosted and integrated in a seamless fashion. This collaborative computing environment will be used to build a "Virtual" Range as well as a "Virtual" Spaceport. This project will work as a technology pipeline to research, develop, test and validate R&D efforts against real time operations without interfering with the actual operations or consuming the operational personnel's time. This report will also focus on the systems issues required to conceptualize and provide form to a systems' architecture capable of handling the different demands.

[1] This report will constitute the basis for future papers to be presented in conferences and submitted to journals. These papers will include my NASA colleagues as co-authors.

[2] Henry McCoy, Robert Turner, Jeppie Compton, and Darrin Skelly have contributed to this report.

[3] Robert Turner is the NASA-KSC Project Manager of the Virtual Test Bed Project. Jorge Bardina from NASA ARC is the associated NASA Researcher.

# THE VIRTUAL TEST BED PROJECT

Luis C. Rabelo, Ph.D. - NASA Fellow

## 1. Introduction

The contributions to the Virtual Test Bed (VTB) Project during this fellowship are extensive. These contributions can be classified into the following areas: (1) systems architecting activities, (2) evaluations of models and processes, (3) evaluations of technologies, (4) systems design, (5) facilitation of technical meetings, (6) support to improve the communications of technical concepts and project management activities, (7) intellectual property consulting, (8) submission of proposals for funding, (9) documentation and technology transfer, and (10) liaison with Academic groups. Due to restrictions of space in this report, we will focus on the following four areas: systems architecting activities, evaluations of models and processes, evaluations of technologies, and systems design.

This report is organized in the following sections. Section 2 introduces general concepts. These concepts will be needed to understand the objectives of the project and its implications for the future of NASA. Systems Architecting contributions is the topic of Section 3. The systems architecting activities deal with the utilization of the House-of-Quality to translate the users' needs to the design requirements. Section 4 discusses the models and process evaluations activities accomplished during this fellowship. The CALPUFF model was reviewed from its computing characteristics viewpoint (operating system, hardware platform, computer programming language). In addition, Section 4 presents a process flow to determine the $E_c$ based on the Blast Overpressure Assessment Model (BLASTX). The technology evaluations and recommendations are presented in Section 5. Section 5 presents a summary of different tools and technologies that can be used to develop the Virtual Test Bed. The last section is Section 6. Section 6 describes the systems design contributions to the project.

## 2. General Concepts

Virtual test bed environments are the next S-curve[1] of the computer-aided design (CAD) systems. These "advanced" CAD environments will combine not only the 3D solid modeling capabilities of the current CAD systems but also will integrate in a seamless fashion models that represent the different stages of the life-cycle of a system. Therefore, the 3D solid models of the mechanical/physical design will be integrated to the materials models, dynamics models, environmental models, operational models (including operators, crew members), safety models, and other type of models. These virtual environments will go beyond the virtual world attempts of the 90's. The virtual worlds of the 90's concentrated on the look and feel dimensions. However, a virtual environment will go beyond the former cosmetic approach with higher realism and fidelity, and more emphasis on engineering. Virtual test beds will create multi-disciplinary and collaborative design spaces.

From another viewpoint, the current CAD environments concentrated not only in a single dimensional view of a system but are also not adequate for complex.systems design. One interesting characteristic of a complex system is that it is by default a system of systems. To be faithful to concurrent engineering principles, you have to study the interactions among the different systems that are elements of the complex systems. This system of systems is non-linear in nature and the interactions among the different components bring interesting emergent properties that are very difficult to visualize and/or study by using the traditional approach of decomposition.

A very interesting outcome of the current CAD systems is that the single view, which they display, have limited them to just being geometric design collaborative environments. For example, to develop a specific product, a CAD system will only serve as a collaborative environment to those

[1] S-curves are used to study the evolution path of technologies.

members of the design team that only need to know or use 3D Solid Modeling characteristics and/or geometric features. A virtual test bed environment is not only related to geometric design but also includes other properties such as materials, operational behavior, and interactions. A more comprehensive modeling environment such as a virtual test bed will need to have enhanced usability and connectivity capabilities to become an effective collaborative environment.

Spaceports are complex systems. According to Barth [1] "Spaceport technologies must employ a life-cycle "system of systems" concept in which major spaceport systems – launch vehicle processing systems, payload processing systems, landing and recovery systems, and range systems – are designed concurrently with flight vehicle systems and flight crew systems." Therefore, it seems logically to think that a virtual test bed can host the different models that represent different systems and elements of a spaceport. These models in the virtual test bed will work in an integrated fashion synthesizing in a holistic view and becoming together a Virtual Spaceport. This Virtual Spaceport can be utilized to test new technologies, new operational processes, the impact of new space vehicles in the spaceport supply chain, and the introduction of higher schemes of decision-making. A Virtual Spaceport will allow an intelligent visualization of the entire spaceport concept and the implementation of knowledge management strategies.

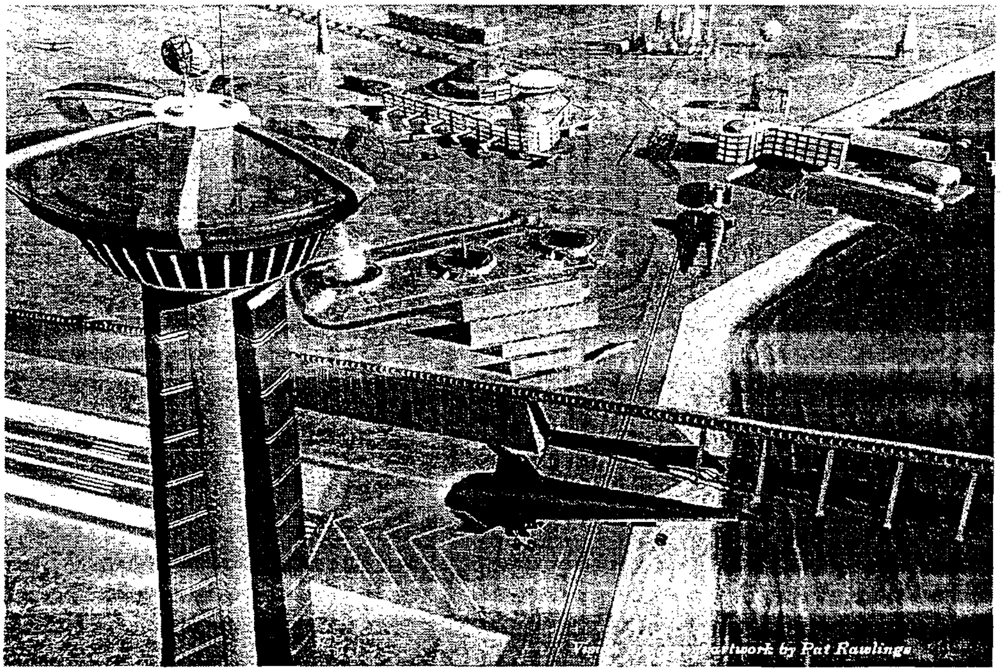

**Figure 1. Conceptualization of a future spaceport.**
**(Vision Spaceport artwork by Pat Rawlings/SAIC)**

Another important and interesting system of the spaceport is the range. The range has different definitions depending on the NASA Center. In this report the range will follow the NASA-Kennedy Space Center (KSC) definition. According to KSC, the range is a "physical volume of space that opens during launch. It's used for vehicle tracking, telemetry, and communications, among other things." [1] The range encompasses many different operations. One of them, range safety has a high level of complexity. The Eastern Range (ER), with its launch base at Cape Canaveral Air Station, is under the observance of the 45th Space Wing (Patrick Air Force Base is the headquarters). The responsibilities of the safety offices at the ER and NASA-KSC include three areas [19]: (1) System safety reviews, (2) Flight safety, and (3) Ground safety. These responsibilities include the study, modeling, and analysis of the hazards from potential launch accidents and the corresponding calculations of $E_c$ ("Casualty Expectation"). Toxic effects, debris, and blast overpressures and their effects in the population are the primary hazards to be studied. The fuel of the vehicles may cause toxic effects. Vehicle explosions caused by system failures can produce debris. In addition, these explosions may also create blast overpressure. Modeling of these effects is required for launch safety. However, hosting the range models and vehicle models in the VTB will support the development of risk management and reduce risk avoidance approaches. A Virtual Range can be used to design and evaluate flight termination systems (FTS), to evaluate and implement a risk assessment scheme of legacy and new vehicles, and to develop emergency responses. A Virtual Range can help reduce the cycle time to certify legacy vehicles (change from one range to another, or when modifications have been performed) and new vehicles. These reductions of cycle times can have profound impacts on time and money.

The Intelligent Launch and Range Operations (ILRO) Program at NASA Ames Research Center (ARC) was started to perform initial studies of a test bed with a demonstration [5]. An evolution of the ILRO test bed is the Virtual Test Bed Project. The objective of the Virtual Test Bed (VTB) Project is to provide a collaborative computing environment to support simulation scenarios, reuse, and integration of multidisciplinary models that represent elements of the range and operations at spaceports. VTB will provide several benefits such as a risk management evaluation of legacy and new vehicles framework, a technology pipeline, and a knowledge management enabler. VTB will leverage current technological developments from NASA ARC in intelligent databases to present data and results as usable knowledge with associated security constraints (Developmental Aeronautics Revolutionizing Wind-tunnels with Intelligent Systems of NASA - DARWIN) and methodologies for remote access to research facilities employing interactive and virtual reality interfaces (Virtual Laboratory - VLAB).

## 3. Systems Architecting Activities

It is very well known that systems architecting integrates systems theory and systems engineering with architecting theory and practice of architecting [21,22]. Conceptualization is the keyword for architecting. System conceptualization involves creativity and the recognition of potential users and perceived needs. System architectures are driven by the function, instead of the form, of the system. Systems Engineering is the one that provides the form. We utilized in this project QFD (Quality Function Deployment) and in particular a modified house of qualities to guide our architecting activities.

QFD can be used to improve the process of introducing new ideas that translates the user requirements from concept to development and beyond [20,23]. In the application of QFD, the initial phase involves the creation of a matrix called the House-of-Quality matrix due to its roof-like format as shown in Figure 2. The House-of-Quality used here follows modifications of the matrix of change [4] (MOC). This House-of-Quality has a list of the user needs/benefits (organized on the left side of the House). This list of needs/benefits is translated into the features (technical) needed (design requirements of a solution to satisfy the needs and/or provide the benefits).

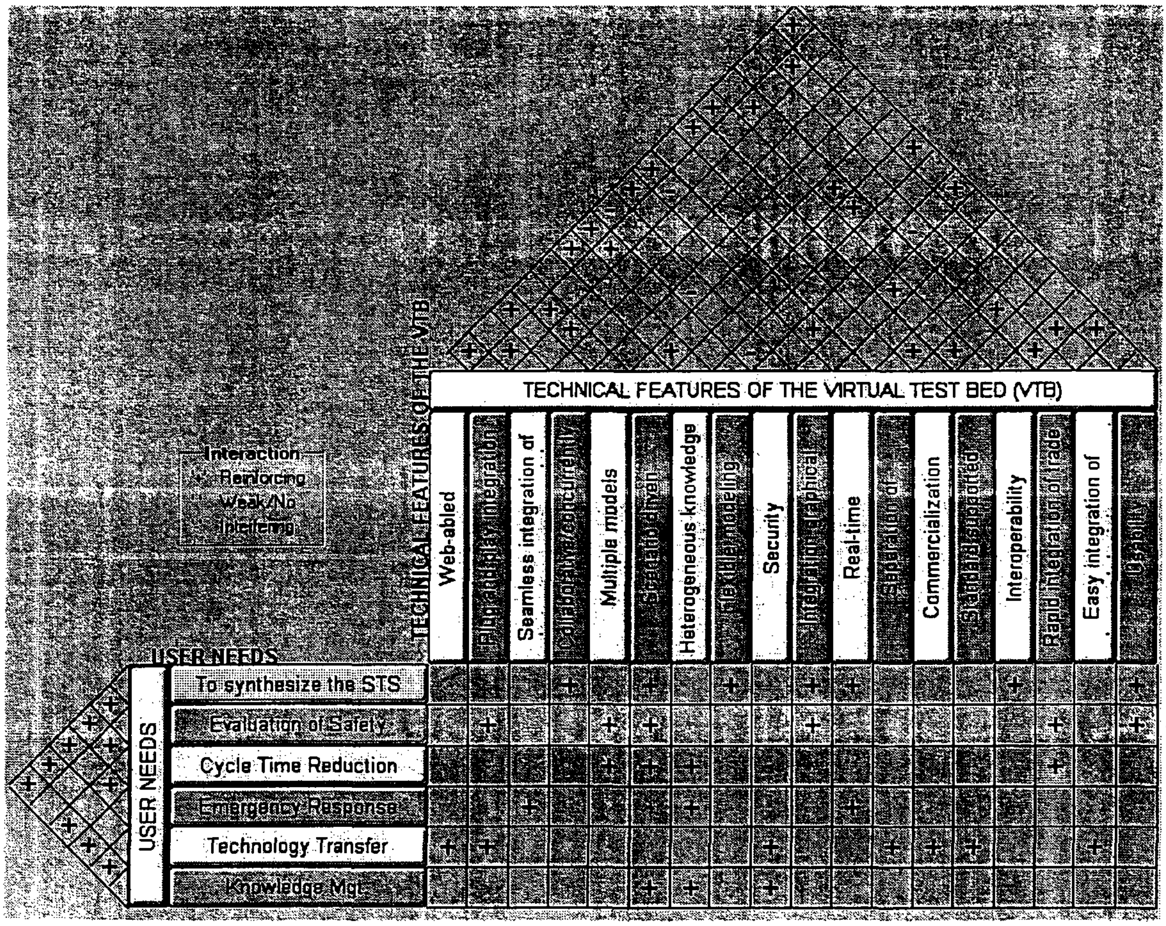

**Figure 2. On going modified House-of-Quality for the VTB project. Look and feel adapted from MOC**

Robert Turner, the project manager of the VTB project, facilitated several meetings with potential users (e.g., Federal Aviation Administration (FAA), NASA Wallops Flight Facility (WFF), NASA Range Safety, NASA-KSC Environmental Office, NASA-KSC Weather Office, Space Launch Initiative (SLI), NASA ARC Researchers and coordinators, the ER Range Operations Control Center (ROCC), and others). The potential users "verbalized" their needs and benefits desired. These "verbalized" needs and benefits were clustered in six areas:

1. *To decrease cycle time during the evaluation of vehicles and systems compliance to safety criteria.*
2. *To improve the evaluation of vehicles and systems compliance to safety criteria.*
3. *To synthesize in a single view the different elements of the Space Transportation System.*
4. *To develop precautions/emergency response measures to support range safety & security in a systematic manner.*
5. *To accelerate the introduction of new technologies (technology pipeline), new operational procedures, and help configure the current ones.*
6. *There is a need for a repository of knowledge and Knowledge Management strategies.*

The next step was to translate these needs/benefits to the desired design requirements/technical features of the VTB. The identification of the different features was the product of discussions with Henry McCoy. Henry is the Scientist assigned for this project by All Points Logistics. In addition, the different interactions among the needs, technical features, and between the needs/benefits and the technical features were carefully studied. I would like to say that this in an on-going process. The ranking of the technical features is providing a guideline for the selection of a viable computing architecture for the VTB project. We understand that these technical features are very ambitious to obtain and some of them required trade-off to be implemented. The following is a listing of the technical features:

**Web-Abled:** The VTB must be able to use the Web and Internet resources. This will guarantee the distributed nature and user acceptance required for the different applications of the VTB.
**Plug & Play Integration:** The models should be able to be integrated without complications. Essentially, the models should not be re-written or reverse engineered to be integrated in the VTB.
**Seamless Integration of Different Types of Models:** A spaceport can only be represented using different types of models (sizes and nature). The VTB will have to be scaleable to support the growth in the size of the models. The nature of the models can differ. Examples of these models [8]: (1) Nonlinear first principles models of systems or subsystems that are based on the physics and chemistry of their underlying phenomena; (2) Linear regression models derived from analysis of steady-state system input/output data; (3) Fuzzy rules expressing a human expert's predictions of system behavior; (4) Linear state-space models developed by linearizing first-principles models; (5) Neural network models for inferring properties of a system; and (6) Discrete-event models.
**Collaborative/Concurrently Environment:** The computing environment should allow the implementation of collaborative design spaces (not limited by geographical constraints). This should be also implemented in a concurrent fashion.
**Multi-Model:** To represent the range and operations of a spaceport multiple models describing the different elements and processes will be integrated. In addition, scaleability (from the number of models to be hosted) is an important design requirement. The existence of scaleability allows for smooth growth of a computer system.
**Scenario Driven:** Scenarios representing impediments or lessons learned should be able to be executed and archived.
**Extensive/Heterogeneous Knowledge Repository:** The databases and knowledge bases should have enough capacity to handle large databases of numerical data, sounds, pictures, videos, and other media. In addition, knowledge integration of different forms is expected.
**Flexible Modeling Environment:** To satisfy the needs the VTB must provide a flexible modeling environment. This flexible modeling environment should be able to allow for queries and what-if scenarios. Ontologies could be used to provide the semantic primitives for a modeling language.
**Real-Time Environment:** There is a need in launch operations for real-time simulation and decision-making. It is important to have a real-time baseline and feedback.
**Security Layers:** Security has to be implemented in different manners and layers to provide the required level. Schemes based on SSL, Passwords, IP-addresses, and software agents could be designed to provide the required level of security.
**Integrated Graphical Environment:** Intelligent Visualization should be the result of this integration. The user should be able to see the output in different media and ways. The environment can be integrated with Visual Query Languages (VQLs).
**Separation of Control/Application Logic/Architecture/Domain Knowledge:** The VTB architecture should be independent of the domain knowledge. This will facilitate the technological products derived from the VTB platform. Maintenance and software life cycle assessment process can be easily implemented.
**Commercialization (Business Development):** One of the goals of NASA is to commercialize the technologies developed.
**Standard Supported Protocols:** Standardization is an important goal in the design. This will provide a cost-effective development process and user/industry acceptance.
**Interoperability/Cross Platform Capabilities:** Metadata mappings among different formats, common representations, reusability, and cross platform capabilities are envisioned capabilities.
**Rapid Integration of Trade Studies:** The rapid integration of trade studies for road mapping and technology transfer activities is a strong design requirement (in specific for groups such as ARTWG and

ASTWG).

**Easy Integration of Advanced Decision Support Tools:** One of the objectives of the VTB project is to advance the decision-making capabilities for range and operations (e.g., more risk management and less risk avoidance).

**Usability (and User Friendliness):** Human-centered computing principles taking into consideration the physical and cognitive models of the users must be implemented. The user should be able to configure the interface based on his/her needs.

## 4. Models and Process Evaluations

The CALPUFF model was reviewed from its computing characteristics (following the list of deliverables - The list of deliverables clearly stated the determination of computing characteristics as the only obligation for KSC). In addition, the subsection related to BLASTX explains the current flow of activities (operational and computational) to calculate the $E_c$.

### 4.1 The CALPUFF Model

CALPUFF is a Lagrangian puff dispersion model of the "plume segment" puff model family [6,7]. CALPUFF is applicable to complex terrain and coastal situations. This model is a multi-layer, multi-species non-steady state puff dispersion model. The user has to define a grid size for high-resolution simulation of episodes as well as for runs for one year or more with a one-hour time step. It can simulate the effects of time- and space-varying meteorological conditions on pollutant transport, transformation, and removal. This dispersion model is used for environmental impact assessments, and studies of air quality and pollutant transport on regional scales. Different forms of meteorological input can be used by CALPUFF. The time-dependent three-dimensional meteorological fields generated by the diagnostic meteorological model CALMET (which is a component of the CALPUFF modeling system), is the preferred form for regulatory application. The code has been used and reviewed by several different organizations for long-range transport distances, intermediate distances, and short to intermediate distances (e.g., EPA, NASA Marshall Space Flight Center).

The standard version of the code is Version 5. This version does not have the ability to accept and process changes in emission sources that occur over times shorter than one hour. However, the time-stepping algorithms within CALPUFF allow it to take very short time steps if necessary. NASA Marshall contracted the authors of CALPUFF, Earth Tech, Inc., to modify the code. Modifications were made to CALPUFF and CALPOST Version 5 to create Version 6. Version 6 has the following additions:

- □ The dispersion modeling system can accept emissions data with temporal variations as small as 1 second, and report average concentrations for intervals as short as 1 minute.
- □ The user can change the value of the variable EPSRAD via the CALPUFF control file. EPSRAD is the emissivity used in the radiative heat loss component of the energy equation of the numerical rise module for buoyant area sources.

These modifications contracted by NASA Marshall Research Flight Center (MRFC) were part of an effort to analyze PC-based modeling capabilities for the dispersion of hydrogen chloride from the catastrophic failure of a Space Shuttle early in flight [2,17]. The purpose of this initiative by NASA Marshall was to [2] *"support the development of a Launch Commit Criteria (LCC) and complimentary Emergency Response Planning for the launch of a Space Shuttle."* The source codes, executable codes, ancillary programs, and user manuals and reports for CALPUFF 5 and CALPUFF 6 were obtained. They were reviewed to obtain their computing characteristics as summarized in Table 1. These two programs were executed and tested with the example grids provided. We tried to translate from Fortran to either Java or C. The problem is that this required a better translator. In addition, we contacted Jeff Anderson and Joe Scire (from Earth Tech) to obtain more information about the modifications made to CALPUFF. The computer programs and the documentation were delivered to Jorge Bardina (NASA ARC on June 26, 2002). Jorge's group will take of the integration (with DARWIN) and modifications to the model.

### 4.2 Process Flow for Safety Risk Evaluation Taking into Consideration Blast-effect modeling.

Blast risks are estimated using BLASTX at the ER. BLASTX uses wind and temperature profiles to determine the risk of casualties. This process of calculating the Ec using BLASTX is a very interesting one. We have studied a process flow for the different activities. The activities can be in order, however, some of them can occur concurrently or be repeated [3]. We studied these activities and documented them with examples (due to restrictions of the length of this report, these activities are not included).

**Table 1. Computing characteristics of CALPUFF [6,7,17]**

| | |
|---|---|
| **Which machine it currently runs (Platform)** | PC with WINDOWS (if GUI is used) or DOS |
| **Specify language it is written in (Language)** | FORTRAN (approximately 51K lines). There are no versions in C, C++, or Java. |
| **Input** | File containing the filename and path for each of the input and output (I/O) files used in the current run. Geophysical and hourly meteorological data. Source and emissions data for point sources with arbitrarily varying emission parameters. |
| **Output** | Unformatted data files containing gridded fields of time-averaged concentrations, time-averaged dry deposition fluxes, and time-averaged wet deposition fluxes are created with each run. |
| **Size and storage requirements (Disk space requirements)** | 3 MB for unzipped files, 18 MB for installed system. |
| **Run execution time and memory requirements** | The memory required by CALPUFF is a strong function of the specified maximum array dimensions in the parameter file. For studies involving long-range transport, memory requirements will typically be at least 8 MB, with more required for simulations involving large numbers of sources. The run time of CALPUFF will vary considerably depending on the model application. Variations of factors of 10–20 are likely, depending on the size of the domain, the number of sources, selection of technical options, and meteorological variables such as the mean wind speed. From our own experience, and using CALPUFF 6, the execution is very fast for small grids. However, the execution time grows exponentially when the size of the grids increases. |
| **Disclose and provide source code as well as executable code (Source & Executable)** | Source Code and Executable Code was obtained and delivered for CALPUFF Version 5 and Version 6 to NASA ARC (Jorge Bardina). |
| **Other Interface Issues** | There are some interfaces/GUI developed with some minor graphics capabilities. However, these interfaces work in batch mode (and they are very primitive). In addition, NASA Marshall reports that in order to efficiently run CALPUFF for large sets a batching system was developed using a set of Visual Basic macros executed via Microsoft Excel. |
| **Points of Contact** | **Jeff Anderson, Ph.D.** B.J.Anderson@msfc.nasa.gov<br>**Joe Scire, Ph.D.** Atmospheric Studies Group Internet: jss@src.com or<br>EARTH TECH, 196 Baker Avenue telephone: (978) 371-4270<br>Concord, MA 01742 fax: (978) 371-2468<br>**Mark Bennett, Ph.D.** 619-208-7055. San Diego office of CH2MHill |

## 5. Technologies, Techniques, and Information Tools Evaluations and Recommendations

This section will discuss several technologies, techniques, and information tools researched during this fellowship.

### 5.1 Leveraging Current NASA ARC Developments

Henry McCoy and I researched several of the developments of NASA ARC to see the suitability for the VTB project. Two technologies were found with good potential: DARWIN (an intelligent database) and VLAB (a virtual reality program that allows the user to configure his/her interface).

- DARWIN. DARWIN is based on a three-tier client/server/server architecture. The first server is the DARWIN Server. It runs a secure HTTP server, the executive server software, and the meta-database. The DReAM server, which is hosted in the aeronautics test area of choice, is the third tier server. The remote DReAM client system hosts the DARWIN Workspace environment that is composed of a secure HTTP browser, an executive, and a tool kit of additional visualization and collaboration software. Java or JavaScript are the programming environments of choice to implement the "small" remote applications. Implementation of larger applications, such as remote video tools are authorized and initiated through the DARWIN server (larger applications can be also initiated from the client site, however, knowledge of the middleware becomes even more critical for these applications). DARWIN can be used as an intelligent, distributed, and scaleable repository for the models, scenarios, and multi-media data used by the VTB.

- VLAB. VLAB is currently an environment programmed in the C language. It implements a virtual laboratory where the user can walk and select the different displays and databases that he/she wants to visualize in her/his desktop. It is currently used in NASA ARC to visualize the VMS. VLAB can be modified to serve as a way for the user of the VTB to configure his/her display and collaborate with other researchers (geographically distributed).

### 5.2 2D and 3D Graphics and Animation Environments

Several environments were researched. Among them, OpenGL, OpenFly, and Multigen Authoring Software.

- OpenGL. OpenGL [9,10,12,16] is the premier environment for developing portable, interactive 2D and 3D graphics applications. A low-level, vendor-neutral software interface, the OpenGL has often been called the "assembler language" of computer graphics. In addition to providing enormous flexibility and functionality, OpenGL applications enjoy the broadest platform accessibility in the industry. OpenGL has high visual quality and performance. Any visual computing application requiring maximum performance-from 3D animation to CAD to visual simulation-can exploit high-quality, high-performance OpenGL capabilities. There are various advantages of OpenGL: (1) An independent consortium, the OpenGL Architecture Review Board, guides the OpenGL specification; (2) Backward compatibility requirements ensure that existing applications do not become obsolete, (3) OpenGL API-based applications can run on systems ranging from consumer electronics to PCs, workstations, and supercomputers, and (4) numerous books have been published about OpenGL, and a great deal of sample code is readily available, making information about OpenGL inexpensive and easy to obtain.
- OpenFly. OpenFly [15] is an open source replacement game engine enabling designers of flight simulations created with Domark Flight Simulation Toolkit created by SIMIS Limited (Kuju Entertainment Limited) to take their simulations to the next level of graphical excellence.
- **Multigen Authoring Software [11,13,14].** A key requirement of any modeling system is the ability to import and export models from/to other CAD standards. **On July 1, 2002,** MultiGen-Paradigm, the leading developer of realtime 3D graphics software solutions, announced the availability of version

1.1 of SiteBuilder 3D, an extension to ESRI's popular ArcView® GIS desktop mapping program. SiteBuilder 3D provides users with a solution to quickly and easily transform 2D map data into realistic, fully interactive 3D scenes. In addition, the company is announcing the initial release of ModelBuilder 3D, an optional authoring software toolset that gives users the power to generate 3D models of real-world buildings, objects and vegetation for incorporation into 3D scenes generated by SiteBuilder 3D. Both products facilitate the simple creation of 3D scenes from GIS and geospatial data, without requiring a high-level of technical or 3D modeling experience. By giving users expanded functionality to create 3D terrain models from common GIS data, create paths and export them to digital movie files, SiteBuilder 3D v1.1 is the right choice for an affordable, easy to use 3D GIS application (currently used in the Control Tower application in NASA ARC).

### 5.3 Dccision-Making Technologies

Several decision-making technologies were researched during this summer 2002. A simple example was developed for neural networks. However, due to the limitations of this report, we discuss three of them.

- □ **Bayesian Networks.** A method of reasoning using probabilities, called Bayesian Networks, has become popular in the artificial intelligence community. Bayesian networks are directed acyclic graphs (DAGs), where the nodes are random variables. The random variables can have two values (True and False), or several values (discrete or continuous). The arcs specify the independent assumptions that must hold between the random variables. These independent assumptions determine what probability information is required to specify the probability distribution among the random variables in the network. To specify the probability distribution of a Bayesian network, one must give the prior probabilities of all root nodes (nodes with no predecessors) and the conditional probabilities of all non-root nodes given all possible combinations of their direct predecessors. It is possible to appreciate that the concept of joint distributions plays a very important role in Bayesian networks. Bayesian networks allow one to calculate the conditional probabilities of the nodes in the network given that the values of some of the nodes have been observed.
- □ **Neural Networks.** There are two types of supervised neural networks researched this summer, which can be useful in for the VTB decision-making:
- □ Supervised neural networks based on minimizing errors
- □ Supervised neural networks based on forecasting by analogy.
- □ **Support Vector Machines.** Support Vector Machines (SVMs) are a new generation of algorithms in machine learning. SVMs are based on recent advances in statistical learning theory. There are two interesting features of SVMs:
- □ Complex Sructures. SVMs are able to generate structures that are complex enough to deal with practical applications. These structures contain classes of radial basis functions, neural networks, polynomials and splines-based functions as particular implementations.
- □ Mathematical Analysis. SVMs are able to generate complex structures but they are simple enough to be examined mathematically. SVMs can be conceptualized as a linear method in a high-dimensional feature space nonlinearly associated to input space.

## 6. System Design

Henry McCoy has developed a preliminary architecture ("form") of the VTB. Our contributions here were to validate his concept. We had many discussions about this architecture. We added some concepts to this architecture such as the utilization of Ontologies based on the Semantic Web Project. In addition, currently, we are developing a concept for a decision-making tool for safety evaluation/risk management of legacy and new vehicles.

### 6.1 Preliminary Concept

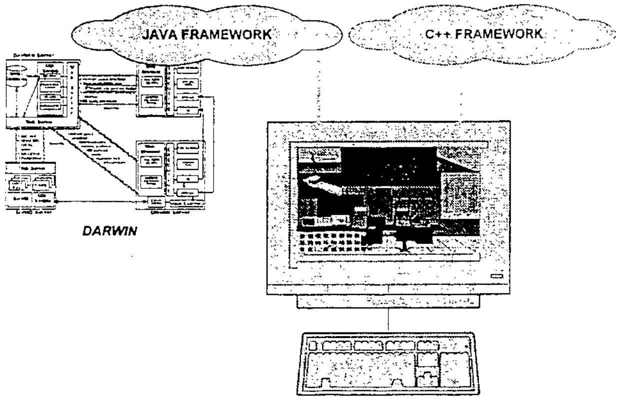

Figure 3. Preliminary architecture of the VTB.

There are intellectual properties issues. Therefore, we will sketch some of the concepts used in the preliminary architecture. The preliminary concept utilizes two frameworks: a C++ framework and a JAVA framework. The JAVA framework is used to provide the necessary services/handshakes with DARWIN. DARWIN is utilized as a repository of multi-media information, knowledge, and the models. The C++ framework is very important at the client side to implement the simulator and the client interfaces. VLAB will be modified (and translated to C++/object oriented) to operate in the client side. Ontologies will be used to develop a modeling language depending on the application. Kernels, security schemes (SSL, passwords, IP-addresses, and software agents), and innovative schemes of software re-configuration and data management are contemplated for this architecture.

### 6.2 Decision-Making Tool for Safety Evaluation/Risk Management

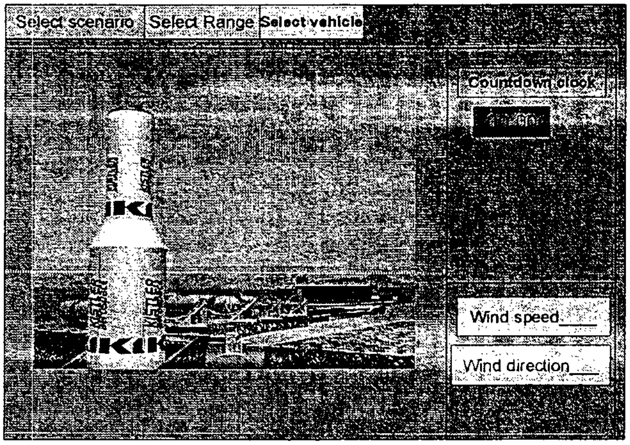

Figure 4. Look & Feel of a Decision Support Tool to evaluate safety/risk management.

The aerospace industry is developing commercial RLVs [18], some of which may someday operate from the ER. "The basic safety criteria for RLVs should be the same as for expendable launch vehicles in terms of $E_c$, $P_c$, and $P_i$. However, range safety processes for RLVs require special attention because RLV concepts will vary greatly in design and operational characteristics." [19] The development of a decision-making tool for safety evaluation/risk management of these new vehicles is very important. In addition, RLVs that carry human beings raise additional safety of flight issues, especially with respect to the acceptability of autonomous or semi-autonomous FTSs. [19] We have been supporting the conceptualization and providing form to this tool. The development of this tool will require the VTB to host different models:

- □ Vehicle component reliability
- □ Nominal trajectory and expected variations from nominal
- □ Vehicle failure modes, probabilities, and effects
- □ Wind modeling and debris-dispersion
- □ Population and geographical
- □ Debris
- □ Blast-effect
- □ Toxic-effect
- □ Impact limit lines
- □ Instantaneous impact point

- Destruct lines
- Collision avoidance

The VTB will have to host knowledge base systems containing knowledge and with capabilities of inferencing about:

- FAA regulations
- EWR 127-1
- Case Scenarios
- Tree analysis and interpretation
- Controllers/Schedulers of knowledge bases

In addition, optimization engines with capabilities to:

- Optimize trajectories
- Selection of Gates
- Compare scenarios (e.g., different fuel types, different ranges)
- Maximize or Minimize Objective Functions Provided by the Users

**References**

1. Barth, T., "Found in Space," IEE Solutions, April 2002.
2. Bennett, M., Technical Memorandum: Launch Commit Criteria Modeling Assistance, Analysis Summary, May 6, 2001.
3. Boyd, B., Harms, D., Rosati, P., Parks, C., and Overbeck, K., Wheather Support to Range Safety For Forecasting Atmospheric Sonic Propagation, Ninth Conference on Aviation, Range, and Aerospace Meteorology, 2000.
4. http://ccs.mit.edu/moc/.
5. Command Control Technologies Corporation, Intelligent Launch and Range Operations Testbed: Concept Paper, December 6, 2001.
6. Earth Tech, Addendum for CALPUFF Version 6, 2002.
7. Earth Tech, User Manuals for CALPUFF Version 5, 1997.
8. Godblole, D., Samad, T., and Gopal, V., "Active Multi-Model Control for Dynamic Maneuver Optimization of Unmanned Air Vehicles," Proceedings of the 2000 IEEE International Conference on Robotics & Automation, April 2000.
9. http://www.berkelium.com/OpenGL/
10. http://www.codeguru.com/opengl/index.shtml
11. http://www.dawnbreaker.com/navy_briefings/antiterror.html
12. http://www.frii.com/~martz/oglfaq/
13. http://www.multigen.com/news/pr/0701_02.shtml
14. http://www.multigen.com/news/pr/0309_00.shtml
15. http://www.openfly.org.uk/
16. http://www.opengl.org/
17. NASA MRFC Environmental Office Contacts.
18. National Research Council, Reusable Launch Vehicle, National Academy Press, 1995.
19. National Research Council, Streamlining Space Launch Range Safety, National Academy Press, 2000.
20. Portanova, P. and Tomei, E. Quality Function Deployment in Launch Operations, Aerospace Report No. TOR-0091 (6561-04)-1, November 1990.
21. Rechtin, E., Systems Architecting, Prentice Hall, 1991.
22. Rechtin, E. and Maier, M., The Art of Systems Architecting, CRC Press, 1997.
23. Zapata, E., " A Quality Function Deployment Method Applied to Highly Reusable Space Transportation," Space Technology & Applications International Forum in Albuquerque, NM, January 26-30, 1997.