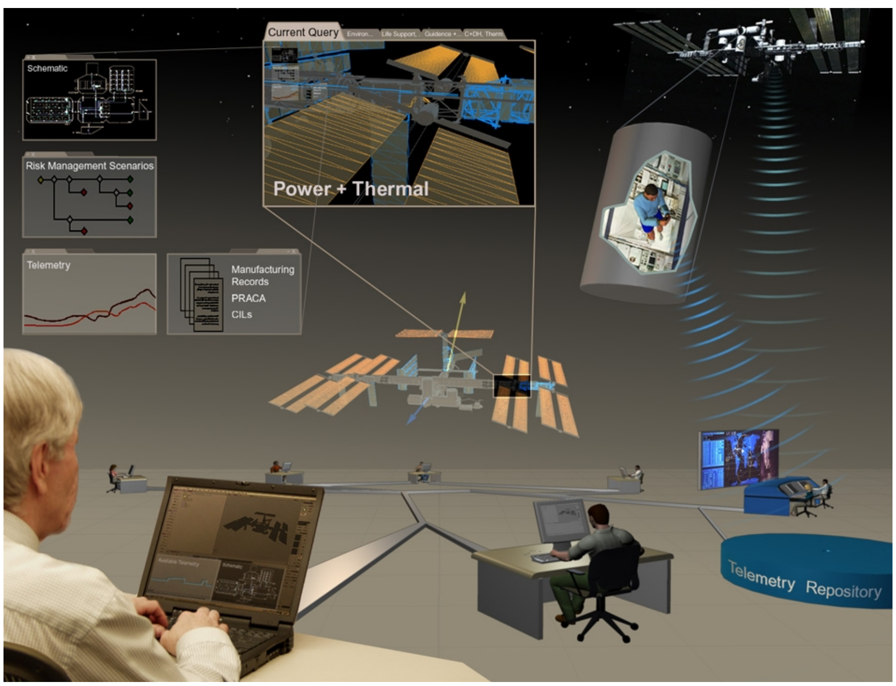
White Paper for the NASA Workshop on Virtual Iron Birds

SimStation: A Knowledge-Integrating Virtual Vehicle

Mark Shirley and Tom Cochrane

April 2, 2004
Naval Postgraduate School
Monterey, CA

SimStation is a *Knowledge-Integrating Virtual Vehicle*. It is a 3D CAD-based visualization-model integrated with functional and behavioral models of the International Space Station (ISS). It is intended to assist vehicle system engineers as they research and evaluate design and operational tradeoffs. Here is an illustration of the idea:



The illustration shows an imaginary case of an engineering team researching a concern with the rotary joints that keep the solar array wings pointed at the sun. The team shares ready access to geometry, schematics, telemetry histories, risk scenarios, and engineering documents. A behavioral simulation is in use to evaluate operational tradeoffs, such as how changes in vehicle attitude control strategies will reduce wear on the joints. The datasets are cross-indexed and linked within the context of the analysis, so the team spends less time looking for basic information and more time thinking about solutions and communicating them within a shared environment.

There are already a great many tools for simulating and managing the ISS. For the most part, however, these tools are disconnected from each other. They're useful because skilled engineers run them while working within a large engineering organization. Similarly, NASA and its contractors already have an enormous and reasonably effective engineering repository. However, it is a *document* repository. While documents are currently the lifeblood of engineering organizations, they cannot be operated on by software and combined to answer unanticipated queries.

Given the constraints of these tools, it takes years of intensive study and working experience for engineers to develop a *big picture* understanding of space station design and operations. Knowledge is widely scattered and no tools exist that cover all aspects of station design and operations. SimStation's goal is to span this range of knowledge to a moderate depth and provide guidance to appropriate in-depth tools. As such, SimStation sits on top of the document repository, helping users navigate to find answers in ways complementary to the full-text indexing capabilities already available. Similarly, SimStation will in a sense sit on top of existing high-fidelity simulations, by providing a quick-look capability to help system engineers brainstorm potential solutions quickly and decide which potential solutions merit more costly detailed analysis.

The initial customer for SimStation is the ISS VIPER team. The mission of the Vehicle Integrated Performance Environments & Resources Team (VIPER) is to provide vehicle systems engineering, including optimization of vehicle integrated performance, vehicle resources, system architecture, vehicle configuration, and mission design. This mission involves the constant interplay of extremely broad analyses encompassing most vehicle subsystems, contrasted with highly focused and detailed case studies.

To address the dual nature of the challenge, SimStation builds upon two powerful ideas: System Dynamics (Forrester, Jay W. 1961 Industrial Dynamics, Portland OR, Productivity Press) and Mirror Worlds (Gelernter, Mirror Worlds).
System Dynamics or Systems Thinking was developed by Forrester as a method of understanding and characterizing extremely complex systems like ecologies or economies. Whereas traditional forms of engineering analysis focus on separating the individual pieces of what is being studied, Systems Thinking focuses on how

the individual pieces being studied interact with the other elements of the system. Systems Thinking involves including more and more interactions to study the often non-intuitive behavior that immerges from those interactions. System Dynamics has proven its value in a wide range of situations, particularly in complex problems that involve helping many actors see the *big picture* and not just their localized portion of it. This approach has been popularized by the SimCity series of educational software games developed by Will Wright, Jeff Braun and a development team at Maxis/Electronic Arts, and by a wide range of business simulation packages. By choosing the name SimStation, a name that echoes these predecessors, we have set a very high bar for our project.

The second powerful idea in SimStation is Gelernter's concept of a Mirror World [Gelernter93]. A Mirror World is "a software model of some chunk of reality, some piece of the real world going on outside your window. Oceans of information pour endlessly into the model (through a vast maze of software pipes and hoses); so much information that the model can mimic the reality's every move, moment-by-moment. A Mirror World is some huge institution's moving, true-to-life mirror image trapped inside a computer - where you can see and grasp it as a whole." (Gelernter, Mirror Worlds, P.3). Mirror Worlds are a technology for supporting systems thinking. Their goal is to provide broad context and access to detailed information to improve the ability of actors in very large institutions to deeply understand the *big picture*, their role in it and how their actions impact other actors, either directly or indirectly. SimStation can be connected either to real-time ISS telemetry or recorded telemetry. However, the process is the same.

SimStation is an instance of a virtual vehicle that is actually a knowledge management system for the vehicle. We want to disseminate SimStation broadly throughout the ISS engineering community, and a limited access version to the public for education and awareness of the program. Many SimStation design choices stem from these twin foci of wide dissemination and helping people explore the big picture, notably in the areas of simulation level-of-detail and in the use of commercial off-the-shelf technology.

SimStation runs on engineering desktops and laptops and can perform some functions standalone, particularly the 'quick look' trade studies. We are also developing a web-based version, called SimStationOnLine, with a subset of capabilities.

The *complex systems* description fits the ISS perfectly and aerospace systems in general. We believe that new ideas are needed to enable engineering organizations to meet the challenge of NASA's new exploration vision. How NASA solves the challenges of the International Space Station will form a baseline for future exploratory missions.

The remainder of this white paper is organized into several sections:

- **Customer Needs**
- **Collaborative Approach**
- **System Components**
- **Status**
- **Conclusions**
- **Acknowledgements**

Customer Needs:

Simulations and knowledge management are increasingly requirements for the development of ambitious aerospace vehicle projects. Aerospace systems are challenging to manage, and system interactions are growing more complex. Success often depends on a limited number of people with amazing abilities to mentally visualize complex systems. Sharing these visions within the team is a challenge that we are trying to support with SimStation.

We observe that working meetings frequently come to a halt at the point where a senior person asks how something is happening, and a team leader has to answer with a promise to come back with a response within the next few days. We imagine great gains in efficiency if a team can access existing knowledge right there when the question first comes up, and then continue the reasoning process with minimal interruption. We seek to provide that sort of access to information.

As an organization, NASA is more family than bureaucracy. NASA's high profile disasters are taken very personally by all involved, and have been thoroughly studied and reviewed over and over again; notably Apollo 1, Apollo 13, Challenger, and Columbia. These have been very expensive disasters, physically, financially and emotionally, and all the more so in that the public shares in this. In retrospect each of these disasters could have been prevented by better understanding, communication, and foresight. Nobody on the NASA team wants to repeat such lessons. Getting a coordinated view is at best extremely tedious and bordering on impossible. Sharing a coordinated view through a knowledge management system is a high priority for all

Providing improved situation awareness and communication is at the heart of the challenge:

- On Apollo 1, a textbook-level knowledge of the hazards of an oxygen rich environment at normal atmospheric pressures was overlooked until a crew was lost by fire during training.
- On Apollo 13, during the early stages of failure, the mission control team was assuming their sensor indications must be wrong, even though they had the crews' report of an explosion on board. The delay nearly caused loss of the crew, as precious oxygen leaked away from the ship. And the explosion itself was caused by a failure to communicate a design change throughout the organization, compounded by other communication errors.
- On Challenger STS-51L there was abundant warning of the failure of booster O-rings at low temperatures, but the information was not correctly prioritized in front of crucial decision makers.
- On Columbia STS-107 there were stacks of warning memos that again didn't get correct prioritization in front of crucial decision makers. And during the early stages of the (by then unavoidable) disaster, there was again an assumption of incorrect sensor readings.

Situational awareness and communication are crucial to survival and success, and we need all the help we can get with complex space systems for NASA's exploration missions. The SimStation project is making major strides in that direction. A recent example here is that an early version of the SimStation geometry was being explored by a senior engineer who has been deeply involved with Station for over ten years, and in the space of browsing for only a few minutes, he noticed two important components that he had not seen before.

The challenge to understand system interactions on Station grows more complex with each mission. ISS represents a large multiple-constraint optimization problem, feeding a large scheduling problem; two of the more challenging fronts in computer science. Station is the largest flying vehicle ever built, and being assembled while in flight. Each addition of a new element changes Station configuration, management and handling characteristics. Planning, negotiating and coordinating systems design and resource assignments within the team are extraordinarily challenging, with sixteen international partners and their commercial and academic partners working together to build Station. When complete, Station will be the largest multi-science lab ever assembled. Station also serves as home, living quarters, office and workshop for crew. Many Station systems run under computer control, using software and hardware that cannot be fully tested in their real operating environment prior to being deployed for use on-orbit. Coordinating international partners and their commercial and academic partners is organizationally and geographically overwhelming. No full-vehicle physical engineering model is technically or financially feasible. Performing "what-if" studies upon multiple system configurations is very labor intensive. Procedures generation, validation and training overstretch current organizational and resource capacity. Determining the functional impact of anomalies requires extensively distributed data resources. And the unknown unknowns present the highest risks. Any issues not resolved on the ground get shipped upstairs to be resolved on the fly by a few astronauts.

While the initial customer is VIPER, SimStation is designed to be generally applicable in the broader contexts of operations, training and management, and SimStation has received extensive contributions of knowledge from ISS representatives in those areas.

Another application of SimStation is under development for the EVA & Crew Systems Office to support a space walk on the next shuttle mission. This **SimStation Procedure Module** links text procedures to a visualization of the execution of those procedure steps. The intention is to use the SimStation Procedure Module for crew refresher training and operational awareness on this crucial Return-To-Flight (RTF) mission.

Collaborative Approach: SimStation's place within the wider group of ISS modeling projects

The vision of creating a virtual space station isn't new with the SimStation project. SimStation builds upon several other NASA projects, most notably:

- The Intelligent Synthesis Environment Program's Station **Large Scale Application**, which continues as **SEE**, the Synergistic Engineering Environment, developed by the Spacecraft & Sensors Branch, Aerospace Systems, Concepts & Analysis Competency at NASA Langley in Virginia.
- The **Intelligent Virtual Station** (IVS) developed by the Smart Systems Research Lab (SSRL), in the Computational Sciences Division at NASA Ames Research Center under funding from several programs: CETDP, ECS and CICT.
- **Birds Eye View** (BEV), by NASA JSC Engineering (predating both of these).

SimStation draws heavily on the work and ideas of the individuals involved in these efforts, and in many cases the same individuals are involved directly in developing SimStation.

In addition to these, there are many other projects that couple visualization of the ISS with some analysis capability, notably:

- **DOUG Dynamic Onboard Ubiquitous Graphics** by the NASA JSC VR Lab
- **Intelligent Flight Support System** at NASA JSC MOD QUEST Lab
- The Space Station Visualization at NASA GRC Immersive VR Lab
- And others…

We are coordinating with each of these projects and understand the elements of similarities and differences. For instance, SimStation differs from the Station LSA in de-emphasizing its multi-user, immersive virtual-reality vision and its goal of integrating the ISS program's existing, currently incompatible, high fidelity subsystem simulations. SimStation instead targets a quick-look capability with a lower fidelity requirement in order to provide integration and flexibility. The move away from immersive VR to desktops and laptops is in recognition of the way engineers currently work at NASA JSC and most other organizations. This same move to widely accessible computers was made by SEE, which is supplying critical simulation components to SimStation.

ISS modeling is a large and complex area of work, and it is appropriate to have a range of related efforts. SimStation's approach to working within such an environment is to build and share components within NASA. We seek to co-create a pool of reusable software components and models. From that pool, will be drawn the components used to construct the SimStation application.

SimStation is a tool development platform as well as being a tool itself. SimStation uses the standards and conventions of .NET, one of the widely used enterprise IT platforms. SimStation contains components implementing a range of vehicle-independent capabilities as well as the vehicle-specific models for Station.

SimStation is becoming an effective platform for building custom analysis software that can benefit from the integrated 'backbone' vehicle representation. In particular, many safety-related analysis and synthesis tasks involve combining information across the traditional engineering disciplines. Applications in which we have done exploratory work include training, procedure development, diagnostics and anomaly response.

SimStation Components:

At the heart of SimStation is a first-order, multi-subsystem model of Station's *structure, function and behavior*.

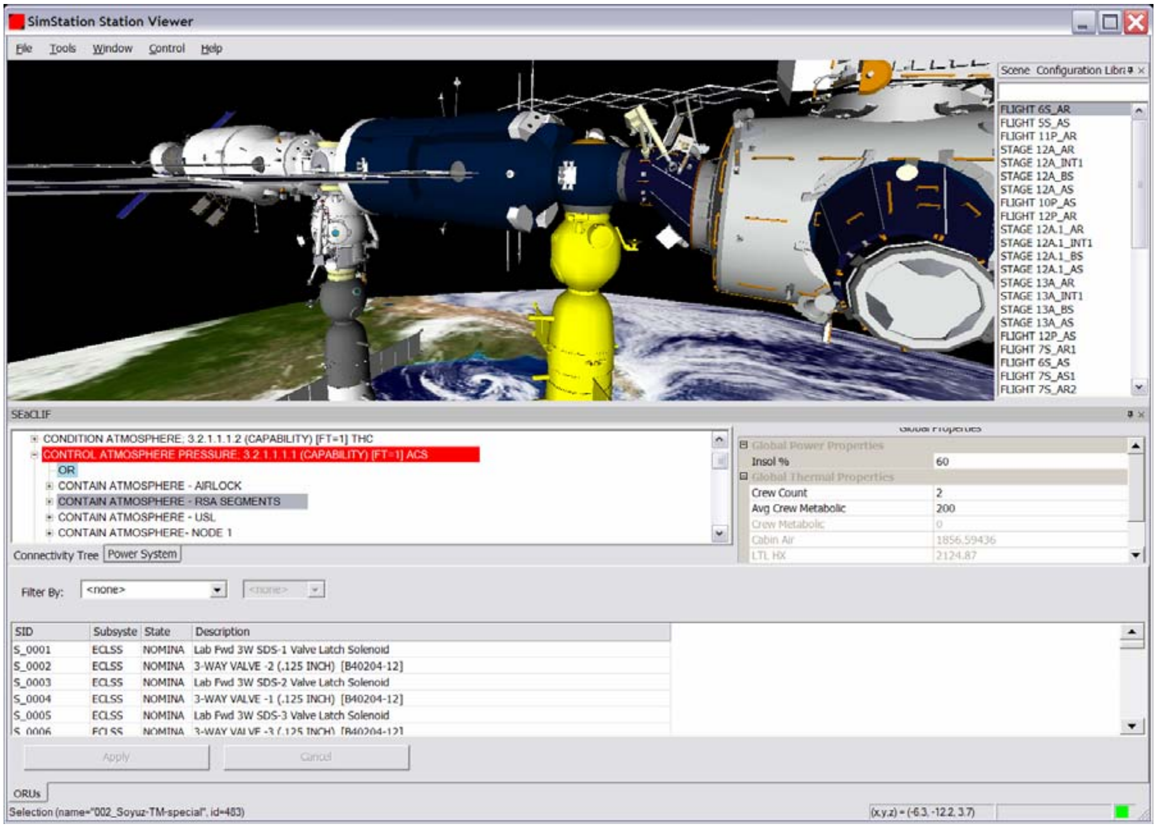
By *structure*, is meant the physical parts, including their 3D geometry, how they are connected, and their spatial inter-relationships as derived from station program qualified CAD models, schematics, functional block diagrams, and related data sources.

By *function* is meant what roles the parts play or what mission requirements they support. This dataset is derived from a combination of the station program qualified reliability and risk models. This combination of the structure and functional representations is referred to as the **SimStation Connectivity Model**. being developed by a modeling team at NASA Johnson Space Center in Houston. Parts and functions in the connectivity model are associated with a wide array of properties, e.g., mass and volume. This also includes mappings between the various identifiers used for a given part, as found in the many databases maintained by the Station program. This crosswalk between datasets is the **SimStation Rosetta Stone**. The SimStation Connectivity Model supports a wide range of user queries, such as:

- What components are supplying power to a particular rack?
- What else will fail if that component fails?
- What are all the electrically powered components in the US lab that support air circulation (perhaps sorted by power draw)
- What components in the airlock store energy?

These common query types currently require considerable time and expert effort to analyze from multiple information resources.

By *behavior* is meant a simulation model (including faults) that can be used to explore and optimize operational scenarios. SimStation contains a multi-subsystem 'quick look' behavioral model, the **SimStation Simulation Module**. Initial work focuses on Power and Thermal and will extend further as resources allow. This requires orbit and attitude simulation as well. The Simulation Module is tightly integrated with the Connectivity Model.

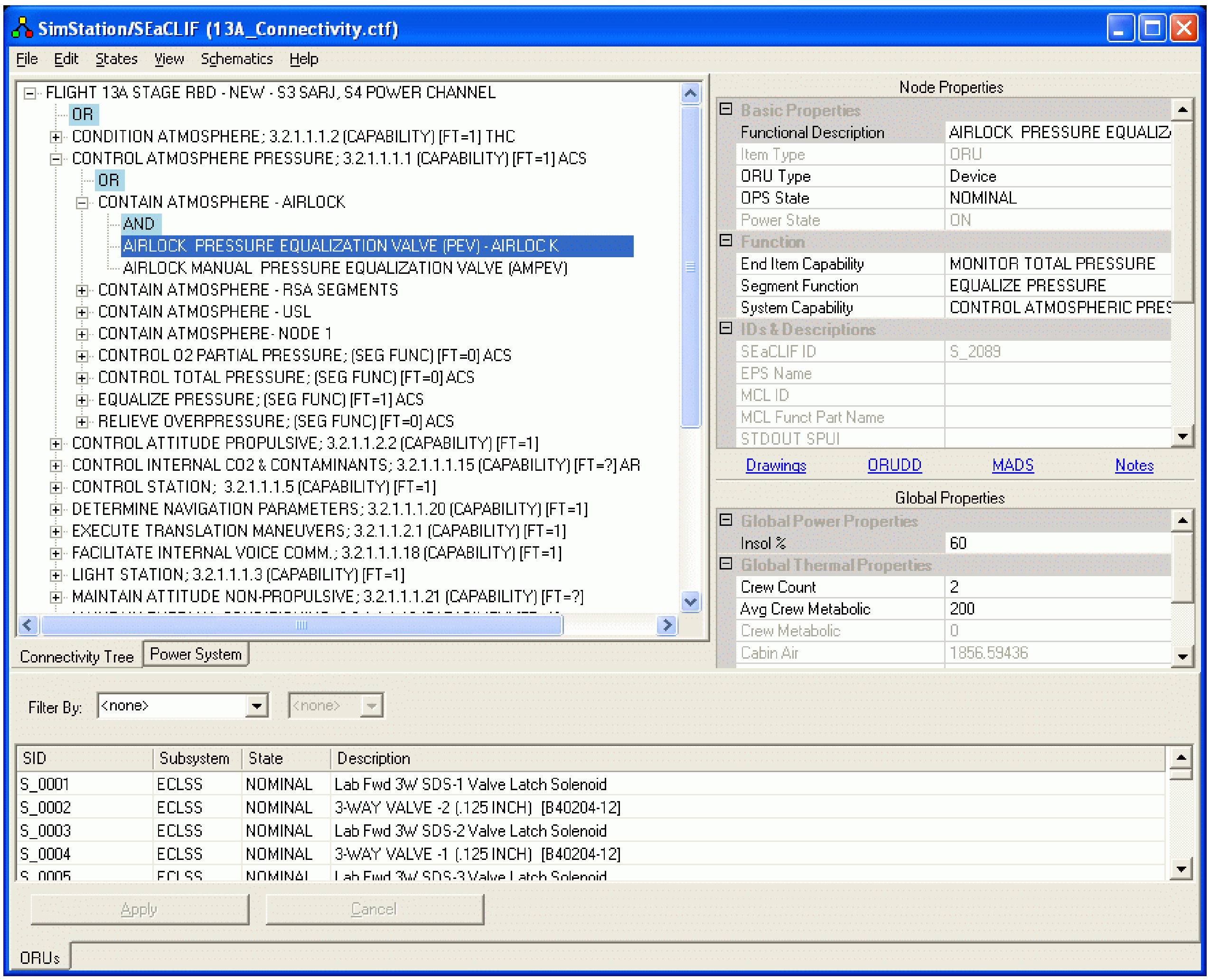


SimStation is not a Product Data Management (PDM) system and does not store master copies of controlled documents. Rather, it is intended to demonstrate a means of building thinking tools on top of a PDM system. An important part of the effort in creating SimStation involves demonstrating the feasibility of pulling datasets from the current ISS databases, integrating them, and organizing them so as to support the system thinking tools in SimStation.

SimStation integrates a subset of the datasets in the Vehicle Master Data Base (VMDB), which form a 'backbone' in the sense that they link to most of the remaining datasets. Due to this linking, the 3D and Connectivity Models in SimStation provide an alternative way to find information in VMDB and JSC DDMS/EDMS that complements the existing web portals. SimStation may allow the option of end-user editing of these datasets under ISSP access controls, and will thus feed information back into VMDB and the PDM system through their change management processes.

The Connectivity Model links to a dataset containing information about the parts extracted and updated from VMDB. This dataset pulls together frequently accessed pieces of information from several VMDB databases, e.g., PUIs, part numbers and other identifiers, masses, and power and thermal loads under different modes of operation. This allows more rapid access to this information. The simulation also uses the power load information. A record of how this information was extracted from VMDB will be maintained, and the capability to check VMDB for the latest information will be present.

SimStation will optionally allow the Simulation and Connectivity models to be edited by users. SimStation will keep track of modifications so that predictions made by the program can be marked if other than a validated dataset was used to make them. The granularity with which these modifications are tracked is TBD.



SimStation supports several modes of operation. We expect users to switch between these modes frequently and rapidly:

- Explore Geometry

- Explore Datasets
- Explore Connectivity
- Explore Procedures
- Explore Simulation Scenarios
- Explore Trade Studies

Here are the major SimStation reusable components:

- **SimStation Top-Level Interface** is a user interface for the setup and viewing of components integrated into SimStation, under development by NASA ARC IC
- **Geometry Library** is a library of 3D geometry compiled for SimStation from multiple sources and transformed into a standard ISS spatial reference framework. The exterior model comes originally from the Lockheed Martin Systems Engineering Modeling and Design Analysis Laboratory and subsequently processed at NASA LaRC.
- **Orbital Dynamics & Vehicle Attitude Module** is a mathematical correct-to-scale model of the solar system; including the earth, moon, sun, and the ISS; used for example to set up sun tracking/lighting/shadowing to initiate calculations of photovoltaic power generation, developed by NASA LaRC AMA
- **Simulation Module** contains various mathematical models, such as one that takes lighting and shadowing data from the vehicle orbit and attitude along with sail angle to calculate photovoltaic power production, under development by NASA ARC IC
- **Energy Balance Model** calculates battery-state-of-charge based on the power generation simulation and the load sums from the power distribution hierarchy, under development by NASA ARC IC
- **Simulation Variable History Repository** archives the analysis of various scenarios as input for decision-making trade-studies, under development by NASA ARC
- **Scenario Editor & Scripting Language** allows setting up various station configurations for decision-making trade-studies analysis, under development by NASA ARC & JSC
- **Rosetta Stone** is a mapping between the different nomenclatures and part numbers used in the many Station datasets, partially automated by the **Reconciler** tool, under development by NASA JSC
- **Connectivity Model** traces the various modes of connectivity between Station systems and components, developed by NASA JSC. This model currently includes electrical power and cooling connections plus the functional hierarchy. Command and control connections will be added shortly.
- **Various browsers for the connectivity model** under development by NASA JSC

- **Procedures Module** allows stepping through textual procedures coupled with the Avatars Module to view performing the actions of the procedures steps, under development by Common Point, Inc.
- **Avatars Module** under development by HPN, Inc and Common Point, Inc.
- **Problem-Solving Context Manager** allows the user to establish the analysis context for browsing multiple SimStation views, configurations and scenarios while maintaining the same context in each, in design at NASA ARC
- **External Connections Manager** in design at NASA ARC
- **Diagnostic Data Server** provides storage, indexing and retrieval of station telemetry, under development by NASA ARC & JSC ISS Avionics Branch
- **Real-time Telemetry** allows real-time access to station telemetry streamed in Mission Control's custom format, developed by JSC Engineering and Lockheed Martin
- **Textual Database Access Module** allows content + context schema-less browsing of multiple datasets using the NetMark tool, under development at NASA ARC and being piloted on servers at NASA JSC
- S**imStation Online** allows users to explores the 3D station geometry via a browser plugin, under development by Digital Space Commons

SimStation is intended both as a production tool and a demonstration. As a production tool, it is specialized to the needs of the VIPER team. Requirements from VIPER are mostly concerned with having a quick-look simulation capability. Portaling capability to VMDB/DDMS/EDMS/a generic document store is important, but not at the top of their list.

The basic user experience we want is this:

- A user can browse the virtual vehicle using any of several modalities: 3D geometry, schematic (not there yet), part hierarchy, or text keyword search.
- A virtual vehicle is represented by a set of objects and relations between them. These objects represent parts, functions/requirements, pieces of geometry, or procedures. Defining, describing and connecting them are a rich set of text documents.
- When the user clicks on an object (part, function, procedure...), the user can quickly get access to a list of relevant documents that have been associated with that object. Documents can be associated with multiple objects. This association is explicitly stored and managed.
- Also when the user clicks on an object (part, function, procedure...), the user gets a list of implicitly associated objects. This list is created on the fly by generating a set of keywords associated with the object and some current

'problem-solving context' the user is within, and dropped into the text search engine.

- The user can then select a document to display
- The user can enter search keywords and get back matches across all of the 'SimStation' datasets including the parts database, the functional hierarchy, the simulation model and the documents in storage. Hits are merged and presented to the user. It's an instance of the general notion of content + context searching.
- Users can annotate anything. Visibility of these annotations can be restricted to individuals, groups or open to everyone. These annotations are typed, and so can be filtered by the problem-solving context.
- Many objects (parts, functions, procedures) have a spatial locality. The user can do spatially based queries as well, e.g., find all annotations within 5 meters of this spot of type RISK or that contain the words RISK or SAFETY.

Finally, SimStation has a persistent store for more kinds of data types than just documents. This is implicit in the IO requirements (particularly in the representation of accident models), but isn't called out as a generic service available for representing other sorts of objects (concepts) and relationships.

Development Status:

Most of the features described above have been developed over the last few years by several different projects related to what can be called Virtual Iron Birds or Knowledge-Integrating Virtual-Vehicles at NASA. Some of them are in regular use by individuals within NASA human space flight programs. Others are being piloted within these programs. Many of these functions are currently integrated into the SimStation environment under the ECS VIB project. Some functions are in a design stage, and depend on completion of other SimStation capabilities. SimStation is on track for broader deployment within the human space flight program in spring and summer of 2004

Conclusion:

Developing Knowledge-Integrating Virtual-Vehicles is much more than just a software development effort. It is as much an exercise in organizational management and systems engineering. Every step in VIB KIVVR development faces as many challenges in organizational dynamics as it does in technical development.

The International Space Station is arguably the most challenging engineering construction attempted by humans. It is the largest flying object ever constructed. It is flying at 17,500 miles per hour at about 240 miles above the earth. It is being constructed while it is flying. It involves 16 international partners and their academic and industrial partners. Just getting there and back requires tremendous difficulty and expense.

An individual can take on the responsibility for a task of some limited scope, but beyond a task of some relative size, a team is required. Even larger tasks require teams of teams. So we divide up pieces of the larger task among members of the teams. Then we face the challenge of integrating all the pieces back together. Meanwhile the teams have developed proprietary ownership issues towards their piece of the product. And either intentionally or inadvertently, they frequently don't agree on standards, methods and tools. SimStation is an attempt to nurture the seeds of development for a knowledge-management platform upon which people can work together to analyze, communicate, and share solutions to these types of large complex challenges.

Acknowledgements:

SimStation is the work and inspiration of many people, and any attempt to list them will be incomplete, particularly among our NASA senior management, who have provided great interest, inspiration, basic support and guidance; and without whom there would be no project.

At NASA Ames Research Center, Mark Shirley and Tom Cochrane provide SimStation project coordination and systems engineering design, Brad Betts designs and implements core software architecture, Charles Neveu integrates the kinematics engine and many other functional elements. Richard Papasin and Dawn McIntosh have contributed software engineering and analytical support. Rick Alena, Daryl Fletcher, and Charles Lee have developed the Diagnostics Development Server component. Robert Mah built the team that built the Intelligent Virtual Station. Ted Blackmon and William Briggs have been integrating the Virtual Astronaut and the Procedure Tracking components. Bruce Damer and his team have been developing SimStation Online. Ken Fertig and

Sudhakar Reddy of Rockwell Scientific implemented the first version of the energy balance model.

At NASA Langley Research Center, Jonathan Cruz, Scott Angster, Jim Hoffman, Darrell Caldwell, Laura Brewer are contributing software components and much inspiration through their creation of the Synergistic Engineering Environment, a complementary tool also used by the ISS VIPER team.

Many individuals and teams at NASA Johnson Space Center have provided significant inspiration, guidance, knowledge and software components to the SimStation project. Valin Thorn and Bill Spetch have provided much knowledge, guidance, and an application domain within the VIPER team. Dan Duncavage at JSC ISS Avionics has provided endless knowledge, guidance and support and an application domain for the DDS component and database access component. At JSC EVA & Crew Systems Branch, significant support, knowledge, and an application domain has come from Jim Thornton, Matt Myers, and Glenda Laws. At JSC Space Station Training Facility, Barbara Corbin, Mark Johnson, Stacey Hale, Hal Smith, Mike Belansky, and Louis Malone provide much guidance, knowledge engineering support, an application domain and access to resources at the SSTF. Dave Homan at the JSC VR Lab has provided valuable programmatic insight and steering advice for the SimStation team, and technical inspiration with his related DOUG product. At JSC MOD Steve Gonzales, Anthony Bruins and Lac Nguyen have contributed elements from their Virtual Astronaut project and Intelligent Flight Support System. David Throop, Jane Malin, Jennifer Lewis, David Sumpter, Paul Porter, Van Keeping and Phil Lewis, are creating the component/function connectivity dataset and associated software. Lui Wang, Richard Barton and David Wadsworth supplied software components enabling SimStation to interface to Mission Control's telemetry distribution infrastructure.

National Aeronautics and Space Administration
Ames Research Center
Engineering for Complex Systems
http://ecs.arc.nasa.gov