# Model Development within the RASSP Virtual Company Environment

Ray Dreiling and Paul Kalutkiewicz
Lockheed Sanders, Inc.
Nashua, New Hampshire 03063

Mike McCollough
Hughes Aerospace and Electronics Company
El Segundo, California 90245

**Abstract**

This paper discusses a virtual company concept as part of the RASSP (Rapid Prototyping of Application Specific Signal Processors) program's overall technical objectives. This concept involves the collaboration of distributed design teams utilizing the RASSP Design Methodology. This approach requires close coordination among the development team members, machine independent information management, and company independent resource management. Presented are experiences from the RASSP Demonstration project that used a multi-company, multi-platform approach for developing a VHDL description of an IRST (InfraRed Search and Track) processor.

## 1. Introduction

The notion of a virtual company is a fairly recent development that appeared with the advent of global networks such as the INTERNET and the proposed Information Super Highway. In fact, the notion has already been realized in the form of virtual law firms, advertising agencies, and software retailers. The RASSP program presses this concept to include the technical domain by conducting business that requires a design team for the development of a complex digital computer. The technical domain is more challenging because of the amount of detailed information exchanged by the design team. It is the design aspect that poses the biggest problem for a virtual company because of the intangibles involved in its creative tasks. The technical domain also has demanding resource requirements for creating and exchanging associated data in terms of workstation and network capacity. The RASSP Program Demonstration project challenged the technology limits and explored the development of a complex multi-processor digital computer with a distributed design team. The team included three companies: Lockheed-Sanders, Hughes Aerospace and Electronics Company, and Motorola. They shared the same electronic data base and conducted technical interchange transparently despite the fact they were separated by thousands of miles (Figure 1).

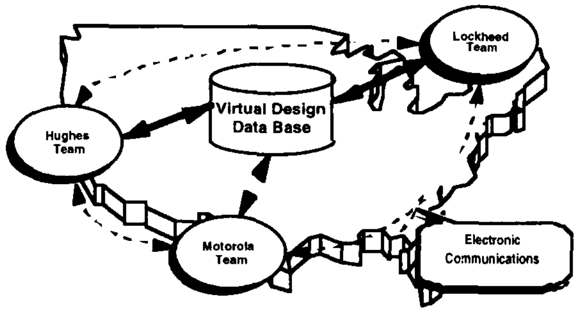


**Figure 1. Virtual Company Concept**

The implementation of this concept allows designers instantaneous access to data and highly interactive and dynamic electronic communication with all design team members. While the technologies to support such goals are still maturing, the RASSP team took a realizable approach by implementing design data bases and technical communications with state of the shelf hardware and software. The cornerstone of the approach was IEEE standard VHDL. The choice was based on the interoperability and design documentation attributes of the language. These features, theoretically allowing designers to share design objects and convey a working understanding of design behavior, enabled the approach to the virtual company. This turned out to be the fundamental challenge.

This paper highlights experiences and lessons learned from modeling in a virtual company with emphasis on managing a VHDL project electronic data base. A description of the issues is given encompassing source code control, infrastructure of project data repositories, and procedures for working within the structure. Presented are current solutions employed by the RASSP Demonstration design team to manage large VHDL data bases in a concurrent modeling environment. Finally, a discussion on lessons learned from the logistics effects of the virtual company that had significant impact on the everyday behavior of the design team is included

## 2. The Current Approach

Operating in a virtual company environment implies a strategy for designing and communicating. Special considerations were made for the characteristics of the design, design cultures and time zones, and resources available at the three companies. The teams adopted an operating strategy based on the following:

- RASSP Design Methodology for Virtual Prototyping based on top down design with VHDL modeling methods.
- VHDL source code management for a multi-designer and multi-team project.
- Design partitioning minimizing team inter-dependency during development.
- Team communication comprising email, tele-conferencing, video-conferencing, and periodic co-location.

The RASSP Design Methodology was utilized because of its strengths rooted in the VHDL language and the Virtual Prototyping concept. The methodology uses a top down design approach producing a totally portable VHDL description. This also includes modeling methods for discrete levels of abstraction that facilitate interoperability. These features are ideal for a multi-company design project that must accomplish virtual system integration and maintain a design data base.

The methodology required a system of procedures for data management across the virtual company. A data promotion scheme was developed that allowed incremental levels of dependency and maturity for source code. The three levels allowed for: Working levels for individual designers to develop and modify source; Gold levels that were used for intermediate models that are visible only at a particular team site; and a Solid Gold level that holds the most mature source code that is visible across the entire project. How and when the models are promoted to the particular levels is governed by two promotion procedures (Figure 2).

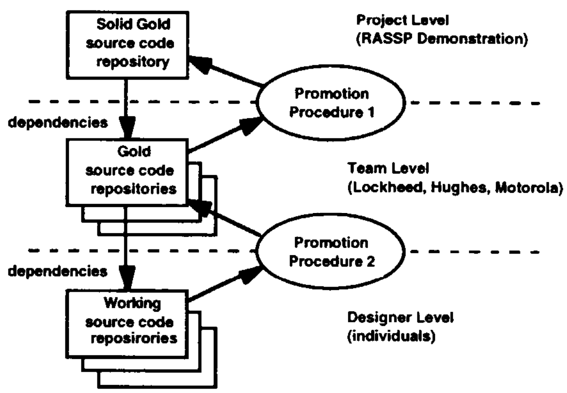


**Figure 2. Two Level Data Promotion Scheme**

Design partitioning of the Demonstration project was an important step. The obvious tact was to partition the design tasks among the three companies to minimize the amount of shared data and technical interchange. This allowed each company to progress as far as possible as well as simplify system integration. The design was partitioned at the circuit card assembly boundaries with all component design dependencies held within the respective company.

The techniques used for technical interchange relied heavily on email. It was used primarily for passing general information and documentation when volatility of data was not critical and time zones were inhibiting. Tele-conferencing was used on a daily basis for status and some code debug during integration time. Video conferencing was used at review points when the whole team needed exposure to presentation material that was to difficult to convey with the other means. Periodic co-location of the designers was still necessary to convey concepts in the early design phases and during debug in the later integration phases.

### 3. The Virtual Data Base

A difficult technical problem for the Demonstration project was the implementation of a design data base. A true virtual data base, where all design data is interoperable across heterogeneous design environments, is still a concept. Likewise, current network capacity is incapable of transferring the enormous amount of design data needed to support transparent access and communication to designers. Additionally, there are security constraints that limit the type of information exchanged in a virtual company environment. To overcome these obstacles, an intermediate implementation of a virtual data base was based on the following:

- VHDL used as the interoperability standard for design data at the source code level.
- Mirror VHDL source code repositories across all companies.
- Source code coherency mechanism based on an INTERNET communications architecture.
- RASSP Design Environment at all team sites including common VHDL design tools.

This approach relies heavily on the interoperability of VHDL and the source code coherency mechanism but is conceptually accurate (Figure 3).

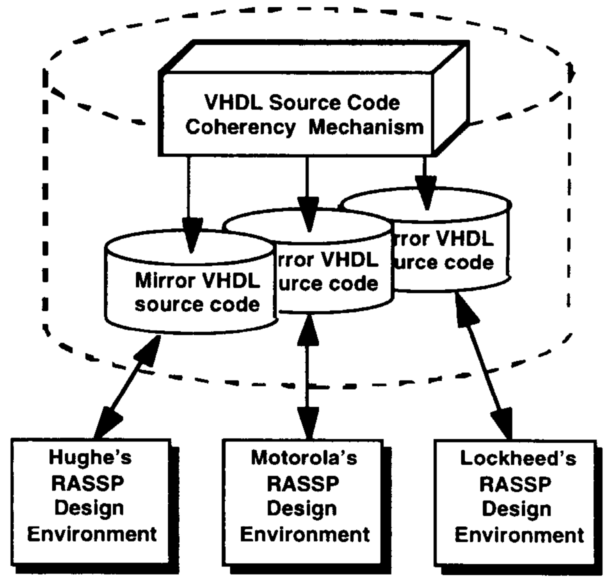


**Figure 3. Approach to Design Data Base**

The data base was implemented as a mirror structure of source code repositories. A communications structure was set up to allow the exchange of source code between companies within the constraints of data security policies. This approach was based on INTERNET communication servers at each of the companies. These servers had security firewalls that restricted access to only the involved companies (Figure 4).

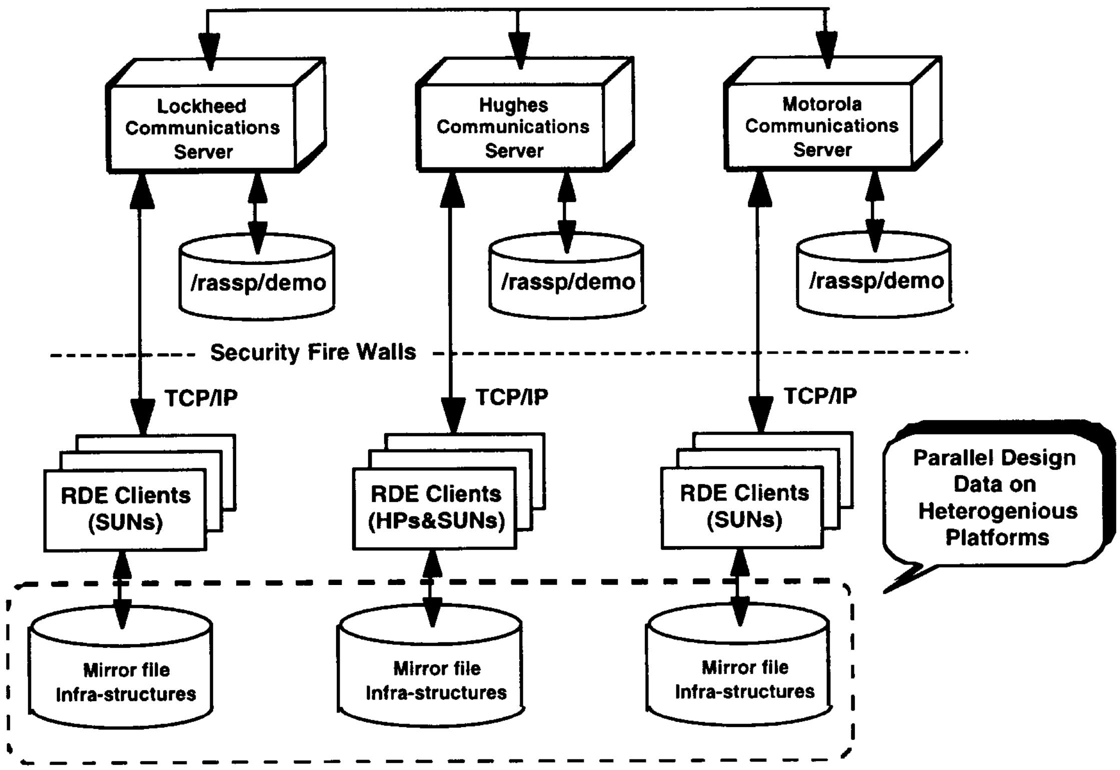


**Figure 4. Communications Architecture**

The basic mechanism to exchange data was by file transfer (FTP). The source code was moved from a company file system to and from the communications server file system. A move from one company to another required two transfers, an **FTP put** and an **FTP get**. The hosts at each company consisted of a mix of SUN Microsystem and Hewlett Packard workstations with UNIX operating systems. The host file system structures for the project areas was mirrored to facilitate the data promotion procedures. The hosts ran segments of the RASSP Design Environments (RDE) applications, in particular, the VHDL design tools such as Vantage Spreadsheet and VHDL mode Emacs. Additionally, other utilities were used such as SCCS source code control system and GNU make files.

The most intricate aspect of this structure was the source code coherency mechanism. This Mechanism had to facilitate the data promotion scheme, maintain the coherency of the mirror source code repositories, and use the established communications architecture and protocols. The data promotion procedures included the appropriate file transfers through the communications servers to the mirror source code repositories. Each team handled their own Gold level promotions (procedure 2) individually but the Solid Gold source code, visible to all, was controlled by one administrator. The promotion to Solid Gold level (procedure 1) and the coherency of all Solid Gold repositories was the responsibility of one team, in this case Hughes. This can be seen in the Figure 5 showing the set of source code repositories and the transfer paths between them.

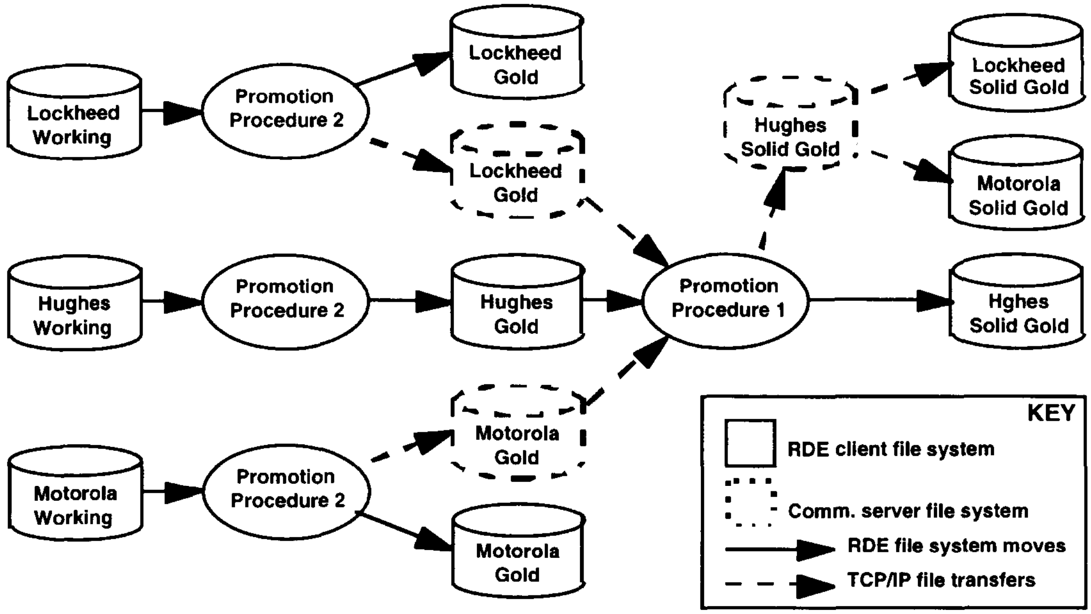


**Figure 5. Source Code Coherency Mechanism**

**4. Lessons Learned**

Upon completion of the core VHDL, work began on the development of test benches for validating the correctness of the core VHDL. One of the issues was the need for both self-contained and file-driven test benches. The resolution of the test benches expanded the methodology to incorporate non-VHDL mechanisms for validating the correctness of the core VHDL.

After the VHDL development was partitioned among the team members work began in developing the necessary VHDL. The process of developing VHDL uncovered missed areas in the description of the VHDL company interoperability simulation plan. One area of the development process that required more work was the build/make process. The differing files storage mechanisms along with the differing compute platforms required the development of a platform independent build/make methodology.

Email was used as one of the primary forms of communication between team members.

Email allowed for:

- Setting up conference calls
- Alert team members that new design data is on the server
- Transfer data and documents via attachments
- Communicate design errors and changes
- Monthly reports

Some of the drawbacks of using email included:

- Communication is not instantaneous
- Incompatible email tools did not transfer data correctly
- Email not delivered due to resource problems

Tele-conferencing and video-conferencing was used extensively to provide the real time feedback needed during status meetings. These meetings were used for status and to re-align critical aspects of the program. Teleconferencing was also used, in conjunction with Timbukto (a presentation manager across multiple companies using the INTERNET), to hold design reviews. This saved primary team members the trouble of traveling cross country to participate in major program reviews and milestones.

Distributed data bases can cause many problems especially when the size of these data bases are on the order of magnitude of tens to hundreds of megabytes. To transfer these large data bases between teams, the files were first combined into one file via the UNIX tar command and then compressed and encrypted using a special password before being placed on the server via FTP protocol. Other team members would then FTP the files to their site, unencrypt, , uncompress, and untar the file to recreate the file structure.

Once the VHDL code was transferred to the other team members as described above, the next step was to compile the code into the respective libraries. Ideally procedures should be in place for the compilation of the code to occur with the least amount of knowledge of the hierarchy of the models. The designer would only have to compile and use the model with some degree of confidence without having to understand to the smallest details how the model works. Trade offs were performed to assess the benefits of using makefiles or script files to create and compile code. Unlike makefiles, script files do not understand dependencies and therefore can not optimize compile times. This means that all files are compiled whether they are out of date or not. If the libraries do not exist and are being created for the first time, then this does not pose a problem. But if there are incremental updates to code then unnecessary compilation can occur. In spite of this, script files were first used to create the initial compile and it was up to the individual team members to manage incremental changes as needed. Regardless of which compile technique was used there had to be enough flexibility to accommodate any VHDL simulators, therefore variables were used to pass in compiler commands and switches.

## 5. Summary

Although there are certain aspects of these procedures that still need refinement, it should not go unnoticed that successful integration of the virtual prototype and successful build of the real hardware occurred with minimum travel. In fact some team members have met each other for the first time at the completion of the project and during the customer review, yet have had the closeness and rapport of a integrated design team in the same location. The only real obstacle was the time zone difference between the east and west coast. This was nothing that a home phone number couldn't fix.

## 6. Acknowledgement

The Lockheed RASSP team is under contract to the Naval Research Laboratory, 4555 Overlook Ave., SW, Washington, DC 20375-5326. The Sponsoring Agency is: Advanced Research Projects Agency, Electronic System Technology Office, 3701 North Fairfax Drive, Arlington, VA 22203-1714. The Lockheed RASSP team consists of Lockheed Sanders, Inc., Motorola, Hughes, and ISX.

## References

[1] Lockheed RASSP Team, "RASSP Process Document", AVY-L-S-00078-101-B.

[2] Jim Summers, Motorola, "Rapid Prototyping and the RASSP Design Environment (RDE).