CROSS-CONTRACTOR, CROSS DISCIPLINE SOFTWARE INTEGRATION AND PRODUCT DEVELOPMENT

Camille Brinkman
Science Applications International Corporation
Orlando, Florida

INTRODUCTION

The Joint Simulation Systems (JSIMS) technical vision is a single, distributed, seamlessly integrated simulation environment. JSIMS is a simulation system that supports the twenty-first century warfighter's preparation for real-world contingencies. The joint environment comprises seven product teams known as Development Agents (DAs): USMC, Maritime, National Air & Space (Warfare) Model (NASM), Warfighter's Simulation (WARSIM), National Simulation (NATSIM), WARSIM Intelligence Module (WIM) and JSIMS Integration & Development (I&D) team.

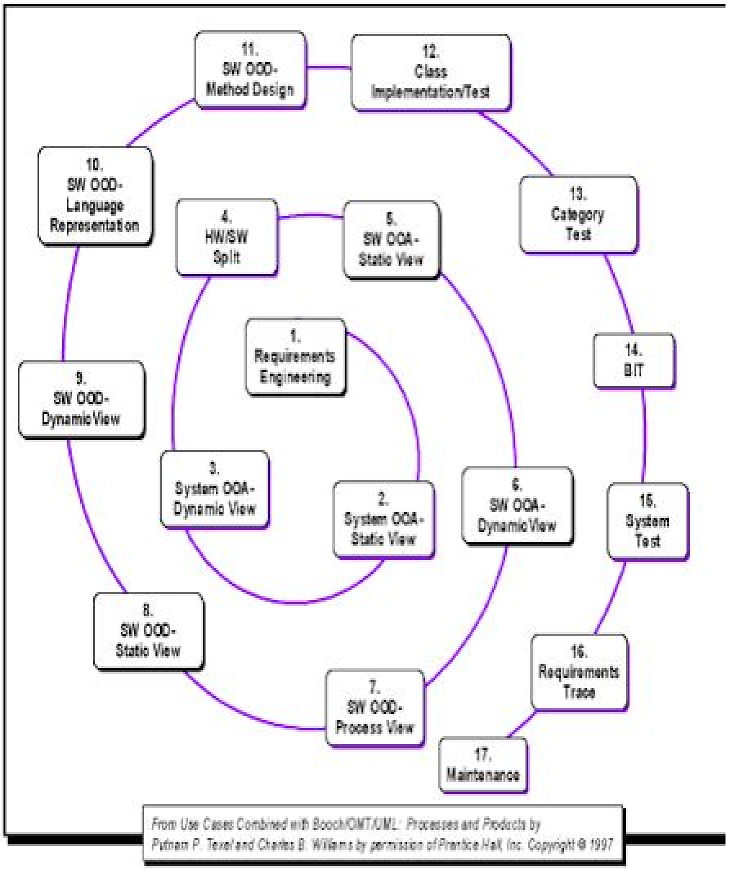
Since JSIMS encompasses several military organizations, this paper is constrained to discussing the Proof of Concept software build, Build 0, for the Mission Space Objects (MSO), an application's piece of the JSIMS architecture. MSO comprises the seven product teams. The focus of this paper is on the development process, products, communication techniques and lessons learned that were useful for succeeding in a diverse development environment.

The Build 0 Proof of Concept for MSO is analogous to building an airplane with seven companies, scattered geographically, each building a different part of the plane and only the parts where the plane fits together is reviewed periodically using paper diagrams and text. It was the MSO I&D Build 0 Manager's responsibility to ensure the product completes successfully and on schedule.

DEVELOPMENT PROCESS

The MSO Integrated Product Team (IPT) used Build 0 development to test and tailor the Texel-Williams Process as documented in the ***Use Cases Combined with Booch/OMT/UML: Process and Products*** by Putnam P. Texel and Charles B. Williams producing the JSIMS Object Oriented Process (JOOP).

Figure 1 depicts the JOOP. Phases 1 through 4 are accomplished by the System Engineers; Phases 5 through 13 are accomplished by the development IPTs; and Phases 14 and 15 are accomplished by the System Integration and Test Engineers. Table 1 outlines the JOOP.



From Use Cases Combined with Booch/OMT/UML: Processes and Products by Putnam P. Texel and Charles B. Williams by permission of Prentice Hall, Inc. Copyright © 1997

Figure 1. JSIMS Object Oriented Process

Each product team either used their development
process or implemented the JOOP for Build 0. Two developmental languages were used, Ada 95 and C++. NASM, JSIMS I&D, Maritime and USMC developed the models using C++ while WIM, NATSIM and WARSIM developed their models using Ada 95. The models communicated using the Frameworks layer developed by the Core Infrastructure IPT.

Table 1 JOOP Outline Description

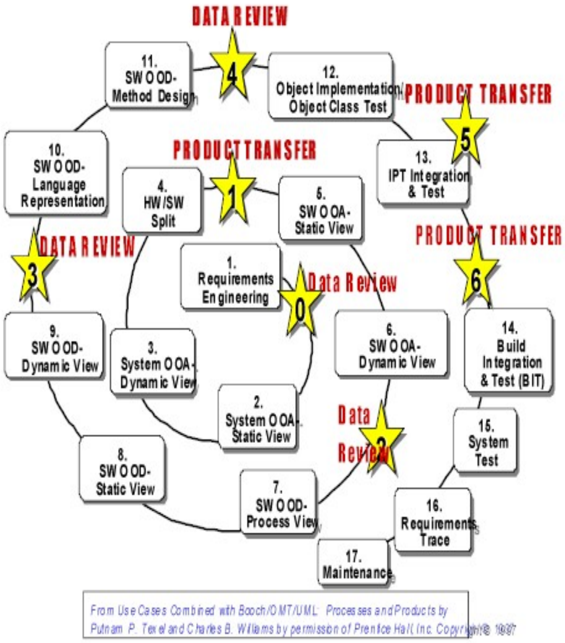
Phase	Title of Phase
1	Requirements Engineering
2 &3	System Object Oriented Analysis
4	Hardware/Software Allocation
5&6	Software Object Oriented Analysis
7,8&9	Software Object Oriented Design
10&11	Software Object Oriented Design (Language Input & Method Design)
12	Object Implementation/Object Class Test
13.1-13.4	Integrated Category & Object Class Test (Non Cross-Domain)
13.5	Cross-Domain Inter-Category Integration & Test
14	Integration & Test Across Integrated Product Teams
15	System Test (Functional Testing)
16	Validation & Verification Testing
17	Maintenance

For Build 0, the seven product teams had representatives on the MSO Build 0 Object Oriented Working Group (MSO B0OOWG) for development. The group pioneered the JOOP. For Build 0, the MSO B0OOWG developed the System OOA phases together (Phases 1-4). The working group used a modified Korean Thread Summary Scenario taken from a "focus context" provided by the Joint Conceptual Model of the Mission Space (JCMMS) IPT. The summary minimized the scope of MSO B0 development while still testing the process and creating models.

It is important to note that Phases 1-4 were developed in a collaborative environment with all DAs participating in the development of the products. A Use Case mapped one-to-one to a requirement. After the categories and Use Cases were allocated to DAs, the DAs developed their products in their environment and facility using their tools and processes and not necessarily following the JOOP. For Build 0, as mentioned earlier, some DAs followed the JOOP for learning and others did not. However, all followed the JOOP for Phases 1-4.

To promote consistency for reviews a constraining agreement was reached concerning the products. It was agreed that all DAs produced Category Interaction Diagrams (CID) and would be in a Rational Rose or a Power Point formatted diagram. In addition, all DAs produced scenarios formatted in MS Word or RTF. During reviews the group became familiar with other DA's products such as Sequence Diagrams or the Shlaer-Mellor process used by one DA using their Bridgepoint tool.

Because products were being developed independently, it was determined early that there should be points created during the development process, called Sync Points, to review or transfer the development products of all team partners. These products were those surrounding the interface areas of the models. Only the team partners who had a "stake" in the final product were required to review the products and all other partners were optional. These Sync Points are numbered 0 through 6. Figure 2 shows Sync Point occurrence in the life cycle.



From Use Cases Combined with Booch/OMT/UML: Processes and Products by Putnam P. Texel and Charles B. Williams by permission of Prentice Hall Inc. Copyright© 1997

Figure 2. Sync Points in JOOP

Based on differing time schedules, each DA scheduled a review of their products with the MSO B0OOWG independently. As a DA approached a Sync Point, a review would be scheduled. It was mandatory that all DAs who had an interface with the reviewing DA be in attendance and have previously reviewed the products. If a required DA was not able to attend or review the products, the review was rescheduled. Only the products that concerned the interface as defined by the JSIMS Federation Object Model (JFOM) was reviewed. Hence, not all the DAs' products were reviewed, only the interface areas.

The following summarizes the Sync Points:

- Sync Point 0 occurred after deriving a set of requirements from the Korean Thread Summary. All DAs participated.
- Sync Point 1 occurred as products were transferred from the System Analysis Engineers to the Software Engineers. For Build 0, all DAs participated in the analysis and product development. There was no formal exchange since the same representatives were working both system and software areas.
- Sync Point 2 was a review of the products that each DA developed. The DA made corrections to the products after the review, and if needed, another review was scheduled.
- Sync Point 3 was a review of the products that each DA developed. The DA made corrections to the products.
- Sync Point 4 occurred after Phase 11, following completion of design. This, again, was a product review and corrections were made accordingly.
- Sync Point 5 occurred after Phase 13.4, code and class/category tests being completed. Each DA was responsible for testing as much as they could within their facility (a whole or part of a category). ***Note: A DA may not have owned the entire category***. At this point a product transfer occurred called Hand Over. A package of the products with the description and Quality Assurance certification was transferred to the JSIMS I&D Configuration Management department. This included source code, requirements tracability, and the results of the testing.
- Sync Point 6 occurred after all inter-category integration and testing was completed. The finished product for MSO was transferred to the System Integration and Test Engineers.

The Hand Over Process as mentioned at Sync Point 5, occurred at the end of Phase 13.4 before beginning Phase 13.5. The JSIMS I&D MSO IPT had the responsibility of integrating all the DAs' models into a working system environment. A web page at the JSIMS Web Site contained a Hand Over form that was created to allow the DAs to easily submit the list of products. An entry on the form identified a File Transfer Protocol (FTP) location of the products. The Configuration Manager imported the products into JSIMS I&D environment.

Since Build 0 was a test of the process of the JOOP, the following products were either produced (yielding more clarification and detail) or reviewed for changes to the current guidelines/process:

- Tech Note 9, ***Process Description For Software Development Coordination Points***
- Tech Note 10, ***JOOP Phase 2-4 and 13.5 Participation***
- JSIMS Configuration Management Plan
- JSIMS Quality Management Plan
- JSIMS Software Development Plan
- A more detailed description of the Hand Over Products and Process
- Improved Hand Over Form

Note: A Tech Note is an agreement of process, products or schedule across the seven partners.

PRODUCTS DEVELOPED

Again, early into development it was decided to create a JFOM to define the interfaces and data types. As part of the JFOM, Phase 3 through Phase 11, an Object Class Structure, Object Interaction Table and Attribute-Parameter Table were developed collaboratively. At Phase 11, the JFOM was baselined, meaning that if a change was required a System Problem Report (SPR) was to be submitted to the JSIMS Configuration Management (CM).

The B0OOWG, during Phase 1, collaboratively developed a requirements trace matrix (RTM), a JSIMS Context Diagram, a JSIMS B0 Context Diagram, a MSO Context Diagram and a MSO B0 Context Diagram. Phase 2 products and activities included identification of Use Cases associated with requirements, development of scenarios and categories, allocation of the categories and methods to Use Cases, development of a System Category Diagram (SCD), and development of the Naming Conventions to be used. Phase 3 activities included assignment of Category Manager and Category Leads, development of the Category Interaction Diagrams (CID) and the beginnings of the JFOM. Phase 4 for Build 0 was eliminated. At this point an unofficial (since the same people were both the System and

Software Engineers) Sync Point # 1 Product Transfer was created and implemented.

Phases 5 & 6 began the Software Analysis. The products and activities that were developed during these phases consisted of the Category Class Diagrams (CCD) and Class Specifications (CS), refinement of the inheritance/aggregation, decomposed scenarios, added attributes and methods, refinement of the CIDs, and the development (as needed) of State Transition Diagrams (STD). All the products developed by the DAs were reviewed at Sync Point # 2.

Phases 7, 8 & 9 entailed Software Design. The products and activities included refinement of the CCD, identified data structures and abstract classes, updated CSs, CIDs, and the JFOM. Also, in these phases the interface with the CI IPT was made known. Sync Point #3 occurred with the participation of all DAs and the CI IPT.

Phase 10 is defined as the representation of the Object Oriented Design (OOD) language independent design. For Build 0, the only controller class is the Application Manager developed and owned by CI IPT. This phase further defined the Naming Conventions.

Phase 11 entailed the Method Design where a loosely defined Program Design Language (PDL) was implemented. This brings us to Sync Point # 4 where the source files containing the PDL was reviewed by all DAs who have an interest in the interfaces. The JFOM was reviewed and baselined.

During Phase 12 through 13.4, the DAs produced code and performed class and category testing within their facilities. Upon completion, Hand Over of the models to the JSIMS I&D for further integration and testing took place.

The Hand Over Package at the end of Phase 13.4 included the following list of products:

- Source Code
- Test scripts
- Test results
- Quality Assurance Certification
- Requirements traceability

Again, since Build 0 was a test of the products of the JOOP, the following products were either produced (yielding more clarification and detail) or reviewed for changes to the current guidelines/process:

- Tech Note 4, ***JSIMS Language Standards for Software and Object Models***
- Tech Note 5, ***JSIMS Single System Definition***
- Tech Note 7, ***JSIMS Integrated Models***
- JSIMS Software Development Plan
- MSO Tech Paper: ***JSIMS Class and File Naming Conventions*** by Patrick Merlot

COMMUNICATION TECHNIQUES

At the heart of every successful project beats flowing open communication. Given that, communication on a small co-located project is a real challenge, how much greater that of a large, geographically scattered project. The following paragraphs describe some of the communication techniques that were used to develop Build 0 involving seven DAs scattered from coast to coast of the United States.

In the beginning, the group met on an average of three times a week for the first two months and then twice a week through Phase 13.4. Local participants came to the group meeting while others used the telecon system. The reviews were generally a combination of both. To accommodate the West Coast DAs, reviews and meetings were scheduled in the afternoons (Eastern Time) as much as possible. Agendas and other information were distributed via e-mail or where there was no e-mail, by fax and the information was re-iterated. Minutes and review products were posted to the World Wide Web or FTP site. MSO IPT had large meetings every four to six weeks at a DA site to discuss all issues and concerns (not just Build 0 or the working group) and to disseminate information.

The working group reviews proceeded successfully for the most part. Since MSO models (software) communicated with each other by the Frameworks developed by the CI IPT, an Alpha Drop by CI was developed and made available for all DAs. Since CI and MSO were on the same schedule path, the development was concurrent which created an unstable development atmosphere.

Another challenge was that most of the DAs did not have the Rational Development Environment in their facility. A requirement was that the DAs' software (both Ada 95 and C++) had to compile in the JSIMS I&D Rational Environment. Telnet capabilities were established, however, this brought on a new set of challenges. For example, one of our West Coast DAs had a time crunch and was going to Telnet into the JSIMS I&D environment during a weekend. This was acceptable, however, a major router between the West Coast and the East Coast was non-functional which put the West Coast DA behind schedule. The DA solved the problem by flying a representative to the JSIMS I&D facility to work on their software. Another challenge was DA Internet Security FireWalls preventing communications with the JSIMS I&D environment.

INTEGRATION PROCESS & EXECUTION

Phase 13.5 is not part of the Texel-Williams Process. It was tailored as a response to the following specific circumstance. A DA owned, developed and tested software classes and possibly full categories. There were cases where a Category was made up of many classes that were owned by several DAs. A DA could only test, in their facility, what they owned and have developed. It was the responsibility of the JSIMS I&D to integrate and test the classes/categories that cross multiple DAs. Hence, Phase 13.5 was created as part of the JOOP.

Build 0 had a hierarchy of Use Cases. There were nine total Use Cases. Use Cases 3 through 9 were independent of each other in the respect that they were on the same hierarchical level. Use Case 2 called, if you will, Use Cases 3 through 9. Use Case 1 called Use Case 2. Therefore, from a high level of integration, Use Cases 3 through 9 were tested independently. Followed by Use Case 2 and Use Case 1 respectively.

Software for each Use Case was spread among several DAs. No Use Case was owned, developed or tested within a single DA. [Hindsight shows this was probably not the best approach]. This meant that in order to test, say, Use Case 3, (which was spread across three DAs), all three DAs' software had to be integrated. Only the categories (software) which made up the Use Case 3 were combined to test only Use Case 3. This made the integration environment complex. Let me explain.

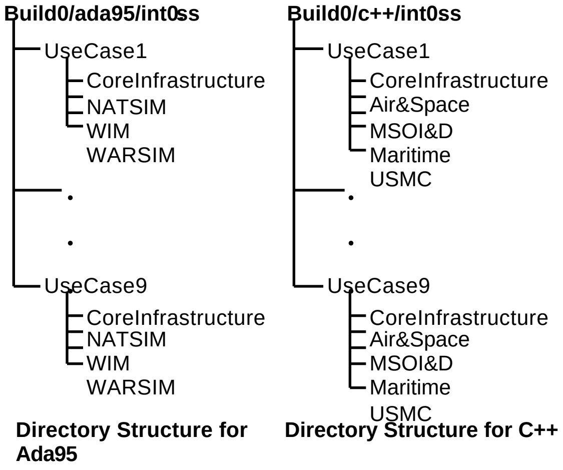


Figure 3. Build 0 Integration Directory Structure

It had to be determined for each Use Case, for each DA, what files needed to be brought into the environment. The directory structure for integration was set up with two sides of the APEX (a product of Rational, Inc.), one each for the Ada95 and C++. See Figure 3.

Then the files required for each Use Case, for each DA, was copied from the Configuration Management area of the environment. Use Case 3 was the first to be integrated. Each DA that had software residing under Use Case 3 was compiled. If there were problems in compilation, the DA had a software representative in the JSIMS I&D lab for corrections.

This brings up two facts needed for clarity. The first is that after the environment was established, each DA had a Software Engineer who was familiar with the software in the JSIMS I&D lab. Each DA was assigned a familiar Use Case to integrate and test. The MSO I&D Build 0 Manager's responsibility was to plan, orchestrate and execute Phase 13.5. The second is that if a correction of the software was required then an SPR was written and assigned. Anyone could write an SPR, but only DAs could make changes to their own software. All DAs made changes to their software within the JSIMS I&D lab.

Because the testing of a Use Case was heavily dependent upon several DAs, if there was a problem with one DA's software, the testing of that Use Case had to stop until it was corrected. Therefore, quick on-site resolution and implementation of SPRs was essential for completion of the Phase in a timely fashion.

The manner in which an SPR was dispositioned is outlined below:

- Assigned to a DA
- Fix the problem described in the SPR
- Test the fix
- Once a week, all DAs and their interested management met to review the SPRs and get agreement from each DA on its closure.

The SPR system was implemented using ClearDDTS and was located on the web with user and password.

A test driver developed by the Air & Space DA was used to test the Use Cases. Since the simulation was event driven, the test driver simulated events that were required to execute the scenario but was not contained within the Use Case. As each Use Case was tested successfully, each participating DA signed a completion form.

MSO Build 0 finished product was 19K lines of software and was integrated in less than two weeks. The models were not built to conceptual models, meaning that correct military behavior of a model was not a part of Build 0. The goal was to determine the effort and prove that software development could successfully work in a cross-contractor, cross disciple environment.

SUMMARY

There were a number of lessons learned during Build 0. As mentioned earlier in the Development Process and Products Developed sections several Tech Notes, White Papers and Documents were established or updated in response to Build 0 activities.

Another lesson that the B0OOWG learned was that even though information was distributed via e-mail, telecon and reviews it should be re-iterated several times. Also, the working group realized that there were a significant number of issues/concerns that the group was not empowered to solve and that these issues/concerns had to be passed up the management chain.

One of the most significant lessons is that with a project of this magnitude, it is best to determine areas of development where there could be commonality and mandate that commonality. For instance, development would have been easier provided the processes, development tools, development products, programming language and schedule were common across all seven DAs. This would have reduced the effort significantly.