LESSONS LEARNED IN DEVELOPING MULTIUSE SIMULATION FOR F-22

Dorothy M. Baldwin, James H. Gault, and Stephen S. Zimmel
Lockheed Fort Worth Company (LFWC)
Fort Worth, Texas

ABSTRACT

Multiuse Simulations are even more critical in light of current budget constraints. Early planning during F-22 development has provided a unique opportunity to maximize simulation synergism across an entire Weapon System. Via Integrated Product Teams (IPTs), the Air Vehicle, the Support System, and the Training System are being developed concurrently. Potential simulations for REUSE by the Training System were identified early to be able to incorporate training requirements into Air Vehicle and Engineering lab developments.

This paper describes "Lessons Learned" in developing simulations to satisfy multiple engineering laboratory and training requirements and also provides examples of specific cases where Training System personnel have acted as "integrators" between various Air Vehicle IPTs.

A good example is the development of the Flight Dynamics Simulation (FDS). FDS has completed Preliminary Design Review (PDR), Critical Design Review (CDR), coding, integration and testing, and will be operational in the *Vehicle Management System (VMS) Integration Facility (a full-up pilot-in-the-loop engineering flight simulator)* by the time this paper is presented. All potential users, including training system personnel, were involved in requirements, review, and approval cycles. All identified training requirements have been met. Examples are given of how FDS development "Lessons Learned" have been shared with other REUSE engineering simulation developers.

Challenges that lie ahead and the processes being put in place include (1) how to develop a robust, flexible design based on early requirements that we know will change, i.e., how to incorporate REUSE simulations into the final media that result from Instructional System Development (ISD) and provide these REUSE simulations to the ultimate training simulator designer and integrator and (2) how to update the REUSE simulations during the Weapon System life-cycle while satisfying the requirements of diverse users.

ABOUT THE AUTHORS

Dorothy M. Baldwin is an Engineering Project Manager at Lockheed Fort Worth Company, currently assigned as the Integrated Product Team (IPT) Manager for the Lockheed portion of the Pilot Training System Devices for the F-22 aircraft. During her ten years at Lockheed/General Dynamics, Ms. Baldwin has led efforts related to foreign customer initiatives in both training and engineering laboratory development; has provided technical and management leadership for a multimillion dollar engineering flight simulator laboratory development; and has led a tank simulation development in support of General Dynamics Land Systems Division. Prior to coming to General Dynamics, Ms. Baldwin spent seven years as a physicist working for the Naval Training Equipment Center (where she led multiagency and industry efforts in sensor and visual simulation technology development) and spent the previous twelve years teaching physics and developing related courseware and curriculum at a number of universities, colleges, and high schools. Ms. Baldwin has a BA from Hartwick College and an MA in physics from Kent State University and is a certified physics teacher in two states.

James H. Gault is an Engineering Specialist Senior at Lockheed Fort Worth Company assigned as Software Product Manager for the F-22 Training Systems group. His current responsibility involves leading an effort to develop a common F-22 flight dynamics simulation for reuse in several engineering laboratories and Pilot Training System devices. Mr. Gault's experience, prior to the F-22 Program, included four years developing engineering and simulation software for AFTI and other F-16 derivatives and over ten years developing real-time software for F-16 avionics test equipment at LFWC. Mr. Gault holds a BS in both chemistry and mathematics from the University of Texas, Arlington, and the University of Houston, respectively.

Stephen S. Zimmel is an Engineering Specialist at Lockheed Fort Worth Company's flight simulation laboratory, where he has been employed as a hardware and systems design engineer since 1983. He is currently responsible for the development of the LFWC F-22 flight simulators being used in the analysis and testing of the F-22 flight controls and vehicle management systems. Mr. Zimmel holds a BS degree in electrical engineering from Pennsylvania State University.

LESSONS LEARNED IN DEVELOPING MULTIUSE SIMULATION FOR F-22

Dorothy M. Baldwin, James G. Gault, and Stephen S. Zimmel
Lockheed Fort Worth Company (LFWC)
Fort Worth, Texas

BACKGROUND

Multiuser (or) REUSE Simulations are becoming even more critical in light of current budget constraints. Early planning during the F-22 Program has provided a unique opportunity to maximize simulation synergism across an entire Weapon System. Via Integrated Product Teams (IPTs) during the Engineering and Manufacturing Development (E&MD) phase[1], the Air Vehicle (A/V), the Support System (SS), and the Training System (TS) are being developed concurrently.

REUSE is an integral part of the F-22 program and F-22 contract; in fact, it is part of the F-22 contractual award fee criteria. References to REUSE appear in a number of F-22 documents, including the Integrated Master Plan and the Integrated Master Schedule (Reference 1), and both the Weapon System (Reference 2), and the Training System Specifications (Reference 3). The F-22 Weapon System Software Development Plan (WS SDP) (Reference 4) and the Training System Software Development Plan (TS SDP) (Reference 5) define the REUSE process.

Air Vehicle IPTs are developing portions of the air vehicle simulation for incorporation into the Pilot Training System devices. Potential reused components for the training system were defined and worked early in the E&MD program to enable their inclusion in air vehicle and engineering lab developments. In parallel with efforts to maximize REUSE, a rigorous Instructional Systems Development (ISD) process is being conducted to define the total training system

The REUSE effort described herein is actually one example of a much larger movement in the industry and at LFWC to improve software development processes. Lockheed is a member of the Software Productivity Consortium formed by aerospace companies for this purpose in 1985. One product of the consortium, Ada-Based Design Approach for Real-Time Systems (ADARTS), was adopted for the F-22 program. In addition, LFWC was recently evaluated by the Software Engineering Institute (SEI) via their Capability Maturity Model as a Level III. Level III certification means, "The process for engineering and management is documented, standardized, and integrated into an organization-wide software process." The Flight Dynamics Simulation (FDS), described herein, was one of the modules tracked for compliance. The Defense Department expects these new processes to save billions of dollars (Reference 6).

A follow-on to a 1991 paper by Baldwin and Landry (Reference 7), this paper describes lessons learned to date in the development of unique processes to allow simulations to satisfy multiple Engineering Laboratory Development and Pilot Training System device requirements. This paper concentrates on the successes encountered and challenges overcome (and yet to be overcome) for LFWC Pilot Training System Devices (PTSD) Software REUSE items. Though REUSE is being addressed in other areas of the F-22 Weapon System, this paper is limited to the areas cited above.

INTRODUCTION

REUSE goals include improved quality, supportability, potential cost reduction or cost avoidance, and potential schedule savings or schedule delay avoidance. The TS SDP (Reference 5) addresses all aspects of Training System Software development through the Weapon System life cycle. The Training System will do a Make vs. Buy for entire systems and applicable subsystems. Any potential pilot simulators, identified by ISD, could be bought from vendors. Applicable REUSE items (that survive the Make vs. Buy process) could then be provided as contractor-furnished equipment (CFE). An F-22 "Joint Procedure" (Reference 8) provides guidelines for implementing REUSE across the entire team. Three types of REUSE are identified in Reference 8: Planned REUSE, Opportunistic REUSE, and Anticipated REUSE.

Planned REUSE – IPTs identify common assets within or across IPT boundaries and enter into a partnership to consolidate commonality to the extent that one IPT becomes a developer and one or more IPTs become reusers.

1 Starting in 1991, E&MD was lead by Lockheed Aeronautical Systems Company (LASC) teamed with Boeing and Lockheed Fort Worth Company (formerly General Dynamics, Fort Worth Division). Boeing is team lead for the Training System.

Opportunistic REUSE – IPTs reuse existing assets and modify them to fit their application.

Anticipated REUSE – The principle of engineering all assets with reuse characteristics to enhance reusability on future programs.

This paper and the previous paper (Reference 1) deal primarily with examples of planned REUSE. REUSE includes design, code, documents, test, data, tools, etc.

DEVELOPING REUSE SIMULATIONS USING THE INTEGRATED PRODUCT TEAM (IPT) CONCEPT

One IPT is assigned responsibility to meet REUSE requirements of several IPTs, with potential reusing IPTs invited to participate in all phases of development and requirements definition through reviews, software product evaluations (SPEs), and testing. Plans are established for including all potential reusers in the software change process through the life cycle. Figure 1 is a summary of planned reuse for the LFWC PTSD responsible areas, including all potential reusers.

	User/Laboratory							
Simulation	VIF	VSS	SDL (1)	SDL (2)	AIL	SIL	FMS	PTSD**
FDS*	✔	✔			✔	✔	✔	✔
CNI			✔		✔	✔		✔
EW			✔		✔	✔		✔
IRS	✔	✔	✔	✔	✔	✔		✔
VMS/U&S			✔	✔				
SMS			✔	✔				

*Development used as case study in this paper.
**Final REUSE items subject to results of ISD and trade studies.

AT02149

Figure 1 F-22 Simulation Planned REUSE

Preliminary PTSD requirements were identified in a timely manner through the use of "quick looks," which consisted of preliminary ISD analysis of critical REUSE areas. Contractual tasks were created to identify training requirements for engineering lab simulations. The PTSD IPT leader had approval rights of the software documents.

We have a program target that PTSD requirements cannot impact engineering simulations more than 20 percent. Technical Performance Metrics (TPM) are used to measure how well we are doing with respect to our target. This TPM (Figure 2) is used to measure the simulation software Source Lines of Code (SLOC) for LFWC-responsible simulations required to support PTSD unique requirements.

The target value and the current estimate are recalculated at each program event along the activity bar. Program events numbered along the activity bar are major reviews for the LFWC PTSD REUSE simulations. Figure 2 shows performance better than the target and, therefore, in the favorable area.

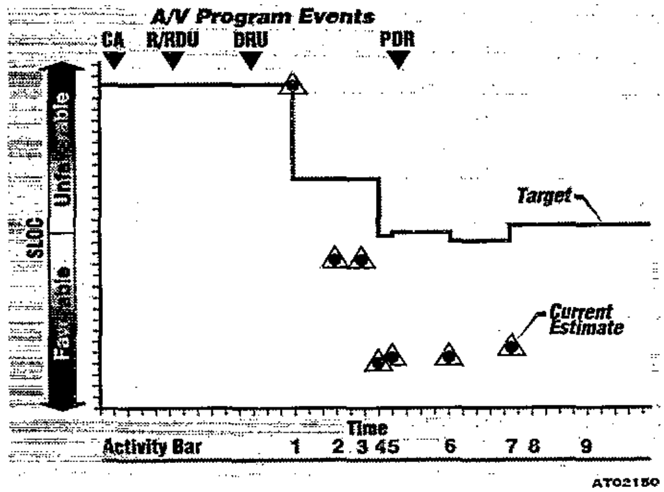

Figure 2 LFWC Simulation Software SLOC Allocated to PTSD Unique Requirements

MULTIUSE SIMULATION DEVELOPMENT – A CASE STUDY

The Role of the Developing Laboratory – The Flight Dynamics Simulation (FDS) will be used to illustrate the progress made on the MULTIUSE (or REUSE) simulation development.

The Flight Dynamics Simulation (FDS) Computer System Configuration Item (CSCI), intended to meet high-fidelity F-22 Airframe Simulation requirements of engineering labs and training systems, was developed as a crossflow item to support the six users shown in the shaded portion of Figure 1. The simulation was originally developed for the VMS Integration Facility (VIF). Because of PTSD deliverability, FDS was required to meet F-22 deliverable standards as specified in the WS and TS SDPs, i.e., to be DoD-STD-2167A, -2168, and MIL STD 1803 compliant and to be written in Ada. The FDS was co-developed by Flight Simulation Laboratory (FSL) and PTSD personnel. The FDS CSCI has been designed with the knowledge that extensive REUSE will occur. All designs, code, documentation, and test procedures will be available for REUSE by the Engineering labs and PTS.

To put the FDS into context, a brief overview of the VIF is provided. The architecture diagram for the engineering development laboratory (Figure 3) illustrates the relationships between the various hardware and software elements that comprise the system.

The software components are shown as circles with the hardware components and their interfaces represented by rectangles.

At the core of the software products are the Flight Dynamics and Vehicle Management System Sensor simulations, that were designed to crossflow, or for

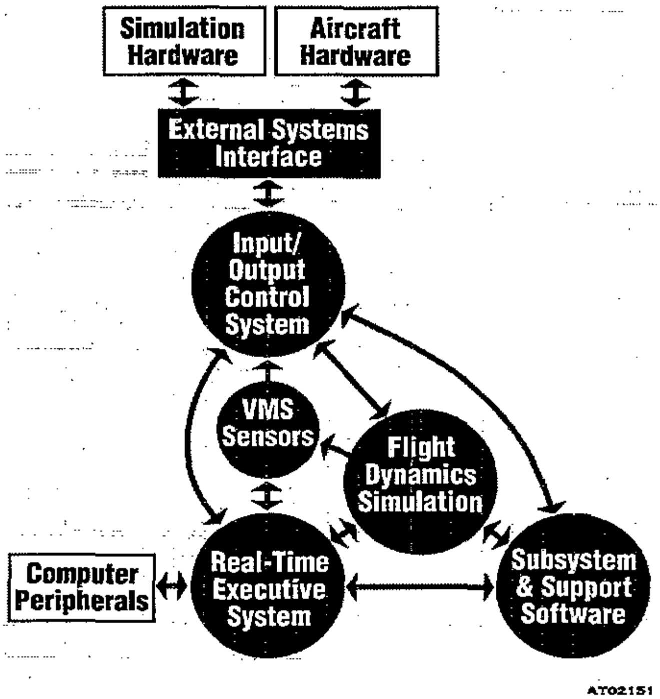

Figure 3 Engineering Development Laboratory System

REUSE, into the Training Simulators. The FDS is the model of the physical motion of the airframe, excluding (to the greatest extent possible) the internal aircraft systems. The VMS Sensor simulation models the behavior of the F-22 Vehicle Management System's unique set of gyros and accelerometers.

The remaining software includes the real-time executive that provides the user interface and overall simulation control function; the input and output control software (which scales, gathers, and scatters data between the simulations and the external hardware systems); and the subsystems and support software (which provides simple models of the other aircraft systems necessary to satisfy interfaces not provided by actual aircraft hardware). A potential layout for the PTSD simulator is described in Reference 7.

Reuse Strategies for FDS – Strategies used to meet requirements of several IPTs include:

- Inviting potential reusing labs and IPTs to participate in review and software product evaluation (SPEs). Plans were to include all potential reusers in the software change process.
- Partitioning the FDS to be as independent of a specific computer as possible through the use of structural modeling techniques.
- Avoiding constant revisions resulting from aircraft interface changes by modeling only physics – not avionics or aircraft subsystems.
- Structuring the software so that it requires only a tailored shell to handle system calls, i.e., no input or output software is required.
- Accommodating different update rates, i.e., programmable Δt.

Role xof Pilot Training Systems – PTSD involvement began in the concepts definition phase and was part of the E&MD proposal. Involvement continued during the requirements definition phase with the review and submission of requirements. During the preliminary and critical design phases , PTSD was a contributor to the Software Product Evaluation (SPE) process. SPEs are required by DoD-STD-2167A to be performed on deliverable products during the software development phases.

PTSD had an integral role during the coding and integration phases by having a number of PTSD software engineers included as part of the FDS development team. Although this was unique among the REUSERs, it was considered to be critically important because of the different philosophies governing the engineering and training simulations.

The role of PTSD during testing of a REUSE item is similar to that of the earlier phases; PTSD participated in the SPE of all FDS test procedures. PTSD IPT members are working within the PTSD IPT (which includes SPO) and with the A/V IPT to come up with tests that will satisfy trainer deliveries and engineering lab developments, through the life cycle. This is ultimately the only way to achieve true synergy between PTSD and the engineering lab developer, and the resulting cost savings from REUSE. The goal is that (throughout the life cycle) once a change is successfully tested in the engineering lab, a copy of the software can be shipped to the trainer – for immediate concurrency between the engineering lab simulator, the updated airplane, and the training simulator.

Phase-by-Phase Results Summary and Lessons Learned – A Case Study

Requirements Definition Phase – During the requirements definition phase, the Software Requirements Specification (SRS) and Interface Requirements Specification (IRS) were developed, SPEed, and released after the Software Specification Review (SSR). All requirements were traced to three areas (Figure 4), which presented a difficult challenge, especially when requirements were fluid and would most likely remain fluid for some time. This influenced our decision to go to the structural model approach described later.

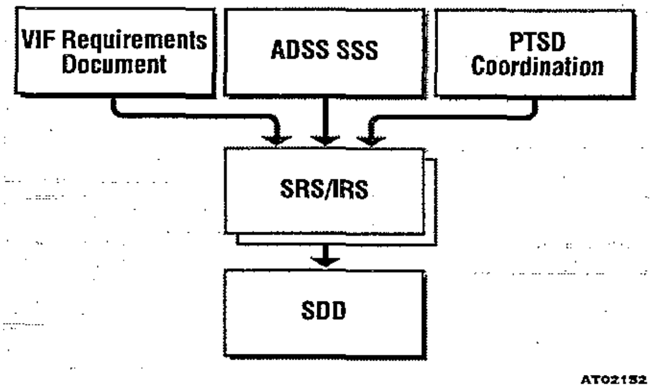

Figure 4 F-22 Simulation Requirements Traceablity

As was touched on earlier, the FDS was functionally partitioned for maximum reuse. FDS provides airframe dynamics, ground handling reactions, aerodynamics forces and moments, mass properties, and atmosphere and wind models. FDS does not provide control and monitoring features, propulsion, actuator, airframe sensors, and flight control models, or aircraft hardware interfaces.

A great deal of confusion resulted from our choice of software product names. Users tended to have entirely different functionality expectations based on their local cultures and previous experiences. For example, the term "flight dynamics" may mean an entire flying airframe – including aerodynamics, flight controls and a propulsion system – to some, but simply equations of motion to others. We would encourage REUSE suppliers to communicate the limitations of the REUSE software as well as the benefits so that potential users are not lulled into the false belief that a single CSCI contains more capabilities than the supplier intends to provide. This should reduce the number of change requests that arise as development proceeds through the design phases and should thereby reduce the likelihood of costly additional development. This clear definition of the boundaries of the REUSE item should also enhance the possibility of potential REUSE on future programs, i.e., "Anticipated REUSE."

A software architecture, consistent with SEI's air vehicle structural model (AVSM) (References 9-12) was developed to aid in understandability, maintainability, extendibility, ease of rehost, ease of adding and modifying malfunctions, and scalability. This architecture is an object-based design with constraints on program communication and coordination. A design decision was made to implement faults at the lowest logical level. Benefits of this structural model architecture include consistent interfaces, i.e., no surprises; proven concept because of common industry use; and easy accommodations for future growth.

In our efforts to "tie down" diverse requirements, we may have gotten carried away. Based on comments by various people at the first walk through of our requirements documents, we added great detail about the system, especially the interfaces, and crossed the fine line between requirements definition and design. Many of our internal interfaces were defined at the CSCI and this presented a problem in two areas, i.e., maintenance and testability. Many times we had to change the SRS and IRS, not due to requirements changes but because of design changes, such as interfaces, which were documented in the SRS or IRS. We also found that many of these "requirements" were not easy to test at the CSCI level because they were really internal. In the future, we should be more careful about what details are included in the requirements documents.

We should also have been more rigorous in establishing only firm requirements, despite our conviction that the software design be reasonably able to accommodate changes. This might seem to be an obvious recommendation, but it is clouded in this case by the wide temporal gap between the initial development and eventual release to the REUSER. For example, the engineering simulation development must necessarily lead the definition of training requirements – the Instructional System Development (ISD) process - by a considerable time. There seemed to be a tendency for the REUSE customers to get caught up in their involvement in the design process and push for the premature inclusion of requirements based on their perception of the future design. A barrier should be maintained between allowing reusers to guide the development toward reusability and allowing them to establish false requirements as firm ones. It is much cheaper in the long run to design a system which can easily accommodate firm future changes than it is to continually redesign the system based on a set of dynamic current requirements.

Since the idea of multiuse simulations is new to most software development teams, it is important at this early stage to define the general configuration management concept which will be used throughout the program life cycle (development, production, and support). The process should not be so burdensome as to preclude the efficient rapid prototyping activities that will be required for the engineering simulator's initial support of the weapon system's dynamic design and integration phases. Conversely, the process must maintain sufficient control of the changes to provide traceability and allow the preparation of detailed formal release documents required for the training system. The apparent disparity between the engineering simulator's requirement for rapid change response and the trainer's requirement for rigorous process control were a primary source of conflict in the FDS development. This became know as the "Rapid Prototype Development Problem" and is illustrated in Figure 5.

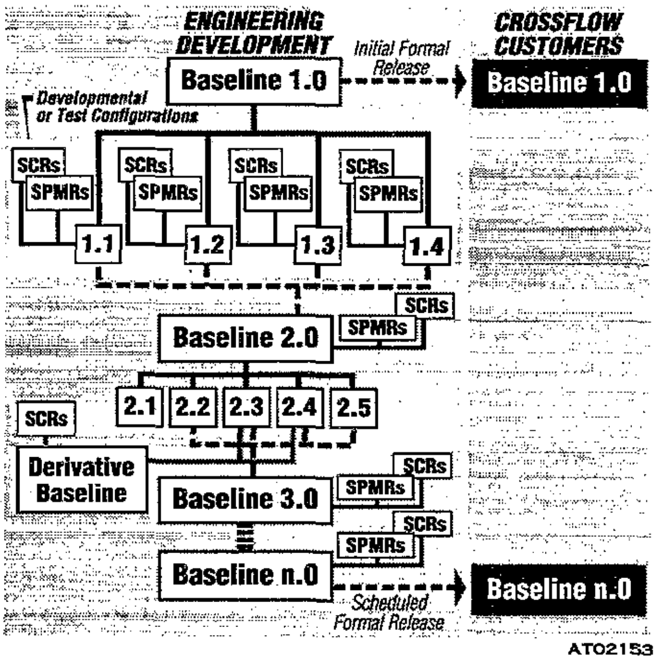

Figure 5 The Rapid Prototyping Development Problem

As is obvious from the illustration, it is frequently the case in an engineering simulator that several versions of the baseline software package may be in use and under configuration management simultaneously. [Note that in the LFWC flight simulation laboratory, two change mechanisms are currently in use. The Simulation Product Modification Request (SPMR) is provided to developers by the end user as an advance notice of changes to the fundamental simulation requirements and typically affects all test configurations simultaneously. The System Change Request (SCR) may be created by either the end user or the simulation developers to make a global or limited change but must be present before a change to the baseline can occur.] The need for simultaneous availability of different test configurations arises because different engineering users need to evaluate the system's behavior based on their unique change or set of changes to the baseline during the same release cycle, often during the same day. Figure 1 shows 1.1, 1.2, 1.3, and 1.4 as user-unique changes that may, or may not, be incorporated into the next formal baseline, 2.0. This is in diametric opposition to the training simulator's situation, where it is not only desirable but mandatory that each trainer of the same A/V configuration, operate with a common software version, regardless of the particular scenario under which the trainer may operate during a given training session. Test configurations that incorporate the individual changes are tracked throughout a given baseline release cycle but are archived at the time of a new baseline release. A new system baseline is defined and released at the discretion of the end user. Figure 1 shows baseline 1.0 and n.0 as formal releases. It is usually based on a related Weapon System milestone (e.g., PDR, CDR) or a major release of some component of the Operational Flight Program (OFP). This release may incorporate any or all of the changes implemented in the test configurations. The entire release-test-release process may be repeated several times prior to a formal release of the software to the REUSE customers. Figure 1 shows baseline 2.0, 3.0, ... n.0-1 with no formal release. With the large number of changes being made in rapid succession, it is obvious that early and thoughtful design of the change control process is very important. (Our method for addressing this problem for crossflow software is described in more detail later in this paper).

By seeing the overall concept defined early, the software developers can become comfortable with the process and gain ownership in it during its refinement in later phases.

Design Phase – Conceptually, the decision to save program resources by REUSE is very attractive, but, as is often the case in the real world, we discovered that the "devil is in the details." Risk Management items and plans were developed early and were constantly modified.

Satisfying requirements of multiple users was on the top of the identified risks list. This proved to be a correct assessment. Common links with the VIF executive software and other F-22 high-end operating systems (HOS) were identified as a REUSE risk. Our risk abatement approach was to minimize links and coordinate with HOS developers. Another risk item was the need to rehost FDS on different platforms. The obvious approach was to design FDS for least dependence upon a hardware configuration. This was accomplished by insulating the FDS from the host computer using the system's executive software and by establishing a common data interface for all applications that was consistent with the structural model.

One of our most significant and painful lessons occurred during the preliminary design phase. All REUSE software had originally been designated as CSCIs for the original developing teams, which, in all cases, were Air Vehicle IPTs. This resulted in the Air Vehicle team having to meet PDR type requirements at Air Vehicle PDR for simulations that did not need to be at that point in their development to meet any IPT's needs. To resolve this, REUSE simulations were designed as non-CSCIs (but developed to deliverable standards) to allow the flexibility to meet schedules and requirements of diverse users. This required that a unique CM process be developed, as described later. We did have sign off on all documents by affected IPT Managers through the development process. This promoted ownership and cooperation but required a willingness to compromise for the overall good of the program.

During the preliminary design phase, developers should be encouraged to resist any temptation to immediately isolate the various pieces of software as independent entities and dismiss external interfaces as unimportant. The external interface names should, if at all possible, be identified early and carried consistently throughout all development phases to ensure that costly discrepancies do not arise during integration.

Between PDR and CDR many changes occurred. Of seventy-eight SRS Requirements, twenty-two were modified, five new requirements were added, thirteen requirements were deleted, the other requirements were unchanged.

Implementation, Integration and Testing Phases – It was during this phase that the benefits of the structural modeling concept and the software architecture chosen during the design phase were demonstrated. Many changes to software were made in an attempt to meet the stringent timing requirements, e.g., the migration of the FDS from two to three and then four processors. The developers believe the chosen architecture supported the rapid restructure and reallocation of this software.

Extensive testing at unit and CSCI levels required considerable coordination with other IPTs for test data. Estimates of the time and effort necessary for this phase were too low because previous simulations had not been tested so thoroughly. This high level of confidence in the simulation was required because the FDS will be used to qualify a safety-of-flight OFP. The proper operation of the real-time simulation was verified by comparison of the output state variable vs. data from the airframe trim, linearization, and simulation (ATLAS) model. Time history comparisons were made by overplotting the data.

CSCI installation, maintenance, and control during FQT were the responsibility of the crossflow software Product Configuration Management System (PCMS) administrator. Tools were provided by F-22 System/ Software Engineering Environment (S/SEE). This worked well.

SHARING OF LESSONS LEARNED WITH OTHER F-22 AIR VEHICLE IPTS DEVELOPING MULTIUSE SIMULATIONS

The VMS IPT, which developed the FDS, has the earliest schedule on the F-22 program for development of its simulations. Many of the lessons learned from FDS development are very useful to other simulation developers on the program. PTSD IPT must cross a multitude of A/V IPT boundaries, thus placing PTSD IPT in a unique position to understand REUSE simulations being developed by various labs (including vendors). The PTSD IPT also has a vested interest in seeing that the REUSE simulations have architectural characteristics that would allow their use in some, yet to be finalized, pilot training device architecture. LFWC's PTSD took the initiative and brought together SEI and FDS developers with vendors who had the responsibility of developing Electronic Warfare Simulation System (EWSS) and later Communication, Navigation, and Identification Simulation System (CNISS). The purpose of these meetings (which took place over several months time for each REUSE item) was to offer the benefits of the experience gained by the FDS development. FDS developers had, under the guidance of SEI, reused a design concept developed under previous trainer programs, i.e., ASVP, C-17, B-2, SOF ATS, etc., (References 9 through 12).

The meetings were very successful, with both the EWSS team and the CNISS team adopting the SEI's structural modeling approach, because it made sense to them. Items shared with the EWSS and CNISS teams that were used for FDS include architecture, specification form templates, code templates, Software Design Document format and words, components, and code. CNISS and EWSS developers reused the specification form templates and the basic concepts. This is potentially a large cost avoidance for the program. Instead of paying for three completely separate developments, much was shared, i.e., the ultimate REUSE and WS IPT in action! The positive for PTS, is that these simulations are now much more attractive for REUSE in a PTSD trainer.

STATUS OF PROCESS DEVELOPMENT FOR ADDRESSING FUTURE CHALLENGES

Providing REUSE Items to Training Simulation Designer – We are faced with the challenge of how to develop a robust, flexible design based on early requirements that we know will change. A further challenge is how to provide resultant REUSE simulations to the ultimate training simulator designer and integrator. We determined that the approach most likely to succeed was structural modeling and PTSD had such a large stake in the outcome that we needed to be proactive.

Structural modeling advantages include ease in changing or adding features without creating a rippling affect. Structural modeling is more understandable, easier to maintain, its subsystems and components are reusable, it eases documentation, and it allows natural propagation of simulated malfunctions. Lessons learned to date, described earlier, support the decision to use the structural modeling approach.

Updating REUSE Simulations During Weapon System Life Cycle – This section only addresses the case study (FDS) described earlier. As stated above in the Rapid Prototyping Development Problem, the requirements of a change process for crossflow software presented many challenges :

- Quick turnaround to support engineering lab activities
- A more controlled release system to ensure commonality across all using communities
- Representation of all users and reusers needs.

A two-tiered change control process (Engineering release and formal release), as depicted in Figure 6, has been set up to address all the above issues.

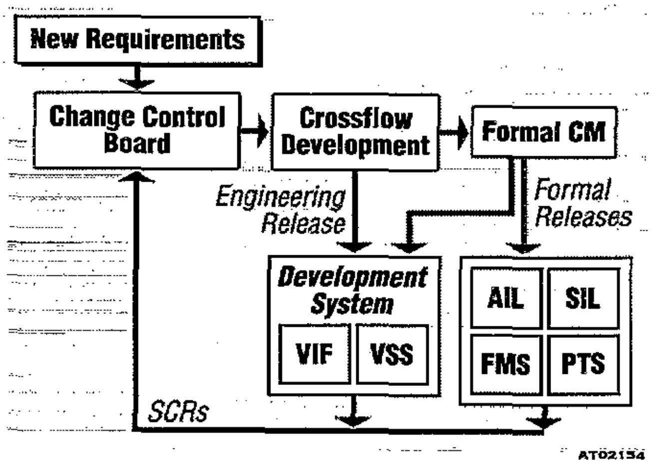

Figure 6 Crossflow Software Change Process

Provisions for an "engineering release" of the software have been made to allow rapid, informal releases. All changes to the baselined software will be tracked via SCRs even within the informal, engineering release process. These releases, while still controlled by the formal CM tool, PCMS, can be generated quickly by eliminating documentation and regression testing requirements at this step. It should be understood that these releases will never be used in support of any formal testing activity in the lab.

Several changes (SCRs) will build up during the engineering release process and those will be grouped together for a block release at appropriate times such as avionics blocks or updated aero data set releases. This process constitutes a more "formal release" to satisfy deliverable requirements. An engineering review board will by tasked with approving all changes that will be incorporated in a particular block release version. Each using lab will be represented directly or indirectly on this board. Each formal release will undergo a regression test appropriate for the types of changes included. Documentation, including a version description document (VDD), will be updated to be representative of that version. Documentation updates will be possible by making use of the SCRs created during the engineering releases. No lab will be forced to immediately update to the newly released version if circumstances dictate, but are encouraged to do so at the earliest opportune moment.

All using labs will be able to submit SCRs to request corrections to problems or to add new requirements they deem necessary. New requirements to accommodate air vehicle changes, new aero data sets or changes due to flight tests may also be input. These new requirements or anomaly corrections must be approved by the engineering review board representing all using labs. The change process will be kept at this level unless disagreements between different labs occur. Any such disagreements would then be elevated up the IPT chain.

CONCLUSION

There are many challenges facing REUSE development, including identifying PTSD requirements in time to achieve maximum synergism across the weapon system. *Meeting deliverable training system standards and* REUSE requirements of several IPTs (labs) is difficult because for the so many different uses for the simulation, e.g., test and verification, integration, analysis and demonstration, and training. Provisions had to be made for at least two different target computers. And a specific training system concern is for REUSE items to meet vigorous life-cycle support requirements.

In addition to the lessons learned during each phase of development described above, some general lessons need to be mentioned. All lessons are preliminary and will be until the ultimate test of the REUSE items, i.e., integration and life-cycle support in all the target REUSE areas.

We applied the ADARTS process but many of the developers doubt the benefit of this process to our specific application. It is our opinion that ADARTS is better suited for event driven or I/O intensive applications. This is not the case of FDS. In addition, we reused algorithms (if not code) which, along with our performance constraints, dictated a design. ADARTS could not add anything to that structure.

One benefit of ADARTS was that it was used to confirm the architecture and structure we had already established. If ADARTS had indicated something drastically different, we would have reexamined our design.

When you change your "mindset" to include REUSE, you potentially find more than you thought you would. Several ATLAS test cases have been identified for REUSE from the Flight Dynamics and Weapons Separation Group. These test cases will be developed for verification use in the Handling Qualities Simulator, VIF, VSS, and PTSD simulator. In addition, we have begun to look at the IRSS in the VIF as a potential candidate for Opportunistic REUSE.

Chances of success of planned REUSE are enhanced by explicit wording in the contract and in the SOW, early involvement by all potential reusers, and co-development. Defining a software architecture up front improves satisfaction of non-functional requirements, i.e., maintainability, modifiability, and scalability.

Co-development of *FDS* had its challenges. For example, merging engineering lab culture (quick turnaround; support of flight test; and informal, in-house documentation); with training device culture (deliverable, formal documentation, life-cycle support) was not easy. However, we discovered advantages to this merge. There was a merging of varied skills and expertise, i.e., Ada and deliverable processes expertise from PTSD; flight dynamics and extensive real-time experience from FSL. The result was that the "whole was stronger than the sum of the individual parts."

We believe that the F-22 is developing REUSE strategies that will ultimately result in deploying and maintaining concurrent trainers, at a reduced life-cycle cost.

REFERENCES

1. F-22 Integrated Master Plan, 2 January 1991
2. F-22 Weapon System Specification. RFP No. F33657-91-R-0006, 2 January 1991
3. F-22 Training System Specification, 2 January 1991.
4. Lockheed Aeronautical Systems Company, F-22 WS SDP for E&MD. Contract No. F33657-91-C-0006, November 4, 1993.
5. Boeing Defense and Space Group, F-22 Training System Software Development Plan for E&MD, January 1993.
6. Nordwall, Bruce. "Defense Department Expects New Strategy for Improving Software to Save Billions," Aviation Week & Space Technology, June 22, 1992.
7. Baldwin and Landry, "F-22 Innovations for Concurrent Development of Pilot Training System Devices," 14th I/ITSEC, 1993.
8. F-22 Joint Procedure No. 169. "Software Reuse Guidelines, July 15, 1993.
9. An Object-Oriented Design Paradigm for Flight Simulation, 2nd Edition Technical Report, CMU/ SEI-88-TR-30, September. 1988.
10. An Introduction to Structural Models; presented to the 14th I/ITSEC, November 1992.
11. Structural Models for Real-Time Simulation; training charts from Software Engineering Institute, Carnegie-Mellon University, March 1993.
12. Draft Structural Modeling Guidebook; Software Engineering Institute, Carnegie-Mellon University, and NSIA, April 1993.

ACRONYMS

A/V air vehicle
ACS air combat simulator
ADARTS Ada-Based Design Approach for Real-Time Systems
ADSS Avionics Development Simulation System
ASVP Ada Simulator Validation Program
ATLAS airframe trim linearization and simulation

CDR Critical Design Review
CM configuration management
CMU Carnegie-Mellon University
CNISS Communication, Navigation, Identification Simulation System
CSCI Computer Software Configuration Item

Dem/Val demonstration and validation

E&MD Engineering and Manufacturing Development
EWSS Electronic Warfare Simulation System

FDS flight dynamics simulation
FSL Flight Simulation Laboratory

IPT Integrated Product Team
IRS Interface Requirements Specification
IRSS Inertial Reference System Simulation
ISD Instructional System Development

LASC Lockheed Aeronautical Systems Corporation
LFWC Lockheed Fort Worth Company (formerly General Dynamics, Fort Worth Division)

Nighthawk Harris Computer
NSIA National Security Industrial Association

OFP Operational Flight Program

PCMS Product Configuration Management System
PDR Preliminary Design Review
PTS Pilot Training System
PTSD Pilot Training System Devices

SCR System Change Request
SDD System Design Document
SDL Software Development Lab
SDP Software Development Plan
SEI Software Engineering Institute
SLOC Source Lines of Code
SOF ATS Special Operations Forces Aircrew Training System
SOW Statement of Work
SPE Software Product Evaluation
SPO System Program Office
SRS Software Requirements Specification
S/SEE System/Software Engineering Environment
SSR Software Specification Review
SSS System Segment Specification

TPM Technical Performance Metrics
TS training system
TS SDP Training System Software Development Plan

VDD version description document
VIF VMS Integration Facility
VMS Vehicle Management System
VSS Vehicle System Simulator

WS weapon system
WS SDP Weapon System Software Development Plan