# Using WorkXpert to Implement RASSP Electronic Design Workflows

Jeffrey Stavash
Lockheed Martin Advanced Technology Laboratories
1 Federal Street, Camden, NJ 08102
jstavash@atl.lmco.com
(609) 338 - 4031

## Abstract

*The goal of the ARPA/Tri-Service sponsored Rapid Prototyping of Application-Specific Signal Processors (RASSP) program is to improve, by at least a factor of four, the time-to-market, life-cycle cost, and design quality of Digital Signal Processor (DSP) systems. One way this goal is being achieved is through the use of enterprise system technology. Lockheed Martin Advanced Technology Laboratories (ATL) is developing an enterprise system which integrates a workflow manager, product data manager, reuse data manager, Computer-Aided Design (CAD) tools, and network services into a single cohesive framework. The workflow manager guides the user through the design process with graphical workflows. The product data manager configuration manages the design data that's being produced. The reuse data manager supports the cataloging and searching for reusable design objects. Network services addresses the design-to-manufacturing interface via a secure network protocol. By integrating workflows, product data, CAD tools, and network services together, an enterprise system enables a concurrent/collaborative approach to DSP design that embraces the entire life cycle from requirements to manufacture. This paper describes how workflows for electronic design were developed on the RASSP program by ATL, and how those workflows were implemented using the Mentor Graphics WorkXpert product.*

## 1. Introduction

Workflows are a graphical representation of ordered tasks that need to be performed to achieve a desired goal. Virtually any effort that can be broken down into a series of individual steps can be modeled by a workflow. Workflows help you do several things:

- **Guides you through the design process.** The graphical representation of the design process indicates which steps come next.
- **Enforce certain design steps.** The dependencies of tasks ensure that key steps of the design process are not omitted.
- **Monitor progress of designs.** Tracking the completed steps allows for better management of the design schedule.
- **Identify problem areas in the design process.** The tracking of design progress helps identify bottlenecks in the workflow.

The RASSP methodology consists of three main design phases: System Definition; Architecture Definition; and Detailed Design. These design phases are hierarchical and are decomposed into leaf-level process flows. The System Definition and Architecture Definition phases address the translation of customer requirements into a candidate architecture of hardware and software elements. The Detailed Design phase contains process flows for designing an Application-Specific Integrated Circuit (ASIC), Field-Programmable Gate Array (FPGA), Multi-Chip Module (MCM), and a Printed Circuit Board (PCB) [1]. Workflows have been developed for all of these design phases in order to achieve design process consistency and reduce errors.

## 2. Workflow Development

Workflow development for the design phases of RASSP began by first developing process models using the IDEF3X modeling method. Rockwell International Corporation developed this method as an extension of the IDEF3 process description capture method [2]. The name IDEF originates from the Air Force program for Integrated Computer-Aided Manufacturing (ICAM), where the first ICAM Definition (IDEF) methods emerged. It is now called Integration Definition. IDEF3 was specifically created to capture descriptions of sequences of activities. It can be distinguished from other process modeling methods because it facilitates the capture of the description of what a system actually does [3].

IDEF3X combines the ICOM (input, control, output, mechanism) feature of IDEF0 with the process flow description of IDEF3, along with some additional features to facilitate implementation by a workflow management tool. The syntactic elements of an IDEF3X model are similar to IDEF3 and include units of behavior (UOBs), junction boxes, and precedence links. Additional features include:

- Identifying ICOMs by naming the object and its life cycle state separated by an asterisk (e.g., Draft*Publication - where Publication is the name of the object and Draft is its current state)
- Object state links identifying the flow of data between UOBs
- Feedback links indicating failback paths
- Annotating the name of the junction boxes with an “A” or “S” to indicate an asynchronous or synchronous junction
- Support for additional precedence links such as Start-Start, Start-Finish, Finish-Start, Finish-Finish, Concurrent, Cascade, Fail-Reset, and Fail-Cascade
- Annotating the precedence links with a “P:” followed by a two-letter code which indicates the precedence between the parent and dependent UOB.

## 3. Modeling Example

This example is taken from the RASSP Detailed Design workflows. All the components necessary to create a leaf-level workflow are summarized. Figure 1 contains a

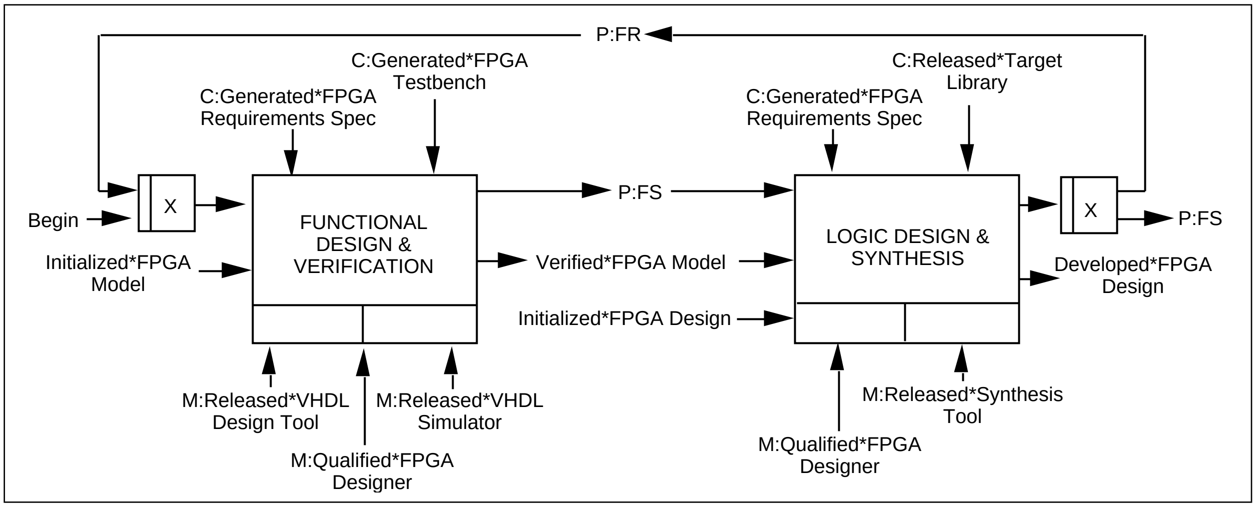
P:FR
C:Generated*FPGA Testbench
C:Generated*FPGA Requirements Spec
C:Released*Target Library
C:Generated*FPGA Requirements Spec
Begin
X
FUNCTIONAL DESIGN & VERIFICATION
P:FS
LOGIC DESIGN & SYNTHESIS
X
P:FS
Initialized*FPGA Model
Verified*FPGA Model
Developed*FPGA Design
Initialized*FPGA Design
M:Released*VHDL Design Tool
M:Released*VHDL Simulator
M:Qualified*FPGA Designer
M:Released*Synthesis Tool
M:Qualified*FPGA Designer

***Figure 1.*** *IDEF3X Process Model*

portion of the FPGA design workflow represented as an IDEF3X process model. The model contains two UOBs that represent design activities:  Functional Design & Verification and Logic Design & Synthesis. Precedence links, object state links, junction boxes, and all of the ICOMs for each UOB are also present. This model was developed using the TopDown Flowcharter tool by Kaetron Software Corporation.

The workflow begins with Functional Design & Verification. This task will develop and simulate an FPGA behavioral model, using VHDL, so it can be synthesized down to the specific FPGA Programmable Function Unit (PFU) level. The input to this task is *Initialized*FPGA Model*. The controls for this task are *C:Generated*FPGA Testbench* and *C:Generated*FPGA Requirements Spec*. The output from this task is *Verified* FPGA Model* and the mechanisms are *M:Released*VHDL Design Tool, M:Released* VHDL Simulator*, and *M:Qualified:FPGA Designer*. Notice that the FPGA Model is both an input and output object, but its state has changed from *Initialized to Verified*. This reflects the processing performed to refine it from the behavioral level of abstraction to the Register-Transfer Level (RTL) of abstraction in preparation for synthesis. The Exclusive OR (XOR) junction box to the left of this activity indicates its execution precedence. This activity can be executed initially via the Begin link or repeatedly via the P:FR feedback link from the Logic Design & Synthesis UOB.

The next task to be performed is Logic Design & Synthesis. This is indicated by the P:FS precedence link between the two UOBs. This type of precedence link indicates that the parent task, Functional Design & Verification, must finish before Logic Design & Synthesis, the dependent task, can start. The purpose of this task is to synthesize the RTL FPGA model into a gate-level design. The inputs for this task are *Verified* FPGA Model* and *Initialized*FPGA Design*. The controls are *C:Released*Target Library* and *C:Generated*FPGA Requirements Spec*. The output from this task is *Developed* FPGA Design* and the mechanisms are *M:Released*Synthesis Tool* and *M:Qualified**

*FPGA Designer*. For this task, the FPGA Design is both an input and output object and its state has changed from *Initialized to Developed*, indicating a synthesized design has been developed. If the results of performing the synthesis meet the requirements, the FPGA design is made available to the placement & routing task (not shown). The XOR junction box to the right of the Logic Design & Synthesis task indicates execution precedence. The remaining activities of this workflow may be executed via the P:FS precedence link or the Functional Design & Verification task may be executed again via the P:FR feedback link. This will occur when Logic Design & Synthesis has not completed successfully and the FPGA model needs to be modified. It is important to note that if the Functional Design & Verification task is re-executed, the Logic Design & Synthesis task will also be re-executed.

## 4. Workflow Implementation

The process models for the Detailed Design phase of RASSP were implemented as workflows using Mentor Graphics' WorkXpert product. Workflow templates were created in XpertBuilder for the ASIC, FPGA, & Module design processes. Figure 2 contains the template for the Module preliminary design process. The RASSP electronic design workflows were designed to be CAD tool independent. Consequently there is not always a one-to-one correspondence between workflow tasks and design tools. Some workflow tasks require the execution of multiple design tools [4]. With the capability of WorkXpert to execute either AMPLE, C, or Shell programs, this is not a problem. A Shell script or C program can be written to invoke as many different design tools as needed within a workflow task. Alternatively, the workflow tasks which require the execution of multiple design tools could be decomposed into subflows. For RASSP however, this is not the desired approach because creating additional levels of workflow hierarchy can result in tool dependent workflows.

Task precedence is another important issue when implementing workflows. The eight different workflow precedences defined for RASSP are *Start-Start, Start-Finish, Finish-Start, Finish-Finish, Concurrent, Cascade, Fail-Reset*, and *Fail-Cascade*. These precedences can be interpreted as a parent-child relationship. For example, a Finish-Start precedence between workflow tasks *Logic Design & Simulation and Preliminary Placement* means that *Preliminary Placement*, the child task, cannot start until the parent task, *Logic Design & Simulation*, has completed successfully. WorkXpert provides a mechanism called Triggers, which allows control over start and finish conditions and reset propagation [5]. Using step-based triggers, the *Start-Start, Start-Finish, Finish-Start, Finish-Finish*, and *Fail-Reset* precedences were implemented. The *Concurrent, Cascade*, and *Fail-Cascade* precedences were not required for the RASSP detailed design workflows. The workflow template shown in Figure 2 uses step-based triggers to implement the *Finish-Start* and *Fail-Reset* precedences.

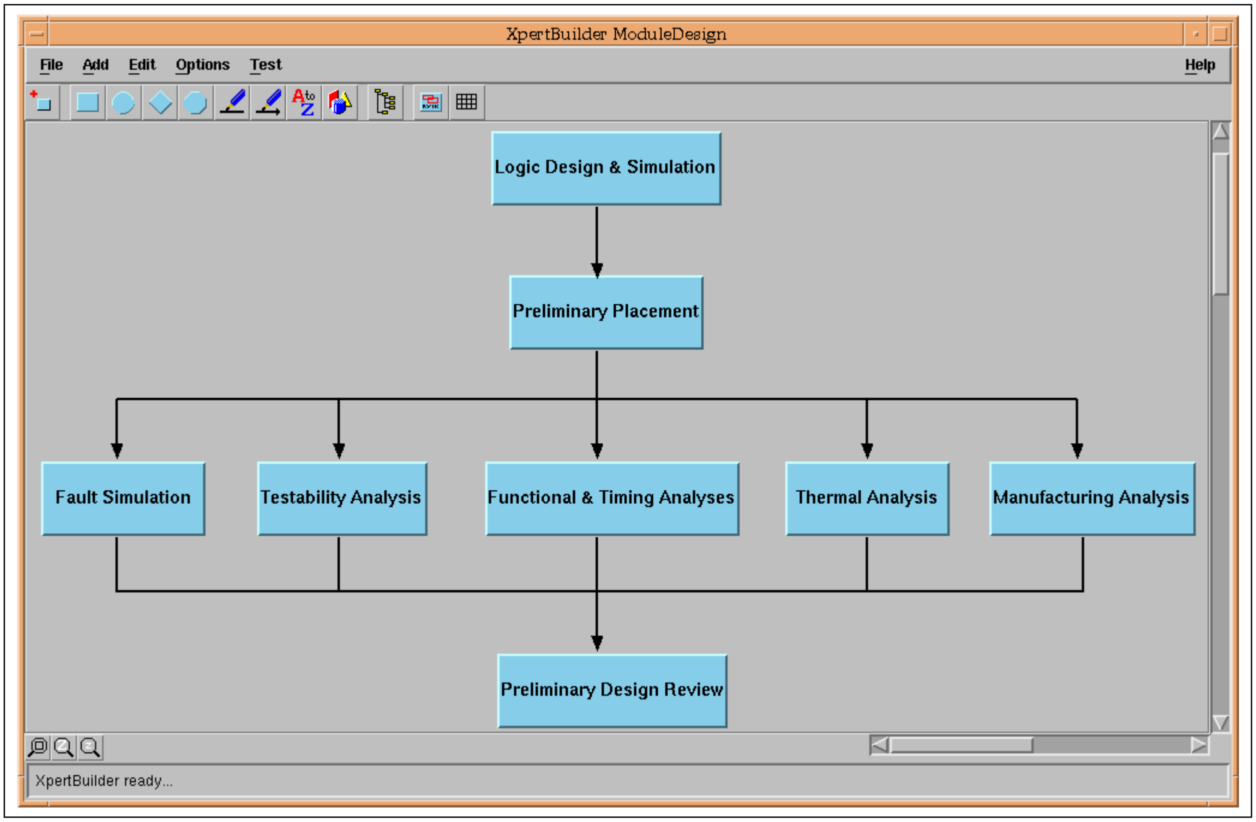

***Figure 2.*** *Module Preliminary Design Workflow Template*

The RASSP electronic design workflows were also designed for use on multi-user projects. Within that environment, it is necessary to restrict workflow access to certain users. For example, the workflow template shown in Figure 2 contains a Preliminary Design Review task. This task must be executed by the Project Manager or the Project Lead. WorkXpert restricts workflow access through the use of Role-Based Access Controls to associate specific operations with specific users.

### 4.1 Integration with a Product Data Manager

Within the RASSP design environment, it is desirable to have the workflow manager integrated with a Product Data Manager (PDM). When a workflow task is executed, any existing required data would be checked-out of the PDM and made available to the user. Upon completion of the workflow task, the modified data is checked back in to the PDM along with any new data that was created. This is the principle behind how PDM systems manage product data. The master copy is held once in a secure ‘vault’ where its integrity can be assured. Duplicate copies can be distributed freely to users. When a change is made to the data, a modified copy is stored in the vault alongside the master copy which remains in its original state as permanent record.

WorkXpert is currently not integrated with a PDM system. However, most PDM systems (e.g. metaphase) have an Application Programming Interface (API) capability. So it should be possible to create an integration by embedding PDM API calls within a C program that's executed by a WorkXpert workflow task. ATL is currently investigating this possibility on the RASSP program. Another approach to integrating a workflow manager with a PDM system is through the design tools. With an increasing number of interfaces between PDM systems and design tools being made available (e.g. MGC-metaphase), it is possible to delegate the PDM requirement to the actual design tools. For RASSP however, this is not the desired approach. The goal is to have the PDM integration implemented at the level of the workflow manager.

## 5. Summar y

Mentor Graphics' WorkXpert product can easily implement the workflows from the design phases of the RASSP methodology. Although only the workflows from the detailed design phase of RASSP have been implemented, the tool has the capability to implement workflows from the other remaining design phases. Its extensive programming features, such as support for preExec, Exec, and postExec actions on workflow tasks, decisions, and subflows, permit greater flexibility in customizing workflows to better fit an organization's particular situation. Through the use of triggers and the many different step states, complex process and data dependencies can be handled. With support for executing AMPLE, C, or Shell programs, a rudimentary form of PDM integration can be achieved. For the remainder of the RASSP program, ATL will continue to investigate the benefits of using WorkXpert as the workflow manager within an enterprise system.

## References

1. Lockheed Martin, RASSP Methodology, Version 2.0, October 1995.

2. Rockwell-NAAD, RASSP Workflow/Information Modeling Overview, May 1994.

3. Mayer, Richard J., Cullinane, Thomas, deWitte, Paula S., Knappenberger, William B., Perakath, Benjamin, Wells, M. Sue, IDEF3 Process Description Capture Method Report, Knowledge Based Systems, Incorporated, May 1992.

4. Stavash J., "Workflow Modeling for Implementing Complex, CAD-Based Design Methodologies," Proceedings of the Mentor Graphics Users Group International Conference, October 1996.

5. Mentor Graphics Corporation, WorkXpert Developer's Guide, V1.6, 1996.