Architectural Patterns for the Extend Enterprise

Janet E. Wedgwood, Jeffrey Stavash, John Sausman, Dr. Bipin Chadha, Harris Zebrowitz

Lockheed Martin
Advanced Technology Laboratories

April 17, 2002
What is a Virtual Prototyping Environment?

- An architecture to dynamically integrate systems that represent the product data and behavior of a complex system in an extended enterprise
- The software that implements the enterprise
- The standards used for interoperability
- A collaborative environment in which a multi-organizational team can:
  - Develop, manage, analyze, and optimize product definition and behavior throughout its lifecycle

A Virtual Prototype is an instantiation of a design in the Virtual Prototyping Environment
Why Build a Virtual Prototype?

- Increased complexity increases lifecycle costs
- Key to addressing lifecycle costs early in the design cycle
  - Perturbations in the design are more practical in the early design stages
- All aspects of product definition and behavior can be represented by the virtual prototype to support complex analysis
  - Trade studies that drive decisions that affect cost in all lifecycle phases can be performed
  - VPE makes multi-domain optimization practical
Properties of the VPE Architecture

- Accessible
- Scalable
- Extensible
- Interoperable
- Reusable
- Available
- Deployable
- Affordable
- Reliable
- Maintainable
How Will the Architecture Evolve?

- Architecture must support the product over a long life-cycle (many decades)
- Must adapt to changes in:
  - Requirements
  - Technologies
  - Business practices
  - Organizations
  - Roles
Architectural Approach

- Managed and controlled evolution of a stable architecture
  - Decouple fast changing things from slow changing things
  - Decouple Information Architecture from System Architecture
  - Decouple Logical Architecture from Physical Architecture
  - Use pervasive standards in the development of information architecture to enable interoperability
  - Use Component Based Architectures with Explicit Interfaces for easier replacement
Components can be implemented incrementally; supports many standard protocols

VPE Logical Architecture

Web Browser

HTML, XML, HLA, JAVA, CORBA

Domain Tools

Visualization Tools

Models

Simulations

Databases
Virtual Prototyping Environment (VPE)

Cost Analysis | Systems Engineering | Performance Analysis | Task Analysis | ...

MDO
MP
IDE

Collaborative Environment

Component based with explicit interfaces for easier replacement; layers isolate changes, spread responsibility to accommodate change, and promote reuse
VPE Design Patterns

- Catalog/occurrence
- Design specific attributes
- MVC
Design Specific Attributes Pattern

The pattern is used to “snap on” functionality to the model without affecting existing data or requiring database schema modifications.
MVC Pattern

The Product Manager MVC architecture mirrors the application thread “snap-on” architecture
The Product Manager MVC architecture mirrors the application thread “snap-on” architecture
Enable different disciplines to view the product structure from their own perspectives; maintain overall integrity between different disciplines; and design phase view vs. perspective (adopted from OMG data architecture)
Product definition includes all design information including a consolidated representation of hardware, software and people.
MDOM enables sophisticated analysis and optimization capabilities by leveraging MPS and enabling a "sandbox" environment.
Lessons Learned

- Special skills required to translate user requirements into a useable, multidisciplinary tool environment
  - Design-a-little, Build-a-little definitely needed for VPE designers to understand domain requirements (i.e. VPE designers are not propulsion engineers…)
  - Use design experiences to prevent documenting bad business processes

- Application threads must be independent of deployment environment
  - Service Architecture is critical

- Design Build plan must mirror thread-oriented design environment
Future

- Application thread tools will be offered as services
  - Jini-like application architecture
  - Users can request services
  - Services can be access controlled

- Design/build environment improvements are being studied
  - Multiple sites, threads, versions, projects

- Integration of tools based on complexity science show promise of increased ability to explore the design space
Examples of Complex Systems
It is desirable to understand the properties of transition points and tools to position and maintain systems at the transition point.