Experiences in Model Driven Architecture for Enterprise IT Management

(Junkyard Modeling)

Dan Matheson
dan_matheson@hp.com

Jishnu Mukerji
jishnu_mukerji@hp.com
OpenView Software Situation

- 10+ development years of network and systems management software in previously independent divisions of HP.
- Acquisitions of products in this area: Trinagy, Compaq TeMIP, etc.
- Targeted at different markets: Telco, IP networks, fault detection, performance, applications.
- Different target platforms: HW, OS, DB, Language
Customer Consequences

• Some products come with a customer model, but can’t be shared among products.
• Most products have configuration items, but all have differences in structure and attributes.
• Each product has its own authentication mechanism.
• Reporting mechanism vary in capabilities.
Architecture Driving Forces

• One OpenView
  – Customers need OV products to work better together as they use multiple products
  – A more uniform look and behavior

• No duplication
  – Remove the inefficiencies from the product development processes
  – Common components for consistent behavior
Business Realities

- Remain Profitable – cannot stop to re-do everything
- Product release compatibility
- Product release process at different stages for different products
- Distributed product development environment
- Islands of specialized knowledge, technology skills, and different processes
The Dilemma

• Do the needs of the many outweigh the needs of the one or do the needs of the one outweigh the needs of the many? - use a common component or have a timely release?

The Solution (in part)

• Modeling as a mechanism to present the alternatives for a more open and thoughtful decision.
Modeling Issues 1

- Lack of knowledge about MDA concepts.
- Differences in UML knowledge.
- Differences in OO analysis and design skills.
- Differences in product vocabulary.
- Different UML CAD tool experience (Rose, Poseidon, Visio, PowerPoint).
Modeling Issues 2

- People were often only email acquaintances.
- Limited face-to-face time.
- Different native languages and work customs.
Approach - Using MDA

- Explain the MDA concepts to other OpenView architects.
- The PIM/PSM concepts helped keep discussions focused.
- The refinement concepts helped us work at higher levels of abstraction.
- There was agreement that we did not want to get bogged down in the code.
Approach - Using MDA - PSM

• The target platforms include multiple dimensions
  – HW/OS: HP-UX, Solaris, AIX, Linux, Windows\textsuperscript{*}, Mac
  – Languages: C, C++, Java, Perl, sh, HTML, XML
  – Databases: Solid, Oracle, Sybase, NIH home brew
  – Middleware: Bluestone, Apache, WebLogic, Tibco, JMS, DCE, CORBA
  – Home Brew libraries: logging, authentication, object server, measurement server, etc.
  – Standards: IETF, DMTF, TMF, OSS/J, Telecom, etc.
  – Future: .NET, …
Approach - Using MDA - PSM

• Step 1 was the initial survey that produced the list on the previous slide.
• The Object Server technology was picked for core use and as the first to model.
• The Object Server is being re-coded in Pure Java from MS J++ for OS platform independence.
Object Server PSM

Object

Entity

Attribute Ui Item

Attribute

LifeCycle

20 September 2002

3rd UML Workshop - MDA for Enterprise IT Management
Approach - Using UML

- Did not try to generate UML from hundreds of thousands of lines of code. No spaghetti UML from spaghetti code.
- Used the experience, partitioning and a top down view to generate first UML.
- Collected use cases as people examined the UML and asked questions.
Some Results – Huge Model from Big Models
Problem – Huge Model from Big Models

- How to manage a set of models that are independently developed, yet need to be brought together for the big view and for cross checking?
- Not working from diagrams at this level.
Suggestions – Huge Model from Big Models

• A UML CAD tool should be able to concurrently have multiple models loaded. Models as standard parts that get assembled.
• Easily switch display and move complete artifacts between models.
• Display concurrently artifacts from different models.
• Cut-out and save part of a model as a new independent model or merge 2 models.
• A note type artifact that appears in the browser window would be handy.
Some Results - PIM/PSM
Some Results - PIM/PSM

PIM

<<refinement>>

<<Abstract>>
AbstractServiceResource
(from OvCoreBaseNetworkTopologyPIM)

<<Abstract>>
Node
(from OvCoreBaseNetworkTopologyPIM)

PSM

<<Abstract>>
ManagedEntity
(from Network_data_primitives)

<<Abstract>>
SNMPNode
(from Network_data_primitives)

Entity

Object

LifeCycle

NetworkNode
(from Network_data_primitives)
Problems - PIM/PSM

• Currently we have a simple 1 PIM to 1 PSM model. We will need >1 PIM to >1 PSM relationships.

• What is an effective way to show the technologies involved in the PSM?

• What is an effective way to relate PIM or PSM models to standards: TMF, DMTF, Java JSRs, etc.?
Problems – Complex Concept Communication

- Documentation of static structure and behavior is spread across multiple diagrams.

- Integration still happens in the engineer’s head, while this will not go away, what can be done to make communication easier?
Suggestions – Complex Concept Communication

• Composite diagrams that include multiple simple diagrams. Typical in mechanical engineering.
• Kinematics diagrams, for example that show creation of instance artifacts as a sequence diagram is enacted or state diagram is traversed.
• Others.
Problems – IDE Tool View

• The current UML tools, in my opinion, are centered around the code Integrated Development Environment.

• This code IDE centric seems to have kept the usefulness of the tools in the “designing and programming in the small” domain.

• This is *not* my current problem.
Suggestions – IDE Tool View

- The CAD/CAM separation in the mechanical engineering world seems reasonable.
- The CAD tools create the model.
- The various CAM tools analyze or realize the model.
- There are 3D scanners to produce a model from a real part.
Suggestions – IDE Tool View

• I would like a CAD tool targeted at model creation, refinement / refactoring and management
  – I want the ability to create and manage alternative designs in parallel
  – I want to create software standard parts and reuse them
  – A tool that does one thing well so I don’t sort through commands
Suggestions – IDE Tool View

• Use XMI as the interchange language from CAD tool to analysis tools
• I want to be able to input the model into various analysis tools and pull a perhaps modified model or a model with improvement suggestions back into the CAD tool.
• Pattern analysis, performance analysis, security analysis, etc.
Suggestions – IDE Tool View

- Use XMI as the interchange language from CAD tool to CAM tools
- I want to be able to input the model into various CAM tools
  - Produce code (Java, C, C++, SQL, Perl, …)
  - Produce compiled code
  - Produce install packages (*ux, Windows, Mac)
  - Produce a particular PSM model from a PIM model
Suggestions – IDE Tool View

• Consequences of this approach
  – A central storage with access control and version capabilities is needed – Product Data Management
  – A tool neutral model storage format – XMI is good
  – A model as an assembly of other models
  – Supports many development process styles
  – Easy to add or change tools used
My personal view.

HP is devoting an increasingly large part of its computer activity to the development of software – the instructions that tell the computer what to do.

Because the design *IS* the product,

“manufacturing” software of high quality presents unusual production challenges.

From the 1982 Hewlett-Packard Company Annual report, page 14