

VALUE OF USING SYSML IN A SPACECRAFT DEVELOPMENT ENVIRONMENT

**An Integrative Project in Partial Fulfillment of the
Requirements for the Degree Master of Systems Engineering**

**Janis Hebermehl
LOYOLA MARYMOUNT UNIVERSITY
December 2007**

Agenda

- **INTRODUCTION**
 - Background
 - Rationale & Objectives
- **OVERVIEW**
 - Useful Terms & Definitions
 - Systems Architecture
 - Architecture Modeling
- **METHODOLOGY**
 - Approach
 - Data Collection
- **DATA ANALYSES AND RESULTS**
 - Challenges & Stakeholder Concerns
 - Value Definition & Top-Level Requirements
 - Modeling Language & Tool Evaluation
 - SysML Evaluation
 - Other Implications
- **CONCLUSION**
- **Q & A**

INTRODUCTION

- Background
- Rational & Objectives

Background

- A system's architecture is the conceptual structure of the system. Its development begins in the earliest phases of customer needs analysis.
- The architectural level of systems development consists of making decisions about overall design goals, requirements, structure, and development strategies.
- Over the last several decades systems architecting has been evolving.
- There have been significant advances in systems engineering and development technologies to directly address:
 - Large, complex system of systems, and systems that present formidable risks and difficulties in their design, construction, deployment and evolution.

Background (continued)

- Many analysis and development innovations have resulted from this attention to the architectural level such as:
 - Tailored architecture modeling languages
 - Extensive architecture frameworks
 - New architecture modeling tools
- Boeing has capitalized on these advancements by investing in the development of processes and tools.

Project Rationale & Objectives

- This project provides a frame of reference to address implementation and adoption decisions for system architecture modeling tools. It attempts to:
 - Define useful terms and guidelines for the consistent application of architecture to systems throughout their life cycle
 - Present system architecture principles and the anticipated benefits for model driven architecture
 - Provide observations on system architecture modeling languages and tools currently in use
 - Identify challenges facing the systems engineering community with model driven architecture
- To these ends, this project is intended to determine the value of implementing SysML in a spacecraft development environment.

OVERVIEW

- Useful Terms & Definitions
- Systems Architecture (SA)
- Architecture Framework
- Architecture Modeling
- Architecture Modeling Language
- Context Diagram for Systems Architecture

Overview – Useful Terms & Definitions

- **Systems Architecture (SA)** - “the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution” (IEEE 1471, 2000)
- **Architecture Framework** – “defines a common approach for architecture description development, presentation, and integration for both Warfighting operations and business operations and processes” (DODAF, Definitions & Guidelines, V1.5)
 - Intended to ensure that architecture descriptions can be compared and related across organizational boundaries, including Joint and multinational boundaries
- **Architecture Models** - a mechanism to communicate the vision and track construction against it for the system to be built as envisioned
- **Architecture Modeling Language** - Visual modeling language that is used to specify, visualize, construct, and document the artifacts of a system
- **Architecture Modeling Tools** - allows users to extend the modeling environment into specific modeling language capabilities.
 - Enables building an architecture — with an integrated collection of models

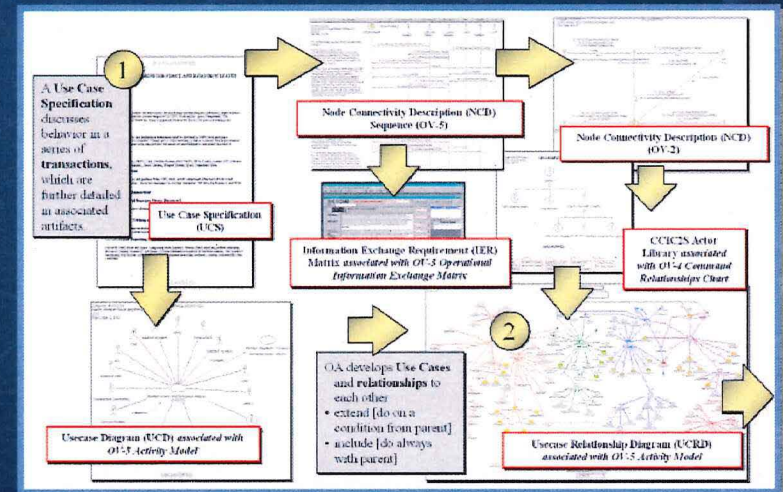
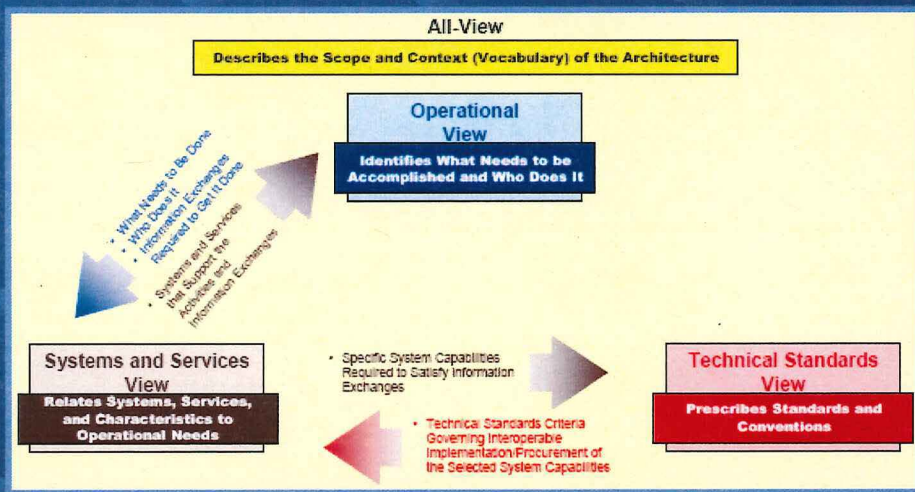
Systems Architecture (SA)

- SA provides traceability between the system requirements and the system functionality, furthermore, between the functionality and the operational context that prescribes that functionality.
- SA provides a framework for a common interpretation of the intended system functionality.
 - It provides a means for early communication of the initial system architecture with the stakeholders.
- SA identifies the common system elements and standards that are to be shared across the system.
- SA identifies all system elements, the behavior allocated to each of the system elements, and the relationships among the system elements.
- The use of a single SA model will result in the successful horizontal integration of the system development team.

The architect must apply a systems approach that focuses on the system as a whole, this requires understanding the stakeholder goals

Architecture Framework

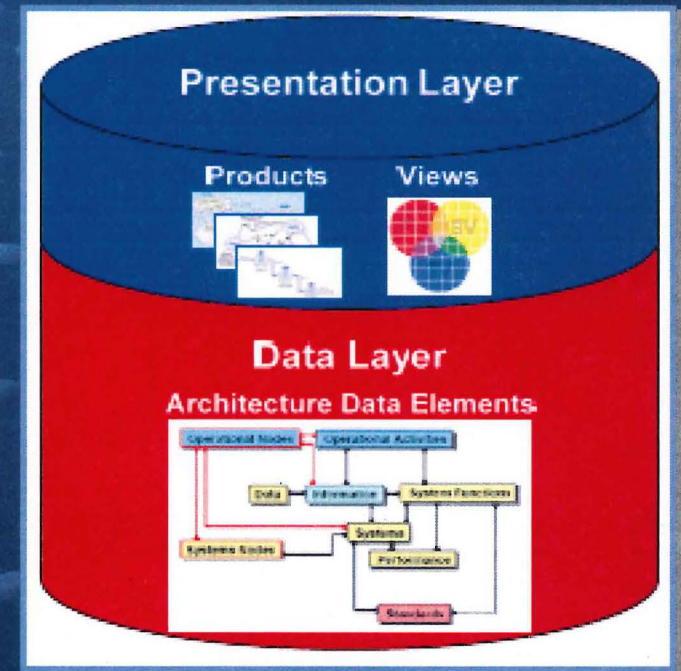
- Provides guidance, rules, and product descriptions for developing and presenting architecture descriptions that ensure a common denominator for understanding, comparing, and integrating systems.
- In the DOD Architecture Framework (DODAF), there are three major perspectives (i.e., viewpoints) that logically combine to constitute a significant portion of an architecture description.



Linkage among the views serves to ensure that a single architecture is described that can actually be developed and operated

Architecture Modeling

- Mechanism to communicate the vision and track construction against it.
- Help minimize the risk that significant elements of an architecture are inadequately communicated to stakeholders.
- Intended to provide insights into particular aspects of the system and related issues.
- Are interrelated because they are elements of a single integrated systems architecture.
- Are abstractions of the as yet unrealized system.
 - Contents are dictated by the purpose of the model.
 - Some are for use in analysis while others are primarily for visualization.

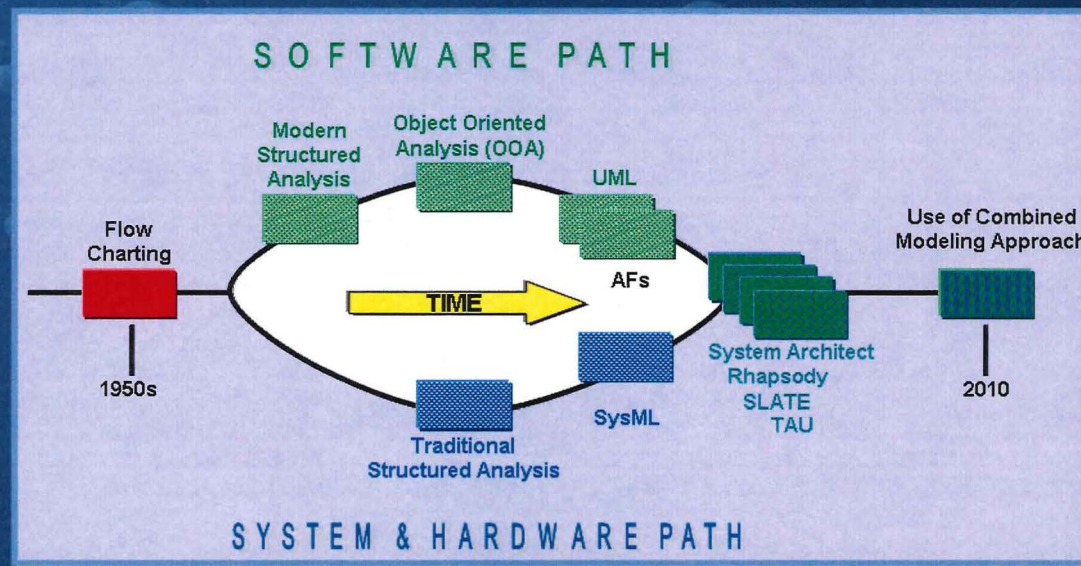


The advantages of models are improved communications, reduced ambiguity and more complete representation

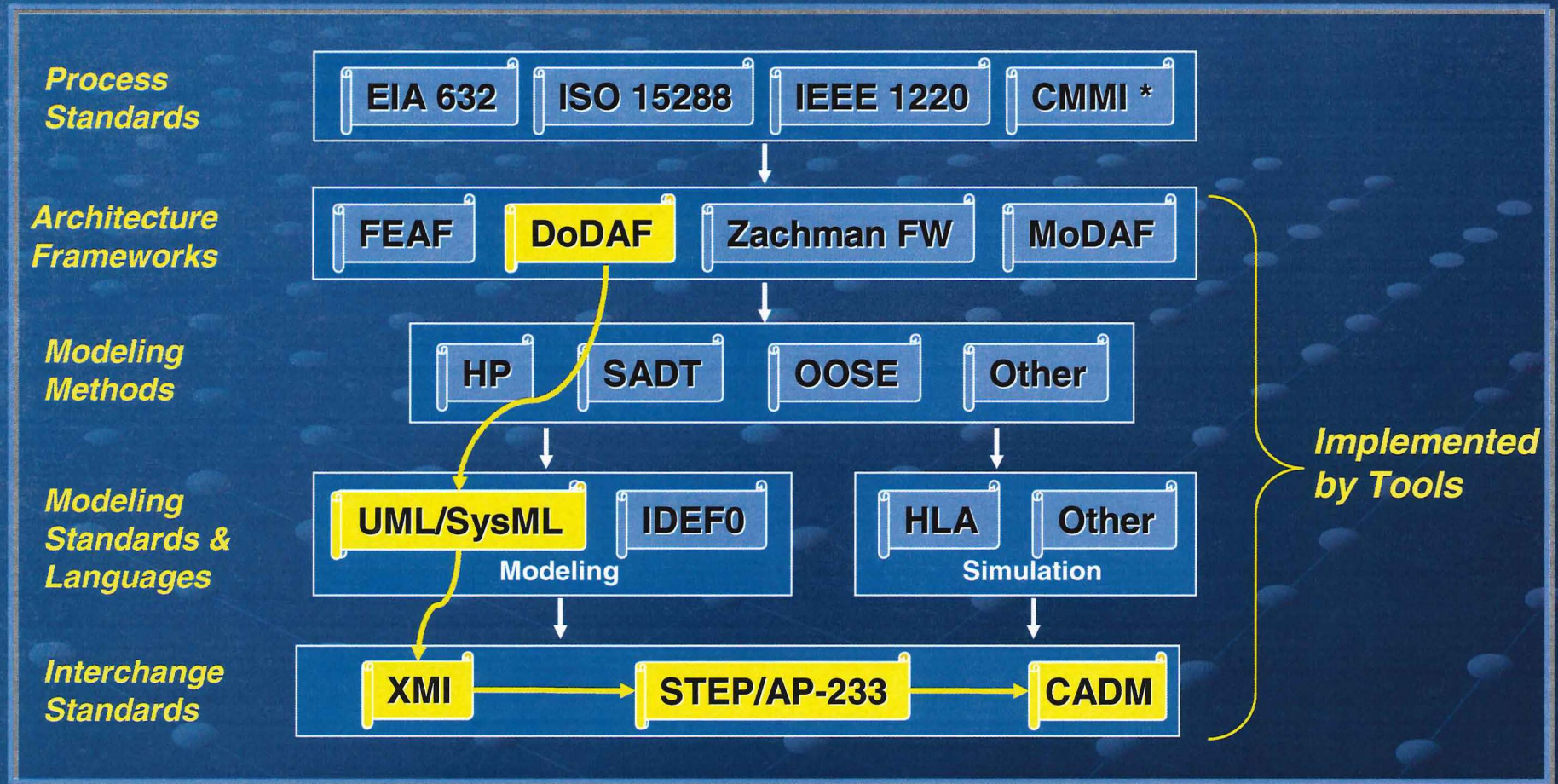
Architecture Modeling Language

The modeling languages addressed in this project are:

- **UML** (Unified Modeling Language) - system level visual language to support tasks of analyzing, specifying, designing, verifying and validating software systems.
- **SysML** (Systems Modeling Language) is based on a subset of UML
 - New standard language for system modeling
 - Built on top of UML
 - Tailored for the specific needs of systems engineers.



Context Diagram for Architecture & Systems Engineering Standards



This shows only one thread through these standards

Architecture Modeling Tools

- Modeling tools provides users a means to extend the modeling environment into specific modeling language capabilities (ie SysML).
 - They enable creation of a collection of models (both visual and data) to communicate the architecture to the stakeholders
- Many modeling tools were reviewed during my project but primary focus was placed on the following subset:
 - System Architect*
 - Rhapsody*
 - Slate
 - TAU*
 - TSA
- Implementation of systems architecture modeling tools often vary.
 - A sampling of modeling tools used on program was taken and their rationale for selection was determined.

Approach & Methodology

● Approach –

- Identify stakeholder concerns and challenges in regards to architecture modeling
- Define value based on top-level requirements
- Review architecture frameworks and modeling languages inclusion of SA products
- Determine the value of SysML
- Evaluate modeling languages & tools ability to meet top-level requirements
- Identify any other implications

● Data Collection –

- Survey stakeholders in the organization regarding concerns and challenges
- Survey of stakeholders top-level needs/requirements
- Identify architecture frameworks and modeling languages inclusion of SA products
- Collected data on architecture modeling tool use and evaluated the ability of a subset of modeling tools to meet top-level requirements
- Collected data on SysML unique functionality and its value

DATA ANALYSES & RESULTS

- Survey Stakeholder Concerns
- Value Definition Top-Level Requirements
- Modeling Language & Framework Evaluation
- SysML Evaluation
- Value of SysML
- Modeling Tool Evaluation
- Other Implications

Survey Stakeholder Concerns

- Stakeholders include (but are not limited to):

- External Customers
- Internal Customers
 - Chief Engineer
 - Systems Engineers
 - Software Engineers
 - Others

- Some key concerns include:

- Does the architecture have an effective framework?
- Does the architecture sufficiently inform hardware and software design?
- Is the architecture definition sufficient to achieve integration across teams?
- What is the process and mechanism to flow requirements discovery in the architecture analysis into corresponding specifications?

Stakeholder Concern

- Does the architecture have an effective framework?
- Do requirements and systems and specialty engineering properly inform architecture?
- Does the architecture have allocated functional and performance requirements?
- Is the architectural approach supported and validated by integrated analysis? As definitions or changes are made to requirements, allocations or performance, are the architecture impacts understood? Can they be modeled?
- Does architecture sufficiently inform hardware and software design?
- Does the architecture capture the required operational information?
- Is the architecture validated against scenarios?
- Is the architecture definition sufficient to achieve integration across development teams?
- How does the Architecture Model incorporate traceability between System Views and Technical Views? TV-1 Expectation: The TV-1 references the technical standards that apply to the architecture and how they need to be, or have been, implemented. These standards
- What are the operational and systems tradeoffs with alternatives?
- Will the architecture account for KPP requirements?
- How do the model items electronically link to requirement origin (ORD para, spec para, analysis paper, flow down)?
- Does the model link TPMs, and KPPs in a way that shows requirement gaps?
- Does the model capture trade study attributes and reference documents?
- How will an analyst be able to use the model as the repository for technical specification and design capture?
- How does the model support growth objectives for spiral development?
- How does the model show support for expandability, evolvability & extensibility?
- How does the model show non-communication doctrinal policies (rules of engagement, weapon target pairing...)?
- How does the architecture model capture non-performance requirements for safety, security, weights, power, supportability, survivability, RAM-T and cost allocations?
- How does the architecture capture product interfaces, behavior and resources required at all system levels?
- Are Mission critical rules represented in the architectural driving use cases? Does the model identify operational level strategies that mitigate mission critical failures? Example: operators rebooting system.
- Do the architectural driving use cases include Operational use cases that reflect constraints on information flows due to associated system owner data security policies, and include definition of those policies?
- Do the architecturally driving use cases and scenarios described in the architecture model identify system level strategies, which mitigate mission critical failures captured within architecturally driving use cases and scenarios?
- Does the architecture model include definition of operational names (e.g. roles and personnel) and the rules associated with their management (e.g. transfer of command in presence of damage, network connectivity, etc)?
- What is the process and mechanism to trace all levels of system architecture analysis to corresponding specifications?
- What is the process and mechanism to flow requirements discovery in the architecture analysis into corresponding specifications?
- Is the strategic and tactical control of the SoS or system(s) represented in the model and presented to a stakeholder so that it is comprehensible in a single diagram?
- What are the architecture requirements for filtering to provide a Layered/Tailored Common Operational Picture that has access constraints for training exercise components?

Value Definition

Top-Level Requirements

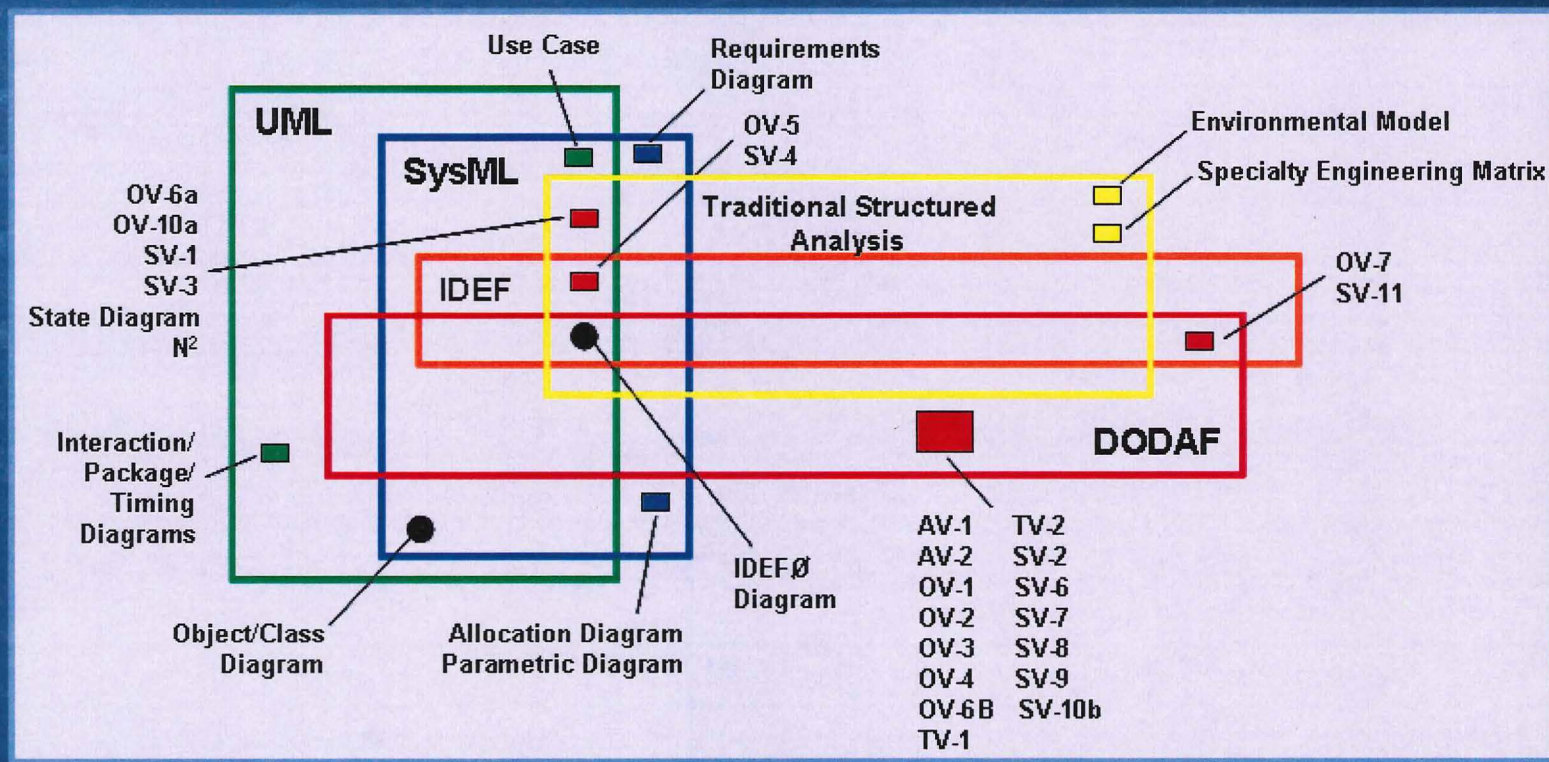
- Surveyed top-level Requirements
- Asked stakeholders to identify:
 - What makes the tool of value to them
 - What are their must-have requirements
- A subset of items identified (those related to systems engineering) became the bases for tool evaluations

Value Definition/Top Level Requirements	
1. Support for UML 2.0.	Must have
2. Integration with configuration management (ClearCase, CM Synergy, CVS, VSS)	Must have
3. Ability to import Rose models	Must have
4. Code generation for C++.	Must have
5. Multi-site support. (distributed development)	Must have
6. Platform support for Windows XP	Must have
7. Documented API to support scripting	Must have
8. DoDAF View support.	Highly desired
9. Integration with DOORS	Highly desired
10. Code generation for Java.	Highly desired
11. SysML part of roadmap	Highly desired
12. Model/code synchronization and/or Code visualization	Highly desired
13. CADM Compliance -	Desired
14. Executable Modeling	Desired
15. Comprehensive report generation	Desired
16. XMI Data Interchange Support	Desired
17. Vendor responsiveness	Desired

Top-level requirements served as the basis for defining value

Modeling Language & Frameworks Evaluation

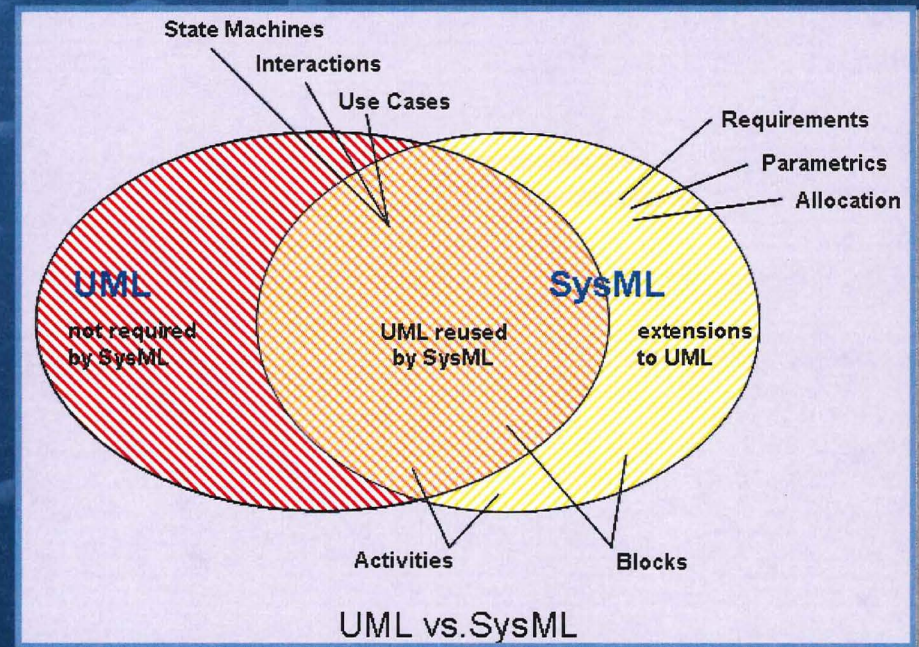
- The figure below depicts system architecture products in reference to modeling language and architecture frameworks



No one modeling language or framework incorporates everything needed by systems engineering

Value of SysML

- SysML was designed with “real” systems in mind.
 - Where as UML is software oriented
- Supports the verification and analysis of complex systems by reusing existing modeling language and extended it.
 - *Allocations*: relation between behavior and structure
 - *Parametric Diagram*: to integrate with engineering analysis models
 - *Requirements Diagram*: to model requirements traceability and relationship to design models



SysML improves the ability to exchange systems engineering information

SysML Evaluation

- Surveyed SysML users to determine value
- Feedback on SysML unique features was as follows:

Requirements	<ul style="list-style-type: none">● Explicitly lay out requirements and consequences
Views & Viewpoints	<ul style="list-style-type: none">● Can separate requirements and model views based on stakeholder concerns
Structures	<ul style="list-style-type: none">● Ability to model structure to verify requirements<ul style="list-style-type: none">▪ Can search for requirements that aren't verified▪ Can search model components that aren't justified
Behavior	<ul style="list-style-type: none">● Dashed line for activity flow is more aesthetically pleasing<ul style="list-style-type: none">▪ Vs. UML solid line

- Particularly effective in specifying requirements, systems structure, functional behavior and allocations during specification and design phases of systems engineering.

Modeling Tool Evaluation

- A subset of modeling tools were evaluated against top-level requirements

Criteria Measured	Rhapsody	Tau	RSA
Support for UML 2.0.	3	3	3
Integration with configuration management (ClearCase, CM Synergy, CVS, VSS)	3	3	1
Ability to import Rose models	1	1	1
Code generation for C++.	3	1	1
DoDAF View support.	3	3	0
Integration with DOORS	3	9	3
SysML part of roadmap	3	1	1
Executable Modeling	3	3	0
Comprehensive report generation	3	1	3
Vendor responsiveness	3	1	1
User Interface	3	1	3
Reliability	3	1	3
User Documentation	3	3	3

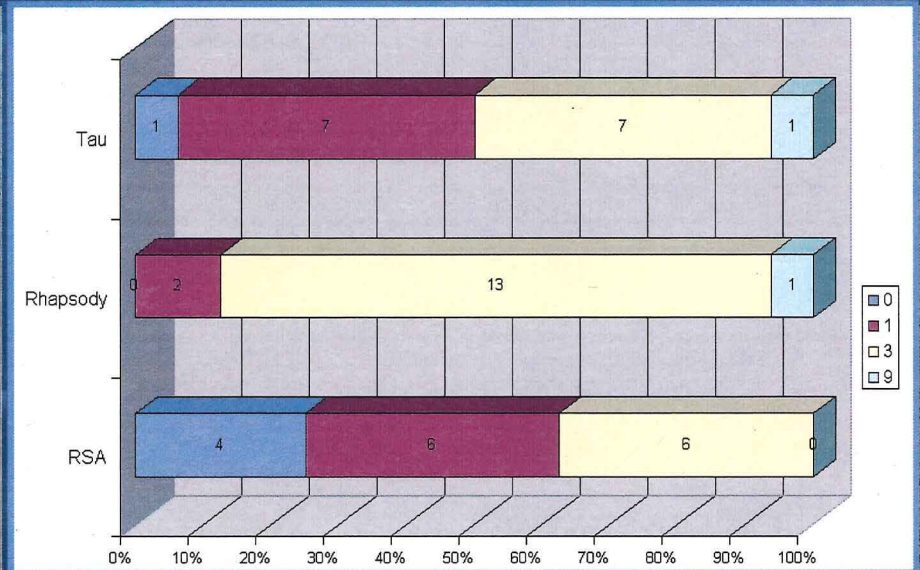
Score:

0 - tool has no support for this criteria

1 - poor or limited implementation

3 - usable / workable

9 - strength of this tool



Rhapsody clearly met more of stakeholders' top-level requirements therefore adds more value

Other Implications

Quality:

- Quality of an architecture description refers to its understandability, consistency, completeness, and analyzability.
- The proposed system must be described with the accuracy and clarity that the stakeholders' need to understand how the architecture addresses their concerns.

Lean:

- There are numerous system architecture tools currently in use
 - With all these different tools there is certainly a lot of waste
 - There is little tool standardization among programs
- To have a more Lean approach to model driven architecture and tools we need to answer the following:
 - Is there some way we can encourage better integration across these models that will allow us to produce better systems?
 - Is there a growth path through which all programs can move on the road to architecture modeling with one tool?

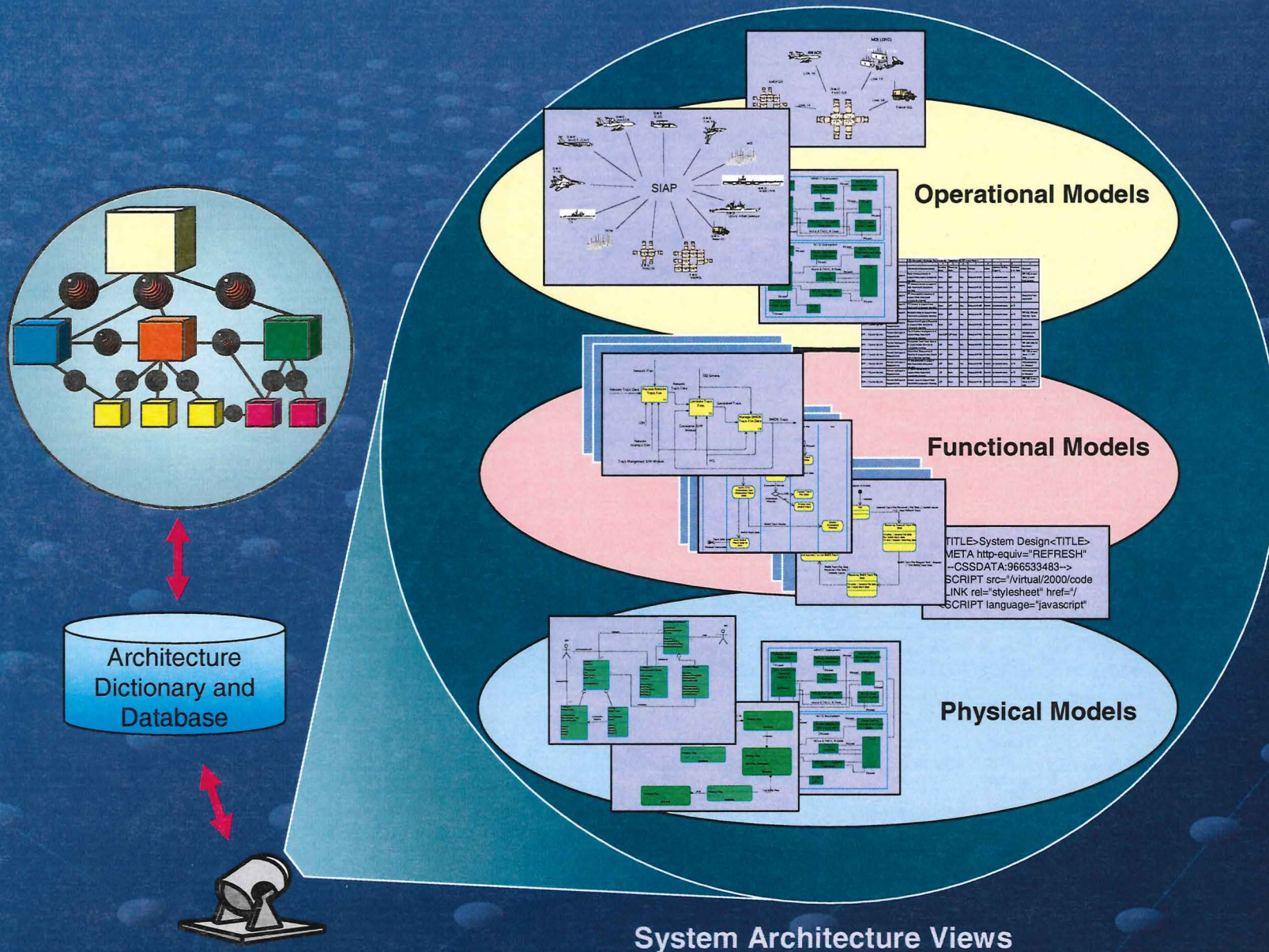
Risk

- If you select an inadequate tool you will be at risk of increasing the time and cost it takes for system architecture development as well as decreasing your ability to communicate and describe the proposed system to the stakeholders

Ethics

- No ethical issues were uncovered during the research

Model Driven Architecture Descriptions



System Architecture Conclusions

- Systems architectures describe the structure of systems, their activity elements, their relationships, and the principles and guidelines governing their design and evolution.
- Architecture models enable, guide, and help assess the construction of systems as they are progressively developed and refined.
- There is a tremendous increase in the number of elements in today's systems and the connections between them.
- Biggest challenge is how to develop successful architectures capable of communicating the higher level of complexity of today's systems.

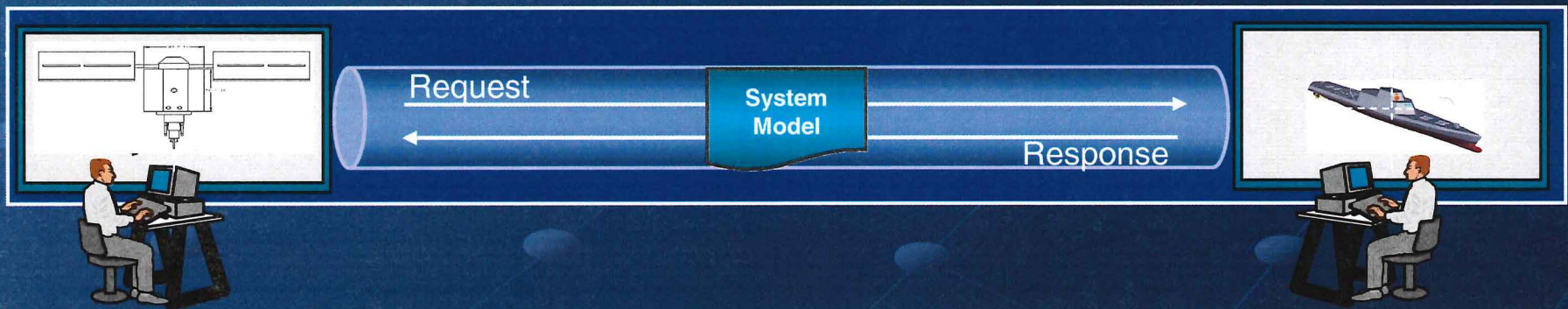
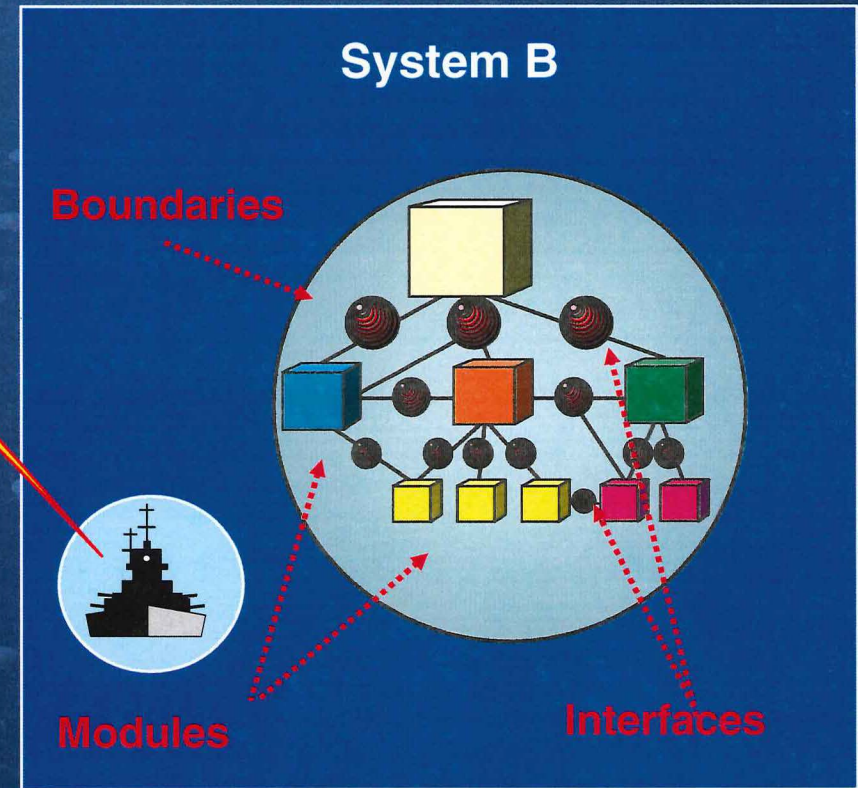
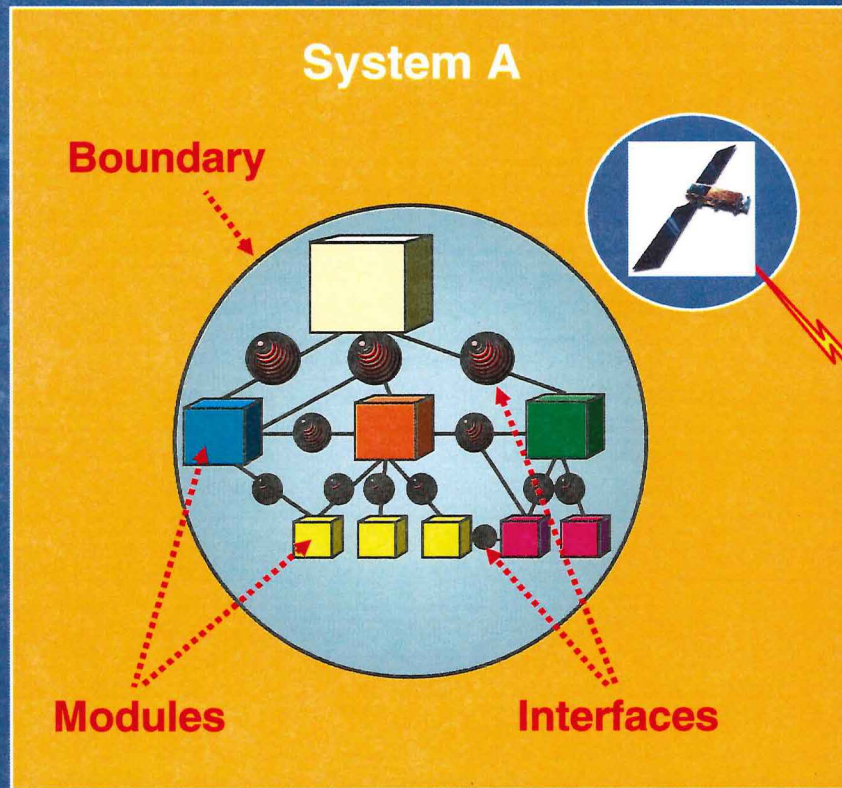
Architecture Modeling Conclusions

- SysML's features were determined to be of value.
 - SysML supports the verification and analysis of complex systems regardless of the domain (i.e.spacecraft environment).
 - SysML improves communication with the customer and throughout the program.
 - The amount of value SysML adds is impacted by the system architecture modeling tool selected.
- There is no one tool or modeling language that fits all of the system architecting needs.
 - But modeling tools and languages are still evolving.
- Selecting one preferred system architecture modeling tool allows for:
 - Re-use of software for system architectures developed from former programs.
 - It will also reduce the waste each program spends paying for system architecting modeling tools development and/or tailoring.

SysML adds value - - the extent is dependant on the modeling tool selected

Focus should be on implementing a single robust modeling technique

Collaborative System Architecture Modeling





Questions?