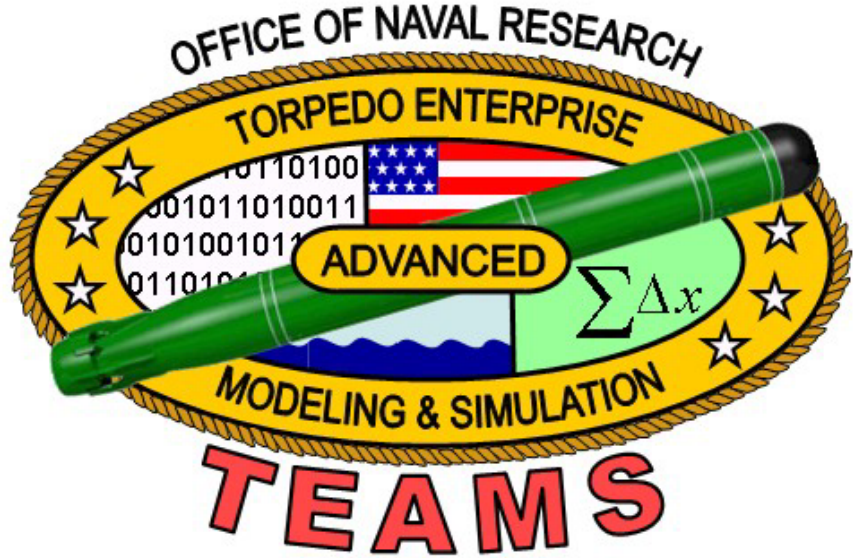
TEAMS and SysML: Proof of Concept Status

Presented to:
Modeling and Simulation Committee of the National Defense Industrial Association Systems Engineering Division

Presented by:
Thomas Haley
Naval Undersea Warfare Center

David Diederich
Applied Research Laboratory
Penn State University

17 April 2007



Note: Slides have been updated to incorporate comments from the original presentation -tbh

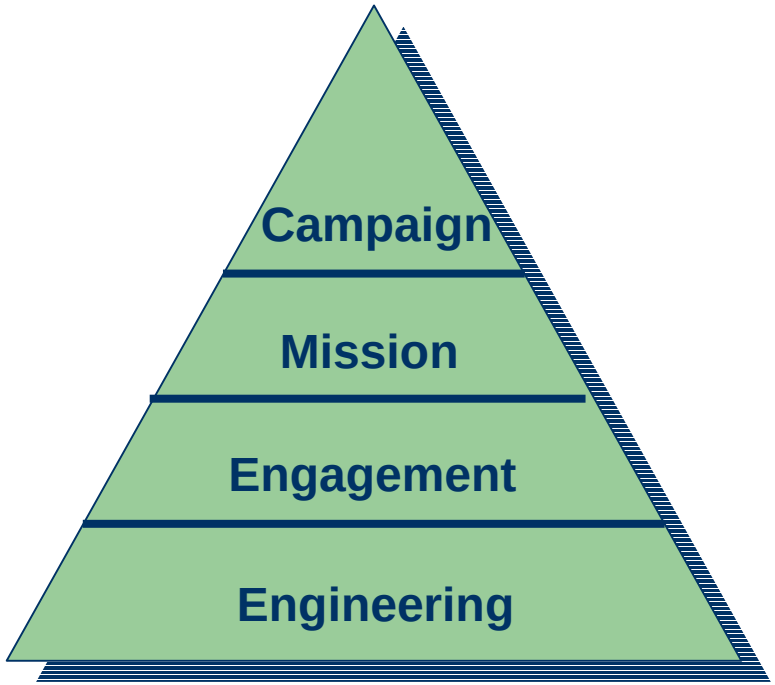
Outline

- SysML Case Study Motivation
- TEAMS Background
- TEAMS SysML Proof of Concept
- Lessons Learned
- TEAMS Perspective: SysML Pros and Cons
- Acknowledgements

Motivation: Feasibility of Open Standards

- Funded by Office of Secretary of Defense, Systems and Software Engineering
- Determine if open standards can be used to describe:
  - System of systems (SoS) architectures based on computer models
  - System components as elements of composable distributed simulations
- Determine whether SysML models can be used in conjunction with performance simulation models

Background: TEAMS Simulation Scope

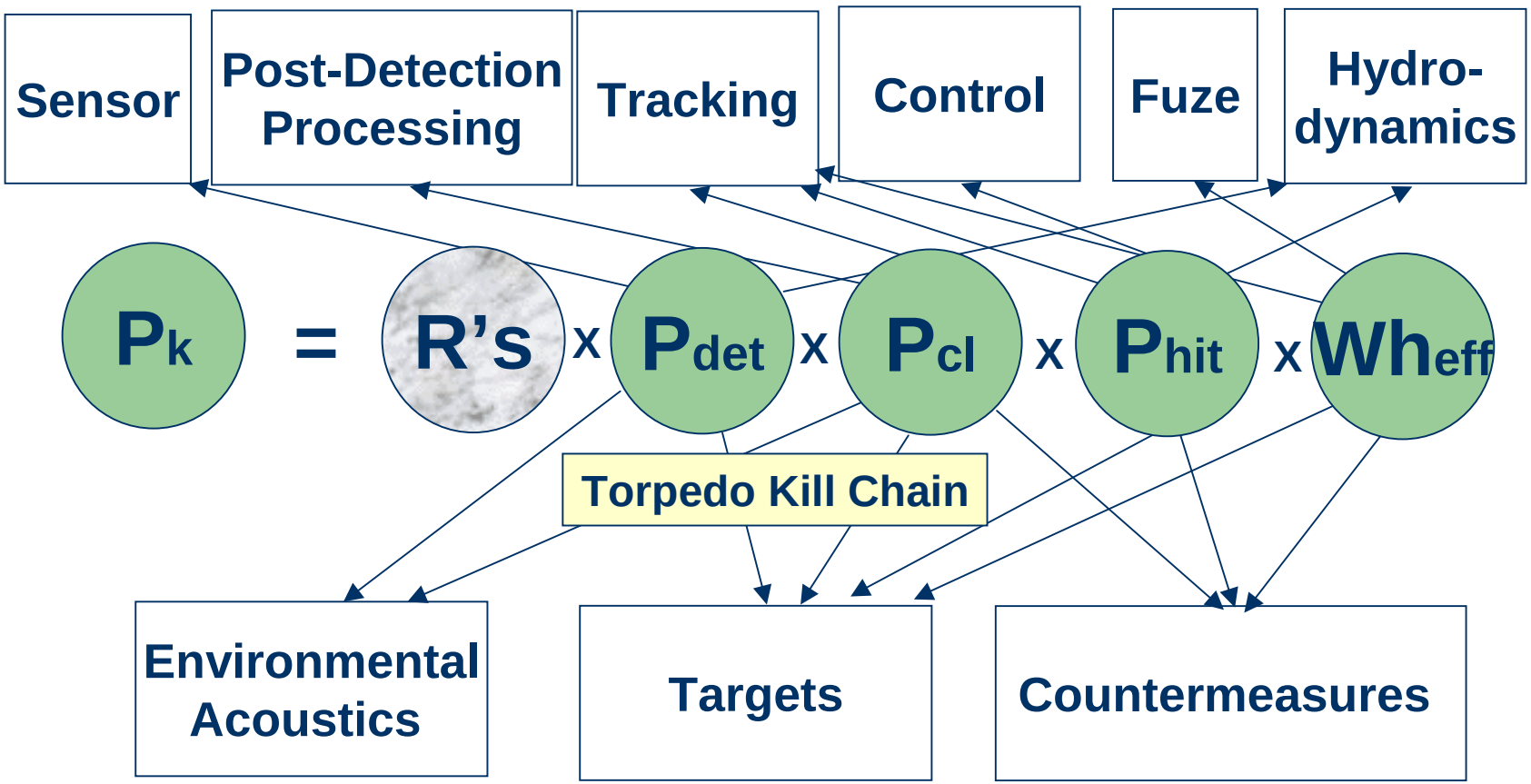


TEAMS Emphasis: "Launch-to-Hit" Analysis

Military M&S Resolution Levels

Background: High-Level M&S Requirements

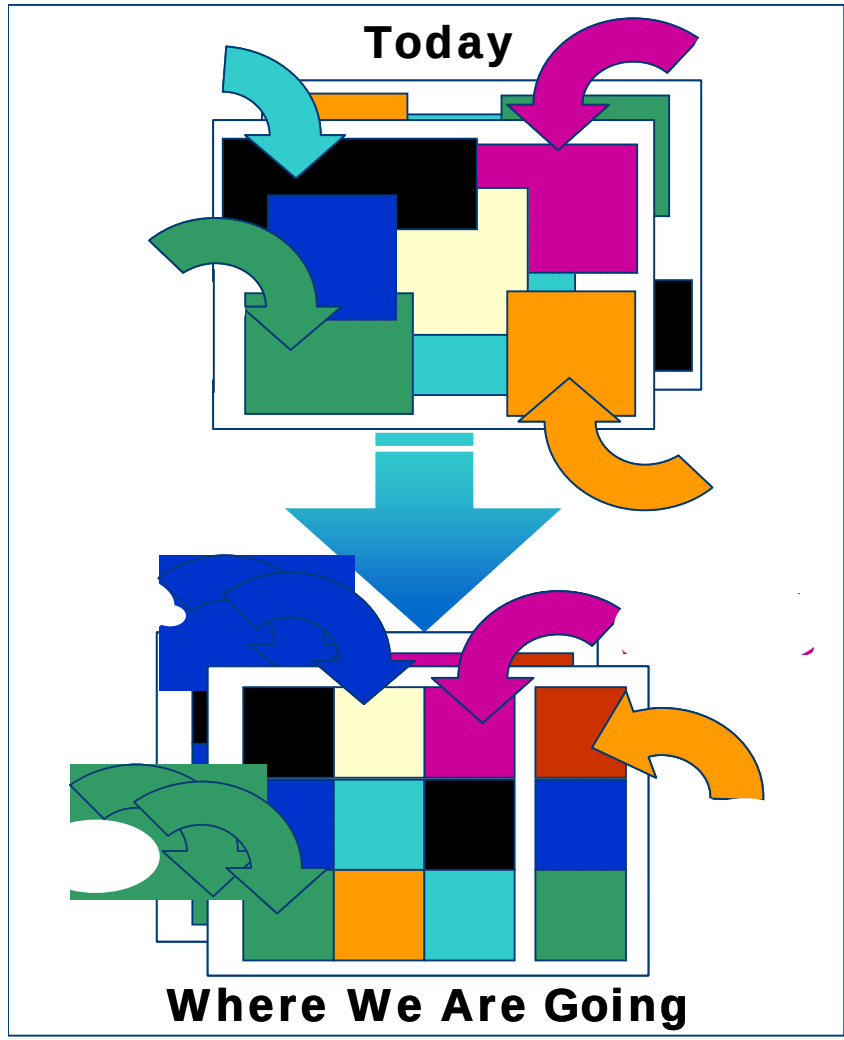
Torpedo M&S Components



Other "Stimulus" M&S Components

TEAMS Background

- **Problem: Modeling & Simulation Business "Model" Obsolete**
  - Monolithic
  - Stove pipes
  - Single developers
  - No communication
- **Solution: Foster Collaborative M&S Development Environment**
  - Standardize M&S architecture framework and component models
  - Reduce the technology development timeline
  - Increase model content, implementation efficiency and reuse
  - Reduce cost



Overall TEAMS Goals

- Modeling and Simulation Community Collaboration
- Standardized architecture framework
  - Conceptual reference model
  - Model-based requirements specifications
- Standardized reference model interfaces
  - Interchangeable & composible components
  - Extendable to other applications (e.g., XML schema)
  - Semantically described (e.g., OWL ontology)
- Document standards and requirements
- Cost effective process to achieve interoperability and composability
- Business model for future cross-organization M&S funded efforts

Organizations Looking to TEAMS

THE Open GROUP

International organization, developers of TOGAF architectural framework

- Wants TEAMS as test case for TOGAF 8.1.1 and 9.0

- Interest in using TEAMS to test synergy between DoDAF and TOGAF frameworks

- Wants TEAMS for its process to incorporate Ontologies (relationships of components)

OMG
OBJECT MANAGEMENT GROUP

International organization, developers of several business communications standards

- Wants TEAMS as test case for their TOGAF/ Model Driven Architecture (MDA) synergy effort

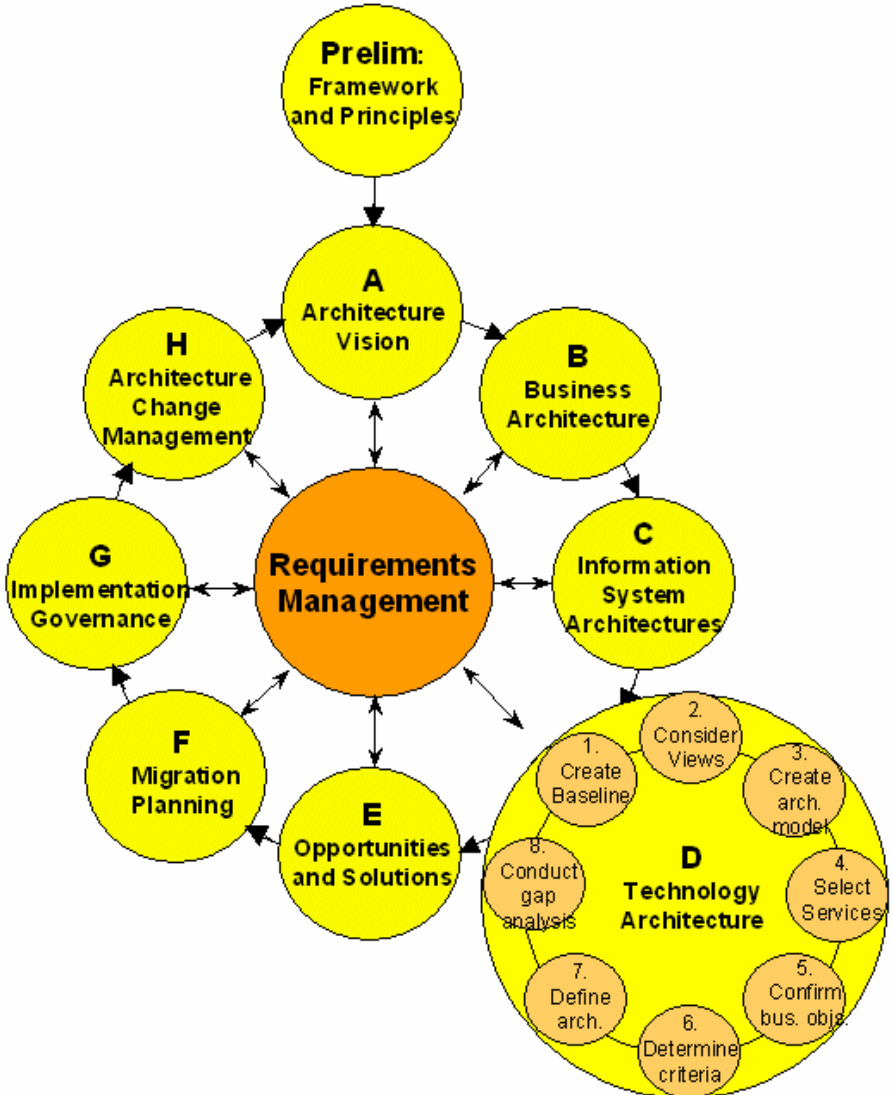
OPEN SYSTEMS
OS
JOINT TASK FORCE

The Open Systems Joint Task Force of the Office of Secretary of Defense (OSD)

- Wants to convert TEAMS UML artifacts to the newly approved SysML standard to demonstrate utility of the new standard

TEAMS is quickly yielding *highly visible* and *transitionable* results.

The Process: TOGAF ADM



The Open Group:
IT Consortium
Offers Consortia Services

TOGAF:
The Open Group
Architecture Framework

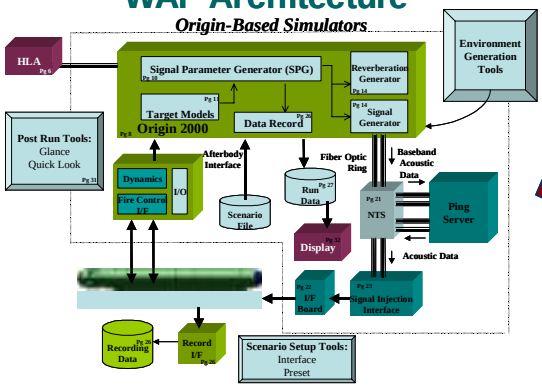
ADM:
Architecture
Development Method

A Architecture Vision

Baseline Technology Architecture

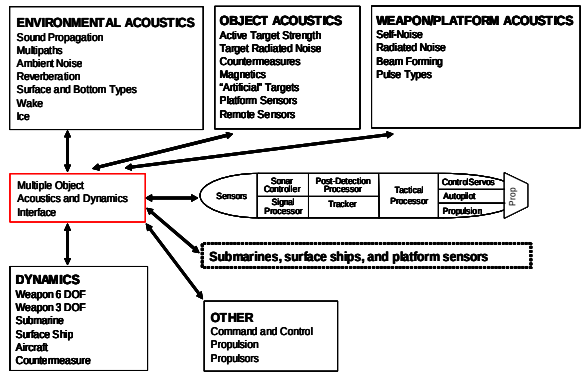
Weapons Analysis Facility (WAF)

WAF Architecture



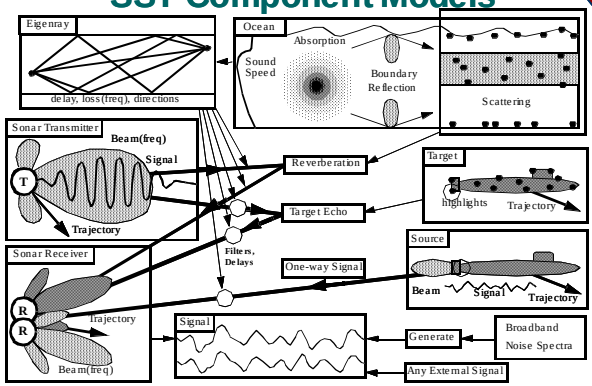
Technology Requirements Model (TRM)

TRM's CARLEE

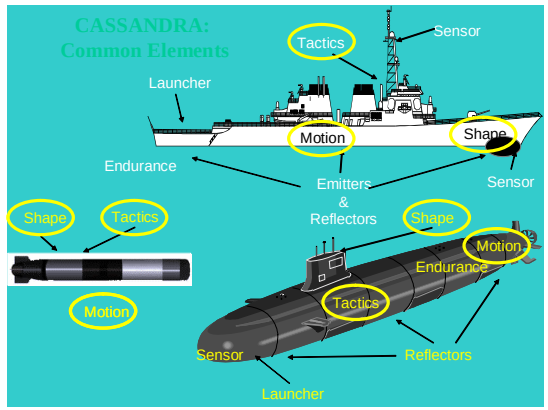


Conceptual Reference Model

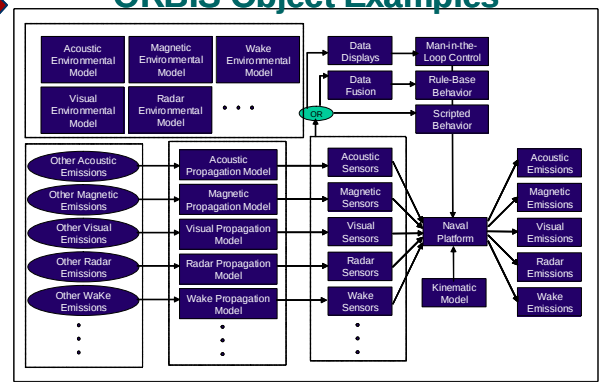
SST Component Models



CASSANDRA Common Elements

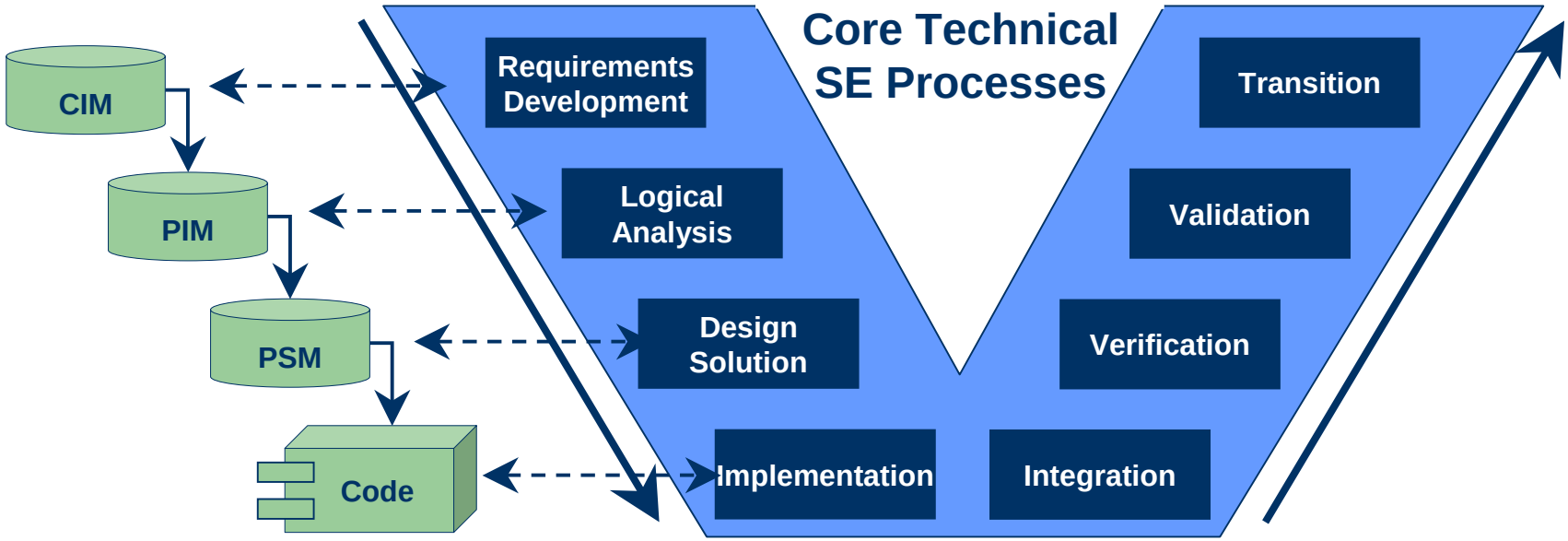


ORBIS Object Examples



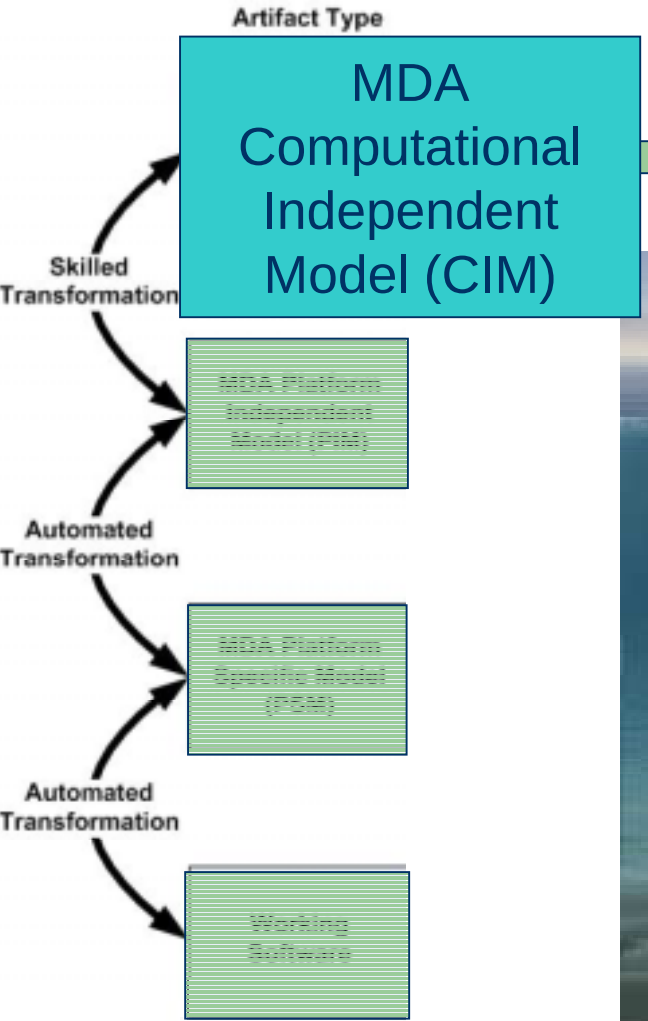
Sonar System Toolset (SST)

Using MDA in SE Context

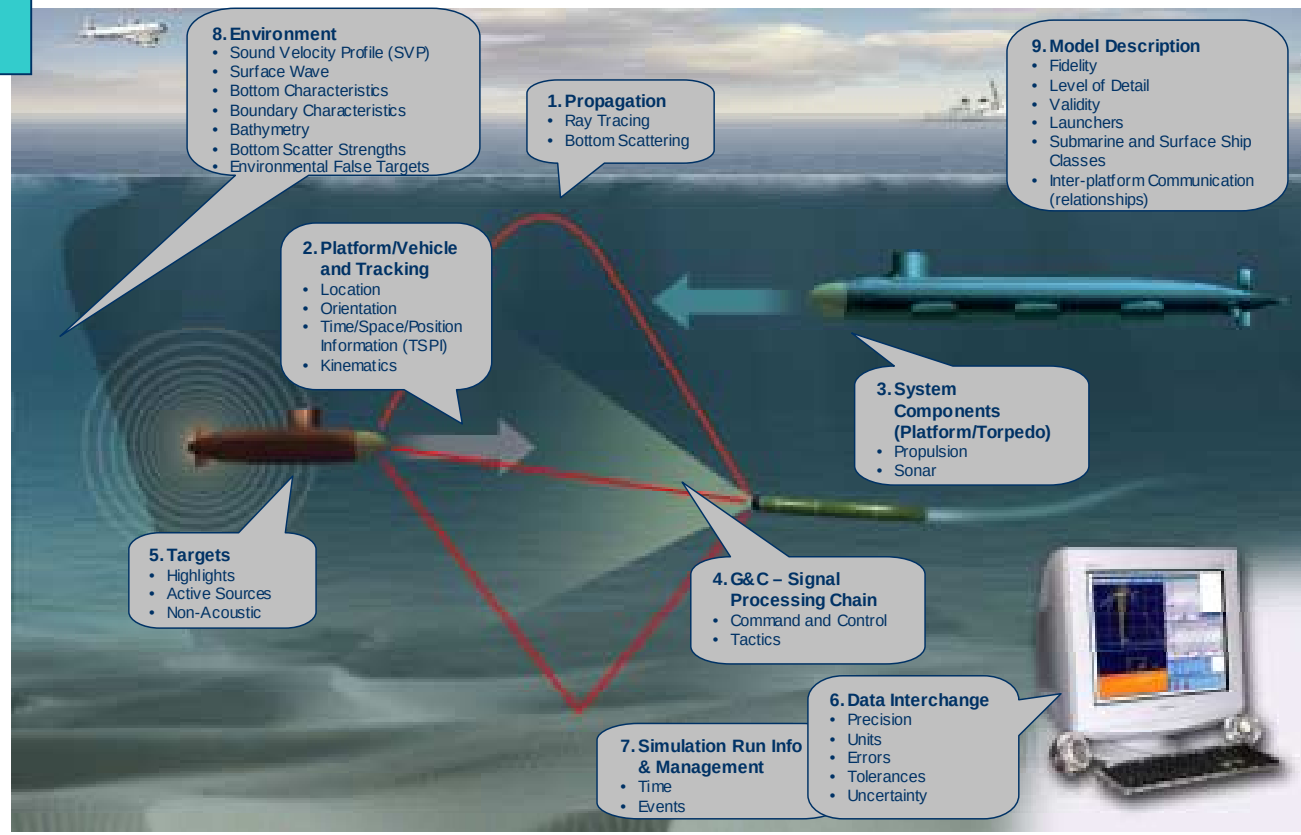


The implementation (code) for technology selected by the developer

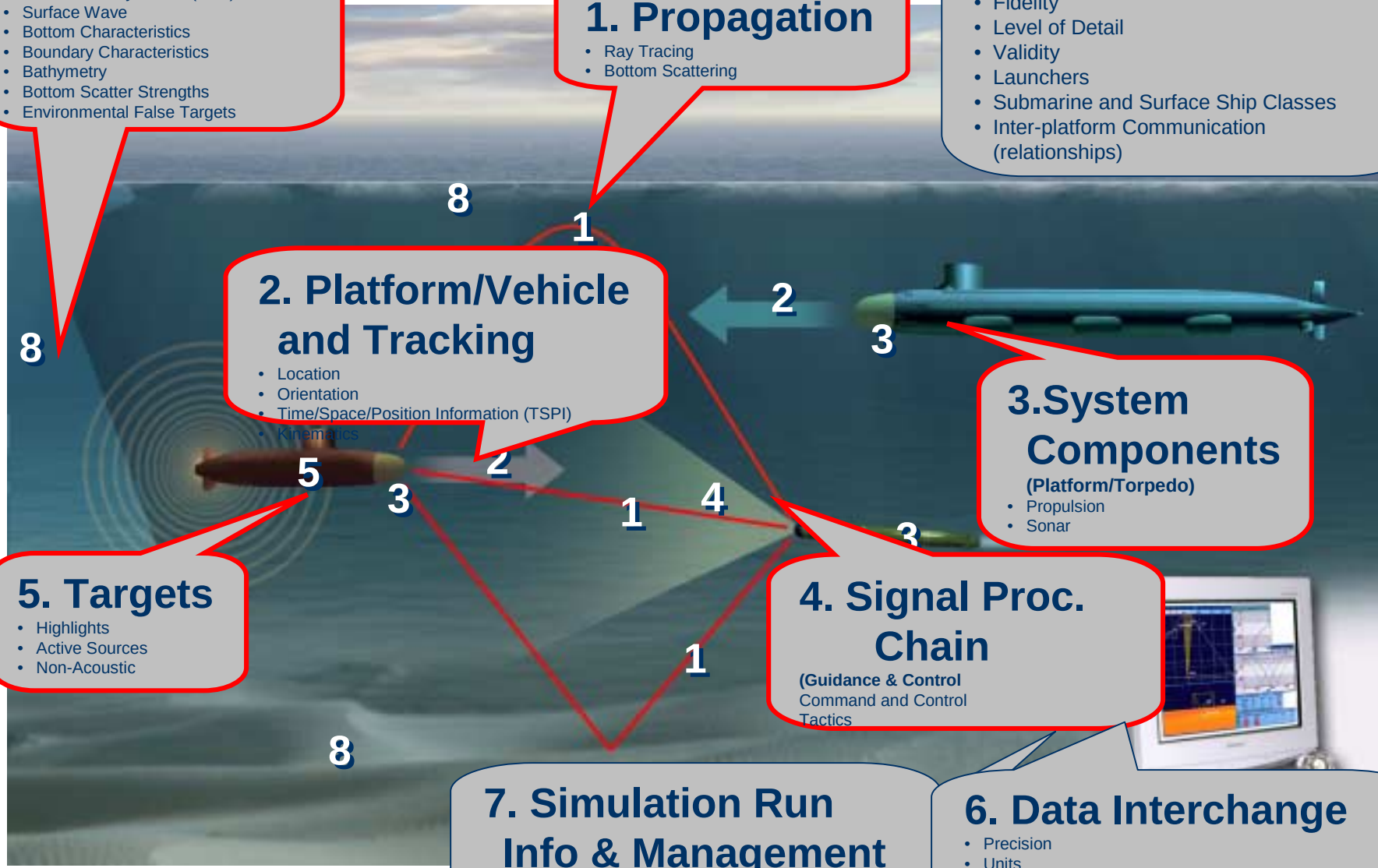
The Method:
Model Driven Architecture (MDA)



TEAMS Conceptual Reference Model



TEAMS Conceptual Reference Model



8. Environment

- Sound Velocity Profile (SVP)
- Surface Wave
- Bottom Characteristics
- Boundary Characteristics
- Bathymetry
- Bottom Scatter Strengths
- Environmental False Targets

1. Propagation

- Ray Tracing
- Bottom Scattering

9. Model Description

- Fidelity
- Level of Detail
- Validity
- Launchers
- Submarine and Surface Ship Classes
- Inter-platform Communication (relationships)

2. Platform/Vehicle and Tracking

- Location
- Orientation
- Time/Space/Position Information (TSPI)
- Kinematics

3.System Components

(Platform/Torpedo)

- Propulsion
- Sonar

5. Targets

- Highlights
- Active Sources
- Non-Acoustic

4. Signal Proc. Chain

(Guidance & Control
Command and Control
Tactics

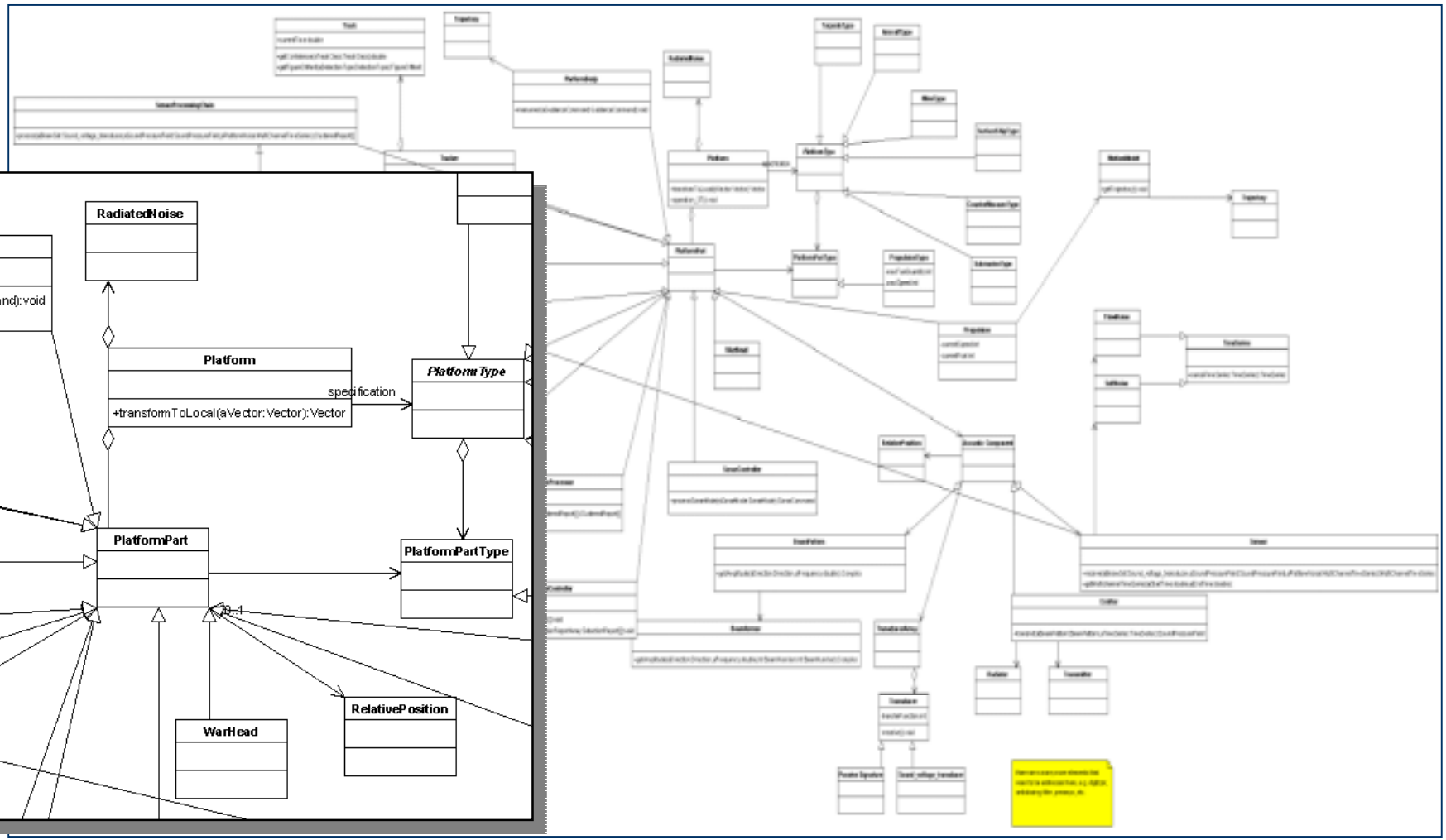
7. Simulation Run Info & Management

- Time
- Events

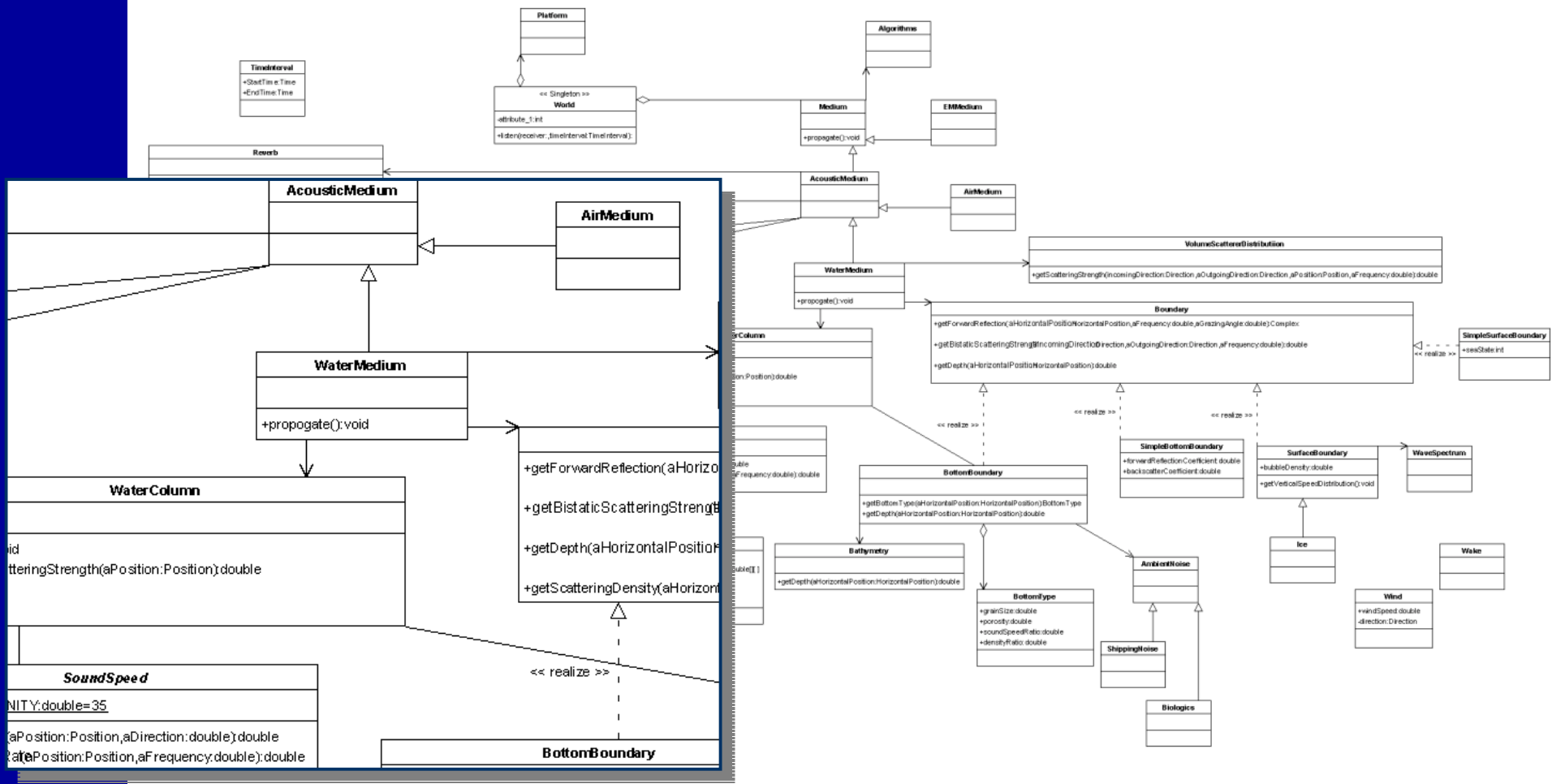
6. Data Interchange

- Precision
- Units
- Errors
- Tolerances
- Uncertainty

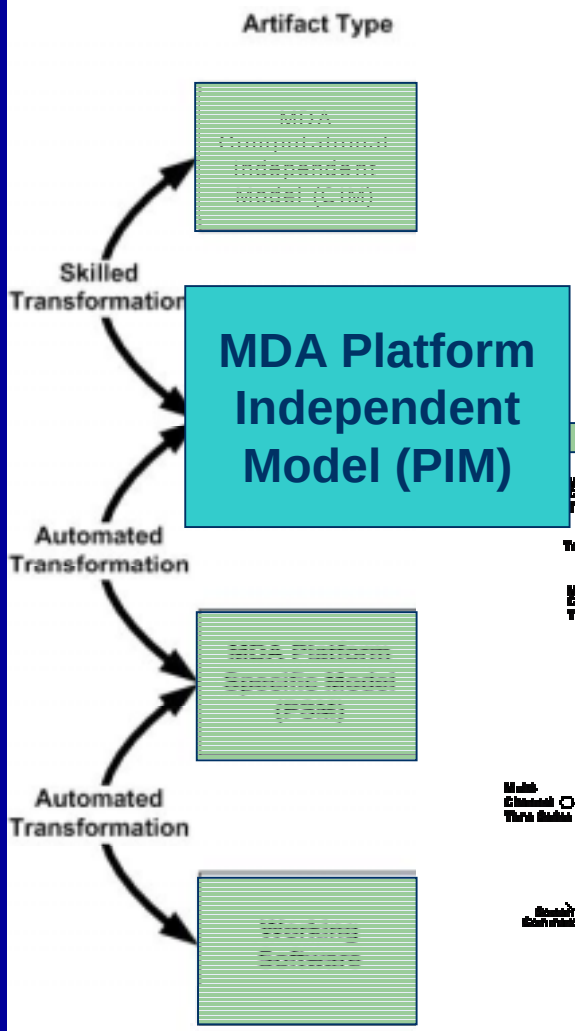
Platform Conceptual Level Diagram



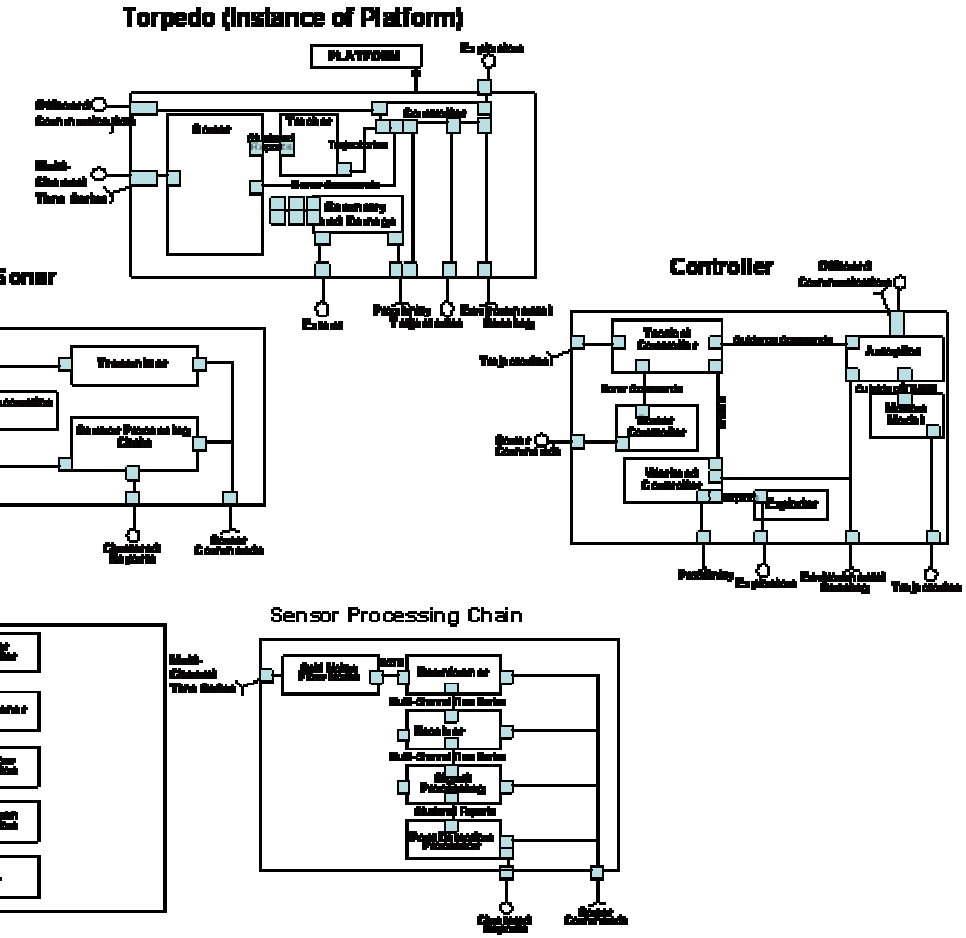
Environment Conceptual Level Diagram



The Method: Model Driven Architecture (MDA)



TEAMS UML Component Diagrams (Now Represented in SysML)



TEAMS PSM: Implementation Planning

In-situ Environmental Data via Web Services

NAVOCEANO
SIPRNET Web Site

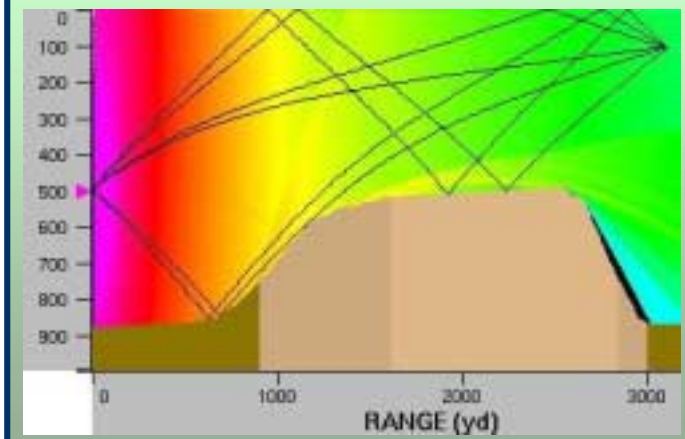
Jackson Bottom Model via CORBA

Applied Physics Lab
University of Washington

Fey Rey Propagation Model via HLA*

Defense Modeling and
Simulation Office

TRM Propagation Tool 'PETE'



TEAMS Proof of Concept

- Port existing UML to SysML
  - Torpedo system components
  - Simulation environment
- Extend TEAMS SysML to include:
  - Requirements traceability
  - Parametrics and constraints
- Share experiences and lessons learned using SysML for architecture and component modeling

UML to SysML Approach

- Convert UML Class Diagrams to SysML Block Definition Diagrams (BDDs)
- Convert UML Component Diagrams to SysML Internal Block Diagrams (IBDs)
- Represent Behavior relationships between blocks as Activity Diagrams *(new!)*
- Capture Requirements Traceability *(new!)*
- Capture Parametric Relationships and Constraints *(new!)*

TEAMS Perspective: SysML Pros and Cons

Pros

- Requirements
  - Explicitly lay out requirements and consequences
- Views and Viewpoints
  - Can separate requirements and model views based on stakeholders concerns
- Structure
  - Ability for model structure to verify requirements
    - Can search for requirements that aren't verified
    - Can search for model components that aren't justified
  - Separation of structure from behavior
    - SysML BDDs vs. IBDs and Activities allow for clear separation
    - UML allows this, but easier to implement in SysML
- Behavior
  - Dashed line for activity flow is more aesthetically pleasing
    - vs. UML solid line

Cons

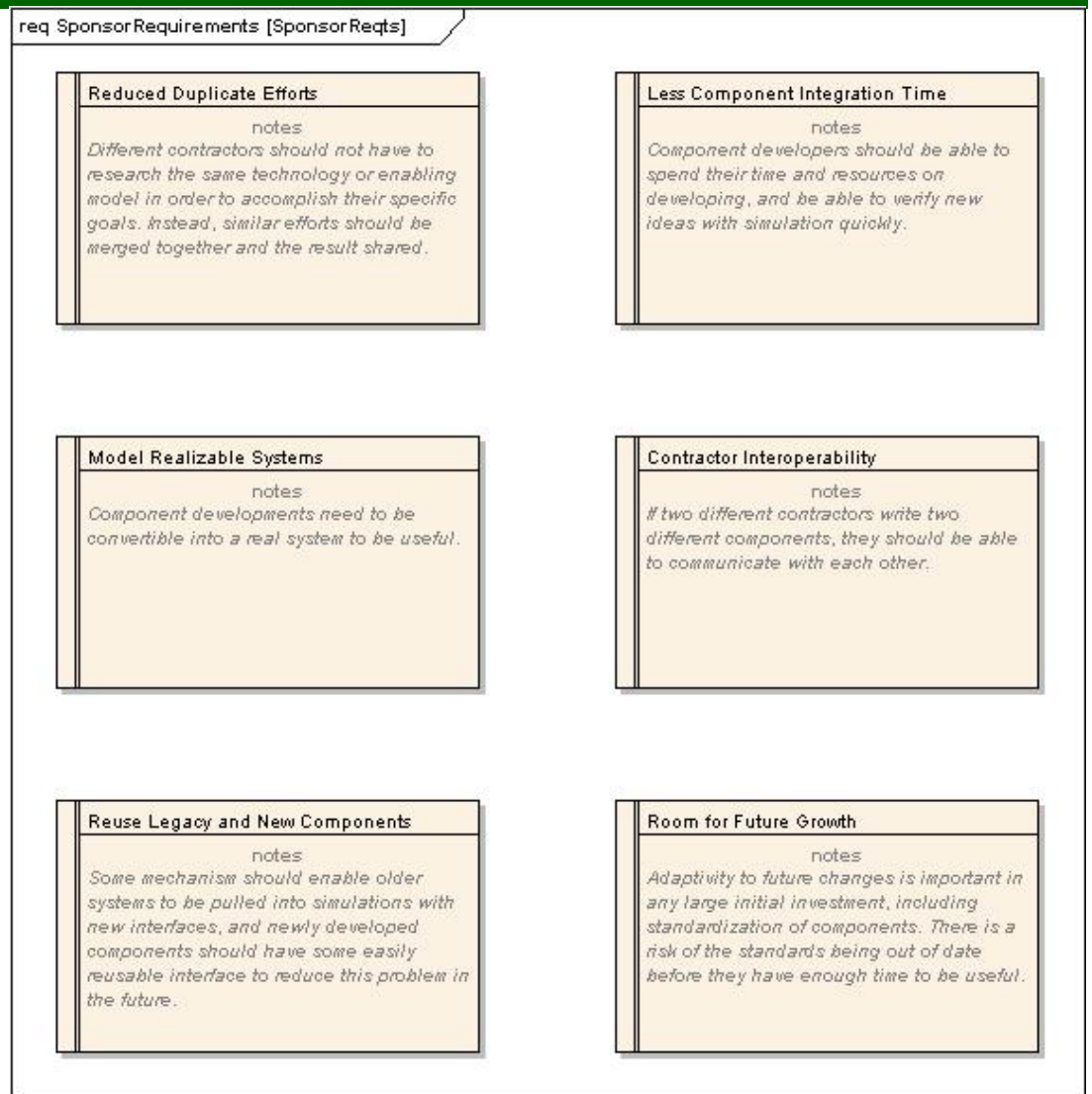
- Allocating CIM to PIM
  - Difficulty with abstract activities
    - Exit path dependent on logic within an activity is not accessible and can't be modeled
    - Not represented well in either UML or SysML – tactical controller example
- Implementing PIM
  - Not "direct" for some SysML features
    - Flow ports, continuous activities, parametric constraints involve more components than just themselves
    - Flows in "real systems" easier to represent
    - Flows in software modeling are open to interpretation
  - Requires additional documentation of model to bridge between SysML feature and executable code

TEAMS Perspective: SysML Pros

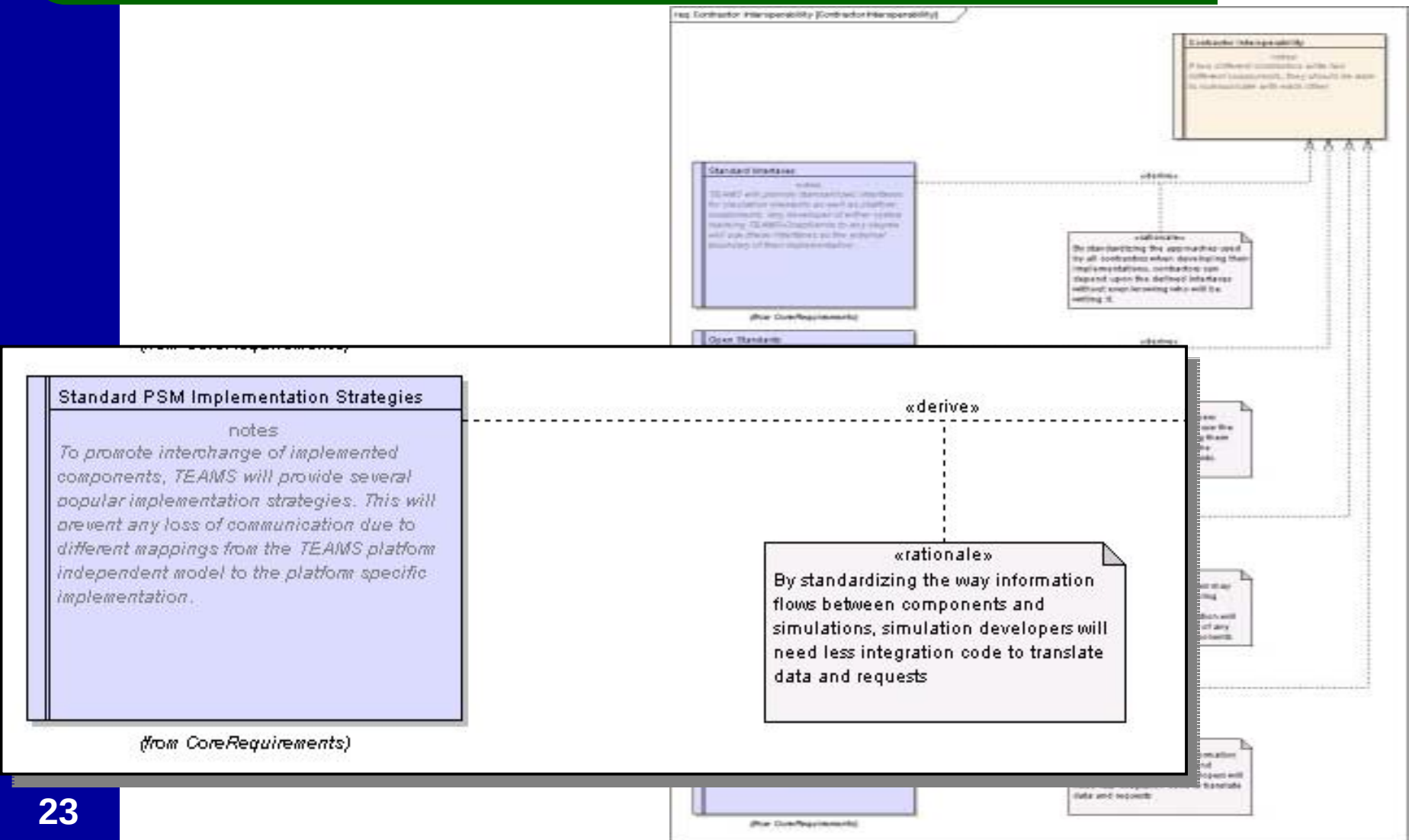
Pros

- **Requirements**
  - **Explicitly lay out requirements and consequences**
- **Views and Viewpoints**
  - **Can separate requirements and model views based on stakeholders concerns**
- **Structure**
  - **Ability for model structure to verify requirements**
    - **Can search for requirements that aren't verified**
    - **Can search for model components that aren't justified**
  - **Separation of structure from behavior**
    - **SysML BDDs vs. IBDs and Activities allow for clear separation**
    - **UML allows this, but easier to implement in SysML**
- **Behavior**
  - **Dashed line for activity flow is more aesthetically pleasing**
    - **vs. UML solid line**

Sponsor Requirements



Rationale for Deriving TEAMS Core Values from Sponsor Requirement(s)



Requirements Traceability: TEAMS Core Values

Standard Interfaces

notes

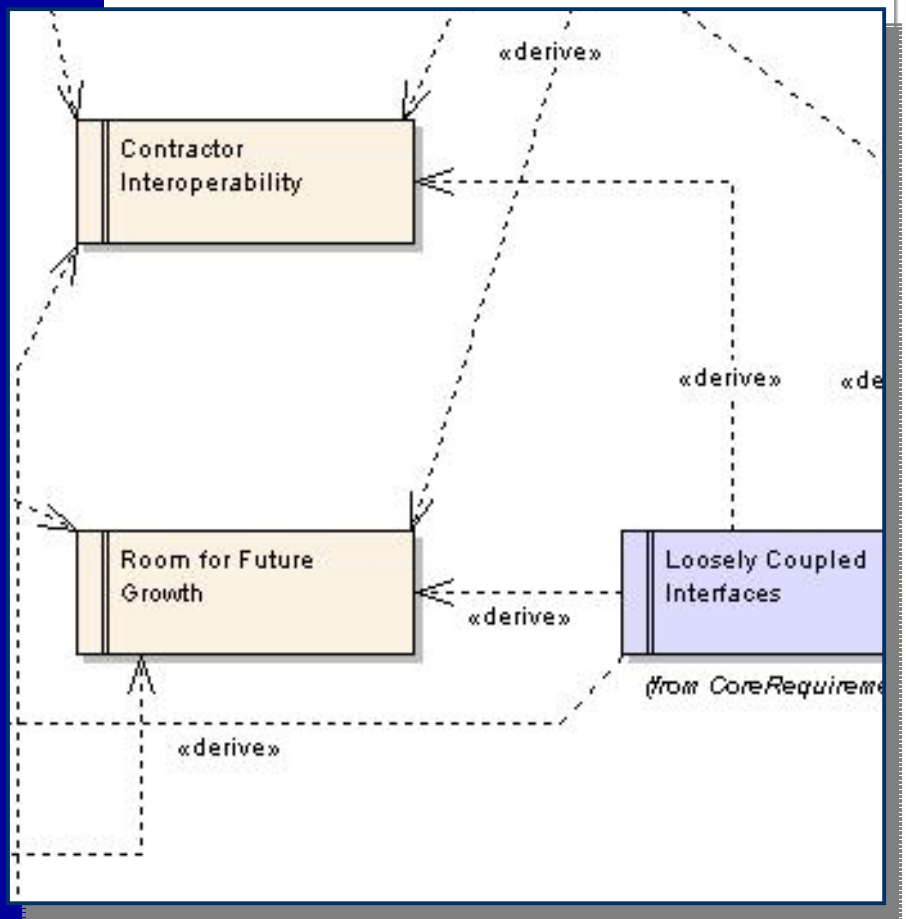
TEAMS will provide standardized interfaces for simulation elements as well as platform components. Any developer of either system claiming TEAMS-Compliance to any degree will use these interfaces as the external boundary of their implementation.

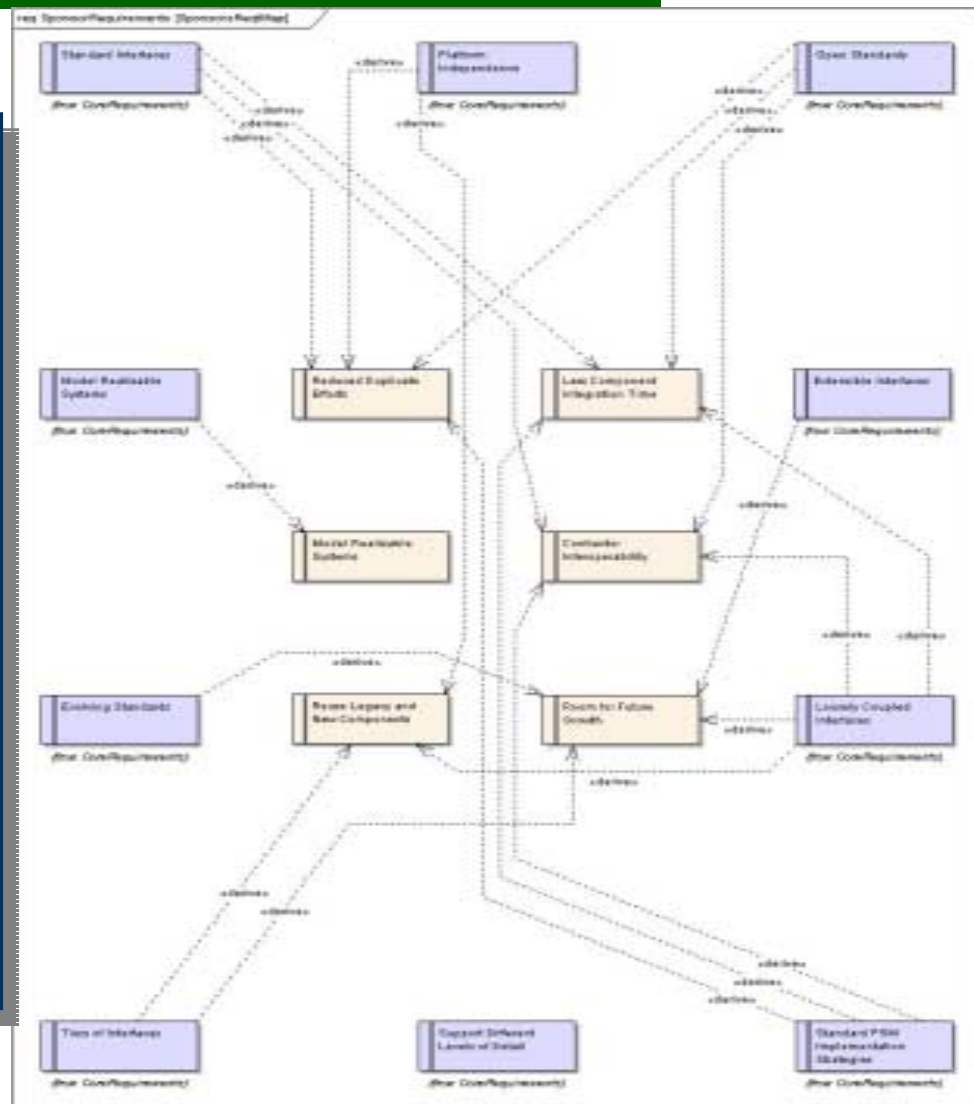
Model Realizable Systems

notes

The interfaces that appear in the TEAMS model will reflect actual systems in the real world. This includes designed systems as well as physical constraints placed by the environment.

Sponsor Requirements Mapped to TEAMS Core Values



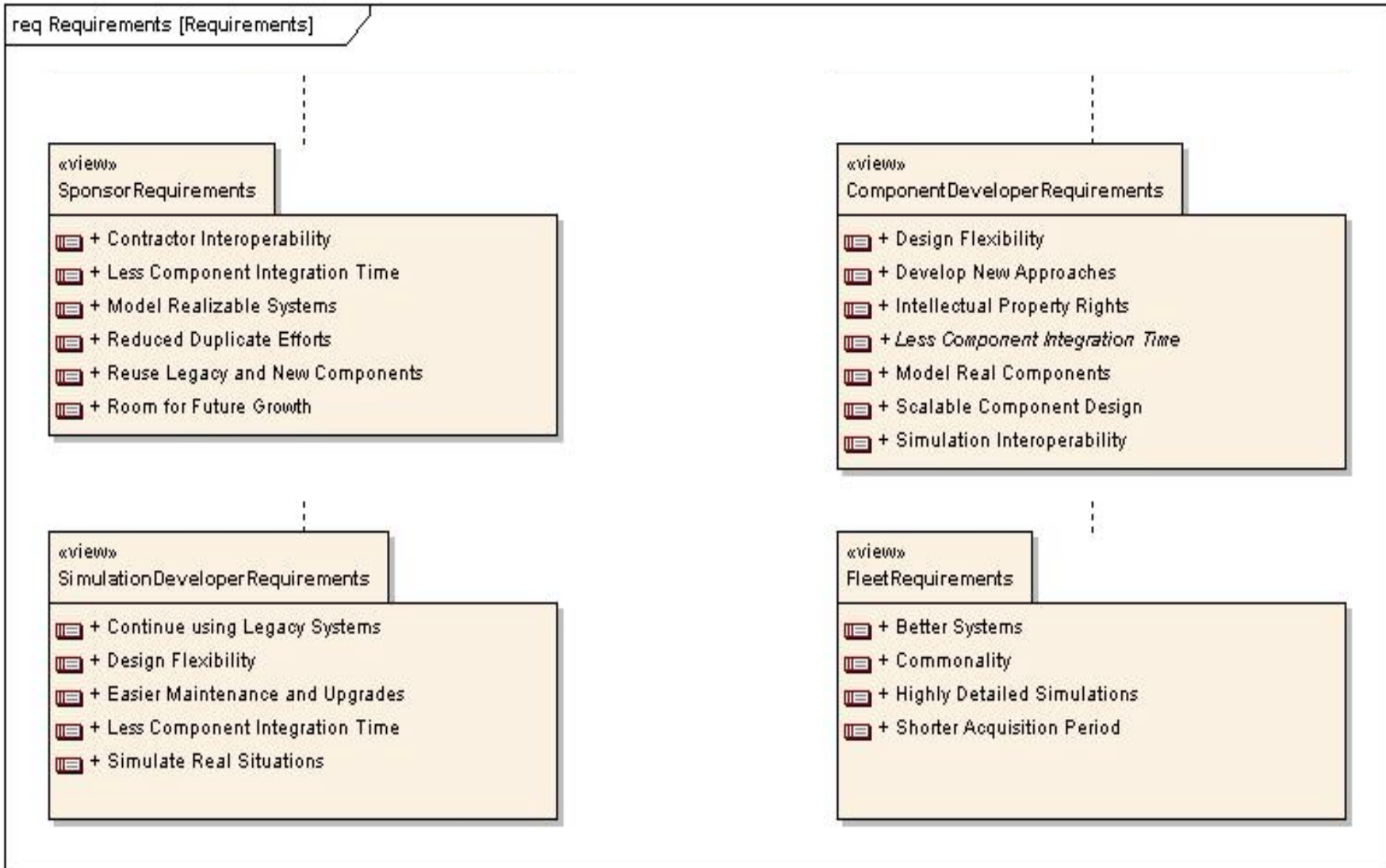


TEAMS Perspective: SysML Pros

Pros

- Requirements
  - Explicitly lay out requirements and consequences
- **Views and Viewpoints**
  - **Can separate requirements and model views based on stakeholders concerns**
- Structure
  - Ability for model structure to verify requirements
    - Can search for requirements that aren't verified
    - Can search for model components that aren't justified
  - Separation of structure from behavior
    - SysML BDDs vs. IBDs and Activities allow for clear separation
    - UML allows this, but easier to implement in SysML
- Behavior
  - Dashed line for activity flow is more aesthetically pleasing
    - vs. UML solid line

TEAMS
Stakeholder Requirements

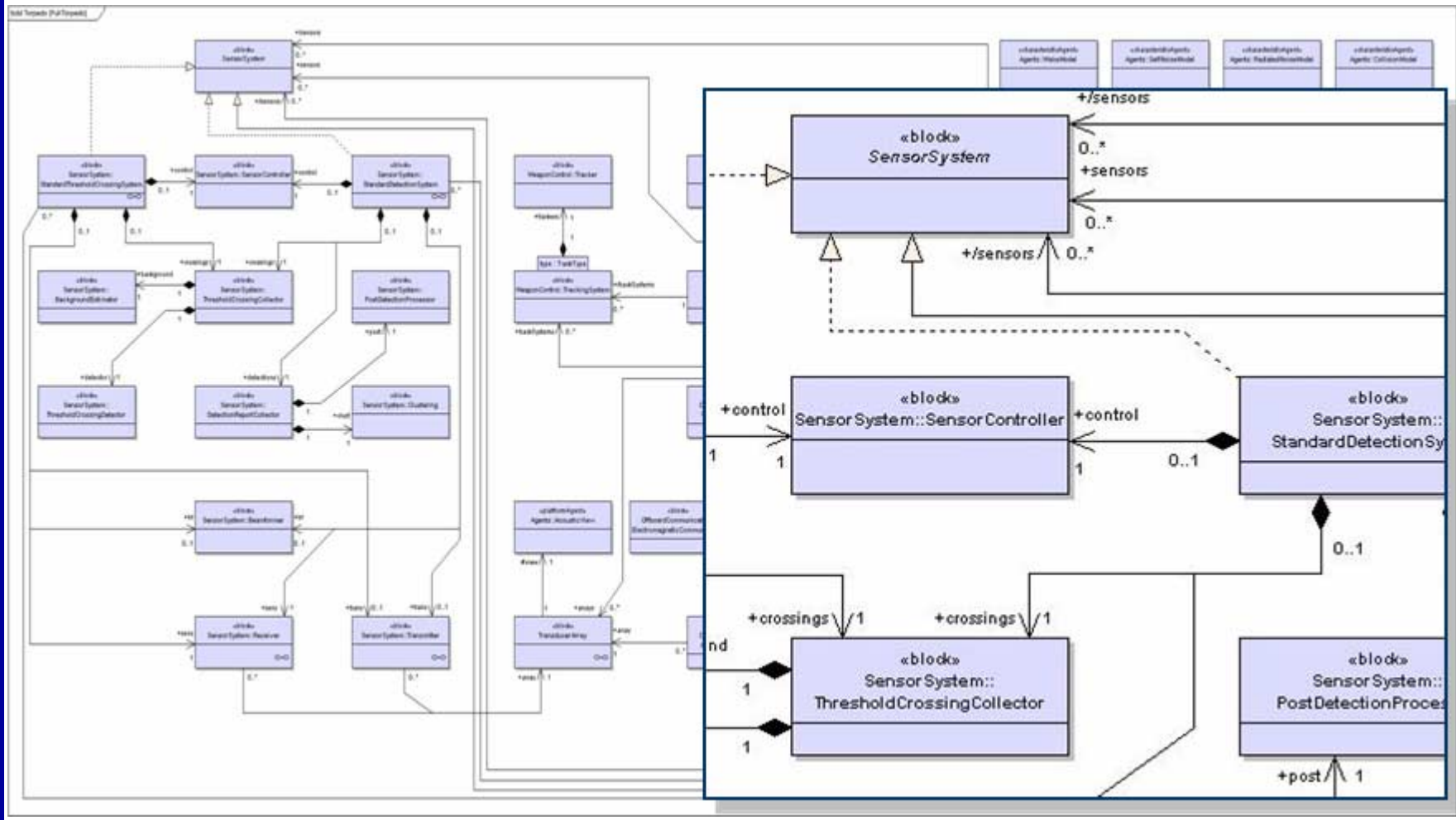


TEAMS Perspective: SysML Pros

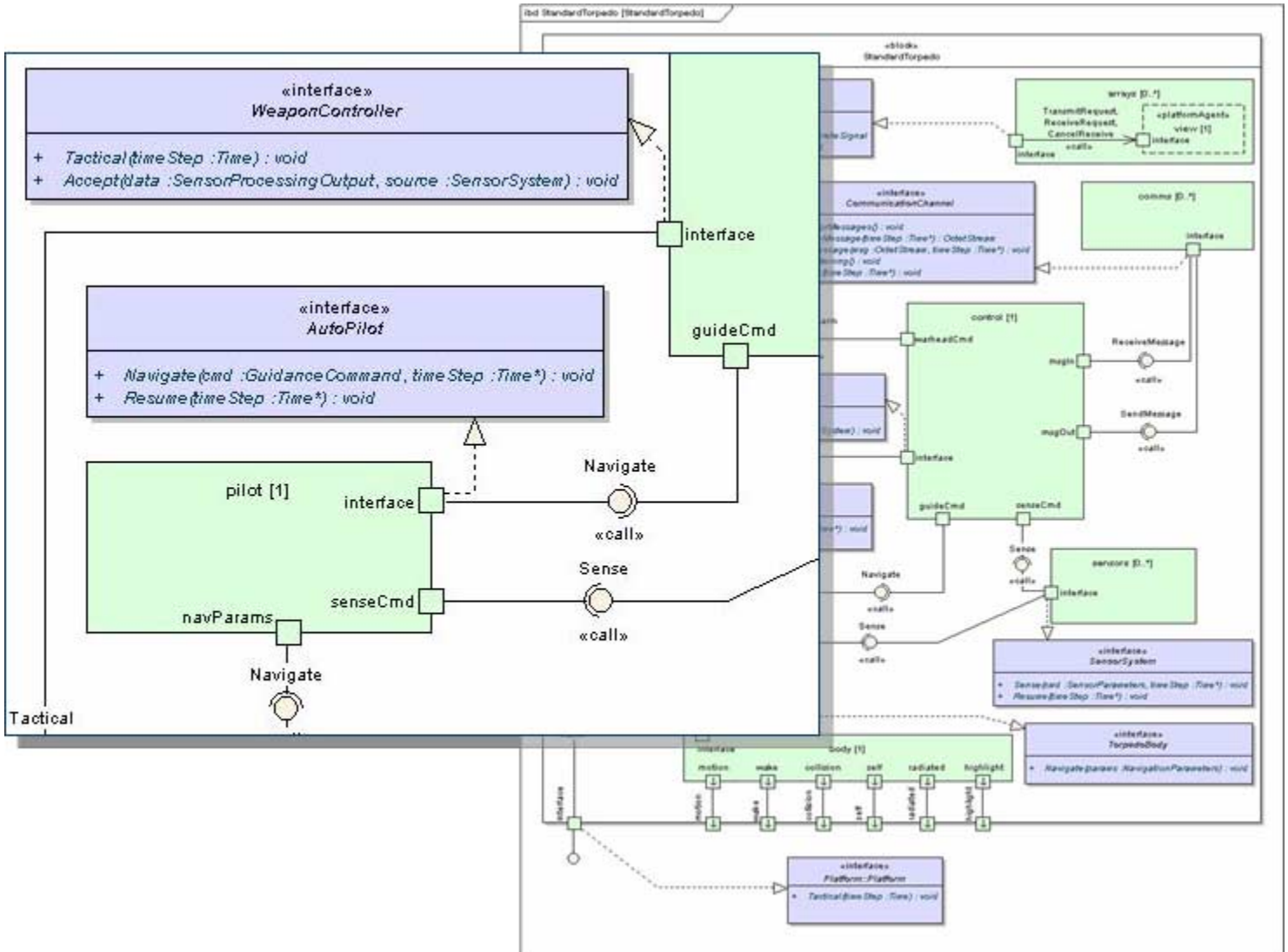
Pros

- Requirements
  - Explicitly lay out requirements and consequences
- Views and Viewpoints
  - Can separate requirements and model views based on stakeholders concerns
- Structure
  - Ability for model structure to verify requirements
    - Can search for requirements that aren't verified
    - Can search for model components that aren't justified
  - Separation of structure from behavior
    - SysML BDDs vs. IBDs and Activities allow for clear separation
    - UML allows this, but easier to implement in SysML
- Behavior
  - Dashed line for activity flow is more aesthetically pleasing
    - vs. UML solid line

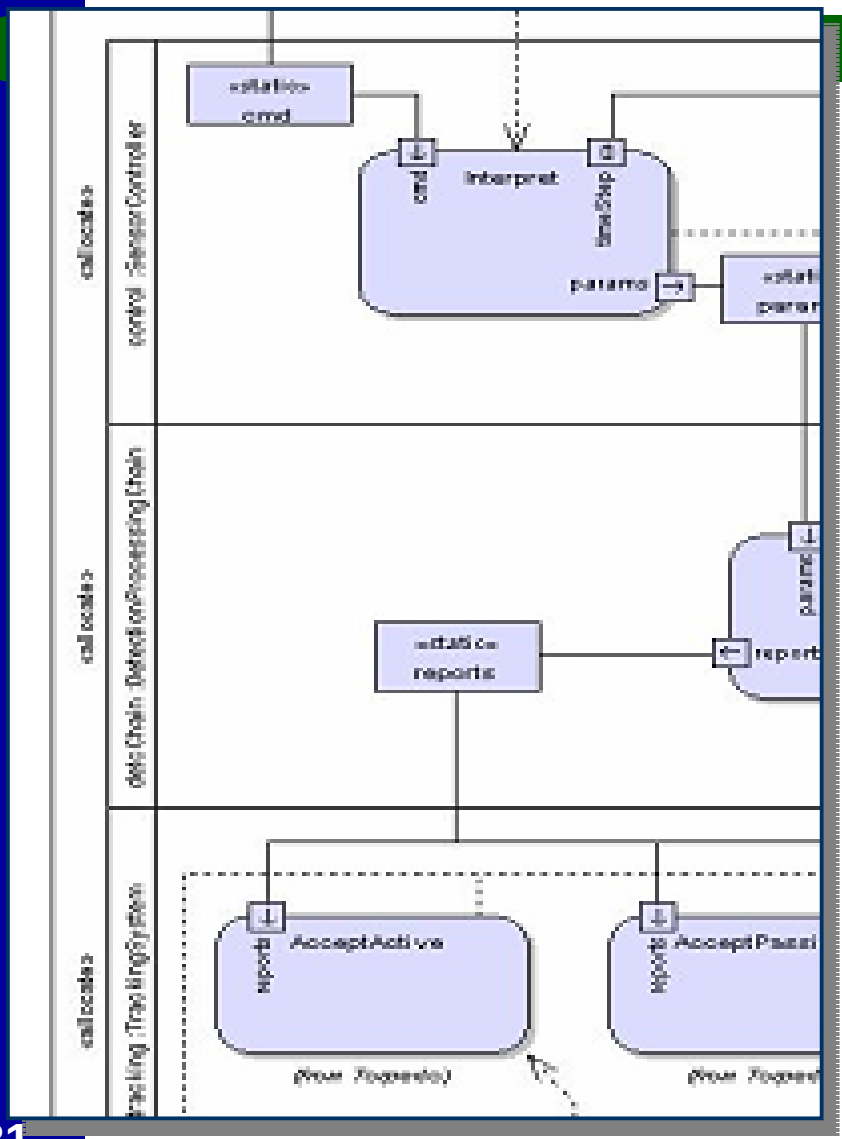
Torpedo Block Definition Diagram

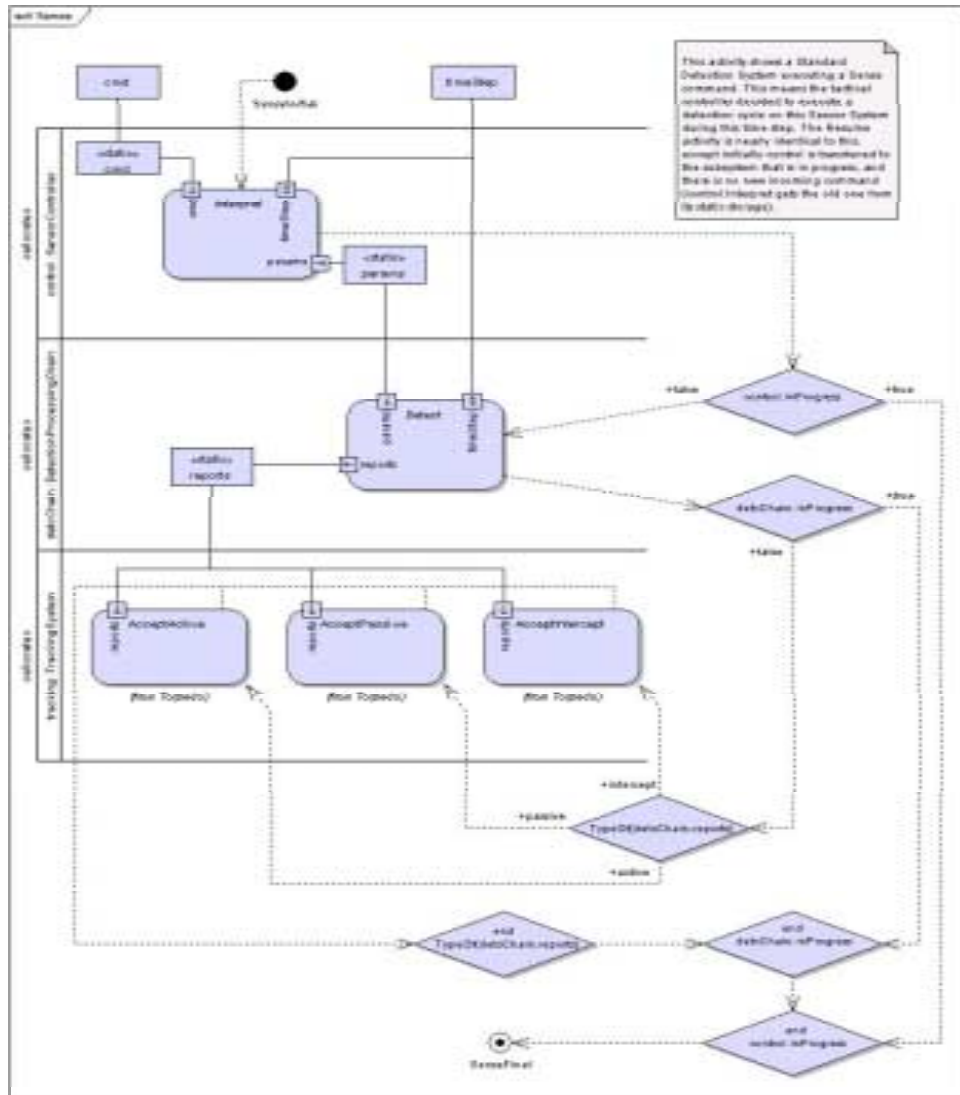


Torpedo Internal Block Definition Diagram

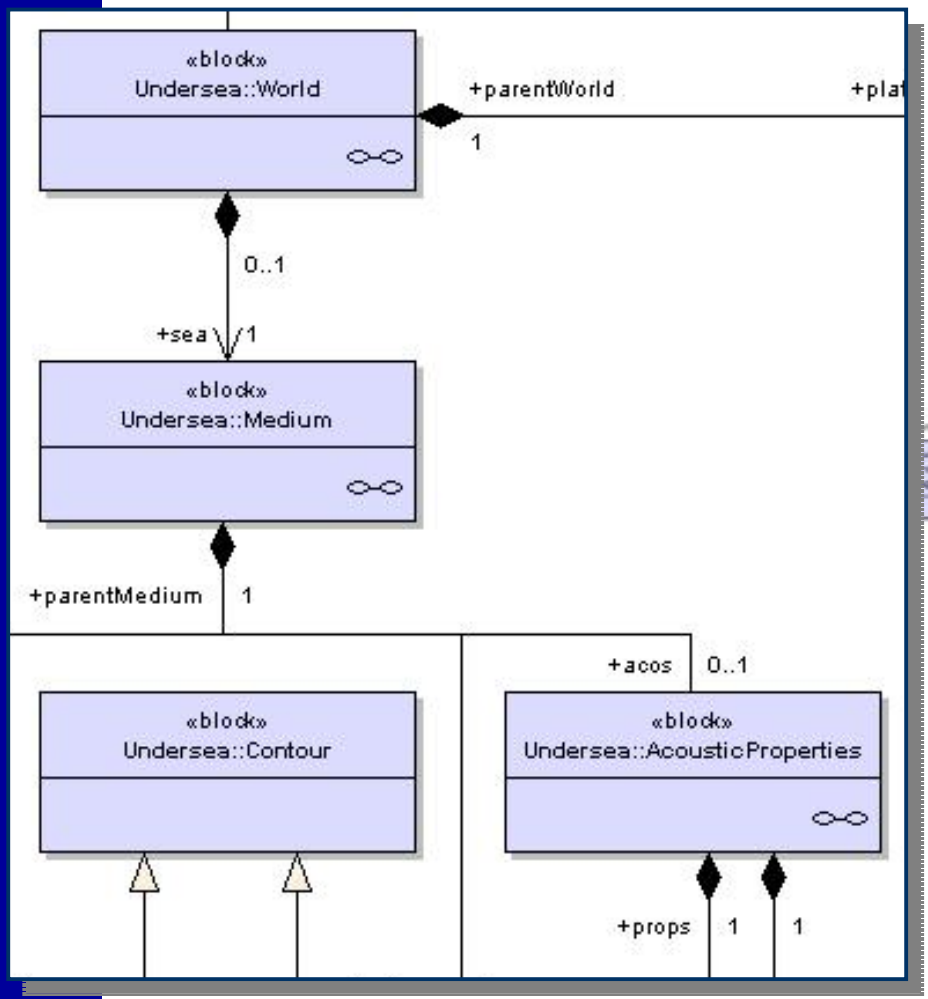


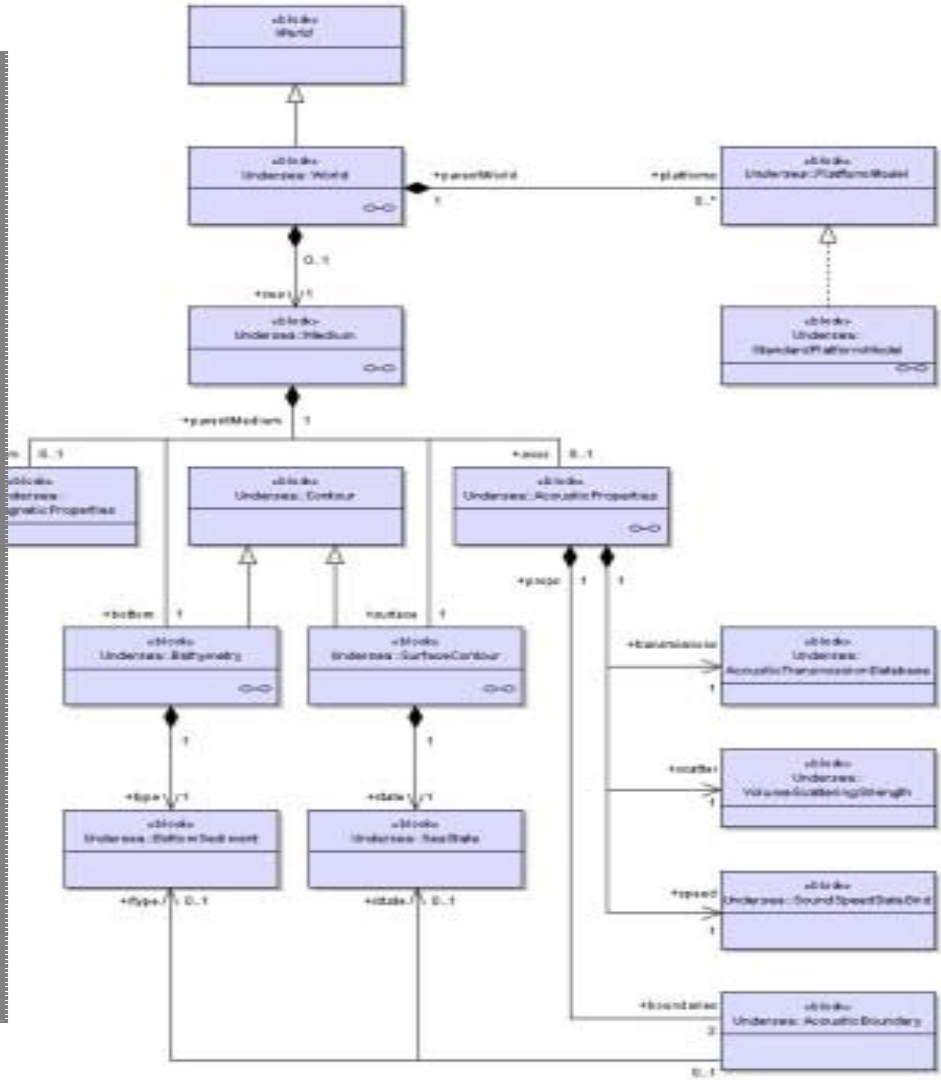
Torpedo Sensor Activity Diagram



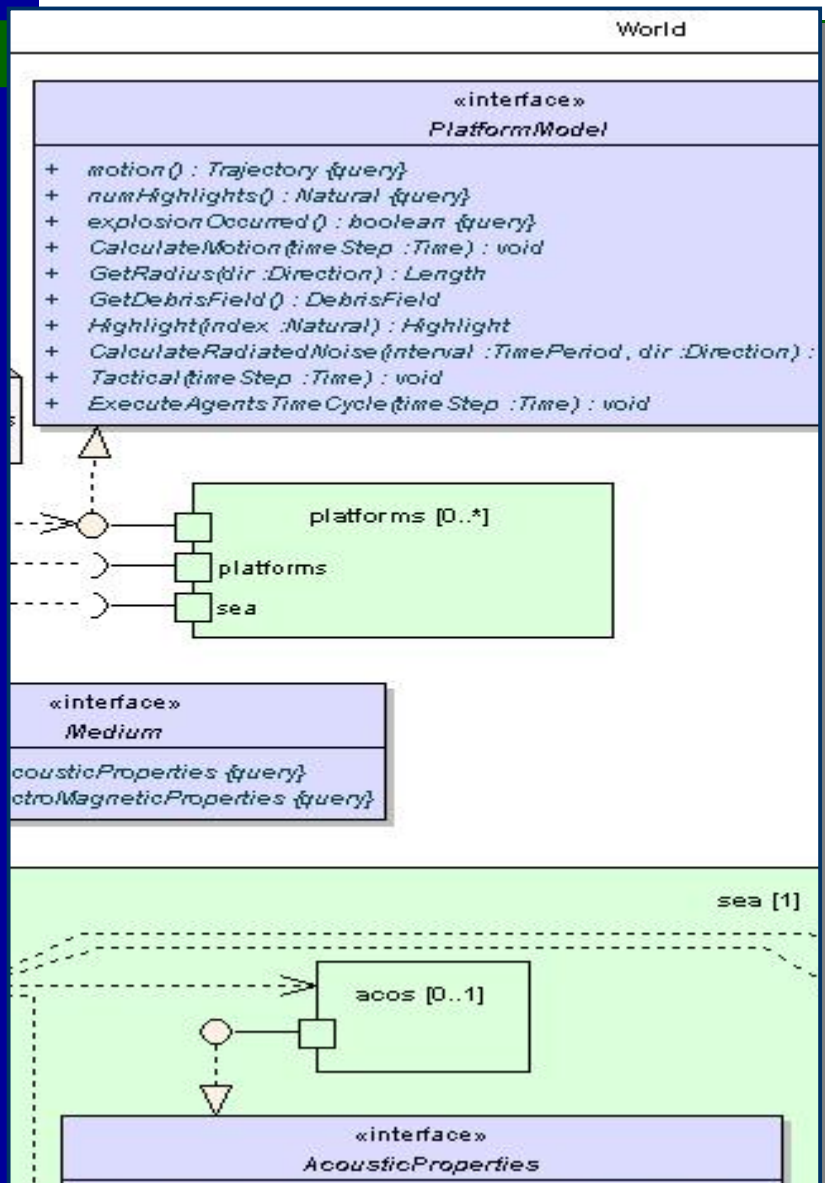


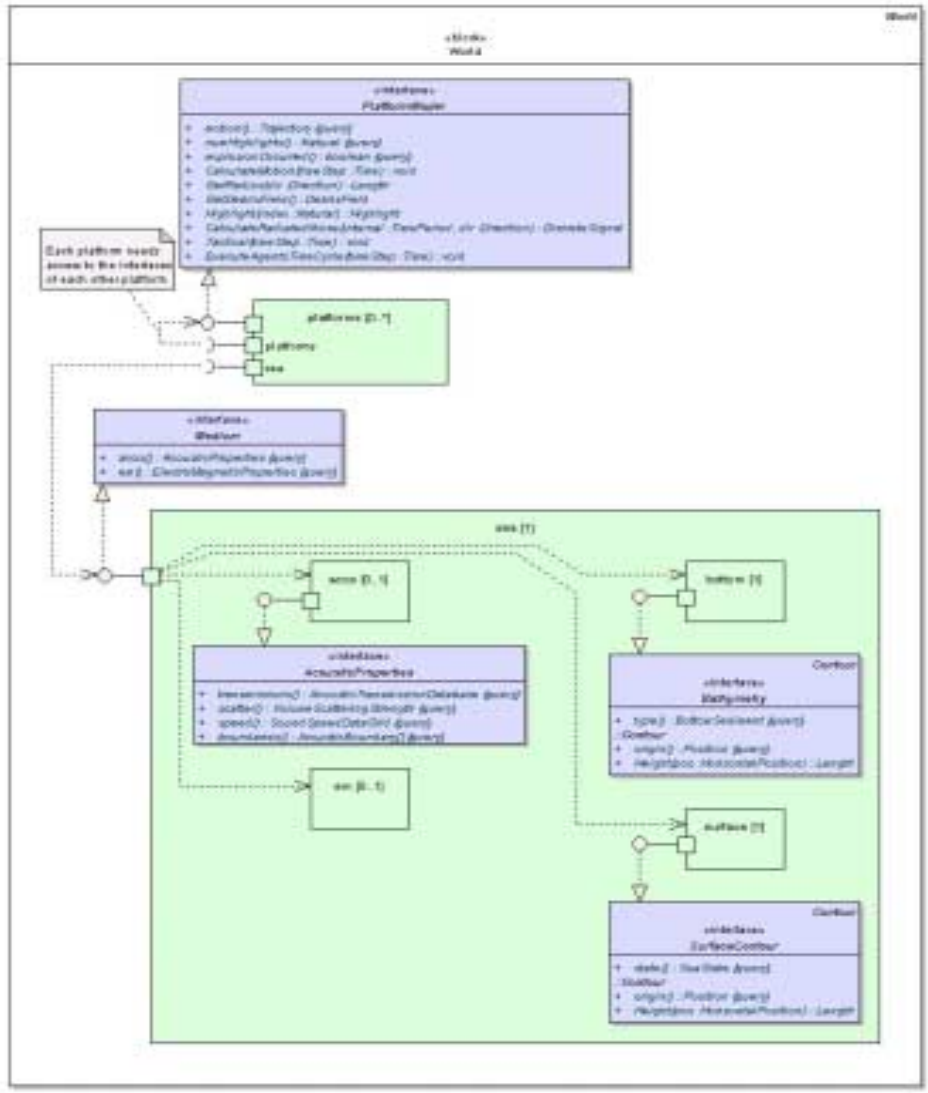
Undersea World Block Definition Diagram



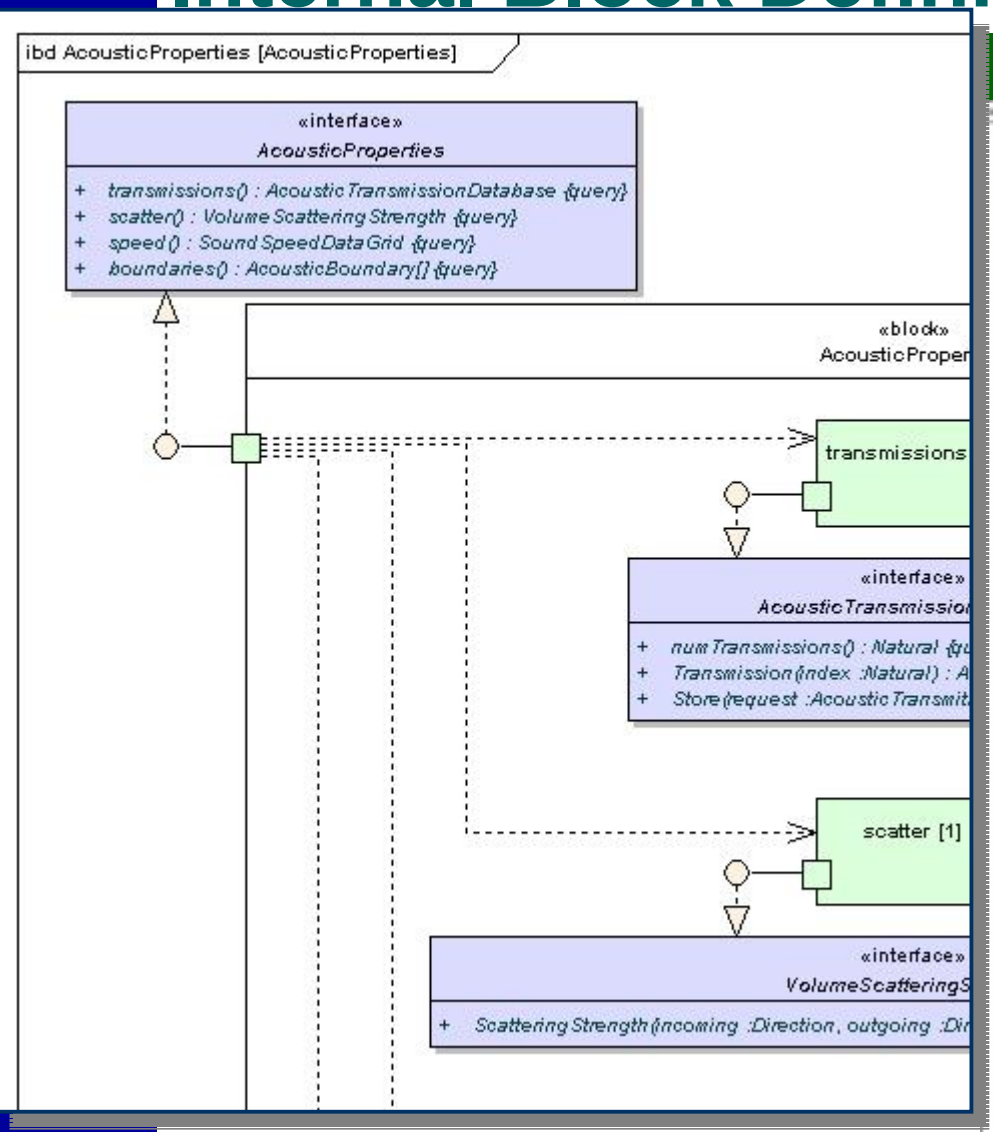


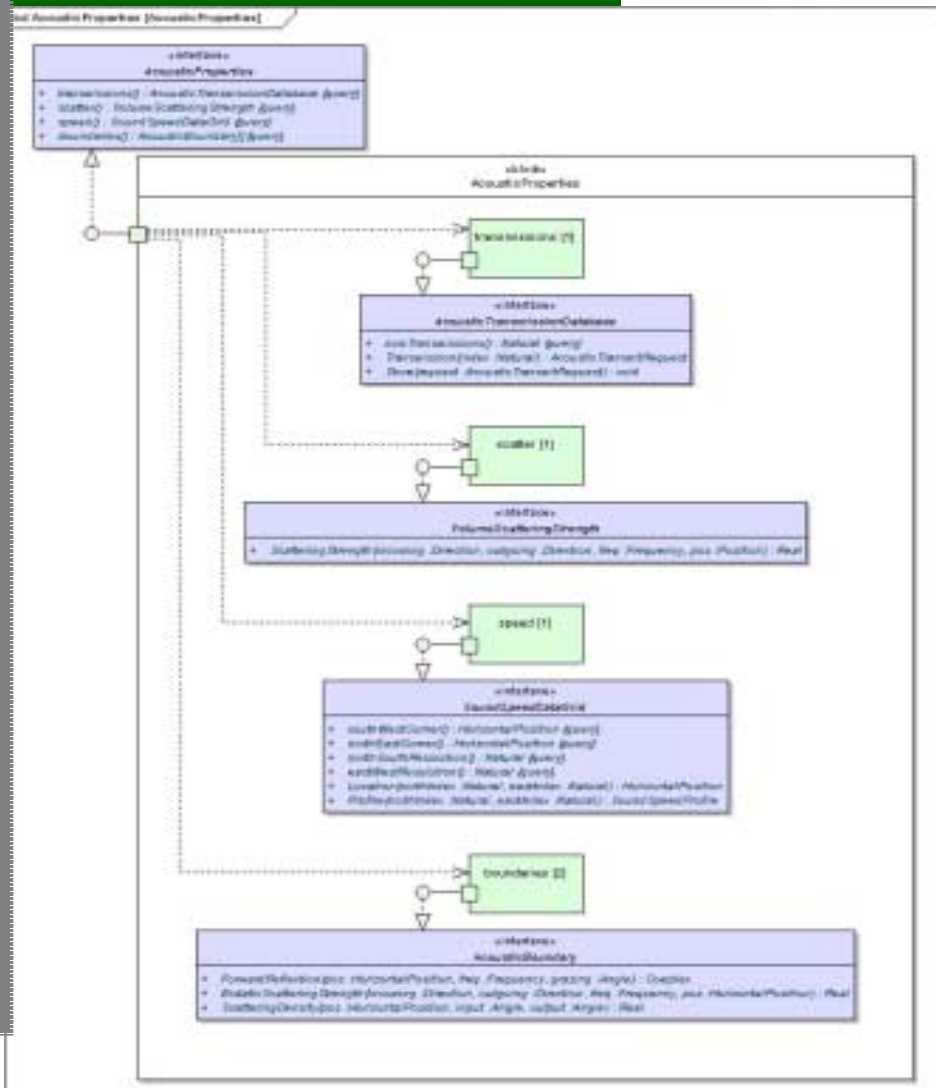
Simulation "World" Internal Block Definition Diagram





Acoustic Properties Internal Block Definition Diagram



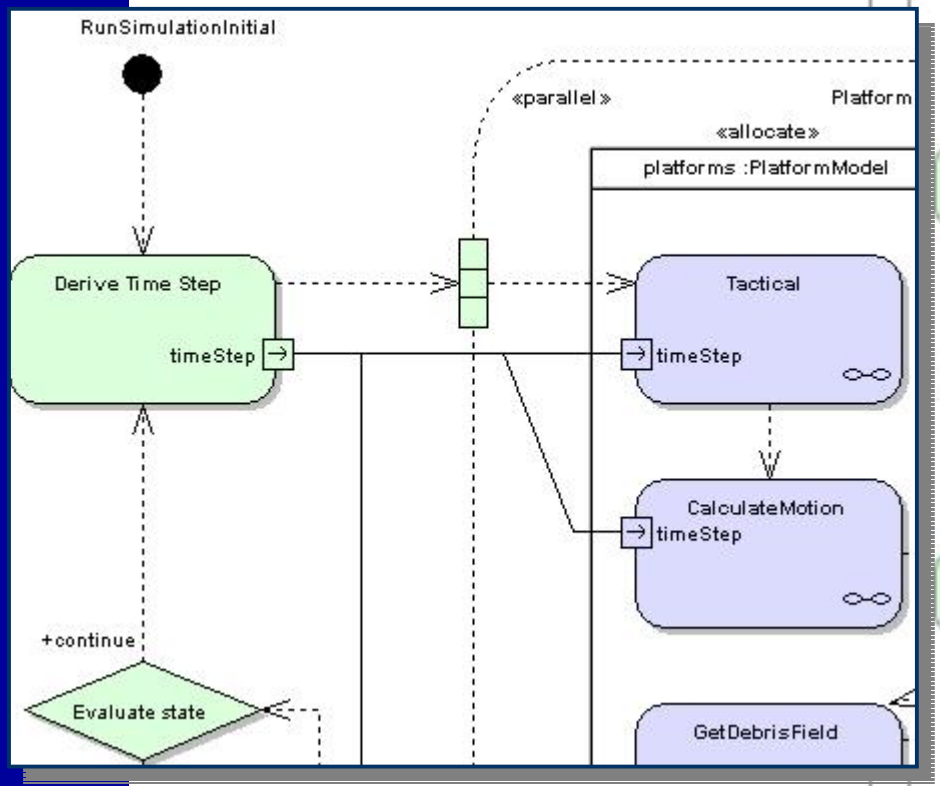


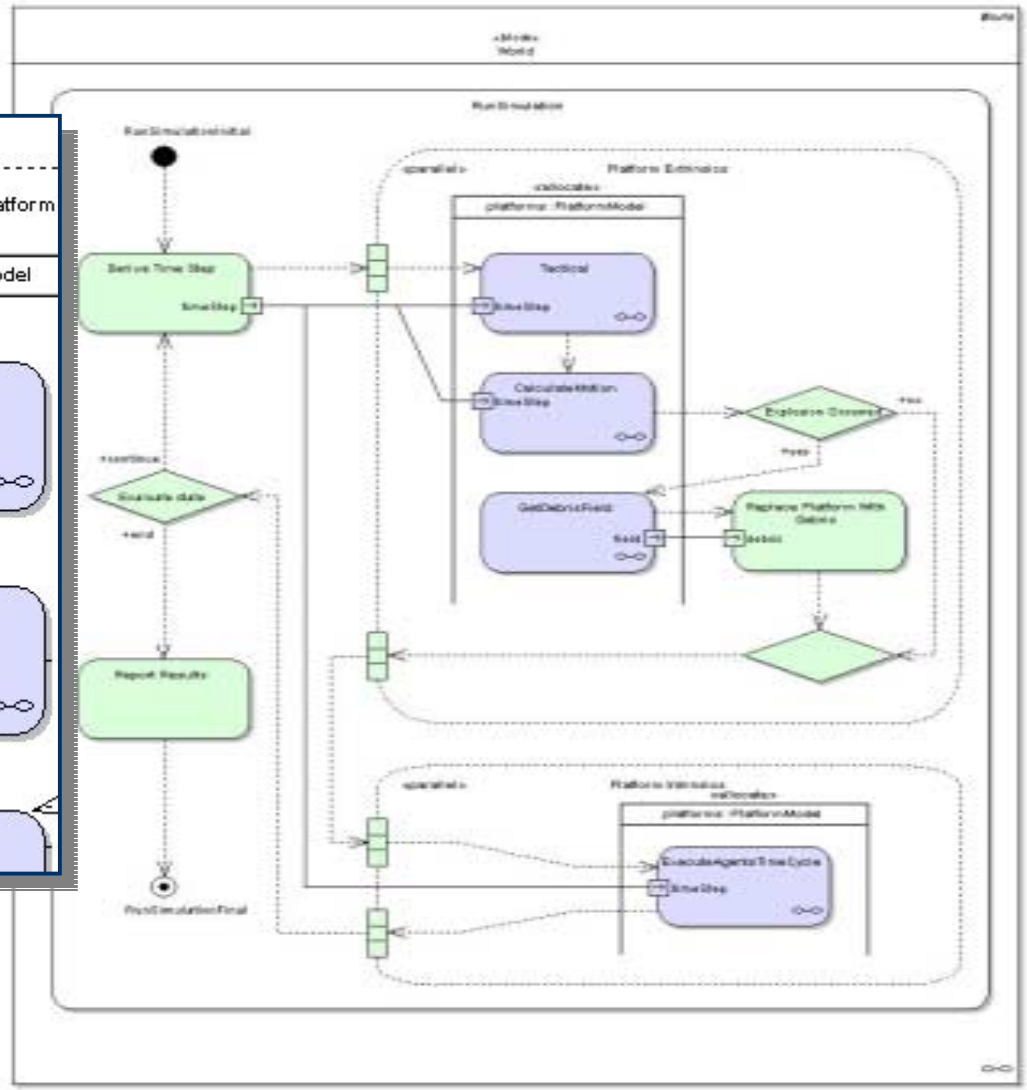
TEAMS Perspective: SysML Pros

Pros

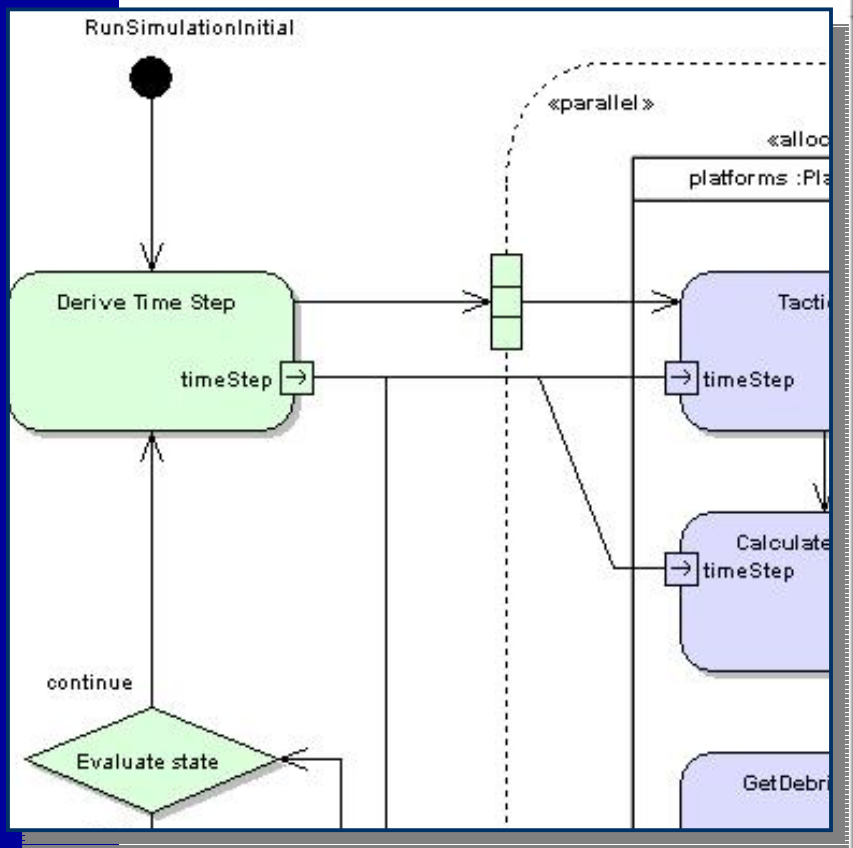
- Requirements
  - Explicitly lay out requirements and consequences
- Views and Viewpoints
  - Can separate requirements and model views based on stakeholders concerns
- Structure
  - Ability for model structure to verify requirements
    - Can search for requirements that aren't verified
    - Can search for model components that aren't justified
  - Separation of structure from behavior
    - SysML BDDs vs. IBDs and Activities allow for clear separation
    - UML allows this, but easier to implement in SysML
- **Behavior**
  - **Dashed line for activity flow is more aesthetically pleasing**
    - **vs. UML solid line**

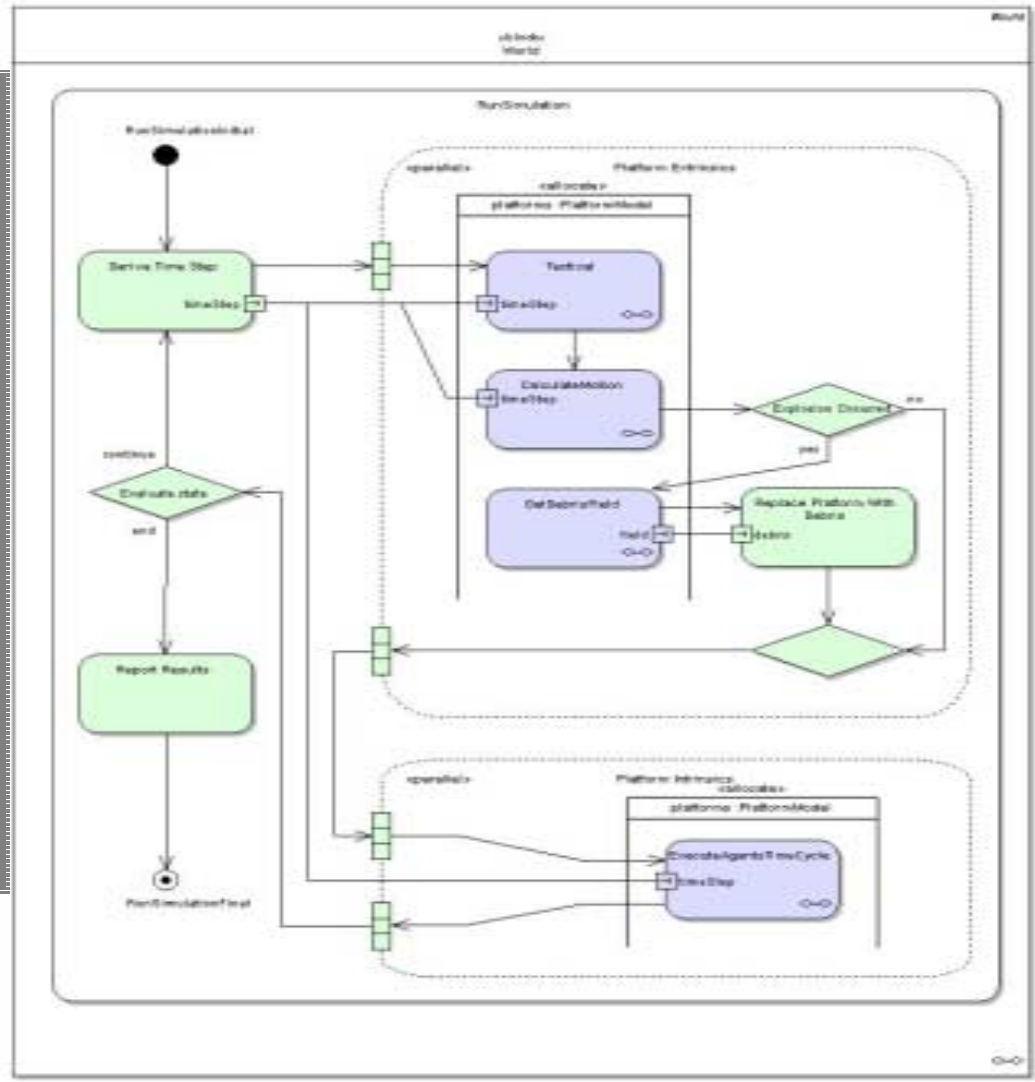
Simulation "World" Activity Diagram





Solid Line Representation



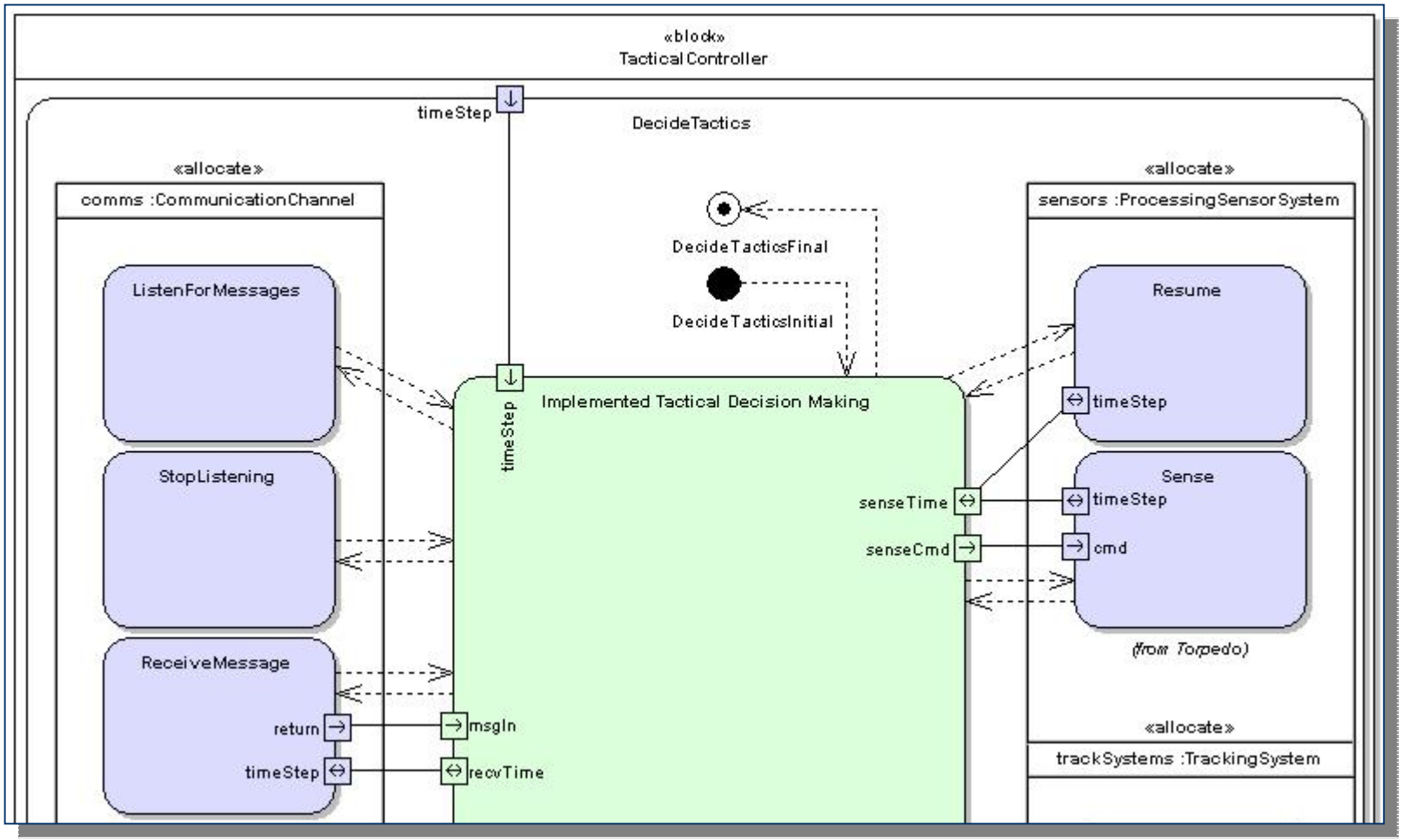


TEAMS Perspective: SysML Cons

Cons

- **Allocating CIM to PIM**
  - **Difficulty with abstract activities**
    - **Exit path dependent on logic within an activity is not accessible and can't be modeled**
    - **Not represented well in either UML or SysML – tactical controller example**
- **Implementing PIM**
  - **Not "direct" for some SysML features**
    - **Flow ports, continuous activities, parametric constraints involve more components than just themselves**
    - **Flows in "real systems" easier to represent**
    - **Flows in software modeling are open to interpretation**
  - **Requires additional documentation of model to bridge between SysML feature and executable code**

TEAMS
Tactical Controller Example



TEAMS Perspective: SysML Cons

Cons

- Allocating CIM to PIM
  - Difficulty with abstract activities
    - Exit path dependent on logic within an activity is not accessible and can't be modeled
    - Not represented well in either UML or SysML – tactical controller example
- Implementing PIM
  - Not "direct" for some SysML features
    - Flow ports, continuous activities, parametric constraints involve more components than just themselves
    - Flows in "real systems" easier to represent
    - Flows in software modeling are open to interpretation
  - Requires additional documentation of model to bridge between SysML feature and executable code

Lessons Learned and Value Added

- Requirements traceability is VITAL to the success of several TEAMS projects
  - ONR TEAMS standard framework and interfaces
  - OSD-ATL feasibility study
  - TOGAF/MDA Synergy Project
- SysML was designed with “real” systems in mind
  - where UML is software oriented
- Perceived concreteness – simulated vs. actual system
  - not just one way to design interfaces, need recommendations for implementation
- Still need some UML features not present in SysML
  - <<Instantiate>> or <<create>> for dynamic allocation
- Still need guidance on how to best implement parametrics and constraints for modeling and simulation

OMG SE DSIG Recommendation

"Clarify the distinction between the domain model and the simulation design model."

Domain Model

- Equivalent to the MDA CIM
- Represents the operational domain (e.g., torpedo, submarine platform, targets, and ocean environment)
- Specifies the requirements for the simulation design
- Capture in SysML model
- Parametrics used to specify constraints (e.g., torpedo dynamics, signal propagation)

Simulation Design Model

- Equivalent to the MDA PIM
- Represents the simulation software design
- SysML model "transformed" into simulation model (e.g. Map SysML structure, behavior, and parametrics into simulation components)
- Use SysML allocations to specify the CIM/PIM mapping (i.e.,transformation)

*Reference SE DSIG minutes from OMG San Diego Meeting on March 27, 2007

Acknowledgements

- LtCol Telford / Dwayne Hardy and OSD-ATL, for their interest and continued support
- David Drumheller and ONR, for their vision and continued support
- Sanford Friedenthal, for his expertise and willingness to educate the TEAMS consortium on the nuances of SysML
- Members of The Open Group, Object Management Group, and TEAMS Initiative who contributed to the success of SysML Project
- Sparx Systems, who provided complimentary licenses for Enterprise Architect 6.5 for this SysML effort