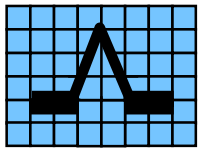
Problems with the UML: Formal versus Informal

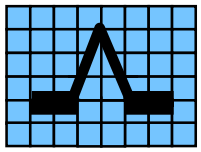
16 November 2000



George Krasovec
AST Engineering Services, Inc.
8100 Shaffer Parkway
Littleton, Colorado 80127
(303) 874-7360
gkrasovec@astes.com

Paul Ward
Paul Ward Associates
185 West End Ave.#27L
New York, NY 10023
(212) 362-1391
paul@p-w-a.com

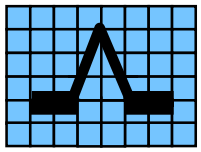
Overview



AST Engineering Services

- **Background**
  - **AST and PWA**
  - **R&D sponsor**
  - **Research objectives**
- **Evaluation of UML as a formally analyzable architecture description language (ADL)**
- **Problems with attribute and composition semantics**
- **Proposed formal semantics and notation**
- **Formal analysis of UML architecture models**
- **Sample application areas for UML-based formal analysis**
- **Conclusions**

Disclaimer Regarding Intellectual Property

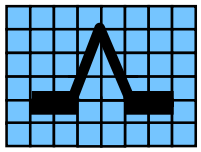


AST Engineering Services

- **Extensions to and clarifications for the use of UML semantics presented herein are intended for use in the public domain and eventual incorporation into the UML specification.**

- **Techniques for the translation of UML semantics into Prolog, along with supporting Prolog encodings and certain predicate definitions, are proprietary to AST Engineering Services and have been documented in a provisional patent application submitted to the US Patent and Trademark Office.**

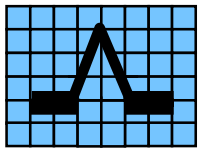
Introductions



AST Engineering Services

- AST Engineering Services, Inc.
  - offices in Littleton, Colorado and Falls Church, Virginia, USA
  - corporate specialty areas include
    - computer system performance engineering
    - system modeling and analysis
    - design methodology and tools R&D
  - George Krasovec, VP of R&D
    - system engineering and analysis
    - conformal mappings of design notations
    - semi-formal analysis
    - orbital mechanics, large-scale simulation
- Paul Ward, Paul Ward Associates, New York, USA
  - system design methodologist
  - author, consultant, lecturer, CASE tool technologist
  - co-developer of the Real-Time Structured Analysis design methodology

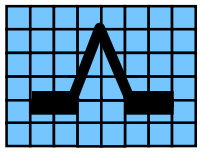
About our R&D Sponsor ...



AST Engineering Services

- **US Naval Sea Systems Command**
  - **DD 21 Surface Combatant currently the largest Naval ship procurement in US**
    - **broad war fighting capabilities for open sea and littoral missions**
    - **reduced manning and increased reliance on automation**
    - **incorporation of open system design processes and standards**
    - **highly distributed computing resources shared among functional applications**
  - **DD 21 Program Office funds R&D related to potential risk areas**
    - **automated support for the design and operation of the Total Ship Computing Environment (TSCE)**
    - **tools and techniques which allow system engineers to “reason about” the design of the TSCE architecture**
    - **administered through the Small Business Innovative Research program**
  - **DD 21 Program Sponsorship**
    - **Modeling and Simulation program element**
    - **TSCE program element**

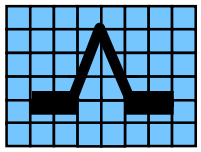
R&D Overview



AST Engineering Services

- **R&D requirement**
  - **develop innovative methods, techniques, and tools for the system engineering of large complex computer-based systems using open system architectures for commercial off the shelf (COTS) components.**
- **Technical challenges**
  - **develop a hardware/software architecture modeling approach based on a standard design notation (UML)**
  - **support scalability to extremely large analysis domains**
  - **develop technique to represent and evaluate compliance with open system standards and profiles**
  - **seamlessly integrate this functionality with a commercial-quality system engineering tool environment**
- **Specific Phase II objectives accomplished over the past 22 months**
  - **design and implement architecture modeling and analysis tool set**
    - **provide ability to capture and "reason about" system architectures**
    - **applicable to hardware and software systems**
  - **develop scalable modeling conventions**
  - **test and document**

Evaluation of UML as a Formal ADL



AST Engineering Services

- **UML selected as the design notation for this R&D**
  - **OMG standard**
  - **widely used**
  - **multi-vendor tool support**

- **Primary UML design views utilized**
  - **Class diagrams**
  - **Instance diagrams**

- **Extensive instance model development exposed early problems**
  - **poor tool support for instance model creation**
  - **scalability problems for graphically specified instance models**
  - **informally defined semantics related to textual attribute and composition constructs**

Semantics of Attributes and Compositions

AST Engineering Services

- **Problems were identified with the descriptions of attribute and composition semantics in the 1.2 and 1.3 versions of the UML Specification:**

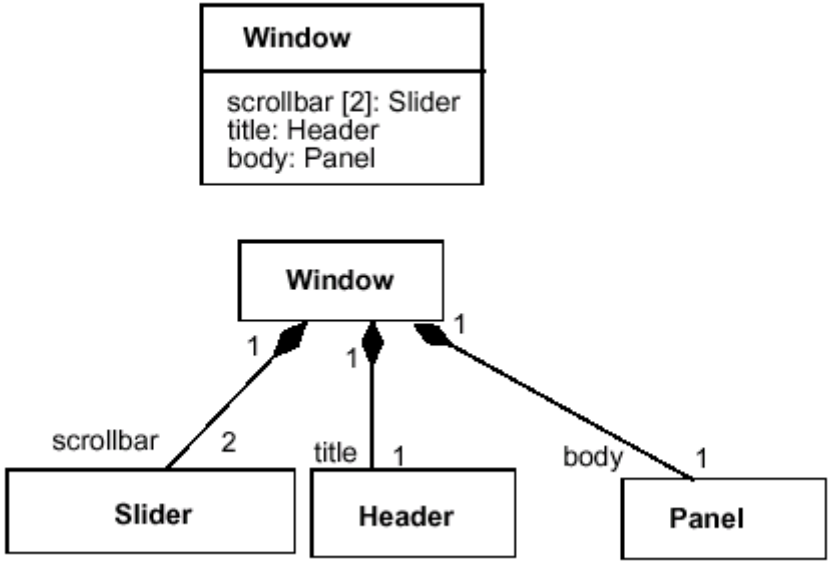
2.5.2 Abstract Syntax

AssociationEnd

name **(Inherited from ModelElement): The rolename of the end. When placed on a target end, provides a name for traversing from source instance across the association to the target instance or set of target instances. It represents a pseudo-attribute of the source classifier (i.e., it may be used in the same way as an Attribute) and must be unique with respect to Attributes and other pseudo-attributes of the source Classifier.**

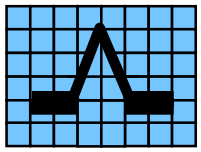
3.47.3 (Composition) Design Guidelines

Note that a class symbol is a composition of its attributes and operations. The class symbol may be thought of as an example of the composition nesting notation (with some special layout properties). However, attribute notation subordinates the attributes strongly within the class; therefore, it should be used when the structure and identity of the attribute objects themselves is unimportant outside the class.



3.47.4 Example

Attributes and Compositions - Semantic Problems

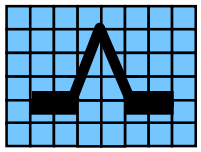


AST Engineering Services

- **Problems with UML 1.3 Attribute and Composition definitions**
  - **composition design guidelines are vague and arbitrary**
  - **determination of outside importance of attributes is subjective and unpredictable**
    - **the 3.47.4 example seems to violate this guidance**
  - **role name usage on compositions is inherited from associations**
    - **does not recognize that instance traversal could be inherently different for compositions versus associations in general**
  - **implies that under certain conditions, compositions do not extend the Classifier nesting (encapsulation) boundary to include the target Classifier**

- **Proposed semantics for compositions**
  - **extend the Classifier nesting boundary to include the target Classifier in all cases**
  - **allows for traversal mechanisms peculiar to compositions**

Proposed Equivalence between Attributes and Compositions



AST Engineering Services

- **Permit use of attribute and composition assertions interchangeably**
  - **precedence given to graphic assertion when both are present**
  - **use of one form or the other is a modeling preference**
    - **compactness versus visualization**

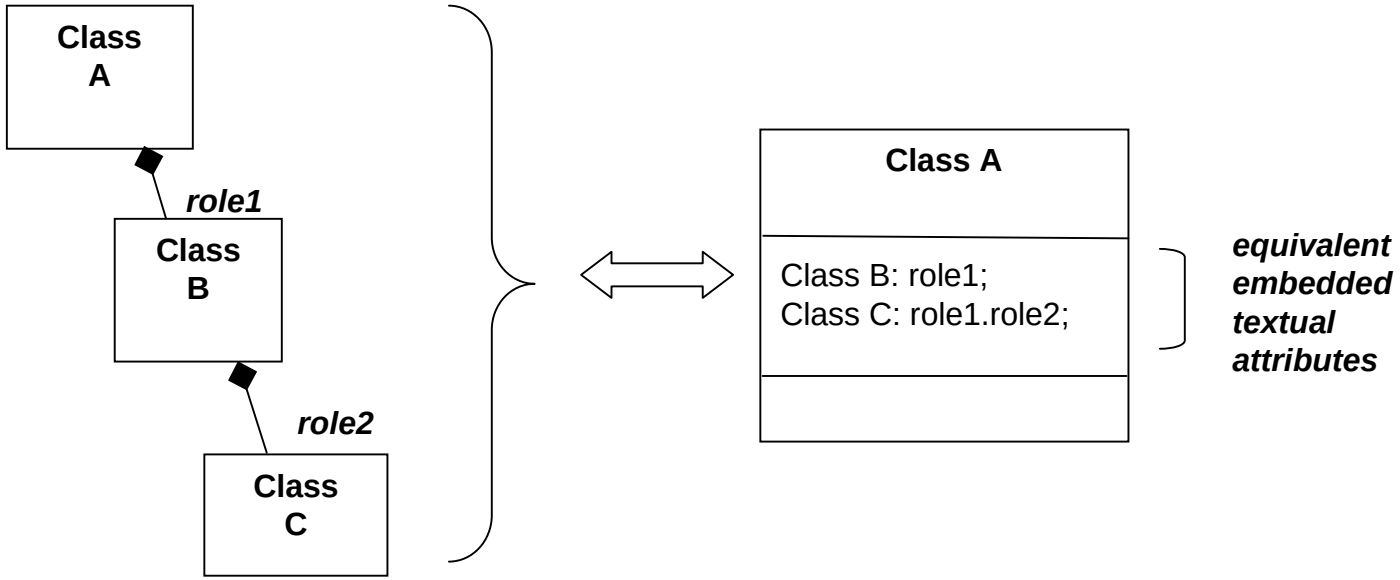
- **Compositions are included within the encapsulation boundary of the source Classifier**
  - **appear as “local” attributes**
  - **similar to nested data structures**
  - **derive attribute name from composition role (on target end)**

- **Require extension of dot notation similar to use in OCL expressions**
  - **hierarchical compositions**
  - **provides unique naming convention for nested objects**
  - **allows for transliteration of textual attributes from composition assertions and vice versa**

Example of Attribute - Composition Equivalence in Class Models

AST Engineering Services

Classes

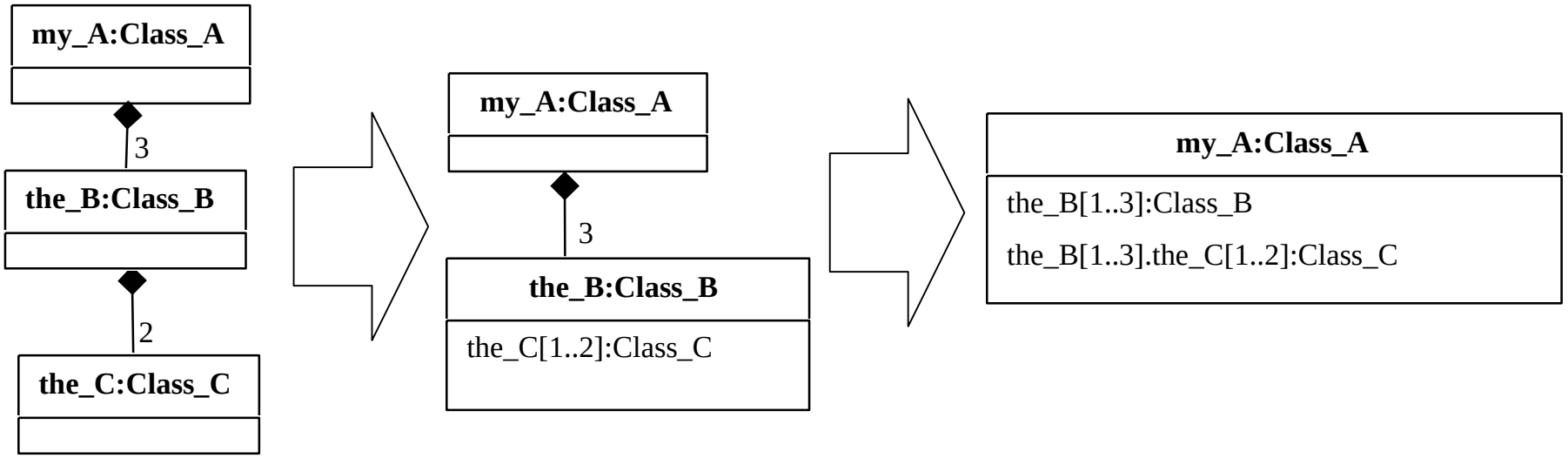


graphical assertions take precedence over textual assertions

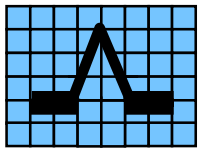
Example of Attribute - Composition Equivalence in Instance Models

AST Engineering Services

Instances



System Modeling Approach with Strengthened UML Formalisms

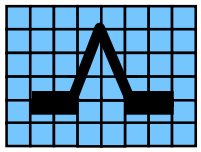


AST Engineering Services

- **Modeling conventions developed**
  - **Based on standard UML notation**
  - **Hierarchical ownership of model objects**
  - **Equivalence of compositions and attributes**
  - **Compaction of graphical representation using object replication**
  - **Structured object naming syntax (a.b.c[3].d.e[1])**
  - **INVARIANT expressions in class model (constraints)**
  - **DEPENDENCY assertion in class model (profile conformance)**
  - **BOUND assertion in instance models (connections)**

- **Significance**
  - **our enhanced UML is a powerful architecture description language**
  - **supports hardware and software co-modeling**
  - **scaleable for extremely large modeling problems**
  - **compact graphical representation**

Formal Representation of UML Models



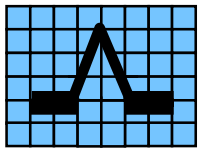
AST Engineering Services

- **Translation of UML models into a formal analysis script needed to support automated reasoning capability**
  - **Prolog chosen**
    - **powerful pattern search mechanisms**
    - **rules expressed as predicate logic**
    - **back tracking for efficient execution**
  - **well suited for identifying and matching patterns inherent in architecture models**

- **Encoding formats optimized for simplification of analysis predicates**
  - **relies extensively on Prolog's list of lists notation**

- **Prolog also chosen as the implementation language for our UML constraint language**
  - **conciseness**
  - **expressivity**
  - **can be translated to/from OCL as necessary**

Prolog Class Encodings



AST Engineering Services

- **Basic class assertion**

  class(className, Concrete|Abstract)

- **Class modifiers**

  isa(subClassName,superClassName).

  containsa(containerClassName,containedClassName,roleName,[multiplicity]).
  hasa(containerClassName,containedClassName,roleName,[multiplicity]).

  method(className,methodName,argumentList,returnClassName,visibility).

- **Example**

class('CISCO_7200', 'Concrete').

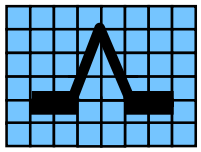
isa('CISCO_7200','CommercialProduct').

isa('CISCO_7200','Router').

containsa('CISCO_7200','NetworkProcessingEngine','ciscoNPE',['1','1']).

method('CISCO_7200', 'reset','','null','External').

Prolog Instance Encoding



AST Engineering Services

- **Instance assertion**

  inst(instanceList, className, value, serialNumber).

  where instanceList contains an ordered containment hierarchy of scalar or subscripted container objects (obj_1.. obj_{n-1}) and a leaf instance object (obj_n)

  [[obj_1{,ss}], [obj_2{,ss}] … [obj_n{,ss}]] ← ***prolog list of lists notation***

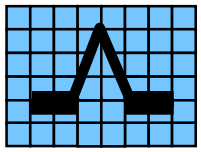
- **Examples**

  inst([[testNetSeg],[router,2],[nPorts]],'int',4,5).

  inst([[testNetSeg],[router,2],[ciscoIOC],[pcCardSlot,2]],'PCIFlashCard',nav,15).

  (nav keyword signifies "not a value")

Other Prolog Encodings



AST Engineering Services

- **Predicates are the building-blocks of the Prolog language**
  - **<head> :- <body>.**
- **Invariants**
  - **expressed via UML binary constraint construct**
  - **encoded as INVARIANT(<prolog body>) in binary constraint description**
  - **future consideration for using UML's object constraint language (OCL)**
  - **example:**

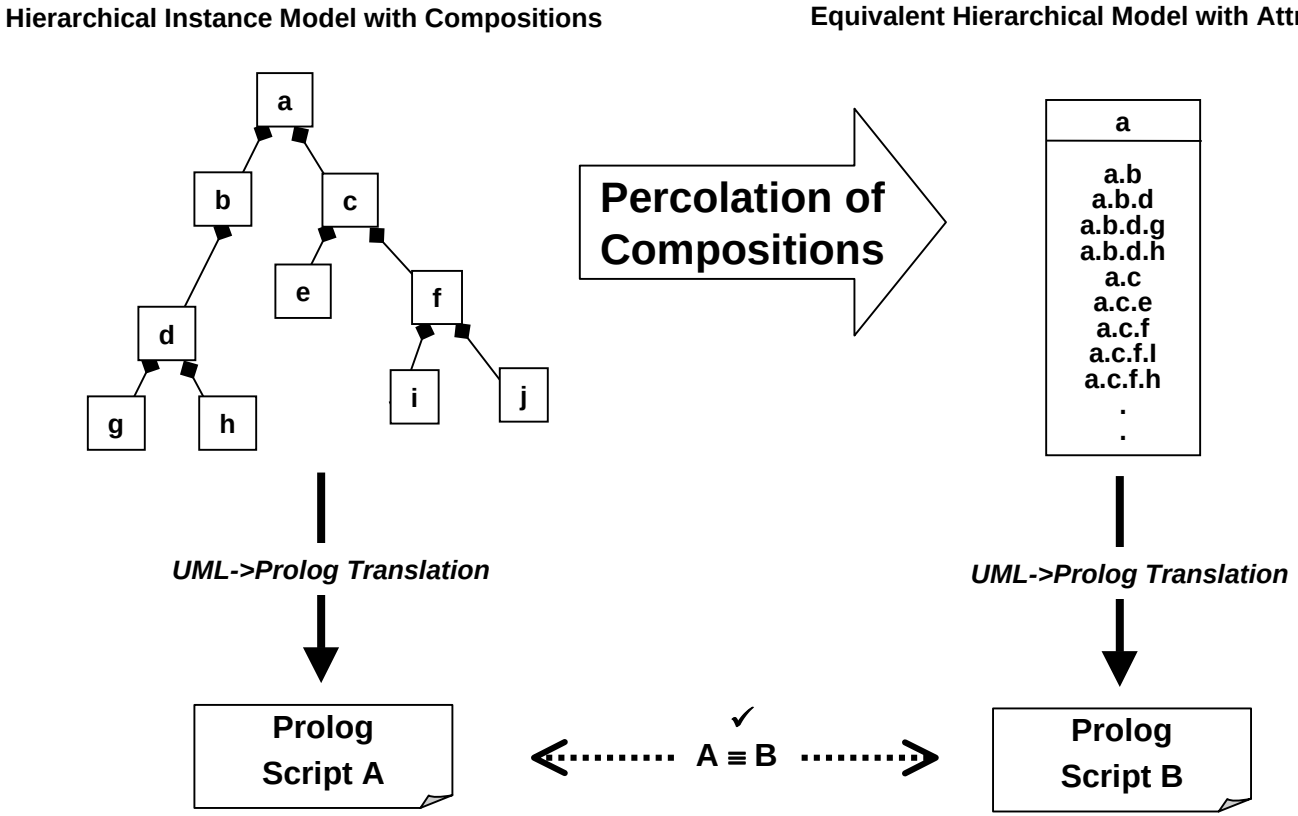
    INVARIANT(cntr_inc(0,_), instr([[partNumber],[ciscoNPE]\|X],_,'NPE-100',ID1), instr([[partNumber],[ciscoPSA,Slot]\|X],_,'PA-A3-OC3MM',ID2), error(301,[ID1,ID2]),fail.)

- **Bound assertions for expressing connectivity**
  - **bound(serialNumber$_1$,serialNumer$_2$).**
  - **serialNumber$_i$ is a link to inst$_i$**
  - **Example: *bound(4,33).***

- **Interface conformance**
  - **uses UML DEPENDS construct with stereotype name**
  - **Example: *conformsTo(36,49).***

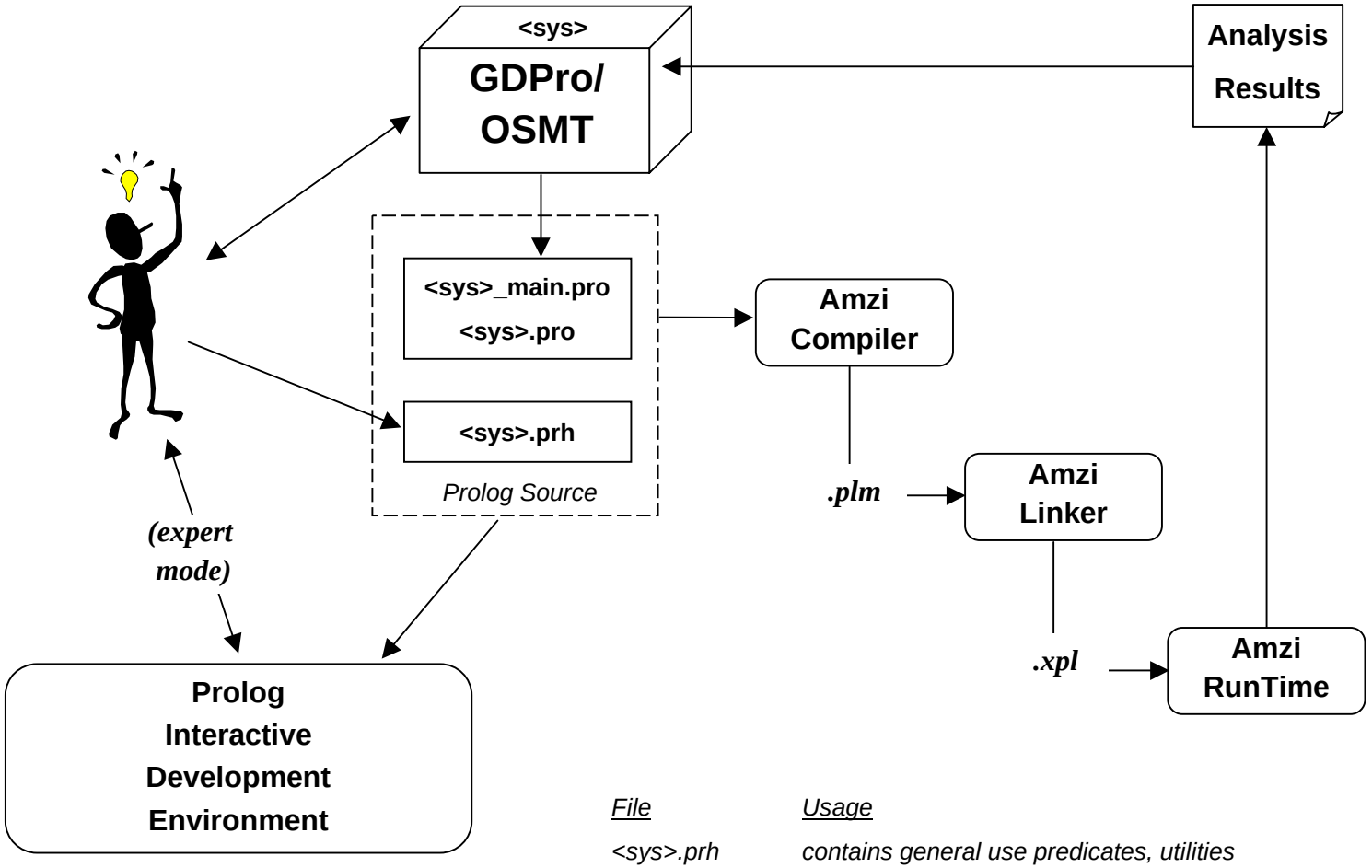
Enhanced Attribute and Composition Definitions - Demonstration of Semantic Invariance

AST Engineering Services



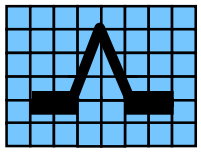
Prototype Prolog Analysis Environment Implemented through GDPro CASE Tool

AST Engineering Services



File	Usage
<sys>.prh	contains general use predicates, utilities
<sys>.pro	contains system fact base
<sys>_main.pro	contains main entry point, top-level logic

Applicability of UML-Based Formal Modeling and Analysis



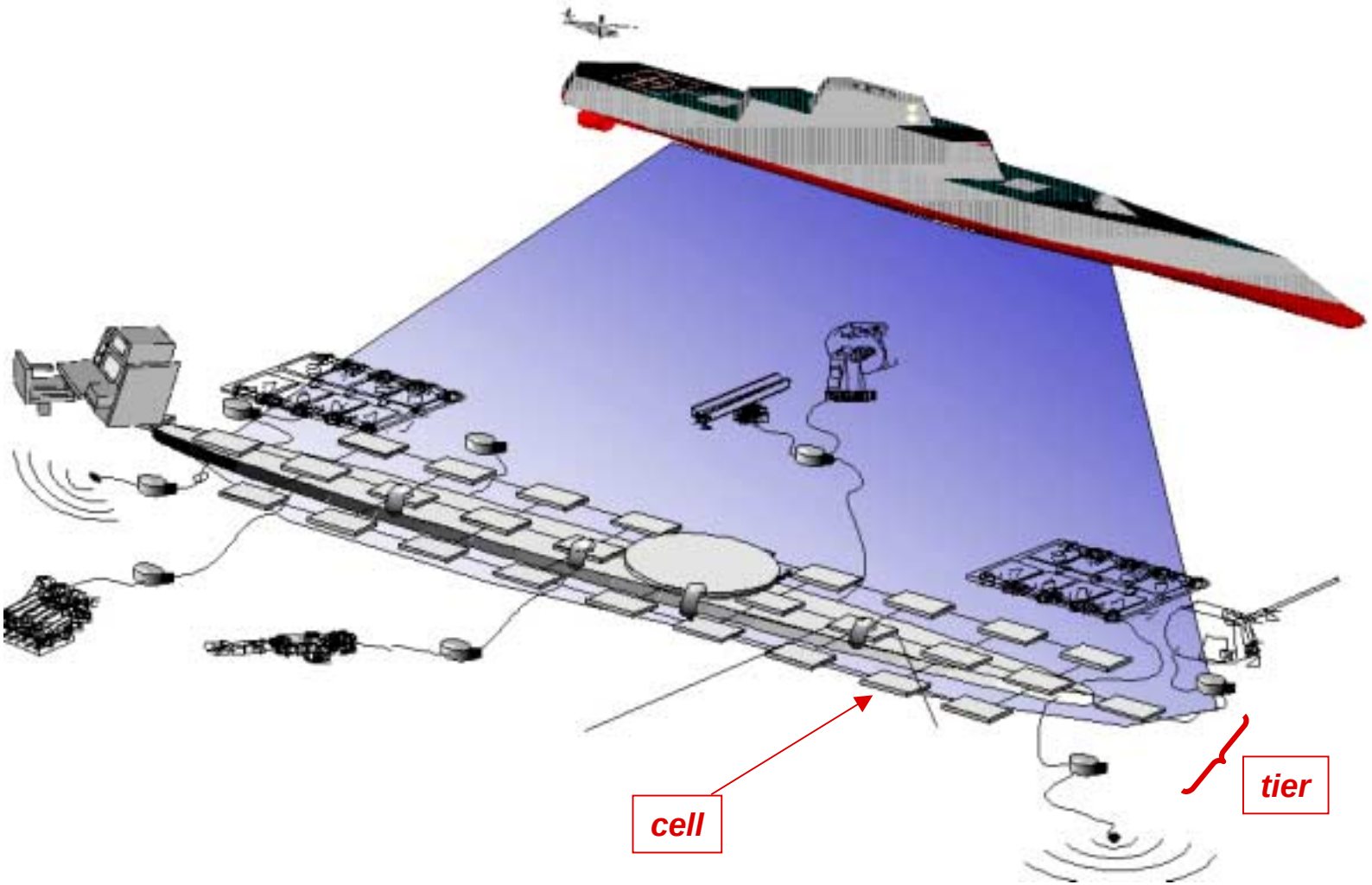
AST Engineering Services

- **UML/Prolog technology is relevant to any system of things with:**
  - **structure**
    - **connectedness**
      - **interfaces**
        - **complexity**
          - **interdependencies**

- **Can provide automated support to help develop, operate, and upgrade complex systems (open or otherwise):**
  - **specify**
    - **synthesize**
      - **validate**
        - **operate**
          - **evolve**

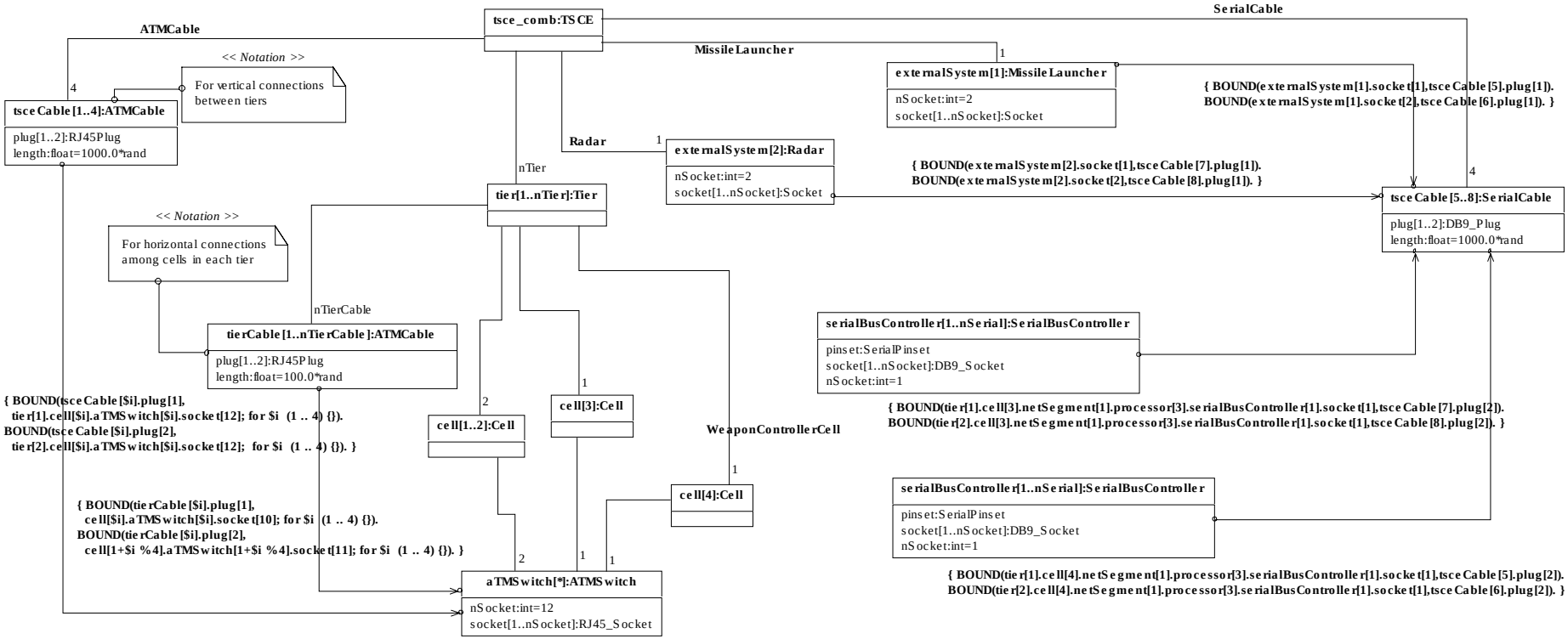
Sample Applications - TSCE Architecture Evaluation

AST Engineering Services



Large-Scale DD 21 TSCE Instance Model with BOUND Assertions

AST Engineering Services

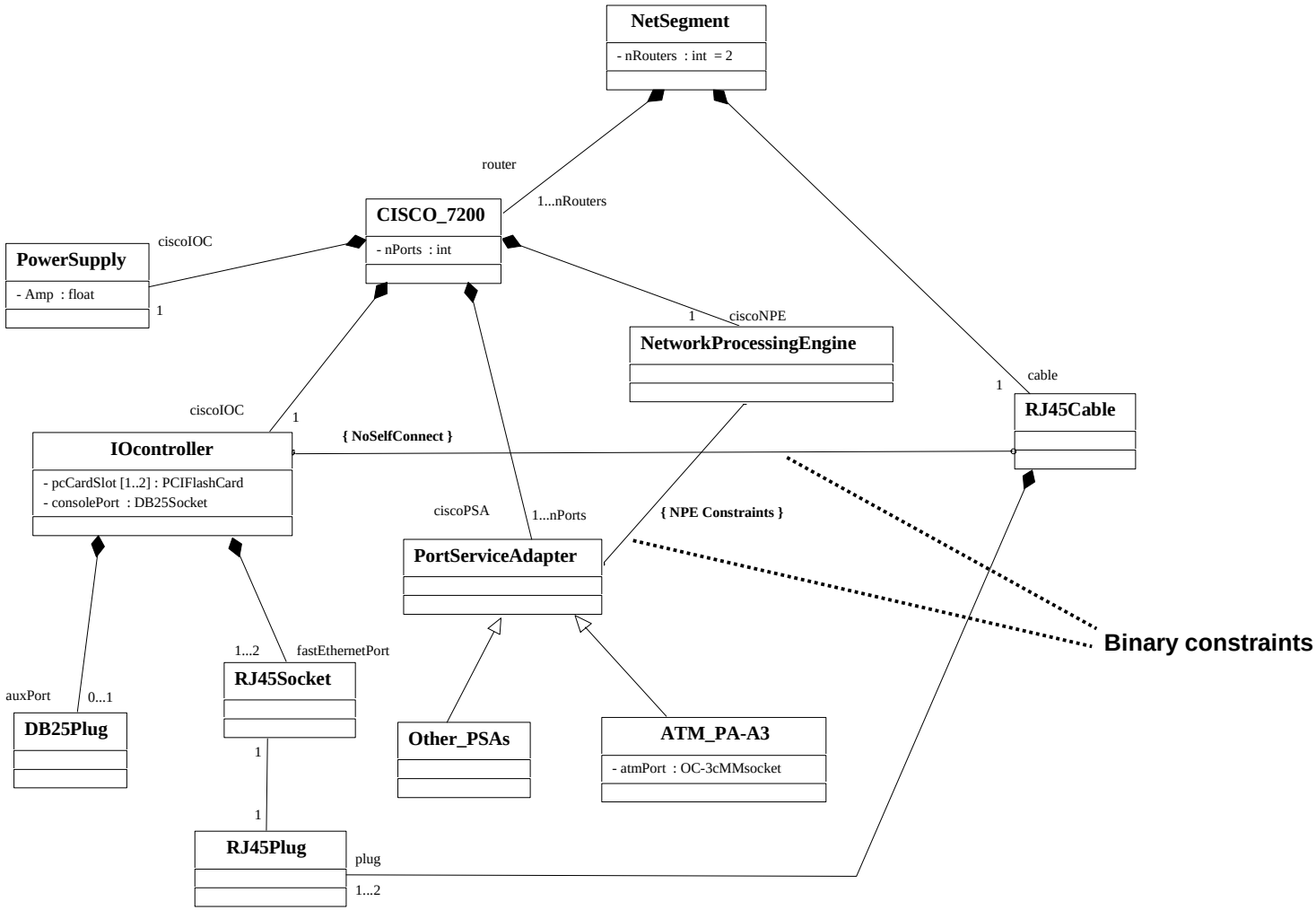


3 tiers, 30 cells/tier, ~600+ Cables, ~400 CPU's, 6 external subsystems,6-way cell connectivity

~ 20,000 objects and 10,000 connections represented on 7 1-page views

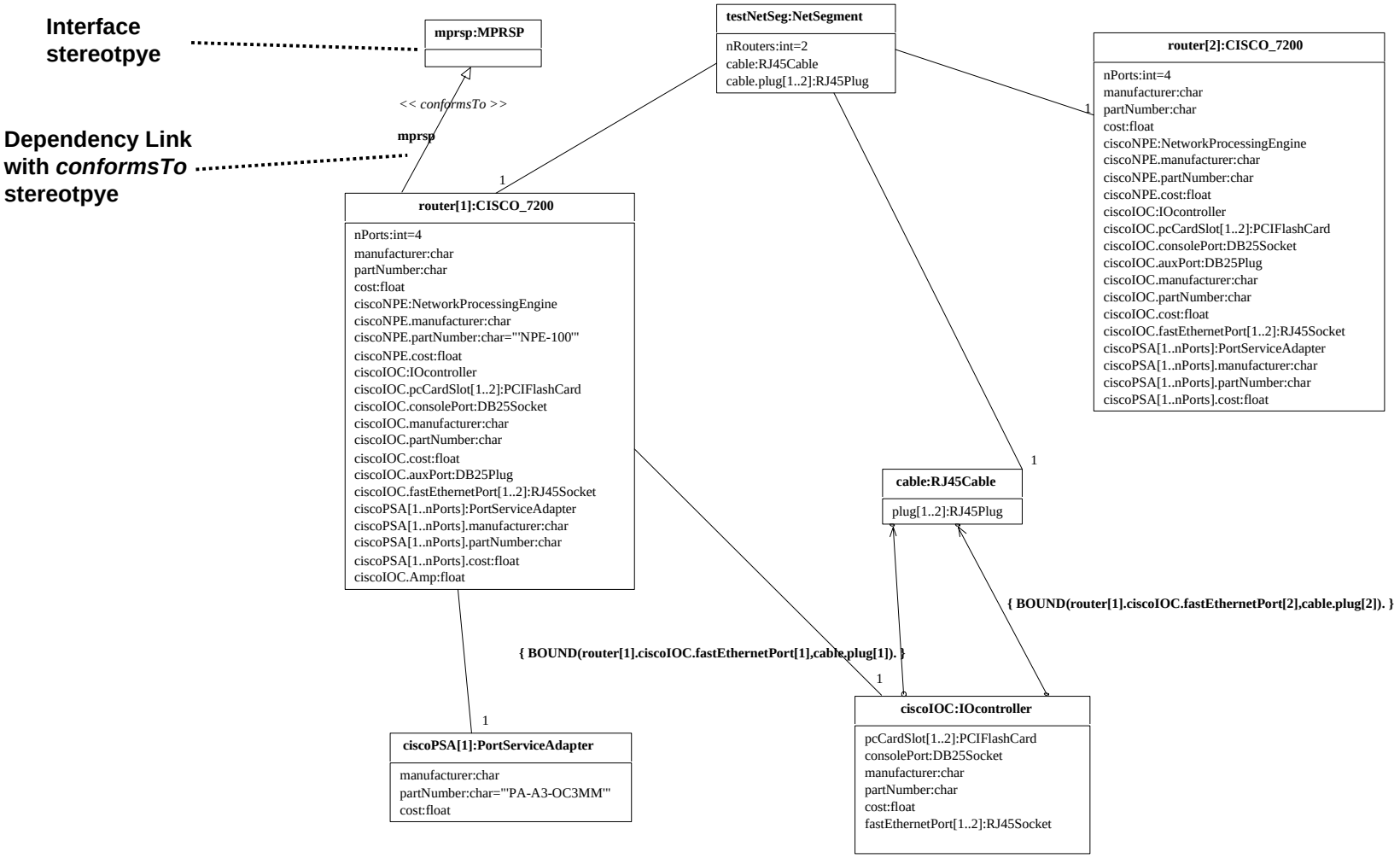
Class Model of Network Segment with Binary Constraints

AST Engineering Services

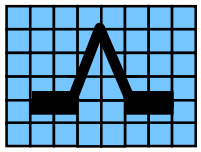


Instance Model of Network Segment with Dependency Link

AST Engineering Services



Prolog Encoding of Preceding Binary Constraint Expressions



AST Engineering Services

- **"NPE Constraint" text**

```
INVARIANT(  instr( [ [partNumber],[ciscoNPE] | X],_, 'NPE-100',ID1 ),
     instr([ [partNumber],[ciscoPSA,Slot] | X],_, 'PA-A3-OC3MM',ID2 ) ,
                    error(301,[ID1,ID2]), fail. ).
```

- **"NoSelfConnect" constraint text**

```
INVARIANT(
   instr([[plug,1],[cable]|X],_,_,ID1), bnd(ID1,ID2), instr([[fastEthernetPort,J],[ciscoIOC]|Y],_,_,ID2),
   instr([[fastEthernetPort,K],[ciscoIOC]|Y],_,_,ID3),  bnd(ID3,ID4),  instr([[plug,2],[cable]|X],_,_,ID4),
    instr([[cable]|X],_,_,ID5),  instr([[ciscoIOC]|Y],_,_,ID6), error(302,[ID5,ID6]), fail.  ).
```

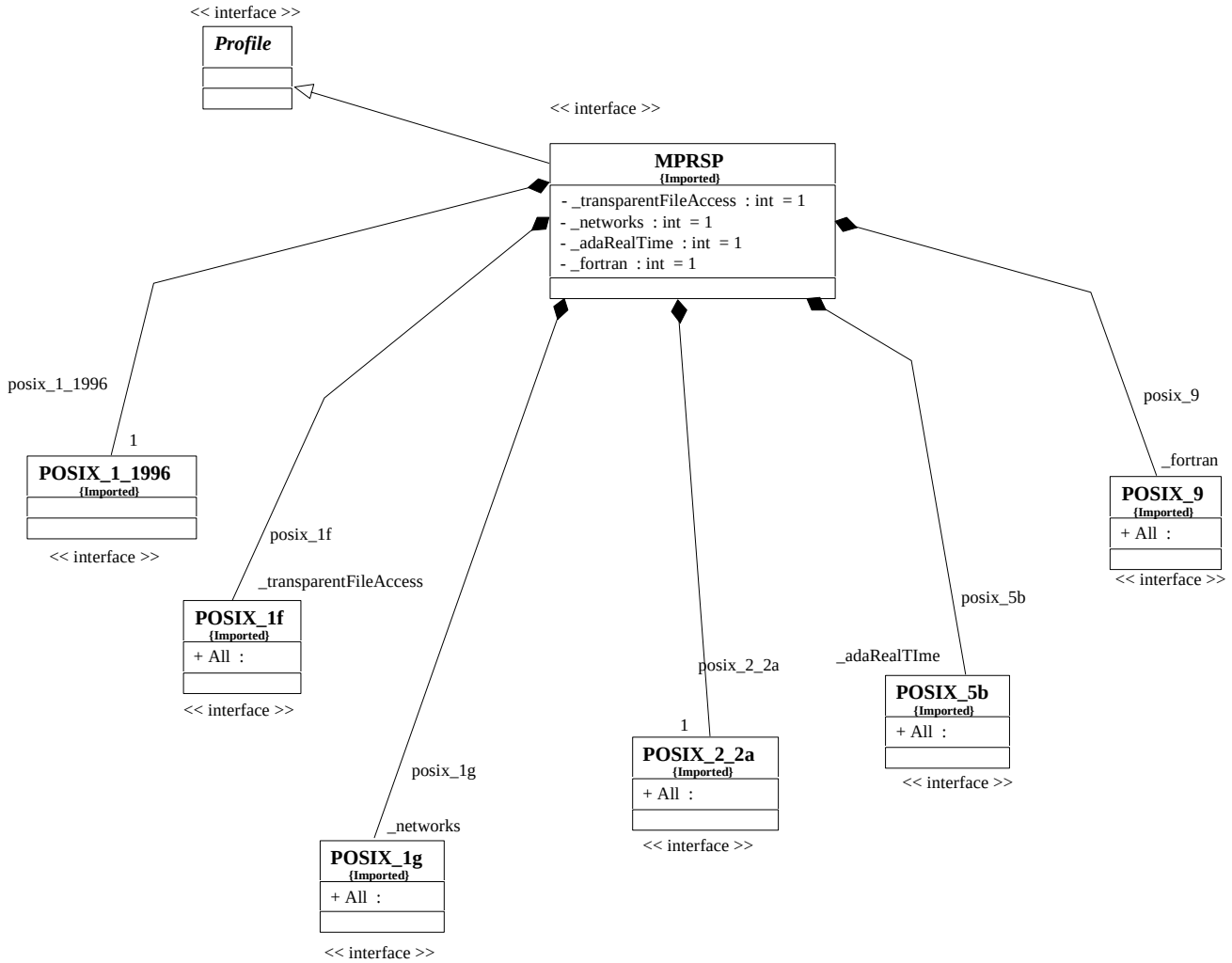
- **Run-time evaluation of constraint expressions**

```
Evaluating invariants ...
Error 301: Configuration Incompatibility - NPE-100 not valid with PA-A3-OC3MM
   <inst 59> [[testNetSeg],[router,1],[ciscoNPE],[partNumber]]
   <inst 50> [[testNetSeg],[router,1],[ciscoPSA,1],[partNumber]]
Error 302: I/O Controller is self-connected
   <inst 1> [[testNetSeg],[cable]]
   <inst 39> [[testNetSeg],[router,1],[ciscoIOC]]

   2 invariants evaluated.
```

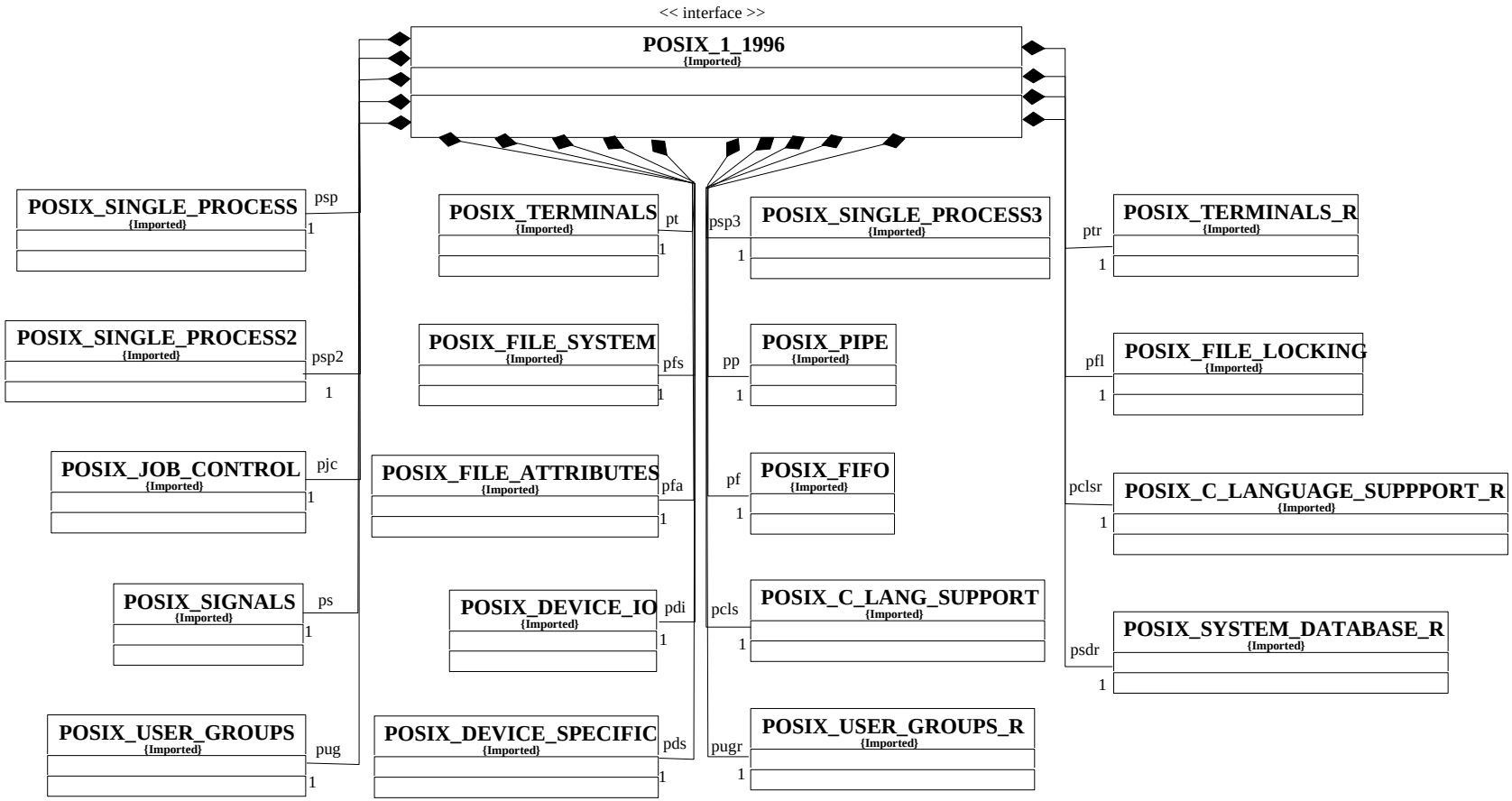
Example of POSIX MPRSP Specification (Multi-Purpose Real-Time System Profile)

AST Engineering Services

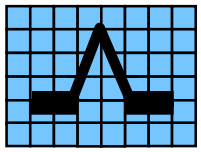


Composition of POSIX Profile Specification

AST Engineering Services



Implementation of Prolog Network Analysis Functions



AST Engineering Services

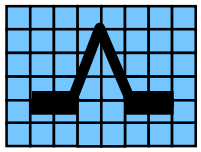
- **Numerous Prolog analysis functions are viable for implementation:**
  - **infer from network connections the necessary protocols of the host**
  - **determine the compatibility between cables and network elements**
  - **count the spare network connections**
  - **identify potential insertion points for switches or bridges**
  - **develop incremental plans for network evolution**
  - **produce a priced parts list for a given network configuration or network upgrade**

Implementation of Prolog Network Analysis Functions (Cont.)

AST Engineering Services

- identify limiting components with regard to performance or supportability
- assess compliance with network cable length restrictions
- identify single points of network failure
- maintain a list of the CPUs attached to each switch or hub port
- summarize the numbers of used and available local bus slots in each CPU by bus type
- identify which components are not compatible with a network protocol upgrade (e.g., 10BaseT to 100BaseT where some NICs are attached to ISA busses).

Commercial Network Management Applications

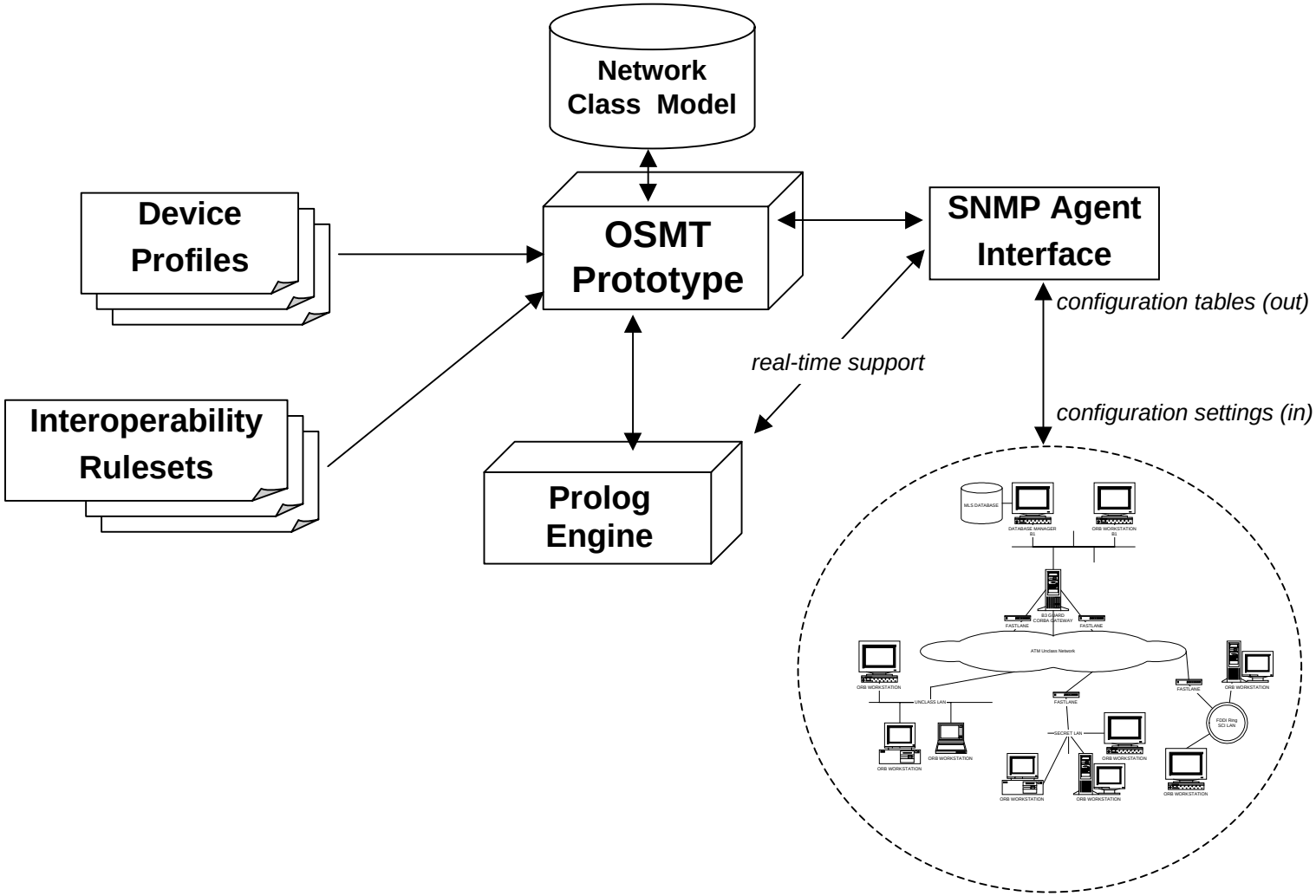


AST Engineering Services

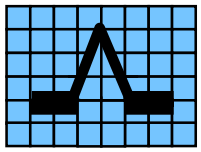
- **Large commercial networks present significant management difficulties to their operators**
  - **case study: major airline network engineer manually reset an airport's router incorrectly twice within 10 days**
  - **airport lost access to reservation system for several hours each time**
  - **resulting delays and rescheduling cost airline an estimated $6M**
- **Key issues**
  - **network complexity mentally overwhelming**
  - **device configurations must match system profiles as well satisfy interoperability constraints with neighboring devices**
    - **case study: airline network backbone routers set with legal but incompatible "hello timer" delay values**
    - **network oscillated between up and down status .. much finger-pointing**
    - **extensive network probing and analysis required to pinpoint the problem**
- **Critical need exists for network modeling and analysis tool to**
  - **periodicallly validate network component configurations**
  - **validate proposed network component configuration changes before implementation**
- **Existing commercial and custom tools**
  - **lack a formal system representation**
  - **don't scale**
  - **can't handle rule complexity**

Real-Time Network Management Concept

AST Engineering Services



Conclusions



AST Engineering Services

- **Problems identified in UML Specification have been reconciled**
  - **attribute and composition semantics**
  - **applicable to method assertions as well**
  - **plan to submit a change proposal to OMG**

- **Demonstrated the ability to translate enhanced UML models into formal analysis scripts**
  - **integrated Prolog development environment with commercial UML CASE tool**

- **Demonstrated synergy of UML-based modeling with Prolog analysis environment**
  - **wide assortment of viable applications identified**
  - **without our UML extensions, many of these applications would not be feasible**

- **Utility of commercial UML modeling tools significantly expanded**
  - **UML demonstrated as a viable ADL**
  - **commercial tool support for instance model development needs improvement**