Ontology-Enabled Traceability Mechanisms

Mark Austin and Cari Wojcik,

Department of Civil Engineering and Institute for Systems Research,

University of Maryland, College Park, MD 20742.

Presentation at INCOSE, Chicago, July 13, 2010.

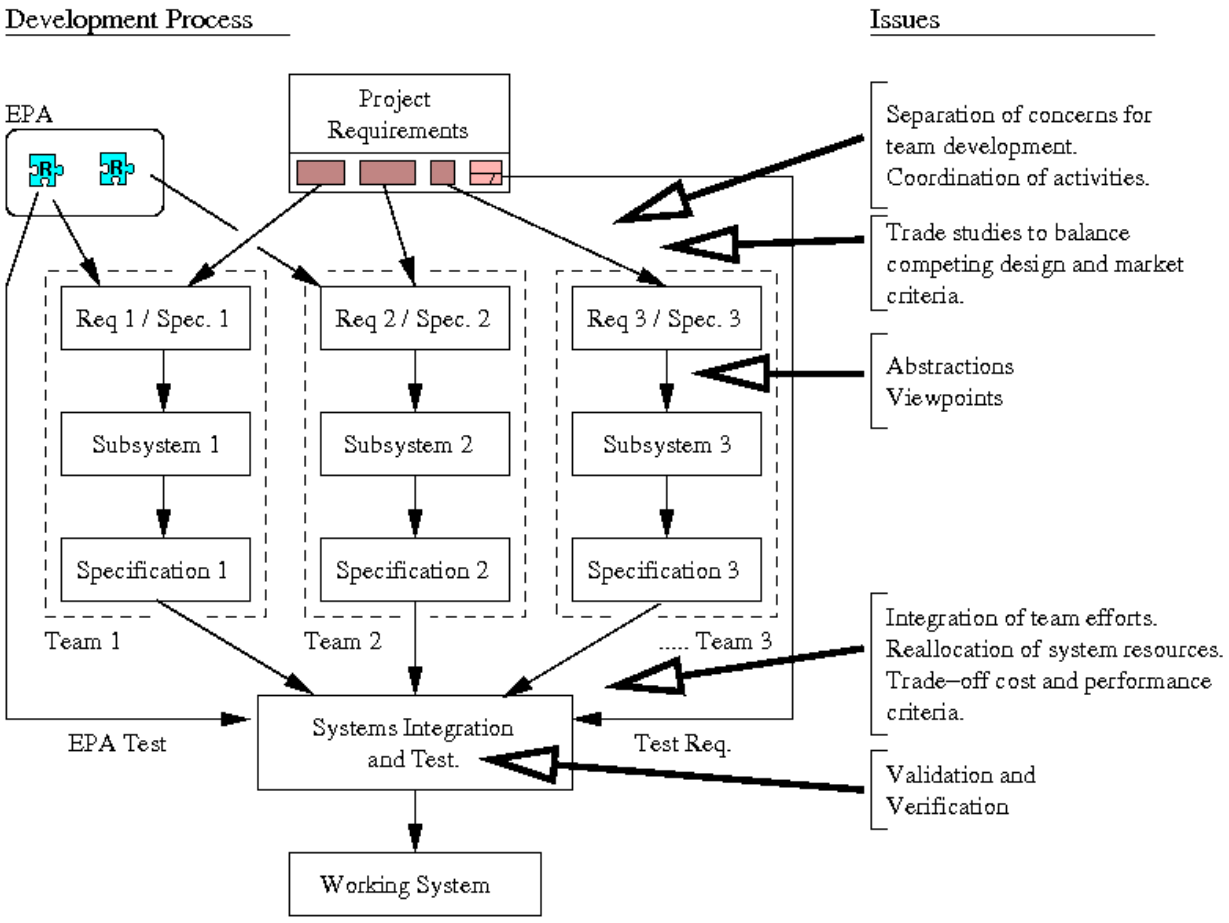
Outline

- Motivating Tenets
- Research Objectives and Approach
- Transfer of Semantic Web Technologies to Requirements Engineering
- State-of-the-Art Traceability (with SLATE)
- Improving upon State-of-the-Art Traceability
- Ontology-Enabled Traceability Mechanisms
- Support for Multiple-Viewpoint Design
- Prototype Implementation: Ontology-Enabled Traceability for the Washington D.C. Metro System
- Current Work
- Future Work and Expected Benefits

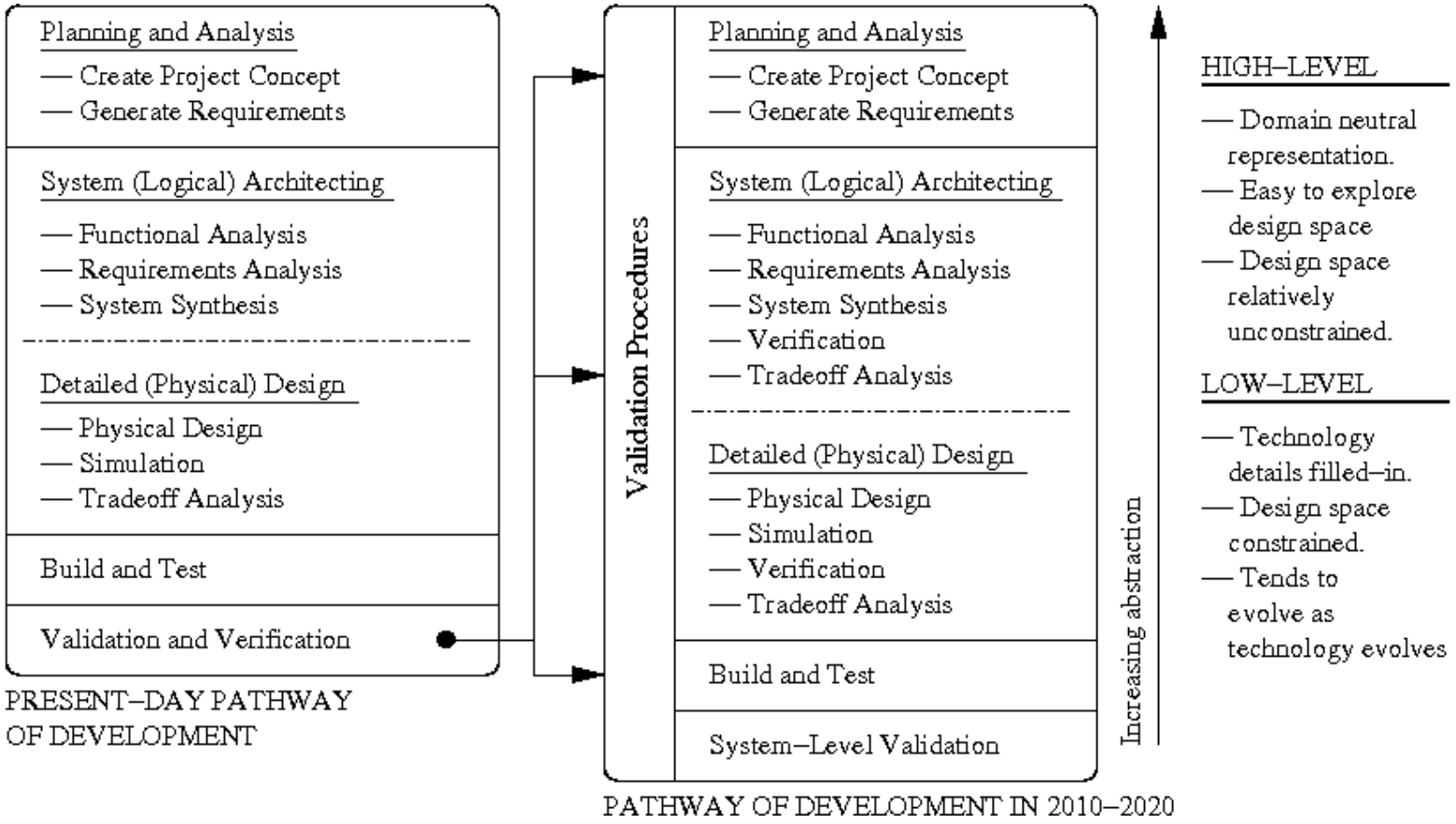
Tenet 1: SE needs to support Team-Based Development

Note: Project development is a SoS problem because the EPA is a separate, independent, process.

SoSs are characterized by three things: (1) emergent properties not part of the individual systems, (2) Heterogeneity of the systems in the SoS configuration and (3) behaviors that evolve over time.

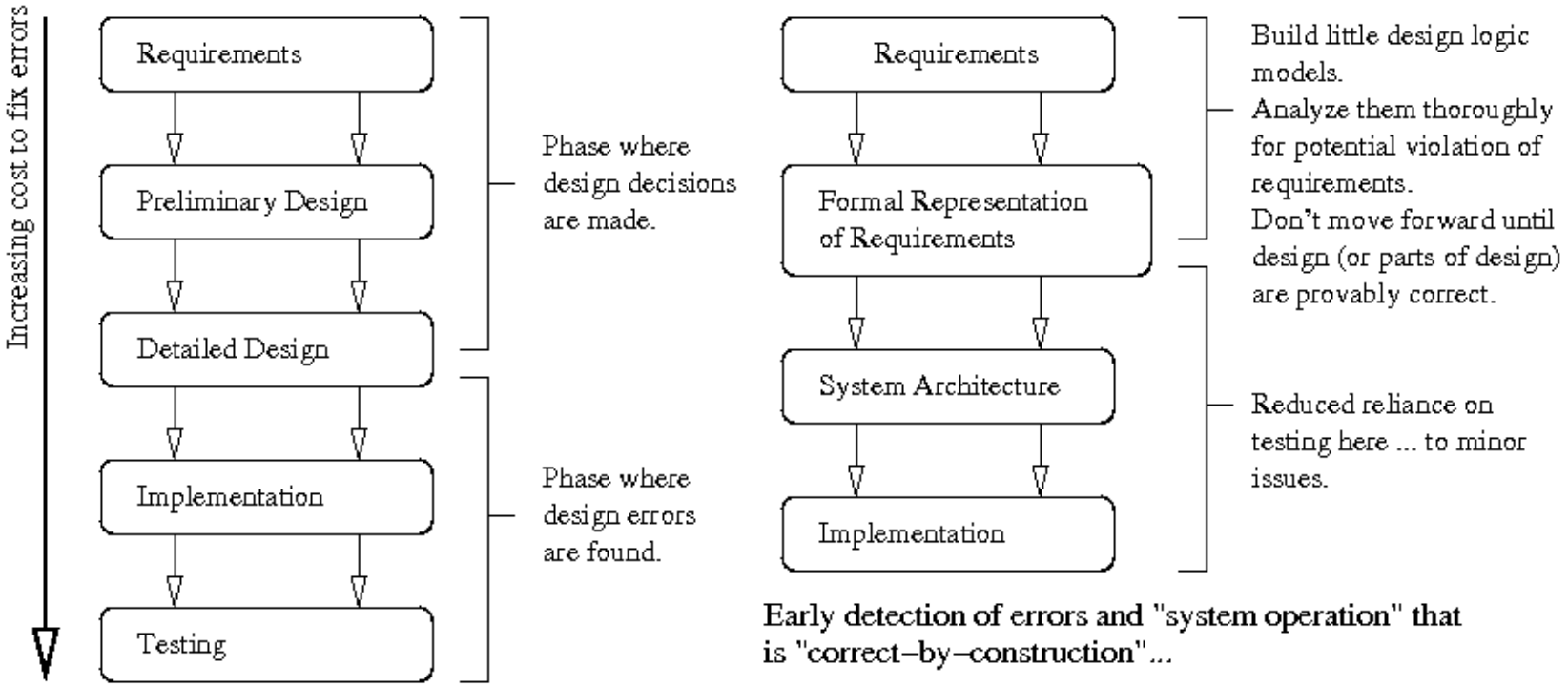


Tenet 2: Validation/Verification needs to be an integral part of the Systems Engineering Lifecycle



Tenet 3: Formal Approaches to Validation/Verification

We need formal methods to keep the complexity of design activities in check.



Research Objective and Approach

Research Objective

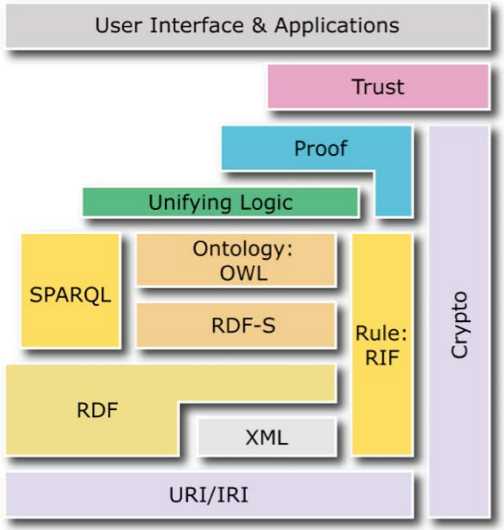
Explore benefits of ontology-enabled traceability mechanisms for team-based design and management of SoS.

Observation

The Internet and “project development problems” are both chaotic systems of systems.

Our research approach:

Compare the needs of a requirements engineering system to the Internet and look for solutions along parallel lines of thought.



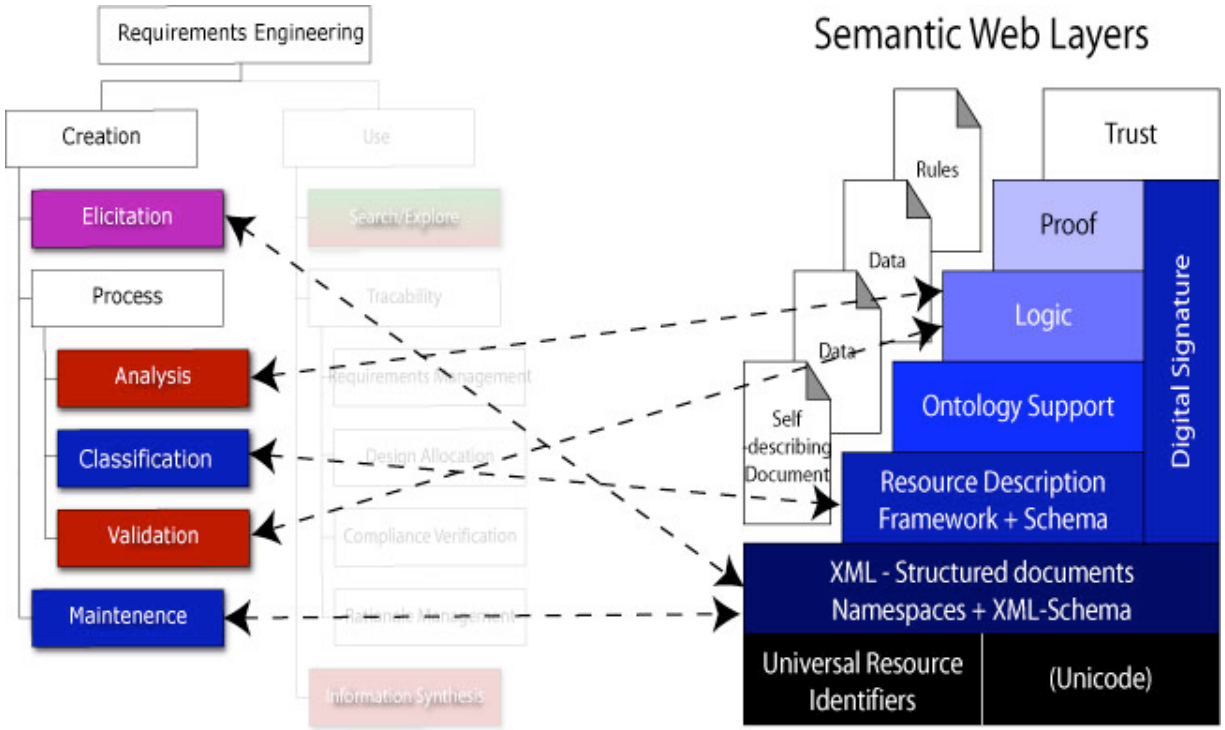
Goals of the Semantic Web:

...give information a well-defined meaning, thereby creating a pathway for machine-to-machine communication and automated services based on descriptions of semantics.

Note: Requirements and UML/SysML diagrams can be encoded in XML and RDF.

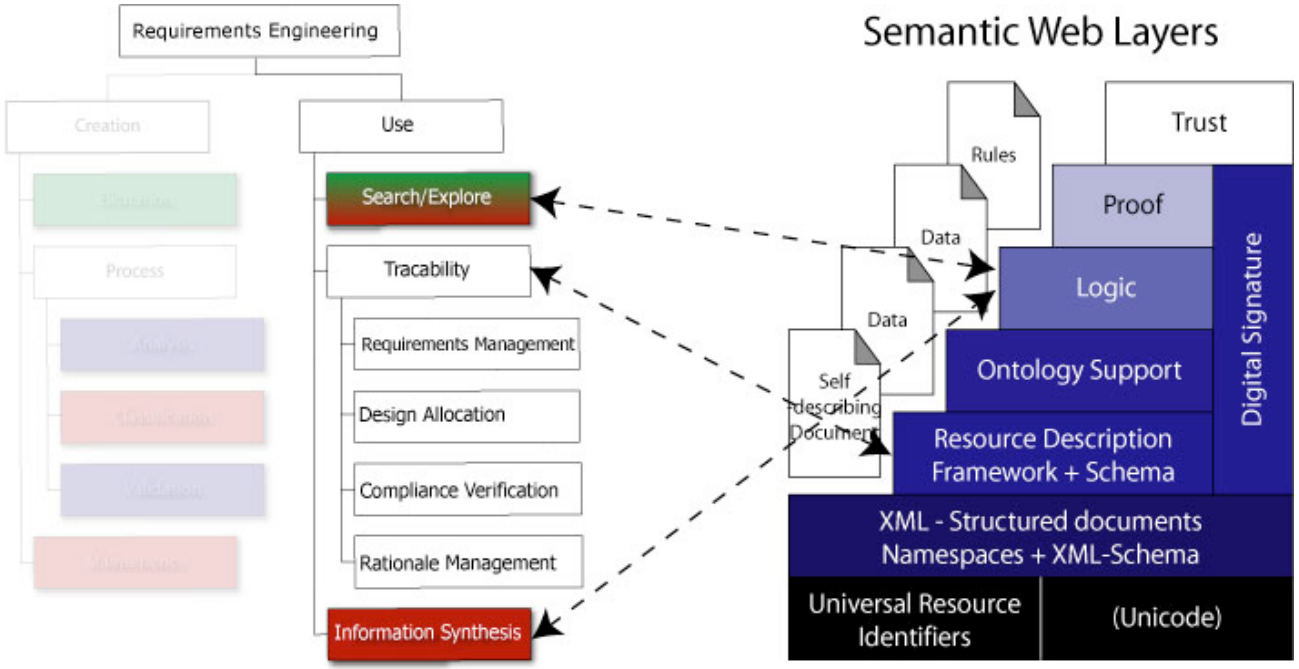
Transfer of Semantic Web technologies to Requirements Engineering

Starting point: Identify tasks associated with requirements creation and required support in the Semantic Web Layer Cake.



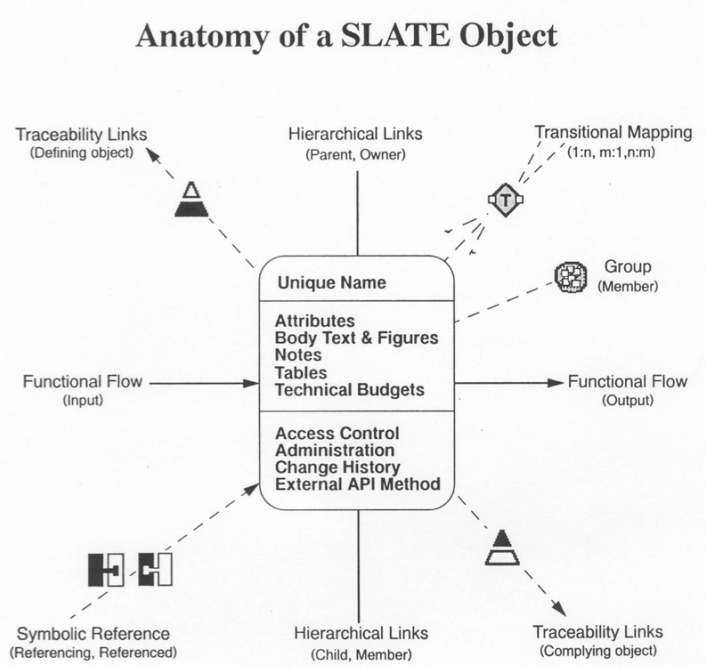
Transfer of Semantic Web technologies to Requirements Engineering

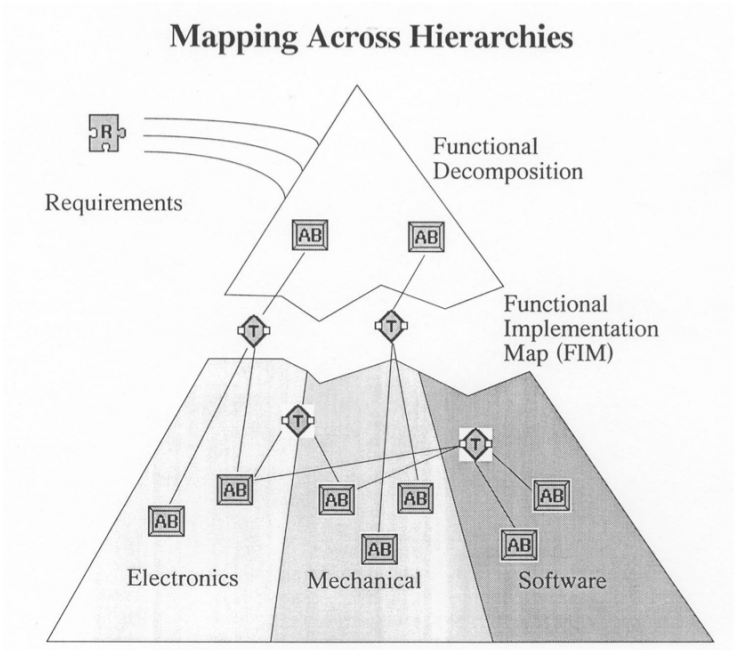
Starting point: Identify tasks associated with requirements usage and required support in the Semantic Web Layer Cake.



State-of-the-Art Traceability

State-of-the-Art Traceability with SLATE..

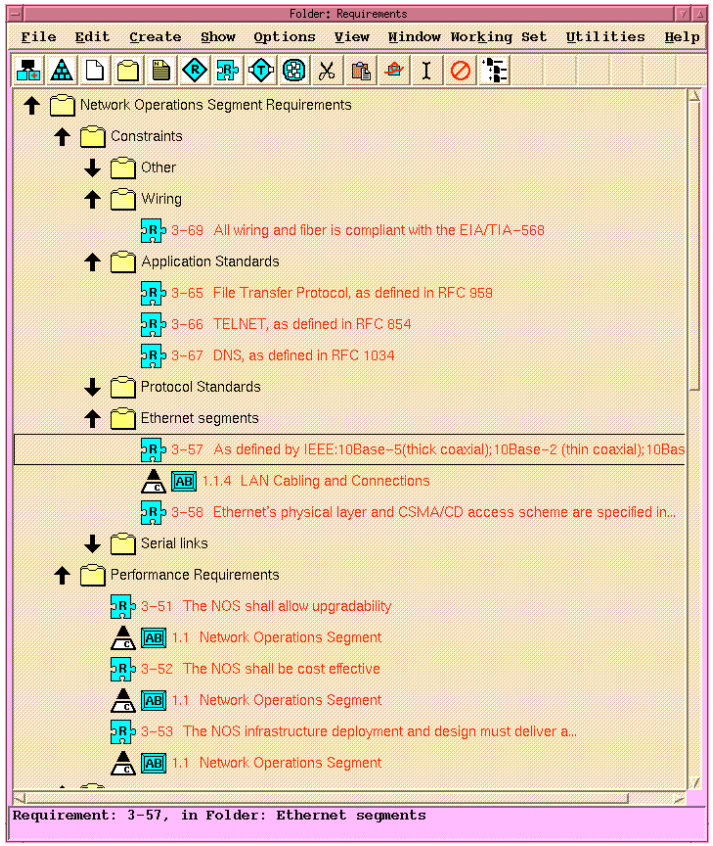


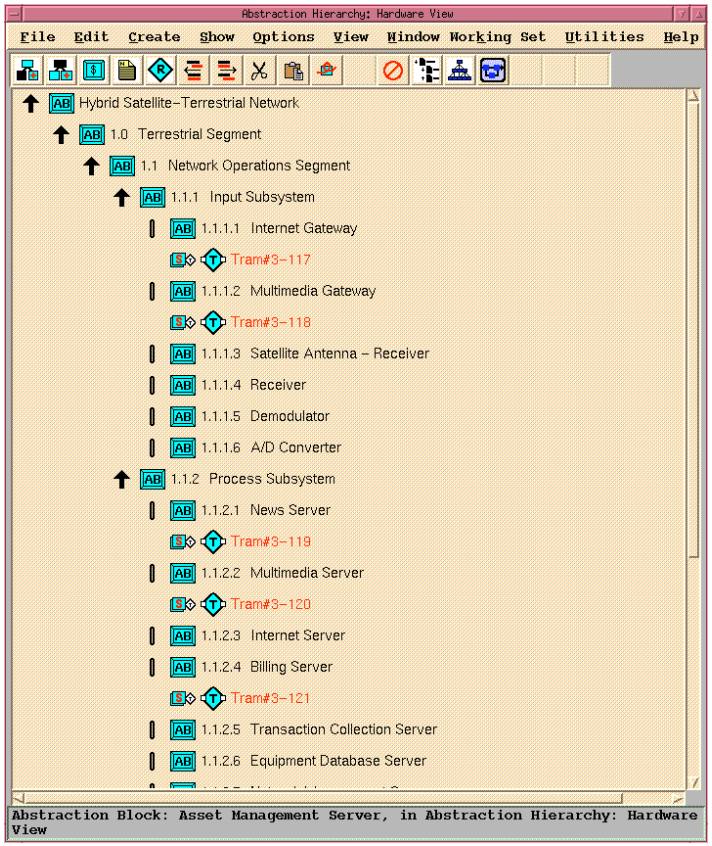


Note: Use of abstraction blocks only makes sense at the earliest stages of development, and where a system doesn't already exist. Doesn't apply for SoS.

State-of-the-Art Traceability

Visualization of traceability relationships is far from intuitive.





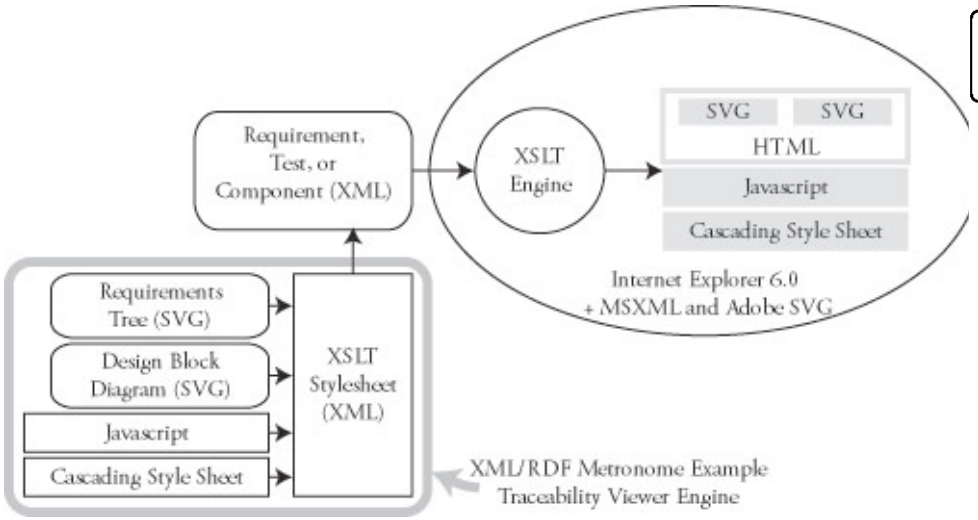
Most engineers want to visualize system developments using notations they are familiar with.

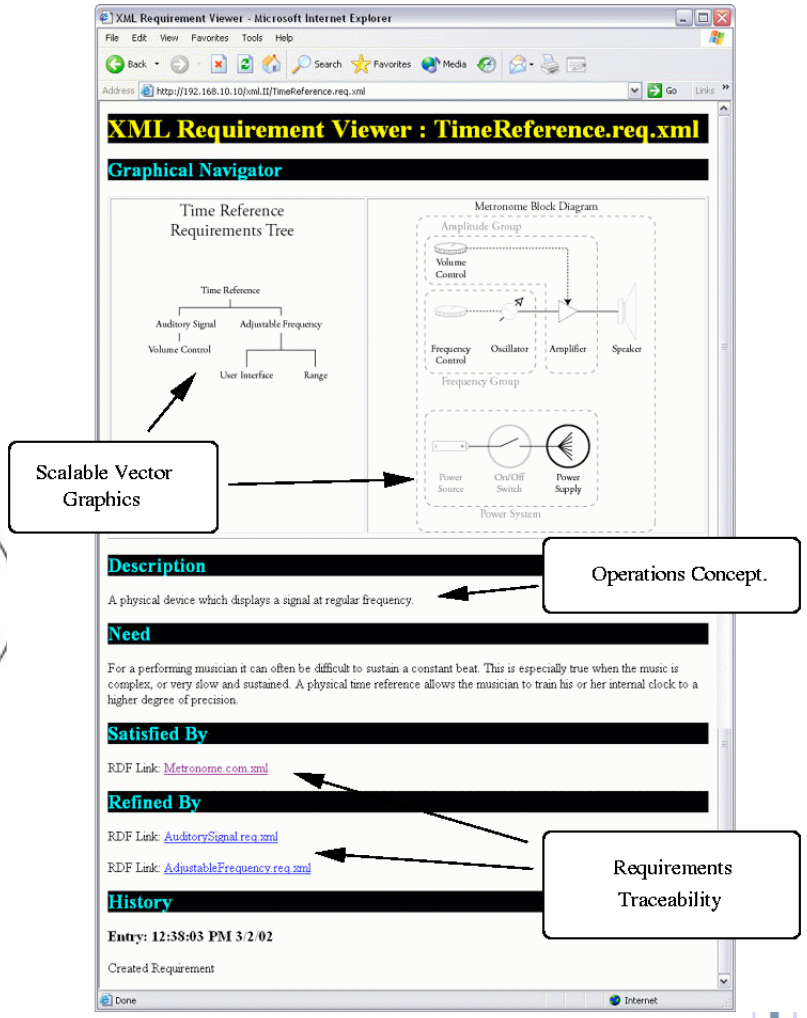
Improving upon State-of-the-Art Traceability

Surely we can do better!!!

Our first step: Explore use of XML and RDF technologies to improve visualization of requirements traceability.

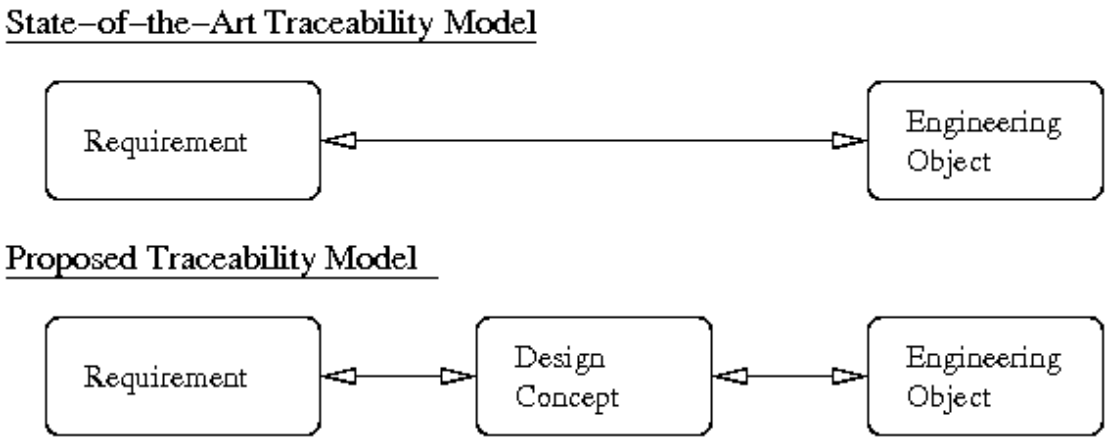
Credit: Web prototype developed and implemented by Scott Selberg in 2003.





Here's what's new …..

New idea: Ontology-enabled Traceability Mechanisms.

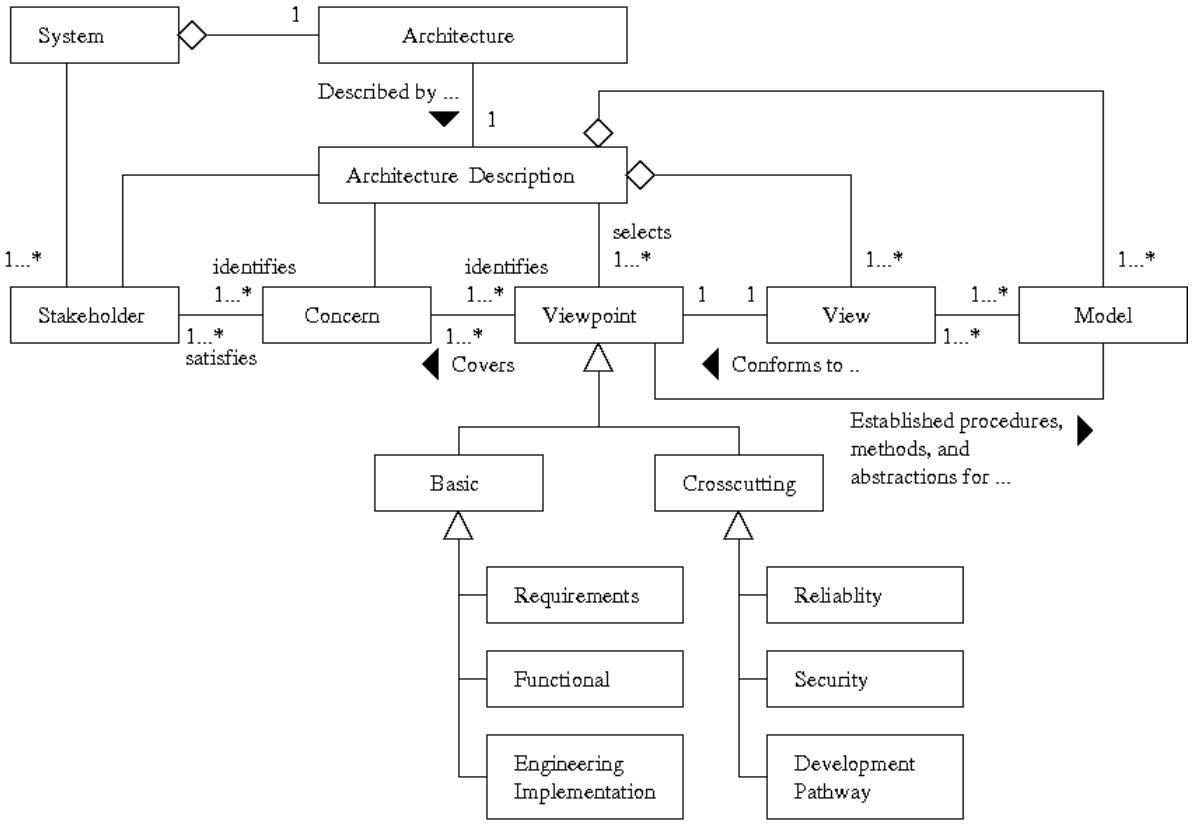


Approach: **Requirements are satisfied through implementation of design concepts**. Now traceability pathways are threaded through design concepts.

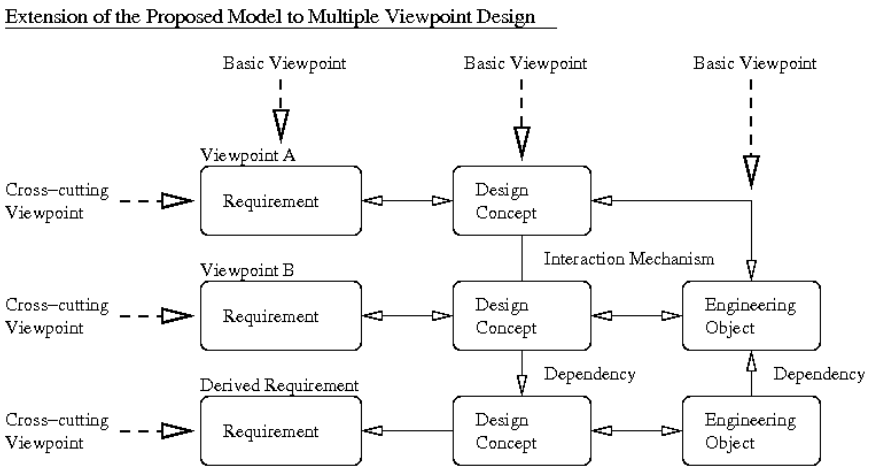
Key Benefit: Rule checking can be attached to "design concepts" – therefore, we have a pathway for **early validation**.

Support for Multiple-Viewpoint Design

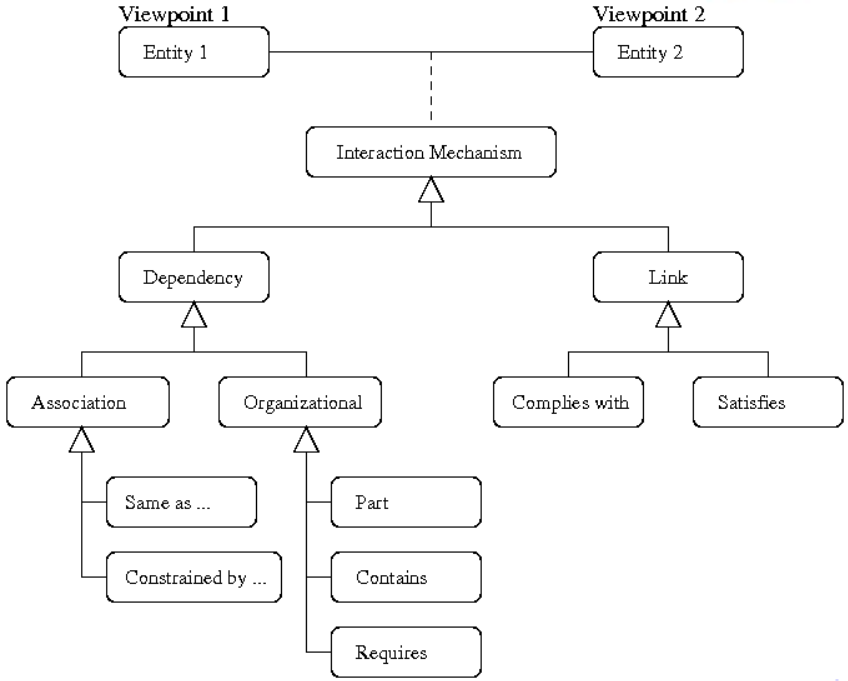
Team-based design is a multi-disciplinary activity. We need a model for multiple-viewpoint design and mechanisms for capturing interactions between design concerns.



So how might ontology-enabled traceability for multiple-viewpoint design work?



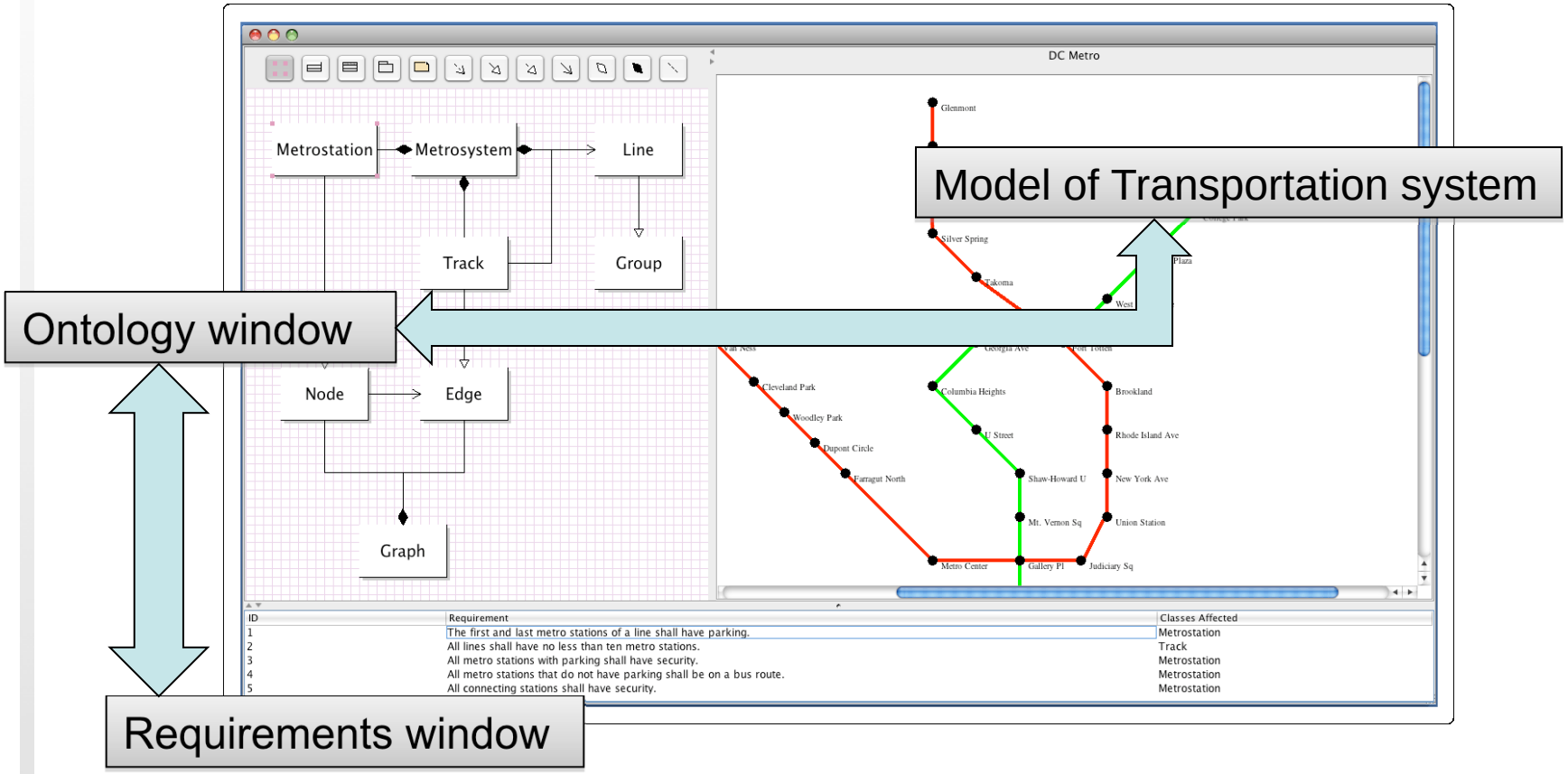
Multiple-viewpoint ontology-enabled traceability will correspond to graph of design entities: requirements, ontologies, and engineering objects.



We need models to capture the various mechanisms of interaction between viewpoints.

Prototype Implementation: Ontology-Enabled Traceability for Washington D.C. Metro System.

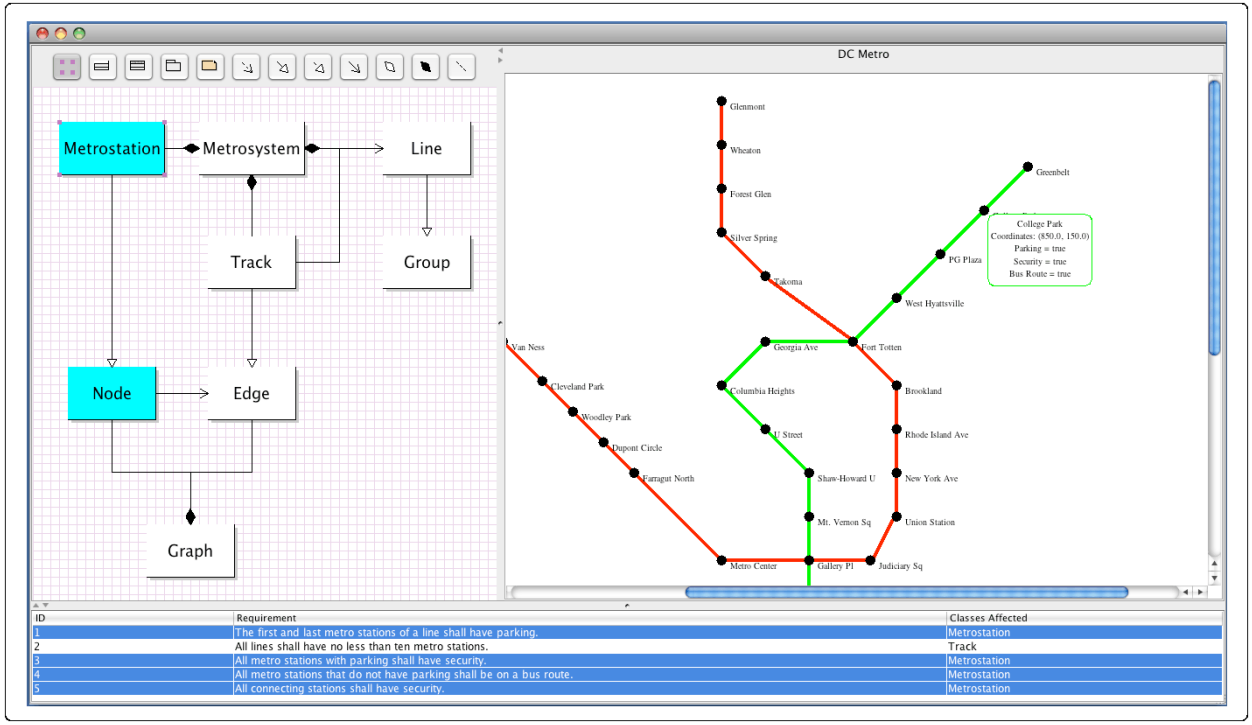
Very simple. UML representation for one ontology. All traceability relationships are hard-coded. Visualization cuts across stages of system development.



Credit: Cari Wojcik, MS Thesis, 2006.

Prototype Implementation: Ontology-Enabled Traceability for Washington DC Metro System.

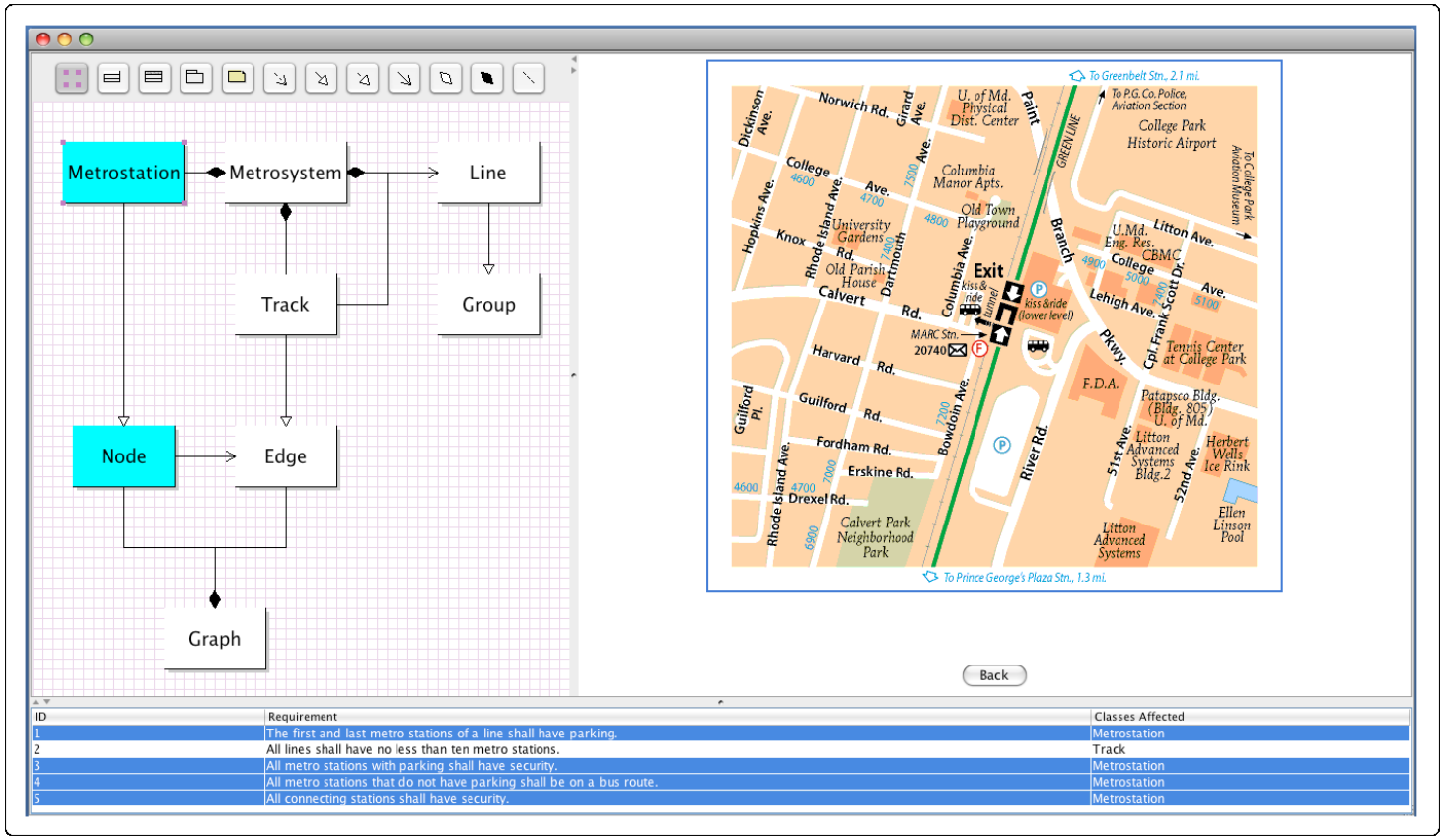
Designers are provided with mechanisms to interact with the system in multiple ways.



Traceability relationship from the College Park Metro Station back to defining design concepts (MetroStation and Node) and defining requirements.

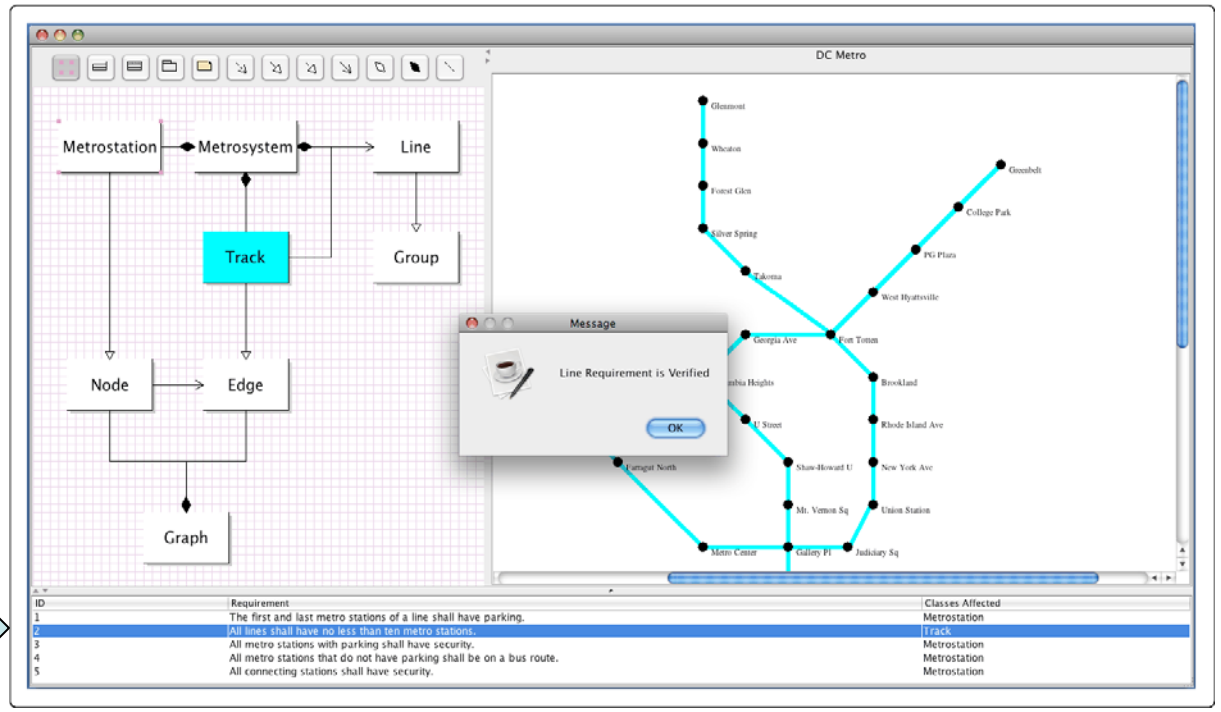
Prototype Implementation

Detailed Map View of the College Park Metro Station



Prototype Implementation: Ontology-Enabled Traceability (with very basic rule checking).

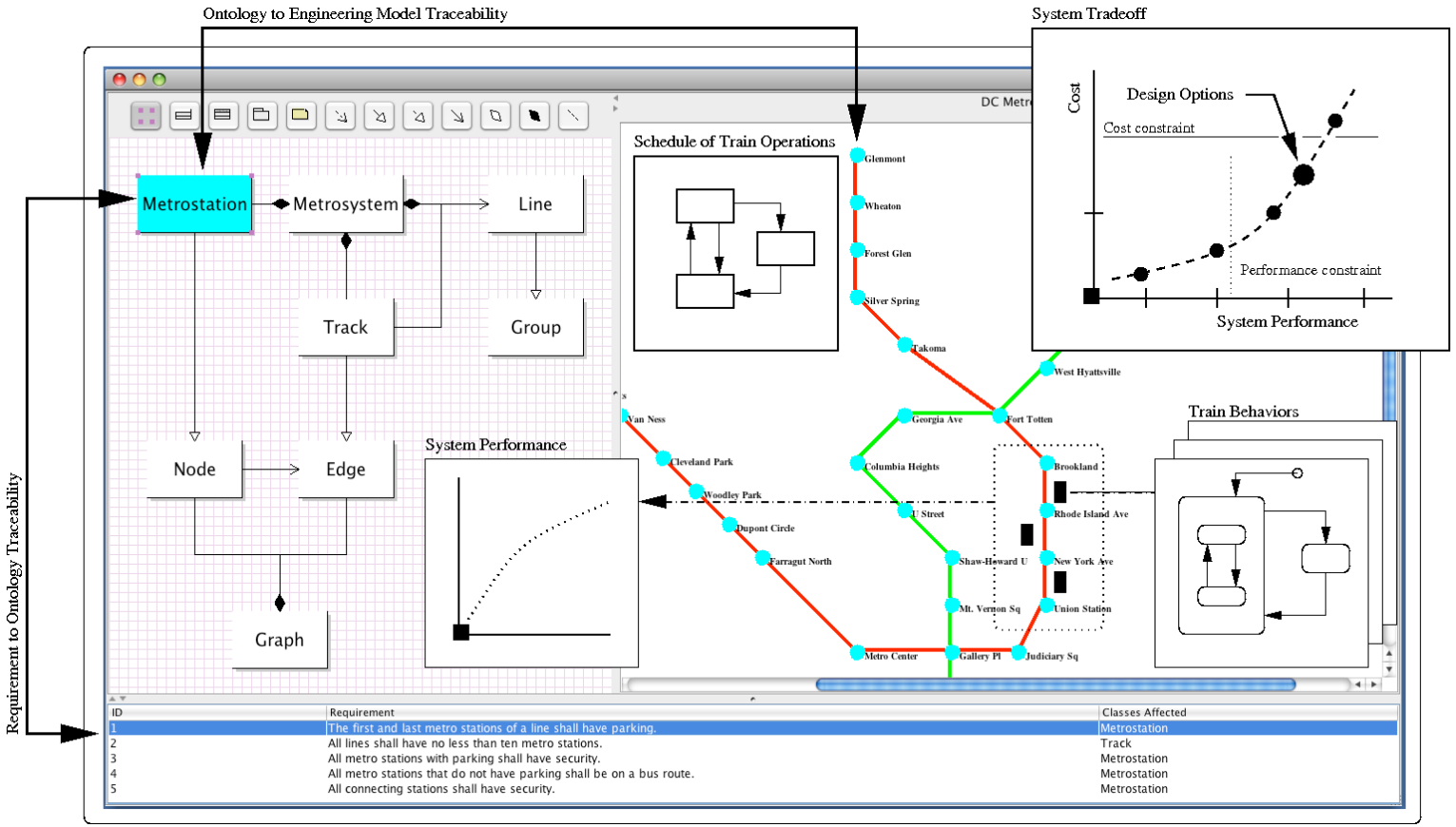
Key Advantage: **Design rules and procedures for design rule checking can be attached to ontologies**.



Design rule checking is triggered by double clicking on a requirement. Visualization shows the extent of ontologies and engineering entities involved in the rule checking.

Current Work

Current work: Re-design implementation to maximize use of software design patterns. Add train behaviors. Student: Parastoo Delgoshaei, MS Thesis.



Future Work and Potential Benefits

Proposed Work:

1 Explore feasibility of extending ontology-enabled traceability mechanisms to multiple-viewpoint design,

2 Explore use of Semantic Web Technologies (e.g., OWL = Web Ontology Language and SWRL = Semantic Web Rule Language) for representation of ontologies and rule-checking,

3 Design software infrastructure to conduct system trade studies.

4 Design and implement a scalable, networked, system implementation.

Potential benefits/payoffs?

Fewer design/management errors due to superior representation of traceability relationships; built-in support for design rule checking at the earliest possible moment; improved economics of SoS development and management.

The End!

Questions?