



Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02

Standardizing Variability – the process towards CVL – a Common Variability Language

by

Øystein Haugen, SINTEF
oystein.haugen@sintef.no



Outline of the talk

- **Motivation**
 - Technical
 - Political
 - Commercial
- **The MoSiS Common Variability Language**
 - Language
 - CVL Tool
 - Canned demo
- **Standardizing CVL**
 - Why OMG?
 - OMG process
 - Initial submission

Motivation for standardizing CVL

- **Political / Commercial**
 - Interoperability
 - Opportunities for SMEs
 - Goals for projects – valuable deliverables
 - Value over time
 - beyond the scope of the individual European project
 - Quality label (?)
- **Technical**
 - Many dialects – more ambiguity
 - More tools
 - Bigger educational market
 - university
 - commercial courses
 - tutorials
 - Serious technical assessment



Eureka Σ! 3674 Programme, ITEA2 project ip066035

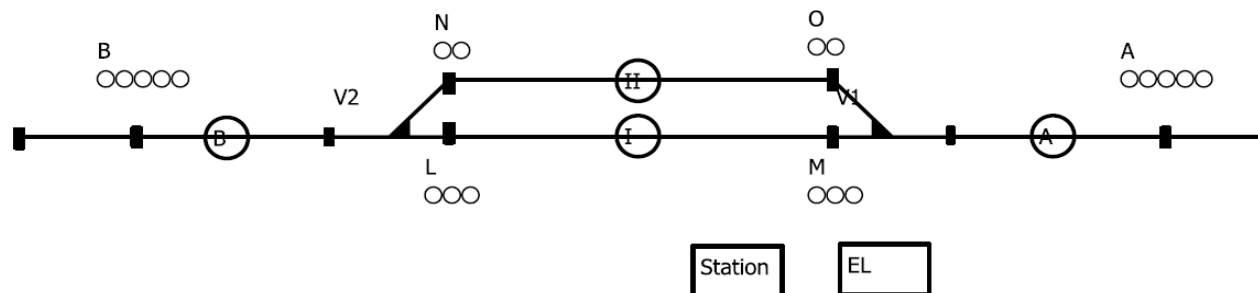
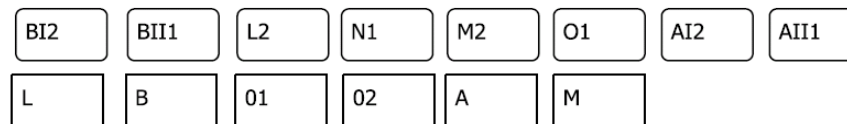
SINTEF / Øystein Haugen, 2010-10-02

MoSiS CVL

coming out of the ITEA project
MoSiS

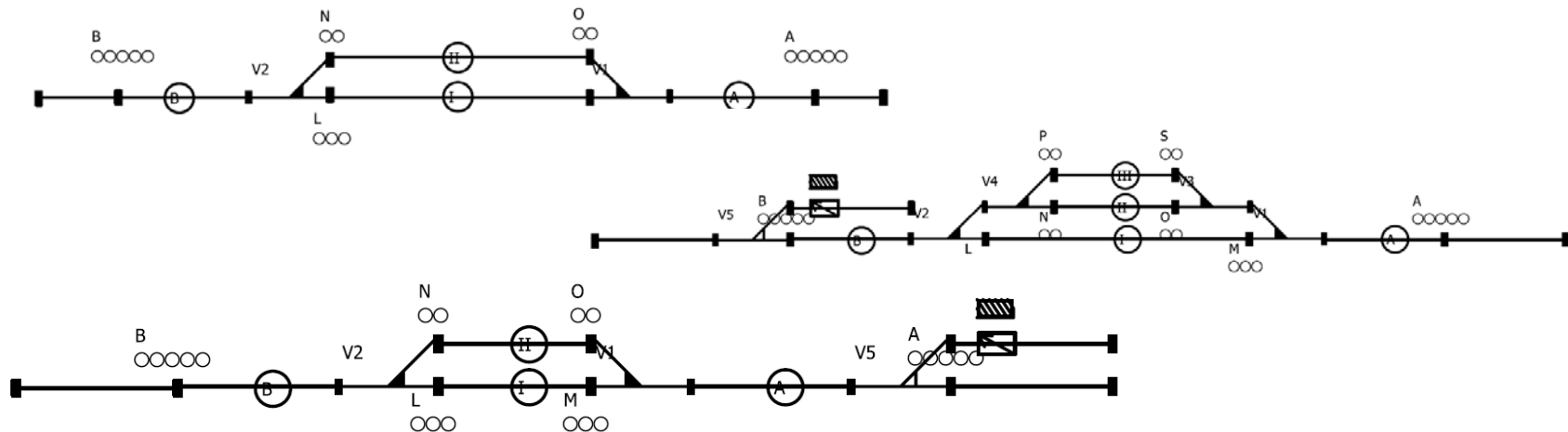
with SINTEF, Univ. of Oslo and ESI
as main contributors

TCL – Train Control Language



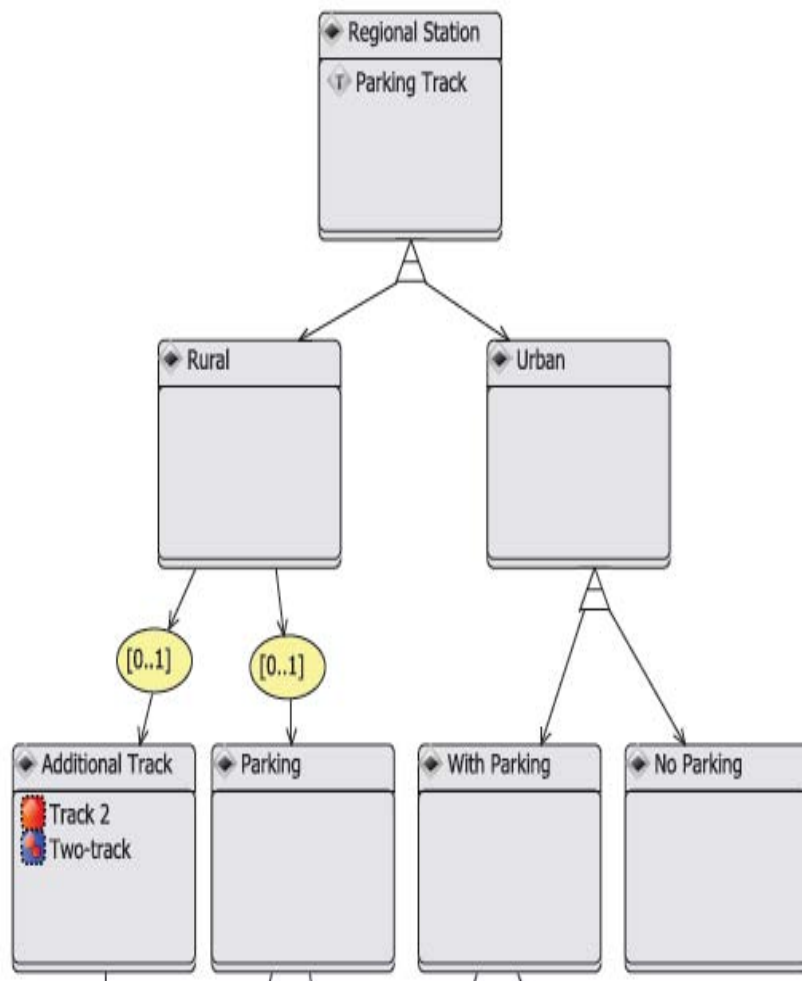
- **TCL represents train signaling domain knowledge**
- **Using TCL provides automation**
 - MoSiS has made code generators for 3 different kinds of representations
- **TCL is a domain specific modeling language**
 - MoSiS has created several such DSMLs

Several Similar Stations



- By using TCL you create each train station from scratch
- This means also maintaining each station separately
- and making test suites for each station separately
- How can we capture the obvious similarities between stations
 - beyond the concepts of the TCL language

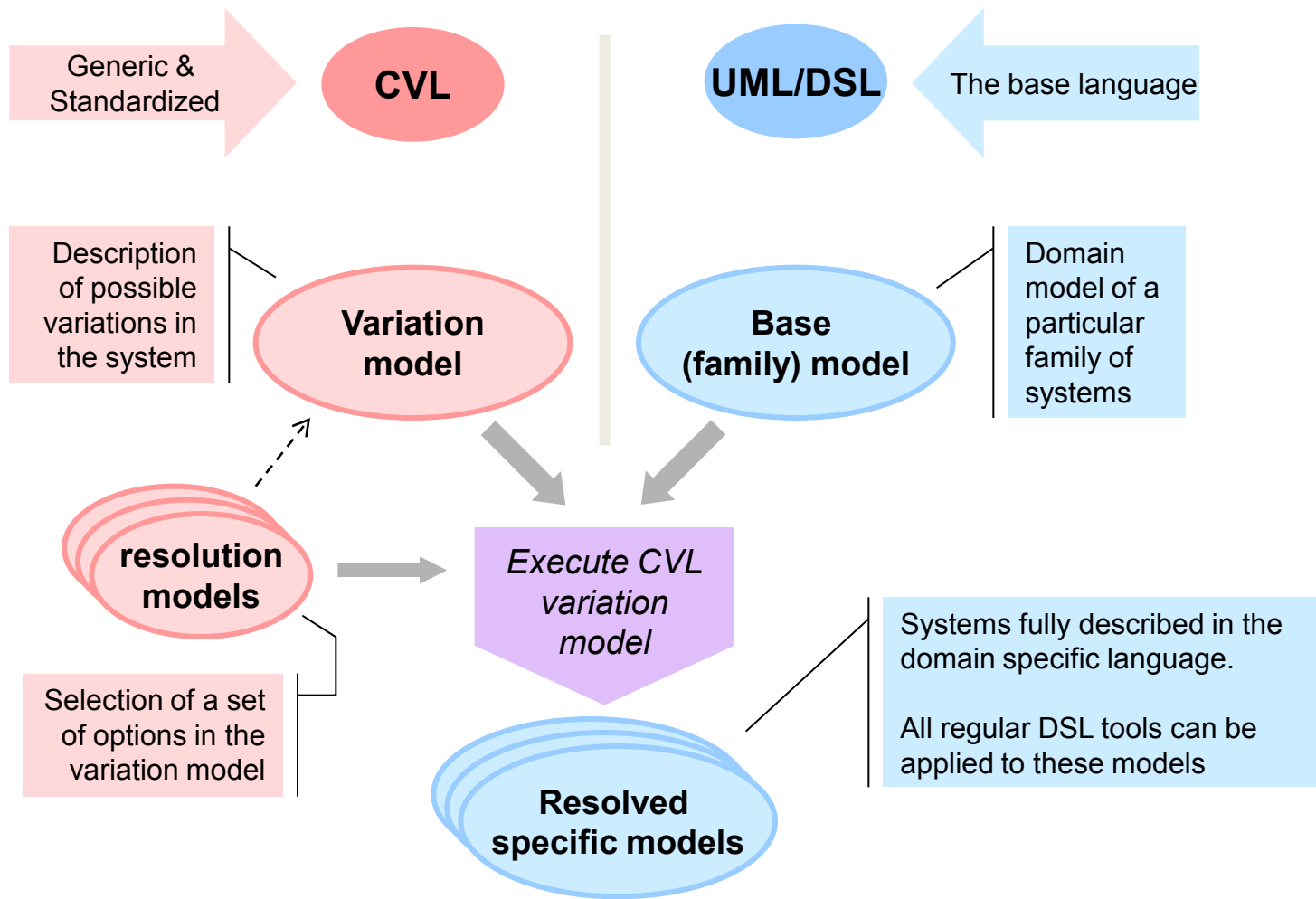
Feature Diagrams for variability



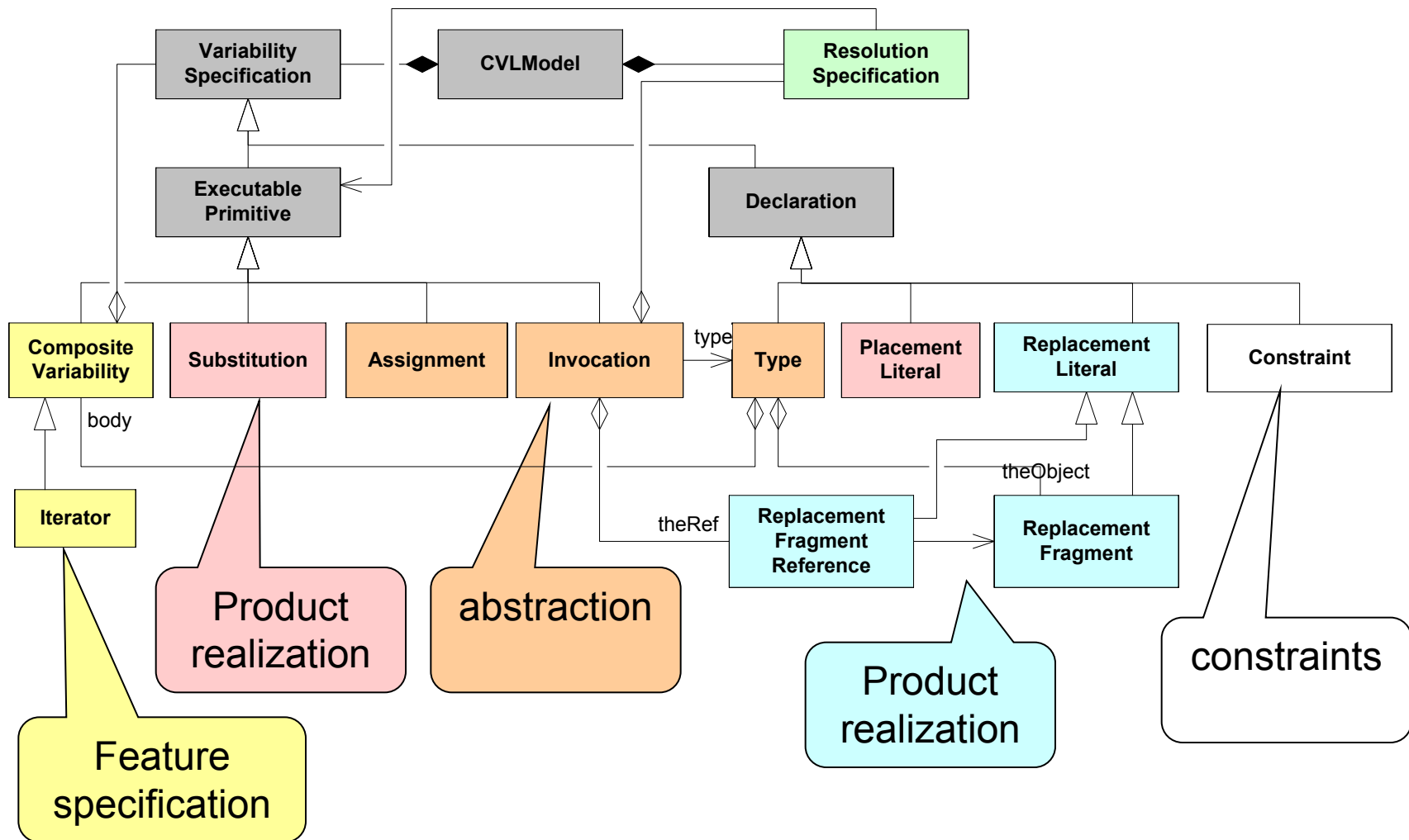
Two solutions to similarity

- **Include concepts for reuse inside the language**
 - add e.g. types or macros to TCL
 - blurs the domain specifics
 - requires deeper language engineering knowledge
- **Add similarity with a separate description**
 - CVL = Common Variability Language
 - Feature Diagrams as partial concrete syntax for CVL

Common Variability Language



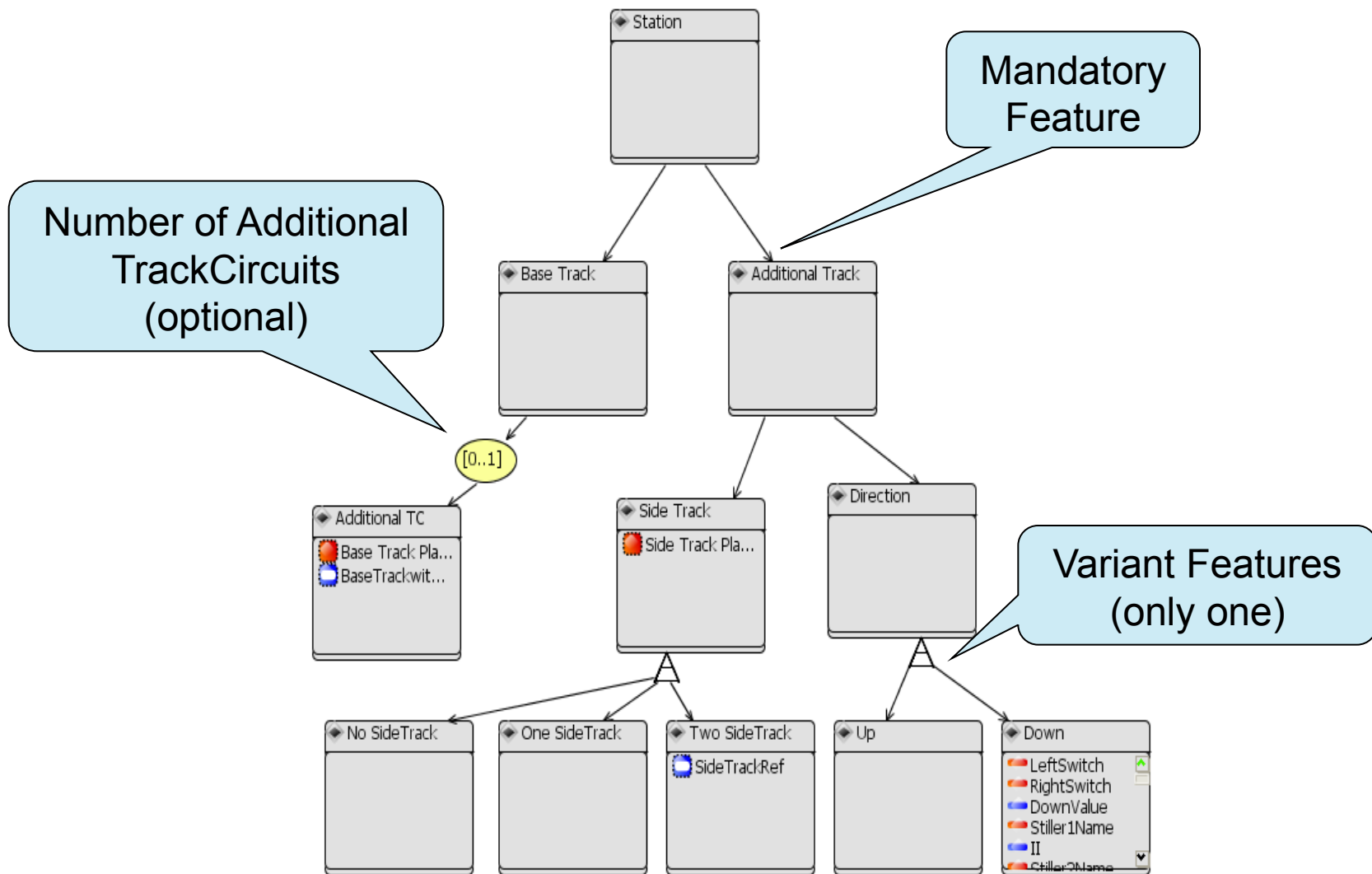
CVL Metamodel (important classes)



Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02

Feature Specification Layer








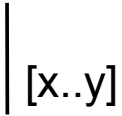
Number of Additional TrackCircuits (optional)

Mandatory Feature

Variant Features (only one)

Eureka Σ! 3674 Programme, ITEA2 project ip06035
 SINTEF / Øystein Haugen, 2010-10-02

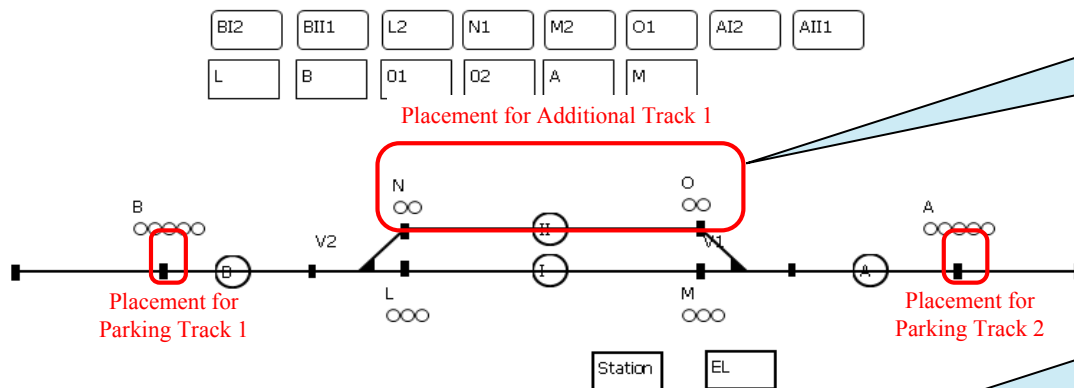
The Feature Diagrams – CVL mapping

Semantics	Symbol	CVL Element	Comment
Mandatory		<code>:CompVar</code>	<i>CompositeVariability</i> makes the resolution of a node mandatory for any instantiation
Optional		<code>:Iterator</code> Lower = 0 Upper = 1	The multiplicity [0..1] reflect the fact that the sub-node can be chosen or not.
AND		<code>:CompVar</code>	<i>CompositeVariability</i> makes the resolution of all sub-nodes mandatory for any instantiation
OR		<code>:Iterator</code> Lower = 1 Upper = -1 IsUnique = true	The multiplicity [0..*] means that any number of sub-nodes can be chosen. <i>IsUnique</i> specifies that each sub-node can only be chosen once (which corresponds to the usual OR semantics)
XOR		<code>:Iterator</code> Lower = 1 Upper = 1	The multiplicity [1..1] means that one and only one of the sub-nodes can be chosen (which corresponds to a classic XOR semantics).
Multiplicity		<code>:Iterator</code> Lower = x Upper = y IsUnique = true	The multiplicities associated to a choice are mapped to an Iterator containing the same multiplicities

Product Realization Layer

Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02



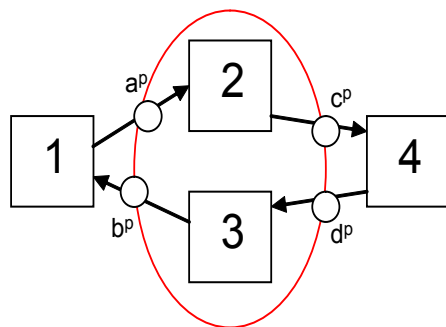
Placement =
the hole to fill

Replacement =
the filling source

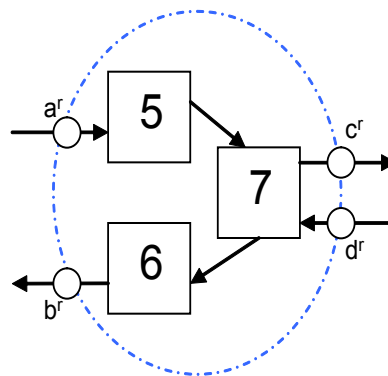
an internal
placement

How CVL relates to any base language

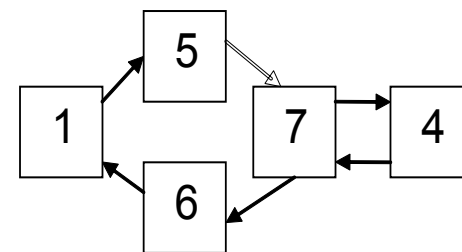
- Any base model defined in a MOF-defined base language can be seen as a general graph of objects and references
- All operations when executing CVL is on general graphs
- Type of the eObject can be obtained through reflection – for the purpose of type checking



Base model plus Placement Fragment



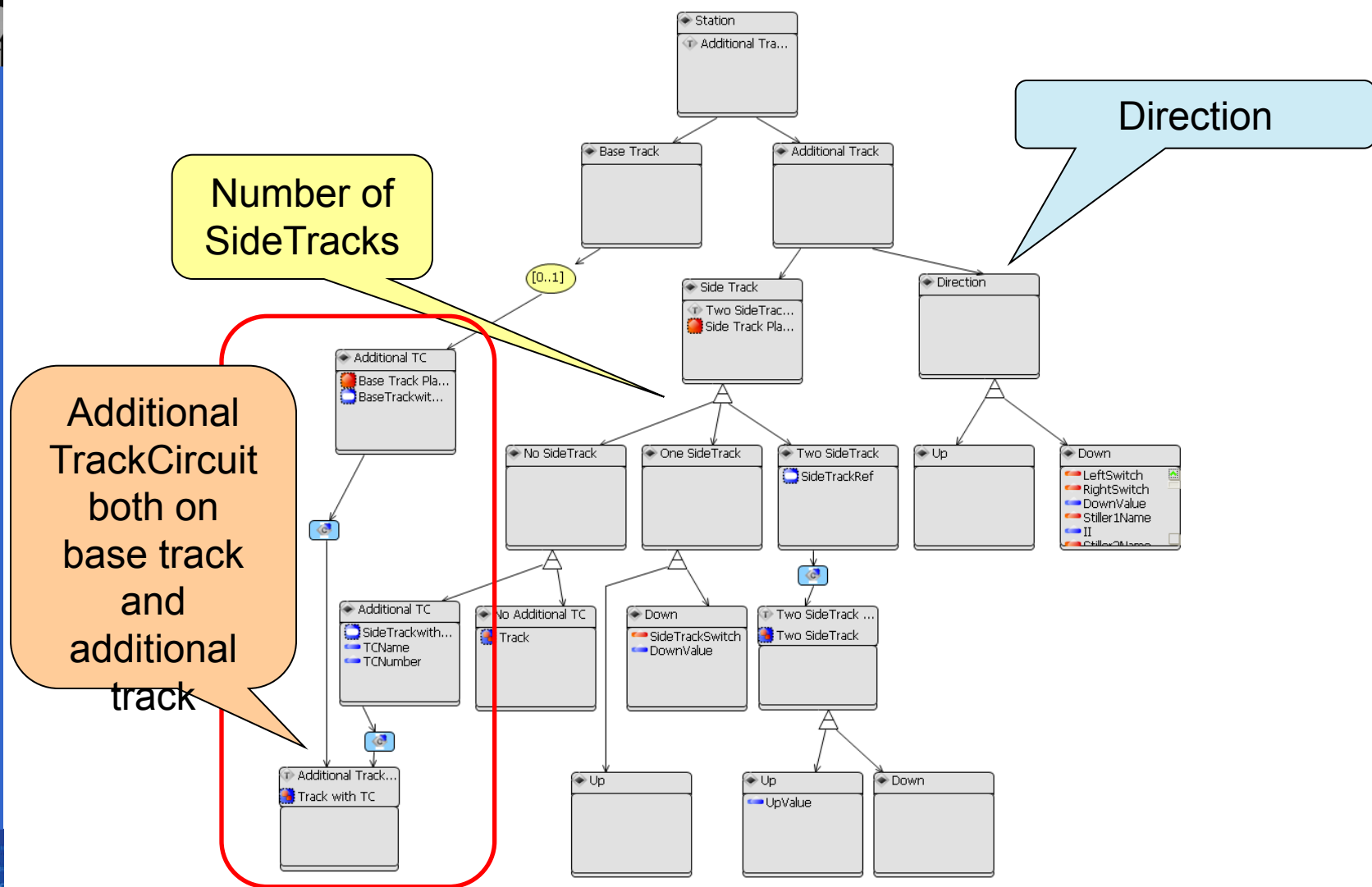
Replacement Fragment



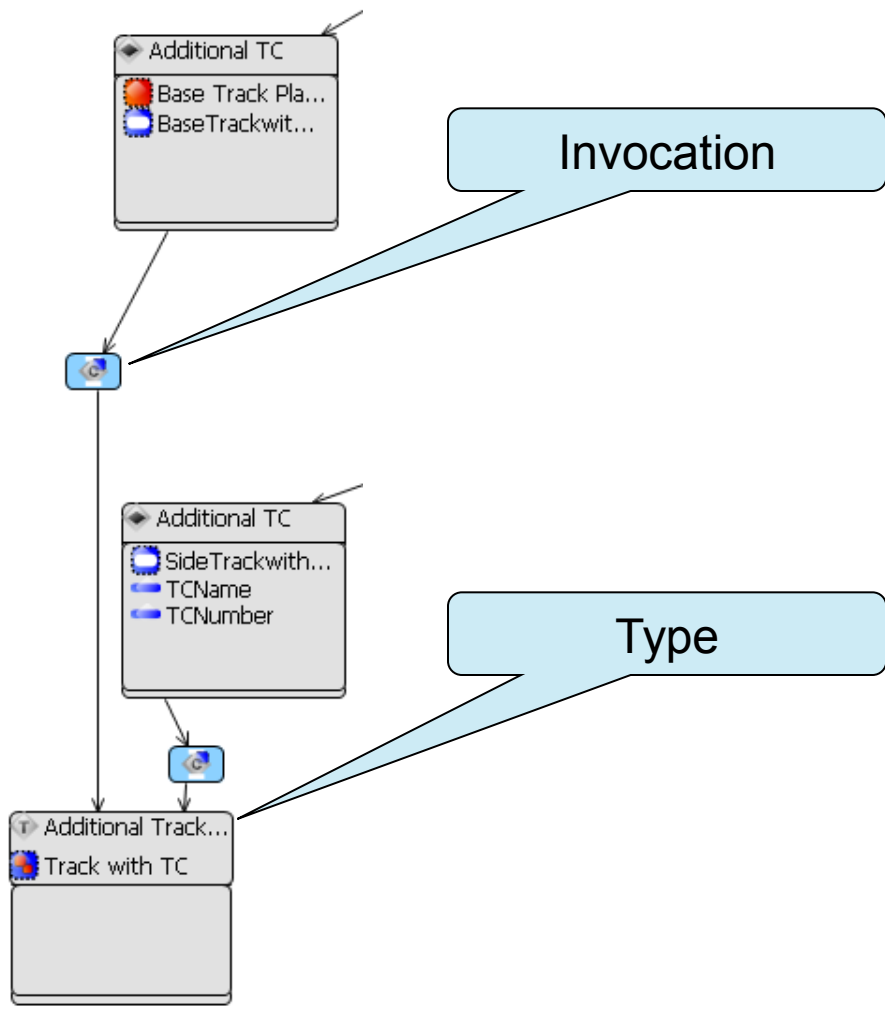
Resolved product model

Station Product Line

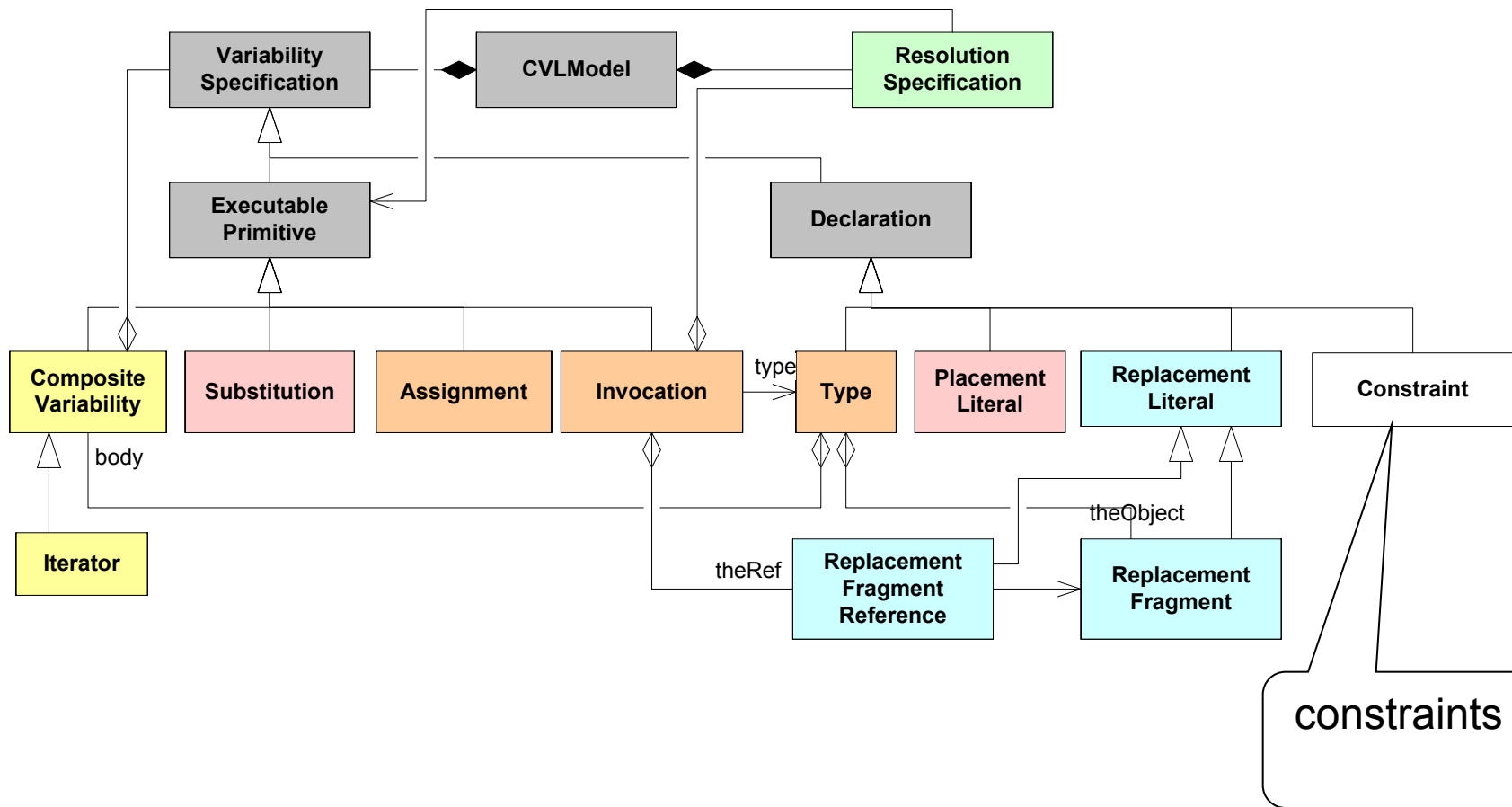
Eureka Σ! 3674 Programme, ITEA2 project ip06035
 SINTEF / Øystein Haugen, 2010-10-02



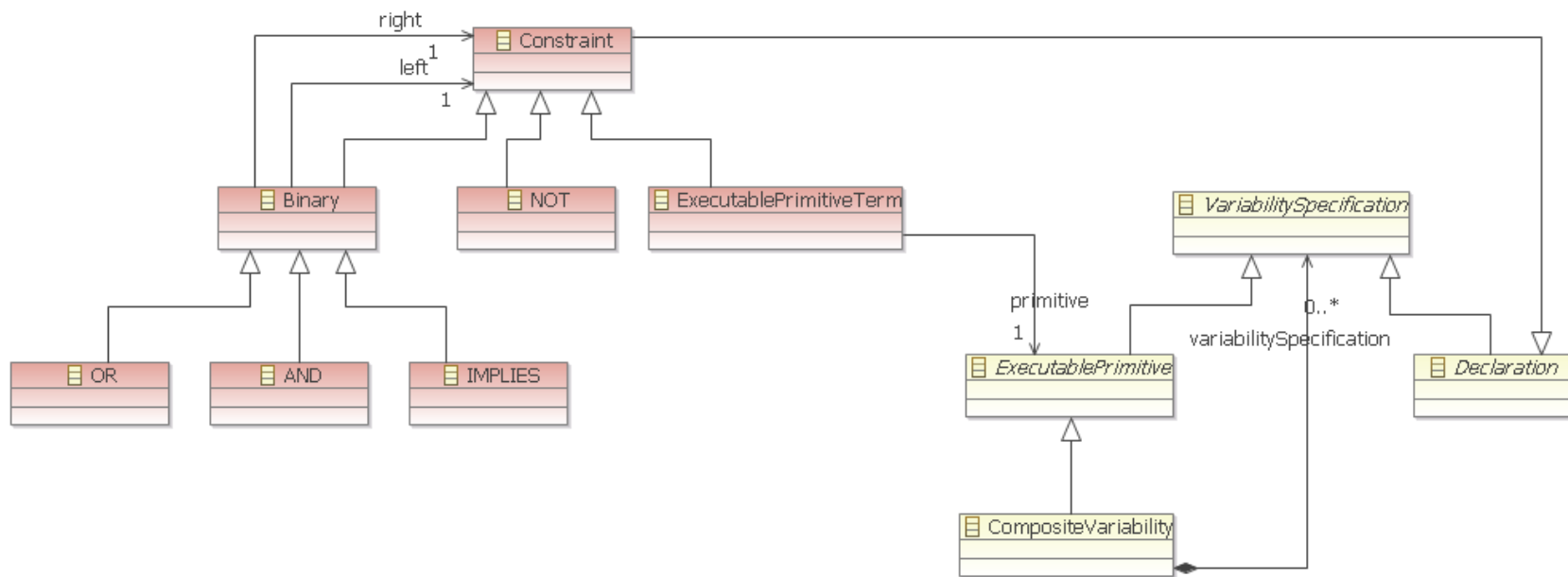
Type and Invocation



CVL Metamodel (important classes)



and the constraint



CVL Editor Integration

Eureka ΣI 3674 Programme, ITEA2 project ip06035
 SINTEF / Øystein Haugen, 2010-10-02

The screenshot shows the Eclipse IDE interface with three main windows:

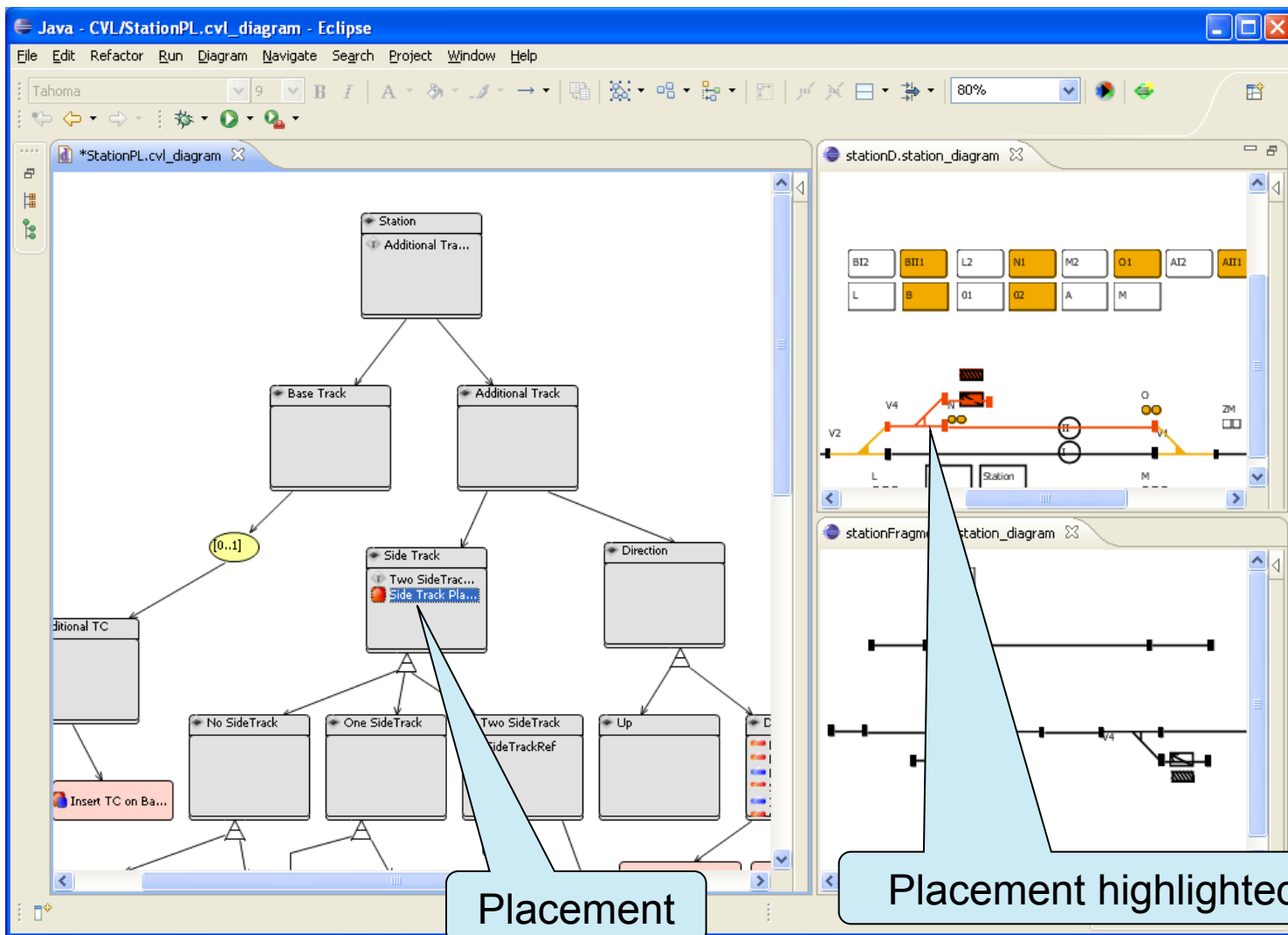
- CVL Editor (feature diagram):** A UML-like feature diagram for a 'Station' class. It shows a hierarchy where 'Station' has two children: 'Base Track' and 'Additional Track'. 'Additional Track' has two children: 'Side Track' and 'Direction'. 'Side Track' has three children: 'No SideTrack', 'One SideTrack', and 'Two SideTrack'. 'Direction' has two children: 'Up' and 'Down'. A multiplicity of '[0..1]' is shown on the arrow from 'Station' to 'Additional Track'. A red box highlights this entire diagram.
- TCL Editor:** A window showing a TCL script for 'stationD.station_diagram'. It contains a grid of track identifiers (BI2, BII1, L2, N1, M2, O1, AI2, AII1, L, B, 01, 02, A, M) and a corresponding track layout diagram with labels V2, V4, L, EL, Station, M, and ZM.
- TCL Editor (fragments):** A window showing a TCL script for 'stationFragments.station_diagram'. It contains a track layout diagram with labels 011, V3, and V4.

Specify Variability

Eureka Σ! 3674 Programme, ITEA2 project ip06035
 SINTEF / Øystein Haugen, 2010-10-02

The screenshot shows the Eclipse IDE interface with a Java project named 'CVL/StationPL.cvl_diagram'. The main editor displays a UML class diagram with classes like 'Station', 'Additional Track', 'Track', 'Side Track', 'Two SideT', 'Additional TC', 'No SideTrack', and 'One Side Track'. A context menu is open over the 'Additional TC' class, with the option 'Create Placement from selection' highlighted. A callout bubble points to the 'Selected elements' in the railway diagram on the right, which shows various track segments and signals. Another callout bubble points to the 'Create Placement from Selection' option in the context menu.

Highlight Variability



Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02



Feature Specification Layer
Editor

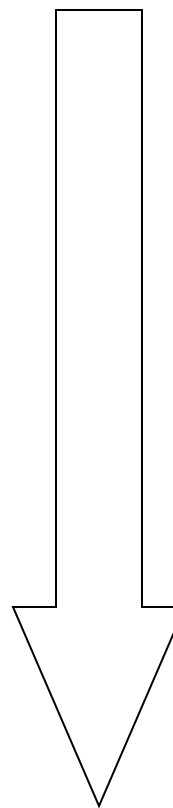
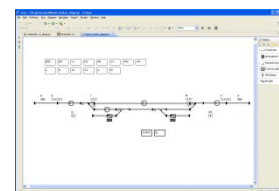
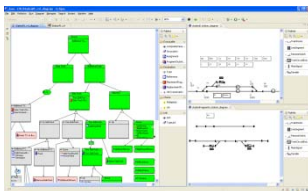
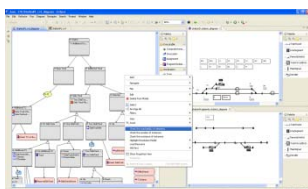
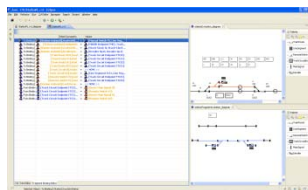
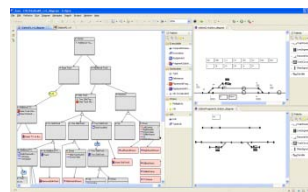
CVL Tools

Product Realization Layer
Editor

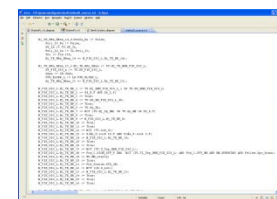
CVL Model Validator

Resolution Editor

Generated TCL Model



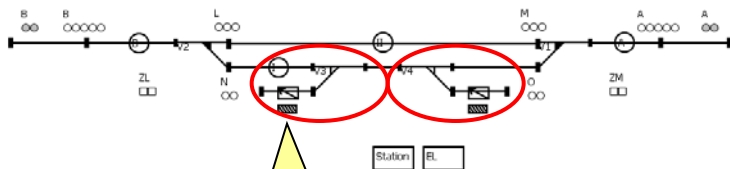
Generated Code



Evaluation of CVL – Case Study

StationB

BI2	BI1	L2	N1	M2	O1	AI2	AI1
L	B	O2	O1	A	M		



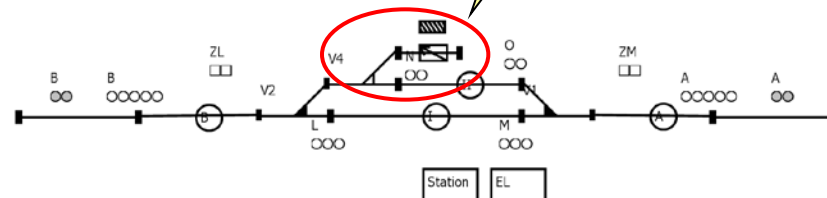
Two SideTracks

Direction: Down

One SideTrack

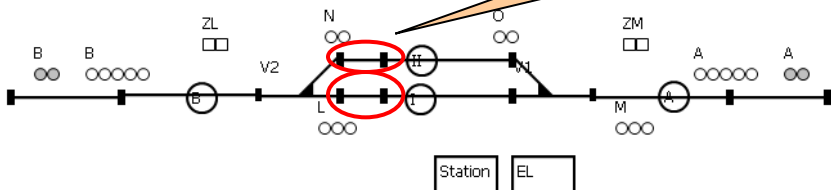
StationD

BI2	BII1	L2	N1	M2	O1	AI2	AI1
L	B	O1	O2	A	M		



StationS

BI2	BII1	L2	N1	M2	O1	AI2	AII1
L	B	O1	O2	A	M	O11	O21



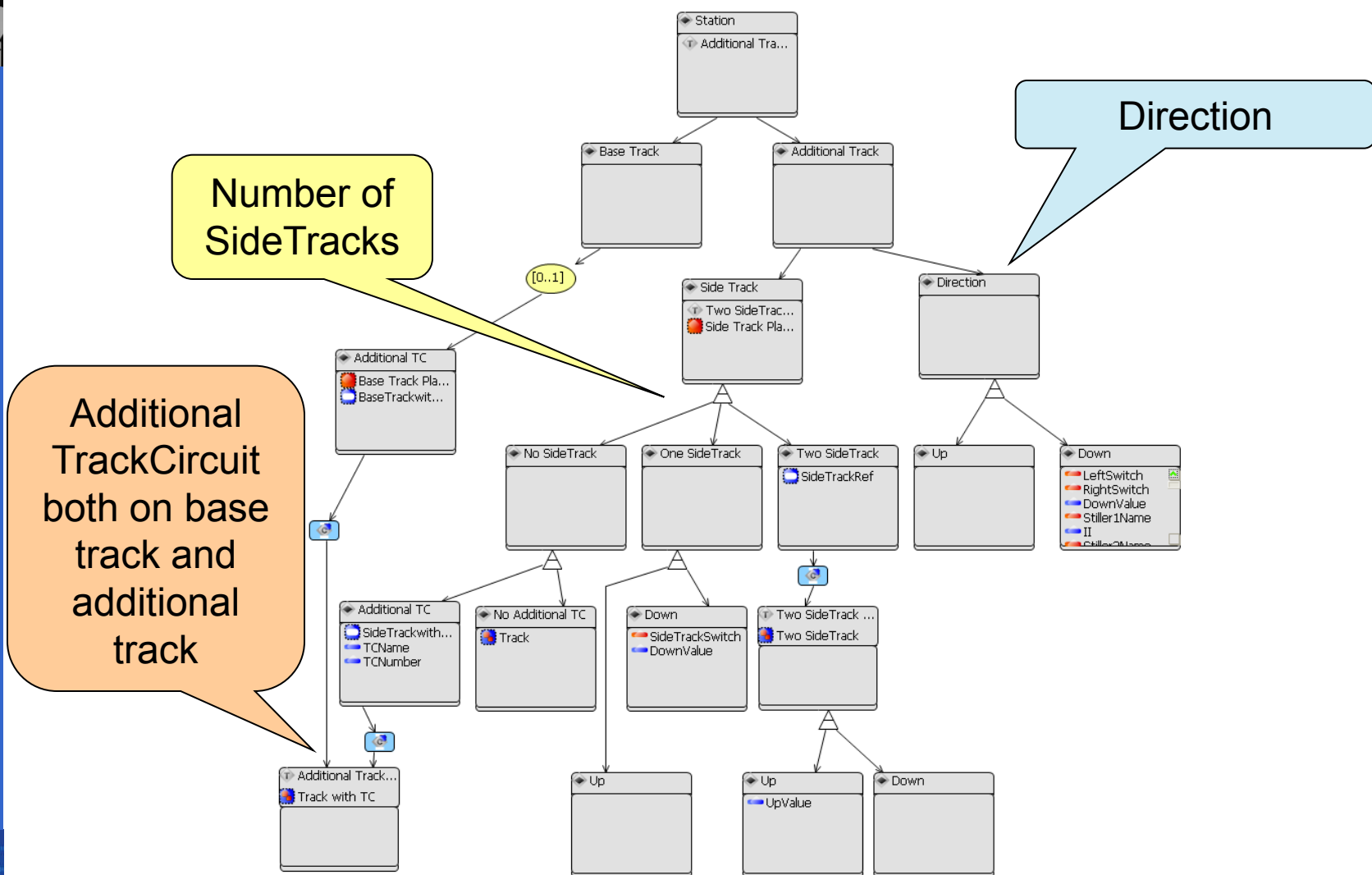
Two Additional TrackCircuits

No SideTracks

Station Product Line

Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02



Number of SideTracks

Direction

Additional TrackCircuit both on base track and additional track

Station Product Resolution



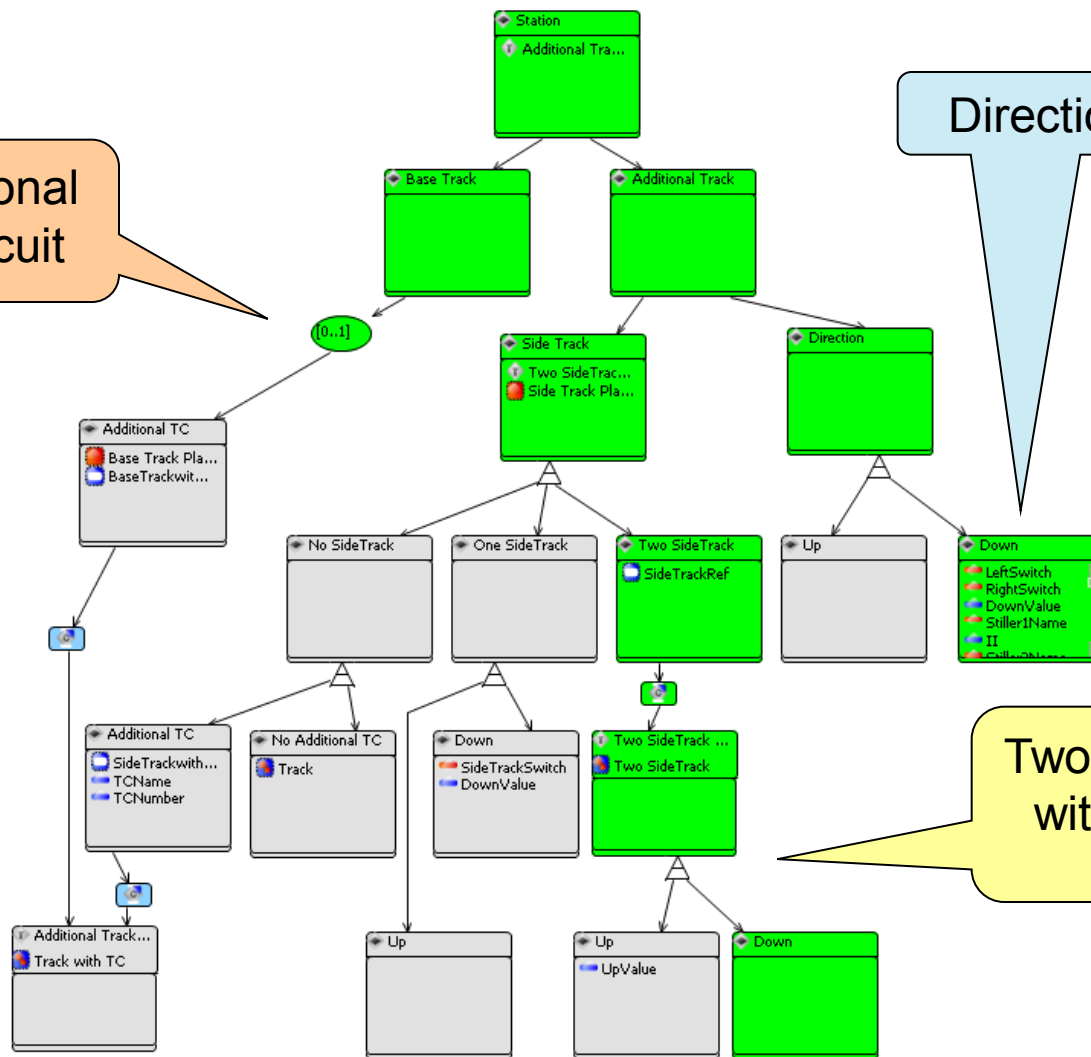
Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02

No Additional TrackCircuit

Direction: Down

Two SideTracks with direction Down





Eureka Σ! 3674 Programme, ITEA2 project ip066035

SINTEF / Øystein Haugen, 2010-10-02

Standardizing CVL in OMG

why?
who?
when?
how?

Standardization: Why OMG?

- **CVL is modeling and OMG is standardizing modeling**
 - MOF, UML, SysML, SoaML, UPDM, BPMN,
- **Eclipse has a set of tools well suited for implementing an OMG standard**
 - eCore, EMF, GMF, ...
- **Some of us have a long history in OMG**
 - and sometimes it helps to know someone
- **OMG is not the only choice**
 - ITU – The International Telecom Union (SDL/MSD)
 - Oasis – standard languages in the business domain
 - W3C – the web standards
 - ISO – a lot of other standards





Who participates?

- **Tool vendors**
 - Protecting their earlier and current investments
 - Opening up new markets
- **Industry partners of different domains**
 - Big companies want standards that comply with their needs
 - This will ensure proper tooling now and in the future
 - Small companies want standards such that they are better protected against the wims of the big ones
- **Research Institutes**
 - Fulfilling dissemination deliverables
- **Academics**
 - Bringing their brilliant thoughts to the people

The Challenges of Standardization

- **Nobody is interested**
 - Forget the whole idea (?)
- **Few tool vendors are interested**
 - And in OMG that means the technology will not become available
- **There is not enough agreement**
 - Within the consortium, or within the organization
- **A lot of effort is put into the technology, but the quality is not good enough**
 - There are not enough users of the technology
- **Your competitors are better**
 - Other standards become more popular



Contents of a good language standard

- **Subsumes de facto standards**
 - CVL will cover feature diagrams
- **Takes advantage of academic advances**
 - CVL can gain experience from
 - MoSiS CVL
 - Staged Feature Models
 - Aspect orientation
- **Incorporates general language stuff**
 - Types, instances
 - Functions, parameters
 - Inheritance
- **Is easy to implement in tools**
 - CVL: both reference prototypes and emerging functionality of a commercial product


CVL in OMG – the early process

- **Presentations for ADTF on variability modeling**
 - 2005: Product families – another area for OMG technology?
 - 2008: Should the OMG standardize variability modeling?
 - 2009: CVL - Common Variability Language
 - 2009: Common Variability Language DRAFT RFP
- **Current resources**
 - ad/2009-12-03: CVL RFP (iteration after AB meeting)
 - RFP = Request For Proposals
 - Wiki: <http://variabilitymodeling.org>
 - E-mail list: cvl@omg.org
 - Series of WebEx meetings

variabilitymodeling.org

Eureka Σ! 3674 Programme, ITEA2 project ip06035

SINTEF / Øystein Haugen, 2010-10-02



OBJECT MANAGEMENT GROUP

navigation

search

Go Search

toolbox

- What links here
- Upload file
- Special Pages
- Printable Version
- Permanent link
- Cite this Article

[article](#) | [discussion](#) | [show pagesource](#) | [old revisions](#)

Trace: » [start](#)

Welcome to the Common Variability Language Wiki

-Table of Contents

- Welcome to the Common Variability Language Wiki
- News items
- Contact

This is a wiki for variability modeling intended to support the work to standardize a language for variability modeling in the [OMG](#).

Variability Modeling is to efficiently describe more than one variant of a system. Variability modeling is often closely associated with product lines. The resulting systems are often fairly complex and variations are described explicitly. Variability can be expressed in stand-alone models, such as feature diagrams. Software Product Lines refer to methods, tools and techniques for creating and maintaining a collection of similar software systems from a shared set of software assets.

The intention of variability modeling is to create and manage many variants of a product, also known as mass customization. Variability modeling is regarded as the enabling technology for delivering a wide variety of software systems of high quality in a fast, consistent and comprehensive way. The key is to build a base on the commonalities and efficiently express and manage the variability of the systems.

Variability can also be described in annotations or extensions to the base model, or totally separated out to represent in an independent variability model.

- Click for further [Variability Modeling Resources](#)

News items

- [OMG](#) issued the CVL RFP on 12. December 2009. The issued RFP cleaned up can be found [here](#). The official page is following this link [here](#)
- Presentation of the CVL RFP by Øystein Haugen can be found [here](#)

Contact

SINTEF [\(Øystein Haugen\)](#) and the ITEA2 [project MoSiS](#) are responsible for this initiative and if you are interested in joining the efforts (you do not need to be an [OMG](#) member), please register for this wiki through the login in upper right corner of the page, and then you may contact us by filling out [this form](#).

start.txt · Last modified: 2010/03/10 09:53 by ohaugen

CVL in OMG – the formal process

Event or Activity	Actual Date
<i>Preparation of RFP by TF</i>	<i>September 2009</i>
<i>RFP placed on OMG document server</i>	<i>September 2009</i>
<i>Approval of RFP by Architecture Board Review by TC</i>	<i>December 2009</i>
<i>TC votes to issue RFP</i>	<i>December 2009</i>
<i>LOI to submit to RFP due</i>	<i>May 2010</i>
<i>Initial Submissions due and placed on OMG document server ("Three week rule")</i>	<i>August 2010 (changed to November 2010)</i>
<i>Voter registration closes</i>	
<i>Initial Submission presentations</i>	<i>September 2010 (changed to December 2010)</i>
<i>Preliminary evaluation by TF</i>	<i>October 2010</i>
<i>Revised Submissions due and placed on OMG document server ("Three week rule")</i>	<i>February 2011</i>
<i>Revised Submission presentations</i>	<i>March 2011</i>
<i>Final evaluation and selection by TF Recommendation to AB and TC</i>	<i>June 2011</i>
<i>Approval by Architecture Board Review by TC</i>	<i>June 2011</i>
<i>TC votes to recommend specification</i>	<i>June 2011</i>
<i>BoD votes to adopt specification</i>	<i>June 2011</i>

RFP Objective

- The objective of this RFP is to enable the specification of the **variability in product line models** in order to support seamless product line modeling across the whole product line engineering process.
- This CVL RFP requests a specification language including a **metamodel, semantics** and **concrete syntax** for variability specification.
- Variability specifications shall relate to a base product line model that describes the whole product line and shall comprise:
 - a variability model with the following elements: a model of possible choices and relationships between those choices and the base model
 - resolution models which resolve variability (by a set of choices) and thus define specific product models.
- CVL shall support base models in languages that are defined by means of **MOF-compliant metamodels**, including UML and Domain Specific Languages.

CVL RFP 6.5.1 Coverage

- **6.5.1.1 Proposals shall define a language that can express variabilities on models in **any** language that is defined by means of a **MOF-compliant metamodel**.**
- **The language shall support:**
 - The most common **variability mechanisms**, like optionality, alternatives, etc. that have been acknowledged as mature within the fields of product line modeling and feature modeling.
 - **Constraints** on the variabilities;
 - **Abstraction mechanisms** that support the definition and application of compound variability specifications.
 - **Resolutions** of the variabilities, defining the set of actual choices.

CVL RFP 6.5.1 Coverage (cont.)

- 6.5.1.2 The proposed language shall specify variability as a model **separate** from the base model on which the variabilities apply.
- 6.5.1.3 The proposed language shall have mechanisms for relating variability specifications to those base model elements that are subject to variation. These **relationship mechanisms** may assume that base models are made in languages that are defined by MOF-compliant metamodels.
- 6.5.1.4 The proposed language shall be defined by means of a **MOF-compliant metamodel**.
- 6.5.1.5 Proposals shall provide a non-normative demonstration of a CVL description **applied to a base model in UML including profiles**.

CVL RFP 6.5.2 Semantics

- **6.5.2.1** The proposal shall **define the semantics** of the variability language **e.g. by using QVT** or other transformation languages. The execution of a variability model with specific resolutions should result in either alterations (at runtime) of an executing product (system), or materialize (by filtering or by generation) as a specific product model in the base language.

CVL RFP 6.5.3 Notation

- 6.5.3.1 Proposals shall specify the complete **concrete syntax** for CVL,
- 6.5.3.2 Proposals shall demonstrate how the notation of **Feature Diagrams** can be integrated within the concrete syntax of the proposed language.
- 6.5.3.3 Proposals shall define the **notation for relationships** either in separate descriptions or as annotations to the base model notation.



Initial Submission: Letter Of Intent

LOI List:

Company	Date	Notes
Fraunhofer FOKUS	May 19, 2010	
Fundacion European Software Institute	May 18, 2010	
International Business Machines	May 24, 2010	
TCS	April 26, 2010	
THALES	May 21, 2010	

- **TCS is in one consortium, and the rest is in another – for now**
- **adding also technical supporters**
 - SINTEF, INRIA, University of Waterloo and IT-Univ. of Copenhagen
- **The main challenge for a large consortium is to merge our ideas and attempts – but we need not do this perfectly in the first round**

Our consortium – who are we?

- **Tool vendor**
 - IBM – represented by the Rhapsody tool people and by the research department
- **Industry partners of different domains**
 - Thales – a big company of users
- **Research Institutes**
 - SINTEF – initiators through MoSiS
 - ESI – in MoSiS and making the tool PLUM
 - INRIA – long term players in the product line area
 - Fraunhofer FOKUS – likewise
- **Academics**
 - U. of Waterloo (K. Czarnecki)
 - IT Univ of Copenhagen (A. Wasowski)



CVL – the future?

- **In December 2010 initial submissions are presented**
 - Next phase can start
- **Can the different submissions merge?**
- **Are there comments that should be taken into account?**
- **Should more supporters be added?**
- **One (or more) revised submissions with complete CVL will be created**
- **CVL will be adopted**
 - and tools made to support it

