

Informatics Seminar

Meta-modelling and SMILE

Prof. Andreas Prinz



Contents

- Problems of SWE
- Model-driven technologies
- The SMILE project
- EIAO WAMs
- Summary



Software Development Crisis

- Standish Group 1999 Study of Large Government and Commercial Systems
 - Only 16% completed on time & within budget
 - 31% cancelled
 - 53% cost growth exceeded 89%
 - Of all completed, on average, only 61% of specified features were included
 - Successful projects are usually small
- Software development techniques are not able to produce good software



Modern Software Challenges

- distributed systems
 - Worldwide distributed systems, Wireless and Handheld Devices
 - New Technologies: XML, Web Services
- heterogeneous systems
 - no consensus on hardware platforms, operating systems, network protocols, programming languages
- long-living systems (including old components)
 - Integration of legacy applications and databases
 - Go fast onto new developments
 - Retaining qualified staff; maintaining existing code base, protect investments
- highly available systems
 - E-business applications, Web applications that will respond in seconds
 - accommodate everything, including predictable changes
 - safety, security
- service provision by cooperation
 - inter-connectivity within and between companies
 - enterprise application integration



Computer Science: Engineering

- Engineering needs reliable methods to come to good results (best practice).
- Computer Science has lots of methods, but none is really engineering.
- Engineering:
 - build models of the product in advance
 - analyse and improve the models
 - build the real systems according to the models



The Good News

- Increased standardization
 - Internet Protocols, SQL, UML
- Increased openness
 - Apache, Linux
- Less custom development
 - Component Reuse, ERP (Enterprise Resource Planning) Applications
- Platform independent intermediate layers (relative term)
- Modern Software Development
 - CASE, CMM
 - XP, Agile Programming



Contents

- Problems of SWE
- Model-driven technologies
- The SMILE project
- EIAO WAMs
- Summary



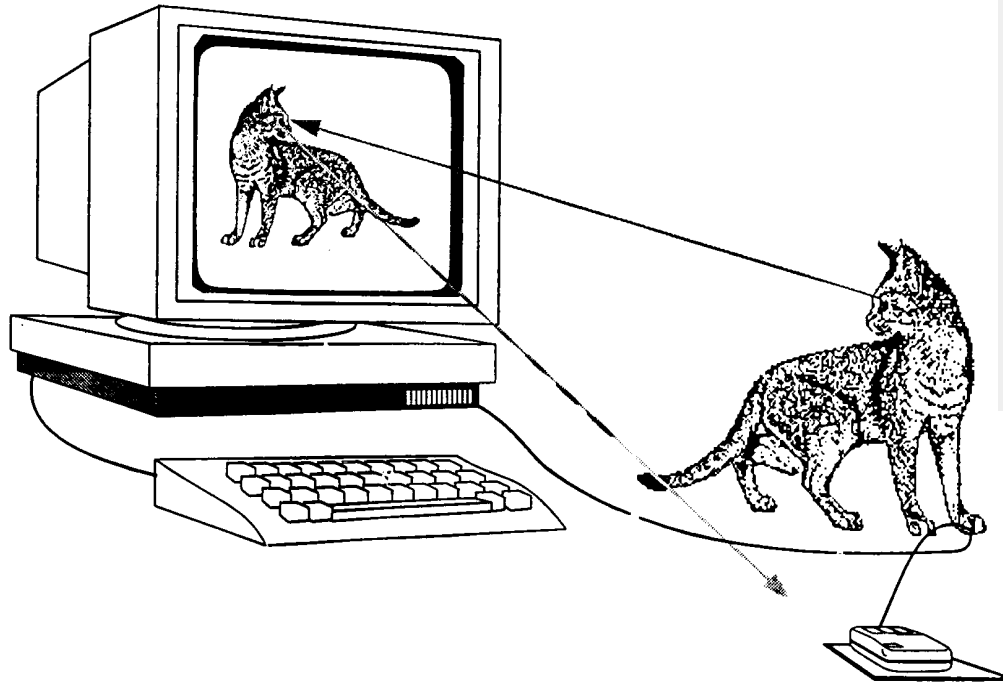
What is a Model?

- A model is a description of a (part of) system written in a well-defined language.
A well-defined language is a language with well-defined form (syntax), and meaning (semantics), which is suitable for automated interpretation by a computer.
- Models are around all the time
 - Models of the real Bits: Assembler
 - Models of the Control Flow: Prog. Lang.
 - Models of data storage: Database descriptions
 - Models of access: Interface languages
 - Important: Abstraction level is different!



Norbert Wiener (1894-1964)

What is the best model of a cat ?



It is a cat.

However it
has to be the
same cat.

Modeling Maturity Levels

- MML 0: No specification
 - Specification of software is kept in the heads of the developers.
- MML 1: Textual Specification
 - Specification of software in natural language documents.
- MML 2: Text with Models
 - Specification of software in natural language documents plus high-level architecture diagrams.
- MML 3: Models with Text
 - Specification of software in models. Natural language text for background and motivation.
- MML 4: Precise Models
 - Specification of software in models (+ comments).
 - Code generation from the models (MDA!).
- MML 5: Models only
 - The models are precise and detailed: complete code-generation.
 - The code is invisible (as assembler is today).
 - Modeling language → High level programming language.



Model-driven Development

- Put the models in the centre of the activities
- develop, analyse and improve the models
- generate all other things from the models: in a way interpret/execute the model
- Models must be precise - Precision and detail are *not* the same!
- MDA = Model Driven Architecture
- see also <http://www.omg.org/mda/>

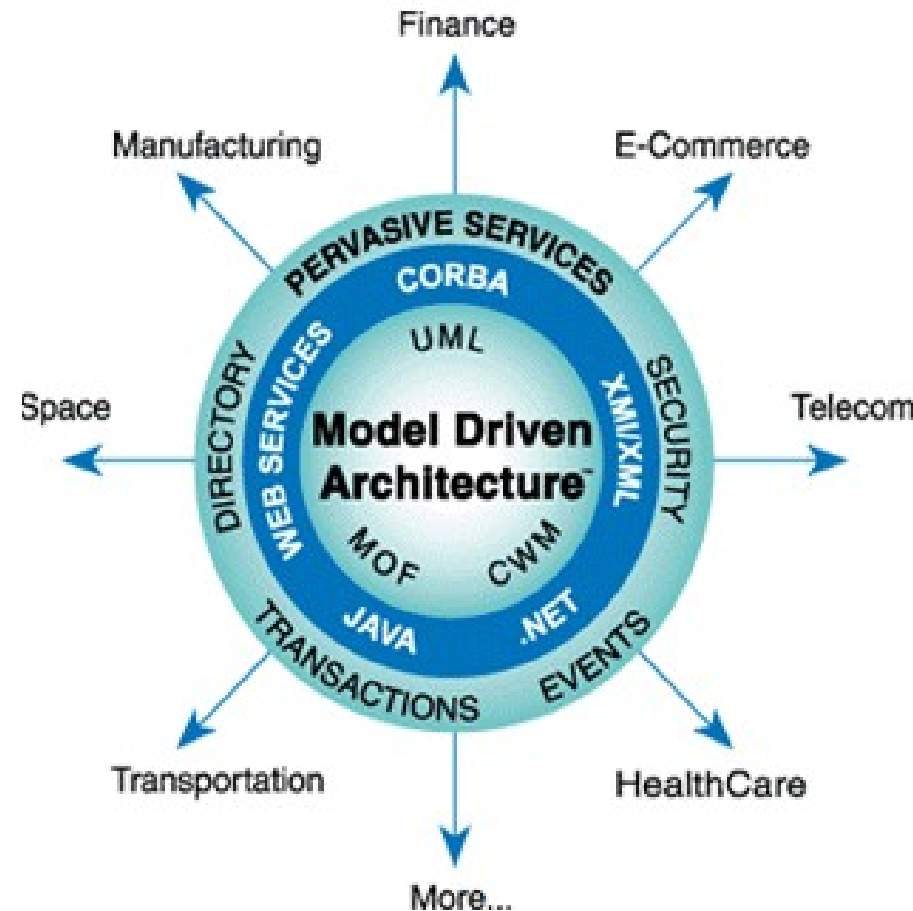


<http://www.omg.org/mda/>

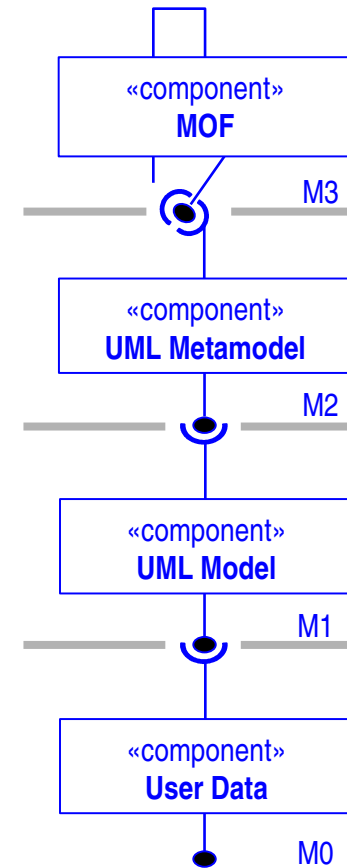
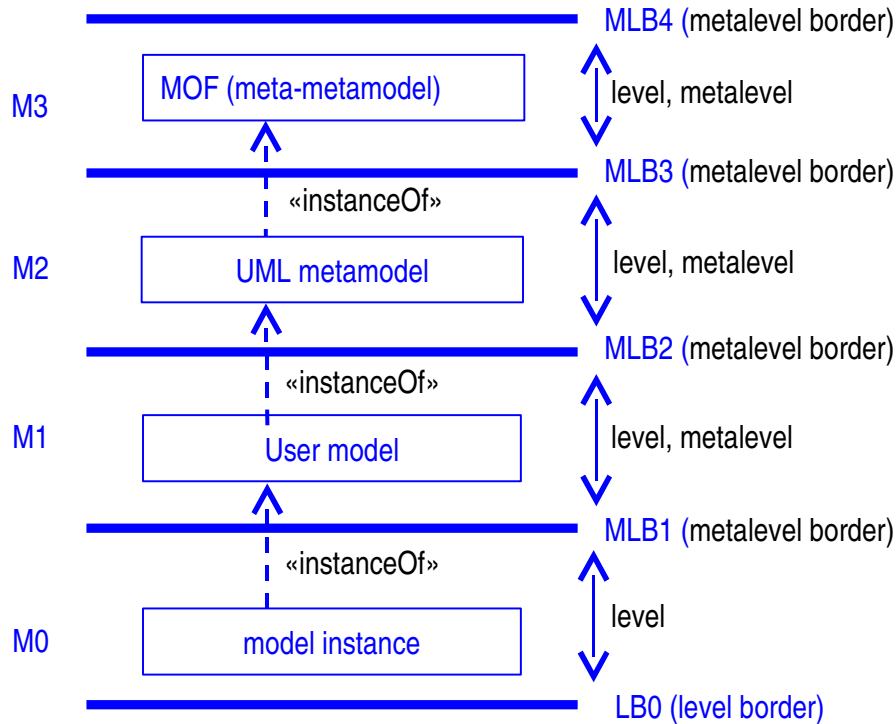
OMG Model Driven Architecture™

How Systems Will Be Built

MDA™ provides an open, vendor-neutral approach to the challenge of interoperability, building upon and leveraging the value of OMG's established modeling standards: Unified Modeling Language™ (UML™); Meta-Object Facility™ (MOF™); and Common Warehouse Meta-model™ (CWM™). Platform-independent Application descriptions built using these modeling standards can be realized using any major open or proprietary platform, including CORBA®, Java, .NET, XMI™/XML, and Web Base platforms.



OMG metamodel levels



Three Abstraction Layers

- explicit separation of three abstraction layers.
- CIM:
 - The most abstract model, the **Computational Independent Model (CIM)**, deals only with concepts of the application domain.
- PIM:
 - A CIM is refined into a **Platform Independent Model (PIM)** that contains already computational information about a system but is free from platform specific realization details.
 - PIM usually written in UML.
- PSM:
 - These details are part of a **Platform Specific Model (PSM)**, which is a refinement of the PIM.
 - PSM can be diverse: Java, J2EE, SQL, C++, .NET, COBOL, C#, CORBA, XML, etc. etc.

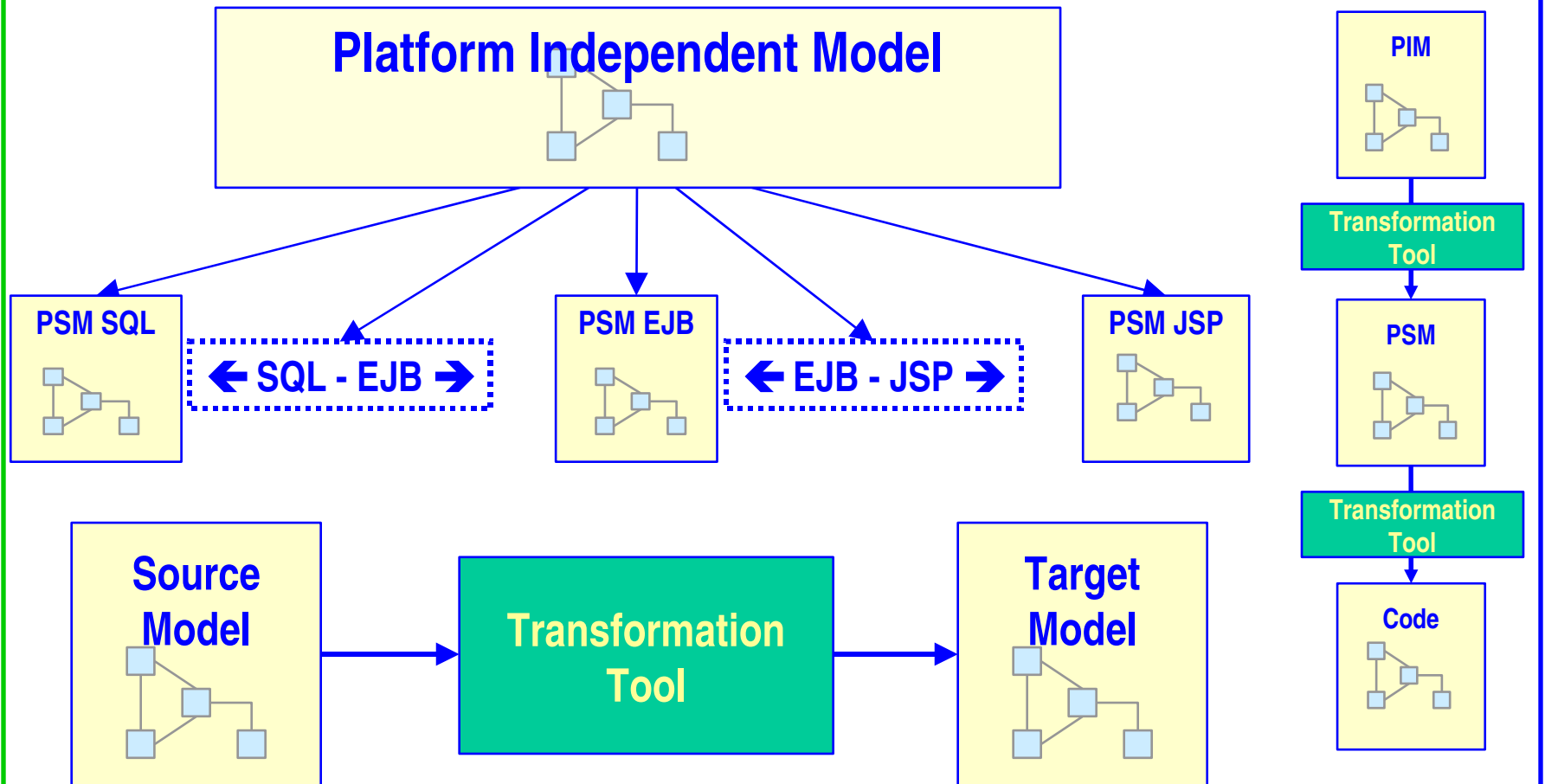


Transformation

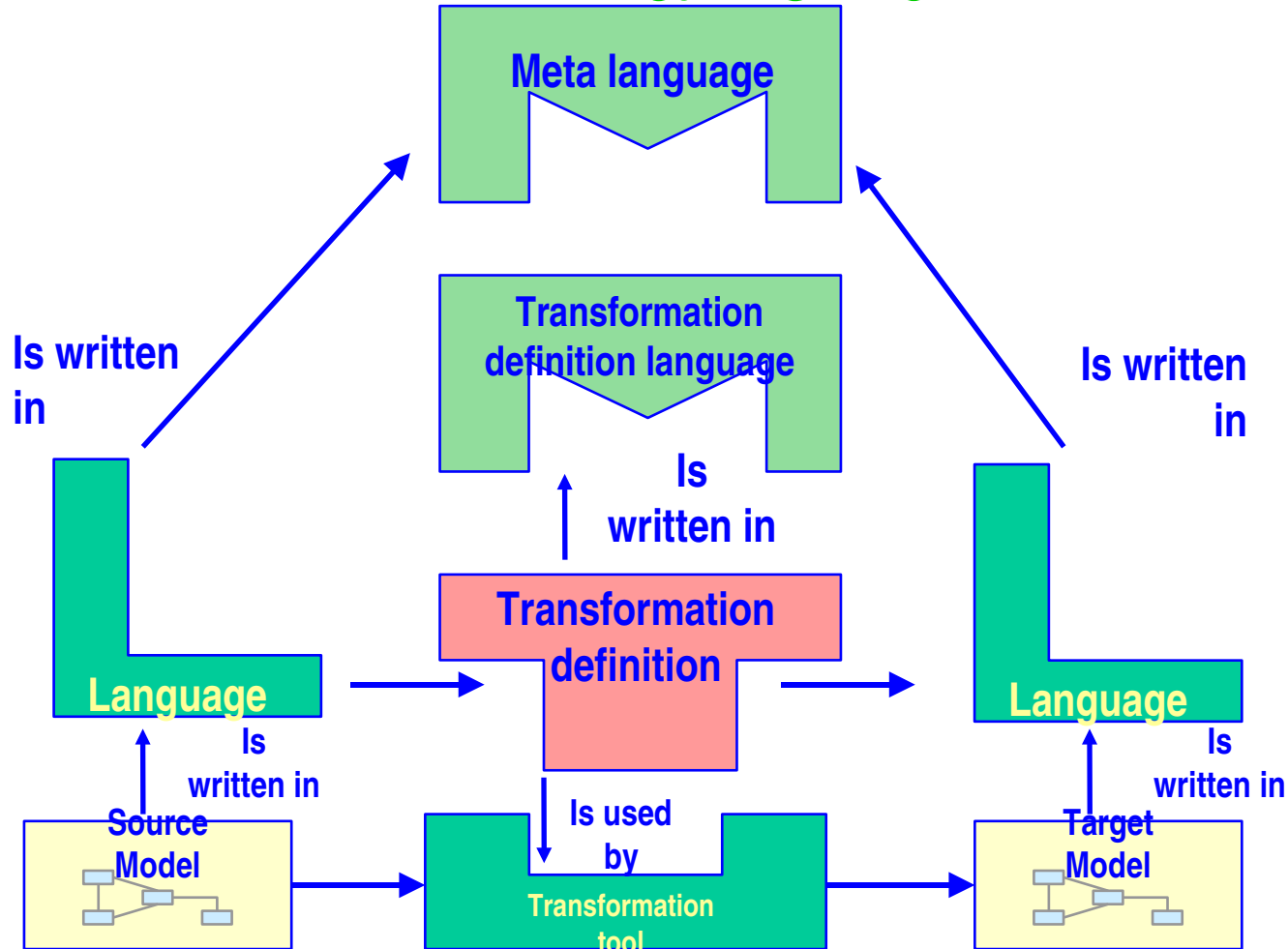
- A transformation is the automatic generation of a target model from a source model, according to a transformation definition.
- A transformation definition is a set of transformation rules that together describe how a model in the source language can be transformed into a model in the target language.
- A transformation rule is a description of how one or more constructs in the source language can be transformed into one or more constructs in the target language.



MDA Basic Framework



MDA Framework



Contents

- Problems of SWE
- Model-driven technologies
- The SMILE project
- EIAO WAMs
- Summary



Which language to take?

- The one that fits best, but
- We need tools for the language!
- Better take a predefined one?
- Define your own language and use MDA for the language description
- This is called Language Driven Development = LDD

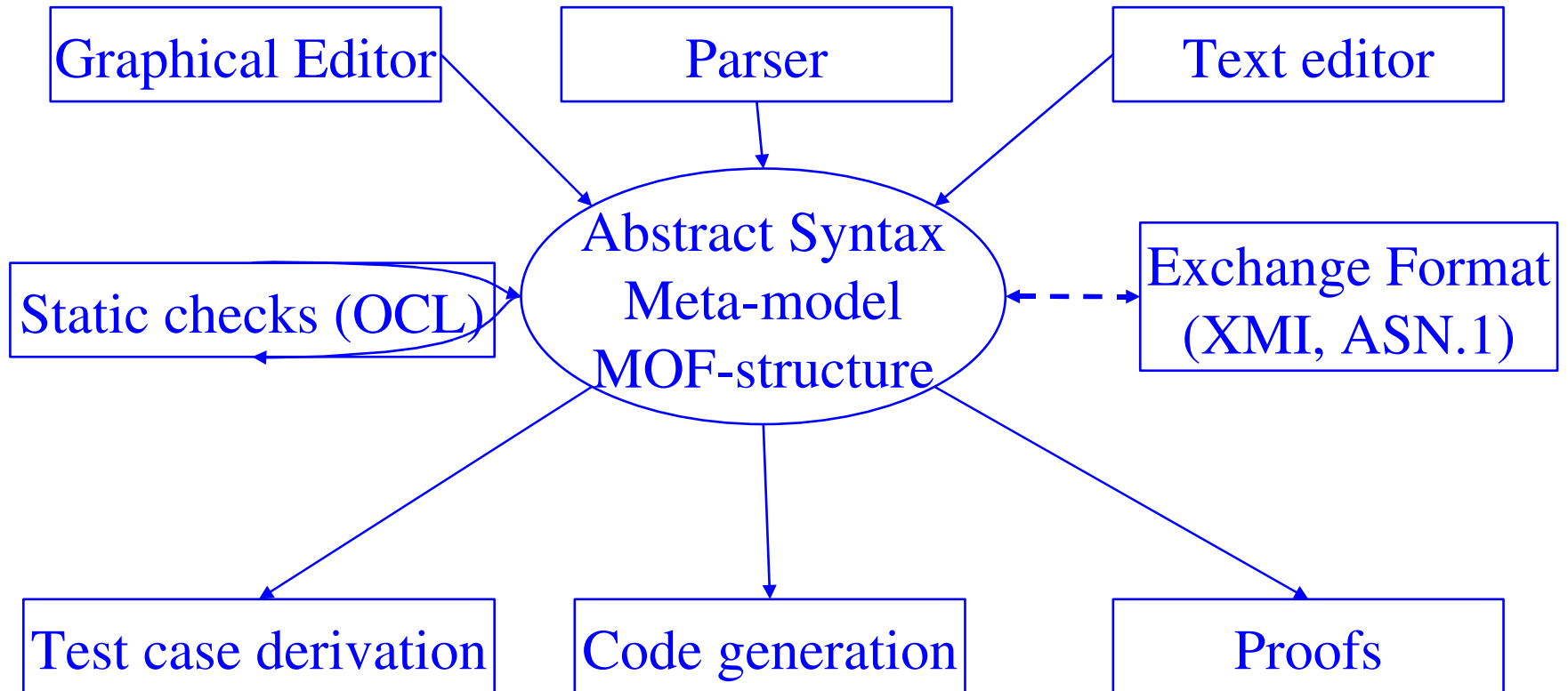


SMILE Project Overview

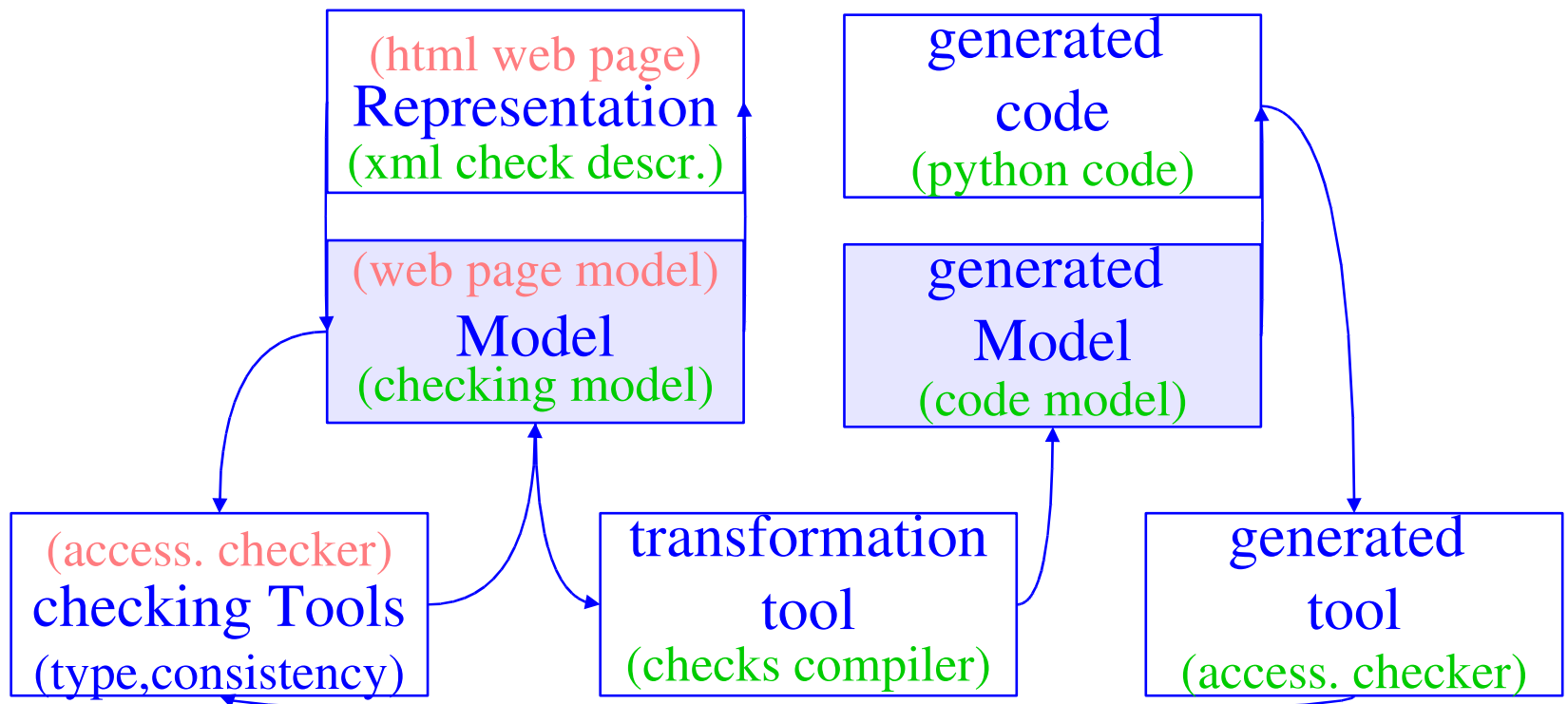
- SMILE = Semantic Meta-model-based Integrated Language Environment
 - Semantic: explicit representation of all semantic (meaning-giving) information
 - Meta-model-based: Model and Meta-model handling is in the centre
 - integrated: We will handle all information in the same way independent of the abstraction, meta-modelling and executability level
 - language: the main focus is on language handling and integration and providing tools for languages
 - environment: the aim is a complete environment allowing all important operations



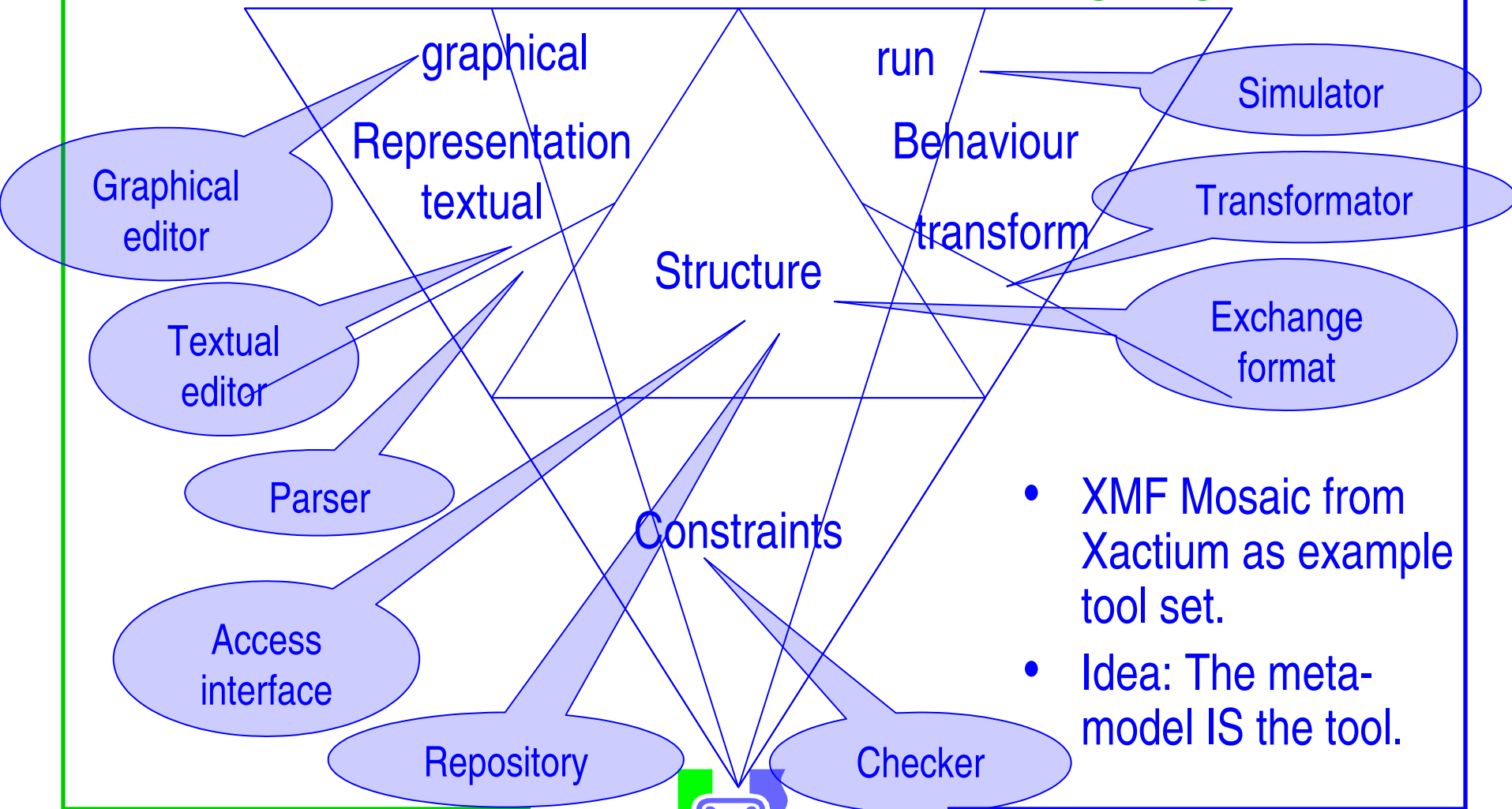
SMILE Approach



SMILE implements MDA



Aspects of a meta-model / language



- XMF Mosaic from Xactium as example tool set.
- Idea: The meta-model IS the tool.



SMILE current state

- Some theory about meta-modelling base framework
- Basic implementation of base framework in Java & Eclipse using EMF
- basic integration with EMF and Eclipse
- work in progress
 - integration with Coral framework
 - graphical user interface
 - inclusion of semantic aspects (constraints etc.)



Contents

- Problems of SWE
- Model-driven technologies
- The SMILE project
- EIAO WAMs
- Summary

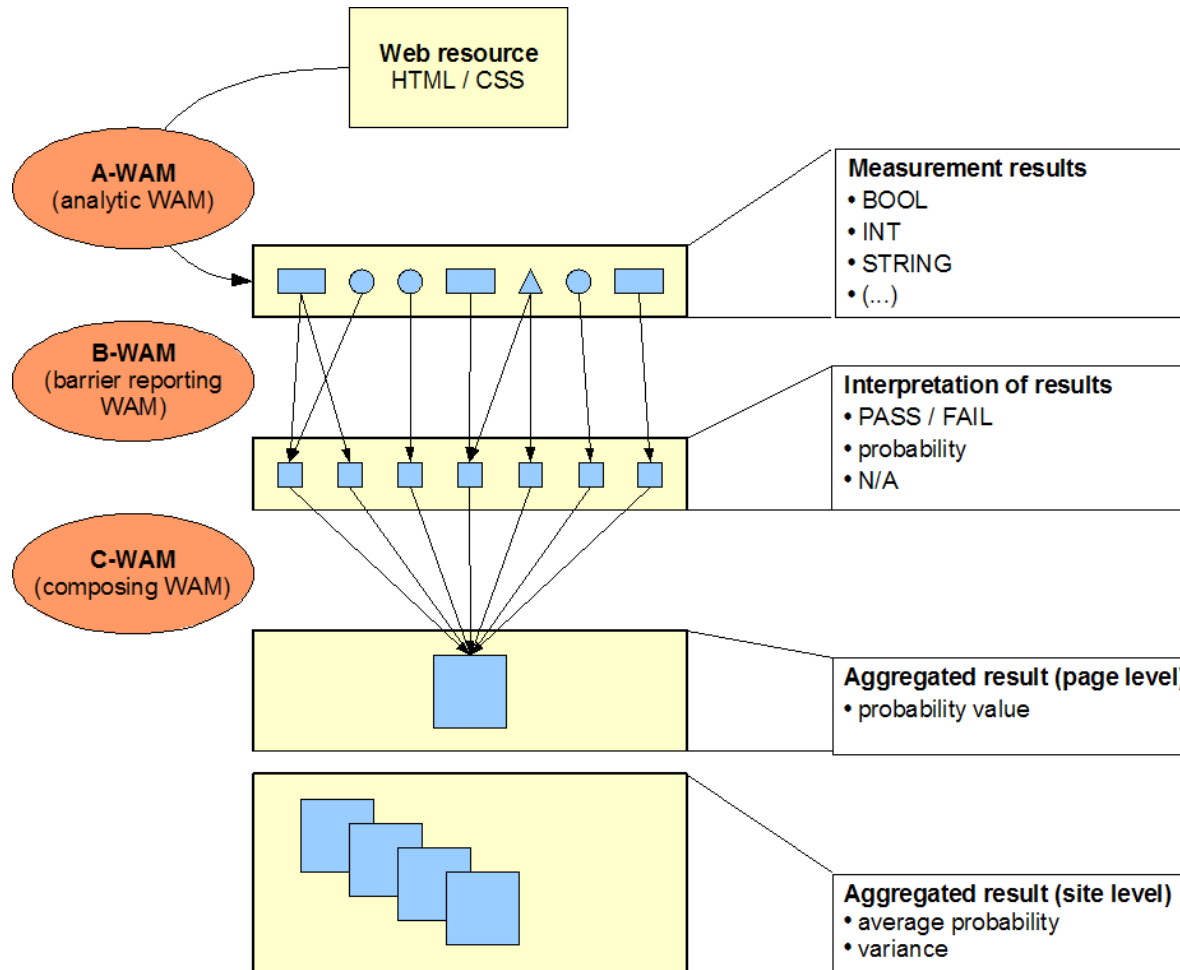


EIAO WAM Concept

- A Web Accessibility Metric (WAM) is a formal rule specifying how to make a statement about accessibility barriers in a web resource.
- The goal of EIAO is to make these statements automatically.
- Therefore the rules are based on the measurable properties of the web resource.
- The required decisions are described in an unambiguous way to ensure that the results are repeatable.
- Furthermore, the meaning of the end result (i.e. the accessibility statement) has to be clear.
- This is achieved by providing a consistent interpretation of the intermediary results.



EIAO WAM architecture



A10.01.01.005.001-HTML01.001

- Does the inspected `<area>` element have an alt attribute?

- **Schematron description:**

```
<sch:rule context="html:area">
```

```
<sch:report test="@alt">1</sch:report>
```

```
</sch:rule>
```

- **OCML description:**

```
context=area
```

```
inv self.exists(alt)
```



A10.03.03.001.001-HTML03.001

- Is the deprecated attribute bgcolor used for the inspected element?

- **Schematron description:**

```
<sch:rule context="html:body | html:table | html:tr | html:th | html:td">  
<sch:report test="@bgcolor">1</sch:report>  
</sch:rule>
```

- **OCL description:**

context=body inv self.exists(bgcolor)

context=table inv self.exists(bgcolor)

context=tr inv self.exists(bgcolor)

context=th inv self.exists(bgcolor)

context=td inv self.exists(bgcolor)



A10.01.01.002.001-HTML02.001

- Is the inspected <object> element the only part of a link?

- **Schematron description:**

```
<sch:rule context="html:object[@type]
[starts-with(@type,'image')]
[ancestor::*[name()='a']
[string-length(normalize-space(text()))=0]]">
<sch:report test="normalize-space(.)">0</sch:report>
</sch:rule>
```

- **OCL description:**

context=object

inv startsWith(self.type,'image') and

self.ancestors->exists(ala.name = 'a') implies normalStringLength(self.text)=0



A10.04.02.001.001-HTML01.001

- Is the inspected <abbr> element the first occurrence of the abbreviation within the web resource (html page), but has no title attribute or an empty (trimmed) title attribute?

- **Schematron description:**

```
<sch:rule context="html:abbr">
```

```
<sch:report test=" not(preceding::html:abbr[string()=string(current())])
```

```
and not(@title and normalize-space(@title))">0</sch:report>
```

```
</sch:rule>
```

- **OCML description:**

```
context=abbr
```

```
inv self.preceding->not exists(sls.string = self.string) implies (self.exists(title) and  
normalStringLength(self.title)>0
```



A10.12.04.001.001-HTML04.001

- Is the corresponding id definition for a for missing?

- **Schematron description:**

```
<sch:rule context="html:label[@for]">
```

```
<sch:report test=" (preceding::html:select/@id = @for)
```

```
or (following::html:select/@id = @for)
```

```
or (preceding::html:textarea/@id = @for)
```

```
or (following::html:textarea/@id = @for)
```

```
or (preceding::html:input[@type!='hidden'][@type!='submit'] [ @type!='button']/@id  
= @for)
```

```
or (following::html:input[@type!='hidden'][@type!='submit']
```

```
[@type!='button']/@id = @for)">0</sch:report>
```

```
</sch:rule>
```

- **OCL description:**

context=label

inv self.exists(for) implies allInstances->forall(i | i.id <> self.for.id)



A10.05.02.001.001-HTML05.001

- Is <th> used without id attribute and without scope attribute (in <table> not containing other <table> with two header level complexity)?

- **Schematron description:**

```
<sch:rule context="html:th[ancestor::html:table[1]
[not(descendant::html:table)]
[ (count(descendant::html:tr[ child::html:th and (not(child::html:td)
or not(normalize-space(child::html:td))))]=2)
or ( count(descendant::html:tr[ child::html:th
and (not(child::html:td)
or not(normalize-space(child::html:td))))]<lt;2
and count(descendant::html:tr[count(child::html:th)=2])
=( count(descendant::html:tr))
-count(descendant::html:tr[ child::html:th
and ( not(child::html:td)
or not(normalize-space(child::html:td))))]]]">
<sch:report test="@id or @scope">0</sch:report>
</sch:rule>
```

- **OCL description not given**



Some more examples

- Does the inspected <area> element have an alt attribute?

- **Schematron description:**

```
<sch:rule context="html:area">
```

```
<sch:report test="@alt">1</sch:report>
```

```
</sch:rule>
```

- **OCML description:**

```
context=area
```

```
inv self.exists(alt)
```

- Is the alt attribute of the inspected <area> element not empty?
- Is the body of the inspected <object> element empty?
- Are absolute values used for the cols or rows attribute in the inspected <frameset> element?
- Is an absolute value used in for the size attribute of the inspected or <basefont> element?
- Is there a <h*> element in the inspected web resource (html page)?
- Does the inspected page, which uses frames, contain a <noframes> element?



A10.01.01.005.001-HTML02.001

- Is the alt attribute of the inspected <area> element not empty?

- **Schematron description:**

```
<sch:rule context="html:area[@alt]">
```

```
<sch:report test="(string-length(@alt) > 0)">1</sch:report>
```

```
</sch:rule>
```

- **OCML description:**

```
context=area
```

```
inv self.exists(alt) implies Length(self.alt) > 0
```



A10.01.01.002.001-HTML01.001

- Is the body of the inspected <object> element empty?

- **Schematron description:**

```
<sch:rule context="html:object[@type]
```

```
[starts-with(@type,'image')] ">
```

```
<sch:report test="normalize-space(.)">0</sch:report>
```

```
</sch:rule>
```

- **OCML description:**

```
context=object[@type]
```

```
inv startsWith(self.type,'image') implies notEmpty(self.body)
```



A10.03.04.001.001-HTML01.001

- Are absolute values used for the cols or rows attribute in the inspected <frameset> element?

- **Schematron description:**

```
<sch:rule context="html:frameset">
```

```
<sch:report test=" @rows and (not(contains(@rows, '%')) and  
not(contains(@rows, '*')))
```

```
or @cols and (not(contains(@cols, '%')) and  
not(contains(@cols, '*')))">1</sch:report>
```

```
</sch:rule>
```

- **OCCL description:**

```
context=frameset
```

```
inv (self.rows->union(self.cols))->  
forall(rlr <> '%' and r <> '*')
```



A10.03.04.001.001-HTML02.001

- Is an absolute value used in for the size attribute of the inspected or <basefont> element?

- **Schematron description:**

```
<sch:rule context="html:basefont[@size]">
```

```
<sch:report test="not(contains(@size, '+')) and not(contains(@size, '-'))">1</sch:report>
```

```
</sch:rule>
```

```
<sch:rule context="html:font[@size]">
```

```
<sch:report test="not(contains(@size, '+')) and not(contains(@size, '-'))">1</sch:report>
```

```
</sch:rule>
```

- **OCL description:**

```
context=basefont
```

```
inv self.exists(size) implies self.size->forall(ele<>'+' and e<>'-' )
```

```
context=font
```

```
inv self.exists(size) implies self.size->forall(ele<>'+' and e<>'-' )
```



A10.03.05.001.001-HTML01.001

- Is there a <h*> element in the inspected web resource (html page)?

- **Schematron description:**

```
<sch:rule context="html:body">  
<sch:report test=" descendant::*[name()='h1']  
or descendant::*[name()='h2']  
or descendant::*[name()='h3']  
or descendant::*[name()='h4']  
or descendant::*[name()='h5']  
or descendant::*[name()='h6']">1</sch:report>  
</sch:rule>
```

- **OCCL description:**

```
context=body  
inv self.elements->  
exists(el e.name in Set{'h1', 'h2', 'h3', 'h4', 'h5', 'h6'})
```



A10.01.01.010.001-HTML01.001

- Does the inspected page, which uses frames, contain a `<noframes>` element?

- **Schematron description:**

```
<sch:rule context="html:html[child::*[name()='frameset']]">  
<sch:report test="descendant::*[name()='noframes']">1</sch:report>  
</sch:rule>
```

- **OCML description:**

context=html

inv self.children->exists(ele.name='frameset') implies
self.children->exists(ele.name='noframes')



Summary

- *Models* (High-level descriptions) used for automatic generation of real Code, thus achieving more productivity, quality, longevity
- Raising the level of abstraction on a broader scale
- Meta models are special models, represented in the same way
- Custom modelling languages will be more and more important
- MDA from OMG provides standards and standard Mappings to XML, Java, and CORBA™
- SMILE project is a HiA attempt to MDD and LDD

