Semantic Model-driven Engineering

Steffen Staab

Acknowledgements to students and colleagues@MOST project
http://most-project.eu
New level in Software Engineering

- Development with untyped languages
- Type-safe development with typed languages
- Constraint-safe development with constraint languages
- Consistency-based development with ontology languages
- Guided development with advising languages

Ontology-Driven Software Development (ODSD)
Marrying Ontologies and Software Technology

[ODSD]
Marrying Ontologies and Software Technology

Software Engineer

Software Modeling World (ModelWare)

EMOF
... MOF

Ontology Expert

Ontology World (OntologyWare)

OWL
RDF

Terminologies with $10^3$ to $10^6$ entities

Gene Ontology

Snomed

FMA

500+ terminologies in the B2B world

[ODSD]
Marrying Ontologies and Software Technology

Software Engineer

Software Modeling World (ModelWare)

EMOF

MOF

Ontology Expert

Ontology World (OntologyWare)

OWL

RDF

Terminologies with $10^3$ to $10^6$ entities

Gene Ontology

Snomed

FMA

500+ terminologies in the B2B world
The MOST Project

ODSD with Process Guidance

Consistency Preservation In ODSD

Foundational ODSD technology

Basic technology

[ODSD]
The MOST Project

Software Modeling World (ModelWare)

ODSD with Process Guidance

Consistency Preservation In ODSD

Foundational ODSD technology

Basic technology

Traceability

Integration of Metamodels and Ontologies

Ontology-based configuration

MDSD

Case Studies

Ontologies

[ODSD]

WeST

Steffen Staab
staab@uni-koblenz.de

Semantic MDE
7 of 25
The MOST Project

Software Modeling World (ModelWare)

ODSD with Process Guidance
- Ontology-Integrated Modeling

Consistency Preservation In ODSD
- Consistency Checking for Structural Models

Foundational ODSD technology
- Traceability

Basic technology
- MDSD

[ODSD]

Ontology World (OntologyWare)

TwoUse: A Platform for ODSD

Consistency Checking for Processes

Integration of Metamodels and Ontologies

Ontology-based configuration

Case Studies

Ontologies

Steffen Staab
staab@uni-koblenz.de
Semantic MDE
8 of 25
Semantic MDE

ONTOLGY BASED
DOMAIN SPECIFIC LANGUAGES
Stakeholders

Bridge Developer specifies Metamodelling Language

DSL Designer specifies DSL Metamodel

does not specify

DSL User builds Domain Model

requires Guidance and services

Constraints defined in [MODELS09] based on
Scenario at Comarch (PL)

- Modeling physical devices, e.g. Cisco network devices

Cisco 7603:

Domain Model:

- Every Cisco 7603 has at least 1 Configuration 7603.
- Every Configuration has at least 1 Slot in which a Supervisor Engine card is plugged in.
- A Configuration 7603 has exactly 3 Slots in which either a Hot Swappable OSM or SPA Interface card is plugged in.

Restrictions modeling a Cisco 7603 device:

- Every Cisco 7603 has at least 1 Configuration 7603.
- Every Configuration has at least 1 Slot in which a Supervisor Engine card is plugged in.
- A Configuration 7603 has exactly 3 Slots in which either a Hot Swappable OSM or SPA Interface card is plugged in.
• Domain Model: (inconsistent)

• Requirements of DSL User:
  • Consistency Checking
  • Debugging of domain models
DSL User interacting with PDDSL

- Domain Model: (consistent)

- Requirements of DSL User:
  - Consistency Checking
    - Debugging of domain models
  - Validate incomplete models
    - Guidance and explanations how to complete the model
DSL User interacting with PDDSL

• Domain Model: (inconsistent)

- Configuration
  - Slot
    - HotSwappableOSM
  - Slot
    - HotSwappableOSM
  - Slot
    - HotSwappableOSM

Explanation:
Configuration hasSlot some Slot and hasCard some SupervisorEngine

• Requirements of DSL User:
  • Consistency Checking
    • Debugging of domain models
  • Validate incomplete models
    • Guidance and explanations how to complete the model

[MODELS09]
**DSL User interacting with PDDSL**

- **Domain Model:**
  - Cisco7603
  - Configuration7603
  - Slot
  - Slot
  - Slot
  - HotSwappableOSM
  - SupervisorEngine

- **Requirements of DSL User:**
  1. Consistency Checking
     - Debugging of domain models
  2. Validate incomplete models
     - Guidance and explanations how to complete the model
  3. Suggestions of suitable domain concepts
  4. Use of services without any extra effort

[MODELS09]
Integrated Modeling – Modeling PDDSL

```java
class Device {
  reference hasConfiguration [1-]: Configuration;
}

class Cisco7603 extends Device {
  equivalentWith restrictionOn hasConfiguration
  with min 1 Configuration7603 {
  }
}

class Configuration {
  equivalentWith
    IntersectionOf(restrictionOn hasSlot with min 1 Slot,
    restrictionOn hasSlot some
    restrictionOn hasCard some SupervisorEngine) {
    reference hasSlot Slot;
  }
}

class Configuration7603 {
  equivalentWith
    IntersectionOf(restrictionOn hasSlot with min 1 Slot,
    restrictionOn hasSlot some
    restrictionOn hasCard some SupervisorEngine) {
    reference hasSlot Slot;
  }
}

class Slot {
  reference hasCard;
}
```

**Description Logics**
- Subset of first order logics
- Reasoning at two levels
  - Class level
  - Object Level
- Sound and complete reasoning
- Pragmatically efficient
Integration Bridge: Metamodelling PDDSL

Bridge Developer

Bridging Technology - Integration -

Ontoware TS

OWL2 Metamodell

Tools

Ontology

TBox

Querying Reasoning Technology

Modelware TS

Integrated (Meta-)Metamodell

Hybrid (Meta-)model

Ontoware

Model

Modelware

Ecore Metametamodell

DSL Metamodell

Model

Model

Integrated (Meta-)Metamodell

Model

Projection

Projection

Projection

Projection

Validation Constraint Technology

[DKE10]
### Evaluation: Stakeholders and measures

**GQM methodology reused**

**Productivity**
- Compare development time under comparable conditions

**Quality**
- Compare development time spent on bug fixing
- Assumption: development is continued until no errors exist

<table>
<thead>
<tr>
<th>Actor</th>
<th>Productivity</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bridge developer</td>
<td>Not measured</td>
<td>Not measured</td>
</tr>
<tr>
<td>DSL designer</td>
<td>manual, pure models vs manual, hybrid vs generalized, hybrid</td>
<td>Productivity measure * bug ratio</td>
</tr>
<tr>
<td>Domain expert/Customer expert</td>
<td>Cisco 7600 with PDDSL vs. Without PDDSL</td>
<td>Productivity measure * bug ratio</td>
</tr>
<tr>
<td>Domain user</td>
<td>Survey customers</td>
<td>Survey customers</td>
</tr>
</tbody>
</table>

[ECMFA10]
Evaluation DSL developer productivity & quality

- Implement consistency guidance services for DSL users/Domain experts
  - Instance validation
  - Types validation
  - Suggestions
- Includes abstract syntax integration
- Excludes concrete syntax integration
- Excludes requirements analysis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Metamodel integration</td>
<td>0</td>
<td>6 h</td>
<td>Re-use</td>
</tr>
<tr>
<td>Projection to OWL</td>
<td>0</td>
<td>40 h</td>
<td>(mostly) automated</td>
</tr>
<tr>
<td>Services implementation</td>
<td>98 h</td>
<td>28 h</td>
<td>Finished Apr 30</td>
</tr>
<tr>
<td>Testing &amp; Bugfixing</td>
<td>34 h</td>
<td>18 h</td>
<td>Finished Apr 30</td>
</tr>
</tbody>
</table>

Productivity improvement - 40 h = 30% (Finished Apr 30)

Quality improvement - 16 h = 47% (Finished Apr 30)
Semantic MDE

PROCESS REASONING
Supporting SAP Development of Processes

From business developer to software developer

BPMN at different levels of granularity

Select Applicant A

Hire Applicant B

Interview Applicant a1

Prepare Contract b1

Send Contract b2

Send Rejection a3

Send Acceptance a2
Transformation Bridge for Process Reasoning

WeST

Steffen Staab
staab@uni-koblenz.de
Semantic MDE
22 of 25

Ontoware TS
OWL2 Metamodel
Tools
Querying Reasoning Technology
Ontology
TBox
ABox
Transformation Definition
Transformation Engine
Model
Model
DSL Metamodel
Ecore Metametamodel
Target model
Source model
Target metamodel
Source metamodel
M3
conformsTo
M2
conformsTo
M1

[DKE10]
Mapping Process Steps

Select Applicant A

Interview Applicant a1

Prepare Contract b1

Send Rejection a3

Send Contract b2

Send Acceptance a2

Hire Applicant B

[DL 2009]
1. Eliminating parallel gateways:
   Executions remain the same
   - Exponential complexity (O(n!))

2. Reduce execution sets to predecessor and successor sets:
   - **Predecessor sets:**
     \( \text{PS(b11)} = \{a11\}; \)
     \( \text{PS(a21)} = \{b11\}, \text{ etc.} \)
   - **Successor sets:**
     \( \text{SS(b11)} = \{a21,b22\}; \)
     \( \text{SS(a21)} = \{b21\}, \text{ etc.} \)
   - Execution sets subsumption can be reduced to PS/SS sets subsumptions \([DL2009]\)
Example for survey question without support

- 3 processes in a refinement hierarchy
- Some models have multiple errors
- Multiple-choice: 2-4 answer options, multiple correct options possible

Experiments performed by J. Lemcke & colleagues, SAP AG
Example for survey question with support

2 main surveys

- Survey 1 with no support to modeller
- Survey 2 with error highlights in the BPMN diagrams or guidance comments, respectively

Experiments performed by J. Lemcke & colleagues, SAP AG
Results per case study

- Process refinement case study
  - Quality: QI = 57%, \( \text{sav}_q = 36\% \)
  - Productivity: PI = 192%, \( \text{sav}_p = 66\% \)

- Process guidance case study
  - Quality: QI = 66%, \( \text{sav}_q = 40\% \)
  - Productivity: PI = 254%, \( \text{sav}_p = 72\% \)

Experiments performed by J. Lemcke & colleagues, SAP AG
Semantic MDE

CONCLUSION
### Bridge Classification

<table>
<thead>
<tr>
<th>Motivation</th>
<th>Bridging definition layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M3 layer</td>
</tr>
</tbody>
</table>
| Bridging     | Integration | • Extended expressiveness of metamodeling language  
|              |           | • Additional Restrictions for M1 models       |
|Transformation|           | • Leveraging models by ontologies  
|              |           | • Knowledge represented independently of modeling language |
|              | • Replacement of coexistence between metamodels (+ conforming models) and ontologies  
|              |           | • Enabling ontology-based querying |
|              |           | • Transformations of models to formal ontological representations  
|              |           | • Enabling formal semantics |
class Configuration7603 extends Configuration
    equivalentWith (hasSlot with exactly 3 Slot) and
    (hasSlot some (hasCard some (hasSlot some HotSwappable)))
{
...

WeST  Steffen Staab  staab@uni-koblenz.de  Semantic MDE  31 of 25
Original Contributions: **Ontology Reasoning**
- Defined for class diagrammes „before“ [BCG05], but also
  - Domain-specific languages
  - Process languages

**Lessons Learned**
- Marriage is nontrivial
  - formalization not straightforward, but re-useable!
  - terminological problems need to be reconciled between the two pillars:
    - E.g. open/closed world vs. draft/deployed models
- **Productivity gains**
Knowledge by the User & for the User

- Domain models must be included, maintained and aligned with other models in the lifecycle
  - Families of business processes [CAiSE 2011]
  - Software product lines [submitted]
- Terminologies and Ontologies are undervalued re-useable artefacts in software development
- Knowledge not contained within one software project alone
  - Semantic Web can span many projects! [TKB10;under development]
- Semantic technologies respond to these needs
http://west.uni-koblenz.de/Projects/twouse
Transforming and Weaving Ontologies and UML for Software Engineering


Further core publications for this area


Thank you for your attention!

Your Questions?