Exposé for B.Sc. Thesis

Megamodel-driven Traceability Recovery & Exploration in an O/R/X-Mapping scenario along the mereological aspects of software artifacts

Maximilian Meffert
210101205

University of Koblenz-Landau

1 Introduction

This exposé[1] outlines the thesis:

Megamodel-driven Traceability Recovery & Exploration in an O/R/X-Mapping scenario along the mereological aspects of software artifacts

for acquiring the degree Bachelor Science (B.Sc.) in Computer Science. The central topic of the thesis will be the study of traceability recovery in a megamodel governed environment, that is MegaL, a modeling language for such models. Furthermore a system for exploring the recovered traceability links will be developed.

2 Motivation

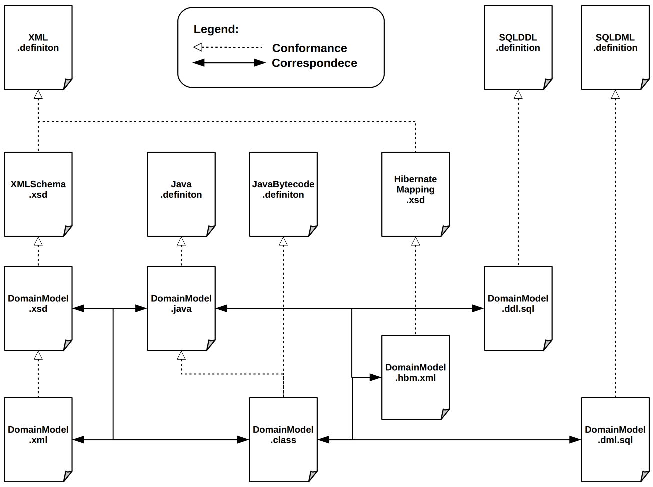
A common task during the development of software systems is to persist and serialize a domain model. For instance, consider a simple ReST-ful web-service where data is stored in a database and served via HTTP in serialized form, e.g. XML. Given such a system, one can observe correspondences (structural similarities) and conformances (compliance with a definition) between different artifacts, i.e. manifestations of the same domain model. Figure 1 shows a non exhaustive depiction of such relationships in O/R/X scenario.

Upon closer inspection, one can also observe that correspondence and conformance are not just interrelations of the artifacts as wholes. In fact the same relations can be found between fragments of the artifacts. Table 1 shows the fragments corresponding to a Java class property in the O/R/X scenario.

The motivating idea behind the thesis is that correspondence and conformance relationships between linguistic artifacts are trace-links left behind by transformations in the sense of [3]. Thus, it should be possible to recover these links with a model describing these linguistic relations. A technology capable of modeling such *"linguistic architectures"*[1] is MegaL[2] [4]

[1] `http://www.softlang.org/info:expose`

[2] `http://www.softlang.org/megal/`



The depicted relationships may not be exhaustive.

Fig. 1. O/R/X Correspondence & Conformance

Language	Fragment Type	Fragment Text
Java	Class Property	`public String name;`
XSD	Attribute Definition	`<xs:attribute name="name" type="xs:string"/>`
XML	Attribute	`name="Alan Turing"`

Hibernate XML	Property Mapping	`<property access="field" generated="never" lazy="false" name="name"`
		`optimistic-lock="true" type="java.lang.String" unique="false">`
		`    <column name="name"/>`
		`</property>`

SQL/DDL	Column Definition	`‘name‘ varchar(255) DEFAULT NULL,`

Table 1. O/R/X Fragment Correspondence

3 Background

The thesis will be based but is not limited to aspects of the following topics:

4 Research Hypotheses & Questions

4.1 Research Hypotheses

Correspondence [3] axiomatizes a strict one-to-one correspondence of software artifacts as follows: Given there is an arbitrary relationship R between to languages: $R \subseteq L_1 \times L_2$, for instance, R may be a transformation from one language to the other.

Then two artifacts $a_1 \in L_1$ and $a_2 \in L_2$ correspond to each other only if for each part of one artifact there exists exactly one corresponding part of the other:

$$\begin{aligned}&(a_1, a_2) \in R\\ &\wedge \forall b_1 \in L_1 : b_1 \text{ partOf } a_1 \Rightarrow (\exists! b_2 \in L_2 : b_2 \text{ partOf } a_2 \wedge b_1 \text{ correspondsTo}_R\ b_2)\\ &\wedge \forall b_2 \in L_2 : b_2 \text{ partOf } a_2 \Rightarrow (\exists! b_1 \in L_1 : b_1 \text{ partOf } a_2 \wedge b_2 \text{ correspondsTo}_R\ b_1)\\ &\Rightarrow a_1 \text{ correspondsTo}_R\ a_2\end{aligned}$$

However, [3] also notes that this kind of correspondence may be unrealistic since real world artifacts may contain two or more parts corresponding to only one part in another artifact, e.g. an XSD documents can contain an element and a complex type definition corresponding to only on Java class declaration. For this reason the thesis will assume a weaker correspondence hypothesis.

Fragment Correspondence Hypothesis One objective of the thesis should be to provide empirical assurance for the axiom above in the sense that correspondence is in fact mereologically induced. So if two artifacts a_1 and a_2 are assumed to correspond to each other, e.g. a_2 could be the result of a transformation of a_1, then parts of both artifacts should exist which also correspond to each other.

$$\forall x \in \text{Any} : x \in L \subseteq \text{Any} \Leftrightarrow \exists d \in D \subseteq \text{Any} : x \text{ conformsTo } d$$

$$\begin{aligned}&\forall a_1 \in L_1, a_2 \in L_2 : a_1 \text{ correspondsTo}_R\ a_2\\ &\ \Rightarrow \exists b_1 \in L_1, b_2 \in L_2 : b_1 \text{ partOf } a_2 \wedge b_2 \text{ partOf } a_2 \wedge b_1 \text{ correspondsTo}_R\ b_2\end{aligned}$$

Fragment Conformance Hypothesis If two artifacts are in a Conformance-Relationship, they contain constituent parts in the same relation to each other, i.e.:

$$\begin{aligned}&A_1 \text{ conformsTo } A_2\\ &\ \Rightarrow \exists a_1, a_2 : a_1 \text{ partOf } A_1 \wedge a_2 \text{ partOf } A_2 \wedge a_1 \text{ conformsTo } a_2\end{aligned}$$

4.2 Research Questions

Research Questions are:

RQ1 description

5 Objectives

Objectives for the thesis are:

O1 Implementation of a MegaL/Xtext-extension capable of recovering traceability links representing PartOf-, Correspondence- and Conformance-Relationships between code fragments.

O2 Implementation of a MegaL/Xtext-extension allowing for an user to visually explore traceability links, i.e. PartOf-, Correspondence- and Conformance-Relationships between code fragments.
O3 Providing an extensive discussion comparing MegaLwith related approaches on traceability recovery.
O4 Providing an extensive discussion comparing MegaLwith related approaches on ontologies for software artifacts or software engineering in general.

6 Methodology

Thesis and research will utilize an example-driven approach inspired by the 101system[3]. For this, the system to study will be an imaginary Human Resource Management System (HRMS). Figure 2 shows an UML-Class-Diagram depicting the model of this system.

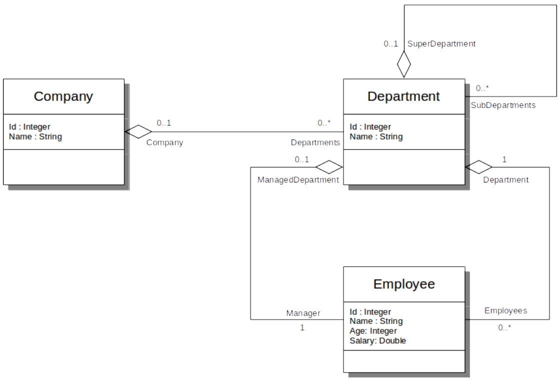


Fig. 2. The Human Resource Management System Model

This HRMS will be implemented in plain Java with two scenarios in mind:

1. XML-Binding with JAXB
2. Persitence with JPA/Hibernate

Both scenarios will be studied with the concrete instance of the HRMS model depicted in figure 3.

7 Structure of the Thesis

The interim structure[4] of the thesis is depicted in figure 4. 2

[4] [1] [6] [2] [5] [3]

[3] `https://101wiki.softlang.org/101system`

[4] `http://softlang.wikidot.com/info:thesis-structure`

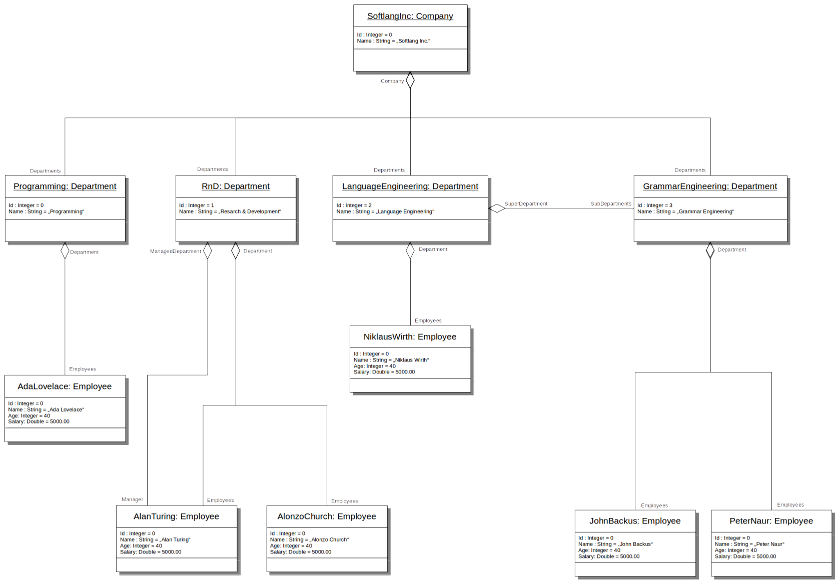


Fig. 3. The Human Resource Management System Instance called *Softlang Inc.*

1. **Introduction**
2. **Background**
3. **Related Work**
4. **Methodology**
5. **Requirements**
6. **Design**
7. **Implementation**
8. **Case Study**
9. **Analysis/Results**
10. **Conclusion**

Fig. 4. Structure of the Thesis

References

1. Favre, J., Lämmel, R., Varanovich, A.: Modeling the linguistic architecture of software products. In: Model Driven Engineering Languages and Systems - 15th International Conference, MODELS 2012, Innsbruck, Austria, September 30-October 5, 2012. Proceedings. pp. 151–167 (2012), `http://dx.doi.org/10.1007/978-3-642-33666-9_11`
2. Favre, J., Nguyen, T.: Towards a megamodel to model software evolution through transformations. Electr. Notes Theor. Comput. Sci. 127(3), 59–74 (2005), `http://dx.doi.org/10.1016/j.entcs.2004.08.034`
3. Lämmel, R.: Coupled software transformations revisited. In: Proceedings of the 2016 ACM SIGPLAN International Conference on Software Language Engineering,

Amsterdam, The Netherlands, October 31 - November 1, 2016. pp. 239–252 (2016), http://dl.acm.org/citation.cfm?id=2997366

4. Lämmel, R., Varanovich, A.: Interpretation of linguistic architecture. In: Modelling Foundations and Applications - 10th European Conference, ECMFA 2014, Held as Part of STAF 2014, York, UK, July 21-25, 2014. Proceedings. pp. 67–82 (2014), http://dx.doi.org/10.1007/978-3-319-09195-2_5
5. Lämmel, R.: Technology modeling with megal in software development (2015), http://userpages.uni-koblenz.de/~laemmel/pttcourse/slides/technology.pdf
6. Varzi, A.C.: Parts, wholes, and part-whole relations: The prospects of mereotopology. Data Knowl. Eng. 20(3), 259–286 (1996), http://dx.doi.org/10.1016/S0169-023X(96)00017-1