A Case of Quality Prediction of Architecture Knowledge Sharing through Model Mapping

Peng Liang, Anton Jansen, Paris Avgeriou

January, 2008
RUG-SEARCH-08-L01

Software Engineering and Architecture (SEARCH) Group
Department of Computing Science
University of Groningen
PO Box 407, 9700 AK Groningen
The Netherlands

Abstract

In this report, we introduce the AK sharing activity with a query-based scenario, and the motivation for the prediction of AK sharing quality prediction. In the end, a concrete case of quality prediction of AK sharing through model mapping was presented with assumptions.

This research has been partially sponsored by the Dutch Joint Academic and Commercial Quality Research & Development (Jacquard) program on Software Engineering Research via contract 638.001.406 GRIFFIN: a GRId For inFormatIoN about architectural knowledge.

Content

1 Introduction

Software architecture is considered of paramount importance to the software development life cycle [1]. It is a key artifact for the early analysis of the system, as it facilitates stakeholder communication and understanding, and drives both system construction and evolution. Various authors [9][10][11][12][13] and the IEEE [4] have proposed their own AK models to document AK concepts and their relationships. Some of these concepts and relationships are different, while others are largely overlapping. These discrepancies between the AK domain models can hamper the effective sharing of AK between these organizations, which in turn results into misunderstandings among stakeholders, expensive system evolution, and limited reusability of architectural artifacts [14].

1.1 Query-based AK-sharing scenario

In our perspective, we envision AK sharing in a heterogeneous AK repository setting, in which different repositories are part of the architectural knowledge GRID. Each repository contains one AK model and its instances. A user can retrieve AK from all involved AK repositories transparently without being conscious of the underlying model differences. To quantify the quality, we use this specific user scenario, which is a key activity for knowledge sharing. The query is a precise request for information, typically keywords combined with Boolean operators and other modifiers.

The query-based scenario is shown in Figure 1. A user who understands only AK model *T* queries the repository of AK model *S* using concepts from AK model *T* as query keywords. The conceptual difference between AK model *S* and *T* poses a problem for AK sharing. The concepts from model *T* queried does not exist (or exists, but has a different meaning) in model *S*. Thus, the repository of model *S* cannot return any data. Using concept mappings from model *S* to *T*, the repository of model *S* could partially return data to the user.

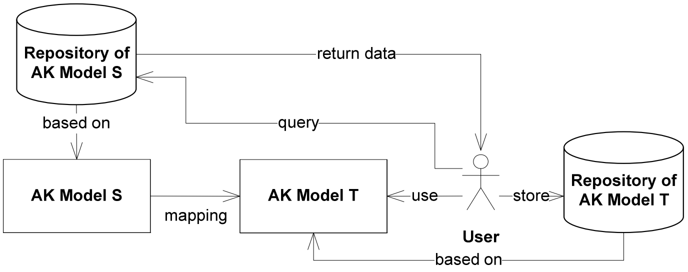


Figure 1 Query-based scenario for AK sharing

2 Motivation

The quality of AK sharing is not only dependant on the models and mappings involved, but also on the actual instances of these models. Only with these instances the real cost and AK sharing quality can be determined. However, creating these instances requires considerable effort, as human intervention is required. Even more troublesome is the fact that most of this effort needs to be redone when due to further insight domain models or mappings are changed. Hence, we would like to *predict* the cost and quality of AK sharing in advance before effort is spend on creating instances. This report contributes such a prediction model for both the direct and indirect mapping approaches.

3 Assumptions

By assigning more practical assumptions, we can come up with better prediction model, with which the prediction of set distribution is closer to the real case. In this report, we make assumptions as follows:

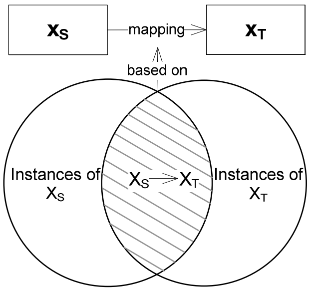
- All instances in a AK repository are evenly distributed over the AK concepts;
- We use a perfect instance mapping tool for the instance mapping by which all instances will be mapped into correct concept smartly;

4 Prediction Rules

4.1 Calculation rules for prediction of set distribution

The AK instances are mapped based on concept mapping relationships between AK models, and AK model mapping is composed of a set of concept mapping relationships. With assumptions of Simple Mapping Quality Prediction Model, the prediction of sets distribution (D'_{DM}, D'_{IM}, and $D'_{DM\cap IM}$) can be calculated using set distribution prediction of all individual concept mapping relationships in a instances even distribution way. To be concise and to differentiate from **set distribution** concept, the set distribution of individual concept mapping relationship is renamed **concept mapping set distribution**. The calculation rules for the prediction of **concept mapping set distribution** based on different concept mapping relationships are presented after the introduction of several mathematical symbols:

- x_S and x_T are concepts from two AK models S and T. A concept mapping relationship from x_S to x_T is normalized as a triple $<x_S,m,x_T>$, in which m is the mapping relationship from concept x_S to x_T or mapping rules applicable from x_S to x_T. The concept mapping relationships includes `equivalentClass`, `subClassOf`, `superClassOf` (`inverseOf subClassOf`), `disjointWith` and `noMatchingPair` which can be represented by RDF [16]/OWL [17] constructors and deduced by RDF/OWL reasoners. For easy introduction of ontology mapping representation using RDF/OWL constructors, other mapping relationships like `partOf`, `compositionOf` are not included.
- $|x|$ is the number of concept x. In triple $<x_S,m,x_T>$,
  (1) if $m\neq$`noMatchingPair`, then $|x_S|=1$ and $|x_T|\geq1$, which means that the concept mapping relationship can be 1 to 1 or 1 to multiple;
  (2) if $m=$`noMatchingPair`, then $|x_S|=1$ and $|x_T|=0$, which means that there is no mapping concept for x_S;
- $D'(<x_S,m,x_T>)$ is the prediction of **concept mapping set distribution** of individual concept mapping relationship represented by triple $<x_S,m,x_T>$, and its real value $D(<x_S,m,x_T>)$ is the percentage of the number of instances mapped from concept x_S to x_T to the instances number of x_S as shown in Figure 2, in which both x_S and $x_S\rightarrow x_T$ are sets. With the assumption of even instances distribution over concepts in SMQPM, we can assume that the instance number of all concepts is 1 (i.e. $|x_S|=1$) for easy calculation, thus for the prediction $D'(<x_S,m,x_T>)$, we can get:
  (1) $0\leq D'(<x_S,m,x_T>)\leq1$;
  (2) $D'(<x_S,m,x_T>)=0$, if $m=$`noMatchingPair`;
  (3) $D'(<x_S,m,x_T>)=1$, if $m\neq$`noMatchingPair` and all instances of concept x_S can be mapped as instance of concepts x_T;



$$D(<x_S,m,x_T>)=\frac{|x_S\rightarrow x_T|}{|x_S|}$$

Figure 2 Concept mapping set distribution calculation with mapping relationship from x_S to x_T

The calculation rules for the prediction of **concept mapping set distribution** $D'(<x_S,m,x_T>)$ based on different concept mapping relationships are presented as follows, and note that besides $D'(<x_S,m,x_T>)$, **side-effect concept mapping set distribution** caused by concept mapping relationship $<x_S,m,x_T>$ can take place, which will be descried in different calculation rules in details.

4.1.1 equivalentClass

R1: `equivalentClass` concept mapping relationship

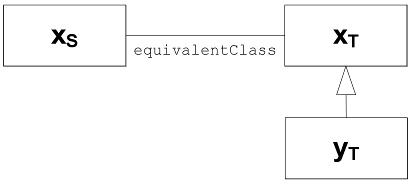


Figure 3 `equivalentClass` concept mapping relationship from x_S to x_T

- **Concept mapping set distribution**
  Calculation: $D'(<x_S,m,x_T>)=1$
  Reason: since x_S `equivalentClass` x_T, any instance of x_S is instance of x_T, i.e. $|x_S \rightarrow x_T|=|x_S|$.
- **Side-effect concept mapping set distribution**
  Condition: y_T is a concept in model T, and is a direct `subClassOf` x_T, and all concepts as y_T are `disjointWith` each other.
  Calculation: $D'(<x_S,m,y_T>)=1/(N(y_T)+1)$, in which $N(y_T)$ is the number of concepts as y_T.
  Reason: With the assumption of even distribution of instances with SMQPM, all instance of x_T will be distributed evenly in its direct subclasses (as y_T) plus 1 dummy subclass, which represents the concept of instances not covered by all the explicit direct subclasses as shown in Figure 4. All concepts as y_T are `disjointWith` each other, so there is no instances intersection between set of instances of different y_T. With $|x_S \rightarrow x_T|=|x_S|$, $|x_S \rightarrow y_T|=|x_S \rightarrow x_T|*1/(N(y_T)+1)=|x_S|*1/(N(y_T)+1)$.

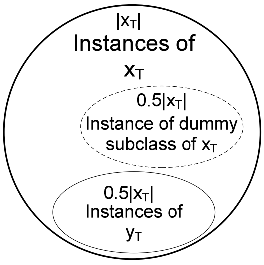


Figure 4 Instances mapping of internal `subClassOf` relationship with 1 subclass case

4.1.2 subClassOf

R2: `subClassOf` with `disjointWith` concept mapping relationship

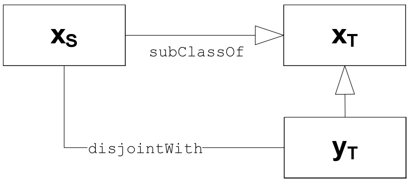


Figure 5 `subClassOf` concept mapping relationship from x_S to x_T with x_S `disjointWith` y_T

- **Concept mapping set distribution**
  Calculation: $D'(<x_S,m,x_T>)=1$
  Reason: since x_S `subClassOf` x_T, any instance of x_S is instance of x_T, i.e. $|x_S \rightarrow x_T|=|x_S|$.
- **Side-effect concept mapping set distribution**
  Condition: y_T is a concept in model T, and is a direct `subClassOf` x_T, and x_S `disjointWith` y_T.
  Calculation: $D'(<x_S,m,y_T>)=0$
  Reason: x_S `disjointWith` y_T, so there is no instances intersection between set of instances of x_S and y_T, $|x_S \rightarrow y_T|=0$.

R3: `subClassOf` without `disjointWith` concept mapping relationship

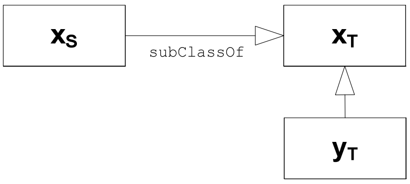


Figure 6 `subClassOf` concept mapping relationship from x_S to x_T without x_S `disjointWith` y_T

- **Concept mapping set distribution**
  Calculation: $D'(<x_S,m,x_T>)=1$
  Reason: the same reason as that for concept mapping set distribution in R2.
- **Side-effect concept mapping set distribution**

Condition: y_T is a concept in model *T*, and is a direct `subClassOf` x_T, and x_S is not `disjointWith` y_T, which is a default concept mapping relationship between x_S and y_T if no mapping relationship defined between them. All concepts as y_T are `disjointWith` each other.
Calculation: $D'(<x_S,m,y_T>)=1/(N(y_T)+1)$
Reason: the same reason as that for side-effect concept mapping set distribution in R1.

4.1.3 superClassOf

R4: `superclassOf` concept mapping relationship

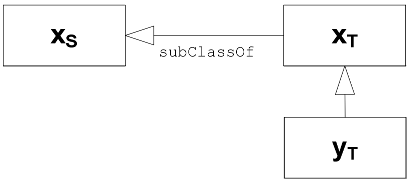


Figure 7 `superClassOf` concept mapping relationship from x_S to x_T

- **Concept mapping set distribution**
  Calculation: $D'(<x_S,m,x_T>)=1/(N(x_T)+1)$
  Reason: in this mapping relationship, x_T `subClassOf` x_S, so the same reason as that for side-effect concept mapping set distribution in R1.
- **Side-effect concept mapping set distribution**
  Condition: y_T is a concept in model *T*, and is a direct `subClassOf` x_T, and x_S is not `disjointWith` y_T, which is a default concept mapping relationship between x_S and y_T if no mapping relationship defined between them. All concepts as x_T are `disjointWith` each other, and all concepts as y_T are `disjointWith` each other.
  Calculation: $D'(<x_S,m,y_T>)=1/(N(y_T)+1)*(N(x_T)+1)$
  Reason: the same reason as that for side-effect concept mapping set distribution in R1. With $|x_S \rightarrow x_T|=|x_S|*1/(N(x_T)+1)$, $|x_S \rightarrow y_T|=|x_S \rightarrow x_T|*1/(N(y_T)+1) =|x_S|*1/(N(y_T)+1)*(N(x_T)+1)$.

4.1.4 noMatchingPair

R5: `noMatchingPair` concept mapping relationship

Figure 8 `noMatchingPair` concept mapping relationship from x_S

- **Concept mapping set distribution**
  Calculation: $D'(<x_S,m,x_T>)=0$
  Reason: since x_S `noMatchingPair` x_T, any instance of x_S is not instance of x_T, i.e. $|x_S \rightarrow x_T|=0$.

4.2 Prediction of sets distribution for precision and recall

In this section, the calculation expression for the prediction of sets distribution (i.e. D'_{DM}, D'_{IM}, and $D'_{DM \cap IM}$) are presented based on calculation rules defined in section 0.

4.2.1 D'$_{DM}$ calculation

By the definition in section 3.4.2, $D'_{DM}=|DM|/|S|$. With the assumption of even instances distribution over concepts in SMQPM and the instance number of all concepts is 1 (i.e. $|x_S|=1$) defined in section 0, the value of $|S|$ is the number of concepts in AK model *S*, and $|DM|$ can be calculated by summary of **concept mapping set distribution** and **side-effect concept mapping set distribution** from concepts of model *S* to *T*. The concept mapping relationship from one concept x_S to concepts in model *T* can be 1 to 1 or 1 to multiple, so we use x_T to represent the set of concepts mapped from x_S. Detailed calculation expressions are shown below, in which *n* is the number of mapping relationships (direct or indirect caused calculation rules) from x_S to *T*, and *NoC(S)* is the number of concepts in AK model *S*:

$$D'(<x_S,m,x_T>)=\sum_{j=1}^{n}D'(<x_S,m,x_T{}^{j}>)\;(x_T=\{x_T{}^{1},..x_T{}^{j}..,x_T{}^{n}\}\subset T);$$

$$D'_{DM}=\frac{|DM|}{|S|}=\frac{\sum\limits_{i=1}^{NoC(S)}D'(<x_S{}^{i},m,x_T{}^{i}>)}{NoC(S)}$$

4.2.2 D'$_{IM}$ calculation

D'_{IM} is prediction of set distribution based on concept mapping from S to T with indirect mapping, in which twice concept mapping relationships from concept of model S to C, and from mapped concepts in model C to T will occur. D'_{IM} can be calculated in the same way as D'_{DM} does. The only difference is that we use $D_{\_C}'(<x_S,m,x_T>)$ to represent the prediction of set distribution of individual concept mapping relationship based on twice mapping relationships represented by triples $<x_S,m,x_C>$ and $<x_C,m,x_T>$, in which x_C represents the set of concepts in central model C mapped from x_S , and x_T represents the set of concepts mapped from x_C. For distinguishability from other kinds of **concept mapping set distribution**, $D_{\_C}'(<x_S,m,x_T>)$ is named **combined concept mapping set distribution**, and its calculation can be described in two steps. In the first step, the **concept mapping set distribution** for each $x_C{}^j$ (concept mapped from x_S to C) is calculated by summary of product of **concept mapping set distribution** and **side-effect concept mapping set distribution** from x_S to $x_C{}^j$ and $x_C{}^j$ to T. In the second step, the **concept mapping set distribution** for x_S is calculated by summary of the **concept mapping set distribution** for each $x_C{}^j$ mapped from x_S. Detailed calculation expressions are shown below, in which n is the number of mapping relationships (direct or indirect caused calculation rules) from x_S to C, and $l(j)$ is a function of parameter j representing the number of mapping relationships (direct or indirect caused calculation rules) from $x_C{}^j$ to T:

$$D_{\_C}'(<x_S,m,x_T>)=\sum_{j=1}^{n}\left(\sum_{k=1}^{l(j)}D'(<x_S,m,x_C{}^{j}>)\times D'(<x_C{}^{j},m,x_T{}^{k}>)\right)$$

$$(x_C=\{x_C{}^{1},..x_C{}^{j}..,x_C{}^{n}\}\subset C,$$

$$x_T=\{x_T{}^{1},..x_T{}^{k}..,x_T{}^{l(1)}\}\cup\ldots\{x_T{}^{1},..x_T{}^{k}..,x_T{}^{l(j)}\}\ldots\cup\{x_T{}^{1},..x_T{}^{k}..,x_T{}^{l(n)}\}\subset T);$$

$$D'_{IM}=\frac{|IM|}{|S|}=\frac{\sum\limits_{i=1}^{NoC(S)}D_{\_C}'(<x_S{}^{i},m,x_T{}^{i}>)}{NoC(S)}.$$

4.2.3 D'$_{DM\cap IM}$ calculation

$D'_{DM\cap IM}$ is prediction of set distribution of the instances that belong to both the DM and IM sets, and it can be calculated in the nearly same way as D'_{IM} does. The only difference is that the **combined concept mapping set distribution** in D'_{IM}, whose concept mapping relationship (caused indirectly by twice concept mappings) does not belong to direct concept mapping relationships from S to T, should be filtered out because this kind of **combined concept mapping set distribution** is not relevant to the **concept mapping set distribution** in D'_{DM}. We use $D_{R\_C}'(<x_S,m,x_T>)$ to represent **relevant combined concept mapping set distribution** in D'_{IM}, and its calculations expression is the same as calculation expression of $D_C'(<x_S,m,x_T>)$ except for an additional parameter r (r=1 when **combined concept mapping set distribution** is relevant , and r=0 when it is not). Detailed calculation expressions are shown below, in which other parameters except for r are the same meaning as in section 4.3.2:

$$D_{R\_C}'(<x_S,m,x_T>)=\sum_{j=1}^{n}\left(\sum_{k=1}^{l(j)}D'(<x_S,m,x_C{}^{j}>)\times D'(<x_C{}^{j},m,x_T{}^{k}>)\times r\right)$$

$$(x_C=\{x_C{}^{1},..x_C{}^{j}..,x_C{}^{n}\}\subset C,$$

$$x_T=\{x_T{}^{1},..x_T{}^{k}..,x_T{}^{l(1)}\}\cup\ldots\{x_T{}^{1},..x_T{}^{k}..,x_T{}^{l(j)}\}\ldots\cup\{x_T{}^{1},..x_T{}^{k}..,x_T{}^{l(n)}\}\subset T);$$

$$D'_{DM\cap IM}=\frac{|DM\cap IM|}{|S|}=\frac{\sum\limits_{i=1}^{NoC(S)}D_{R\_C}'(<x_S{}^{i},m,x_T{}^{i}>)}{NoC(S)}.$$

5 Cases of Quality Prediction of AK Sharing

5.1 LOFAR model

The domain model proposed for the AK documentation for the LOFAR projects, which are due to the long development of more than 10 years, and architectural decisions need to be shared and used over 25 years. The concept mapping between Astron domain model and refined Griffin core model is specified in Figure 9.

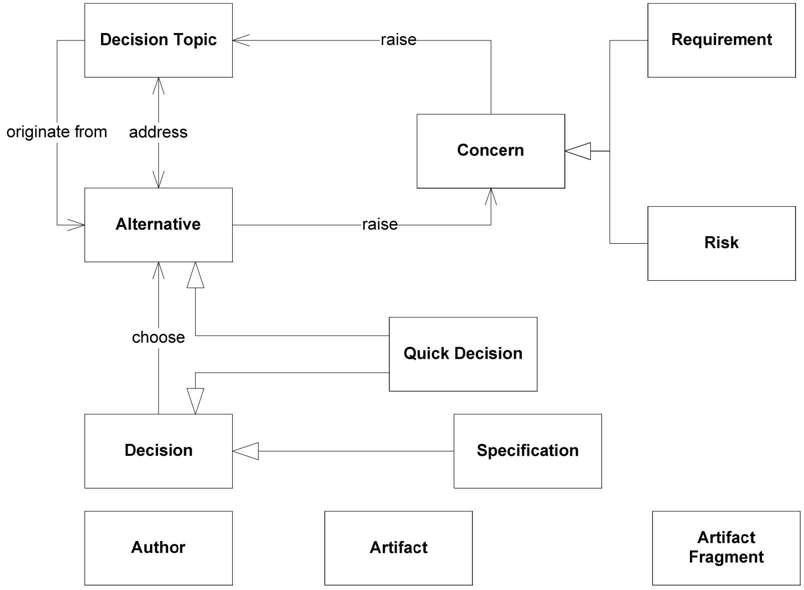


Figure 9 Concepts in LOFAR Model

5.2 Central model

The refined Griffin core model [15] as shown in Figure 10 is taken as the central model for AK model mapping.

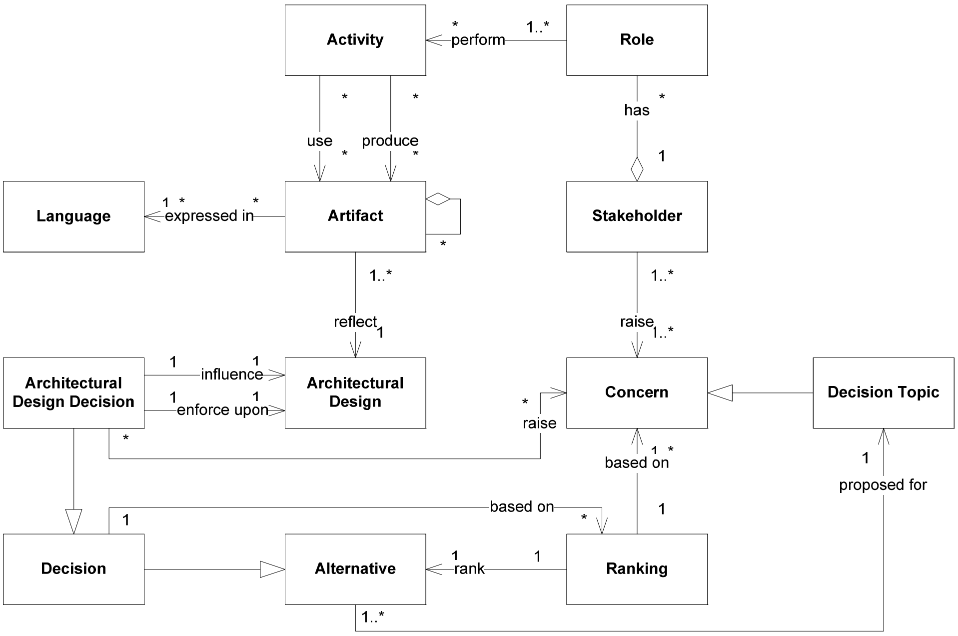


Figure 10 Refined Griffin Core Model represented in UML

5.3 Kruchten’s ontology

Kruchten’s ontology proposed in [9] for documenting mainly Architectural Design Decision, and the concept mapping between Kruchten’s ontology and refined Griffin core model is specified in Figure 11. The exceptional concept mappings are: *Structural Decision*, *Behavioral Decision*, and *Ban Decision* are `subClassOf` *Existence Decision*, *Constraint*, *Design Rule*, and *Guideline* are `subClassOf` *Property Decision*, *Organization*, *Process*, *Technology*, and *Tool* are `subClassOf` *Executive Decision*. The *Design Decision* is `sameAs` Architectural Design Decision or Alternative based on the value of *State*. The concept of *Risk*, *Requirement*, *Plan*, and *Design Element* are not the concepts from Kruchten’s ontology, but the concepts traceable from Kruchten’s ontology, and we map them onto the concepts in the refined Griffin core model as well.

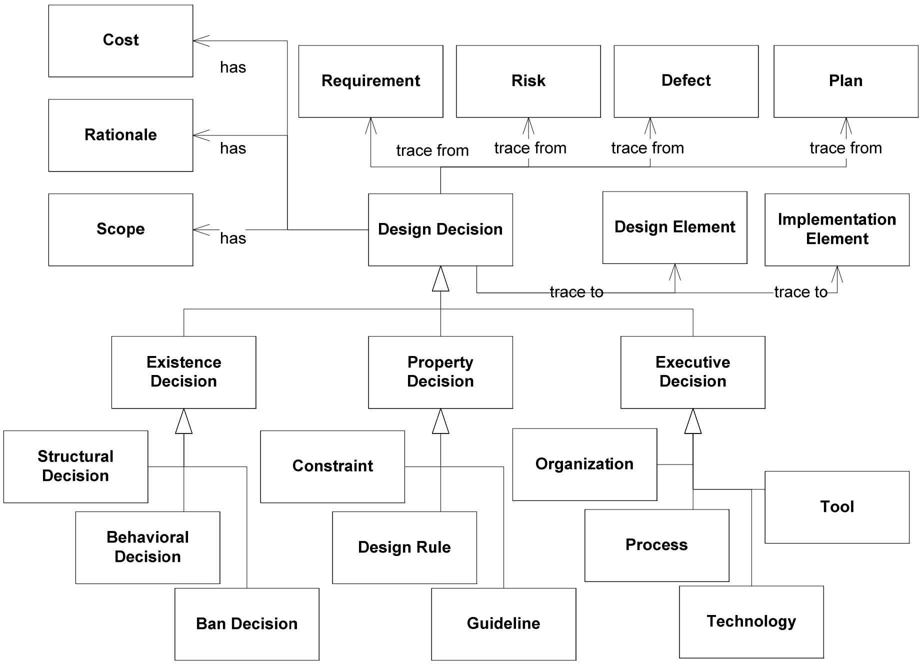


Figure 11 Concepts in Kruchten’s ontology

5.4 Mapping relationships with prediction

5.4.1 D'_{DM} Prediction of set distribution with direct mapping from model S to T

Table 1 Direct mapping relationships from LOFAR domain model to Kruchten’s ontology with $D'(<x_S,m,x_{T>})$ value

Concept of LOFAR domain model	Relationship/Rule	Concept of Kruchten’s ontology	$D'(<x_S,m,x_{T>})$
Author	`noMatchingPair`		0
Artifact	`superClassOf`	Design Element	1/3
Artifact	`superClassOf`	Implementation Element	1/3
Artifact Fragment	`superClassOf`	Design Element	1/3
Artifact Fragment	`superClassOf`	Implementation Element	1/3
Concern	`superClassOf`	Requirement	1/7
Concern	`superClassOf`	Risk	1/7
Concern	`superClassOf`	Defect	1/7
Concern	`superClassOf`	Plan	1/7
Concern	`superClassOf`	Cost	1/7
Concern	`superClassOf`	Scope	1/7
Requirement	`equivalentClass`	Requirement	1
Risk	`equivalentClass`	Risk	1
Decision Topic	`equivalentClass`	Scope	1
Decision	`subClassOf`	Design Decision	1
Decision	`R3`	Existence Decision	1/4
Decision	`R3`	Property Decision	1/4
Decision	`R3`	Executive Decision	1/4

Decision	`R4`	Structural Decision	1/4*1/4
Decision	`R4`	Behavioral Decision	1/4*1/4
Decision	`R4`	Ban Decision	1/4*1/4
Decision	`R4`	Constraint	1/4*1/4
Decision	`R4`	Design Rule	1/4*1/4
Decision	`R4`	Guideline	1/4*1/4
Decision	`R4`	Organization	1/4*1/5
Decision	`R4`	Process	1/4*1/5
Decision	`R4`	Technology	1/4*1/5
Decision	`R4`	Tool	1/4*1/5
Alternative	`subClassOf`	Design Decision	1
Alternative	`R3`	Existence Decision	1/4
Alternative	`R3`	Property Decision	1/4
Alternative	`R3`	Executive Decision	1/4
Alternative	`R4`	Structural Decision	1/4*1/4
Alternative	`R4`	Behavioral Decision	1/4*1/4
Alternative	`R4`	Ban Decision	1/4*1/4
Alternative	`R4`	Constraint	1/4*1/4
Alternative	`R4`	Design Rule	1/4*1/4
Alternative	`R4`	Guideline	1/4*1/4
Alternative	`R4`	Organization	1/4*1/5
Alternative	`R4`	Process	1/4*1/5
Alternative	`R4`	Technology	1/4*1/5
Alternative	`R4`	Tool	1/4*1/5
Quick Decision	`subClassOf`	Design Decision	1
Quick Decision	`R3`	Existence Decision	1/4
Quick Decision	`R3`	Property Decision	1/4
Quick Decision	`R3`	Executive Decision	1/4
Quick Decision	`R4`	Structural Decision	1/4*1/4
Quick Decision	`R4`	Behavioral Decision	1/4*1/4
Quick Decision	`R4`	Ban Decision	1/4*1/4
Quick Decision	`R4`	Constraint	1/4*1/4
Quick Decision	`R4`	Design Rule	1/4*1/4
Quick Decision	`R4`	Guideline	1/4*1/4
Quick Decision	`R4`	Organization	1/4*1/5
Quick Decision	`R4`	Process	1/4*1/5
Quick Decision	`R4`	Technology	1/4*1/5
Quick Decision	`R4`	Tool	1/4*1/5
Specification	`subClassOf`	Design Decision	1
Specification	`R3`	Existence Decision	1/4
Specification	`R3`	Property Decision	1/4
Specification	`R3`	Executive Decision	1/4
Specification	`R4`	Structural Decision	1/4*1/4
Specification	`R4`	Behavioral Decision	1/4*1/4
Specification	`R4`	Ban Decision	1/4*1/4
Specification	`R4`	Constraint	1/4*1/4
Specification	`R4`	Design Rule	1/4*1/4
Specification	`R4`	Guideline	1/4*1/4
Specification	`R4`	Organization	1/4*1/5
Specification	`R4`	Process	1/4*1/5
Specification	`R4`	Technology	1/4*1/5
Specification	`R4`	Tool	1/4*1/5

$$D'_{DM} = \frac{\sum_{i=1}^{NoC(S)} D'(< x_S{}^i, m, x_T{}^i >)}{NoC(S)} =(1/3+1/3+1/3+1/3+1/7+1/7+1/7+1/7+1/7+1/7+1+1+1+1+1+1+1)/11=0.835$$

5.4.2 D'_{IM} Prediction of set distribution with indirect mapping from S to T through central model C

Table 2 Direct mapping relationships from LOFAR domain model to Core model with $D'(<x_S,m,x_{C}>)$ *value*

Concept of LOFAR domain model	Relationship/Rule	Concept of Core model	$D'(<x_S,m,x_{C}>)$
Author	`subClassOf`	Stakeholder	1
Artifact	`equivalentClass`	Artifact	1
Artifact Fragment	`subClassOf`	Artifact	1
Concern	`equivalentClass`	Concern	1
Requirement	`subClassOf`	Concern	1
Requirement	`R3`	Decision Topic	1/2
Risk	`subClassOf`	Concern	1
Risk	`R3`	Decision Topic	1/2
Decision Topic	`equivalentClass`	Decision Topic	1
Alternative	`equivalentClass`	Alternative	1
Decision	`equivalentClass`	Decision	1
Decision	`R1`	Architectural Design Decision	1/2
Quick Decision	`subClassOf`	Decision	1
Quick Decision	`R3`	Architectural Design Decision	1/2
Specification	`subClassOf`	Decision	1
Specification	`R3`	Architectural Design Decision	1/2

Table 3 Direct mapping relationships from Core model to Kruchten's ontology with $D'(<x_C,m,x_{T}>)$ *value*

Concept of Core model	Relationship/Rule	Concept of Kruchten's ontology	$D'(<x_C,m,x_{T}>)$
Stakeholder	`noMathingPair`		0
Artifact	`superClassOf`	Design Element	1/3
Artifact	`superClassOf`	Implementation Element	1/3
Concern	`superClassOf`	Requirement	1/7
Concern	`superClassOf`	Risk	1/7
Concern	`superClassOf`	Defect	1/7
Concern	`superClassOf`	Plan	1/7
Concern	`superClassOf`	Cost	1/7
Concern	`superClassOf`	Scope	1/7
Decision Topic	`equivalentClass`	Scope	1
Alternative	`subClassOf`	Design Decision	1
Alternative	`R3`	Existence Decision	1/4
Alternative	`R3`	Property Decision	1/4
Alternative	`R3`	Executive Decision	1/4
Alternative	`R4`	Structural Decision	1/4*1/4
Alternative	`R4`	Behavioral Decision	1/4*1/4
Alternative	`R4`	Ban Decision	1/4*1/4
Alternative	`R4`	Constraint	1/4*1/4
Alternative	`R4`	Design Rule	1/4*1/4
Alternative	`R4`	Guideline	1/4*1/4
Alternative	`R4`	Organization	1/4*1/5
Alternative	`R4`	Process	1/4*1/5
Alternative	`R4`	Technology	1/4*1/5
Alternative	`R4`	Tool	1/4*1/5
Decision	`subClassOf`	Design Decision	1
Decision	`R3`	Existence Decision	1/4
Decision	`R3`	Property Decision	1/4
Decision	`R3`	Executive Decision	1/4
Decision	`R4`	Structural Decision	1/4*1/4
Decision	`R4`	Behavioral Decision	1/4*1/4
Decision	`R4`	Ban Decision	1/4*1/4
Decision	`R4`	Constraint	1/4*1/4
Decision	`R4`	Design Rule	1/4*1/4
Decision	`R4`	Guideline	1/4*1/4

Decision	`R4`	Organization	1/4*1/5
Decision	`R4`	Process	1/4*1/5
Decision	`R4`	Technology	1/4*1/5
Decision	`R4`	Tool	1/4*1/5
Architectural Design Decision	`subClassOf`	Design Decision	1
Architectural Design Decision	`R3`	Existence Decision	1/4
Architectural Design Decision	`R3`	Property Decision	1/4
Architectural Design Decision	`R3`	Executive Decision	1/4
Architectural Design Decision	`R4`	Structural Decision	1/4*1/4
Architectural Design Decision	`R4`	Behavioral Decision	1/4*1/4
Architectural Design Decision	`R4`	Ban Decision	1/4*1/4
Architectural Design Decision	`R4`	Constraint	1/4*1/4
Architectural Design Decision	`R4`	Design Rule	1/4*1/4
Architectural Design Decision	`R4`	Guideline	1/4*1/4
Architectural Design Decision	`R4`	Organization	1/4*1/5
Architectural Design Decision	`R4`	Process	1/4*1/5
Architectural Design Decision	`R4`	Technology	1/4*1/5
Architectural Design Decision	`R4`	Tool	1/4*1/5

Table 4 Indirect mapping relationships from LOFAR domain model to Kruchten's ontology through Core model with combined concept mapping set distribution $D_{\_C}'(<x_S,m,x_T>)$ value

Concept of LOFAR domain model	**Concept of Core model**	$D'(<x_S,m,x_{C>})$	**Concept of Kruchten's ontology**	$D_{\_C}'(<x_S,m,x_T>$	**Relevant mapping**
Author	Stakeholder	1	`noMatchingPair`	0	*r*=0
Artifact	Artifact	1	Design Element	1/3	*r*=1
Artifact	Artifact	1	Implementation Element	1/3	*r*=1
Artifact Fragment	Artifact	1	Design Element	1/3	*r*=1
Artifact Fragment	Artifact	1	Implementation Element	1/3	*r*=1
Concern	Concern	1	Requirement	1/7	*r*=1
Concern	Concern	1	Risk	1/7	*r*=1
Concern	Concern	1	Defect	1/7	*r*=1
Concern	Concern	1	Plan	1/7	*r*=1
Concern	Concern	1	Cost	1/7	*r*=1
Concern	Concern	1	Scope	1/7	*r*=1
Requirement	Concern	1	Requirement	1/7	*r*=1
Requirement	Concern	1	Risk	1/7	*r*=0
Requirement	Concern	1	Defect	1/7	*r*=0
Requirement	Concern	1	Plan	1/7	*r*=0
Requirement	Concern	1	Cost	1/7	*r*=0
Requirement	Concern	1	Scope	1/7	*r*=0
Requirement	Decision Topic	1/2	Scope	1/2*1	*r*=0
Risk	Concern	1	Requirement	1/7	*r*=0
Risk	Concern	1	Risk	1/7	*r*=1
Risk	Concern	1	Defect	1/7	*r*=0
Risk	Concern	1	Plan	1/7	*r*=0
Risk	Concern	1	Cost	1/7	*r*=0
Risk	Concern	1	Scope	1/7	*r*=0
Risk	Decision Topic	1/2	Scope	1/2*1	*r*=0
Decision Topic	Decision Topic	1	Scope	1	*r*=1
Alternative	Alternative	1	Design Decision	1	*r*=1
Alternative	Alternative	1	Existence Decision	1/4	*r*=1
Alternative	Alternative	1	Property Decision	1/4	*r*=1
Alternative	Alternative	1	Executive Decision	1/4	*r*=1
Alternative	Alternative	1	Structural Decision	1/4*1/4	*r*=1
Alternative	Alternative	1	Behavioral Decision	1/4*1/4	*r*=1
Alternative	Alternative	1	Ban Decision	1/4*1/4	*r*=1

Alternative	Alternative	1	Constraint	1/4*1/4	$r=1$
Alternative	Alternative	1	Design Rule	1/4*1/4	$r=1$
Alternative	Alternative	1	Guideline	1/4*1/4	$r=1$
Alternative	Alternative	1	Organization	1/4*1/5	$r=1$
Alternative	Alternative	1	Process	1/4*1/5	$r=1$
Alternative	Alternative	1	Technology	1/4*1/5	$r=1$
Alternative	Alternative	1	Tool	1/4*1/5	$r=1$
Decision	Decision	1	Design Decision	1	$r=1$
Decision	Decision	1	Existence Decision	1/4	$r=1$
Decision	Decision	1	Property Decision	1/4	$r=1$
Decision	Decision	1	Executive Decision	1/4	$r=1$
Decision	Decision	1	Structural Decision	1/4*1/4	$r=1$
Decision	Decision	1	Behavioral Decision	1/4*1/4	$r=1$
Decision	Decision	1	Ban Decision	1/4*1/4	$r=1$
Decision	Decision	1	Constraint	1/4*1/4	$r=1$
Decision	Decision	1	Design Rule	1/4*1/4	$r=1$
Decision	Decision	1	Guideline	1/4*1/4	$r=1$
Decision	Decision	1	Organization	1/4*1/5	$r=1$
Decision	Decision	1	Process	1/4*1/5	$r=1$
Decision	Decision	1	Technology	1/4*1/5	$r=1$
Decision	Decision	1	Tool	1/4*1/5	$r=1$
Decision	Architectural Design Decision	1/2	Design Decision	1/2*1	$r=1$
Decision	Architectural Design Decision	1/2	Existence Decision	1/2*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Property Decision	1/2*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Executive Decision	1/2*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Structural Decision	1/2*1/4*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Behavioral Decision	1/2*1/4*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Ban Decision	1/2*1/4*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Constraint	1/2*1/4*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Design Rule	1/2*1/4*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Guideline	1/2*1/4*1/4	$r=1$
Decision	Architectural Design Decision	1/2	Organization	1/2*1/4*1/5	$r=1$
Decision	Architectural Design Decision	1/2	Process	1/2*1/4*1/5	$r=1$
Decision	Architectural Design Decision	1/2	Technology	1/2*1/4*1/5	$r=1$
Decision	Architectural Design Decision	1/2	Tool	1/2*1/4*1/5	$r=1$
Quick Decision	Decision	1	Design Decision	1	$r=1$
Quick Decision	Decision	1	Existence Decision	1/4	$r=1$
Quick Decision	Decision	1	Property Decision	1/4	$r=1$
Quick Decision	Decision	1	Executive Decision	1/4	$r=1$
Quick Decision	Decision	1	Structural Decision	1/4*1/4	$r=1$
Quick Decision	Decision	1	Behavioral Decision	1/4*1/4	$r=1$
Quick Decision	Decision	1	Ban Decision	1/4*1/4	$r=1$
Quick Decision	Decision	1	Constraint	1/4*1/4	$r=1$
Quick Decision	Decision	1	Design Rule	1/4*1/4	$r=1$

Quick Decision	Decision	1	Guideline	1/4*1/4	r=1
Quick Decision	Decision	1	Organization	1/4*1/5	r=1
Quick Decision	Decision	1	Process	1/4*1/5	r=1
Quick Decision	Decision	1	Technology	1/4*1/5	r=1
Quick Decision	Decision	1	Tool	1/4*1/5	r=1
Quick Decision	Architectural Design Decision	1/2	Design Decision	1/2*1	r=1
Quick Decision	Architectural Design Decision	1/2	Existence Decision	1/2*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Property Decision	1/2*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Executive Decision	1/2*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Structural Decision	1/2*1/4*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Behavioral Decision	1/2*1/4*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Ban Decision	1/2*1/4*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Constraint	1/2*1/4*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Design Rule	1/2*1/4*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Guideline	1/2*1/4*1/4	r=1
Quick Decision	Architectural Design Decision	1/2	Organization	1/2*1/4*1/5	r=1
Quick Decision	Architectural Design Decision	1/2	Process	1/2*1/4*1/5	r=1
Quick Decision	Architectural Design Decision	1/2	Technology	1/2*1/4*1/5	r=1
Quick Decision	Architectural Design Decision	1/2	Tool	1/2*1/4*1/5	r=1
Specification	Decision	1	Design Decision	1	r=1
Specification	Decision	1	Existence Decision	1/4	r=1
Specification	Decision	1	Property Decision	1/4	r=1
Specification	Decision	1	Executive Decision	1/4	r=1
Specification	Decision	1	Structural Decision	1/4*1/4	r=1
Specification	Decision	1	Behavioral Decision	1/4*1/4	r=1
Specification	Decision	1	Ban Decision	1/4*1/4	r=1
Specification	Decision	1	Constraint	1/4*1/4	r=1
Specification	Decision	1	Design Rule	1/4*1/4	r=1
Specification	Decision	1	Guideline	1/4*1/4	r=1
Specification	Decision	1	Organization	1/4*1/5	r=1
Specification	Decision	1	Process	1/4*1/5	r=1
Specification	Decision	1	Technology	1/4*1/5	r=1
Specification	Decision	1	Tool	1/4*1/5	r=1
Specification	Architectural Design Decision	1/2	Design Decision	1/2*1	r=1
Specification	Architectural Design Decision	1/2	Existence Decision	1/2*1/4	r=1
Specification	Architectural Design Decision	1/2	Property Decision	1/2*1/4	r=1
Specification	Architectural Design Decision	1/2	Executive Decision	1/2*1/4	r=1
Specification	Architectural Design Decision	1/2	Structural Decision	1/2*1/4*1/4	r=1
Specification	Architectural	1/2	Behavioral Decision	1/2*1/4*1/4	r=1

	Design Decision				
Specification	Architectural Design Decision	1/2	Ban Decision	1/2*1/4*1/4	$r=1$
Specification	Architectural Design Decision	1/2	Constraint	1/2*1/4*1/4	$r=1$
Specification	Architectural Design Decision	1/2	Design Rule	1/2*1/4*1/4	$r=1$
Specification	Architectural Design Decision	1/2	Guideline	1/2*1/4*1/4	$r=1$
Specification	Architectural Design Decision	1/2	Organization	1/2*1/4*1/5	$r=1$
Specification	Architectural Design Decision	1/2	Process	1/2*1/4*1/5	$r=1$
Specification	Architectural Design Decision	1/2	Technology	1/2*1/4*1/5	$r=1$
Specification	Architectural Design Decision	1/2	Tool	1/2*1/4*1/5	$r=1$

$$D'_{IM} = \frac{\sum_{i=1}^{NoC(S)} D_{\_C}'(<x_S^{\ i}, m, x_{\mathrm{T}}^{\ i}>)}{NoC(S)} = (1/3+1/3+1/3+1/3+1/7+1/7+1/7+1/7+1/7+1/7+1+1+1+1+1+1+1)/11=0.835;$$

5.4.3 D'$_{DM\cap IM}$ Prediction of set distribution of intersection of sets DM *and* IM

$$D'_{DM\cap IM} = \frac{\sum_{i=1}^{NoC(S)} D_{R\_C}'(<x_S^{\ i}, m, x_{\mathrm{T}}^{\ i}>)}{NoC(S)} = (1/3+1/3+1/3+1/3+1/7+1/7+1/7+1/7+1/7+1/7+1/7+1/7+1+1+1+1+1)/11=0.680;$$

6 Future Work

We outline the following points as future work:

- More AK models should be covered and AK repositories should be included for the validation of mapping quality prediction models.
- Tool support for AK model mapping and the quality and cost perdition calculation needs to implemented in order to automate and provide flexibility to these tasks.
- The relationships [9] between AK instances are lost in the currently proposed AK sharing scenarios, which result in traceability problems. For example, a relationship exists between AK instances of concept "*Alternative*" and concept "*Decision Topic*" in that some "*Alternative*" is proposed for some "*Decision Topic*" in a software architecture design. A solution that retains the relationships between AK instances for AK sharing needs to be investigated.

7 Reference

[1] L. Bass, P. Clements, and R. Kazman. Software architecture in practice 2nd ed. Addison Wesley, 2003.

[2] T. R. Gruber. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993.

[3] Remco C. de Boer, Rik Farenhorst, Patricia Lago, Hans van Vliet, Viktor Clerc, and A. Jansen, Architectural Knowledge: Getting to the Core, in QoSA (3rd International Conference on the Quality of Software Architectures), 2007.

[4] IEEE. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Standard 1471-2000, IEEE, 2000.

[5] Object Management Group. Software Process Engineering Metamodel Specification. Technical Report formal/05-01-06, Object Management Group, January 2005.

[6] Antony Tang, Muhammad Ali Babar, Ian Gorton, and J. Han, A survey of architecture design rationale, Journal of Systems and Software, vol. 79, pp. 1792-1804, 2006.

[7] Object Management Group. Ontology Definition Metamodel Specification. Technical Report formal/06-05-01, Object Management Group, May 2006.

[8] Antony Tang, Yan Jin, and Jun Han, A Rationale-based Architecture Model for Design Traceability and Reasoning, Journal of Systems and Software, vol. 80, pp. 918-934, 2007.

[9] P. Kruchten, An ontology of architectural design decisions in software intensive systems, SVM (2nd Groningen Workshop on Software Variability Management), p. 54-61, 2004.

[10] Tyree J and Akerman A, Architecture Decisions: Demystifying Architecture, Software, IEEE, vol. 22, pp. 19-27, 2005.

[11] A. Tang, Y. Jin, and J. Han. A Rationale-based Architecture Model for Design Traceability and Reasoning, The Journal of Systems and Software, Elsevier, 80(6):918-934, 2007.

[12] R. Capilla, F. Nava, S. Pérez, and J.C. Due?as. A Web-based Tool for Managing Architectural Design Decisions. in Proceedings the 1st Workshop on Sharing and Reusing Architectural Knowledge, ACM digital library, 2006.

[13] M. Ali-Babar, I. Gorton, and B. Kitchenham. A Framework for Supporting Architecture Knowledge and Rationale Management. in Dutoit, A.H., McCall, R., Mistrik, I. and Paech, B. eds. Rationale Management in Software Engineering, Springer-Verlag, 237-254, 2006.

[14] A. Jansen, J.S. van der Ven, P. Avgeriou, and D.K. Hammer. Tool support for Architectural Decisions. in Proceedings 7th IEEE/IFIP Working Conference on Software Architecture, 2007.

[15] Peng Liang, Anton Jansen and Paris Avgerio. Refinement to Griffin Core Model and Model Mapping for Architectural Knowledge Sharing. Technical Report RUG-SEARCH-07-L01, University of Groningen, August 2007.

[16] W3C, RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/, 10 February 2004.

[17] W3C, OWL Web Ontology Language Reference, http://www.w3.org/TR/2004/REC-owl-ref-20040210/, 10 February 2004.