Selecting a High-Quality Central Model for Sharing Architectural Knowledge

Peng Liang, Anton Jansen, and Paris Avgeriou
Department of Mathematics and Computing Science
University of Groningen
P.O. Box 407, 9700 AK Groningen, The Netherlands
{liangp,anton,paris}@cs.rug.nl

Abstract

In the field of software architecture, there has been a paradigm shift from describing the outcome of architecting process to documenting Architectural Knowledge (AK), such as design decisions and rationale. To this end, a series of domain models have been proposed for defining the concepts and their relationships in the field of AK. To a large extent, the merit of this new paradigm is derived by sharing and reusing AK across organizations, especially in geographically distributed contexts. However, the employment of different AK domain models by different parties makes effective AK sharing challenging, as it needs to be mapped either from one domain model to another directly, or indirectly through a central model for simplicity when the number of AK models increases. The indirect mapping approach has proved to be a cost-effective way by sacrificing acceptable sharing quality compared with direct mapping approach. However, there exist no criteria for the selection of a high quality central model besides the intuitive judgment by domain experts. In this paper, we propose to tackle this issue by using the concept of semantic distance between AK models, which is calculated using rules based on the concept mapping relationships between the models. A high quality central model is therefore the one with the shortest semantic distance to all potential AK models.

1 Introduction

Software architecture plays an important role in managing the complex interactions and dependencies among stakeholders, and acts as a central artifact for the communication in the whole software life cycle [5]. Existing notational and documental approaches [10] to software architecture primarily focus on the outcome of architecting process and fail to capture the design decisions that resulted in the architecture as well as the organizational, process and business rationale underlying design decisions, result-ing in high maintenance cost, design erosions and evolutionary risks [6]. The software engineering community, both in industry and academia, is therefore gradually acknowledging capturing and codifying explicit architectural knowledge (AK) as a particularly critical task [21][2][3].

AK is generally defined as the integrated representation of the software architecture of a software-intensive system (or a family of systems), the architectural design decisions, and the external context/environment [2]. However, since the focus on AK is relatively new, there is no commonly accepted definition of what AK entails [22], and different AK practitioners may address AK in different perspectives, including generalized, organizational, view-based, domain-specific and project-specific AK etc. Currently, various specific AK domain models have been proposed by industrial organizations (e.g. Knowledge Architect (KA) [17] AK model from LOFAR [1] project) and AK domain experts (e.g. Kruchten's ontology [20], Tyree's template [27], AREL [26], PAKME [1], ADDSS [9] and Archium [18] AK models).

With the increasing trend of distributed software development, e.g. Global Software Development (GSD) and outsourcing development, sharing and reusing AK across organizations becomes a critical factor for project success [16]. Sharing AK is unavoidable for the effective communication between distributed teams who are responsible for different software development activities, e.g. requirement analysis, architecture design, and detailed design, etc. It is also an important part of all architecting activities like modifying past design decisions, or performing architecture reviews and trading off quality attribute requirements.

The problem of bridging the gaps between the different AK models adopted by various organizations need to be solved in order to achieve effective AK sharing [22]. Two distinct approaches can be employed: a direct map-

---

[1]LOFAR is the abbreviation of Low Frequency Array project undertaken by Astron, the Dutch Astronomy Institute, which is involved in the development of large software-intensive systems used for astronomy research.

ping or an indirect mapping approach. Suppose that n AK repositories based on n different AK models are involved in AK sharing. With a direct mapping approach, all AK models are directly mapped onto each other, while with an indirect mapping approach, all models are mapped onto one central model, and the central model acts as a mediator between different models. The indirect mapping approach has proved to be more cost-effective when the number n increases, while still achieving an acceptable sharing quality level [24].

However, when indirect mapping approach is adopted, the question becomes what a high quality central model is for AK sharing and how to get it? The former question is addressed in this paper by the introduction of semantic distance concept between AK models to quantify the AK sharing quality using certain central model. For the latter question, we prefer to have a bottom-up approach to get the optimal central model based on various existing AK models. The arguments for a bottom-up approach and three ways to get the high quality central model are described in section 3.

The rest of this paper is organized as follows. In section 2, related work on AK sharing based on conceptual models is reviewed and discussed. The problem statement of our research is specified in section 3. In section 4, the concept of semantic distance between AK models in the context of knowledge sharing is introduced with a quantified definition. The calculation rules for predicting the metrics of semantic distance are presented in section 5. A concrete case study including five AK models is presented in section 6. The result of this case study is discussed and analyzed in section 7. The paper concludes with future work in section 8.

2 Related Work

Many practitioners and researchers from the knowledge engineering area argue that ontology is a key technology for knowledge representation, management, reasoning and also sharing activities [25]. An ontology can be in various formats varying from categories to formalized concept specifications, including domain models. Due to the existence of multiple ontologies in certain domains (e.g. AK), reconciliation of heterogenous ontologies is a general request to realize a knowledge sharing vision. However, it has been admitted that an integrated ontology complying with all heterogenous ontologies is hard to maintain and evolve due to the complexity, and is sometimes even impossible when concepts conflict with each other [28]. In another way, ontology mapping provides a common layer from which different ontologies could be accessed and therefore could enable knowledge sharing in semantically sound manners [19]. Ehrig and Euzenat have researched on the quality evaluation of the ontology mapping results by introducing two criteria:

the precision and recall rate [14][13][15], which originate from the Information Retrieval (IR) theory [11]. However, their work focuses on the quality evaluation of the mappings between ontologies in the conceptual level without considering the effect to the knowledge instance level. To the best of our knowledge, there is no research work on predicting the mapping/sharing quality of knowledge instances.

De Boer et al. investigated the issues of AK sharing using an indirect mapping approach, and claim that a newly-constructed AK core model (i.e. central model) can cover all the concepts coming from a series of AK models by concept similarity analysis [12]. The central model acts as a mediator for domain model mapping between different AK models. Their work primarily focuses on providing a conceptual base for AK sharing through conceptual mapping. However, the issue about the quality of this central model with respect to AK sharing has not been addressed.

3 Problem Statement

Recent work on getting central model for AK sharing takes a top-down approach. Firstly, AK domain experts proposes a candidate central model, and then evaluate/refine/justify this candidate central model through concept analysis, which is unqualified in two aspects: (1) there are no general criteria for the evaluation of the candidate central model; (2) the central model proposed by domain experts could be biased due to the personal interest, background and partial understanding on various AK models.

We addressed part of issue (1) by reasoning about the AK sharing quality and cost using both direct and indirect mapping approaches in [24]. In order to avoid the cumbersome mapping work in the AK instance level, a prediction model for the sharing quality and cost was proposed based on the mapping relationships in the model level with assumptions. Validation results show that an indirect mapping approach is more cost-effective than a direct mapping approach by sacrificing acceptable sharing quality when the number of AK models involved increases.

The remaining problems are: there is no comparison in sharing quality between the employed central model [12] and other candidate central models (still issue 1), and the assumptions for the prediction model is too optimistic to be real in practice (more specific issue inside issue 1).

To address issue (2), we take a bottom-up approach: getting the optimal central model based on existing AK models. There are three ways to get the central model for AK sharing: (i) Select a central model from the AK domain models involved with AK sharing; (ii) Select a central model from other existing general AK models; (iii) Construct iteratively a new central model.

All these selection and construction ways need a general criterion to evaluate the sharing quality of the se-

lected/constructed central models (issue 1). We introduce the concept of semantic distance between AK models to tackle this issue.

4 Semantic Distance Definition

To address the quality of candidate central models, a general criterion with quantified metrics should be defined. We propose that the concept of semantic distance, originated from computational linguistics (e.g. Natural Language Processing) techniques [8], is applicable in the current problem context. A general introduction on semantic distance in computational linguistics is presented in section 4.1, followed by a detailed definition of semantic distance in the AK domain in section 4.2.

4.1 Semantic Distance in Computational Linguistics

In computational linguistics, the concept of semantic relatedness involves two lexemes in a lexical resource. Its inverse is the *semantic distance*, which formalizes and quantifies the intuitive notion of similarity and dissimilarity between two lexically expressed concepts. Budanitsky presented an extensive survey and classification of measures of semantic distance in [7]. A domain model is composed of lexical concepts and the relationships between them. Thus the similarity and dissimilarity between models can be represented by the aggregation of the similarity and dissimilarity between their compositional concepts, i.e. the semantic distance between concepts (see Fig 1). Similarly to this idea, we try to define the semantic distance in the AK domain as the quantification of similarity and dissimilarity between AK models aimed at AK sharing.
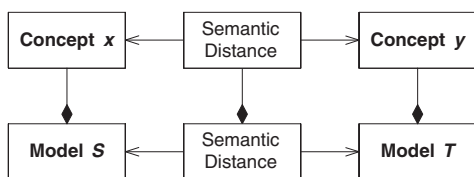
Figure 1. Semantic distance between models as the aggregation of semantic distance between concepts.

4.2 Semantic Distance in AK domain

The concept mapping between AK models provide the conceptual base for AK sharing. The semantic distance between two models depends on two parts as mentioned as follows:

1. Semantic difference in concept mapping: there are different mapping relationships, e.g. `equivalentClass`, `superClassOf`, and `partOf` etc. The semantic distance between two concepts can be different with different mapping relationships. e.g. the semantic distance between two concepts with an `equivalentClass` mapping relationship is shorter (more similar) than those with `superClassOf` mapping relationship.

2. Model coverage difference: due to the unavoidable heterogeneity between different models, the model mapping can not enforce that all the concepts can find their counterparts in other models i.e. the `noMatchingPair` mapping relationship, which increases the semantic distance (more dissimilar) between models.

4.3 Quantified Metrics

The AK sharing activity can be viewed as an information retrieval task [24] and the sharing quality can be quantified in terms of precision, recall, and the F-measure [4]: the metrics for quantifying information retrieval quality. As aforementioned, the semantic distance in the AK domain is concerned with two parts: model coverage difference due to the `noMatchingPair` mapping relationship, and the semantic difference in concept mapping due to other mapping relationships. Both of them can be uniformly measured by the precision and recall rate. The following instance classification scenarios are used to retrieve the data set for the calculation of precision and recall.

An AK repository is composed of AK instances belonging to certain concept types represented in the corresponding AK model. Fig 2 illustrates AK sharing from the perspective of instance classification based on concept mappings. The two bigger circles x and y represent two concepts in different AK models, the small dots inside each concept circle represent the AK instances belonging to them, and the directed arrow between the small dots are instance classifications. If concept x (e.g. **Human**[2]) is `superClassOf` y (e.g. **Man**), three instance classification scenarios may exist:

- an instance of concept x is classified correctly as an instance of concept y, e.g. instance a (*John* is a **Human** and he is also a **Man**);

- an instance of concept x is classified as instance of concept y, but it is not correct mapping, e.g. instance b (*Tom* is a **Human**, but he is not a **Man**, he is a **Boy**).

---

[2]For easier understanding, we take common concepts as an example instead of the specific domain concepts from AK models. A partial concept mapping example between two AK models is shown in Table 2.

This instance classification is possible in practice by an instance classifier since concept **Man** and **Boy** are similar;

- an instance of concept x can not be classified as an instance of concept y, e.g. instance c (*Mary* is a **Human** but she is not a **Man**, she is a **Woman**).
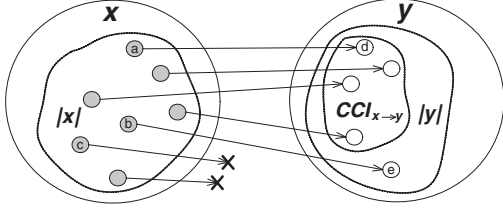
Figure 2. Instance classification scenarios for AK sharing.

Based on the instance classification scenarios, three instance sets can be retrieved (see Fig 2) originating from the IR theory for the calculation of precision and recall:

- $|x|$: all instances to be classified in concept x regardless whether they are classifiable or not, e.g. $a, b, c \in |x|$. *Relevant data* in IR theory;

- $|y|$: all instances classified to concept y regardless whether they are correctly classified or not, e.g. $d, e \in |y|$. *Retrieved data* in IR theory;

- $CCI_{x \to y}$: all correctly classified instances from concept x to y, e.g. $d \in CCI_{x \to y}$. *Relevant retrieved data* in IR theory.

Then the precision (P) and recall (R) rate based on concept mapping from x to y $(x \to y)$ can be defined as follows:

$$P_{x \to y} = \frac{relevant\ retrieved\ data}{retrieved\ data} = \frac{CCI_{x \to y}}{|y|} \quad (1)$$

$$R_{x \to y} = \frac{relevant\ retrieved\ data}{relevant\ data} = \frac{CCI_{x \to y}}{|x|} \quad (2)$$

As proposed in section 4.1, the precision (MP) and recall (MR) rate of the model mapping from S to T can be calculated based on the aggregation of precision and recall for the individual concept mappings. Some symbols for the calculation of MP and MR are defined: x_i denotes a concept of model S, y_i denotes a set of concepts of model T due to multiple mapping relationships from x_i to y_i, W_{x_i} denotes the weight of concept x_i in model S (i.e. the percentage of amount of instances in concept x_i denoted by $|x_i|$ in relation to the whole amount of instances in model S denoted by $|S|$), and $NoC(S)$ is a function to get the number

of concepts in model S. The formulas for the calculation of MP and MR are defined as follows:

$$W_{x_i} = \frac{|x_i|}{|S|} \quad (3)$$

$$MP_{S \to T} = \sum_{i=1}^{NoC(S)} (P_{x_i \to y_i} \times W_{x_i}) \ (x_i \in S, y_i \subset T) \quad (4)$$

$$MR_{S \to T} = \sum_{i=1}^{NoC(S)} (R_{x_i \to y_i} \times W_{x_i}) \ (x_i \in S, y_i \subset T) \quad (5)$$

F-measure (F), the combination of precision and recall in IR theory, and its inverse semantic distance (SD) are defined as follows:

$$F_{S \to T} = \frac{2 \times MP_{S \to T} \times MR_{S \to T}}{MP_{S \to T} + MR_{S \to T}} \quad (6)$$

$$SD_{S \to T} = \frac{1}{F_{S \to T}} \quad (7)$$

We explain the meaning of these formula results (P, R, MP, MR, F, SD) in the context of AK sharing (in IR theory, $P, R, MP, MR, F \in [0, 1]$, then $SD \in [1, \infty]$):

- $P_{x \to y}$ is the percentage of correctly classified instances in concept x to all the classified instances in concept y, and $R_{x \to y}$ is the percentage of correctly classified instances in concept x to all the instances of concept x;

- $MP_{S \to T}$ is the percentage of correctly classified instances in model S to all the classified instances in model T, and $MR_{S \to T}$ is the percentage of correctly classified instances in model S to all the instances of model S;

- $F_{S \to T}$ is a combined criterion to quantify the sharing performance: the greater the F value is, the more similar the two models are, and vice versa;

- $SD_{S \to T}$ is the multiplicative inverse of F, the greater the SD value is, the more dissimilar the two models are, which matches the notion of semantic distance in computational linguistics.

5 Calculation Rules of Precision and Recall

In practice, a domain expert can do the instance classification by following the mapping relationships defined in the concept level, and then calculate the precision and recall for each concept mapping relationship $x_i \to y_i$ using formulas 1 and 2. In order to avoid the cumbersome instance classification work in the AK instance level, the precision and recall can be predicted based on certain prediction model. In our

previous work [24], three prediction models for the calculation of precision and recall were proposed based on the mapping relationships in the conceptual level, and different assumptions were assigned to these prediction models. The SMQPM (simple mapping quality prediction model) was adopted in [24] to predict AK sharing quality and cost. The disadvantage of SMQPM is that it is too optimistic to be real in practice. In this paper, we adopt the prediction model which is more advanced and practical, the RMQPM (random mapping quality prediction model). The assumptions of the three prediction models, including SMQPM, RMQPM and AMQPM (advanced mapping quality prediction model), are specified below.

5.1 Assumptions of Prediction Models

By assigning more practical assumptions, we can come up with a more realistic prediction model, whose prediction of precision and recall results are closer to the real value. Two assumptions are considered: (1) Even distribution of instances (EDI), refers to the assumption that the AK instances are evenly distributed over AK concepts, and (2) Instance classifier (INC), refers to what kind of instance classifier is employed, an intelligent classifier classifies the instances always into the correct concepts, while a random classifier classifies the instances into possible concepts randomly. The three mapping quality prediction models with their assumptions are presented in Table 1 and are compared with the associated assumptions of a real case.

Table 1. Assumptions of three prediction models compared with assumptions of a real case

	EDI	INC
SMQPM	Yes	Intelligent
RMQPM	Yes	Random
AMQPM	No	Random
Real Case	No	Random

Of these three prediction models, the SMQPM has the most optimistic assumptions: the instance classifier always perfectly classify instances into the correct concepts. The RMQPM has the most pessimistic assumptions (but more realistic than SMQPM) in which the instance classifier classifies the instances to possible concepts randomly. The AMQPM has the most realistic assumptions compared with those for SMQPM and RMQPM. In this paper, the RMQPM is adopted for predicting the mapping quality.

With the RMQPM assumption of even distribution of instances (i.e. $W_{x_i} = 1/NoC(S)$), the formulas 4 and 5 can be deduced to the following formulas:

$$MP_{S \to T} = \frac{\sum_{i=1}^{NoC(S)} P_{x_i \to y_i}}{NoC(S)} \quad (x_i \in S, y_i \subset T) \quad (8)$$

$$MR_{S \to T} = \frac{\sum_{i=1}^{NoC(S)} R_{x_i \to y_i}}{NoC(S)} \quad (x_i \in S, y_i \subset T) \quad (9)$$

5.2 Calculation Rules

As mentioned in section 4.2, various concept mapping relationships can be defined between AK models. After a detailed analysis of a series of AK models and concept mappings between them [23], four frequently used concept mapping relationships are selected for effective AK sharing:

- `equivalentClass`, denotes two concepts to be the same;

- `superClassOf`, denotes one concept to be a generalization of another;

- `subClassOf`, denotes one concept to be a specialization of another;

- `noMatchingPair`, denotes that a concept can not find its counterpart in another model.

The calculation rules of RMQPM for the precision (P) and recall (R) are based on above concept mapping relationships. In these rules, the amount of instances belonging to each concept is defined as a constant C, which is feasible due to the assumption of even distribution of instances.

5.2.1 equivalentClass

If concept x is the `equivalentClass` of y, then all the instances of concept x are also the instances of y (see Fig 3). The precision (P) and recall (R) can therefore be calculated as:

$$P_{x\ equivalentClass\ y} = \frac{CCI_{x \to y}}{|y|} = \frac{C}{C} = 1 \quad (10)$$

$$R_{x\ equivalentClass\ y} = \frac{CCI_{x \to y}}{|x|} = \frac{C}{C} = 1 \quad (11)$$

Figure 3. Instance classification with equivalentClass relationship.

5.2.2 superClassOf

The `superClassOf` mapping relationship from concept x to y is more complex. Since one concept x can have multiple `superClassOf` mapping relationships with y. In such a situation, two instance classification approaches can be employed for the random instance classification: a replicating or a splitting approach. RMQPM assumes a random classifier, thus this classifier is unable to recognize the correct concept from multiple candidate concepts (multiple y plus a *dummy* concept representing the concept not covered by y) for an instance. The classifier either classifies all the instances of x to all candidate concepts by *replicating* the instances of x, or classifies part of the instances of x to all candidate concepts by evenly *splitting* the instances of x. Both the replicating and splitting approaches have their specific advantages and disadvantages with respect to the mapping quality. The replicating approach achieves a higher recall since more instances are classified, while the splitting approach achieves a higher precision since less incorrect instances are classified. The precision (P) and recall (R) with `superClassOf` relationship using replicating or splitting can be calculated as follows ($NoS(x)$ is a function to get the number of subclass concepts of x):

- **Replicating Approach**

$$P_{x\ superClassOf\ y} = \frac{CCI_{x \to y}}{|y|} = \frac{\frac{C}{NoS(x)+1} \times NoS(x)}{NoS(x) \times C} \quad (12)$$
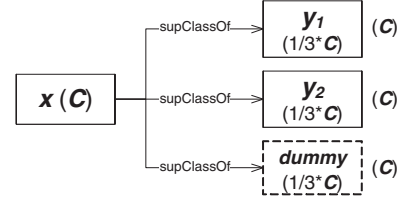
$$R_{x\ superClassOf\ y} = \frac{CCI_{x \to y}}{|x|} = \frac{\frac{C}{NoS(x)+1} \times NoS(x)}{C} \quad (13)$$

An example of an instance classification with two subclass concepts (y_1 and y_2) to x using replicating is presented in Fig 4. The $1/3 \times C$ inside the concept box denotes the amount of correctly classified instances ($CCI_{x \to y}$), and the C outside concept box denotes the amount of classified instances ($|y|$). Note that classified instances to dummy concept are not considered in the calculation due to its irrelevance to any y concepts.
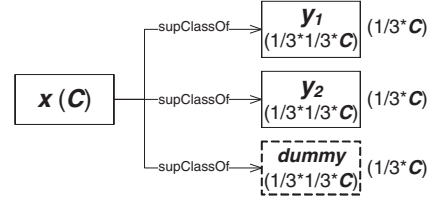
- **Splitting Approach**

$$P_{x\ superClassOf\ y} = \frac{CCI_{x \to y}}{|y|} = \frac{\frac{C}{(NoS(x)+1)^2} \times NoS(x)}{\frac{C}{NoS(x)+1} \times NoS(x)} \quad (14)$$

$$R_{x\ superClassOf\ y} = \frac{CCI_{x \to y}}{|x|} = \frac{\frac{C}{(NoS(x)+1)^2} \times NoS(x)}{C} \quad (15)$$

Figure 4. Instance classification with super-ClassOf relationship using replicating.

Figure 5. Instance classification with super-ClassOf relationship using splitting.

An instance classification example with two subclass concepts (y_1 and y_2) to x using splitting is shown in Fig 5.

When calculating the semantic distance of a concept mapping with the `superClassOf` relationship, the calculation results based on the two approaches are combined to get an average value.

5.2.3 subClassOf

Let concept x be the `subClassOf` y, then all the instances of concept x are also the instances of y (see Fig 6). The precision (P) and recall (R) can be calculated as:

$$P_{x\ subClassOf\ y} = \frac{CCI_{x \to y}}{|y|} = \frac{C}{C} = 1 \quad (16)$$

$$R_{x\ subClassOf\ y} = \frac{CCI_{x \to y}}{|x|} = \frac{C}{C} = 1 \quad (17)$$
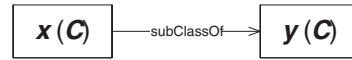
Figure 6. Instance classification with subClassOf relationship.
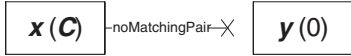
5.2.4 noMatchingPair

If concept x has `noMatchingPair` in the other model, then all the instances of concept x cannot be mapped (see

Table 2. Partial concept mappings from Archium to AREL

Archium	AREL	Relationship
Stakeholder	Supporting information	`subClassOf`
Requirement Category	-	`noMatchingPair`
Requirement	Functional requirements	`superClassOf`
Requirement	Non-functional requirements	`superClassOf`
Architectural Design Decision	Design rationale	`superClassOf`
Trade-off	Tradeoffs	`equivalentClass`
Motivation	Design concerns	`subClassOf`
Pros	Design strengths and weaknesses	`subClassOf`

Fig 7). The precision (P) is not taken into account since there is no instance been classified ($|y| = 0$), and recall (R) can be calculated as:

$$R_{x\ noMatchingPair\ y} = \frac{CCI_{x \to y}}{|x|} = \frac{0}{C} = 0 \qquad (18)$$

Figure 7. Instance classification with no-MatchingPair relationship.

6 Case Study: Five AK models

In this section, the calculation rules defined in section 5 are applied in a concrete case study including five AK models. The input is the concept mapping relationships between these five AK models, and the output is the semantic distance between them. An optimal central model with a high quality will be selected in the end among these five models. Note that the semantic distance as a general criterion for the quantification of sharing quality can also be applied in other ways to get a central model as stated in section 3: one can select a central model from other existing general AK models, or construct a new central model. There options will be further investigated in our future work.

6.1 Five AK models

For the purpose of future validation in the AK instance level, the five AK models selected are all supported by AK management tools. AREL (Architecture Rationale and Element Linkage) [26] is a tool to capture architecture decisions and design rationale which focus on traceability between requirements, design and design rationale. Archium [18] tool integrates a requirement, decision, architecture, and implementation model and aims at fine-grained

traceability between them. ADDSS (Architecture Design Decision Support System) [9] is a web-based tool for storing, managing, and documenting architectural decisions, which is the core concept in ADDSS AK model. PAKME (Process-based Architecture Knowledge Management Environment) [1] is a prototype web-based system to provide knowledge management support for improving architecting activities, with a special AK interest on collaborative features, e.g. contact management, project management, online collaboration tools etc. KA (Knowledge Architect) [17] is a tool suite for capturing, managing, and sharing architectural knowledge, and emphasizes on providing generic AK management function.

6.2 Concept mappings between models

The number of concepts mapping pairs is $\mathcal{O}(n^2 \times m)$ where n is the number of AK models, and m is the average number of concepts per AK model. Due to space limitations, only a partial part of the concept mappings from Archium [18] to AREL [26] are shown in Table 2.

6.3 Calculation Result

With the mapping relationships between all five models, the semantic distance and average semantic distance results can be calculated, as shown in Table 3. Each value in cell represents the semantic distance from the model in the first column to the model in the first row, e.g. 2.5136 is the semantic distance based on the model mapping from AREL to Archium, and 2.8398 is the semantic distance based on the model mapping from AREL to ADDSS. The implication of these two values is that a better sharing quality can be achieved when sharing AREL AK using the Archium model than using the ADDSS model. Among the five models, AREL achieves the shortest average semantic distance: 1.6382, which implies that the best sharing quality can be achieved when sharing AK based on other models using the AREL model. Consequently AREL is the optimal central model for sharing AK in high quality among the five models using an indirect mapping approach.

From↓ To→	AREL	Archium	ADDSS	PAKME	KA
AREL		2.5136	2.8398	2.1707	1.6869
Archium	1.2747		2.4003	1.9396	2.2951
ADDSS	1.6009	1.8194		1.5885	1.4959
PAKME	1.8124	1.9583	1.9583		2.2697
KA	1.8649	1.8366	3.9444	2.3400	
Average SD	**1.6382**	2.0320	2.7587	2.0097	1.9369

7 Discussions and Analysis

In this section, the implications between the semantic distance calculation result and AK model comparison analysis is discussed and analyzed to justify the concept of semantic distance in AK sharing.

- Mapping from Archium to AREL achieves the shortest semantic distance (1.2747), compared to those from Archium to other models. The reason as found in the model comparison analysis is that: both Archium (concepts e.g. Component Entity, Delta, Port, Interface, Connector, Abstract connector) and AREL (concepts e.g. Design outcome, Application model, Technology model, Data model) have an extensive architecture model, which is missing in KA, ADDSS, and PAKME.

- Mapping from KA to ADDSS achieves the longest semantic distance (3.9444) in the table. The reason as found in the model comparison analysis is that: The AK described by these two AK model are in different design levels. KA primarily focuses on the design decision on the quantitative aspect of AK (concepts e.g. Number, Value, System parameter, Analysis function) which is close to the detail design level, while ADDSS is mainly concerned with the design rationale on architecture views (concepts e.g. Architectural views, Traditional views, Decision views) which is close to the architecture design level.

- AREL achieves the shortest average semantic distance: 1.6382, and is therefore selected as the optimal central model with high quality among the five models. The reason as found in the model comparison analysis is the following: we try to look at the concept coverage of each AK model in a characteristic view which has broader scope than individual concept, and categorize all the AK concepts into a set of characteristics as shown in Table 4, and the cell with (+) mark means that the AK model can support the AK of this characteristic (e.g. describing Patterns, Quality model, etc). The number attached with (+) mark denotes the

	AREL	Archium	ADDSS	PAKME	KA
Architecture model	+ (1)	+ (1)			
Indirect requirement	+ (2)	+ (2)			+ (2)
Explicit arch. evolution	+ (3)	+ (3)		+ (3)	+ (3)
Scenario support	+ (2)	+ (2)		+ (2)	
Quantitative analysis				+ (1)	+ (1)
Quality model	+ (2)		+ (2)		+ (2)
Reviewing					+ (0)
Patterns		+ (2)	+ (2)	+ (2)	
Decision tracking			+ (1)		+ (1)
Decision classification			+ (0)		
Risk identification	+ (2)			+ (2)	+ (2)
Number in Common	12	10	5	10	11

number of characteristics in common with other models. Bottom row is the sum of *number in common*, which demonstrates the concept coverage of each AK model. The value of *number in common* is perfectly matched with the semantic distance calculation result in Table 3: the shorter the semantic distance is, the more common parts the model has with other models.

8 Conclusions and Future Work

In this paper, we introduced the concept of semantic distance between AK models to tackle the issue of quantifying the AK sharing quality, in particular to select a high quality central model for an indirect mapping approach. The major contributions of this paper are the following: (1) Introduction and definition of the semantic distance between AK models in the context of AK sharing. The benefit is two-fold: use the semantic distance to quantify the AK sharing quality, and act as a general criterion to quantify the similarity between AK models. (2) Introduction of quantified metrics (precision and recall) and the definition of calculation rules for the semantic distances with different concept mapping relationships.

We outline our future work in several points: (1) Validation of the prediction result against the AK instance level in the practical AK sharing scenarios; (2) Adoptation of more advanced and practical prediction model for the mapping quality prediction, i.e. AMQPM; (3) Inclusion of additional concept mapping relationships (e.g. `partOf`, `compositionOf`, `disjointWith`, etc.) besides the four mapping relationships used in this paper for the purpose of more comprehensive concept mapping and better sharing quality in the end; (4) Tool support, including concept mapping tool and semantic distance visualization tool, to assist the selection of high quality central model; (5) Application of semantic distance in another domain besides AK, e.g. requirement knowledge sharing.

9 Acknowledgements

References

[1] M. Ali-Babar, I. Gorton, and B. Kitchenham. A Framework for Supporting Architecture Knowledge and Rationale Management. *Rationale Management in Software Engineering, A.H. Dutoit, et al., Editors*, pages 237–254, 2006.

[2] P. Avgeriou, P. Kruchten, P. Lago, P. Grisham, and D. Perry. Architectural knowledge and rationale: issues, trends, challenges. *ACM SIGSOFT Software Engineering Notes*, 32(4):41–46, 2007.

[3] P. Avgeriou, P. Lago, and P. Kruchten. Third international workshop on sharing and reusing architectural knowledge (SHARK 2008). *ICSE Companion*, pages 1065–1066, 2008.

[4] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Addison-Wesley Harlow, 1999.

[5] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice Second Edition*. Addison Wesley, 2003.

[6] J. Bosch. Software Architecture: The Next Step. *Proceedings of the First European Workshop on Software Architecture (EWSA)*, pages 194–199, 2004.

[7] A. Budanitsky. Lexical semantic relatedness and its application in natural language processing. Technical Report CSRG390, University of Toronto, 1999,ftp://ftp.cs.toronto.edu/csrg-technical-reports.

[8] A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. *Second meeting of the North American Chapter of the Association for Computational Linguistics*, pages 29–34, 2001.

[9] R. Capilla, F. Nava, S. Pérez, and J. Dueñas. A web-based tool for managing architectural design decisions. *ACM SIGSOFT Software Engineering Notes*, 31(5), 2006.

[10] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley Professional, 2002.

[11] C. Cleverdon. The Cranfield tests on index language devices. *Morgan Kaufmann Multimedia Information And Systems Series*, pages 47–59, 1997.

[12] R. de Boer, R. Farenhorst, P. Lago, H. van Vliet, V. Clerc, and A. Jansen. Architectural Knowledge: Getting to the Core. *Proceedings of the 3rd International Conference on the Quality of Software-Architectures (QoSA)*, pages 197–214, 2007.

[13] M. Ehrig and J. Euzenat. Generalizing precision and recall for evaluating ontology matching. *Proceedings of the 4th International Semantic Web Conference (ISWC) poster session*, pages PID–54, 2005.

[14] M. Ehrig and J. Euzenat. Relaxed precision and recall for ontology matching. *Proceedings of the K-Cap 2005 workshop on Integrating ontology*, pages 25–32, 2005.

[15] J. Euzenat. Semantic precision and recall for ontology alignment evaluation. *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, 2007.

[16] A. Jansen and J. Bosch. Software Architecture as a Set of Architectural Design Decisions. *Proceedings of the 5th IEEE/IFIP Working Conference on Software Architecture (WICSA)*, pages 109–119, 2005.

[17] A. Jansen, T. de Vries, P. Avgeriou, and M. van Veelen. Sharing the Architectural Knowledge of Quantitative Analysis. In *Proceedings of 4th International Conference on the Quality of Software-Architectures (QoSA)*, 2008 (to appear).

[18] A. Jansen, J. van der Ven, P. Avgeriou, and D. K. Hammer. Tool support for Architectural Decisions. *Proceedings of the 6th IEEE/IFIP Working Conference on Software Architecture (WICSA)*, pages 44–53, 2007.

[19] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(01):1–31, 2003.

[20] P. Kruchten. An ontology of architectural design decisions in software intensive systems. *Proceedings of the 2nd Groningen Workshop on Software Variability Management (SVM)*, pages 54–61, 2004.

[21] P. Lago and P. Avgeriou. First workshop on sharing and reusing architectural knowledge. *ACM SIGSOFT Software Engineering Notes*, 31(5):32–36, 2006.

[22] P. Lago, P. Avgeriou, R. Capilla, and P. Kruchten. Wishes and Boundaries for a Software Architecture Knowledge Community. *Proceedings of the 7th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 271–274, 2008.

[23] P. Liang, A. Jansen, and P. Avgeriou. Refinement to griffin core model and model mapping for architectural knowledge sharing. Technical Report RUG-SEARCH-07-L01, University of Groningen, 2007, http://www.cs.rug.nl/~liangp/download/liang2007rgc.pdf.

[24] P. Liang, A. Jansen, and P. Avgeriou. Sharing architecture knowledge through models: quality and cost. *The Knowledge Engineering Review*, 2008 (to appear).

[25] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. Swartout. Enabling Technology for Knowledge Sharing. *AI Magazine*, 12(3):36–56, 1991.

[26] A. Tang, Y. Jin, and J. Han. A rationale-based architecture model for design traceability and reasoning. *The Journal of Systems & Software*, 80(6):918–934, 2007.

[27] J. Tyree, A. Akerman, and C. Financial. Architecture Decisions: Demystifying Architecture. *Software, IEEE*, 22(2):19–27, 2005.

[28] A. Valente, T. Russ, R. MacGregor, and W. Swartout. Building and (re) using an ontology of air campaign planning. *Intelligent Systems and Their Applications, IEEE*, 14(1):27–36, 1999.