Evaluation and Improvement of MoSaIc Reference Models Comparison Approach

Chittaphone CHANSYLILATH, and Simona JENERS

***Abstract*—**The process improvement nowadays is playing an important role in software development. The quality of software product is strongly influenced by the quality of development process applied. In order to improve the quality of software product; the process improvement needs to be closely take a look and developed. Moreover, the impact of people and technologies is rapidly increased as can be seen from the number of increasing of the competitor in software market. Thus, to reach the customers with better, cheaper and faster products, organizations need to have good strategies for dealing with their software development process.

***Keywords*—**Reference models; software process improvement; comparison; meta models; similarity metrics.

I. Introduction

DUE to the complexity of reference models is made their comparison becomes a complicated task. Currently, there are some people are researching on similar topic with reference model comparison approach such as: Ferchichi et al.[3] developed an approach for integrate two reference models CMMI and ISO 9001:2000, Liao et al.[4] presented the common structure of reference model which can make their comparable by designing a software process ontology and the department of computer science , RWTH university, Aachen Germany is also tries to solve the problems of adoption and assessment of reference models. The project is started since January 2010, aiming for developing a model based integration approach of the reference models called MoSaIC (Model-based Selection of Applied Improvement Concepts)[1].

MoSaIc is a comparison approach was developed to compare two or more reference models by their similarity procedures. This can be done in three levels: Firstly, is the comparison of basic elements which are elements of an activity unit knows as input and output artifacts, roles and contexts of reference model.

The accuracy in this level is based on relations between these concepts. Secondly, is the comparison of activity units, this level is based on the aggregation results from previous level and the last level is the comparison and similarity between two or more procedures, one procedure is consists of one or many activity units.

For example: "**Establish and maintain the overall project plan**" is a procedure which consists of two activity units: "**Establish the overall project plan**" and "**Maintain the overall project plan**". This level is based on a list of results from previous levels [1]. According to an evaluation of MoSaIc comparison approach, this approach still needs to improve in order to provide the organization with the good accurate comparison approach. The main drawback that we have seen is that the way of structuring Integration Concept Model (ICM concepts), this approach doesn't has any standard for structuring or grouping the set of concepts; Therefore, all concepts were only structured and checked by expert. It seems to work for small practice, but it becomes really hard for a huge number of practices. In addition, as stated earlier, this approach is based on the similarity of procedures from both reference models. The similarity in each level is based on the relation between those corresponding concepts. But this still hard to find out which concept has relation with other concepts, since we have to take a look many times when we need to add new concept and relation. In conclusion, it becomes the harder task for defining the similarities and compares them afterward.

Chittaphone Chansylilath, The Sirindhorn International Thai-German Graduate School of Engineering (phone: 85620-22234883; e-mail: chittaphone@gmail.com).

Simona Jeners, RWTH Aachen University, Research Group Software Construction Aachen, Germany (email: simona.jeners@rwth-aachen.de)

II. Challenge And Goals

Main goals of paper are to evaluate and improve the existing comparison approach (MoSaIc's comparison approach) aim for developing the systematic calculation tool for similarity and differences between concepts from different reference models. Another goal is to analyze and improve the structure of existing ICM models. This means that all the abstraction of concepts are need to be identified and categorized in proper ways, the relation among concepts and their abstract concepts need to be assigned correctly. As mentioned earlier, lack of a proper structuring of concepts can cause a lot of problems especially in determining the similarities and differences between two concepts. Therefore, the idea of improvement was prompted, the clear working task is determined as the real objective for this research.

III. Related Works

From literature reviews can be concluded that, with some domain ontology, concepts are categorize as aspects or group, depending on specific domain model. They were structured as a hierarchy tree and each tree has its own sub concepts which are disjoint with each other. The relationship between parent or root concept to its sub concepts is clearly defined. In addition, some papers consider the type of concepts such as role concepts and value concepts. For example: Alans

Rector[7][8] grouped concepts as Seft-Standing Entities, Playable Roles and Value types and all combinations of those type concepts can makes new possible sub concepts. Thus, from those mentioned ideas, we applied the idea for categorizing and structuring concepts. With the idea of aspects we have categorized one concept by one or more aspects. For example: "stakeholder" was categorized by department, importance, experience and etc. while "requirements" was categorized by context, type, status and parties. In addition, with the idea of concept type, we classified concepts into attribute concepts which have higher importance weight than non-attribute concepts.

In our approach, each structuring concept has a tree that specialize it and all sub concepts come from integration structure meta model of specific domain. For example, one reference model contains one or many categories and one category contains one or more process areas and so on. While from this paper, Alans Rector structured concepts based on the logical language like for health care service ontology, the Playable Roles concepts must have child concepts like Patient Role, Doctor Roles and etc.

The relationship between concepts applied for all in hierarchies' tree is "is-kind-of" while our approach has "is-Generalization-of" inside category tree and "is-Compossed-of" for relating one abstract concept with other abstract concepts. Our approach considers the weight of each category, which one has more importance and which one has less importance while the paper wasn't covered.

Faceted based approach, all combination between facets are made by the combination of facet values. For example, in book library, a book could be classified as "fiction/UK/19th century" using the three facets "topic", "geography" and "time".

In our approaches, all related concepts are define by corresponding categories and divided into three levels which are high, medium and low level (We defined the weight of abstract concept to attribute concepts as high level, for the weight of abstract concept to either non-attribute or attribute concepts we defined as medium and for the weight of abstract concept to non-attribute concepts as low level). The combination of two concepts can be calculated depending on the weight of the relation between abstract and its child concepts. For example, stakeholder which has attributes key or non-key representing key stakeholder or non-key stakeholder. It is impossible that the combination can be created by attribute itself.

IV. MOSAIC INTEGRATION APPROACH

Reference models have different application areas, different basic elements and they have written by different authors for different purposes. In order to make an accurate reference models comparison, it is necessary to have the good process of bringing those reference models to same structure and terminology. Thus, to success this- Integration Concept Meta-Model(IC Meta-Model) and Integration Structure Meta-Model (IS Meta Model) are needed.

IC Meta Model- Consists of concepts and concept relation which is created for helping reference model understands other reference model by using same language for communication. It is a word or set of words that took from the analyzed reference models. For instance, concept is a combination of words "project schedule". IC Meta-Model is can be atomic or not by its definition; It is atomic, if it is independent from other concepts. For example, some concepts like "project", "activity" or "Management framework" are atomic concepts while others like "maintain project plan" and "establish training needs of the organization" are not atomic.

IS Meta-Model- comprises of basic elements such as outputs, inputs, roles and context artifacts. MoSaIc had defined the syntactical rule [2] for extract all these elements from context of reference model. IS Meta-Model has two packages- Core package contains elements defined in most of the meta-models of reference model which available now. The reference model represents and structured by Categories. A category defines a certain topic that is addressed in one or more Process Areas. A process area addresses a topic to be improved by defining Procedures. The dependencies between procedures can be classified by dependsOn relation. And other package is Concepts package contains elements use to model concept information of RMs .We differentiate between Activities, Artifacts (Inputs and Outputs), Purposes for activities and Roles. These Procedure Concepts are offered to model IRM procedures. Each procedure concept from an IS Meta-Model relates to a concept in the IC Meta-Model [1]. Fig.1 describes in details the IS Meta-Model and IC Meta-Model with their correspond relations.

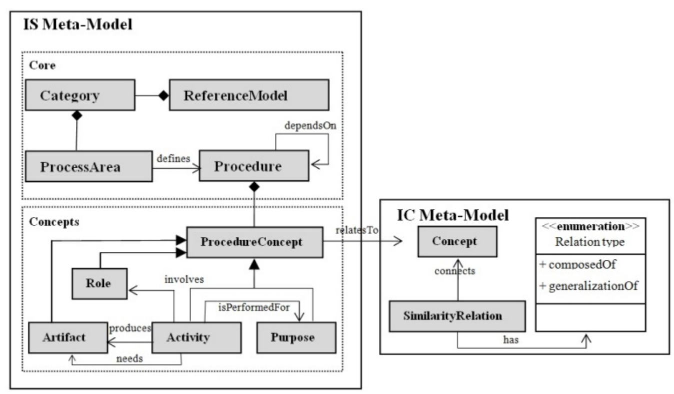

Fig. 1 MoSaIc Integration Meta-Model and Concept Meta-Model

MoSaIC comparison approach can be done on four levels - The similarity of procedures, this comparison is based on a list with the results of previous level, followed by the similarity of activity units which is based on the aggregation results from previous level. Then is similarity of basic elements of the same type, based on relations between these concepts and the last comparison is similarity of concepts in ICM concepts.

Fig. 2 describes the relation of each level in detail. This figure shows that a reference model comprises of many procedures which contain one or many activity units and one activity unit is composed of one or many concepts. For example, CMMI, "Establish and maintain supplier agreements" by mean of using the modeling rule[3] we can split its activity units into two activity units-Establish supplier agreement with one output-supplier agreement and Maintain supplier agreements with one input supplier agreement and one output supplier agreement.

Fig. 2 MoSaIc Comparison approach

V. Comparison Approach

This approach is based on 3 main processes. The first process is the structuring of concepts. In this process, we define some rules (in 4.1) which can be used for categorizing and positioning the concepts. The Second process is mapping ISM concept with its corresponding ICM concepts using 1:m mapping relation. For instance, one concept in ISM can be mapped with one or more ICM concepts. The third process is the process of computing the similarity of concepts (it is described in detail in next section)

A. Structuring Concepts

As mentioned before, the existing ICM structure doesn't scientific supports our comparison approach. After some evaluation we have some ideas for improving and re-structuring the concepts. At first, we categorized the ICM abstract concepts by the various type of aspects then we related them with their corresponding sub-concept by the "**ComposedOf**" relation. For each abstract concept, we categorized its sub-concepts by their corresponding category and eventually we related them with "**GeneralizationOf**" relation. For example: we categorized project stakeholder with by different aspects such as Importance: key, context: SW: programmer, experience: expert and so on. In meantime, all concepts are connecting with each other by Compose of and Generalization of relations. Below are the rules for structuring ICM concepts:

- ***R1:*** ICM abstract concepts are connected by "**ComposedOf**" relation (e.g. "stakeholder involvement plan" is compoasedOf "stakeholder" and "activities")
- ***R2:*** For each ICM abstract concept there is a tree of specializations (with ICM sub-concepts that specializes the ICM abstract concept- "**GeneralizationOf**" relation. In the specialization tree there is no relation between ICM sub concepts.
  - o The ICM sub-concepts are categorized (e.g. Category "Importance" categorizes the ICM sub-concepts "key stakeholder" and "non-key stakeholder").The value "key", "non-key" will be saved in the concepts.
  - o The ICM sub-concepts can also have their own ICM sub-concepts (e.g. "software stakeholder" has sub-concepts "programmer" and "tester"). These sub-concepts only inherit the category of their parent sub-concepts.
  - o ISM concepts can be connected to one or more ICM sub-concepts (e.g. ISM "software key stakeholder " is connected to two ICM sub-concepts "software stakeholder" and "key stakeholder")
- ***R3:*** In specialization tree of ICM model, there is no relation between sub-concept in one category and another category.

B. Mapping ISM with ICM Concepts

As I mentioned earlier, reference model (RM) itself is large, complex and not transparent. Thus, it hard to understand and it brought the difficulty to the organization to select the most suitable RM. For this reason, the ISM is presented as the process of analysis and classifying basic elements of RM to be mapped with ICM. For instance, ISM Modeling of SPICE:"ACQ.13.BP6: Define responsibilities and goals"} can be modeled as following: the meaning of this procedure is to define the responsibilities and goals of the team members}, with this description can be classified into two activities; the first activity is to define responsibilities of team members" which can be produced an output responsibilities of the team members". The last activity is define goals of the team members" which produces an output "goals of the team members" and the one in charge for these two activities is represents as a role "team members".

Fig.3 describes in details the ISM Modeling.

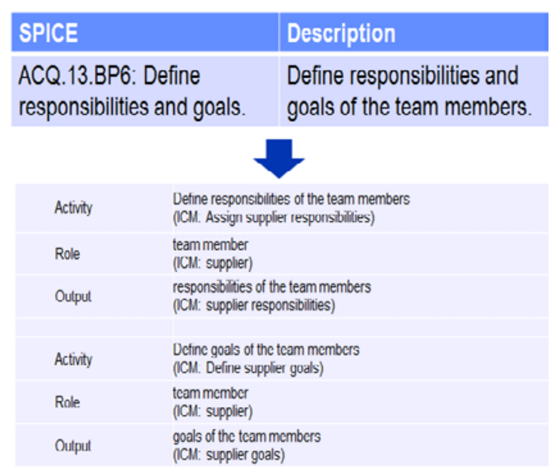

Fig. 3 ISM Modeling

The process of mapping firstly is identifying the ICM corresponding concepts. For example: defined ICM concepts which are match with specific ISM concept (e.g. "key software stakeholder" can be mapped with two corresponding concepts "software stakeholder" and "key stakeholder". This mapping is occurs when the corresponding concept is available otherwise the new concept is needs to be created.

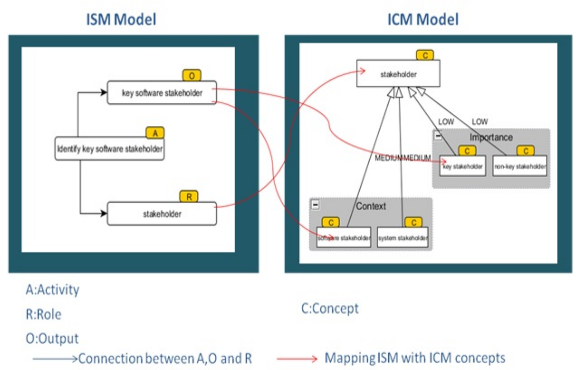

Fig.4 ISM and ICM Mapping

Mapping algorithm

- Mapping ISM concept with ICM concepts(if the concept already exists)
  - o Identification of the corresponding ICM-concepts, including:
    - Finding the abstraction of ISM concept (e.g. "key software stakeholder" has an abstraction and "programmer" have same abstraction concept "stakeholder")
    - Filtering the ICM abstract concepts by their process areas and goals(this means that all available ICM abstract concepts are defined accordingly with specific domain)
    - Identifying the abstract concept by searching in its specialization tree and then look for the best category which can be mapped with ISM concept.
  - o Connecting the ISM concept to the corresponding ICM concepts
    - See rule R2.b
- Adding new concepts(if these concept haven't existed)

An ISM concept can be mapped with one or more concepts in ICM. Thus, all new concepts will be generated according to existing ISM concepts. For example: ***"expert SW key stakeholder"*** in ISM will be mapped with three difference concepts in ICM. Therefore, new concepts can be added according to the steps below:

- o Understand new concepts and cate***"expert, key and SW stakeholder"***gorize them with aspects in ICM model such as ***"key programmer"*** is a combination between two difference concepts ***"stakeholder: aspect: Importance: key"*** and ***"stakeholder: aspect: Context: SW: programmer"***.
- o Checking for the corresponding aspect and concept in ICM
  - If aspect was found -> identify the corresponding concepts
    - If concept was found-> connect ISM concept with ICM concept in either 1:1 or 1:M mapping
    - If concept was not found -> add new concept and connect ISM concept with ICM concept in either 1:1 or 1: M mapping.
  - If aspect was not found -> add new aspect and corresponding concept.

C. *Similarity Algorithms*

The comparison approach considers the similarity between two concepts according to their semantic relations ("generalizationOf" and "composedOf").We differentiated the similarity values between concepts in two different relations as following:

Similarity in Specialization

In the reference model content, the similarity between two concepts depends on their related information such as the position where they are located, what type of category that they are belong to and the relations among them. Therefore, we try to cover all cases that concepts are supposed to have. *Firstly*, we compute the similarity between two concepts in same category. *Secondly*, we consider the similarity between two concepts along with the specialization hierarchy and *finally*, we compute the similarity for those concepts that located in different category.

- For all cases, the similarity value between two ICM concepts in specialization tree can be calculated as following:

$$\mathbf{Sim}(\mathbf{ce1}, \mathbf{ce2}) = \frac{\mathbf{2depth}(\mathbf{LCA}(\mathbf{ce1}, \mathbf{ce2}))}{\mathbf{depth}(\mathbf{ce1}) + \boldsymbol{depth}(\boldsymbol{ce2})} \qquad (4.1)$$

- o LCA (ce1, ce2): Lowest Common Ancestor; entity of maximum depth that is ancestor of ce1 and ce2.
- o Depth (ce1): number of edges from LCA (ce1, ce2) to ce1.

The Similarity between concepts in same category

In this case, we consider only a small part of concept abstraction. As mentioned earlier, an abstract concept contains one or more sub concepts and those sub concepts are positioned as a small hierarchy tree. The similarity in this case is the computation for similarity between members in same parent (category).Figure 5. Represents the hierarchy tree of one abstract concept "**A**" with its sub concepts "**B**" and "**D**" which are sometimes share some common features or disjoint with each other.

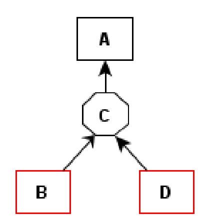

Fig. 5 Hierarchy tree for two concepts in same category

Therefore, we also consider the relation between them (sub concept (**B** and **D**) and the similarity between these two concepts is the multiplication of the similarity between both concepts by their relation (we have classified the relation within one category into three levels High, Medium and Low) and the actual similarity in specialization tree (as defined in previous section).Thus, the experts have defined the default similarity values as following assumptions.

- S1 (*High level*): For concepts which are related or not disjoint. We consider the similarity between them 100% ***S1=1***
- S2 *(Medium level):* For some concepts which are sometimes can be related with some concepts, the similarity will be calculated as 50% ***S2=0.5***
- S3 *(Low level):* For concepts which are not related or disjoint with other concepts in same category, the similarity will be calculated as 25% ***S3=0.25***

Based on these assumptions above, we defined the formula for compute the similarity between ce1 and ce2 as follows:

$$Sim(ce1, ce2) = S(ce1, ce2) X Sim(ce1, ce2) \quad (4.2)$$

- S(ce1,ce2) is the similarity between ce1 and ce2, and
- Sim(ce1,ce2) is an actual similarity between ce1 and ce2.

Example: An example below is showing how our formula is works

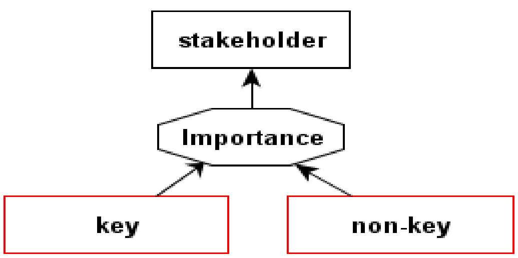

Fig. 6 The hierarchy tree of stakeholder in various type categories

From Fig. 6 we have known that, both concepts are disjoint then the category similarity is = 25% thus, from formula 4.2 we have:

$$Sim(key\ stakeholder, non-key\ stakeholder)$$
$$= S(key\ stakeholder, non-key\ stakeholder)\ x$$
$$Sim(key\ stakeholder, non-key\ stakeholder)$$
$$= 0.25\ x\ \frac{2depth(LCA(key\ stakeholder, non-key\ stakeholder))}{depth(key\ stakeholder) + depth(non-key\ stakeholder)}$$
$$= 0.25\ x\ \frac{2(1)}{2+2}$$
$$= 0.12$$

Similarity between concepts in specialization hierarchy

In this case, we consider the concepts as parent-child concepts in horizontal and categorized those children concepts into two types. Those do not have their own children will be considered as "***attribute concepts***"; in contrast, those concepts which are not used as attribute for others and have child concepts will be considered as ***"non-attribute concepts"*** (e.g. key and non-key is attributing for some concepts like *software stakeholder*, *system stakeholder*)

Besides looking at the specialization tree for calculating the similarity between parent and child concepts, we also consider the weight of their relation. The weight from parent (abstract concept) to attribute concept is higher than non-attribute concepts. For example, weight from "*stakeholder*" to "*software stakeholder*" is lower than weight of "*stakeholder*" to "*key stakeholder*". Therefore, the similarity between parents to attribute concepts is also higher than the similarity between parents to non-attribute concepts. Following are the default values of weight from parents to child concepts:

- WofSep1(High level): the weight of abstract concept to attribute concept's and it has default value 95% (WofSep1 = 0.95)
- WofSep2(Medium level): the weight of abstract concept to either non-attribute and attribute concept's and it has default value 90% (WofSep1 = 0.90)
- WofSep3(Low level):the weight of abstract concept to non-attribute concept's and it has default value 85% (WofSep1 = 0.85)

VI. Evaluation

First, we re-structuring the CMMI's ICM models b categorized them into 10 type categories and then we related them with their corresponding abstract concepts and assigned the weight for them. In mean times, we had defined the weight judgments as follows:

[LOW] -> 0.85;
[MEDIUM] -> 0.9;
[HIGH] -> 0.95;

Second, we determined the similar CMMI/SPICE and CMMI/COBIT procedure pairs to compute the similarity between CMMI and SPICE procedure. The "*Mapping of CMMI for Development, V 1.2*" and "*Automotive SPICE Process Assessment Model*" documents were used. With this evaluation, we too the most similarity pairs from CMMI and SPICE and calculated the similarity in procedure level.

Third, we computed the similarity for 76 activity unit pairs (36 of CMMI-COBIT and 18 CMMI-SPICE).

Then, we asked professional experts to subjectively evaluate the similarity of the activity units according to five categories: identical, high, medium, low, different.

We mapped the computed similarity values on the defined categories in order to compare them with the expert judgments:

[1, 1] -> identical
[0.68, 1] -> high
[0.3, 0.67] -> medium
[0, 0.3] -> low
[0,0] -> different

Finally, we asked professional expert to subjectively evaluate the results on the procedure level.

Table below is the lists of some positive example of compared activity units showing the similarity metric values(SM) and expert's judgments (EJ).

Compared Activity Units	SM	EJ
SPICE ACQ.13.BP11	**0.67**	**M**
Identify risk associated with project life cycle	**(m)**	
CMMI RD RSKM SP2.1		
Identify/document risks		
SPICE ENG.2.BP2	**0.88**	**H**
Analyze system requirements	**(h)**	
CMMI RD SP 3.3.3		
Analyze requirements		
SPICE SPL.1.BP8	**0.44**	**L**
Define the roles of the stakeholders involved in the strategic planning process.	**(l)**	
CMMI SAM SP 1.3		
Plan the involvement of identified stakeholders.		
COBIT PO10.7	**0.87**	**H**
Establish a integrated project plan	**(h)**	
CMMI PP SP2.7		
Establish the overall plan		

Fig. 7 Evaluation example

VII. Conclusion

This paper represents the methodology for structuring the existing comparison approach called MoSaIc comparison approach. The main objective is to evaluate and improve the above mentioned comparison approach aims for developing the systematic similarity calculation tool of different reference models.

The paper is stresses on three processes, re-structuring the MoSaIc's ICM concepts, mapping ISM with corresponding ICM concepts and computes the similarity of reference models. For the formal process, we used the idea of aspects to categorize all ICM concepts and then applied the "ComposedOf" relation to abstract concept and its sub concepts and then the structuring rules which defined in 4.1 was applied for structuring the ICM concepts.

The second process was done by analyzed the 90 similar procedure pairs from CMMI and SPICE and applied the ISM modeling rules [Appendices A.1] to generate the ISM models; in meantime, the ICM was also modeled accordingly. Based on the structure and similarity algorithm, we finally can calculate the similarity between two reference models.

References

[1] Automated Comparison of Process Improvement Reference Models based on Similarity Metrics,Simona Jeners, Horst Lichter, Elena Pyatkova RWTH Aachen University, Research Group Software Construction Aachen, Germany simona.jeners, horst.lichter, elena.pyatkova@rwth-aachen.de

[2] MoSaIC Reference Model Modeling Rules, Simona Jeners RWTH Aachen University, Research Group Software Construction, 2012 simona.jeners@rwthaachen.de

[3]Ferchichi, A., Bigand, M., Lefebvre, H.: An Ontology for Quality Standards Integration in Software Collaborative Projects. In Proceedings of MDISIS (Model Driven Interoperability for Sustainable Information Systems) 2008. Montpellier, France, 1998.

[4]Liao, L., Qu, Y., Hareton, K. N. L.: Software Process Ontology and Its Application. In Proceedings of the IWFST-2005 (International Workshop on Future Software Technology, pages 1□10. Shanghai, 2005.

[5]Normalization of ontology implementation by towards modularity, re-use and maintainability. Alans Rector, Department of computer science, university of Manchesterrector@cs.man.ac.uk

[6]A.Rector,C.Wroe,J.Rogers,andA.Roberts.Untangling taxonomies and relationships: Personal and practical problems in loosely coupled development of large ontologies. In Y.Gil, M. Musen, and J. Shavlik, editors,Proc. Of the 1st Int. Conf. on knowledge Capture (K-CAP2001), page139-146.ACM, 2001.

[7]A.L. Rector. Clinical terminology: Why is it so hard? Method 7 Information in Medicine, 38:239-252, 1999.

[8]D.S. Solomon, C. Wore, J.E. Rogers, and A. Rector. A reference terminology for drug. In Proc. American Medical Informatics Association Annual Symposium 1997(AMIA-1997), page152-155, 1997.