Achieving ontology interoperability using formal concept analysis: an approach to inter-organizational collaboration

N. El-Gohary & T. El-Diraby
Department of Civil Engineering, University of Toronto, Toronto, On, Canada

ABSTRACT: Semantic systems that are based on ontologies are required to facilitate the collaboration between project teams and stakeholders in the infrastructure and construction domain. However, since no one ontology will be universally accepted, techniques are needed to achieve interoperability between different ontologies. This paper proposes an approach based on formal concept analysis (FCA) and relational concept analysis (RCA) for ontology merging that aims at enhancing inter-organizational collaboration among different organizations that use different ontologies / models.

Construction Informatics Digital Library http://itc.scix.net

1 INTRODUCTION

Advanced systems are needed to facilitate the collaboration between project teams and stakeholders in the infrastructure and construction domain. However, this requires more than the technical connectivity of entities or the sharing of online data. What is required is a semantic system that embeds human expertise and knowledge in a computer-understandable format, i.e. an ontology-based web system. However, since the development of a single ontology is not feasible given the multidisciplinary parties that are involved and the unique nature of each project; any semantic system has to provide for ontology interoperability. Thus, there is a need to develop a web-based system that depends on two components: 1) a core ontology for the infrastructure and construction domain, and 2) an ontology merging mechanism to allow for interoperability between different ontologies.

2 ONTOLOGY INTEROPERABILITY

Different definitions are provided in the literature for terms related to ontology interoperability including mapping, aligning, articulation, matching and merging (Klein 2001, Kalfoglou & Schorlemmer 2003). Mapping, aligning, and articulation are referred to in this paper as ‘mapping’, while ‘matching’ and ‘mapping’ are used interchangeably. In this paper, ontology mapping / matching is defined as the process of finding semantic correspondences between entities (i.e. concepts, relations, axioms etc.) of two or more ontologies. On the other hand, ontology merging is defined as the process of deriving a new (target) ontology from two or more existing (source) ontologies (McGuinness et al. 2000). Generally, combining two ontologies is composed of two main steps: 1) finding correspondences between both ontologies (i.e. mapping) and 2) merging both ontologies based on the established correspondences. Different methodologies and approaches for ontology mapping and merging are being developed and used, such as Chimaera PROMPT, ONION, GLUE, FCA-Merge, IFMap, KAON, Edamok, Matchmaker, and OBSERVER. The presentation and discussion of these different approaches are beyond the scope of this paper.

3 REQUIREMENTS FOR ONTOLOGY MERGING TOOLS

This paper classifies requirements for ontology merging tools as either practical or technical requirements. Practical requirements are those defined by ontology users and are mainly related to the type of input available, type of output they require, and level of user interaction that is needed. On the other hand, technical requirements are those defined by the developer in response to user requirements. Two of the major or challenging requirements that face developers of ontology merging tools are: 1) merging other relations besides taxonomic relations and 2) merging axioms. Most of the available ontology mapping and merging tools allow for the generation of taxonomic relations for the new ontology. However, ontologies usually have many other ontological relations that are fundamental to their semantics. For example in an infrastructure ontology, common relations would be performed_by, uses, and controlled_by. A process is performed by an actor, uses resources, and is controlled by constraints. Such relations need to be reflected in the new ontology based on source ontologies. Similarly, current ontology mapping and merging tools do not allow for the merging of functions and axioms. However, axioms are necessary to define the semantics of both concepts and relations. Therefore, tools for merging ontological relations and axioms are needed.

4 FORMAL CONCEPT ANALYSIS

4.1 *Formal Concept Analysis*

Formal Concept Analysis (FCA) is a mathematical theory that models the concept of 'concept' using lattice theory. A formal concept is modeled as a unit that has an extent and intent. A formal context is introduced to describe extensions and intensions mathematically. A formal context is defined as a triple (G,M,I), where G is a set whose elements are called objects, M is a set whose elements are called attributes, and I is a binary relation between G and M ($I \subseteq G \times M$). The relation I expresses which attributes describe each object or which objects are described by an attribute (Stumme 2003). A formal concept represents a group of objects. It is described by objects (the extent of the concept) and attributes (the intent of the concept). The extent represents all objects belonging to the concept, while the intent includes all attributes shared by these objects. A formal context is usually represented in a form of a table, where the left column represents the object set G, the upper row represents the attribute set M, and the cross cell values represent the relationship I. On the other hand, formal concepts are usually described using a *Hass Diagram*, forming a concept lattice. In the lattice, each node represents a formal concept. The ascending path of line segments represents the subconcept-superconcept relationship and vice versa (Ganter & Wille 1999).

Formal contexts discussed above deal with binary attributes only. However, in practical applications, attributes are usually represented by a value, such as 'price' of product. These multi-valued attributes are called 'many-valued attributes', in contrast to previously discussed 'one-valued attributes'. As presented by Ganter & Wille (1999), these many-valued attributes form a many-valued context. A many-valued context is represented by (G, M, W, I), where G is a set whose elements are called objects, M is a set whose elements are called attributes, W is a set whose elements are called attribute values, and I is a ternary relation between G, M and W ($I \subseteq G \times M \times W$). Similar to one-valued contexts, a many-valued context is represented in a form of a table, where the left column represents the object set G, the first upper row represents the many-valued attribute set M, the second upper row represents the attribute value set W, and the cross cell values represent the relationship I (Ganter & Wille 1999).

In order to derive concepts from a many-valued context, the many-valued context is transformed into a one-valued context by means of conceptual scaling for attributes. Only then, the concepts of the derived one-valued context are interpreted as the concepts of the original many-valued context. Conceptual scaling is simply developing a 'conceptual scale' for each attribute. The scale is a context that is used to interpret the attributes, such that a scale for an attribute m of a many-valued context is a one-valued context given by: $Sm := (Gm, Mm, Im)$, where $m(G) \subseteq Gm$ and Mm is a set of new attributes and Im is the relation between attributes Gm and new attributes Mm. The choice of conceptual scales depends on the interpretation of the context and is not mathematically determined (Ganter & Wille 1999).

4.2 *Relational Concept Analysis*

Relational concept analysis (RCA) is an extension to formal concept analysis (FCA) proposed by Huchard et al. (2002). It allows for presenting relations between formal objects through the use of relational context family (RCF), so that formal concepts not only present shared attributes, but also reflect shared relations between objects.

A RCF is a set of formal multi-valued contexts and a set of relational attributes (extracted from relations between objects). In other words, a RCF R^S is a pair (K_R, M_R) where K_R is a set of s multi-valued contexts $K_i = (G_i, M_i, W_i, I_i)$ and M_R is a set of p relational attributes α_j such that for each j, $1 \leq j \leq p$ there exist r and q in $[1,s]$ with $\alpha_j : G_r \rightarrow 2^{Gq}$. The mappings domain and co-domain are $dom: M_R \rightarrow \{G_i\}$ and $cod: M_R \rightarrow \{G_i\}$; such that for all $\alpha : G_r \rightarrow 2^{Gq}$, $dom\ (\alpha_j) = G_r$ and $cod\ (\alpha_j) = G_q$. In addition, $rel: K_R \rightarrow 2^{M_R}$ with $rel(K_i) = \{ \alpha \mid dom\ (\alpha) = G_i \}$.

Rouane et al. (2004) propose the application of RCA to UML class hierarchies. Their proposed approach is composed of three steps: 1) encoding, 2) abstraction, 3) reverse encoding. For the encoding, classes and associations of the UML model are each presented in a separate context of the RCF. The class-association relation is reflected in both contexts as relational attributes, out-association vs. in-association for the class context and source-class vs. target-class for the association context. During the abstraction phase, a set of inter-related concept lattices are built. The initial lattice is constructed from classes and non-relational attributes. Based on this lattice, relational scaling is performed, resulting in relational attributes scaled along the lattice. Thus, the attribute name will have reference to both the association type and the formal context of the lattice.

Construction Informatics Digital Library http://itc.scix.net

Table 1. Presentation of ontology 1.

Processes (Org A)	Processes	Products		Resources			Actors			Controls	
	Design Coordination Process	Architectural Drawings	Architectural Specifications	CAD Software	Constr. Lessons Learned	Hardware	Architect	Senior Architect	Drafter	Code Specifications	Available Construction Techniques
Architectural Design Process	supported_by	impacts	impacts				performed_by	managed_by	participates_in	controlled_by	
Constructability Analysis Process		could_modify	could_modify	aided_by	aided_by	utilizes	performed_by	managed_by	participates_in	controlled_by	controlled_by
Architectural Drawing Development Process		results_in		aided_by		utilizes	supervised_by		performed_by	controlled_by	
Specification Development Process			results_in			utilizes	performed_by			controlled_by	
Design Coordination Process							participates_in	performed_by	participates_in		

Table 2. Presentation of ontology 2.

Processes (Org B)	Products			Resources	Actors				Controls	
	Master Plan	Schedule	Budget	Scheduling Software	Planner	PM Engineer	PM Estimator	Project Manager	Budget	Design Documents
Planning Process	results_in				performed_by	participates_in	participates_in	managed_by		impacted_by
Scheduling Process		results_in		aided_by	performed_by	participates_in		managed_by		impacted_by
Budgeting Process			results_in			participates_in	performed_by	managed_by		impacted_by
Cost Control Process						performed_by	participates_in	managed_by	controlled_by	
Resource Management Process						performed_by		managed_by	controlled_by	
Project Coordination Process					participates_in	performed_by	participates_in	managed_by		

A process of mutual enrichment continues until isomorphism between two consecutive lattices is achieved.

RCA provides a relational connection between formal concepts of two formal contexts. So, the intent of a formal concept of a class context may include a relational attribute referring to a formal concept of the association context. This reference reflects the links between formal objects (classes) and association type.

5 DYNAMIC ONTOLOGY MERGING - PROPOSED APPROACH

5.1 *Introduction to the Proposed Approach*

Formal and relational concept analysis could be used as a key tool for merging two or more ontologies for information exchange. For example, if we need to exchange information between two ontologies, FCA algorithms would be used to link the two ontologies by producing a merged lattice that combines the two ontologies, thus allowing for a seamless exchange of information. In this case, each ontology will be represented in a form of context table. Consequently, the result will be a number of formal contexts that need to be merged in a dynamic environment. The methodology of the proposed approach consists of five main steps: 1) encoding, 2) axiomatic scaling, 3) mapping, 4) merging, and 5) reverse encoding.

5.2 *Encoding*

The aim of the encoding is to present each ontology in two main contexts. The first context, K_1, includes ontology concepts (including attributes of concepts). The second context, K_2, presents ontology relations (including attributes of relations). The concept-relation link is reflected in both contexts as relational attributes, out-relation vs. in-relation for the ‘ontology concepts context’ and source-class vs. target-class for the ‘relation context’. For example, consider that Tables 1 and 2 present two different ontologies developed by two different organizations. Table 3 then shows a formal context presenting concepts (of ontology 1) as formal objects and both their non-relational attributes (such as process and product) and relational attributes (out-relations and in-relations) as formal attributes. On the other hand, Table 4 presents ontology relations as formal ob-

Construction Informatics Digital Library http://itc.scix.net

jects, while their non-relational attributes (such as name) and their corresponding relational attributes (source-concepts and target-concepts) are reflected as formal attributes. In order to reduce the scale of the example, only few relations of ontology 1 are considered.

Table 3. Concepts of ontology 1 and their attributes.

Concept	process	product	out-Relation	in-Relation
Architectural Design Process	x		supported_by, impacts	
Constructability Analysis Process	x		could_modify	
Architectural Drawing Development Process	x		results_in	
Specification Development Process	x		results_in	
Design Coordination Process	x			supported_by
Architectural Drawings		x		impacts, could_modify, results_in
Architectural Specifications		x		impacts, could_modify, results_in

Table 4. Relations of ontology 1 and their attributes.

Relation	Name	Source	Target
Supported_by	supported_by	Architectural Design Process	Design Coordination Process
Impacts	impacts	Architectural Design Process	Architectural Drawings
Could_modify	could_modify	Constructability Analysis Process	Architectural Drawings
Results_in	results_in	Architectural Drawing Development Process	Architectural Drawings
Impacts	impacts	Architectural Design Process	Architectural Specifications
Could_modify	could_modify	Constructability Analysis Process	Architectural Specifications
Results_in	results_in	Specification Development Process	Architectural Specifications

5.3 *Axiomatic Scaling*

Axiomatic scaling aims at reflecting some of the axioms of the source ontologies in the formal context to achieve partial axiom merging. For example, if ontology 1 has an axiom stating that if an actor has a role it implies that he has a right too, then the formal context of Table 5 could be scaled according to this axiom, as shown in Table 6: $\forall(x)\,(\text{actor}(x) \wedge \text{has\_role}(x)) \supset \text{has\_right}(x)$. However, this approach does not yet propose a way for incorporating all axioms of the source ontologies.

Table 5. Partial formal context.

	actor	has_role
Architect	x	x
Senior Architect	x	x
Drafter	x	x

Table 6. Partial axiomatically scaled formal context.

	actor	has_role	has_right
Architect	x	x	x
Senior Architect	x	x	x
Drafter	x	x	x

5.4 *Mapping*

Mapping candidates will be suggested to the user or supply chain manager based on pre-defined algorithms. Four main types of heuristics will be used to suggest mappings: 1) name-similarity between concept names, 2) definition-similarity between concept definitions, 3) hierarchical-similarity based on similarity between taxonomical hierarchies and is-a relationships, and 4) relational-similarity matches based on similarity of other taxonomical and ontological relations between concepts. More discussion about the mapping methodology will be presented in future work.

5.5 *Merging: Multiple Lattice Construction*

A set of inter-related concept lattices are constructed at this stage (Rouane *et al* 2004). The initial lattice is constructed from concepts and non-relational attributes. For example, lattice $L^{0}_{concpet}$ (Fig. 1) is built from context $K^{0}_{concpet}$ (Table 7). This lattice constitutes the first iteration of the construction process.

The second iteration starts by relational scaling based on lattice $L^{0}_{concpet}$, resulting in relational attributes scaled along the lattice. Thus, the attribute name in the scaled context will have reference to both the relation type and the formal context of the preceding lattice. A process of mutual enrichment continues until isomorphism between two consecutive lattices is achieved. For example, Table 8 shows a scale context with: a) ontology concepts that are source concepts as objects, and b) relational attributes scaled along the existing lattice $L^{0}_{concpet}$ and thus corresponding to formal concepts of $L^{0}_{concpet}$ which have these ontology concepts in their extents.

Construction Informatics Digital Library http://itc.scix.net

Table 7. Formal context K^{0}_{concpet}.

	process	product	resource	actor	control	design Document	software
Architectural Design Process	x						
Constructability Analysis Process	x						
Architectural Drawing Development Process	x						
Specification Development Process	x						
Design Coordination Process	x						
Planning Process	x						
Scheduling Process	x						
Budgeting Process	x						
Cost Control Process	x						
Resource Management Process	x						
Project Coordination Process	x						
Architectural Drawings		x				x	
Architectural Specifications		x				x	
CAD Software			x				x
Constr. Lessons Learned			x				
Hardware			x				
Architect				x			
Senior Architect				x			
Drafter				x			
Code Specifications					x		
Available Construction Techniques					x		
Master Plan		x					
Schedule		x					
Budget		x			x		
Scheduling Software			x				x
Planner				x			
PM Engineer				x			
PM Estimator				x			
Project Manager				x			
Design Documents					x	x	

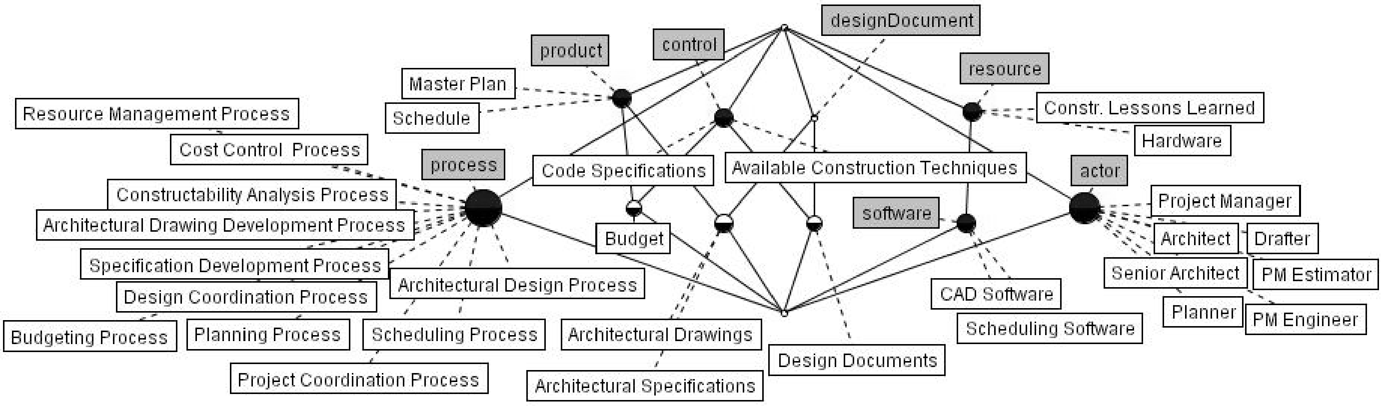

Figure 1. Lattice L^{0}_{concpet}.

Table 8. Scale context $K_{\text{source-concept}}$.

	sc-L10-Fc0	sc-L10-Fc5
Architectural Design Process	x	x
Constructability Analysis Process	x	x
Architectural Drawing Development Process	x	x
Specification Development Process	x	x

The relational attribute has the following notation r-L*ij*-FC*k*, where r refers to the relation type (being source concept in this case, abbreviated as sc), *Lij* is a reference to the existing lattice (*i* stands for the iterative step of the lattice construction and *j* is the context number), and *k* is the formal concept index in lattice *Lij*. A similar scale context is built for target concepts, as per Table 9. Table 10 shows the relational extension of context K^{1}_{relation}, which is de-

Construction Informatics Digital Library http://itc.scix.net

rived by linking relations to concepts and thus having relations as objects and relational attributes corresponding to formal concepts of L^0_{concpet}. The relation context K^1_{relation} is then constructed by adding the non-relational attributes of relations to the relational extension of context K^1_{relation}, as per Table 11. Accordingly, the relational lattice L^1_{relation} (Fig. 2) is constructed from the context K^1_{relation}.

Table 9. Scale context $K_{\text{target-concept}}$.

	tc-L10-Fc0	tc-L10-Fc3	tc-L10-Fc5	tc-L10-Fc7
Design Coordination Process	x		x	
Architectural Drawings	x	x		x
Architectural Specifications	x	x		x

Table 10. Relational extension of K^1_{relation} through $K_{\text{source-concept}}$ and $K_{\text{target-concept}}$.

	sc-L10-Fc0	sc-L10-c5	tc-L10-Fc0	tc-L10-Fc3	tc-L10-Fc5	tc-L10-Fc7
Supported_by	x	x	x		x	
Impacts	x	x	x	x		x
Could_modify	x	x	x	x		x
Results_in	x	x	x	x		x

Table 11: Relation context K^1_{relation}.

	name	sc-L10-Fc0	sc-L10-c5	tc-L10-Fc0	tc-L10-Fc3	tc-L10-Fc5	tc-L10-Fc7
Supported_by	supported_by	x	x	x		x	
Impacts	impacts	x	x	x	x		x
Could_modify	could_modify	x	x	x	x		x
Results_in	results_in	x	x	x	x		x

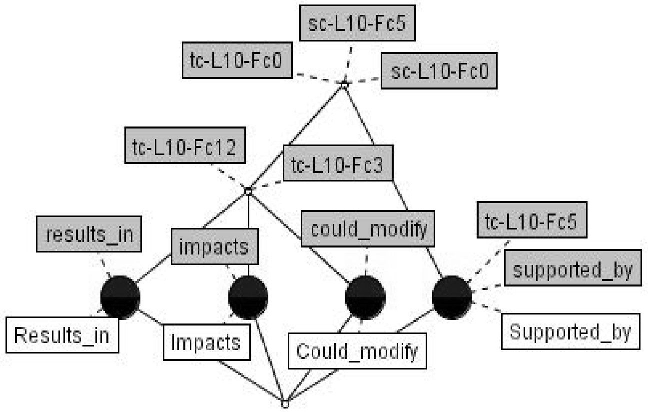

Figure 2. Relational lattice L^1_{relation}.

The second part of the second iteration deals with constructing the relational 'ontology concept lattice'. For example, Table 12 shows: a) a scale context with ontology relations that are out-relations as objects, and b) relational attributes scaled along the existing lattice L^1_{relation} and thus corresponding to formal concepts of L^1_{relation} which have these ontology relations in their extent. The similar scale context is built for in-relations, as per Table 13. Table 14 shows the relational extension of context K^1_{concept}, which is derived by linking concepts to relations and thus having concepts as objects and relational attributes corresponding to formal concepts of L^1_{relation}. The ontology concept context K^1_{concept} is then constructed by adding the non-relational attributes of concepts to the relational extension of context K^1_{concept}. Accordingly, the relational lattice L^1_{concept} (Fig. 3) is constructed from the context K^1_{concept}.

Accordingly, lattices L^1_{relation} and L^1_{concept} are both interpreted together. RCA provides a relational connection between formal concepts of the two formal contexts. So, the intent of a formal concept of an 'ontology concept lattice' may include a relational attribute referring to a formal concept of the 'ontology relation context'. This reference reflects the links between formal objects (ontology concepts) and ontology relations.

5.6 *Reverse Encoding*

The main objective of the reverse encoding step is to represent the merged combined lattice into a merged ontology (using the same format as the original source ontologies). Details of reverse encoding will be presented in future work.

6 CONCLUSION

Collaboration between semantic ontology-based systems is hindered by the lack of interoperability between different ontologies that are used by collaborating organizations. This paper proposed an approach based on FCA and RCA for developing a merged ontology for creating interoperability among more than one organization. Accordingly, organizations can use different process models and ontologies, but can still collaborate since interoperability is facilitated though dynamic ontology merging.

Construction Informatics Digital Library http://itc.scix.net

Table 12. Scale context K $_{\text{out-relation}}$.

	out-L21-Fc0	out-L21-Fc1	out-L21-Fc2	out-L21-Fc3	out-L21-Fc4	out-L21-Fc5
Supported_by	x					x
Impacts	x	x		x		
Could_modify	x	x			x	
Results_in	x	x	x			

Table 13. Scale context K $_{\text{in-relation}}$.

	in-L21-Fc0	in-L21-Fc1	in-L21-Fc2	in-L21-Fc3	in-L21-Fc4	in-L21-Fc5
Supported_by	x					x
Impacts	x	x		x		
Could_modify	x	x			x	
Results_in	x	x	x			

Table 14. Relational extension of K$^1_{\text{concept}}$ through K $_{\text{out-relation}}$ and K $_{\text{in-relation}}$.

	out-L21-Fc0	out-L21-Fc1	out-L21-Fc2	out-L21-Fc4	in-L21-Fc0	in-L21-Fc1	in-L21-Fc5
Architectural Design Process	x						
Constructability Analysis Process	x	x		x			
Architectural Drawing Development Process	x	x	x				
Specification Development Process	x	x	x				
Design Coordination Process					x		x
Architectural Drawings					x	x	
Architectural Specifications					x	x	

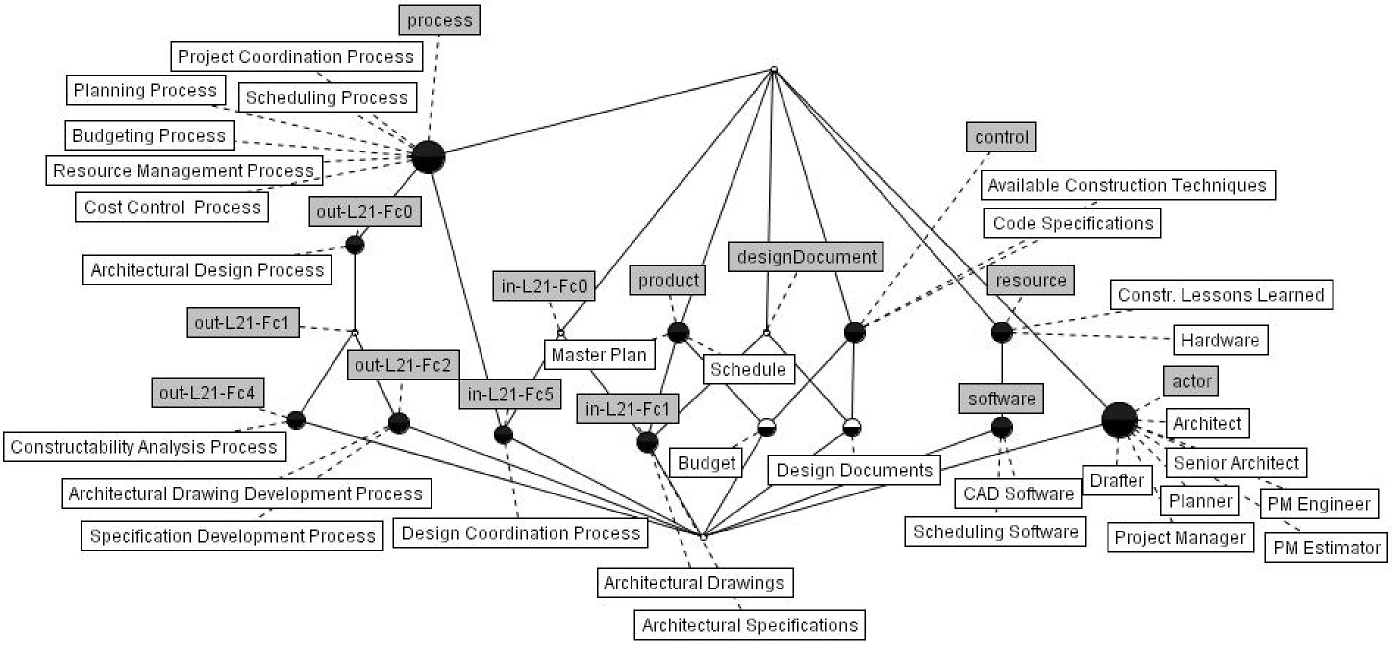

Figure 3. Lattice L$^1_{\text{concpet}}$

REFERENCES

Ganter, B. & Wille, R. 1999. Formal concept analysis: mathematical foundations. New York: Springer.

Huchard, M., Roume, C. & Valtchev, P. 2002. When concepts point at other concepts: the case of UML diagram reconstruction. Proc. 2nd Intl. Workshop on Advances in Formal Concept Analysis for Knowledge Discovery in Databases (FCAKDD), Lyon, 23 July 2002: 32-43.

Kalfoglou, Y. & Schorlemmer, M. 2003. Ontology mapping: the state of the art. The Knowledge Engineering Review 18(1): 1-31.

Klein, M. 2001. Combining and relating ontologies: an analysis of problems and solutions. Proc. IJCAI-2001 Workshop on Ontologies and Information Sharing, Seattle, 2001: 53-62.

McGuinness, D. L., Fikes, R., Rice, J. & Wilder, S. 2000. An environment for merging and testing large ontologies. Proc. 7th Intl. Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, 12-15 April 2000.

Construction Informatics Digital Library http://itc.scix.net

Rouane, M. H., Valtchev, P., Sahraoui, H. & Huchard, M. 2004. Merging conceptual hierarchies using concept lattices. MechAnisms for SPEcialization, Generalization and inHerItance Workshop (MASPEGHI) at ECOOP 2004, Oslo, 15 June 2004.

Stumme, G. 2003. Off to newshores: conceptual knowledge discovery and processing. International Journal Human-Computer Studies 59: 287-325.

Construction Informatics Digital Library http://itc.scix.net