On the Evaluation of Reference Models for Software Engineering Practice

Balbir S. Barn
Middlesex University
The Burroughs, Hendon
LONDON, UK, NW4 4BT
+44 208 411 4563
b.barn@mdx.ac.uk

ABSTRACT

This paper argues that conceptual models and more specifically reference models play a key role in the specification and design of information systems. However, an effective evaluation strategy of such models is a relatively immature field. The paper presents the key challenges for this evaluation activity and articulates an approach for understanding how to evaluate models based on the information and cognitive theories of structuralism and conversation theory. An example of a reference model developed for the Higher Education domain is used as a case study to illustrate how the approach may be applied.

Categories and Subject Descriptors

D.2.1 [**Requirements**]: Requirements / Specifications – *elicitations methods, languages, methodologies.*
D.3.1 [**Specification**]: Specifications *–methods, languages, methodologies.*
D.2.11 [**Software Architecture**]: *information hiding, languages*

General Terms

Management, Documentation, Design, Experimentation, Standardization, Languages, Theory.

Keywords

Reference Model, Conceptual Model, Evaluation, Variability

1 Background

Conceptual modeling is a viewed as a capstone of many software engineering approaches where it is used to as an approach to user requirements definition and as a basis for developing information systems to meet those requirements [12]. More generally, conceptual modeling may be used to support acquisition, standardization, and integration of information systems. Such uses may happen at multiple different levels:

- Application level: models are used to define requirements for specific applications
- Enterprise level: models are used to define requirements for an entire organization and provide the basis for data management across the enterprise
- Industry Level / Sector Level: a variant of conceptual models – reference models are used to define information and behavior requirements for a sector as the basis of standardization and development of generic software solutions.

ISEC'09, February 23–26, 2009, Pune, India.

The centrality of conceptual modeling to systems design has led many notations and methods. Further the quality of conceptual models has the potential to the make a significant impact on other IT artifacts. For example, inadequacy of conceptual models may lead to problems in the implemented system and thus may determine the acceptability and usability of software systems [10]. One commonly cited reason for systems failure is miscommunication between business and IT personnel [18]. Modeling or more specifically, conceptual modeling [17] provides a mechanism by which a shared understanding between business domain specialists and IT specialists positively enhances the alignment of business and IT goals leading to improved quality of IT Solutions [16].

More recently models and modeling has become important as the Object Management Group (OMG) in particular has been leading industry initiatives in the promotion of technologies, methods and standards under the banner of model driven architecture (MDA) [8, 13]. Mellor et al. [11] articulate key aspects of model driven development including: raising the level of abstraction where the knowledge of a particular domain is formalized in a high a level of language as possible; raising the level of re-use (the ability to reuse previous development effort – moving re-use from the code level to domain model re-use; and supporting design-time interoperability which enables re-use of applications.

Conceptual models can address different aspects of information systems requirements. The most commonly used application scope of conceptual models is in some form of data (information) modeling. However, it is possible to have conceptual models for process description, and even functional description such as use cases. Moody identified range of models and their scope with a view to assessing their quality. Such a range requires different languages and structural rules for capturing the model [12].

1.1 Reference Models

A special form of conceptual modeling is *Reference modeling* – where the model represents a general solution to a particular class of problem in a specific domain. The current general trend from bespoke development to tailoring and adaptation (exemplified for example by the SAP approach [20]) means that the importance of *Reference modeling* is increasing and the oft-cited benefits include: re-use of knowledge, a rise in quality, and corresponding reductions in risk, cost and time [7, 19]. Even allowing for these attributed benefits, Thomas [19] further argues, “no uniform grasp of the term reference model exists”. This confusion partially arises out of the tendency to declare Application information models and/or enterprise Information models as “reference models”. One approach to seeking clarity on what is meant by reference model were to review the outputs of several existing reference models (ranging from domains such as workflow, security and education) and from these synthesizing a definition [2].

> “A reference model is based on a small number of unifying concepts and is an abstraction of the key concepts, their relationships, and their interfaces both to each other and to the external environment. A reference model may be used as a basis for education and for explaining standards and methods to a non- specialist and can be viewed as a framework for comparing architectures and operations of existing and future systems. “

More recently recognizing some of the difficulties in engaging with stakeholders in the production of reference models following from a series of funded reference model projects [http://www.jisc.ac.uk/whatwedo/programmes/elearning_framework/refmodelssept05.aspx], the Joint Information Systems Committee (JISC) in the UK has embarked on the development of “domain maps” these are more relaxed forms of reference model but still attempt to achieve the same purpose as a reference model. JISC has set out to ‘map out’ the domain of higher education in the UK with the intent of defining canonical descriptions of informational and functional elements of the wider domain (and sub-domains) in order to support the development, purchase and integration of information systems in higher education. For JISC, domains can be sub-divided for mapping analysis purposes. Example: Course Management is a sub-domain of Learning and Teaching.

A domain map description comprises informal domain maps (main business concepts and their relationships), formal domain models (captured in a more rigorous modeling language such as UML) and functions which can ultimately be described as business processes. Eventually, it is intended that a fully mapped out domain will also describe services (as part of a service oriented architecture) that implement the processes belonging to the domain. Collectively the artifacts are conformant to the reference model described above. In describing the domain maps – multiple language forms can be used for example, topic maps [http://www.frema.ecs.soton.ac.uk/], information models (as UML Class Diagrams), process models (UML Activity Diagrams or BPMN compliant diagrams) [l]. This range and complexity of multiply deployed languages raises one of several key challenges when faced with the need to evaluate the quality of a reference model.

1.2 The structure of the paper

In this paper, the focus is on reference models and their evaluation and assessment needs. Section 2 will outline the main challenges and the motivation for this research. Section 3 provides a description of the proposed evaluation framework. In particular two theories of information and cognitive understanding are used to support the evaluation approach. Section 4 uses a case study as a vehicle for exploring some of the artifacts of the reference model process and their assessment against the framework. Finally a summary of the outlook for the ongoing research is presented.

2 Challenges for evaluation, quality and synthesis of reference models

Conceptual or reference modeling in general has the potential of significant impact on systems success/failure. Conceptual models address the gap between business/domain experts and IT specialists so an analysis of systems design based on faulty models may result in expensive decisions. Thus conceptual models raise several key challenges. These are:

1. Evaluation of the quality of an IT artifact such as a conceptual model
2. Reference models in particular need to support variability within a vertical market. How can variability be supported?
3. Community acceptance of reference models

2.1 Evaluation of Quality

Evaluation of quality of conceptual models is relatively under researched. While there are international standards for software systems there is “little agreement among experts as to what makes a “good” model” [12]. Partly this may be attributed to relatively immature field, or more likely the production of a conceptual model is socio-technical event so evaluation is against a person’s tacit needs. The production of a model could be seen as an instance of the SECI (Socialization, Externalization, Combination and Internalization) cycle – the externalization (making explicit) of tacit knowledge [http://www.12manage.com/methods_nonaka_seci.html]. Efforts have been made to contextualize production of conceptual models in extant theories such as structuralist appoaches [15] or philosophy of languages [17] as means of addressing evaluation requirements. The framework in the next section develops aspects of these approaches.

2.2 Variability

A reference model sets out requirements for information systems for a particular vertical market to support development, integration and standardization etc. Within a particular market though, while there may be a canonical model, there is still a need to support variability. This can be further characterized by the dichotomy of the two polar positions sometimes described as commonality versus variability – that is describing those software features that are common and fit for purpose for multiple clients and those features that are adaptable by the end-user or by the software provider in order to maximize software fit for a particular client.

Software design has always needed to consider the tension between designing for purpose for a specific client versus designing for potential re-use. Common and varying features of software can be supported by a number of implementation technologies such as interface based design (a specification focus) or class inheritance frameworks (an implementation focus), understanding and supporting C/V tension earlier in the software lifecycle is much more limited. Jacobsen et al [9] discuss how software product families / product line architectures can be developed by placing notions of designing for re-use at the centre of method architecture. In particular, variation at use case level is described. However, there is limited evaluation of the effectiveness of the methods and techniques described. The approaches presented are largely theoretical. Similarly The UML 2.0 Use Case Diagram provides extensive abstraction concepts for specifying and accommodating C/V but it is difficult to ascertain if features (beyond "extends" and "uses" relationships) are extensively used. Dobing and Parsons [5] identified a research agenda for exploring the application of Use Case Diagram in more detail and in a more recent empirical study of the use of UML diagrams presented evidence that focused mainly on specification of use case narratives [6]. When variability is considered in the context of reference models – where there are different forms (models in different languages for example) - variability needs to addressed for each form. For reference modeling Becker et al have been examining the use of ontologies and related toolsets as an approach for configurative reference modeling. In their work the context in which an element is used to enrich the model and the information and relationships forms an ontology which can be used to support reasoning and deduction [3].

2.3 Community Acceptance

The success of a reference model depend upon the acceptance of a model by a community. The assertion by developers of a reference model is not sustainable unless there is at least one application of the reference model – "This attribute can ultimately be proved only by way of the model being applied at once" [19, p22]. While this is a start, community engagement with a reference model for review, usage and refinement is also essential.

2.4 Motivation of this work

This paper seeks to address these challenges by proposing an approach located in theories of information systems development and provides an illustration of the application of the approach in the development of a reference model for a sub-domain in higher education.

3 A Structural framework for evaluation

This section describes the proposed reference model evaluation framework. The structural concepts discussed are based on the last two decades of research on conceptual model evaluation. Two different theories are used to help explain the approach to evaluation.

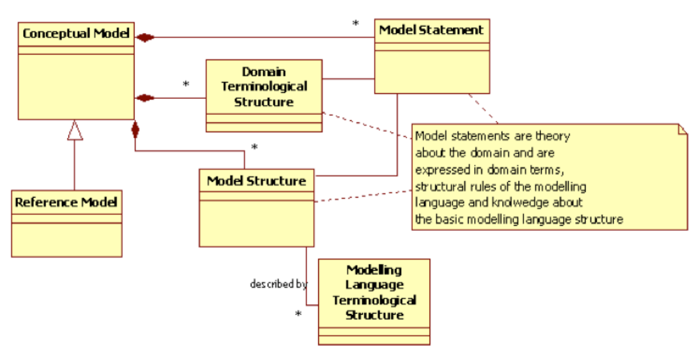
3.1 Model Structure

Model quality has largely focused on a structural language based view - models depend upon a language based meta model – without a meta model the syntax and semantics of constructs used in a model would be ambiguous. In this area, notions of completeness, construct deficit, overload, and redundancy have been developed based on the work done by Wand and Weber [22]. This language based view is inherently limited as it not contextualized in the domain. More recently, Pfeffffer[15] argues that a conceptual model is a theory and has used structuralism as tool for understanding the conceptual model as theory that is as a conjunction of three levels of language discourse: the meta modeling language, the conceptual modeling language and the domain specific terms used in the conceptual model. This has the potential to allow evaluation at three levels but requires tool vendors to develop interlinks between the levels. New research in domain specific languages (and tool support) could present opportunities for making evaluation of quality easier. In the end, though, implementation of a model is likely to be the ultimate arbiter.

Table 1: Structural Framework

Challenge	Evaluation Criteria	Evaluation Approach
Model Quality	Construction adequacy Language adequacy Clarity Comparability Completeness Links to methods	Language tailoring Domain Specific Languages Implementation Conversation theory
Variability Modelling	Support for canonical representation Support for variability modeling Multiple languages	Reference modeling rules Language tailoring and use of UML profiles Implementation
Method	Appropriateness Completeness Consistency Adaptability Tool support Link to model artifacts	Method engineering
Community Acceptance	Agreement Visible, public discussion Embedding in the community of practice	Conversation theory enactment

Figure 1 Structuralist view of reference models



The conceptual model presented above represents the author's interpretation of structuralist theories applied to conceptual model understanding. Thus the conceptual model comprises a set of model statements. These model statements are expressed in the terminological structure of the domain (e.g. notions of Customer, Order etc.), Rules of constructing models based on the meta mode l of the modeling language, and the terminological structure of the meta modeling language. The interactions between the levels is the challenge which requires enhanced tool support.

Evaluation of the support for variability of reference model can be determined by assessing the following: Is it possible to separate out a canonical model from variants? Are there mechanisms for documenting the rules of how variability can be defined? Is variability definition possible for all types of artifacts produced for the reference model? E.g. Variation in information models as well as variation in core business processes documented for a particular domain. Approaches to variability can include the use of meta language extensions using techniques such as UML profiles.

3.2 Interpreting Conversation Theory

A conceptual model represents the arrival of a shared understanding of a subject area between two different actors – the domain expert and the systems designer. One way of viewing the process of understanding is through the lens of Conversation Theory [14].

There is a considerable body of literature, including by the originator Pask and his colleagues, some of which is included in the references in this paper. However, here an overview restricted to those elements deemed pertinent by the author, is presented. While Pask and his co-researchers perceived CT as general theory of human communication, in this paper the embodiment of CT as a theory of learning and teaching is the focus. From this perspective one participant (say, the domain expert) describes a body of knowledge to a second participant (the Systems Analyst). Both these participants are a type of organization – the psychological (p-) individual. A p-individual is a stable closed system comprising memory (facts), rules for interpreting the memory (concepts), rules for structuring the derivation of concepts – "how to" understand concepts and rules for understanding how topics in the memory relate to each other. In a basic conversation ("skeleton of a conversation"), there are two levels – the "how" and "why". The "how" level describes how to do a topic for example, recognizing, constructing and maintaining a topic, while the "why" level is focused on explaining or justifying the topic perhaps in terms of other topics. The basic conversation is provocative, that is participants are provoked into constructing understandings of each others' beliefs. A "modeling facility" provides the medium in which concepts are understood between individuals. In the context of this paper, the medium may comprise tools and conceptual models that are produced.

A key aspect of CT is the embodiment of knowledge. A body of knowledge (e.g. the workings of the combustion engine, finite state machines or any other coherent whole) is viewed as a set of topics or facts that are related to each other. Relations between topics are either decompositional (hierarchical) or analogous (heterarchical), when such relationships and topics are static then that static representation is called an entailment structure. When a topic is understood by a learner (via a reproducible procedure) then the topic also exists as a concept for potential sharing with another p-individual. In order to arrive at an understanding of a topic, an equivalent task structure (task breakdown and analogous tasks) serve to provide the procedures for understanding it. A learner will select and perform appropriate tasks within the modeling facility to understand a topic.

As an example of the use of CT, consider the conceptual model presented in Figure 1. In this "conversation" the two actors or participants could be the author who is presenting the model and the reader who is interpreting the model. Through the medium of this paper, the concepts or knowledge is presented to the reader who would via email perhaps, describe their interpretation of the model. After a series of talkbacks, the reader is able to convey the meaning of the model. Conceptual models aid in this process.

Reference models do not generally mandate approaches to deriving the key IT artifacts. A method will address the needs of intended applications, describe the conditions of applicability of the method and define the products and results of an application of a method or technique. However, given the link between methods and their resulting artifacts (e.g. conceptual models, process models etc.) the evaluation of methods for the reference model would appear to be an appropriate dimension. Examples of criteria for the methods dimension would be appropriateness of a particular method or technique, the completeness of the method – does it address production of all artifacts of the reference model and the consistency of approaches mandated.

In the next section an elaboration of how the framework can be used to examine the evaluation of a reference model produced for higher education is developed.

4 Case study

4.1 Context

This section provides a short description of the context of the case study for this paper. The e-Framework (http://www.e-framework.org) is an initiative by the U.K's Joint Information Services Committee (JISC) and Australia's Department of Education, Science and Training (DEST) to build a common approach to Service Oriented Architectures for education and research across a number of domain areas including course management.

In higher education, the course (or programme management) is an area of activity that involves key roles from across the university community. The domain includes many functions:

1. course design – the process which a programme of study is designed, validated for use and then advertised for potential students
2. course management – ongoing changes to a programme – maintainance, update, communication with students, assessment design and changes
3. liaison with external bodies – such as examiners, professional bodies etc.

The domain may also share interactions with other domains such as student achievement, results and so on.

The domain has a specific domain vocabulary and may incorporate rules, which affect the governance of the activities within the domain.

JISC has funded several projects, which have explored the construction of reference models to describe the essential characteristics of the course management domain. While various reference models have been produced the challenges of evaluating the models has led to this notion of an evaluation framework. The following provides some sample results of the reference model and their respective evaluation strategies as constrained by the framework described earlier.

The basis approach used in the research was a case study approach. Systems analysis was undertaken at four institutions. Visual models were constructed and evaluated and an approach to synthesizing the models from each institution into a single canonical model was developed and then applied. This approach includes rules for identifying variances between processes and is described in more detail elsewhere [2]. These models were used as input to the software design and implementation stages to develop a set of software services that allowed us to automate part the business process.

4.2 Model quality (structure, language)

For the course management domain – several conceptual model artifacts have been produced to capture both the information model and functional requirements of the domain. Information models were defined using Class diagrams so the meta model language was UML. Domain concepts were identified using a series of workshops and interviews conducted by experienced business analysts. Verification of the information models (quality assessment) was by a series of "talkbacks" using principles from conversation theory [14] where the models were re-presented back to the domain experts for validation. Only minimum language constructs were explained. A second iteration of the information model definition used a different approach – we focused on so called problems in the domain as a means of requirements definition. A key result here was the identification of a conceptual model as a theory. As indicated earlier, the course management domain is concerned with how a programme specification of a course of study may evolve over time. New study elements may be added, new modes of study be possible and these changes are governed by clearly stated business rules which determine the integrity and quality of the award. As these problem patterns were identified and we constructed information models to represent them, the emerging model displayed many similarities to standard software configuration management. We postulate that a conceptual model for configuration management is a solution pattern for addressing course specification management. So quality of information models in the programme specification domain can be compared to a generic conceptual model for software configuration management.

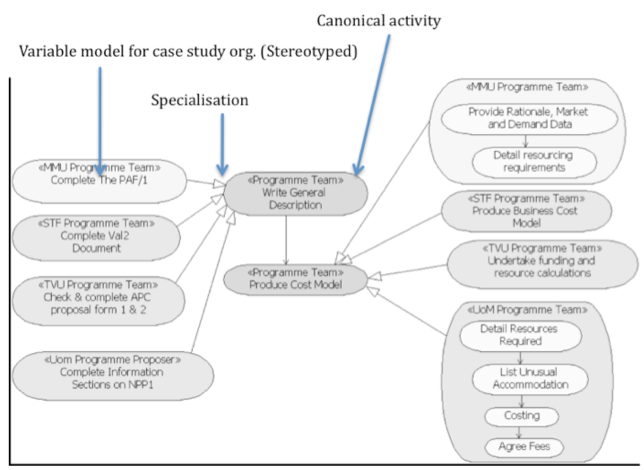
For functional models in the domain, in the first iteration of the research, UML activity diagrams were produced, again, the primary means of verification was by re-presenting the models back to the experts. It was noted that business experts found the basic activity modeling language intuitive and useful as an articulation of their own understanding of the domain. A second iteration of the functional or process view of the domain was the use of the Business Process Modeling Notation (BPMN) [4] – this did not add any additional ease to understanding despite the promotion of BPMN as targeted at business users.

Variability in the reference model was addressed at two different areas – informational and functional (business process). As indicated earlier, four case study organizations were subject to the analysis. From these studies – canonical models (those features / constructs common to all) were constructed. Rules for extending and supporting variability from these canonical models were defined, and additional language semantics to the core UML modeling semantics specified. For example, abstract activities and specialization of activities stereotyped with domain information of the case study organizations.

Table 2 Variability Rules

Condition	Action
Commonality	Separate out activities and objects that are common to all cases under study
Activity variability	If an activity varies by case study institution, insert an abstract activity and create case study specialized activities (Use extended semantics – UML profiles to support distinctions)
Nested Activities	If a set of activities correspond to one abstract activity – use nested activities to represent the mapping

Figure 2 Variability Specification in a Process Model



One such case is shown above:

In this diagram (a subset of one of the key business processes in the course management domain) Two canonical activities are shown – because they indicate a variation point for individual case study institutions, they are specialized by activities which are specific to an institution. UML stereotypes are used to indicate how the variation for a particular institution is recorded. While these models exist at specification these rules have also been extended to support execution variations in Business Process Execution Language.

5 Conclusion

The starting premise of this paper was the importance of the role of conceptual and reference models to the IS development. Such models are one of the primary means of communication thus an effective evaluation of these artifacts is likely to be critical to the success of an IS implementation. By building on work in the use of structuralism applied to conceptual models, the paper has identified the role of variability modeling for reference models which is an example of the context of the domain being inserted into the conceptual model. Additional areas of evaluation for example the extent of community acceptance have also been identified (although not addressed in detail here) as important. While case studies are an important research tool, they do not permit a ready generalization so it is important to collect further results. As part of continued work in the area, the domain of course management and the artifacts produced so far, there is ongoing engagement with the stakeholder community to review, enhance and further experiment with reference models. A follow up project is part of an action research approach to re-interpret the function aspects of the reference model (the core business process of course validation) as an implementation using a community edition of a business process management toolset.

References

[1] Barn, B. S. and Oussena, S. (2008): BPMN, Toolsets and Methodology: A case study of business process management in Higher Education. In Information Systems Development 2008, Springer. To appear.

[2] Barn, B.S., Dexter, H., Oussena, S. Petch, J.: An Approach to Creating Reference Models for SOA from Multiple Processes. In: IADIS Conference on Applied Computing, Spain 2006.

[3] Becker J, Knackstedt R, Fleischer S, Stein A (2008) Ontology Support for Configurative Reference Modeling. In 16th European Conference on Information Systems (Golden W, Acton T, Conboy K, van der Heijden H, Tuunainen VK eds.), 1668-1679, Galway, Ireland. (ISBN 978-0-9553159-2-3)

[4] BPMI.org, OMG: Business Process Modeling Notation Specification. Final Adopted Specification. Object Management Group (2006), available at: http://www.bpmn.org

[5] Dobing, B. and Parsons, J. (2000) Understanding the role of Use Cases in UML: A review and research agenda. *Journal of Database Management* 11, 4, 28–36

[6] Dobing, B. and Parsons, J. (2006) How UML is used. *Communications of the ACM*, May 2006. Vol.49, No. 5.

[7] Esswein W., Zumpe DKS., Sunke N. (2004) Identifying the Quality of Ecommerce Reference Models, *Proceedings of the 6th international conference on Electronic Commerce*, Delft, The Netherlands

[8] Frankel, D (2005) Business Process Trends. http://www.bptrends.com/publicationfiles/07%2D05%20COL%20BP%20 Platform%20%2D%20Frankel%2Epdf

[9] Jacobsen, I., Griss, M., and Jonsson, P. (1997) “Software Reuse: Architecture, Process and Organisationn for Business Success” Addison-Wesley, USA.

[10] Lauesen, S. and Vinter, O. (2000). Preventing Requirement Defects. In Proceedings of the Sixth International Workshop on Requirements Engineering: Foundations of Software Quality (REFSQ 2000), Stockholm.

[11] Mellor, S.J., Scott, K., Uhl, A. and Wiese, D. (2004) MDA Distilled. Addison Wesley

[12] Moody, D. L. (2005) Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering* 55, 243-276.

[13] Object Management Group (2003) The Unified Modeling Language http://www.omg.org/technology /documents/formal/uml.htm

[14] Pask, G. *Conversation, Cognition and Learning*. (New York: Elsevier,1975).

[15] Pfeiffer D, Niehaves B (2005) Evaluation of Conceptual Models - A Structuralist Approach. In Proceedings of the Thirteenth European Conference on Information Systems (Bartmann D, Rajola F, Kallinikos J, Avison D, Winter R, Ein-Dor P, Becker J, Bodendorf F, Weinhardt C eds.), 459-470, Regensburg, Germany. (ISBN 3-937195-09-2)

[16] Reich, B. H. and Benbasat, I. (2000). Factors That Influence the Social Dimension of Alignment between Business and Information Technology Objectives. MIS Quarterly, 24 (1), 81-113.

[17] Ribbert M, Niehaves B, Dreiling A, Holten R (2004) An Epistemological Foundation of Conceptual Modeling. In Proceedings of the Twelfth European Conference on Information Systems (Leino T, Saarinen T, Klein S eds.), 1557-1568, Turku School of Economics and Business Administration, Turku, Finland. (ISBN 951-564-192-6)

[18] Standish Group, The CHAOS Report, in: The Standish Group International, Available on-line at http:// www.standishgroup.com/sample_research/chaos_1994_1.php, 1994.

[19] Thomas, O (2006) Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation, *Lecture Notes in Computer Science* ,Springer-Verlag

[20] Themistocleous, M., Irani, Z., O'Keefe, R.M. (2001), "ERP and application integration", *Business Process Management Journal*, Vol. 7 No.3, pp.195-204.

[21] Thomas, O (2006) Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation, *Lecture Notes in Computer Science* ,Springer-Verlag

[22] Y. Wand, R.A. Weber, Research commentary: information systems and conceptual modelling—a research agenda, Information Systems Research 13 (4) (2002) 363–376.