

From Concept Models to Conceptual Data Models

Bodil Nistrup Madsen & Anna Elisabeth Odgaard
DANTERMcentret, Copenhagen Business School

Keywords: concept model, ontology, data model, database, knowledge modelling

In order to develop a harmonised and efficient IT system, such as a database, it is important to be familiar with the underlying concept model (concept systems) for the relevant domain which the IT system should be designed to accommodate, as this forms the necessary firm foundation for designing the conceptual data model. Although there is no one-to-one correlation between concept and characteristic features in the concept model and classes and attributes in the conceptual data model, there are many similarities between concept modelling and conceptual data modelling, and by closely examining the relationship between the two models, we have strived to construct an algorithm for creating conceptual data models in Unified Modelling Language (UML) on the basis of concept models that adhere to the traditional principles and methods of terminology work.

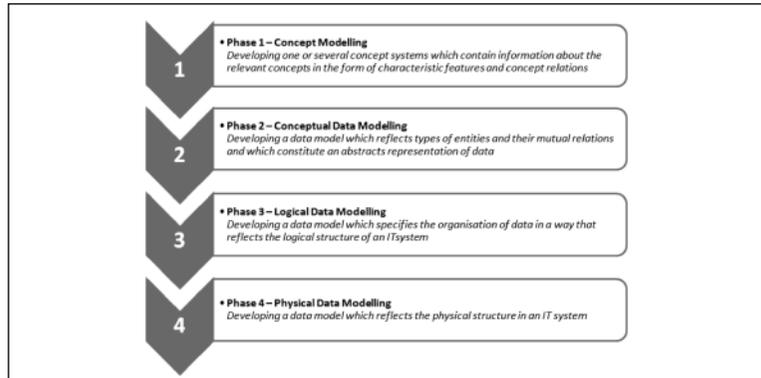
The Four Modelling Phases

Ideally, the modelling procedure for developing an IT system that comprises a database should consist of four phases: Concept modelling, conceptual data modelling, logical data modelling and physical data modelling, cf. Figure 1.

Unfortunately, the concept modelling phase is often omitted and IT developers set out to design the conceptual data model first off. This causes problems as a conceptual data model, which is also inappropriately referred to as a semantic model, contains no information about meaning, rather, what *kind* of information should be recorded in the database. If concepts are not clarified and consensus regarding the content of concepts and usage of terms has not been reached, problems and misunderstandings could arise in connection with the data models. Therefore conceptual data models should be constructed on the basis of

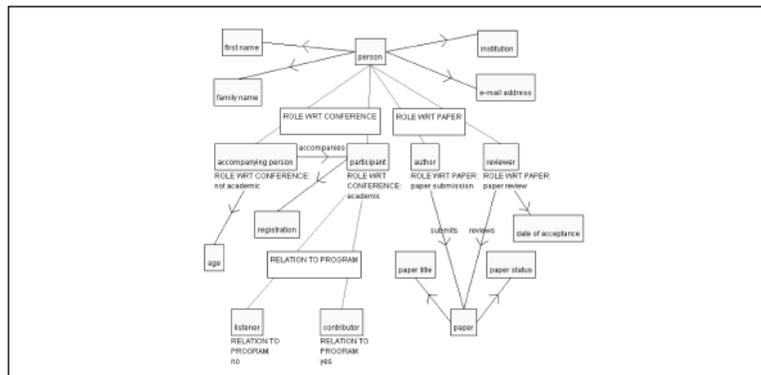
concept models and recent research in this area is already beginning to yield valuable findings (Kop 2008). However, we will go one step further as we set out to construct an algorithm for automatically generating conceptual data models on the basis of concept models.

Figure 1. The Four Modelling Phases



In our description of the four phases below we will use the design of a database for information about a conference system as an example, cf. Figure 2. This example has been inspired but not constrained by a case presented by Mathiassen et al, as we include a concept model and present a different data model (Mathiassen *et al.* 2001).

Figure 2. Extract of a concept model for a conference system



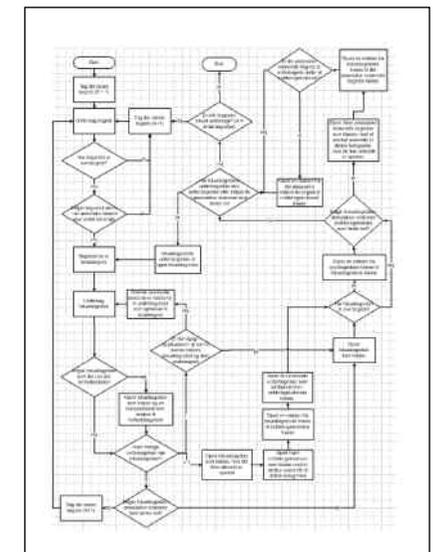
PHASE 1:

In the first phase of concept modelling, the domain experts draw upon their knowledge and understanding of the domain to identify the concepts and terms which are relevant for the proposed IT system. This information may also be extracted from requirement and design specifications. Assisted by terminologists or having the necessary terminological prerequisites, the domain experts then proceed to organise concepts visually in a concept model where the mutual semantic relations between the concepts are recognized and defined. Finally, the subdivision criteria and characteristic features are identified allowing for the production of short, simple and precise concept definitions.

PHASE 2:

Once a concept model has been produced, a conceptual data model in UML can be created on the basis of the conceptual information contained in the concept model in the form of characteristic features and concept relations. In this work, we have attempted to closely study and formalise to the extent possible, the many considerations and decisions of this conversion process. As a general trend, it can be observed that some concepts correspond to classes, attributes or attribute values while other concepts in the concept model are not to be found at all as elements in the data model. Furthermore, the relations between concepts in a concept model can differ from the associations in the corresponding conceptual data model.

Figure 3. Algorithm for creating a conceptual data model on the basis of a concept model

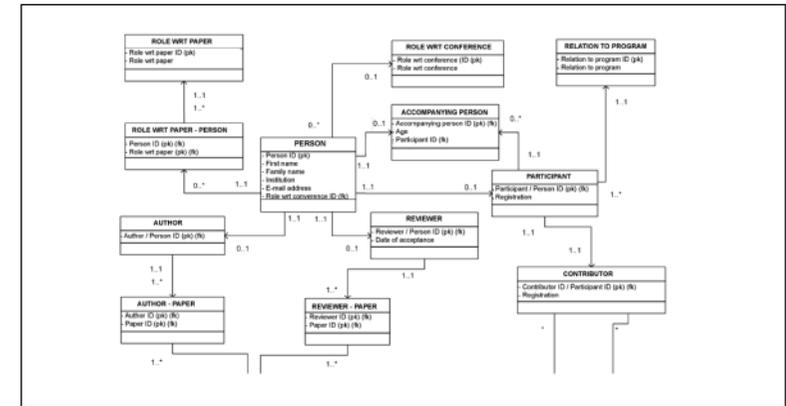


By applying the algorithm on a concept model, it would be possible to generate a conceptual data model in UML semi-automatically. However, this does imply that the concept model is complete, that concepts which are irrelevant to the IT system are identified manually, that multiplicity is specified manually and that any simplifications are carried out manually. Figure 3 shows the draft algorithm.

In order to move through the concept model and process all the concepts, subdivision criteria and relations in the appropriate way, the algorithm identifies the initial focus concepts which are characterized by not having any superordinate concepts nor do they solely enter into an associative relation as the target concept. Subsequently, a series of properties are analyzed to establish which role the focus concept, relevant subdivision criteria and relations should play in the conceptual model. For instance, if a focus concept has several subordinate concepts, then the focus concept is converted to a class in UML and each subdivision criteria is similarly converted to a class with an attribute corresponding to the name of the class. In addition, a “has” relation is created from the class of the focus concept to the class of the subdivision criteria, and the associated subordinate concepts become attribute values in the class of the subdivision criteria.

Another part of the algorithm accounts for the measures to be taken when the focus concept enters into associative relations as the source concept. In this case, the focus concept is converted to a class, if it has not been converted already, and the associatively related concepts become classes as well with attributes corresponding to the names of the classes. Lastly, a “has” relation is created from the class of the focus concept to the associatively related classes. Another rule in the algorithm states that if a subconcept of a focus concept has subconcepts itself or enters into associative relations as the source concept, then this subconcept should be treated as a focus concept too. The algorithm is thus repeatedly traversed until all concepts have been processed. Figure 4 shows an example of a conceptual data model generated on the basis of the concept model in Figure 2.

Figure 4. Extract of a conceptual data model generated on the basis of the concept model

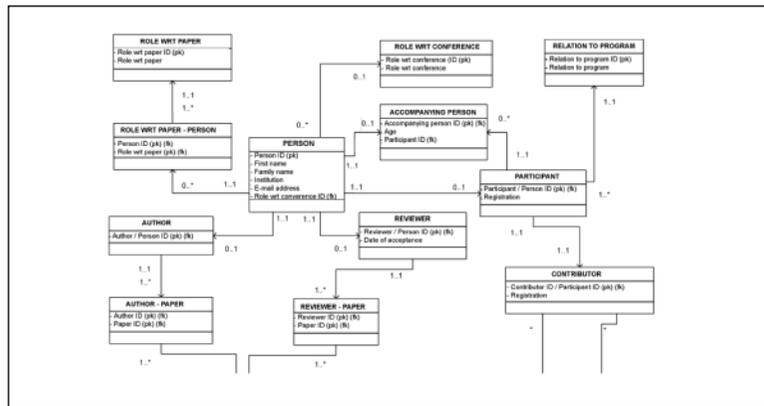


Next, each relation in the new intermediate model will have to be carefully scrutinized to determine multiplicity. The domain experts and IT developers will also be able to make simplifications in order to facilitate implementation or cater for specific user requirements, before the conceptual model can be considered actually complete and ready for phase 3.

PHASE 3:

Concerning the logical data modelling phase, principles and recommendations for converting a conceptual data model to a logical data model already exist (Connelly and Begg 2004) (Applied Information Science 1997). During this conversion, all classes are provided with a primary key corresponding to the name of the class joined with the identifier string “ID” and the relevant attribute is given a (pk) marking. Many-to-many relations transformed into a new intermediate class and two one-to-many relations from the two original classes to the new intermediate class are created. All classes which take part in a one-to-many relation are joined by a primary key (pk) on the “one” side and a foreign key (fk) on the “many” side. An example of a logical model, which is created on the basis of the conceptual data model in figure 4, is given in Figure 5.

Figure 5. Extract of a logical data model created on the basis of the conceptual data model



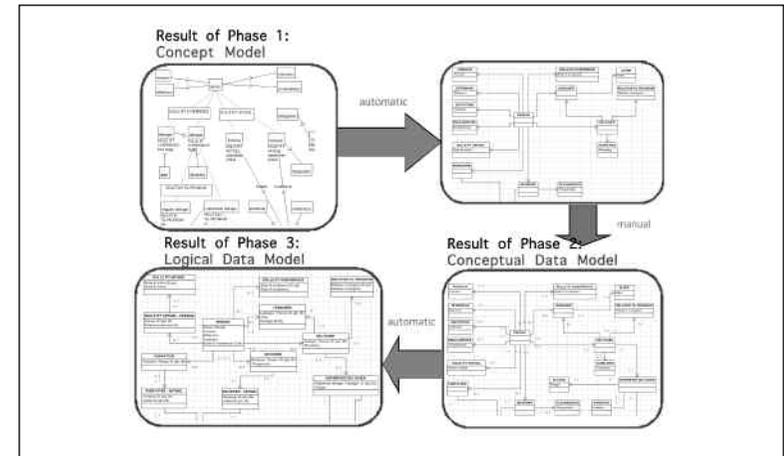
PHASE 4:

Implementing a physical data model in the last phase calls for considerations regarding the facilities and constraints of a given schema and query language, such as SQL or XML, including technical specifications, such as data types or the sequence order of elements. Once this has been established, the database can be implemented. Figure 6 shows an overview of the phases 1 to 3: from concept model to logical data model.

Considerable Potential

Formalising the crucial step of converting a concept model to a conceptual data model has clear benefits and considerable potential, especially in the current age of digitalization where the necessity of organising and clarifying concepts as a basis for data modelling has become apparent. By providing guidelines for converting a concept model to a data model, we hope to pave the way for consistently anchoring data models in concept models, implementing a mediating framework for a dialogue between domain experts, terminologists and IT developers, and reducing resource costs by avoiding errors, ambiguity and vagueness during IT system development.

Figure 6. Overview of the phases 1 to 3: from concept model to logical data model



Future Work

In future work, we plan to investigate the relationship between concept models and data models even further and apply the algorithm to a wide range of concept models and subsequently potentially extend and improve the algorithm. An interesting direction for future research in this area would also be to devise a methodological workflow for progressing through the four modelling phases and specify the necessary skills and competencies for the various profiles involved both directly and indirectly in IT system development.

Authors' Details

Bodil Nistrup Madsen (bnm.danterm@cbs.dk) & Anna Elisabeth Odgaard (aao.danterm@cbs.dk)
 DANTERMcentret, Copenhagen Business School, Dalgas Have 15,
 DK-2000 Frederiksberg

References

Ambler, S. (2004) *The Object Primer 3rd Edition - Agile Model-Driven Development with UML 2.0*, New York: Cambridge University Press. Available online at

[<http://www.agiledata.org/essays/dataModeling101.html>] (accessed 20 August 2009).

Applied Information Science (1997) *Conceptual - Logical - Physical Models*, Available online at [<http://www.aisintl.com/case/CDM-PDM.html>] (accessed 20th August 2009).

Connolly, T. and Begg, C. (2004) *Database Systems 4.ed - A Practical Approach to Design, Implementation and Management*, Pearson Education Limited.

Kop, C. (2008) 'Conceptual modelling tool for novice designers', *International Journal of Metadata, Semantics and Ontologies*, 3(2) 151-165.

Mathiassen, L. Munk-Madsen, A., Nielsen, P.A., and Stage, J. (1998) *Objekt-Orienteret Analyse og Design*, Forlaget Marko Aps, Aalborg.

Yeung, A. (1998) Unit 051 - *Information Organization and Data Structure*, Available online at [<http://www.ncgia.ucsb.edu/giscc/units/u051/u051.html>] (accessed 20 August 2009).