Working Paper Series
ISSN 1170-487X

Teaching Novice Conceptual Data Modellers to Become Experts

by John R. Venable

Working Paper 95/28

September 1995

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Teaching Novice Conceptual Data Modellers to Become Experts

John R. Venable

Department of Computer Science
University of Waikato
Hamilton, New Zealand
email: jvenable@cs.waikato.ac.nz

Abstract

This paper describes teaching practices designed to help novice data modellers become expert data modellers. We base these practices on extant empirical research which highlights the strengths of expert data modellers and reveals the weaknesses of novices. After reviewing this research and analysing the causes of the novices' difficulties, we describe a strategy and specific techniques for helping novices to overcome their weaknesses and acquire the strengths and skills of expert data modellers. Techniques recommended include explicit comparison and teaching of novice and expert characteristics and behaviours, providing students with a realistic plan for how to acquire expert data modellers' capabilities, exposure to and comparison of a wide variety of data modelling approaches and topics, extensive amounts of practice on a wide variety of application domains, and critique of practical work in light of the understanding of novice errors and expert behaviours. Our intent is not just to make significant progress during a course, but to provide students with a means to continue to learn and improve in the long term.

1. Introduction

This paper describes an overall strategy and specific techniques for teaching novice data modellers to become expert data modellers. The strategy and techniques are intended to be applied during a second course on data modelling (i.e., a course beyond introductory courses to database and systems analysis and design). The techniques we discuss here are being applied primarily to the task of *conceptual* data modelling, rather than to later data modelling tasks, such as logical or physical database design, or to other areas of system development, such as process modelling or system design. However, we believe the overall approach is suitable for obtaining expert skills relevant to those other systems development areas.

Conceptual modelling is "the process of formally specifying the data and processing requirements of an information system. ... This model is independent of how the information system is to be constructed." [Moody et al., 1995] Conceptual data modelling is a sub-area concerned with modelling the static structure of the problem domain, usually reflecting the things about which we need to store information. Conceptual data models often serve as input to database design activities.

Conceptual modelling, including conceptual data modelling is very important. Boehm [1987] has estimated that the cost of making changes during system development (including fixing errors) increases by a factor of 10 at each succeeding phase of the development process. E.g., fixing an error at the implementation stage that occurred at an earlier stage will cost approximately 10 times what it would have cost to fix it at the design stage and 100 times what it would have cost to fix at the systems analysis stage. Moreover, Martin [1989] has conducted empirical research that shows that more than half the errors in systems development result from inaccurately defined requirements, of which the conceptual data model is an important part. Therefore, from both the cost and error likelihood points of view, the proper conduct of the conceptual data modelling task is very important.

What makes someone an expert is rather imprecise and difficult to say. Experts in a particular domain generally have substantial training and/or practical experience, have developed excellent skills in their domain, and generally perform better. In the conceptual data modelling domain, empirical research has shown that expert data modellers perform significantly better than novice data modellers [Batra and Davis, 1992; Chaiyasut and Shanks, 1994]. They develop data models that are more complete. Their models provide for future expansion, thereby lowering maintenance costs. Their models contain fewer errors. Therefore, there is strong motivation to teach novice data modellers to acquire expert data modelling skills - and as quickly as possible.

Novices in a particular domain have received some training, but have little-to-no practical experience and poor skills. For the purposes of the teaching practice recommendations in this paper, novice data modellers have generally had a typical course in systems analysis and design, including coverage of entity-relationship modelling and elementary relational database design, a typical computer science database course, and a project course in which they apply their knowledge to developing an information system for a real-world situation.

At the University of *** (unnamed for review), we teach an advanced level course which includes a significant component on data modelling. One objective is to help our students to progress from the level of novice data modeller to a more advanced level. We don't expect our students to become experts just by taking one course. However, we do expect them to make a significant advance in their skills and to have the foundation and means to become expert data modellers in the near-term future.

The remainder of this paper describes our teaching strategy for this course and the rationale behind it. The next section summarises empirical research into novice and expert data modeller characteristics and behaviours. Section 3 analyses those characteristics to identify the root causes and problems to be overcome by novices who want to become expert data modellers. Section 4 presents our strategy and specific techniques for overcoming those problems and encouraging students to acquire expert data modelling skills. We conclude with a summary of our findings and recommendations for future research.

2. Expert vs. Novice Data Modellers

As noted in the introduction, experts perform significantly better than novices at conceptual data modelling. This is a good thing, as our experience is that novice data modellers generally produce data models of unacceptably poor quality. This section summarises existing empirical research on novice and expert conceptual data modellers [Batra and Davis, 1992; Chaiyasut and Shanks, 1994; Batra and Antony, 1994]. We identify the characteristics that contribute to the improved performance of experts or conversely hinder the performance of novices. Some of the factors that contribute to improved conceptual data modelling performance by experts include greater referent experience, greater understanding of data modelling concepts, different goal structures, different emphases on data modelling levels and activities, and better use of heuristics.

2.1 Greater referent experience

An obvious difference between experts and novices is that experts typically have significantly more experience actually *doing* conceptual data modelling than novices. Experts typically have acquired much of their expertise over years of practice at data modelling. They have gained experience with a broad cross-section of application domains. Experience in a number of application domains leads to experts possessing a sort of library of generalised conceptual data models, upon which they then draw for reuse when modelling data [Chaiyasut and Shanks, 1994]. Experts recognise analogous situations, then adapt these pre-stored general models to the specific situation, using them as a sort of template. In their study, Chaiyasut and Shanks [1994] observed that experts spent 6.2% of their time recognising and reusing experience, while novices, lacking the requisite referent experience, spent 0% on reuse.

2.2 Greater understanding of data modelling concepts

Experts often have completed degree programs with multiple courses in data modelling. This gives them both more training on data modelling concepts and more experience using them in courses. Certainly, this will be the case for graduates of our degree program. Moreover, experts typically also have extensive experience using the data modelling constructs at work. This means that experts do not have to think much about which constructs to use or how to put them together.

2.3 Different goal structures

Novices' goals are typically rather narrow, revolving around capturing the specific semantics of the problem description into the conceptual data model. Typically, they do not relate the various parts of the problem domain to other parts [Batra and Antony]. Instead, they consider the parts of the problem description in isolation from each other. This leads to sub-optimisations in the modelling process and to literal translations without an adequate understanding of the problem domain.

Experts, on the other hand, have several goals which lead to better data models. First, they have a goal of developing a holistic understanding of the problem domain [Batra and Davis, 1992; Chaiyasut and Shanks, 1994]. The conceptual data model that they develop must then be consistent and faithful to that understanding. All of the parts of the conceptual data model must be coherent with the other parts of the model. Secondly, experts try to be sure that the data model supports all of the system requirements (information needs) and goals identified in the problem domain [Chaiyasut and Shanks, 1994]. They use this as a way of extending and verifying the data model. Third, experts have the goal of creating a data model that takes future requirements into account, by anticipating likely changes and allowing for organisational growth.

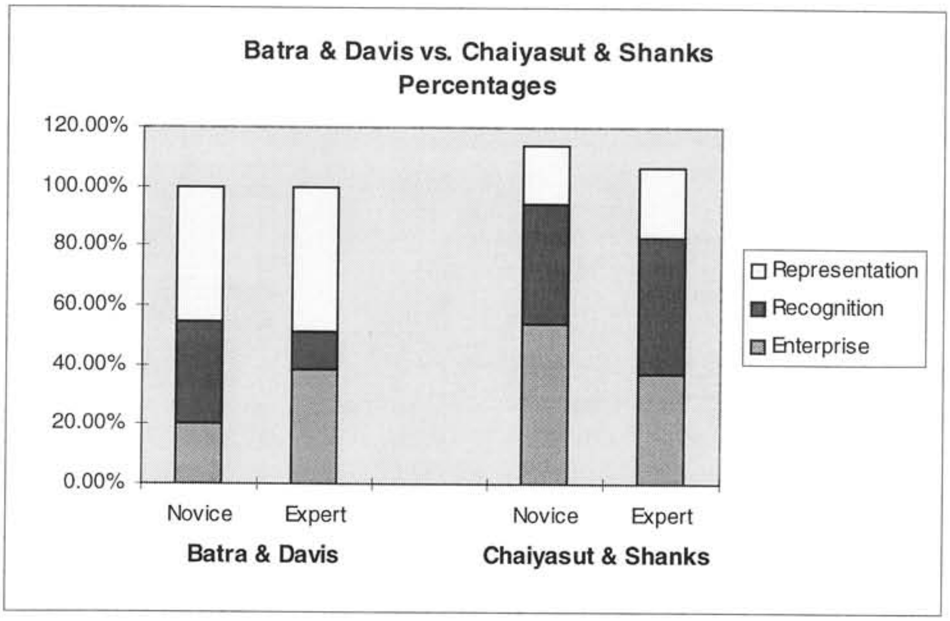

Chart 1: Comparison of percentages

2.4 Different emphases on data modelling levels and activities

Studies by Batra and Davis [1992] and Chaiyasut and Shanks [1994] have clearly shown that experts and novices spend substantially different percentages of the time they spend data modelling at different levels of the task. But, *how* are they different? Unfortunately, many of the results are conflicting. Batra and Davis analysed the percentage of time that novices and experts spent at 3 different levels: enterprise, recognition, and representation. Chaiyasut and Shanks analysed the percentages of time novices and experts spent on 6 different classes of activities: understanding, searching for solutions, representing information, recognising goals, reusing , and planning. The first three of these activities are roughly equivalent respectively to the three levels of Batra and Davis. Recognising goals additionally corresponds to the enterprise level, so here we take understanding and recognising goals together as roughly equivalent to work at the enterprise level. chart 1 compares the two studies' results, based on the data contained in these two papers.

In chart 1, note the complete reversal between the two studies of the percentages at the enterprise and recognition levels. Batra and Davis found a significantly higher percentage of the time for experts than for novices at the enterprise level, while Chaiyasut and Shanks found the reverse. How to explain this? One explanation is that in Chaiyasut and Shanks' study, activities could be categorised into more than one group at a time. Indeed their time allocations total 117.83% for novices and 116.38% for experts. However, this is insufficient to explain the differences. The activities (behaviour categories) in Chaiyasut and Shanks also do not correspond exactly to the levels in Batra and Davis, but they are still close enough that there should not be such a difference. A third possibility is that the differences in the experimental task performed by the research subjects or the administration of the experiments were sufficient to cause different behaviours. For example, it may be that the task in Chaiyasut and Shanks' study had more complexities or more of a design component, therefore requiring more time searching for appropriate data modelling solutions. Due to the small sizes of both studies, individual participant differences is another possibility.

However, we can also explain at least *some* of the differences in percentages of time by instead comparing at the *amount of time* spent by novices and experts on each level or behaviour. By multiplying the average percentages against the total average time for each group, we instead get the information shown in chart 2.

In chart 2, the relative amounts of time spent that novices and experts spent at the enterprise level in the two studies are more in accordance. In the Batra and Davis study, experts spent 62.3% more time than novices at the enterprise level. In the Chaiyasut and Shanks study, experts spent only 23.5% more time than novices on recognising goals and understanding the problem domain. While the amounts are different, we can now conclude that the *amount of time* spent by experts is *greater* than that spent by novices. This is consistent with the conclusion of both studies that experts try to develop a holistic understanding of the problem domain (while novices don't).

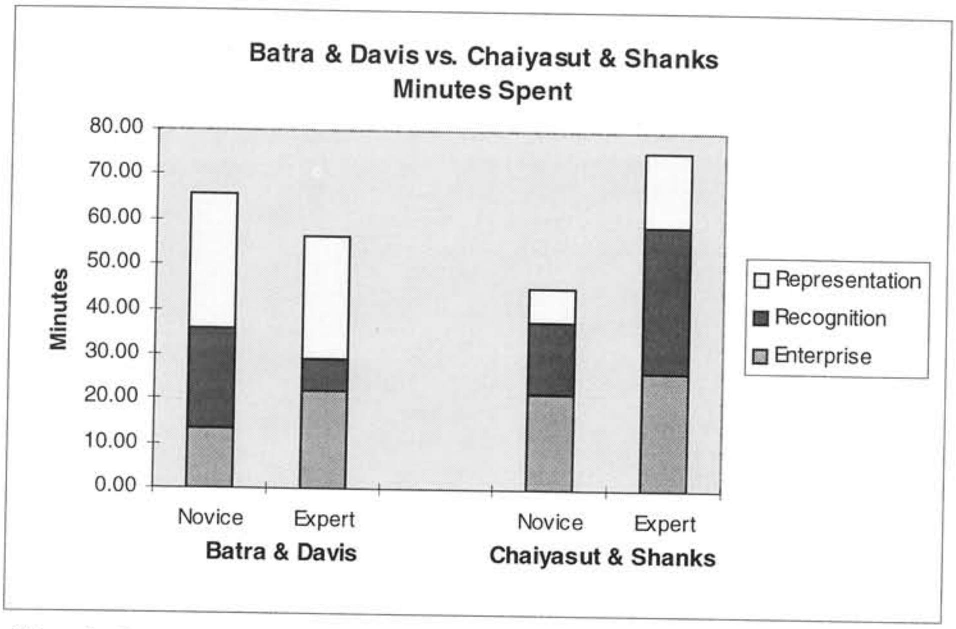

Chart 2: Comparison of total mintues spent

Clearly, though, more studies need to be made to clarify this and other information. For example, the differences between experts and novices in total time and time spent on recognition are still reversed in the two studies.

2.5 Better use of heuristics

Experts generally make effective use of high level heuristics to guide the conduct of the conceptual data modelling process. Novices, on the other hand, tend to use fairly low level data modelling heuristics, such as translating nouns to entities and verbs to relationships. [Chaiyasut and Shanks, 1994]. This mechanical translation without understanding leads to problems such as "literal translation" [Batra and Antony, 1994].

When novices attempt to use higher level heuristics, they often misuse them, leading to "biases" [Batra and Antony, 1994]. One such bias is that of anchoring. Anchoring is an appropriate high-level heuristic used by experts in which an initial conceptual data model is developed and then refined. Novices, however, tend to be reluctant to make changes. They are unwilling or unable to modify their initial data models or to discover flaws in them. Therefore, their initial models become fixed, even if the initial model contains errors. Any additions to them must be adapted to them, errors and all, which leads to even more errors. Other aspects contributing to novice biases include data saturation (a sort of information overload), availability (what is easily available in the problem description gets incorporated into the model, even if inappropriate), order effects (aspects of the problem description that are encountered close to each other in time are related while those encountered separately are not), and outcome irrelevant learning systems (lack of feedback on model quality - i.e. no means to evaluate the model) [Batra and Antony, 1994].

3. Analysis of Difficulties that Novices Encounter

In this section, we analyse the differences described above to identify the root causes of the difficulties that novices encounter in the conceptual data modelling process.

3.1 Lack of referent experience

The most obvious difference is that novice users have only been exposed to a small number of different kinds of problem domains (or Universes of Discourse). Additionally, the problem domains that they have encountered are usually contrived, as well as being smaller and simplistic when compared with real-world problem domains. This is the root cause for their inability to reuse and their difficulty in understanding problem domains. They have little experience to guide them and little or no referent experience to recognise and to which they can draw analogies. The lack of referent experience means that novices are more likely to misunderstand the situation, make incorrect hypotheses and assumptions about the problem domain, treat the situation literally - rather than sensibly from their understanding, and to spend more time on trying to comprehend the organisational situation, detracting from the time they have to perform other data modelling tasks.

3.2 Poor understanding of data modelling constructs

Unfamiliarity and lack of understanding of data modelling constructs, both from lack of training and lack of real-world data modelling experience, leads to a number of problems dealing with translations of the novice's understanding of the problem domain to a conceptual data model representation. These problems include misuse or even nonsensical use of data modelling constructs, inappropriate correspondence of constructs to the problem domain, incorrect statement of the novice's understanding, and use of only a small subset of the conceptual data modelling language's constructs (e.g., not using higher arity relationships or generalisation structures) [Batra and Antony, 1994].

3.3 Lack of understanding of appropriate heuristics and processes

Novices' use of heuristics is naive and low level [Batra and Antony, 1994]. They believe that using these simple heuristics is appropriate and non-problematic. They have generally been unexposed to the higher level sorts of heuristics (or even data modelling processes) used by experts and hence have little or no understanding of them. Consequently, heuristics such as divide and conquer or anchoring (together with incremental adjustment) are improperly applied.

3.4 Work at too detailed a level

We have observed that novice data modellers tend to work at a detailed level too much of the time. Their use of low level heuristics tends to make them work at a detailed level. They also tend to work on only one area at a time without relating it to other areas. They don't take the time - or don't know to take the time - to step back and reflect on the larger problem, such as the overall goals of the system. Similarly, they rarely relate their data models to the system's processing characteristics.

3.5 Poor ability to exercise quality control over their own models

There are three primary factors that are basic causes of this problem.

First and foremost, the practice of conceptual data modelling, especially in isolation from other system development activities, is hindered by *outcome irrelevant learning systems* [Batra and Antony, 1994]. This means that the conceptual data modeller does not receive feedback on his or her resulting conceptual data model. Typically, such feedback would come from the system users, from whom the problem description is obtained, and from system designers, who must understand and use the resulting conceptual data model. This feedback is often delayed or never reaches the data modeller. Expert data modellers deal with this by ensuring that they understand the problem domain thoroughly, by seeking such feedback after modelling, and/or by exercising quality control over their models in other ways. In the case of conceptual data models developed for instructional purposes (e.g. a course assignment), the feedback comes from instructors or their agents.

Secondly, novice conceptual data modellers have *no practical goal of quality control.* This is reflected in the absence of such activity in empirical studies of novice data modellers [Batra and Davis, 1992; Chaiyasut and Shanks, 1994]. Novices have only rarely been taught that exercising of quality control is an important goal. They believe that arriving at an initial solution, having applied their (naive) heuristics is sufficient for creating an adequate (if not optimal) conceptual data model.

Third, novices *lack knowledge about appropriate quality control techniques.* They do not know to seek feedback from outside sources, such as users, system designers, or their peers. They also do not know appropriate techniques for internal quality control, such as simulating the system's ability to meet all system requirements from the information modelled. They often do not even know how to apply basic quality control heuristics, such as examining the conceptual data model for completeness (e.g. missing cardinalities), meaningful names, synonyms (where illegal), and homonyms.

3.6 Rush to closure and avoidance behaviours

Finally, we believe from our experience that novice data modellers often suffer from a problem that is not discussed in the literature. When beginning data modelling, novices are confronted with a difficult and unpleasant task. They are asked to use constructs that they don't really understand to describe problem domains that they don't have the necessary background to understand. They feel that they look stupid when they are forced to ask questions about aspects of the problem domain that the users seem to think are obvious (and to the users, they *are* obvious). The situation forces them to work in an environment with a great deal of uncertainty, which naturally tends to make them uncomfortable. Additionally, they may view such exercises as pure busywork rather than something that they actually learn from. The negative aspects of data modelling lead quite naturally to a strong desire to be finished with the unpleasant process (rush to closure) and to avoid doing parts of it, such as raising questions with users or examining their own models for correctness. We believe that these tendencies are reflected in many of the characteristics described above.

4. Strategies and Techniques for Overcoming Novice Difficulties

This section describes the general strategies and specific techniques that we use in a high level (fourth year) course that includes a significant component (approximately 1/2) on advanced conceptual data modelling. As mentioned in the introduction, this course follows an introductory database course, as well as a systems analysis and design course and a project course, i.e. the students have reached the novice level as data modellers. While the techniques are applied in the context of our course, our goal is to provide a sustainable basis for continued learning and acquisition of expert data modeller characteristics following course completion and transition to the work force.

4.1 Teach and compare novice and expert characteristics

Our experience has shown that novices benefit from learning about the differences between novice and expert data modellers, as well as specific kinds of errors that novices make and weaknesses to which they are prone. First, it makes them understand that there really is a difference. Second, it gives them something to strive for. Third, students come to realise the limitations of the lower level heuristics they might otherwise rely blindly on. Fourth, they become aware of expert goals and behaviours that they can seek to learn and adopt.

One technique that we use in teaching novices to become expert data modellers is to expose them to the literature on this topic. Early in the course, students are required to read the main empirical research articles cited in this paper and are assigned to write a 3-5 page paper integrating and summarising the findings of those articles. Additionally, the articles are discussed in class and the students' papers are carefully marked. Doing so ensures that the students realise that the instructors consider the material to be important, pay attention to it, and have an understanding of it.

4.2 Provide students with a realistic plan for acquiring experts' capabilities

One reason we spend so much time on expert characteristics is that we don't want to be simplistic about it. Otherwise, there is a tendency for the students to say "Of course experts are better, they have years of experience. We just have to wait until we have years of experience. Nothing else can be done." In our experience, this is not true. Worse, it is counterproductive.

Therefore, besides making our students aware of the differences between themselves and experts (as described above), we try to operationalise a method for the students to acquire experts' skills. Students should be given a plan for how to acquire expert characteristics, both in the context of the course and beyond the course. Part of the plan is already incorporated in being made aware of experts' characteristics. The remainder of the plan incorporates four main goals and strategies for reaching each of them (we describe them here in the imperative voice, as we would communicate them to our students).

> "First, become very familiar with data modelling constructs. Develop an intimate understanding of them so that they become second nature. Reading examples and practising are both very helpful for this. Also, you can use data modelling as a tool for understanding problem domains that you encounter in your day-to-day life.
>
> "Second, become very familiar with and adopt an expert's process for data modelling. Remember that experts always make sure that they understand the problem domain correctly. Where they can do this more quickly, you will have to make special efforts. Don't make assumptions about the problem domain; ask questions about it. Remember also that experts carefully verify their models, by checking whether they capture the information necessary to meet system requirements. The best way to remember these things at first is to review expert characteristics while practising - and to practice developing your own models whenever possible.
>
> "Third, gain exposure to a wide variety of different application domains. Consciously

begin to abstract common patterns out of them. You don't have to wait for the application domain to come to you in the course of your studies or work. You can model things that you read about in other courses. You can read different examples in data modelling textbooks.

"Fourth, after performing data modelling, review your data models and the way that you developed them, and critique them in light of what you know about experts and novices. This will tell you something about how well you are progressing toward becoming an expert data modeller. For example, did you rely heavily on lower-level heuristics? Were you uncertain whether to use binary, ternary, or higher level relationships? In that case, you probably aren't familiar enough with the data modelling concepts. Does the resulting data model make unwarranted assumptions or include misunderstandings of the problem domain? Are you uncertain that you have modelled the domain correctly? If so, you haven't concentrated sufficiently on understanding the problem domain. Are you uncertain that the system will provide the information needed? Then either you haven't satisfactorily established the system requirements, or you haven't verified that the data model accounts for them."

In the conduct of our course, before providing students with this plan, we additionally assign students to come up with their own plan, as part of the paper assigned above. We ask the question, "What can you do to become an expert data modeller?"

After reviewing the plan, we can then begin putting it into action within the context of the course, using additional techniques discussed in the following sections.

4.3 Expose students to many general data modelling approaches and specialised topics

In our course, we teach our students a number of different general data modelling approaches, as well as covering some specific topics or domains in data modelling.

In the general approaches covered, we begin with entity relationship (ER) modelling, as covered in their systems analysis and design course. This is extended to cover various forms of extended entity relationship (EER) modelling. We further extend their knowledge of data models by covering object-oriented modelling, as used in object-oriented analysis (OOA), which we see as very similar to EER, but further extending it. We also carefully cover fact-based or object-role models (ORM), specifically using NIAM [Nijssen and Halpin, 1989], as a contrasting approach. We do this partly to show a different view of data modelling, and partly to highlight interesting characteristics of fact-based modelling, such as lexical object types (LOTs), the use of populating diagrams with sample data, and NIAM's extensive and rigorous set of constraints.

As far as the topical areas of data modelling covered, we spend time discussing temporal data modelling, geo-spatial data modelling, and meta-modelling. These specialised areas highlight some of the difficulties encountered in data modelling and some potential solutions.

One of these topics -- meta-modelling -- deserves special mention. We have two objectives in using meta-modelling in our course. First, it is an exercise in using a conceptual data model (as a meta-modelling language). Second, it is a way of seeing how the concepts of different conceptual data modelling languages are related. For example, Venable [1994] has meta-modelled and integrated each of the conceptual data models covered in our class (i.e., EER, OOA, ORM/NIAM). Figure 1 shows a partial meta-model (created using the conceptual data modelling language CoCoA [Venable, 1994]). It shows how fact-based models (ORM) aggregate the primary concepts of entity (here called general entity type to distinguish it from ER models' entity concept) and label (here called identifier as in ER models) and how those concepts relate to other concepts (attribute, descriptor, and object types) in ER and O-O conceptual data models. Figure 1 shows only a portion of the meta-models covered. Other meta-models cover relationships, aggregation, etc.

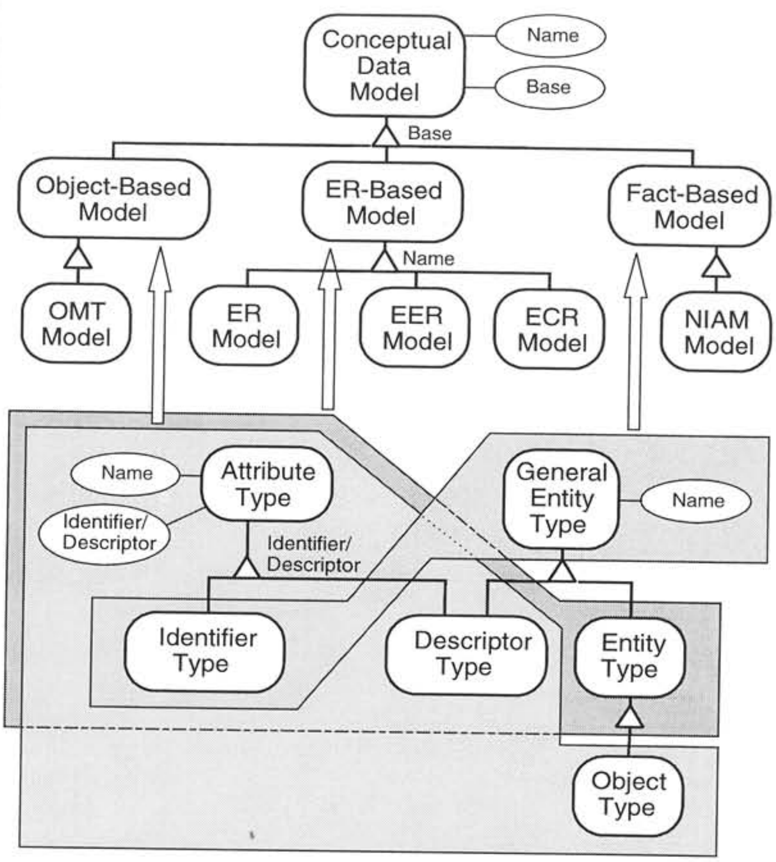

Figure 1: Example (partial) meta-model of conceptual data modelling languages

4.4 Practice modelling on a wide variety of application domains

In addition to the topical areas of data modelling, throughout the course, students are required to develop conceptual data models both in assignments and in small exercises to be discussed in class. The problem domains for these exercises and assignments (approximately 10) are deliberately chosen to give a broad cross section. The exercises require the use of all of the concepts taught and confront a number of classic data modelling problems. For example, we make sure that we assign problem domains that contain recursive relationships and ternary or higher level relationships, as well as generalisation and aggregation structures. Problems cover such areas as reservations, part assemblies, logical concepts with physical embodiments, such as films and videotapes of them, resource allocation across multiple consumers, etc., in addition to temporal, spatial, and meta-modelling. We also show students the generalised solutions (templates) behind each problem and show how such a template could be applied to similar problems. This is done in order to help the students begin to organise a set of template solutions in their own minds. Also, by discussing solutions in class, students can see their classmates' solutions and discuss the advantages and disadvantages of different solutions to the same problem.

4.5 Critique practical work based on novice and expert characteristics

When reviewing students' exercises or assignments, we try to relate that back to novice and expert characteristics. For example, where translation of requirements to a data model representation seems to have been literal (and naive), we mention this and relate it back to that novice characteristic and the expert solution, which is to develop a real (not naive) understanding of the problem domain. Where students have used binary relationships and a ternary relationship would have been better, we point out how this could occur and suggest that they review the rules for when to use ternary relationship.

5. Summary and Future Research

This paper has presented an overall strategy and a number of specific teaching techniques designed to facilitate novices becoming expert conceptual data modellers. It builds upon existing empirical research on expert and novice conceptual data modelling. This paper has reviewed, summarised, and integrated (where possible) these empirical findings. Further, we have analysed the existing findings to identify specific novice deficiencies that need to be addressed in teaching novices to become experts at conceptual data modelling. Finally, we have presented a coherent program that attempts to provide students with an adequate background and long-term plan for becoming expert conceptual data modellers. Our strategy is to (1) study the specific characteristics of novices and experts, so that the students understand the nature of the gap between themselves and experts, and (2) provide students with the means for overcoming the gap. Specific techniques described include encouraging students to adopt specific expert behaviours and goals, studying and comparing a wide variety of conceptual data modelling techniques, extensive practice using various techniques, emphasising practice on the more difficult conceptual data modelling constructs, and exposing students to many different problem domains and classical data modelling problems.

Further research is needed to validate, improve, and extend this approach. First, the existing empirical work on which this approach is based is still very incomplete and somewhat inconsistent. More research is needed on expert and novice conceptual data modelling, as well as on how novices become experts. Furthermore, although our subjective evaluation of the improvement of our students is very positive, we need careful empirical research to validate this sort of approach. We are only making the beginnings of this research and do not yet have results to report here. We also need research to identify the best cases and application domains for novices to practice with while utilising this approach. For example, what are the main generic models that experts utilise and should we teach them explicitly to novices? If so, how?

Research is also needed on novices and experts in other areas of systems development (e.g. [Sutcliffe and Maiden, 1992] on systems analysis) and extending this approach to teaching in those areas.

Finally, research is needed on other approaches to helping students learn, such as computer-aided instruction. [Batra and Davis] point out the possibility of knowledge-based support tools. These could include such instruction. [Chaiyasut and Shanks] point out that CASE tools supporting conceptual data modelling could be enhanced with facilities to support specific expert behaviours.

Acknowledgements

Many thanks to Robert Barbour for very helpful comments and suggestions on this paper. Many thanks also to our students for their feedback on these techniques.

Bibliography

Batini, Carlo, Stefano Ceri, and Shamkant B. Navathe, *Conceptual Database Design: An Entity-Relationship Approach*, Benjamin/Cummings Publishing Company, Redwood City, California, 1992.

Batra, Dinesh and S.R. Antony, "Novice errors in conceptual database design," *European Journal of Information Systems*, 1994, Vol. 3, No. 1, pp. 57-69.

Batra, Dinesh and Joseph G. Davis, "Conceptual data modelling in database design: similarities and differences between expert and novice designers," *International Journal of Man-Machine Studies*, Vol. 37, 1992, pp. 83-101.

Boehm, Barry, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981.

Chaiyasut, Piengjit and Graeme Shanks, "Conceptual Data Modelling Process: A Study of Novice and Expert Data Modellers," in *Proceedings: First International Conference on Object-Role Modelling*, 4-6 July 1994, Terry Halpin and Robert Meersman (editors), Key Centre for Software Technology, The University of Queensland, Brisbane, Queensland, Australia, 1994, pp. 310-323.

Martin, James, *Strategic Data Planning Methodologies*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

Moody, Daniel, Graeme Simsion, Graeme Shanks, Nancy Olson, and John Venable, "Stakeholder Perspectives in Conceptual Modelling," in *ACIS'96, Proceedings of the 6th Australasian Conference on Information Systems* (forthcoming), Curtin University of Technology, Perth, Western Australia, 1995.

Sutcliffe, A. G. and N. A. M. Maiden, "Analysing the novice analyst: cognitive models in software engineering," *International Journal of Man-Machine Studies.* Vol. 36, 1992, pp. 719-740.

Venable, John R., *CoCoA: A Conceptual Data Model for Complex Problem Domains*, PhD Dissertation, Binghamton University, Binghamton, New York, 1994.