A Decomposition Method for Entity Relationship Models: A Systems Theoretic Approach

Daniel L. Moody

Department of Information Systems,
University of Melbourne, Australia
d.moody@dis.unimelb.edu.au

Andrew R. Flitman
School of Business Systems,
Monash University, Australia
email: a.flitman@infotech.monash.edu.au

ABSTRACT. *This paper defines a method for decomposing a large Entity Relationship model into a hierarchy of models of manageable size. The purpose of this is to improve user understanding and simplify documentation and maintenance. We define the problem as an instance of the general systems decomposition problem or systems simplification problem. We first define a set of principles for decomposing Entity Relationship models based on systems theory and human information processing. These define the desirable characteristics of a decomposition, and may be used to evaluate the quality of a decomposition and to choose between alternatives. We then define a procedure which can be used by humans to develop a relatively optimal ("good") decomposition. Finally, we define a genetic algorithm which automatically finds an optimal decomposition based on the principles defined.*

Keywords: *Entity Relationship model, systems theory, decomposition, data model clustering, genetic algorithms*

INTRODUCTION

The Problem of Complexity in Entity Relationship Models

One of the most serious limitations of the Entity Relationship Model in practice is its inability to cope with complexity (Simsion, 1989; Gandhi et al, 1994; Allworth, 1996; Moody, 1997). With large numbers of entities, data models become difficult to understand and maintain. Feldman and Miller (1986) argue that this is the major reason why data modelling techniques have not realised their full potential in practice. The problem is multiplied many times over at the enterprise level, where models typically consist of hundreds or even thousands of entities.

This paper develops a method for decomposing a large data model into a hierarchy of data models of manageable size. This process is called *data model clustering* (Akoka and Comyn-Wattiau, 1996) or *data model decomposition* (Moody and Flitman, 1999). The purpose of decomposing a data model in this way is to improve human comprehension and simplify documentation and maintenance (Moody and Flitman, 1999). The research question can be simply stated as "How can a data model be decomposed in a way which maximises human comprehension and minimises documentation and maintenance effort?"

The Systems Decomposition Problem

The problem of clustering data models is an instance of the *systems decomposition problem* (Weber, 1997), or *systems simplification problem* (Klir, 1985). Decomposition is the process of breaking complex systems down into a set of smaller subsystems or modules. This is one of the most common ways of dealing with complexity in large and complex systems and is also called the "divide and conquer" strategy (Flood and Carson, 1993).

The major reason for decomposing large systems is to improve human understanding (Davis and Olson, 1985). A system which is too complex to be understood as a whole by the human mind can be broken down into a set of cognitively manageable units (Klir, 1985). Decomposition into subsystems also has advantages for development and maintenance, since subsystems can be added on, removed or modified relatively independently of each other (Wand and Weber, 1990). Finding a "good" decomposition has been identified by authors from a number of disciplines as the fundamental problem in design (e.g. Alexander, 1968; Simon, 1982; Klir, 1985).

A major problem in practice is the large number of alternative decompositions that can be produced for a particular problem (Simon, 1982; Wand and Weber, 1990). There are a tremendous number of alternative resolution forms, even for relatively simple problems. The number of possible decompositions of a system into subsystems forms a *resolution lattice* (Klir, 1985). For this reason, formal guidelines are required for choosing between alternative decompositions.

Objectives of this Paper

According to Wand and Weber (1990), a comprehensive method for decomposition should be able to:
(a) Evaluate the quality of decompositions and choose between alternatives
(b) Prescribe how to generate a "good" or optimal decomposition.

The objective of this paper is to develop a comprehensive method for decomposing Entity Relationship models. Section 2 defines a set of principles for evaluating the quality of decompositions and choosing between alternatives—this addresses requirement (a). Sections 3 and 4 describe how to generate a "good" or optimal decomposition based on these principles—this addresses requirement (b).

PRINCIPLES FOR DECOMPOSING DATA MODELS

In this section, we define a set of principles for clustering data models—these define the characteristics of a "good" decomposition. The principles provide the basis for evaluating the quality of a decomposition, and choosing between alternatives. The principles are derived from systems theory and principles of human cognition. We have also defined formal metrics for each principle to enable them to be *operationalised* and incorporated in an optimisation algorithm. Detailed discussion of metrics are beyond the scope of this paper but are defined in Moody and Flitman (1999).

What Makes A Good Decomposition?

There are an enormous number of alternative decompositions that can be produced for a particular system. As the number of components in a system increases, the number of ways in which it can be decomposed increases exponentially, which results in a "tyranny of choice" (Weber, 1997). According to Weber, the number of possible decompositions is 2^n, where n is the number of components in the system. This means that for an application data model ($\approx$ 95 entities), there are over 10^{28} different ways that it can be decomposed into subject areas. In the case, of an enterprise data model ($\approx$ 536 entities), there are over 10^{160} possibilities!

While there is probably no single "correct" decomposition in an absolute sense, clearly some will be better than others for the purposes of understanding and/or maintenance. For this reason, we need formal guidelines or *principles* for choosing between alternatives.

Formal Statement of the Problem

We formally define the problem of data model decomposition as follows:

- The initial system (or *problem state*) is a data model (D), consisting of a set of entities (E) and a set of relationships (R). Each relationship in R defines an association between two entities in E, although the entities may not be distinct (i.e. recursive relationships are allowed).
- The terminal system (or *solution state*) is a hierarchy of n subject areas (S), organised into a finite number of levels (L_1, L_2, ...). Successive levels in the hierarchy represent increasing levels of abstraction from the original data model. Each subject area will consist of either a subset of entities in E (L_1) or a subset of subject areas at the next level down.

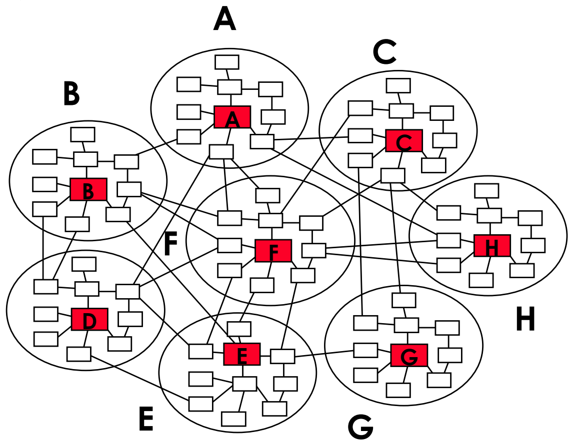

Figure 1. Level 1 Decomposition (Level 1 Subject Areas)

At the lowest level (L_1), each subject area is defined as a subset of entities in E. Each Level 1 subject area is named after one of the entities it contains, called the *central entity* (see Figure 1). At higher levels, each subject area is a subset of subject areas at the next lower level—for example, each subject area in L_2 is an aggregation of subject areas in L_1 and so on. This results in a hierarchy in which elements at each level are groupings of elements at the next level down (Figure 2). This is called a *multi-level structure system* (Klir, 1985) or *level structure* (Weber, 1997).

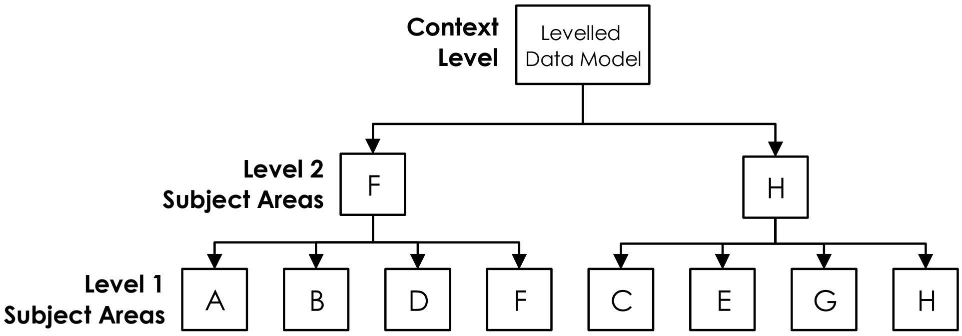

Figure 2. Multi-Level Structure System or Level Structure

Principle 1: Completeness

This principle requires that each entity must be assigned to *at least one* subject area—in other words, the decomposition should *cover* the entities in the underlying model. This principle should be applied at each level of the hierarchy (i.e. each element at a particular level of the hierarchy belongs to at least one subsystem at the next level up).

The objective of decomposition is to reduce the complexity of a system while preserving all information in the original system (Davis and Olson, 1982; Klir, 1985). In the context of data model clustering, this means that all entities and relationships in the original data model should be preserved in the decomposition process. This principle ensures that the decomposition is *lossless* (Weber, 1997).

Principle 2: Non-Redundancy

This principle requires that each entity must be assigned to *at most one* subject area. This ensures that subject areas form disjoint subsets of E. This principle should be applied at each level of the hierarchy (i.e. each element at a particular level of the hierarchy belongs to at most one subsystem at the next level up).

This principle minimises redundancy between subject areas. This reduces maintenance effort because changes to each entity can be made in a single place. It also improves understanding because overlap between subject areas can lead to confusion in user validation (Moody, 1997).

Principle 3: Integration

This principle requires that each subject area forms a *fully connected subgraph* of the original model (D). This means that each entity on the subject area must be related to all other entities on the subject area via an unbroken sequence of internal relationships.

This principle ensures that each subject area forms a fully integrated cluster of entities. This improves understandability by making sure that each subject area can be understood as an meaningful whole. This principle effectively defines a "minimum cohesion" condition for each cluster (Weber, 1997).

Principle 4: Unity

Each subject area should be named after one of the entities on the subject area, called the central entity. The central entity forms the "nucleus" of the subject area. This helps to ensure the *unity* of the subject area—that is, that all entities in the subject area relate to a single business concept or subject. Central entities should be chosen as the entities of greatest business significance to ensure that clusters are as meaningful as possible (Moody, 1997).

We proposed that *connectivity* (the number of relationships an entity participates in) be used as a surrogate measure of business importance. The psychological justification for this is based on two theories of human memory: *semantic network theory* (Collins and Quillian, 1969, 1972) and *spreading activation theory* (Anderson and Pirolli, 1984). According to these theories, semantic memory is structured as a network of related concepts. The concept of spreading activation says that nodes in a semantic network remain in a quiet state until they are activated or "primed". The activation then spreads with decreasing intensity along all pathways connected to the initial node. The level of activation decays exponentially as a function of the distance that it spreads. Spreading activation theory predicts that recall accuracy will be highest and response latency will be lowest for concepts with large numbers of connections to other concepts, because they will receive higher levels of priming. In the case of a data model, entities with large numbers of relationships would therefore be more likely to be recalled. If we assume that "ability to recall" equates to importance, we can conclude that entities with the most relationships will also be perceived as the most important. Experimental evidence has confirmed that connectivity is highly correlated with perceived importance (Moody and Flitman, 1999).

Principle 5: Cognitively Manageable

This principle requires that each subject area is of cognitively manageable size. We operationalise this principle by requiring that each subject area consists of a maximum of nine concepts—this represents the upper limit of human cognitive capacity. There is universal agreement among cognitive psychologists that due to limits on short term memory, the human mind can only handle "seven plus or minus two" concepts at a time (Miller, 1956; Baddeley, 1994). Once the amount of information exceeds these limits, it must be organised into larger and larger chunks, each containing more information and less detail (Uhr et al, 1962).

Limiting the size of subject areas helps to overcome both the limitations of the human mind in dealing with large amounts of information (*understanding*) and the restrictions of physical sheets of paper (*documentation and maintenance*). If a maximum of nine concepts is used for subject areas at each level, diagrams can be easily drawn on standard sized paper, and the need for reduced fonts and/or crossed lines is virtually eliminated.

Principle 6: Flexibility

An important characteristic of the quality of a decomposition is its flexibility to change. Systems need to adapt to changes over time, and should therefore be organised in a way which is resilient to change (Davis and Olson, 1985; Wand and Weber, 1990; Simon, 1982). Data models tend to increase in size over time, as new requirements are added or the system expands in scope. The partitioning of the data model into subject areas should therefore allow adequate capacity for growth. A data model which consists of subject areas that are all of the maximum size (nine) will have to be repartitioned if even a single entity is added.

We operationalise this principle by requiring that the average size of subject areas is as close as possible to seven entities. This allows, on average, 30% capacity for growth. This reduces the need for future re-partitioning of the model, which in turn simplifies documentation and maintenance. Note that choosing a lower optimal size would reduce the complexity of individual subject areas, but would increase the number of subject areas and the number of levels required. This increases the structural complexity of the model (which is determined by the number of subsystems) and the need to navigate between subject areas.

There is also a strong cognitive justification for using seven as the optimum number of concepts for each subject area. Recall that the limits on short-term memory are defined as a range—"seven, plus or minus two". This means that some people will have a limit of five concepts, others will have a limit of nine concepts, while most people will be around the average (seven). Therefore to maximise understandability to all people, it is preferable to use the average rather than the upper limit of human cognitive capacity as the optimal size of clusters.

Principle 7: Equal Abstraction

Another important requirement of a good decomposition is the principle of *equal abstraction* or *balancing* (De Marco, 1978; Klir, 1985; Francalanci and Pernici, 1994). This states that each subsystem should be approximately equal in scope. In the context of a levelled data model, this means that all subject areas should be similar in size. Equal abstraction is an important principle in hierarchical organisation (Klir, 1985). We operationalise this principle by defining the *minimum size* of subject areas as five entities. An alternative metric which could be used is the standard deviation in size of subject areas, but a minimum size constraint is much easier to apply in practice.

Principle 8: Coupling

Coupling is defined as the strength of association *between* different subsystems, and is widely accepted to be one of the most important measures of the quality of a decomposition (Simon, 1982). In the context of data model decomposition, minimising coupling means minimising the number of relationships between entities from different subject areas (called *boundary relationships*).

Coupling should be minimised to increase the independence of the parts of the system (Wand and Weber, 1990; Flood and Carson, 1993). Systems that have low coupling are generally easier to maintain because subsystems can be maintained relatively independently of each other (Yourdon and Constantine, 1979; Davis and Olson, 1985; Weber, 1997; Flood and Carson, 1993). The fewer the interactions between subsystems, the less likely changes to one subsystem will affect other subsystems. In addition, minimising coupling improves understandability by reducing the need to navigate between subject areas.

Principle 9: Cohesion

The complementary concept to coupling is *cohesion,* which is defined as the strength of association *within* each subsystem. Cohesion should be *maximised,* to increase independence of subsystems. In the context of data model decomposition, maximising cohesion means maximising the number of relationships between entities on the same subject area (called *internal relationships*).

Subsystems which are highly cohesive are likely to be more independent of each other, which simplifies maintenance (Yourdon and Constantine, 1978; Flood and Carson, 1993). It is also believed that subsystems that are highly cohesive are easier to understand. Presumably this is because they can be encoded as a single integrated "chunk" of information rather than a set of relatively independent concepts which must be separately encoded (Eysenck and Keane, 1992; Weber, 1997). Grouping together entities which are strongly related together is likely to result in a unit of information which can be understood as a meaningful whole.

Coupling vs Cohesion

Note that the total cohesion of a decomposition (the number of internal relationships) plus the total coupling of a decomposition (the number of boundary relationships) will always equal the total number of relationships in the model. As a result, increasing coupling will decrease cohesion by an identical amount. Therefore maximising coupling will minimise cohesion, so these two principles are logically *dependent*. As a result, following the rule of parsimony, we can eliminate one of them.

Alternatively, we can combine the two principles together into a new concept called *relative cohesion,* which is the *ratio of cohesion to coupling* of the decomposition (the number of internal relationships divided by the number of boundary relationships in the decomposition). Relative cohesion provides a means of comparing the quality of decompositions independent of the size of the underlying data model. As a general rule, the level of cohesion should be at least twice the level of coupling (internal forces twice as strong as external forces).

In the following sections, we show how these principles can be applied using a manual procedure (Section 3) and an optimisation algorithm (Section 4).

MANUAL DECOMPOSITION PROCEDURE

In this section, we define a manual procedure for decomposing a data model based on the principles defined in Section 0. The principles defined are sufficient for a human to carry out the clustering task, but a procedure simplifies the task by breaking it down into manageable steps. This will reduce the conceptual difficulty of the task and should therefore improve task performance (Flood and Carson, 1993). The procedure is summarised in Figure 3.

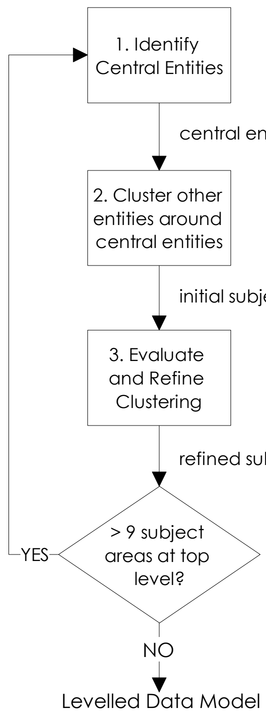

Figure 3. Manual Decomposition Procedure

Step 1: Identify Central Entities

The first step of the clustering process is to identify the central entities. For a model with *n* entities, *n*/7 central entities (or Level 1 subject areas) will be required to allow capacity for growth [Principle 6]. The entities which have the highest number relationships should be chosen as the central entities [Principle 4], although user input may also be used to help identify the most important entities.

Step 2: Cluster Entities Around Central Entities

Other entities should then be clustered around each of the central entities, making sure that each entity is assigned to one and only one subject area [Principles 1 and 2]. Each subject area should consist of a minimum of five and a maximum of nine entities [Principles 5 and 7]. This is literally, *seven, plus or minus two* entities per subject area. The upper bound ensures that subject areas are within the bounds of human cognitive capacity, while the lower bound helps to achieve balancing between the size of clusters. (Note: Experience with the method in practice shows that Principle 3 does not need to be included in the procedure, because this requirement is implicit for humans performing the task).

There is a clear symmetry in using "seven plus or minus two" as the limit on the size of subject areas. Grouping of entities into subject areas is done to facilitate "chunking" in the human mind. For this to be most effective, clustering of entities should mirror as closely as possible the way concepts are chunked in the human mind. It is therefore appropriate that the "chunking factor "used in clustering is identical to that used by the human mind.

Step 3: Evaluate and Refine Decomposition

Calculate the number of internal relationships (cohesion) and the number of boundary relationships (coupling) for the decomposition. The level of cohesion should be at least twice the level of coupling [Principles 8 and 9]. Alternative ways of grouping entities should be explored to try to reduce the coupling between subject areas.

Higher Level Decomposition

If at the end of this process, there are more than nine subject areas, Steps 1, 2 and 3 will need to be applied recursively:

- The number of Level 1 subject areas should be divided by seven to determine the number of second level subject areas required. The n/7 most important Level 1 subject areas should then be identified based on their connectivities. The connectivity of a subject area is the number of boundary relationships it has.
- Clustering of Level 1 subject areas into Level 2 subject areas can then be done in a relatively deterministic way. The objective will be to minimise coupling between subject areas while obeying the seven, plus or minus two rule.

This process should be repeated until there is less than ten subject areas at the top level.

AUTOMATIC DECOMPOSITION PROCEDURE

In this section, we describe a genetic algorithm for automatically clustering a data model using the principles defined in Section 2. Because of the enormous number of decompositions that are possible in even small data models, it is beyond human cognitive capacity to find an optimal solution. Given the number of alternatives that are possible, combined with the range of objectives we need to satisfy, a robust technique capable of solving complex non-linear systems was required. Since the robustness of traditional optimisation techniques has been frequently questioned, we decided to use a genetic algorithm to solve the problem. *Genetic algorithms* differ from standard optimisation techniques in four ways (Goldberg, 1989):

- They work with a coding of the parameter set, not the parameters themselves
- They search from a population of points
- They use payoff information, not derivatives
- They use probabilistic transition rules, not deterministic rules.

Genetic Algorithms

Genetic algorithms are a computer simulation of genetic theory (Goldberg, 1989). Each *generation* consists of a population of individuals, represented by *chromosomes,* which have varying levels of *fitness*. "Fitness" is a measure of performance, and in this case, is defined by the principles described in Section 2. After the fitness of each member is determined, the next generation is created. The chances of an individual surviving to the next generation are proportional to its fitness. *Random mutations* and *crossovers* between population members also affect each generation. The likelihood of mutations and crossovers are determined by the mutation and crossover rate respectively. After a number of generations, the population will tend toward "fitter" members and therefore a more optimal solution.

We used a non-repeating enumerated chromosome for this application. This is simply a chromosome consisting of an ordered string of numbers. The numbers used represent either entities or subject areas. In our case the numbers were the integers 1 to n (where n represents the number of entities in our diagram), plus integers n+1, n+2, …, n+m-1 where m was the number of subject areas. The process of mutation and crossover outlined earlier in our basic discussion of genetic algorithms was modified to take account of the special non-repeating and order dependant characteristics of our chromosome.

Formulation of the Algorithm

The pseudo-code for the genetic algorithm is as follows:

```
Generate random population
DO for all possible subject counts
   DO WHILE populations since optimum chromosome < 200
         Determine population fitness
         Make probability of survival proportional to fitness
         Recombine via single point crossover and mutation
   END WHILE
END WHILE
```

A population of 32-bit length 'chromosomes' was initialised randomly. Each chromosome was passed through the code represented by the pseudo code below to calculate its interdependency. The probability of survival was inversely proportionate to this interdependency. The surviving chromosomes in the population were combined via single point crossover and finally a number of bits were mutated.

Empirical Testing

The algorithm was coded in Delphi 3 and tested on a number of Entity Relationship models of varying sizes. The algorithm was found to outperform a human expert in terms of both time taken and the quality of the result (as measured by the principles defined). The improvement in performance was found to increase with the size of the model used.

CONCLUSION

Summary

This paper has defined:

1. A set of principles and metrics for evaluating the quality of a decomposition and choosing between alternatives
2. A manual procedure for decomposing a model based on these principles, which enables a human to produce a relatively optimal solution
3. A genetic algorithm that automatically finds an optimal decomposition based on the principles defined.

In this paper, decomposition is carried out based entirely on relationships between entities (*form*), rather than on any concept of meaning (*content*). It is therefore a *syntactic* rather than *semantic* approach to decomposition. Some approaches have attempted to incorporate meaning into the process by explicitly defining semantic associations between entities (e.g. Francalanci and Pernici, 1994). However to do this is so time-consuming and subjective, it would be quicker and more effective to do the partitioning entirely manually.

Practical Significance

The complexity of data models is a major barrier to the effective communication of data models in practice. Clustering provides a solution to this problem by dividing the model into conceptually "bite-sized" pieces. The development of formal principles to guide the clustering process should improve the quality of the result, reduce cognitive uncertainty and improve consistency between analysts. Because the principles are soundly based on principles of human information processing, the model should be clustered in a way which maximises human understanding.

The paper defines a manual procedure that can be used by a human designer to produce a relatively optimal solution to the problem. Providing a structured procedure for carrying out the clustering should reduce the conceptual difficulty of the task (*efficiency*) and improve task performance (*effectiveness*).

Finally, the paper defines an algorithm for decomposing a data model which automatically finds an optimal or nearly optimal solution. This will reduce the effort required to apply the method (*efficiency*) and should result in a better solution than would be possible using a manual procedure (*effectiveness*) because of the limitations of human information processing.

Theoretical Significance

The major theoretical contribution of this paper is to define a comprehensive set of theoretical principles for decomposing data models. All of the principles defined are formally justified in terms of theory, defined in mathematical terms and measurable. These principles provide a solid foundation for future research in this area.

The method used in this paper also provides a general approach or *methodological paradigm* for solving design problems in the information systems field and in other fields. At some level, all design problems are decomposition problems. A wide range of design problems can therefore be formulated as systems decomposition problems and solved in a similar manner. The general approach is as follows:

1. Formulate problem as a systems decomposition problem.
2. Define principles and metrics for decomposition—what are the characteristics of a good decomposition?
3. Develop procedure (manual and/or automated) for generating a good or optimal decomposition.

Further Research

We have argued on theoretical grounds that this method offers a better prospect of success in clustering large data models than previous approaches proposed in the literature. While theoretical justification is important and necessary, ultimately the soundness of any method is an empirical rather than a theoretical question (Ivari, 1986). Some limited empirical testing of the method has been carried out as part of this paper. However it is the task of further research to systematically evaluate the efficacy of the method in practice.

REFERENCES

Akoka, J. and Comyn-Wattiau, I. (1996) "Entity Relationship and Object Oriented Model Automatic Clustering", *Data and Knowledge Engineering*, 20.

Alexander, C. (1968): *Notes on the Synthesis of Form*, Harvard University Press, Boston.

Allworth, S. (1996): "Using Classification Structures to Develop and Structure Generic Industry Models", *Proceedings of the First Australian Data Management Conference*, Data Management Association (DAMA), Melbourne, Australia, December 2-3.

Anderson, J.R. and Pirolli, P.L. (1984): "Spread of Activation", *Journal of Experimental Psychology: Learning, Memory and Cognition,* 10, 4.

Baddeley, A. (1994): "The Magical Number Seven: Still Magic After All These Years?", *Psychological Review,* 101, 2.

Collins, A.M. and Quillian, M.R. (1969): "Retrieval Time from Semantic Memory", *Journal of Verbal Learning and Verbal Behaviour,* 8.

Collins, A.M. and Quillian, M.R. (1972): "How to Make a Language User", *in Organisation and Memory,* E. Tulving and M. Donaldson (ed.s), Academic Press, New York.

Davis, G.B. and Olsen, M.H. (1985): *Management Information Systems: Conceptual Foundations, Structure and Development*, McGraw-Hill.

De Marco, T. (1978): *Structured Analysis and System Specification*, Yourdon Press, 1978.

Eysenck, M.W. and Keane, M.T. (1992): *Cognitive Psychology: A Student's Handbook*, Lawrence Erlbaum Associates, Hove and London.

Feldman, P. and Miller, D., (1986): Entity Model Clustering: Structuring a Data Model by Abstraction, *The Computer Journal*, Vol. 29, No. 4.

Flood, R.L. and Carson, E.R. (1993): *Dealing With Complexity: An Introduction to the Theory and Application of Systems Science*, Plenum Press.

Francalanci, C. and Pernici, B. (1994): "Abstraction Levels for Entity Relationship Schemas", in P. Loucopoulos (ed.) *Proceedings of the Thirteenth International Conference on the Entity Relationship Approach*, Manchester, December 14-17.

Gandhi, M., Robertson, E.L. and Van Gucht, D. 1994): Levelled Entity Relationship Model, *Proceedings of the Thirteenth International Conference on the Entity Relationship Approach*, Manchester, December 14-17.

Goldberg, D. (1989): "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley, p15-21.

Ivari, J. (1986): Dimensions Of Information Systems Design: A Framework For A Long Range Research Program. *Information Systems*, June, 39-42.

Klir, G.J. (1985): *Architecture of Systems Problem Solving*, Plenum Press, New York.

Miller, G. (1956): The Magical Number Seven, Plus Or Minus Two: Some Limits On Our Capacity For Processing Information, *The Psychological Review*, March.

Moody, D.L. and Flitman, A.R. (1999): "A Methodology for Clustering Entity Relationship Models: A Human Information Processing Approach", *Proceedings of the Nineteenth International Conference on Conceptual Modelling (ER'99)*, Paris, France, December 3-6, 1999.

Moody, D.L., "A Multi-Level Architecture for Representing Enterprise Data Models", *Proceedings of the Sixteenth International Conference on Conceptual Modelling (ER'97)*, Los Angeles, November 1-3, 1997.

Simon, H.A. (1982): *Sciences of the Artificial*, MIT Press.

Simsion, G.C. (1989): A Structured Approach to Data Modelling, *The Australian Computer Journal*, August, 1989.

Uhr, L., Vossier, C., and Weman, J. (1962): Pattern Recognition over Distortions by Human Subjects and a Computer Model of Human Form Perception, *Journal of Experimental Psychology*, 63.

Wand, Y. and Weber, R.A. (1990): A Model for Systems Decomposition, in J.I. DeGross, M. Alavi, and H. Oppelland (Ed.s), *Proceedings of the Eleventh International Conference on Information Systems*, Copenhagen, Denmark, December.

Weber, R.A. (1997): *Ontological Foundations of Information Systems,* Coopers and Lybrand Accounting Research Methodology Monograph No. 4, Coopers and Lybrand Australia, Melbourne, Australia.

Yourdon, E. and Constantine, L.L. (1979): "*Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design*", Prentice-Hall, Englewood Cliffs, NJ.