UNDERSTANDING THE WAREHOUSE: A GRAPHICAL SEMANTIC MODEL FOR DATA WAREHOUSING WITH EXPLICIT DIAGRAMMATIC CONVENTIONS

A dissertation submitted in partial fulfilment

of the requirements for the Open University's

Master of Science Degree

in Computing for Commerce and Industry.

Colin McGowan

(U0828809)

March 2007

Word Count: 14,053

PREFACE

I would like to my thank my fiancée, Julie, for her understanding and time organising our wedding arrangements whilst I have been working on my dissertation. I would also like to thank my supervisor, James Matthews, for his positive feedback throughout the duration of the research project.

I chose the research area of data warehousing because I think it presents one of the most exciting opportunities to leverage the capabilities of existing information systems. The last 30 years of technological change have seen a maturity in our ability to collect, store, and distribute data efficiently. The next 30 years will surely see a rapid increase in our capability to use this data effectively.

In my view, the challenge of effective data usage is as much, if not more, about how we view our data assets, as about new technologies. For this reason, I have chosen to focus on *understanding the data warehouse*.

I hope readers will find this research informative. In particular I hope those interested in the field of data warehousing will find the ideas presented in this paper thought provoking and worthy of consideration.

CONTENTS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Data warehousing (DW) is increasingly being used by business to store data for the purpose of decision support (Inmon, 1996). A DW is populated with data from pre-existing business systems and/or other external sources. The data is transformed and integrated to provide a more complete picture of the business (Inmon, 1996; Kimball and Ross, 2002). The assumption of the DW process is that decision makers, when presented with this richer source of information, will be able to make more informed decisions.

Although the purposes and goals of DW are widely understood and agreed upon there is less consensus about the optimal approach. Central to the debate is whether DW requirements can be derived from the data itself or whether, as with traditional application development, a user driven requirements specification should be the basis for development.

The literature in this area suggests that a DW process should reconcile user requirements with the available data. If such a reconciliation is not performed there is a risk of populating the DW with data that cannot be interpreted by users and therefore used for decision making (Artz, 2006). The aim of the research is therefore: to identify a technique that helps users reach the level of understanding necessary to guide the creation and use of the DW for its intended purpose of decision support.

The traditional method for communication with users in data centric systems design is graphical conceptual data modelling (GCDM). Several authors have proposed methods to facilitate conceptual modelling for DW. The research evaluated these

methods from two perspectives: firstly their ability to represent the semantic requirements of a DW, secondly to see whether the models could communicate the semantic requirements in a way that could be easily interpreted by users.

The dual focus of semantic and cognitive properties of DW models differentiated this research from previous work, which was predominately concerned with the semantic richness of the model.

The survey revealed that whilst there is consensus in the need to conceptually partition data into *facts* and *dimensions*, there are a number of discrepancies between the modelling techniques in the amount of support offered for temporal properties, the impact of systems integration, and derived data. Furthermore, it was observed that cognitive properties are often given little explicit consideration. Many of the models did not explain how their choice of layout, decomposition, and abstraction helped emphasise the semantic properties of the DW and enhance user understanding.

These findings guided the proposal for a new DW conceptual modelling technique. The technique should be capable of modelling the common semantic requirements of a DW. The model is presented using a template approach, which offers explicit guidance on layout, decomposition and abstraction.

1. INTRODUCTION

1.1 Problem Overview

Data warehousing (DW) is increasingly being used by business to store data for the purpose of decision support (Inmon, 1996). A DW is populated with data from pre-existing business systems and/or other external sources. Data from these sources are transformed and integrated to provide a more complete picture of the business (Inmon, 1996; Kimball and Ross, 2002). The overriding assumption of the DW process is that decision makers, when presented with this richer source of information, will be able to make more informed decisions.

The assumption that business users will immediately understand and appreciate the data contents of the DW is one that has been challenged by empirical studies. Sampson et al. (2002) observed that the complexity of tool, data model, and interface to the DW were a barrier to user understanding. Shanks et al. (2003) found that a DW initiated by the IT department was later abandoned because users did not understand the contents of the DW or how it would improve their decision making. A case study by Hess and Wells (2002) pointed to the central importance of metadata, being data that helped users understand the context of the DW data. One analyst commented that they spent between 20%-50% of their time trying to track down such data. The study found that lack of current and quality metadata was a barrier to effective analysis.

More recently Artz (2006) observed that data in the DW is of little value unless the meaning of the data has been validated and agreed upon by users of the system. Artz argued that previous research on DW had been too focused on methods for populating the DW, without regard for the usefulness of this data to the user.

1.2 Aims of research

This study will consider the challenge of communicating the semantic content of the DW to the user. Traditionally graphical conceptual modelling has been seen as the most effective way of communicating technical database specifications to non technical users.

A number of conceptual models have already been proposed for DW. Each model emphasises different semantic characteristics of the DW. It is not clear whether any of these models have the necessary expressiveness to fully represent the DW contents to users. Furthermore given the range of notation and graphical constructs proposed, there is clearly no consensus on what is the most effective means of representing the DW semantics to users. This study will build on this previous work by considering the following two questions:

What semantic information needs to be communicated to users of a data warehouse?
The DW development process integrates data from heterogeneous sources and requirements from heterogeneous user groups. It is necessary to understand and reconcile both user requirements and available data (Winter and Strauch, 2004).

What is the most effective way of representing these semantic requirements?
Larkin and Simon (1987) demonstrated that in addition to presenting all the information, it is also necessary to consider the cognitive load that the representation places on the user. Representations that have high cognitive load will be difficult for users to understand and reason with.

Answering these questions should provide DW developers and researchers with a comprehensive set of semantic requirements for DW modelling. The study should also provide guidance to DW developers on how to best represent the information content to users.

1.3 Assumptions

Implicit in the overview have been three assumptions.

1. DW users need to understand the data model if they are to make the most effective use of the DW contents. What differentiates the DW from the many other systems that people are exposed to on a daily basis?

2. A graphical representation will be the most effective way to communicate the DW semantic content to users. Previously research has found that diagrammatic representation offer a significant advantage over propositional representations in certain circumstances (Larkin and Simons, 1987).

3. A conceptual model offers advantages over traditional storage models. Studies suggest that this is because the representation is closer to the problem domain (Chan et al., 1998; Sinha and Vessey, 1999).

1.4 Contribution to knowledge

This research will contribute to the knowledge on DW in the following ways:

- *A discussion of user understanding in the context of DW development and use*
- *Identification of the semantic requirements for a DW modelling approach*

- *A framework for evaluating DW modelling techniques that gives explicit consideration to usability*
- *The formulation of a graphical modelling technique for capturing the semantic requirements of a DW*

1.5 Objectives

1. *Establish the case for user understanding of the DW data model as a key component of the DW development lifecycle and use*
2. *Construct a framework of the general semantic requirements of a DW*
3. *Evaluate the expressiveness of existing data models against the semantic framework*
4. *Establish the case for a graphical diagrammatic approach to documenting DW semantic content*
5. *Construct a framework of diagrammatic conventions applicable to the area of DW that can be used to evaluate the computational effectiveness of existing data models*
6. *Use diagrammatic framework to evaluate the computational efficiency of existing data models in expressing semantic information*
7. *Propose extensions to model and possible avenues of future research*

2.0 LITERATURE REVIEW

2.1 User interaction with the data model

2.1.1 The data model and user interaction with the data warehouse

Inmon (1996) describes the DW as a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management decisions.

From the perspective of user understanding, the DW exists to support decision making whereas the OLTP (online transaction processing) system exists to support a business process. When interacting with an OLTP system the user generally has a set task to perform. The system is built to support that task and a user interface will have been designed to guide the user through to successful completion. A DW is built to aid decision making, however the exact decisions to be made and the analysis required to make them cannot normally be specified in advance (Kimball and Ross, 2002).

Chenoweth et al. (2006) describes the need to create a power user, one who understands the business and the DW structure, as one of the seven key interventions for success of the DW. Their field study found that one of the most successful uses of the DW occurred when a user interacted directly with the underlying data to access a wide variety of information.

Users will need to specify the data they require from the DW if they are to perform ad hoc exploratory analysis. In a study performed by Chan et al. (1998) they found that errors in interpreting the data model propagated into query formulation. This suggests that even if users employ a technical specialist to write the queries, the query specification must be based on a correct view of the data model.

Artz (1997) argues that DW users are farther away from the underlying table structures. Firstly the process of extracting and integrating data from various sources may in itself complicate the semantics of the data. Additionally, DW users have far less control over of the semantics and business rules of the data than users of OLTP systems. An Account Payable Manager may decide which fields to use for data, what days of the week to perform certain input etc, and thus shape the semantics of the data in the system. This may be tacit knowledge to the Accounts Payable department but remains unknown to the DW analyst.

2.1.2 The data model and requirements specification for the data warehouse

Approaches to DW are often categorised as an Inmonite, data-driven philosophy (Inmon, 1996), or alternatively, a requirements-driven Kimballite approach (Kimball and Ross, 2002). This oversimplifies the views of the authors however the debate has impacted the direction of research in this field.

Inmon (1996), states that the DW starts with the Corporate Data Model (CDM). The CDM is an integrated model of the existing information assets of the organization. The DW is then developed incrementally from the CDM by adding an element of time to the model and categorizing data elements by their temporal volatility. The premise here is that the DW is developed from existing information systems. Inmon argues that the data in these systems will be useful for analysis once integrated, even if users cannot necessarily perceive exactly how they will use it in advance.

Kimball (2002) rejects the idea that it is necessary to pre-integrate the organization's entire data model before DW development can begin. Kimball believes that requirements should be specified by the business users along business process lines.

Central to both approaches is the need for communication between the users and the DW developers. Inmon concedes that although not all requirements can be predicted by users, "on the other hand, anticipating requirements is still important. Reality lies somewhere in between." Kimball for his part states that a dual pronged approach is required, where the needs of the business are taken in the context of the realities of the data.

The commonality here is that during the process there must be reconciliation between data and requirements. In Winter and Strauch (2002) the authors propose performing this reconciliation at an aggregate level before considering detailed requirements.

From a theoretical standpoint Artz (2006) highlights the inherent danger of a DW methodology that does not rely on users expressly understanding and validating the semantics of the data. Artz argues that if the data in the DW has not been specified by user requirements then:

> "*The strongest validity claim that can be made is that any information derived from this data is true about the data set, but its connection to the organization is tenuous"*.

This discussion suggests that user understanding in communication between user and developer is essential if the DW contents are to have meaning. To form a correct semantic representation of the information assets of the organization, either partially,

using a Kimball approach, or completely, by specifying a CDM, business users and developers must unambiguously agree on the data structure and semantics.

2.1.3 The data model as a medium for communication of data requirements and semantics

Sampson and Atkins (2002) refer to the correlation between user understanding and the actual data as the semantic integrity of the DW. They suggested that exposing the user to a formal data model such as the entity-relationship model (ER) may be problematic and instead propose the use of structured sentences (Atkins and Patrick, 1998).

A propositional sentence based approach was not supported by an empirical study that tested human understanding of functional dependencies. Artz (1997) found natural English too awkward to express all but the most obvious data relationships.

Kim (1995) and Parsons (2003) provide direct support for use of data models with non technical users. Both studies found that users were quick to pick up the modelling notation and validate data models to a high degree of accuracy.

In conclusion, reconciliation between requirements and data is essential regardless of whether a data-driven or requirements-driven approach is chosen. To ensure the semantic integrity of the DW the data content must be unambiguously validated by the business users. Unambiguous validation can be problematic using informal methods like natural language or interviewing. Empirical research has found that users can accurately validate a data model given reasonable training on the model's constructs.

2.2 Prerequisites of diagrammatic representation and reasoning

Since the ER model was first proposed (Chen, 1976), diagrammatic representation has increased in popularity in the software community. However those in the research field of cognitive science have been circumspect about their use. The title of Larkin and Simon's much cited paper (Larkin and Simon, 1987) contains the caveat that diagrams are 'sometimes' more effective. This suggests we should consider when and under what circumstances a graphical representation will be more effective.

What are the prerequisites for successful interaction with a given representation? In addition, what are the properties of graphical representations that potentially make them a more effective communication tool in the DW environment?

2.2.1 Productions

Larkin and Simon (1987) make the point that any representation will be of little value to the viewer if they lack the necessary productions to interpret it. By the term *productions* the authors refer to the set of rules that govern the domain and the specific representation.

The need for relevant productions is demonstrated by Cheng et al. (2001). In their paper they present a weather map. To a trained meteorologist the map can be used to make inferences about future weather patterns. To most other users the picture can be seen to represent the country of Australia, but little other information could be gleaned.

From this we can conclude that diagrams require the user to learn the rules necessary to interpret them. This learning should be supported as part of the modelling process.

It is also important to consider the time taken to gain the necessary productions. If the effort required is too great then the user is unlikely to learn them.

2.2.2 Correlation between representation and problem domain

Larkin and Simon (1987) differentiate diagrams from other forms of representation on the basis that they can preserve the topological and geometric relationships among the components. Their pulley diagram supported the problem solving exercise by its representation of the component parts. A diagram should therefore preserve some attributes of the problem explicitly in its representation.

2.2.3 Problem complexity

Carlson et al. (2003), state that learning imposes two types of *cognitive load*. Cognitive load is the mental effort required for the exercise and can be *intrinsic* or *extraneous.* Intrinsic load is that imposed by the complexity of the problem domain. Extraneous load is that imposed by how the information is presented. Their empirical studies found that diagrammatic representations were only more effective in instances where intrinsic cognitive load was high. The inference drawn from this study is that diagrammatic representations do offer advantages when learning complex domains.

This discussion has highlighted three high-level constraints on the use of diagrammatic representation. These are:

- Users must be taught how to use a diagram effectively;
- The representation should directly reflect some aspect of the problem domain;
- Diagrams only offer significant advantage in modelling non-trivial problems.

2.3 Diagrammatic properties for representation and reasoning

2.3.1 Abstraction

Degani (2004) used the London Underground map to demonstrate the power of abstraction by contrasting the current London Underground map layout, with the original version that users found confusing. The latest version ignores most of the geographical information presented in the original and instead concentrates on the relationships between stations. By abstracting out only those details relevant to the user task of navigating the underground network, the map became much more effective (Degani, 2004).

An empirical study conducted by Moody (2002) supported the use of representing complex data models at different levels of abstraction. The study found that the Levelled Data Model (LDM) performed significantly better in terms of the users' ability to verify the data model. The LDM multi-levelled approached reduced the complexity of the user view.

2.3.2 Decomposition

Decomposition is the division of knowledge into meaningful units (Hahn and Kim, 1999). Diagrams can exploit this by representing each unit as a different graphical component. In doing so, the representation allows natural grouping of objects. Hahn and Kim's experiment showed that diagrams with effective decomposition supported analysis of the problem domain. This resulted in participants making fewer errors in their interpretation of the models.

2.3.3 Layout

Hahn and Kim (1999) observed that explicit layout conventions had a positive effect on users' ability to represent a design using a given syntax. However, the problem of determining the optimal layout for interpretation of a diagram has proved difficult. Kulpa (1994) observes that generally, no computationally tractable algorithm exists for finding the optimal layout of complex diagrams. He states that a heuristic, knowledge based approach is a necessity.

More recently, research by Purchase et al. (2002) looked at the impact of various graph layout algorithms on user preference and syntactic performance. The study highlighted that different layout aesthetics are often mutually exclusive. Therefore, it is important to establish which is most appropriate for a given diagram.

It is clear that there is not a one-size-fits-all solution to spatial layout. Specific instances of good layout as seen in Larkin and Simon (1987), and Degani (2004) demonstrate that it has a significant impact on the computational efficiency and perceived usability of the representation where used appropriately.

2.4 Conceptual data modelling

The bases for focusing on conceptual modelling are:

- A body of empirical evidence in this area suggests that conceptual models are more effective in conveying semantics of a data model than alternative approaches (Chan et al., 1998, 2003; Liao and Palvia, 2000; Sinha and Vessey 1999)

- Conceptual models are capable of supporting richer domain semantics than alternative approaches (Siau et al., 1992)
- DW methodologies use both relational (Inmon, 1996) and multidimensional (Kimball and Ross, 2002) logical models as the basis for DW design. Conceptual models can map to either or both of these views for different user groups (Chen et al., 1997).

3. DATA WAREHOUSE MODELLING SEMANTIC REQUIREMENTS

In Chapter 2, we considered why DW use required an unambiguous and rich representation of its semantic properties. Elmasri and Navathe (2004) use the term Knowledge Representation [KR] to describe these richer schematic representations whilst acknowledging this approach has a lot in common with conceptual modelling. Hess and Wells (2002) found that the lack of high quality rich metadata was an impediment to effective use of the DW. Gemino and Wand (2005) demonstrate that increased complexity may not be so detrimental to cognition if it leads to increased conceptual clarity. Given the support and direction in the research community for richer semantic modelling and representation, the semantic requirements identified below may go beyond those represented in traditional conceptual models.

Much of the previous literature on DW conceptual modelling has focused exclusively on the requirements of the multidimensional (MD) database model. DW semantic requirements should include, but not be limited to, those found in MD modelling.

As a means of finding a high level classification scheme for DW semantic requirements it is helpful to consider Inmon's widely accepted description of a DW (Inmon, 1996):

*"A data warehouse is a **subject-oriented**, **integrated**, **time-variant** and non-volatile collection of data in **support of management's decision** making process" (emphasis added)*

3.1 The data warehouse is subject-oriented

Research related to MD modelling is helpful as it decomposes a subject area into concepts that can be mapped to modelling constructs.

The following classification of modelling constructs is intended as an overview of MD semantic requirements and summarises concepts discussed in the following references: Golfarelli et al., (1998); Sapia et al., (1998); Franconi and Kamble (2004b); Husemann et al. (2000); Abello et al., (2002); Malinowski and Zimanyi, (2004).

3.1.1 Facts

A subject has a focus of analysis. In MD modelling this is represented by a set of *facts.* Each fact represents measurements of an event related to the subject area. The exact terms of measurement are contained in *fact-attributes*.

3.1.2 Dimensions

Dimensions are an abstract concept that provide context for the facts. They provide different analysis perspectives for the fact-attributes.

3.1.3 Levels

Each level of a dimension represents a component of the dimension analogous to an entity. A level has attributes that a form criterion for analysing the associated fact-attributes.

3.1.4 Relationships

Relationships link the other constructs in the model. Relationship types include aggregation, association, generalization, and membership. Depending on the relationship type, the model should be capable of expressing the properties of: multiplicity, inclusion, strictness, completeness, and disjoint or overlap.

3.1.5 Hierarchies

Related levels in a dimension form *hierarchies*. Hierarchies are useful in DW because they describe frequently occurring organizational, temporal and geospatial structures in a way that is natural to analysts. Malinowski and Zimanyi (2004) provide a useful categorisation and analysis of hierarchies.

3.1.6 Fact-attribute constraints

Abello et al. (2002) demonstrate the need to specify the additivity of fact attributes as they apply to dimensions. Specifically it may not be valid to analyse facts across all dimensions using certain operators.

In summary, MD modelling helps the analyst think about the DW in a subject-oriented manner by:

- Differentiating the focus from the context of analysis
- Accurately representing real world relationships between data
- Explicitly representing constraints on analysis through the definition of valid hierarchies and operations on fact-attributes.

3.2 The data warehouse is integrated

The DW does not generate its own data, but captures data from other systems. System integration potentially complicates the semantics of the data. The modelling technique should support understanding of the data integration and any limitations or constraints on this integration.

Srivastava and Chen (1999) comment that data integration brings complexities to constraint definition due to constraint mismatches between source systems. They argue the strictness of constraints often signal the quality of the data. In the OLTP environment constraints help maintain the integrity of the data. In the DW environment constraints help us understand the data. Constraints come in several forms:

3.2.1 Granularity

Defining and declaring the granularity of the data is a vital step in DW design (Kimball and Ross, 2002; Inmon, 1996). A common grain is necessary for data integration to proceed. In addition, the analysis that can occur is constrained by the level of granularity set in the DW.

3.2.2 Data constraints

Operations on data and inferences about data are constrained by the domain and data type of each item in the DW. Declaring these properties should benefit analysis by restricting the possible inferences.

3.2.3 Application software constraints

Applications often contain many constraints in the software layer, not the data layer. If application constraints are explicitly modelled this will help analysts who may not have a good knowledge of the source application.

3.2.4 Business rules

Badia (2004) demonstrates that traditional ER modelling fails to capture many business rule constraints. Business rules differ from application constraints in that they are often: company specific, not directly enforced by the application or the data model, and exist as tacit knowledge to operations personnel. In the DW environment, there is a need to communicate these constraints to a broader range of users. Khan et al. (2004) propose a technique to incorporate these business rules into the data model. They argue that this should facilitate communication between stakeholders.

3.2.5 Vagueness and uncertainty

Experience has shown data integration to follow the law of diminishing returns (Srivastava and Chen, 1999). There may remain a number of anomalous entries even when the vast majority of data is integrated. Pure set related constraints are often too strict or too permissive. Work on relaxing constraints using fuzzy logic offers a solution. This allows meaningful constraint definition on the integrated data without the risk of constraint violation by a minority of noisy data (Galindo et al., 2004). Removing noisy data is only a reasonable alternative if the data is actually incorrect. Constraints must have the flexibility to handle a degree of uncertainty in an environment that integrates data from heterogeneous sources.

If the same information exists in two or more source systems a decision must be made about which source will supply the DW. Osei-Bryson and Ngenyama (2004) raised the issue of the *face* of the attributes. For example, school grades can be recorded as A-F or as a number 1-100. The mapping between these faces is not necessarily obvious and there are potentially differences in precision (Badia, 2004). Osei-Bryson and Ngenyama (2004) argue that multi-faced attributes should be supported where there are heterogeneous user groups.

3.3 The data warehouse is time-variant

Everything recorded in the DW should be associated with an element of time (Inmon, 1996). Therefore, a DW model should be capable of expressing a rich array of temporal properties.

3.3.1 Temporal data strategies

Bruckner et al. (2001) identifies four different strategies that may be used for capturing data over time:

- Transient data does not capture a history of alterations and deletions, only the current state is available
- Periodic data captures each change as a new record and stores a history of these changes permanently
- Semi-periodic data occurs where a limited history of alterations and deletions are stored
- Snapshot data represents a stable view of data at a certain point in time

3.3.2 Time context types

Bruckner et al. (2001) classify three timestamps that may be of interest to the DW user:

- Real world event (Valid time in Gregersen and Jensen (1999))
- Revelation (transaction) time is the point at which the data relating to the event was captured in electronic form
- Load time is the point at which the data relating to the event was loaded into the DW

3.3.3 Sampling period

It may be necessary to know when sampling of data in the DW commenced and finished. A DW integrates data from a number of different systems and these may not all have been available for the entire life of the DW. The conceptual model should be able to incorporate information about the evolution of the DW (Abello et al., 2002).

3.3.4 Update frequency/sampling rate

The model should show the update frequency and/or sampling rate of the data in the DW. A time lag between a real world event occurring and it being available in the DW may have an impact on the validity of any conclusions reached using the DW.

Different source systems will introduce different degrees of time lag and sampling rates by the ETL procedures. Ravat et al. (1999) introduce the concept on an *environment* to define temporal constraints and behaviour on a subset of the DW model.

3.3.5 Temporal precision

Different levels of temporal precision may exist within a DW. Ravat el al (1999) use the TEMPOS model to partition the DW into multiple levels of granularity thus supporting different levels of precision.

3.4 The data warehouse supports management decisions

The ability to support management decisions is not in itself a semantic requirement. Instead, it is an indication of the level of semantic support required in describing the properties of the DW. Decision-making requires a full understanding of the strengths and limitations of the data at hand.

3.5 Data warehouse semantic framework

Table 1 Data warehouse modelling semantic requirements

Data warehouse category	Sub Category	Concept / Reasoning	Citations *	Comment
Subject-oriented	Fact	Separation of context from content	Abello et al. (2002)	
	Dimension	Separation of context from content	Abello et al. (2002)	
	Levels	Hierarchical analysis	Abello et al. (2002)	Ragged hierarchies cannot strictly define levels
	Relationships	Association	Tryfona et al. (1999)	
		Generalisation	Tryfona et al. (1999)	
		Aggregation	Tryfona et al. (1999)	
		Membership	Tryfona et al. (1999)	
	Hierarchies	Strictness	Malinowski and Zimanyi (2004)	
		Symmetry	Malinowski and Zimanyi (2004)	
		Simple/Multiple	Malinowski and Zimanyi (2004)	
		Parallel/Independent	Malinowski and Zimanyi (2004)	
	Attribute Constraints	Fact-attributes	Abello et al. (2002)	Additivity/inclusion along dimensions
Integrated	Granularity		Inmon (1996); Kimball (2002)	
	Constraints	Data/domain constraints		

Data warehouse category	Sub Category	Concept / Reasoning	Citations *	Comment
		Application constraints		
		Business rules	Khan et al. (2004); Badia (2004)	
	Ambiguity/uncertainty	Fuzzy constraints	Galindo et al. (2004)	
		Multi face attributes	Osei-Bryson and Ngenyama (2004)	
Time-variant	Time classification	Valid Time	Bruckner et al. (2001)	
		Transaction Time	Bruckner et al. (2001)	
		DW Load Time	Bruckner et al. (2001)	
	Time lag			Explicitly documents possible data inconsistencies
	Sample period	Over what period was the data updated from source systems	Abello et al. (2002)	
	Sample frequency	How regularly data is updated from source systems		
	Precision	Grain of time attribute	Ravat el al (1999)	
	Volatility	Stability analysis	Inmon (1996)	Make explicit difference between sample frequency and validity

4. DATA WAREHOUSE MODELLING COGNITIVE PRINCIPLES

Gemino and Wand (2003) hypothesise that decreased usability may be the trade-off of a richer semantic model. The number of semantic properties identified in Chapter 3 indicates that this could be an issue for DW modelling. However, failure to fully represent the DW semantics could lead to the data being misinterpreted by analysts, with a possible negative impact on the decision making process. The ideal is a semantically rich model that remains usable for the consumer.

Chapter 2 identified abstraction, decomposition and layout as key to the cognitive efficacy of diagrammatic representation. This section expands on these general concepts by considering how they might apply to DW modelling. We also look at the idea of giving explicit opportunities for interacting with the model.

4.1 Abstraction for data warehouse modelling

Examples of abstraction are present in DW design methodologies and conceptual modelling. Winter and Strauch's (2004) method includes the creation of an aggregate information map as a first step to data analysis. The problem domain is then modelled at increasing levels of detail. Sen and Sinha (2005) observe a commonality of DW methodologies is the creation of a high level (subject-oriented) conceptual model before detailed data modelling.

Chen et al. (1997) argue that failures attributed to conceptual modelling are generally caused by not adopting a top down approach. They conclude that conceptual modelling should work like a multi-level map. Moody (1997) proposes a multi levelled data model as a means of handling complexity. Lujan-Mora (2003) and

Abello et al. (2002) provide examples of DW models that explicitly support different levels of detail and abstraction.

The discussion highlights the relationship between abstraction and cognition. By representing a problem at different levels of detail, abstraction helps control complexity. Empirical research into quality metrics for DW conceptual modelling found a correlation between increasing complexity (as measured in number of elements) and decreasing cognition (Serrano et al., 2004). A DW is a complex entity with many semantic properties. Communicating all these properties in a single representation would exceed the capacity of most humans to absorb the information. For this reason, a DW modelling technique should have the ability to represent the problem domain at different levels of detail.

It may be appropriate to extend the metaphor of a street directory used by Chen et al. (1997) and Moody (1997), to an atlas. An atlas not only represents information at different levels of detail – world, continent, country for example – but also from different perspectives – temperature, topology etc. Degani (2004) observed that when geographic detail was removed from the underground map, commuters preferred the representation. Parsons (2003) found that users should be given different views (local or global) depending upon the presence of conflicts between source schemas. These examples demonstrate how abstraction has a role in presenting both different levels of detail and different perspectives.

4.2 Decomposition for data warehouse modelling

Hahn and Kim (1999) demonstrated that good decomposition – the mapping of concepts to graphical constructs – supported effective analysis of diagrammatic representations.

Burton-Jones and Weber (1999) urge care in the mapping of concepts to constructs. In their empirical study on the use of ER diagrams, they found the problem-solving performance of users deteriorated in diagrams where relationships had attributes. The authors concluded that allowing relationships to assume attributes reduced ontological clarity of the construct because the relationship started to exhibit properties of an entity. A strict one-to-one mapping of concepts to constructs should exist to prevent confusion.

Gemino and Wand (2005) found that decomposition of entities with optional properties into separate entities with mandatory constraints resulted in better user understanding.

Whilst decomposition with a one-to-one mapping supports reasoning and discrimination of concepts, the caveat to this is that too many different constructs may cause cognitive overload for the user. Koning et al. (2002) recommend a maximum of 6 different constructs per diagram. This limitation reinforces the role of abstraction in supporting complex modelling.

4.3 Layout for data warehouse modelling

Layout that directly represents the problem domain promotes inference and reasoning (Larkin and Simons, 1987). However, Kulpa (1994) cautions that the *emergent*

properties resulting from layout manipulation can be a mixed blessing. Kulpa demonstrates that sometimes the inferences suggested by a particular layout may in fact be erroneous.

The layout of the data model should allow a more direct representation of the problem domain without leading the user to make incorrect inferences. Koning et al. (2002) give a number of guidelines that could help minimise such problems. They recommend that objects of the same type should be the same size within individual diagrams and sets of related diagrams. This avoids incorrect inferences about the importance or relevance of same-type objects. With respect to object layout, the guidelines recommend object placement on horizontal and vertical lines. A non-uniform layout may lead to unwanted inferences.

Layout of text in relation to graphical elements can also influence cognitive load. Sweller et al. (1990) found a detrimental impact on performance of instructional materials where explanatory text and diagrams were poorly integrated. The authors reasoned that the lack of integration placed a high cognitive load on users. Switching focus between text and diagram in different locations was the likely cause of this load.

Automatic layout algorithms have been the subject of a number of recent research papers (Purchase et al., 2002; Gutwenger et al., 2003). These algorithms focus on the optimal placement of objects relative to one another and the organisation of connectors that represent the relationships between objects. Purchase el al. (2002) conducted an empirical study that concluded minimisation of bends and crossed edges were important aesthetics for users.

Koning et al. (2002) cautions that automatic layout may distort the natural hierarchical relationships in the model. It is therefore important to evaluate whether a given algorithm supports the properties of the domain. DW models should emphasise the semantic properties discussed in Chapter 3. These include:

- Subject-oriented nature of the DW
- Hierarchical data relationships
- Differentiation of conceptual constructs
- Clarity of relationships

4.4 Interaction and reasoning with data warehouse models

Scaife and Rogers (1987) comment that opportunities for external manipulation of the model aid the formulation of productions. Koning et al. (2002) concur with this view. In their synthesis of diagrammatic properties, they recommend that users be encouraged to look at the diagram and asked thought provoking questions about it.

Atkins and Patrick (1998) claim that their NaLER technique may also assist users interacting with a data model. The technique encourages the use of structured sentences to promote understanding of the data model.

Parson (2003) evaluated another technique to promote reasoning with data models. This study considered data model integration and found that local schema verification was superior where conflicts existed between models. In contrast, global schemas were superior when the models contained complimentary information. Koning's guidelines (Koning et al., 2002) recommend users be given the opportunity to compare old and new versions as an aid to visual reasoning. These techniques should assist in the understanding of an integrated DW schema.

Gutwenger el al. (2003) demonstrated that colour could assist in reasoning with diagrams. The authors proposed an automatic graph layout algorithm with colour used to differentiate class and inheritance hierarchies. Koning et al. (2002) provide further guidelines on the use of colour recommending different shades of non-saturated colour; this supports black-and-white printing, colour blindness, and avoids distracting the user by over emphasising a particular object.

4.5 Data warehouse cognitive principles

Table 2 Data warehouse modelling cognitive requirements

Data warehouse category	Sub Category	Concept / Reasoning	Citations *	Comment
Analysis / Inference	Decomposition	1:1 mapping	Burton-Jones and Weber	Avoid additional cognitive overload by not requiring additional reasoning
		Remove optional properties	Gemino and Wand (2005)	
		Legacy	Gregersen and Jenson (1999)	Avoid incorrect inference and support metaphor
	Abstraction	Different levels of detail	Chen et al. (1997) and Moody (1997)	To avoid overloading user with number of elements and/or different constructs
		Different perspectives	Parsons (2003); Degani (2004)	To emphasis different properties
		Limiting constructs to less than 6	Koning et al. (2002)	Practical guideline for determining abstraction levels
	Layout	Emergent properties	Larkin and Simon (1987)	Enhance direct representation
			Kulpa (1994), Koning (2002)	Avoid incorrect inference
			Gutwenger et al. (2003) and Koning et al. (2002)	Consistent layout of hierarchies
Analysis / Inference	Layout	Crossed edges/bend minimisation in relationship representation	Purchase et al. (2002)	Aesthetically pleasing to users/encourages use
		Text and picture integration	Sweller et al. (1990)	Reduces cognitive load minimising context switch
	Interaction / Reasoning	Explicit support for direct manipulation	Koning et al. (2002); Golfarelli et al. (1998)	Encourages user to interact and reasoning with the

Data warehouse category	Sub Category	Concept / Reasoning	Citations *	Comment
				diagram
		Explanatory text	Atkins and Patrick (1998)	NaLER technique - structured sentences help clarify meaning to user and avoid misinterpretation
		Local schema verification	Parsons (2003)	Allows users to contrast and compare
		Colour	Gutwenger et al. (2003) and Koning et al. (2002)	Aids differentiation of hierarchies and same type constructs

5. RESEARCH METHOD

5.1 Research methodology

The primary research will be a survey of existing DW data modelling techniques. The survey is a common approach used in the field of data modelling to analyse how current models compare to a proposed framework.

In Gregersen and Jensen (1999), the authors used a survey methodology to compare and contrast temporal extensions to the ER model formalism. They expounded the following benefits and outcomes of this approach:

- Obtain a comprehensive list of properties
- Characterise models according to those properties
- Consolidate ideas to facilitate ease of access for future research
- Allow a comparison using consistent terminology

Gemino et al. (2003) provide explicit direction on evaluating modelling techniques. They argue that empirical observation alone is insufficient to contrast the attributes of different modelling techniques. The method they propose encompasses three stages:

- Establish a benchmark based on an existing ontology
- Use the benchmark to find clear differences between the alternative models
- Study the implications of these differences by generating predictions on performance of the various grammars

This approach is recommended for evaluating the expressiveness of different models. However, the authors note that the cognitive properties of the grammar should be

tested empirically. In addition they hypothesise that increased expressiveness may lead to greater complexity with possible reduction in cognitive performance.

These previous studies lend support to the proposed methodology for this study. The commonality between them is the requirement to establish a set of criteria and evaluate the models based on these criteria. This study will follow a similar pattern to that recommended by Gemino et al. (2003). A benchmark will be established and used to compare the expressiveness of the existing data modelling techniques. In my study, literature on DW and data modelling will be used to create the benchmark. Blair et al. (1995) lends support for this approach to framework creation. Here the authors analysed existing techniques presented in literature. They then used these ideas to formulate a conceptually complete representation of requirements.

This study will also assess the usability of these data models. Usability correlates highly to predicted computational efficiency of the modelling representations. Gemino et al. (2003) argued this can only be performed with an empirical study. It is my contention that heuristics for diagrammatic representation can be derived from analysis of previous empirical studies on the subject. The literature review will therefore include an analysis of this body of knowledge.

Gregersen and Jensen (1999) provide direction on activities for surveying modelling techniques. Their survey first established a problem-domain scenario. The scenario encompassed the temporal properties they wished to compare and was used to construct a diagrammatic representation for each technique. This approach allowed both authors and readers to make direct comparisons of the surveyed models.

My study will mirror this approach. Once I have established a complete set of semantic properties, it will be possible to create a suitable domain scenario. Using the notation prescribed by each of the modelling techniques in the survey, a set of data models will be constructed for the problem domain. These data models will then form the basis for a comparison of semantic and representational properties.

Finally, the survey results can be used to propose extensions or modifications to existing approaches that should further enhance user understanding of the semantic content of the DW.

5.2 Conceptual data models in the survey

Table 3 Conceptual data models in survey

Year	Model Name	Full Name	Primary Reference	Supporting References
1998	DFM	Dimensional Fact Model	Golfarelli et al. (1998)	Golfarelli and Rizzi (1999)
1998	ME/R	Multi dimensional entity relationship model	Sapia et al. (1998)	
1999	starER	starER	Tryfona et al., 1999	
1999	CDWDM	Data warehouse conceptual data model	Franconi and Kamble (2004a)	Franconi and Kamble (2004b); Franconi, and Sattler, (1999)
2000	Husemann	Husemann	Husemann et al. (2000)	
2001	GOLD	Object Oriented multi dimensional data model	Trujillo et al. (2001)	Lujan-Mora and Trujillo (2003); Lujan-Mora (2005);
2002	YAM^2	Yet another Multidimensional Data Model	Abello et al. (2002)	Abello et al., 2006
2004	MultiDimER	MultiDimER	Malinowski and Zimanyi (2004)	Malinowski and Zimanyi (2006)

6. RESEARCH RESULTS

6.1 Survey method

The literature review identified eight graphical conceptual models. Between them they represented a broad array of modelling styles. Although the earlier models were first proposed almost a decade ago, graphical conceptual modelling for DW is still an area of active research with recent publications by Malinowski and Zimanyi (2006), together with updated and refined versions of previously proposed models (Abello et al., 2006).

I conducted the survey in three stages. The first stage considered each of the models and their properties. I then combined the synthesis of the semantic and cognitive properties of the models with more general findings on the requirements for DW modelling. This resulted in the more detailed criteria as presented in Table 4.

The second stage concerned developing a graphical representation using each one of the models. To achieve consistency a DW requirements specification was defined (see Appendices 1-4). The specification should be detailed enough so as to test the full expressiveness of each one of the models, but not so detailed that it became the object of study in its own right.

Development of the graphical models had two benefits. Firstly, the act of using the modelling notation facilitated a better consideration of the nuances of the model, its usability and any constraints. Secondly, it allowed for a more consistent approach to the assessment of the cognitive properties of the models. When constructing the graphical representations I was careful to follow any precedents for layout and style

(explicit and implicit) that might influence the final representation. The graphical representation should as far as possible reflect the spirit within which the original model was proposed. This did not mean that exactly the same set of diagrams was developed for each of the models. Instead, the domain scenario was used to the extent necessary to demonstrate the use of the models constructs. This is consistent with the approach taken by Gregersen and Jensen (1999).

The final stage concerned a second review of the articles relating to the surveyed models. With the benefit of a more detailed survey criteria and having spent time working with the models I was able to complete a full review of their semantic and cognitive properties (see Table 4).

6.2 Surveyed models - introduction and diagrams

6.2.1 Dimensional Fact Model (DFM)

The Dimensional Fact Model (Golfarelli et al., 1998) was the earliest published paper in the survey. The authors propose their conceptual model as part of a broader method for deriving a DW schema from operational data sources. It presents a custom notation for representing facts, dimension levels, and hierarchies. In addition the notation is capable of representing query patterns and shared dimensions across multiple facts.

Figure 1 Dimensional Fact Model - Billing fact and dimensions (custom notation)

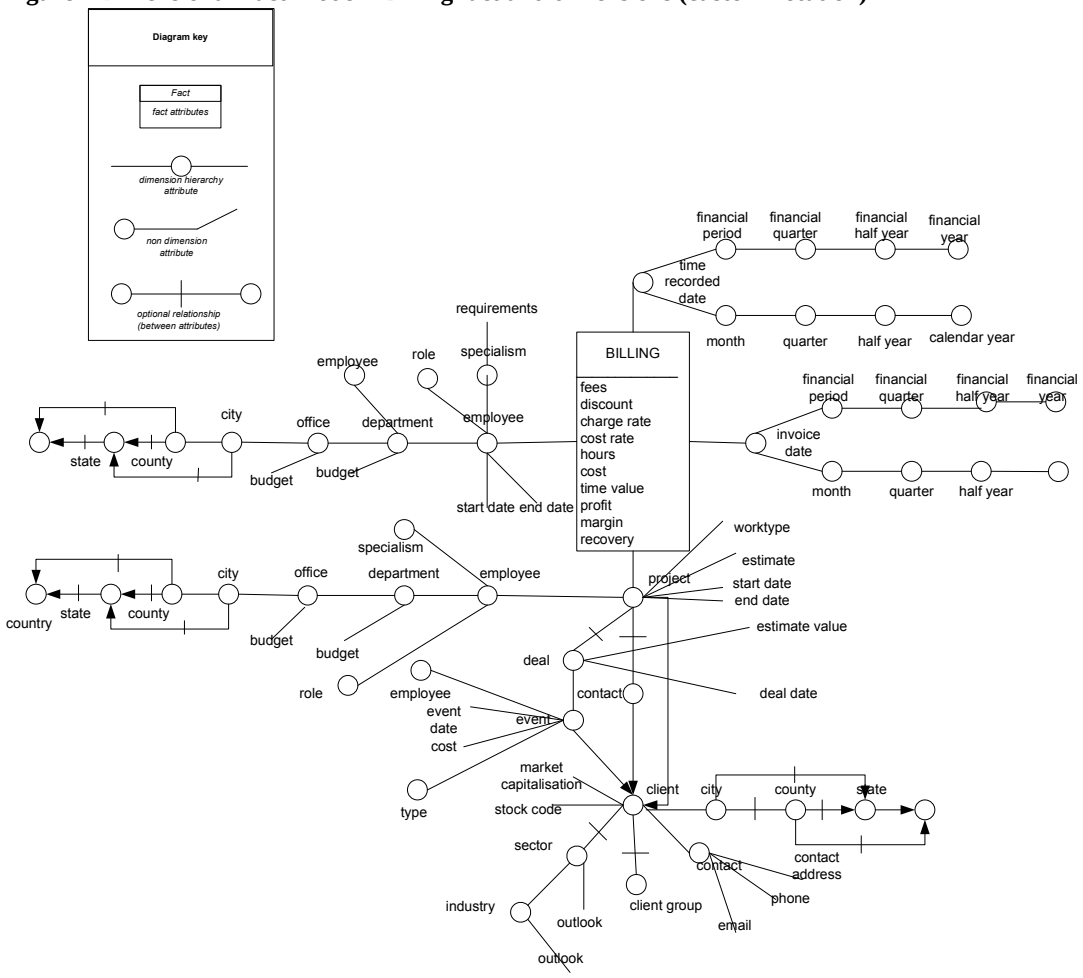
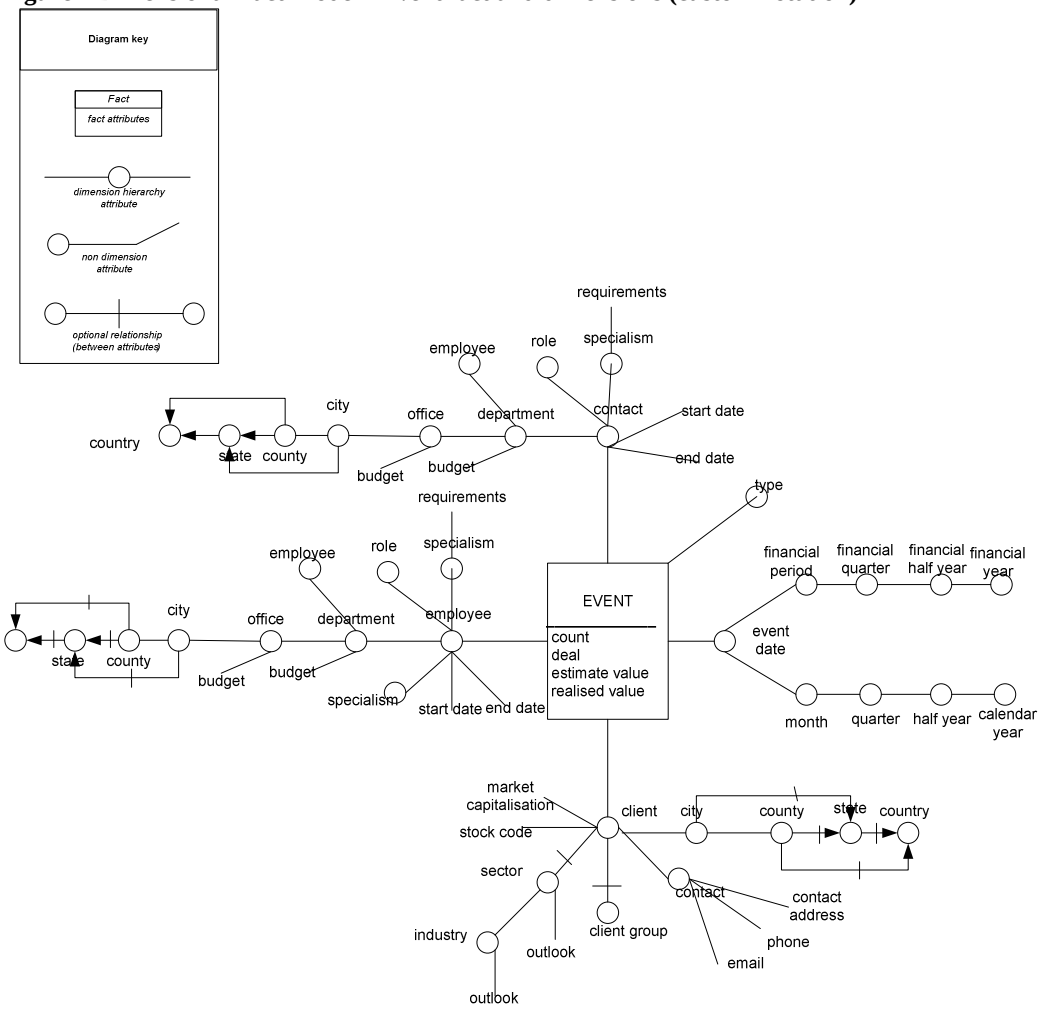


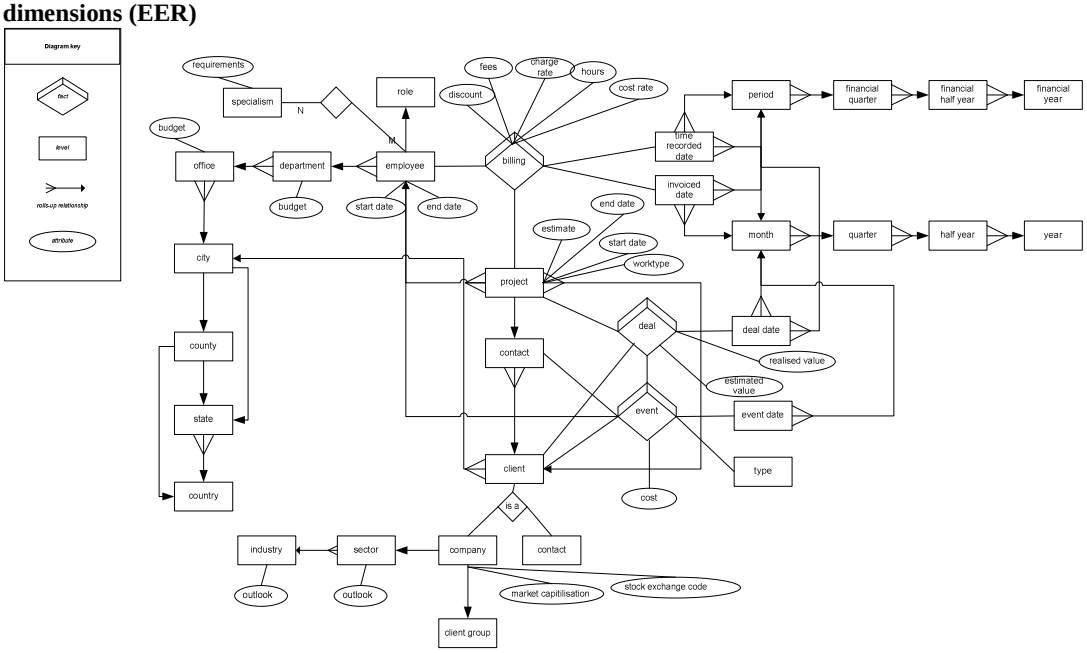
Figure 2 Dimensional Fact Model - Event fact and dimensions (custom notation)



6.2.2 Multidimensional Entity Relationship model (ME/R)

The Multidimensional Entity Relationship model (Sapia et al., 1998) introduces three new graphical constructs to the ER model (Chen, 1976). These specialisations of ER constructs allow for the explicit representation of facts, dimension levels, and the strict and complete *roll-up* relationship that commonly feature in multidimensional hierarchies. The additional constraints imposed on the extensions emphasise that this model deliberately restricts rather than extends the expressiveness of the more general ER data modelling formalism.

Figure 3 Multidimensional Entity Relationship Model (ME/R) - Billing, Deal Event fact, and dimensions (EER)



6.2.3 starER

starER (Tryfona et al., 1998) seeks to extend the expressiveness of the standard ER diagram. The model introduces a number of additional graphical constructs for representing facts, dimensions, measures, and various relationship types. The authors emphasise that the model can be used to represent both multidimensional and more general data structures.

Figure 4 starER - Billing fact and dimensions (EER)

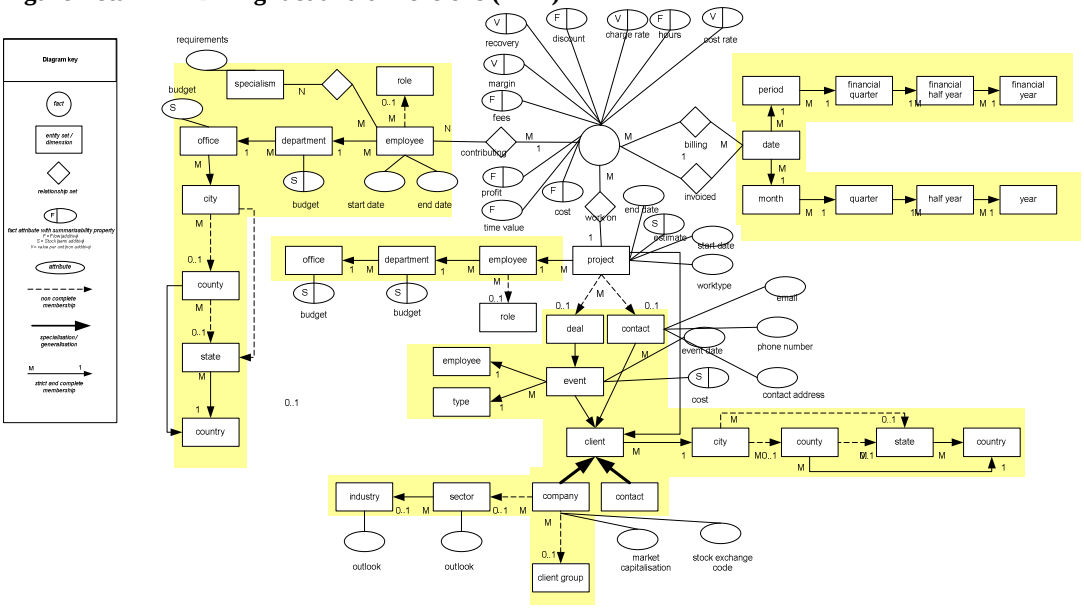
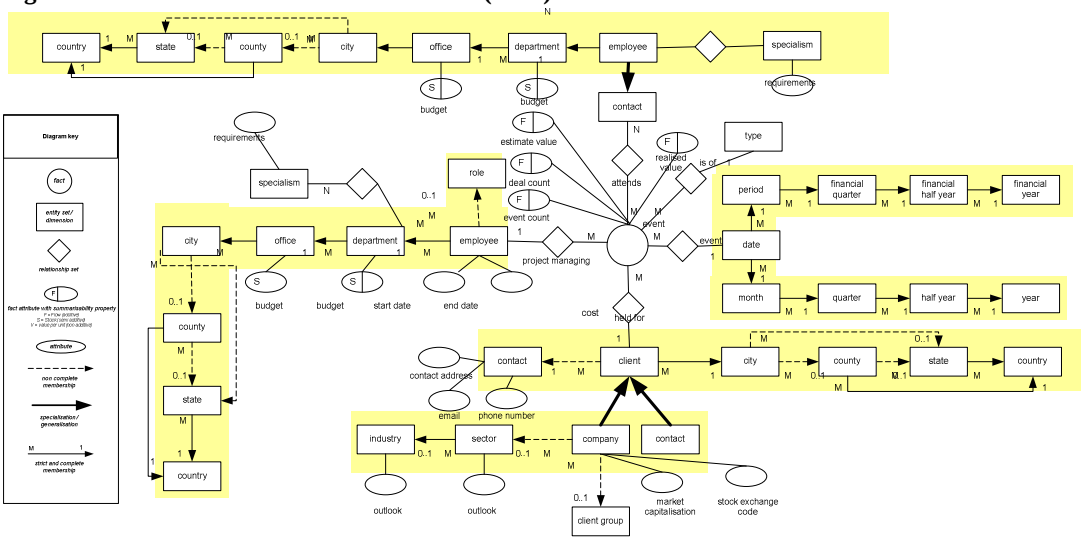


Figure 5 starER - Event fact and dimensions (EER)



6.2.4 Data Warehouse Conceptual Data Model (DWCDM)

The Data Warehouse Conceptual Data Model (Franconi and Kamble, 2004a) proposes two extensions to the ER model based on the concept of aggregated entities. The authors claim that this notation, combined with the semantics of the *GMD* data model, can be used to represent complex data structures as found in DW. The related papers are relatively brief in their consideration of how the graphical model can be used to

represent the DW schema as whole. However, the ability to explicate additional meaning from existing data structures is a powerful concept and should be considered in the field of DW.

The representations of the data model in Figure 6 & 7 (below) do not attempt to model the ‘Billing’ fact or ‘Project’ dimension as with the other examples. The authors state this would be achieved using a standard ER notation and so might be represented similarly to the enterprise data model (see Appendix 4). Figure 6 shows how the notation could be used to model the ‘Performance’ fact which is essentially a consolidation of data derived from other facts and dimensions. Figure 7 shows how a particular measure of the ‘Performance’ fact table might be calculated from aggregating the ‘Billing’ fact at the ‘Client’ level of the ‘Project’ dimension and the ‘Period’ level of the ‘Date’ dimension.

Figure 6 Data Warehouse Conceptual Data Model (DWCDM) - Performance fact with custom aggregation (EER)

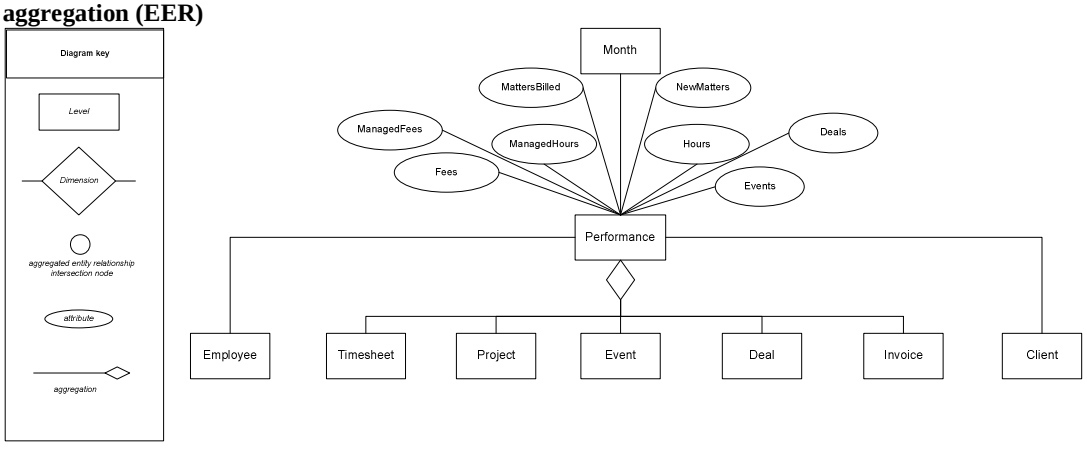
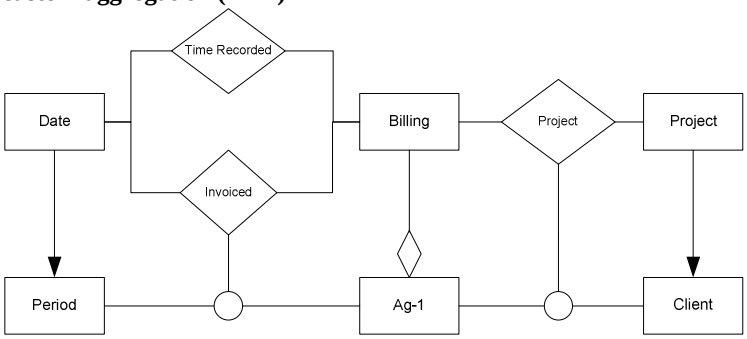


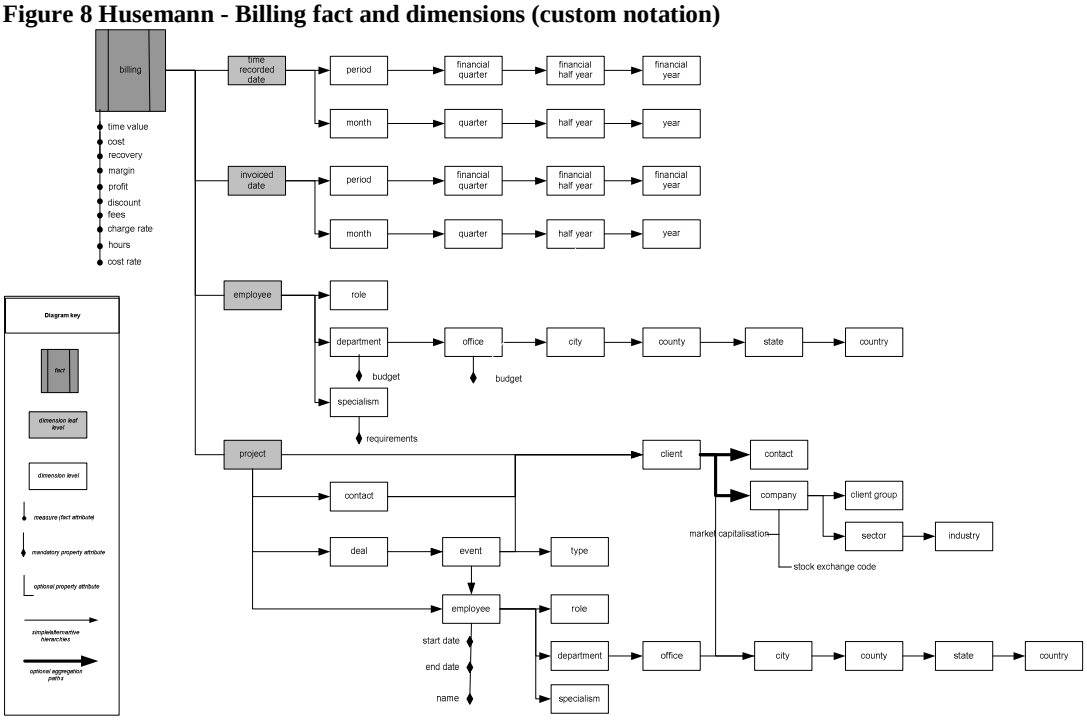
Figure 7 Data Warehouse Conceptual Data Model (DWCDM) - Billings by client with period custom aggregation (EER)



6.2.5 Husemann

Husemann's model (Husemann et al., 2000) uses a custom notation to represent the transformation of an operational data source into a multidimensional model. The authors' emphasis is on the actual process of deriving the multidimensional model from that of its source systems. They claim this is best achieved through the analysis of functional dependencies between fact measures and dimensions. The model defines graphical constructs for facts, terminal levels/dimensions and dimension levels as well as three types of attributes.

Figure 8 Husemann - Billing fact and dimensions (custom notation)



6.2.6 GOLD

The GOLD model originated in work by Trujillo and Palomar, (1998) and fully developed in a thesis by Lujan-Mora (2005).

The conceptual model uses the UML language and stereotype facility to define sub classes of existing UML constructs. The model allows direct representation of facts, dimensions, dimension attributes, and strict-and-complete relationships used for defining well-formed hierarchies. In addition to formally defining specialised constructs, the authors also proposed custom icons to differentiate the stereotypes from their base classes.

Figure 9 GOLD Level 1 - Star schema package dependency model (UML package with custom icons)

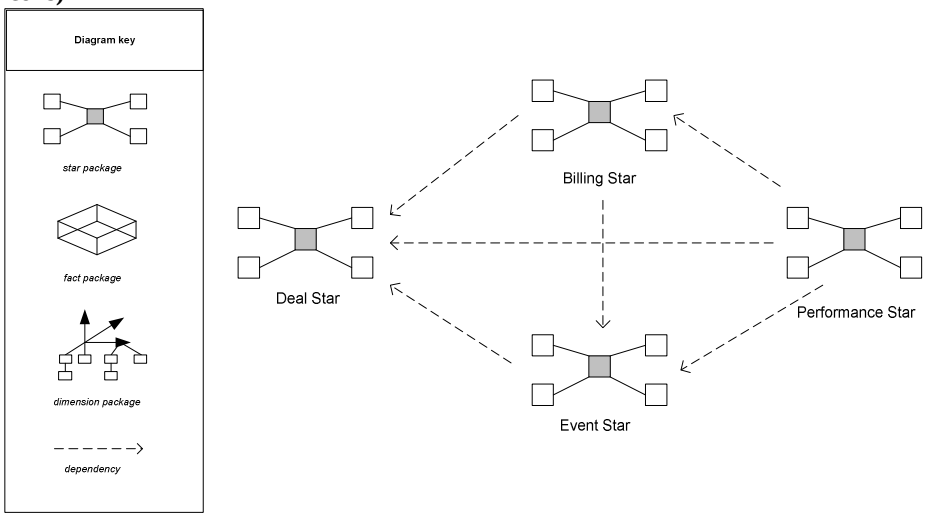


Figure 10 GOLD Level 2 - Billing fact package dependency model (UML package with custom icons)

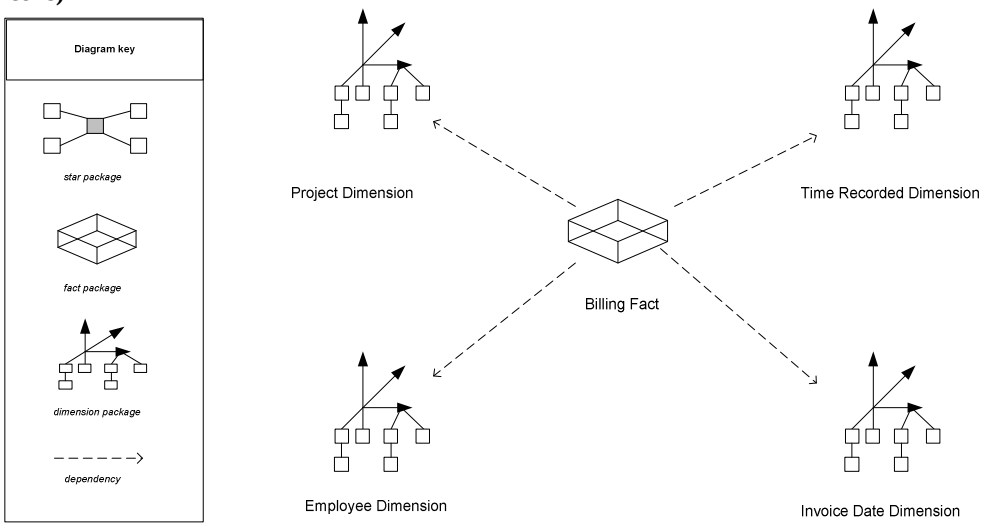
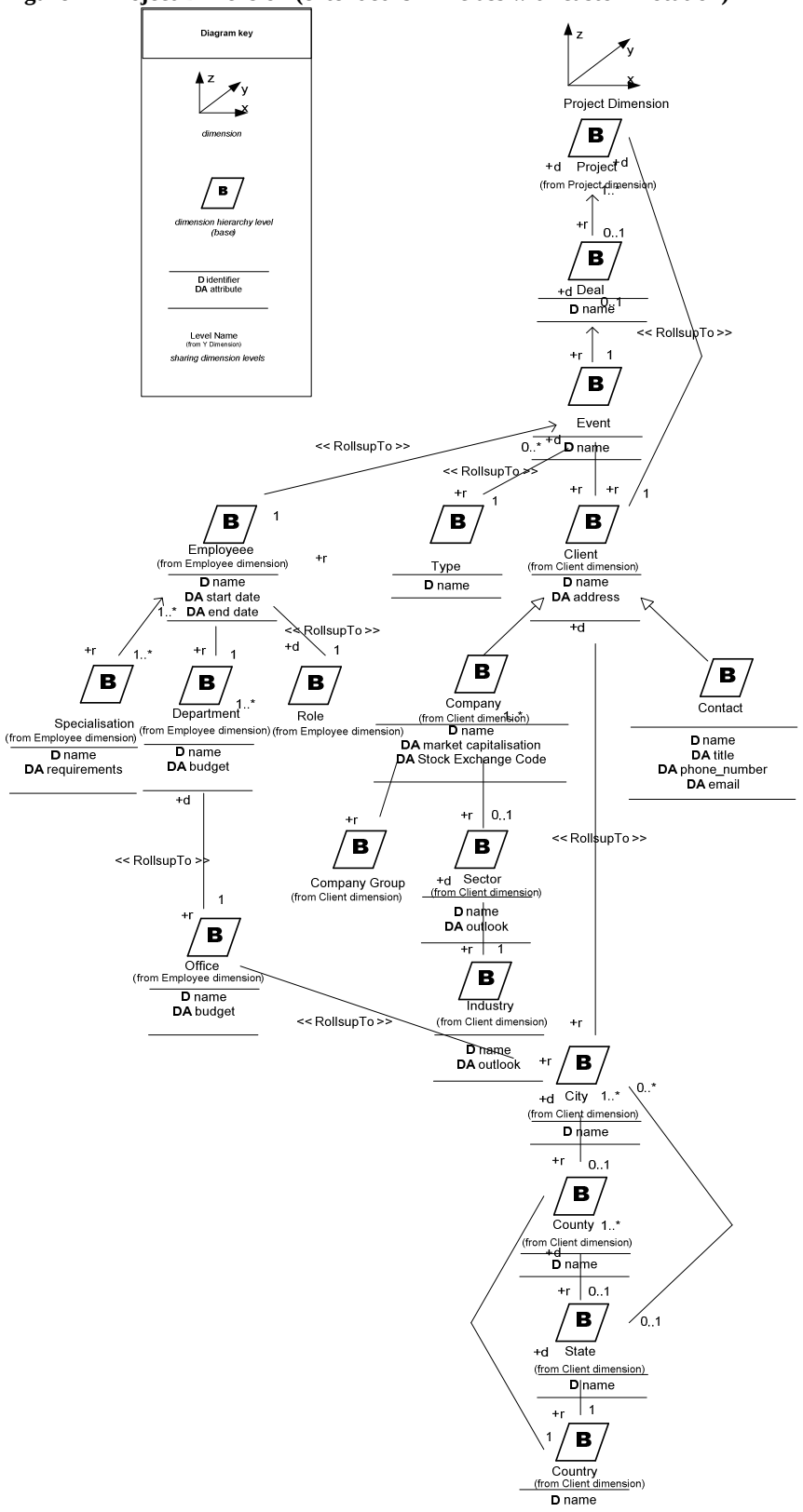


Figure 11 Project Dimension (extended UML class with custom notation)



6.2.7 YAM²

YAM² (Abello et al., 2006) is based on a very detailed consideration of the semantic properties of multidimensional modelling. The model defines three levels of abstraction to control complexity. For each level, the authors systematically explore the set of valid constructs and the inter-relationships between these constructs.

The authors extend UML by defining sub classes to represent the necessary semantics of the multidimensional model more explicitly. In addition to considering facts, dimensions, levels, and attributes, the model also introduces the concepts of Cell and Base constraints that allows for more flexible modelling of the relationship between fact measures and dimension levels.

Figure 12 YAM² Upper Level - Star schema package dependency model (extended UML package/class)

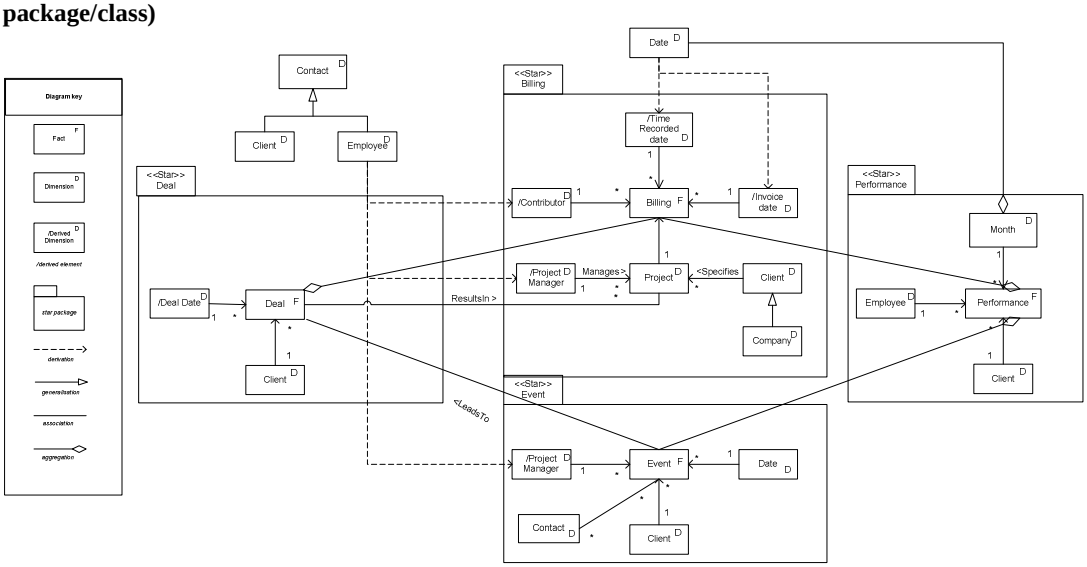


Figure 13 YAM^2 Intermediate Level - Billing fact dimensions (extended UML package/class)

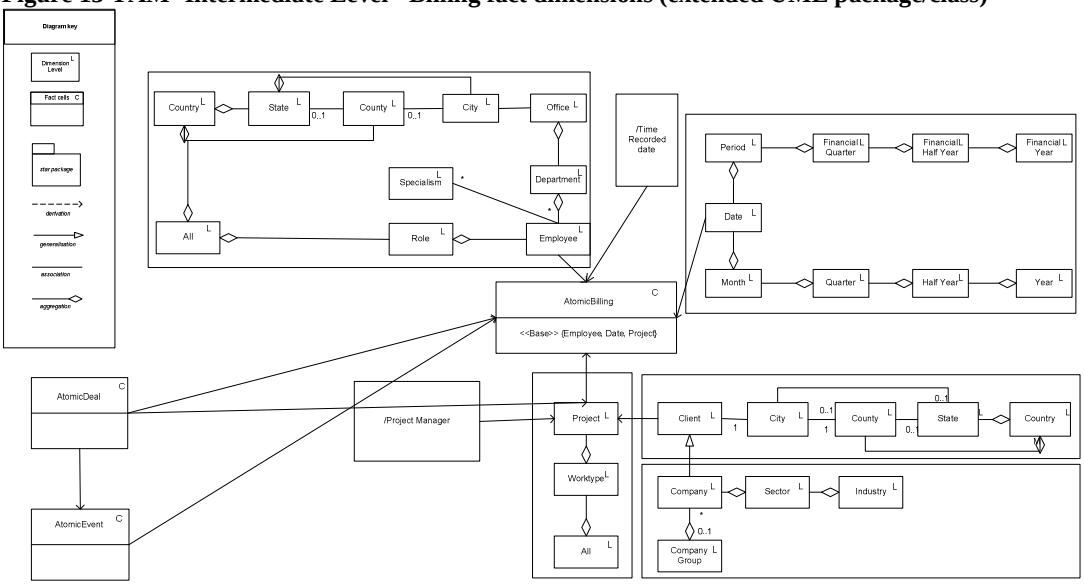
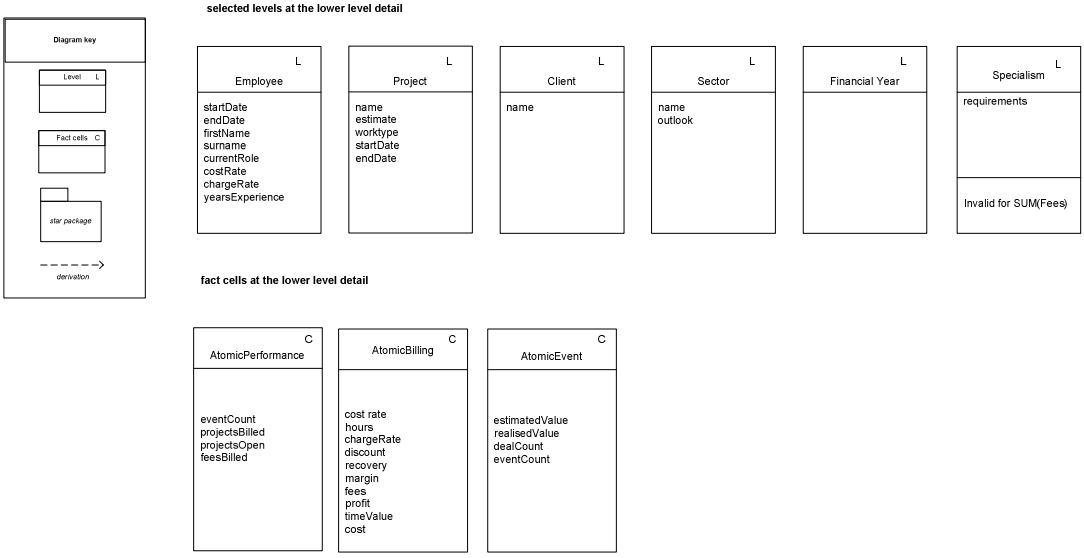


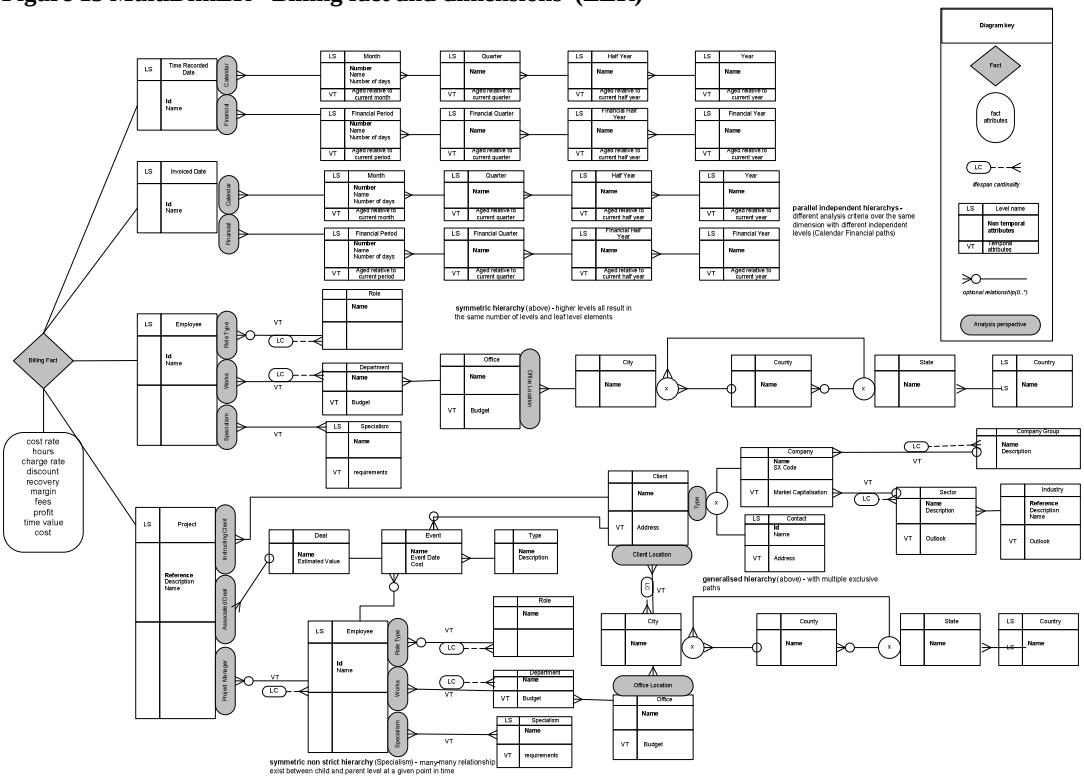
Figure 14 YAM^2 Lower Level - Dimension attribute level (extended UML class)



6.2.8 MultiDimER

Malinowski and Zimanyi (2004) first proposed the MultiDimER model as part of a detailed study into classification of hierarchies in multidimensional modelling. Malinowski and Zimanyi (2006), extends the model to include temporal properties The graphical notation combines elements from existing formalisms. Relationships types are similar to those used in ER modelling. A multi box format, more commonly found in UML class diagrams, represents entities.

Figure 15 MultiDimER - Billing fact and dimensions (EER)



6.3 Survey Results

Table conventions

Explicit Property explicitly considered and supported

Implicit Property not explicitly considered but may be supported by underlying modelling language

Partial Property partially supported

N/A Property not considered and may not be supported

Excluded Property considered but judged outside the scope of the conceptual data modelling technique

Table 4 Data warehouse model survey results

Models	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
General								
Style/notation	Other	ER	ER	ER	Other	UML	UML	ER/UML
Year first proposed	1998	1998	1999	1999	2000	2001	2002	2004
Type *(DW Method or Standalone Conceptual Model)*	DW Method	Standalone	DW Method	Standalone	DW Method	DW Method	Standalone	Standalone
Diagrams								
Figure references	1, 2	3	4, 5	6, 7	8	9, 10, 11	12, 13, 14	15
Billing Fact	Figure 1	Figure 3	Figure 4		Figure 8	Figure 9, 10	Figure 12, 13	Figure 15
Event Fact	Figure 2	Figure 3	Figure 5			Figure 9	Figure 12	
Deal Fact		Figure 3				Figure 9	Figure 12	
Performance						Figure 9	Figure 12	
Project Dimension	Figure 1	Figure 3			Figure 8	Figure 11	Figure 13	Figure 15
SEMANTIC PROPERTIES	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
SUBJECT ORIENTED								
Facts								
Measures	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Derived measures	N/A	Excluded	N/A	Explicit	N/A	Explicit	Explicit	N/A
Dimensions								

Models	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
Derived dimensions (roll playing)	N/A	N/A	N/A	Explicit	N/A	Explicit	Explicit	N/A
Levels	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Attributes	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Derived Levels	N/A	N/A	N/A	Explicit	Explicit	Explicit	Explicit	N/A
Hierarchies								
Hierarchy properties include: (leaf, root, levels, path, path length)								
Symmetric	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Asymmetric	Excluded	Excluded	N/A	N/A	Excluded	N/A	Excluded	Explicit
Generalised	N/A	Implicit	Explicit	Implicit	Explicit	Explicit	Partial	Explicit
Non covering *(ragged)*	Explicit	Excluded	Implicit	Explicit	Explicit	Explicit	Excluded	Explicit
Non strict	Excluded	Excluded	Explicit	Implicit	Excluded	Explicit	Explicit	Explicit
Multiple *(one analysis criteria:many non exclusive simply hierarchies)*	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Implicit	Explicit
Alternative	Explicit	Explicit	Explicit	Explicit	Partial	Explicit	Implicit	Explicit
Relationships								
Types								
Aggregation	Implicit	Implicit	Explicit	Explicit	Implicit	Explicit	Explicit	N/A
Association	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Derivation	N/A	N/A	N/A	N/A	N/A	Explicit	Explicit	N/A
Flow	N/A	N/A	N/A	N/A	N/A	Implicit	Explicit	N/A
Generalisation	N/A	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Membership/rolls-up up (Directed Acyclic Graph)	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Cardinality								
Fact - Dimension	1:N	1:N	M:N	M:N	1:M	M:N	M:N	1:M
Level:level	1:N	1:N	M:N	M:N	1:M	M:N	M:N	M:N
Flexibility								
Interchange of levels, dimensions	Explicit	Explicit	N/A	Explicit	N/A	Implicit	Implicit	N/A
Interchange of levels (dimensions) and facts (summary attributes)	N/A	N/A	Explicit	Explicit	N/A	N/A	Explicit	N/A
Multi grain measures within fact	Explicit	N/A	N/A	Explicit	N/A	Implicit	Explicit	Implicit
INTEGRATED								
Constraints								
Granularity	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Data type	N/A	Implicit	Implicit	N/A	N/A	Implicit	Implicit	N/A
Application constraints	N/A	N/A	N/A	N/A	N/A	N/A	Implicit	N/A
Business rules	N/A	N/A	N/A	N/A	N/A	Explicit	Implicit	N/A
Aggregation constraints								
Fully Additive	Explicit	Implicit	Explicit	Partial	Explicit	Explicit	Explicit	N/A

Models	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
Semi Additive	Explicit	N/A	Partial	Partial	Explicit	Explicit	Explicit	N/A
Non Additive	Explicit	N/A	Explicit	Partial	Explicit	Explicit	Explicit	N/A
Other calculation constraints/expressions								
Spreading *(root to leaf value allocation)*	N/A	N/A	N/A	Implicit	N/A	N/A	N/A	N/A
Full measure-dimension aggregation constraint matrix	N/A	N/A	N/A	Implicit	Explicit	N/A	Explicit	N/A
Mapping from source system	Explicit	Excluded	N/A	N/A	Explicit		N/A	Partial
Multiple fact integration	Explicit	Explicit	Partial	Explicit	N/A	Explicit	Explicit	N/A
Ambiguity/Uncertainty								
Constructs available in model	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	N/A
Mechanisms for handling								
Fuzzy constraints	N/A	N/A	Partial	N/A	N/A	N/A	N/A	N/A
Multi faced attributes	N/A	N/A	N/A	N/A	N/A	N/A	Implicit	N/A
TIME VARIANT								
Time classification								
Measures (fact attributes)								
Lifespan	N/A	N/A	Excluded	N/A	N/A	N/A	Explicit	Explicit
Valid time	Implicit	Implicit	Explicit	Implicit	Implicit	Implicit	Implicit	Explicit
Transaction time	Implicit	Implicit	Implicit	Implicit	Implicit	Implicit	Implicit	Explicit
Data warehouse load time	Implicit	Implicit	Implicit	Implicit	Implicit	Implicit	Implicit	Explicit
Attributes								
Lifespan	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Valid time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Transaction time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Data warehouse load time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Dimensions/level/entities								
Lifespan	N/A	N/A	Excluded	N/A	N/A	N/A	Explicit	Explicit
Valid time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Transaction time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Data warehouse load time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Relationship cardinality								
Snapshot	Partial	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit	Explicit
Lifespan	N/A	N/A	Excluded	N/A	N/A	N/A	Implicit	Explicit
Valid time		N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Transaction time		N/A	Excluded	N/A	N/A	N/A	N/A	Explicit
Data warehouse load time	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Explicit

Models	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
Time lag	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	Implicit
Sample Period	N/A	N/A	Excluded	N/A	N/A	N/A	Explicit	Implicit
Sample frequency	N/A	N/A	Excluded	N/A	N/A	N/A	N/A	N/A
Precision	N/A	N/A	Excluded	N/A	N/A	N/A	Partial	N/A
Volatility	N/A	N/A	Partial	N/A	N/A	N/A	N/A	Explicit
COGNITIVE PROPERTIES	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
Specific guidance on cognitive considerations	No	No	No	No	No	Yes	Yes	No
ANALYSIS								
Decomposition								
1:1 mapping (concept:graphical construct)								
Star	No	No	No	No	No	Yes	Yes	No
Fact	Yes	Yes	Yes	No	Yes	Yes	Yes (label)	Yes
Fact measure	Yes	No	Yes	No	Yes	Yes	Yes	No
Dimension	No	No	Partial	Yes	Partial	Yes	Yes (label)	No
Level	Yes	Yes	Yes	Yes	Yes	Yes	Yes (label)	Yes
Hierarchy	No	No	No	No	No	No	No	No
Hierarchy analysis criteria	No	No	No	No	No	No	No	Yes
Hierarchy classification *(Temporal, spatial, organisational)*	No	No	No	No	No	No	No	No
Level Attribute	Yes	No	No	No	Yes	Yes (label)	Yes	Yes
Relationships 1:1 mapping (concept:construct)								
Drill down / rollup *(representing a strict complete relationship)*	Yes	Yes	Yes	No	Yes	Yes	Yes	No
Constraints (constraint type:graphical construct)								
Aggregation of measures	Yes	Excluded	Yes	Yes	Excluded	Yes	Yes	No
Optional properties	No	Yes	No	Yes	No	Yes	Yes	Yes
Supporting metaphors (*via established diagrammatic constructs*)	N/A	Explicit	Explicit	Explicit	N/A	Explicit	Explicit	Partial
Abstraction								
Number of graphical abstraction levels	1	1	1	1	1	3	3	1

Models	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
Count of other supporting documents					1			
Abstraction mechanisms								
Role playing dimensions	N/A	N/A	N/A	N/A	N/A	Implicit	Explicit	N/A
Shared hierarchies	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Shared dimensions	Explicit	Explicit		N/A	N/A	Explicit	Explicit	
Shared levels (inter dimension)	N/A	Explicit	N/A	N/A	Excluded	Explicit	Explicit	Explicit
Shared levels (intra dimension)	Explicit	Explicit	N/A	N/A	Explicit	Explicit	Explicit	N/A
Multiple perspectives	Partial	N/A	N/A	Explicit	N/A	Explicit	Explicit	N/A
Layout								
Implied default layout style	star	star	star	network	hierarchical (left to right)	hierarchical (top to bottom)	star/network	hierarchical (left to right)
Label encapsulation	No	Yes	Yes	Yes	Partial	No	Yes	Yes
Likelihood of cross edges	Low	High	Medium	Medium	Low	High	High	Low
Likelihood of bends in edges	Low	High	Medium	Medium	Low	High	High	Medium
Emergent properties								
Separation of subject and context	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Analysis Hierarchy	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Dimension ordering	No	No	No	No	Yes	Yes	No	No
Dimension precedence (in terms of owning a level)	No	Yes	No	No	Yes	Yes	Yes	No
Level precedence	No	Yes	No	No	Yes	Yes	No	Yes
INFERENCE								
Interaction								
Text and picture integration								
Integrated diagram key	N/A	N/A	N/A	N/A	N/A	N/A	Explicit	N/A
Explanatory text mechanism		N/A		N/A	N/A	Implicit	Implicit	N/A
Graphical meta model	N/A	Explicit	N/A	N/A	N/A	Explicit	Explicit	Explicit
Integrated constraints		N/A		N/A	Excluded	Explicit	Explicit	N/A
Interaction and reasoning								
Query/aggregation patterns	Explicit	Excluded	N/A	Explicit	N/A	N/A	Explicit	N/A
Use of colour/shading	N/A	N/A	Yes	N/A	Explicit	N/A	N/A	Explicit
Local schema verification	Explicit	Excluded		N/A	N/A	Excluded	Explicit	N/A
Different perspectives	N/A	N/A	N/A	Explicit	N/A	Explicit	Explicit	N/A

Models	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
OTHER OBSERVATIONS	DFM	ME/R	starER	DWCDM	Husemann	GOLD	YAM^2	MultiDimER
Diagram preparation								
Custom symbols	Custom Basic Shapes	Custom ER extensions	Custom ER extensions	Extended ER	Custom Basic Shapes	Custom Icons	Custom UML	Complex Shapes
Graphical Construct Count (excl Relationship arcs)	3	3	5	4	6	5	5	4
Distinct graphical relationship types (excl text variations for cardinality)	3	3	4	3	2	4	5	6
Other symbols and keywords	2				0	6		6
Total constructs	**8**	**6**	**9**	**7**	**8**	**15**	**10**	**16**
Total constructs at given level	8	6	9	7	8	10	4	16

7. ANALYSIS AND DISCUSSION

7.1 Overview of survey results

The survey considered eight DW conceptual models (Table 3, and 4 *above*).

I grouped the semantic and cognitive properties in the survey as follows:

Semantic sub categories

- *Subject-oriented* - representing the domain in terms of facts, dimensions, hierarchies, and relationships in a flexible way
- *Integrated* - representing constraints and the impact of data integration
- *Time-variant* - representing the temporal properties of the DW

Cognitive sub categories

- *Decomposition* - the level of support for representing semantic concepts with distinct graphical constructs
- *Abstraction mechanisms* - support for abstraction techniques that should reduce the complexity of the models thereby increasing usability
- *Layout* - use of effective layout heuristics
- *Inference* - use of techniques to support making inferences from the model

Figures 1 and 2 show the relative support offered by each model for the survey categories. Ordering the models by first publication date helps to highlight any evolutionary trends.

Figure 16 Relative support for semantic properties in the survey

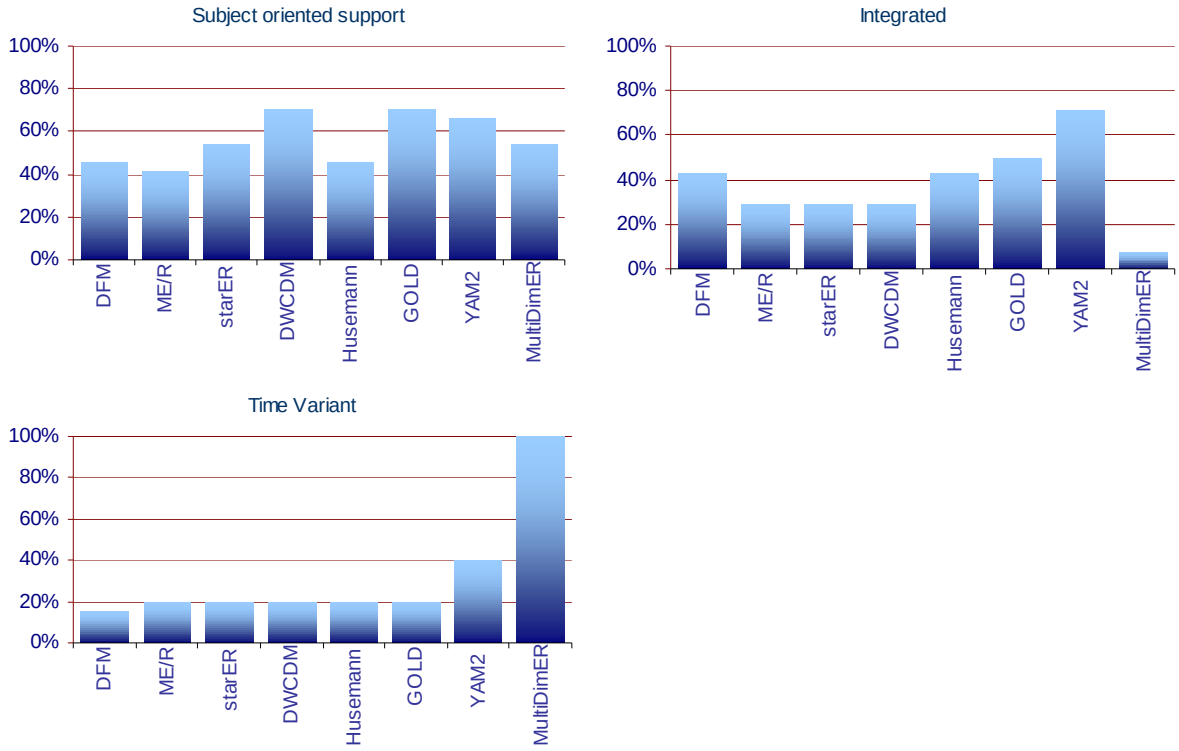
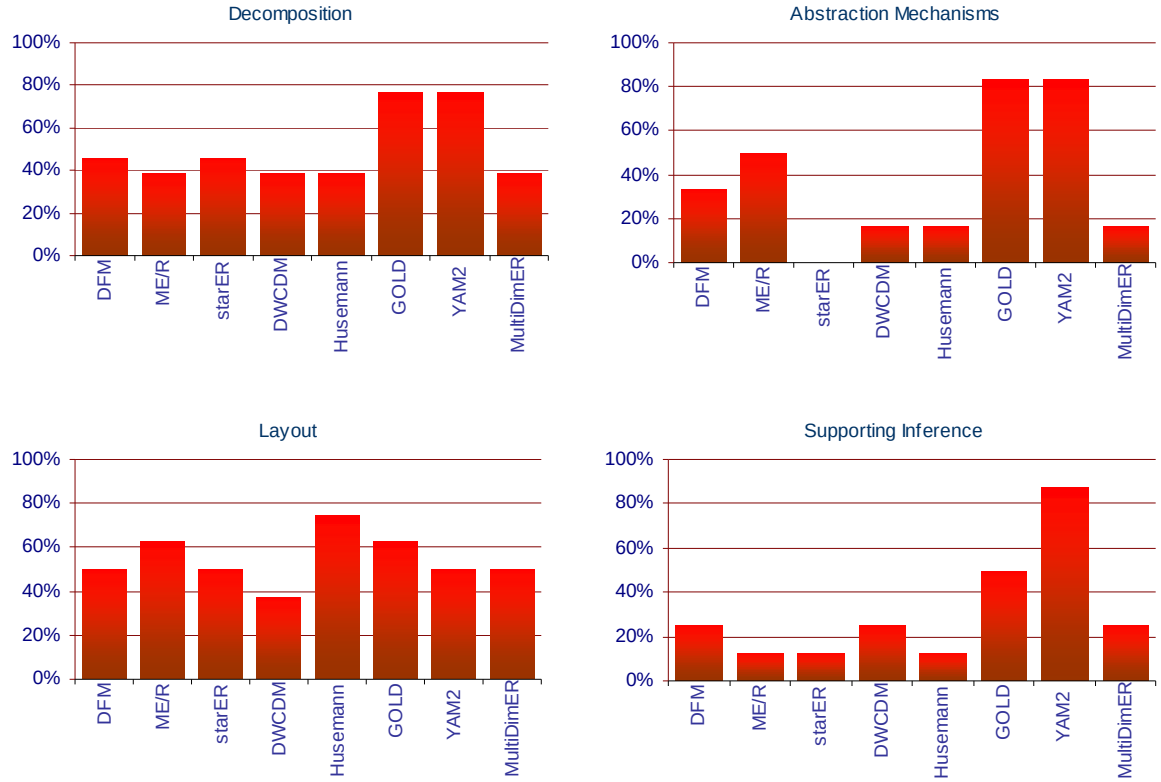


Figure 17 Relative support for cognitive properties in the survey



An analysis of the survey results led to the following observations:

- Consensus in the need to support a subject-oriented approach to DW modelling
- Increasing recognition in the later models for the need to support temporal properties and the effects of data integration
- Increasing awareness of the need for greater decomposition and abstraction
- Limited consideration of the impact of increasing the semantic richness of the model on the layout and clarity of the resulting diagrams
- Models are often strong in a particular category but no single model has strong support for all the categories

7.2 Format of survey results

When considering the semantic properties of the data models it was not always possible to say without qualification whether the model did support the particular property or not. This is because the majority of the models are based on a generalised data modelling formalism that may be able to support the requirement if so adapted. However, because we are interested in the usability of the modelling language, both for developers and users, there is a need to distinguish between the explicit definition of a construct and the mere possibility of its inclusion.

For the reasons outlined the properties were generally assessed using a five point scale where ‘Explicit’ offers the strongest support, and ‘N/A’ implies a lack of support. Where it was possible to be more definitive about a property existing then ‘Yes’ replaced ‘Explicit’ and meant the property was present, and ‘No’ replaced ‘N/A’ meaning the property was not present.

7.3 Semantic Properties

7.3.1 Subject-oriented - facts, dimensions, attributes and levels

There is consensus among the models on the need to bisect the data into facts and dimensions. Where the models differ is in the level of support offered for derived data. ME/R explicitly excludes derived measures on the basis they belong to a functional model and not a data model. Only DWCDM, GOLD and YAM^2 explicitly consider the need to support derived measures in the DW model.

The three models that explicitly support derived measures (DWCDM, GOLD, and YAM^2) also support derived dimensions. The ability to do this depends upon having an explicit graphical construct to represent a dimension. Derivation relationships or relationships edges with distinct descriptors represent the different roles a dimension can play. The CCL domain scenario highlights the importance this facility. Consider that the 'Employee' plays a number of roles in the organisation: contributor of work to projects; project manager; event organiser; and event attendee. In models where no derivation technique exists the levels associated with 'Employee' must be separately represented along each dimension (Figure 1), or complex joins between dimensions must be represented to show that the same data is being referenced (Figure 3).

Four of the models support derived levels. An example is Husemann et al.(2000) where *balanceClass* and *turnOverClass* levels are derived from attributes in the sources systems. Techniques like this emphasise that the DW should not be a simple re-representation of source systems. Where additional semantic information important to decision making can be derived from the underlying data models, it should be represented explicitly to increase understanding and facilitate new inferences.

In contrast to the position taken by Sapia et al. (1998), in ME/R I take the view that explicitly supporting derived data is crucial in a DW data model. All the data in a DW is effectively derived from one source or another and during the extract, transform and load (ETL) process there are likely to be a number of transformations to present a more business-oriented view of the data. The anomaly in the ME/R model is that it excludes derived measures but then includes levels like ‘Day’, ‘Month’, and ‘Year’ in the time dimension. These levels are derived from a date and would not normally be defined as standalone entities.

7.3.2 *Subject-oriented - hierarchies and relationships*

The hierarchies have been analysed using a criteria very similar to that presented by Malinowski and Zimanyi (2004). The ability to represent complex hierarchies depends mainly on the restrictions placed on relationships between levels. Models that permit the full range of cardinalities are capable of representing most hierarchical types identified in the paper. Earlier models tended to restrict relationships to those that could ensure strict-and-complete hierarchies. These hierarchies allow correct additivity of fact measures for each level in the hierarchy. It is important to identify where this behaviour exists; however, some commonly occurring organisational hierarchies do not follow such strict rules. Later modelling techniques have acknowledged this.

Asymmetric hierarchies were the least supported category in the survey with some of the models explicitly excluding them. MultiDimER demonstrated an asymmetric hierarchy but did not show how the resulting dimension would join to the fact. The reasons for restricting the model to support only some hierarchies were generally implementation specific or based on the assumption that a MD model need not

support those types of hierarchies. However, a conceptual DW model should not exclude relationships that occur in a domain due to implementation concerns. On the other hand, where complex hierarchical relationships are included there should be a way of quickly identifying the type and implications of the hierarchy from the model. In this respect, some of the earlier models may have an advantage. By limiting the valid relationship types and explicitly defining graphical constructs for these relationships, models such as DFM and ME/R do allow for a clear representation of the behaviour of hierarchies.

7.3.3 Subject-oriented - flexibility of the model

One problem with trying to model a DW based on the assumption that data can be separated into dimensions and facts is that it depends very much on the perspective of the modeller as to how a particular entity is classified. However, this does not mean the approach is at fault. Any form of data modelling ultimately leans towards an exercise in categorisation to achieve a better understanding of the domain. In the CCL domain scenario we can see that ‘Event’ can play the part a fact and be analysed using a number of pre-existing dimensions. However, it can also play the role of a dimension as seen in the ‘Billing’ fact where it is possible to analyse billings using the ‘Event’ level in the ‘Project’ dimension. Similarly, a particular entity can play the role of a dimension or a level. In the CCL ‘Billing’ fact model ‘Client’ is a level in the ‘Project’ dimension and a dimension in the ‘Event’ fact model.

This leads to the conclusion that DW modelling concerns modelling the domain from a number of perspectives. To this end the DW modelling technique should acknowledge the need for flexibility in the model and the ability of data to assume different roles depending on the context.

The need for flexibility is acknowledged explicitly by a number of the modelling techniques although they adopt different approaches. In the DFM a dimension is defined by the level connected to the fact. In this way a level is implicitly capable of becoming a dimension based on the granularity of the measures in the fact. In starER the authors acknowledge that although they create a conceptual distinction between facts and dimensions there is nothing to prevent the model being used to represent a given entity in both ways.

Flexibility can lead to additional complexity where there is an attempt to model the complete DW schema. YAM^2 uses derivation and association relationships to show that a dimension can be derived from a fact and a fact can be a level in a dimension. However even in a relatively simple scenario like CCL this can generate a complex network of interrelationships and dependencies (Figure 12). Similarly ME/R allows the representation of multi fact schemas but does not describe how the constructs would be represented where an entity plays the roles of both fact and dimension within the same schema. This ambiguity is illustrated in Figure 3 where the relationship between ‘Deal’ and ‘Project’ should be interpreted as being that ‘Deal’ is a parent level of ‘Project’. Instead it is more likely to be misinterpreted as ‘Project’ being a dimension of the ‘Deal’ fact.

7.3.4 Integrated - constraints

Granularity of a fact is a composite constraint incorporating a set of orthogonal dimensions and the level at which a fact measure is represented along each dimension. In the majority of the models this is derived by considering the set of dimension levels that are directly related to the fact. To give additional flexibility YAM^2 introduces

Cell and *Base* constructs that define the grain of a measure with reference to a set of levels from the dimensions associated with the fact. This allows for the definition of derived measures at different levels of granularity within the same fact.

Only GOLD explicitly considers the need to include business rules in the model. GOLD also introduces the concept of Business Models; subsets of the overall DW that model the domain from different perspective. GOLD suggests additional constraints be incorporated using the Object Constraint Language (OCL). Both GOLD and YAM^2 allow the inclusion of constraints using OCL and UML comments notation.

Aggregation constraints are given varying levels of support by the different models. DFM uses query patterns to show legal aggregation paths across multiple dimensions. Husemann's model defines the requirement for an additional matrix showing the type of operators that can be applied to each measure along each dimension. starER defines three aggregation constraint types using a modified attribute notation. The approach taken by Husemann is very comprehensive but has the disadvantage of not incorporating any aggregation constraints into the graphical model. starER graphical notation is simple and easy to understand but may not support more complex scenarios.

Although the models often consider it necessary to express how operators will act on measures when traversing a dimensional hierarchy from leaf to root, none consider the need to represent the reverse situation. It is not uncommon in dimensional modelling scenario to spread measures at a higher level of granularity across instances of lower level entities. If we extend the CCL scenario we might say that a fees budget is set at

‘Office’ level and is then allocated to the instances of the ‘Group’ and then ‘Employee’ based on some expression. To calculate % of budget achieved it is necessary to spread the budget measure defined at ‘Centre’ level and aggregate the fees measure defined at (‘Project’, ‘Employee’) level. This concept is another illustration of the blurred line that DW conceptual modelling may assume between a functional and a data model.

7.3.5 Source system integration

A method of implicitly introducing application and business rule constraints into the conceptual model is to try and preserve some mapping between the conceptual DW model and the data sources. This way the data can inherit the constraints of the source systems. These systems may be better understood by business users than an abstract representation within the model itself. The authors of ME/R argue that such a mapping is unnecessary because their model is requirements driven rather than data driven. However just because requirements have been established independently of sources systems does not preclude a representation of how these requirements will be fulfilled in relation to the source data. DFM and Husemann incorporate an explicit technique for deriving DW schema from source systems. However none of the models consider the possibility of representing some form of mapping to the source system in the final DW model as a means of defining implicit constraints.

One of the benefits of a DW environment is the ability to combine measures from different business processes across common dimensions. For this reason, integration of heterogeneous facts in DW conceptual model should be given explicit consideration.

starER gives some support by representing aggregate attributes in entities not explicitly defined as facts. DFM supports fact integration using schema overlap facts. GOLD and YAM^2 partition the DW conceptual model into three levels. At the higher level of abstraction, they define mechanisms for defining how different facts can share common dimensions. Husemann and MultiDimER do not consider fact integration.

Only starER considers the need for a DW modelling technique to allow for uncertainty in the data. The authors of starER argue that probability can be added to the diagram where the uncertainty relates to the existence or not of particular constructs.

YAM^2 may be capable of incorporating ambiguity in how data is represented. The use of derivation allows for a given entity to be represented from multiple perspectives.

7.3.6 Time-variant

The work by Malinowski and Zimanyi (2006) highlights the deficiencies of the other models when representing the temporal properties of the DW. All the models support the inclusion of a time dimension related to the fact table which allows fact-attributes to be analysed over time. In addition, YAM^2 includes the flow relationship to show lifespan of the facts and their evolution. starER rejects the need to represent temporal attributes of the model stating that this is logical consideration. This seems too generalised a statement. The logical design level is used to decide how things will be represented but it is first necessary to model what needs to be represented. Without due consideration to the temporal properties of the DW at the conceptual level there will be no way of knowing what needs to be represented at the logical level.

The CCL scenario provides a good example of the where it is necessary to consider the temporal properties of dimension data. In most of the models the relationship between ‘Employee’ and ‘Role’ is represented as one-to-many because at a given point in time each ‘Employee’ has one ‘Role’ and a number of ‘Employees’ may have the same ‘Role’. However, an ‘Employee’ can change roles over time and users of the DW will need to know whether this change is captured by the DW. If a history of relationships between entities is not captured over time then users of the DW will only be able to analyse historical measures against the current relationships. In the MultiDimER model a second relationship is defined between temporal levels that defines the cardinality of the relationship over time. From this, it can be ascertained that an employee’s role history is captured.

The MultiDimER model gives a detailed consideration of the temporal properties of all the main constructs used in data modelling of DW. This is consistent with the Inmon’s philosophy (Inmon, 1996) that an element of time should be attached to all data in the DW. It is also necessary if the conceptual model is to capture the requirements for implementation strategies such as Kimball’s slowly changing dimension (SCD) technique to capture temporal properties of dimensions.

7.4 Cognitive properties

Only YAM^2 and GOLD consider cognitive properties of their model. Both techniques define three levels of abstraction with the aim of controlling complexity and facilitating understanding.

7.4.1 Decomposition

The UML based models (YAM^2 and GOLD) use the package construct to represent star and dimension concepts. These concepts are only implicit in the other models. All the models except DWCDM define an explicit construct for a fact. However, DFM, ME/R, and MultiDimER have no one-to-one mapping between the dimension concept and a graphical construct. starER uses colour shading to show the boundaries of dimensions and Husemann uses shading to imply that the terminal dimension level represents a dimension.

Given the importance of hierarchies in understanding the semantics of a domain it is surprising that none of the models offers a construct to capture and encapsulate a hierarchy. The result is that a good opportunity to reduce complexity is missed. The figures in Chapter 6 illustrate that hierarchies relating to time, geographic location, and organisational structure reoccur in a number of orthogonal dimensions. Often we see duplication of the entire hierarchical within a diagram. GOLD and ME/R reuse levels in different dimensions. This approach has two problems. Firstly, in the case of ME/R (Figure 3) it leads to many crossed edges and this decreases the readability of the diagram. Secondly, it introduces the emergent inference that a particular level belongs to one dimension over another. It does not seem correct to say that ‘City’ belongs to the ‘Employee’ dimension and is being borrowed by the ‘Project’

dimension. MultiDimER goes some way towards a solution by introducing an analysis criteria construct represented by a rounded rectangle. This promotes reuse of levels within a dimension hierarchy. However, it fall shorts of encapsulating common hierarchies so that they can be reused intra and inter dimensions and fact schemas.

7.4.2 Abstraction

Only YAM^2 and GOLD explicitly define different levels of abstraction as part of the modelling technique. No guidance is given in the other models about how to handle complexity.

The need for an abstraction mechanism can be seen by considering the figures in Chapter 6. The CCL scenario is relatively simple, especially with respect to the number of attributes. Despite this, it was often impossible to represent all the attributes of each level without the diagram becoming incomprehensible. Semantically rich models like MultiDimER are most at risk of becoming difficult to reason with. It is necessary to enlarge Figure 15 to A3 paper size before the whole diagram is readable. When we consider that this is just for a single fact schema then it is clear that there is an issue of scalability. Even with an abstraction mechanism in place the authors of YAM^2 comment that they have excluded some details from their example diagrams to avoid complexity (Abello et al., 2006).

Leaving decisions of abstraction and representation to the modeller may provide the most flexible approach but it also increases the cognitive load on that person. Without explicit guidelines the layout of diagrams are less likely to be consistent. This increases the cognitive load on the consumers who have to make their own inferences.

The possibility of modelling the same data from multiple perspectives is partially considered by DFM using the schema-overlapping notation. GOLD proposes Business Models as a mechanism for defining different perspectives of the same data. YAM^2 handles this through derivation mechanisms.

7.4.3 Layout

The variation of layout styles used by the models highlights the need to define the inferences that should be promoted through graphical modelling of the DW. Kimball (Kimball and Ross, 2002) argues that business users find the concept of the dimensions unfolding around a central fact intuitive. This may explain the preference for a star layout in the some of the earlier conceptual models. The star layout gives prominence to the fact and therefore supports the inference of separation of contextual data from business process measures data.

However, the star layout is less appealing when representing hierarchies. Figures 1-5 reflect the suggested layout of their respective models by having hierarchies moving out in all directions from the fact. A hierarchy is defined in *Dictionary.com* as any system of persons or things ranked one above another. The definition implies the properties of order and direction. These properties are not emphasised by a layout that allows hierarchies to be represented in multiple directions. It is interesting to contrast these diagrams with the Husemann model in Figure 8. The consistent layout of each of the dimensions and associated hierarchies results in a more readable representation.

Following this logic, the GOLD methodology uses an appropriate mix of layout techniques. At a higher level of abstraction the dimensions unfold around the fact in a star like pattern (Figure 9). At the lower level of abstraction, where hierarchies are

represented (Figure 11), the layout assumes a hierarchical and directional layout style moving from top to bottom. Given that the instances of a strict and complete hierarchy form a tree structure this lends further support for a top-to-bottom layout, with the most granular leaf level at the head of the page and the root level at the foot.

Another area of disparity between the models concerns whether labels are enclosed within the shape representing the concept. From a modellers perspective it was more difficult to avoid ambiguity in the model where labels were not encapsulated. Examples of this can be seen in Figures 1, 4 and 11. Careful placement of the labels is necessary to avoid them being mistakenly related to another construct.

Crossed and bent edges were a hazard for all the models that attempted to show more complex intra/inter relationships between dimensions, dimension levels, and facts. DFM avoids crossed edges by restricting the representation of star schemas to relatively simple acyclic graphs. However, this is at the expense of representing the realities of complex enterprise data. Where a more comprehensive representation is attempted (YAM^2, ME/R) the result is a network of inter-relationships that tends to reduce readability. The only way to avoid crossed edges but also represent all the necessary relationship is to have multiple perspectives of the same data each emphasising different semantic properties.

7.4.4 Interaction

To interact effectively with a representation the consumer must first understand what is being represented. In a departure from my survey rule to follow all layout precedents, I have included a legend with each of the figures in Chapter 6 to facilitate reader understanding. However only YAM^2 actually includes a legend along side each

graphical model it presents. The other models define the constructs at various points during the article and then force the reader to retrace their steps when they come to examine the sample diagrams. This does not set a good precedent for modellers who will be presenting their graphical models to users who are generally unfamiliar with data modelling notations.

A graphical metamodel can also facilitate interaction with a diagram by allowing a more technical user to quickly assimilate how the constructs are derived and inter-relate. A graphical metamodel is included in the ME/R, GOLD, YAM^2 and MultiDimER techniques.

The models have very different approaches to the incorporation of query patterns and aggregation patterns as part of the conceptual model. On one hand ME/R excludes them completely stating that they are part of a functional model. In contrast the DWCDM is defined with the explicit purpose of representing custom aggregations. DFM uses query patterns to augment the model with legal aggregations. This mechanism also allows for the definition of aggregation constraints. YAM^2 similarly allows the representation of interesting aggregations and aggregation constraints in the lower level of the model.

7.4.5 Other diagrammatic properties

Little use is made of colour and shading in the models. However where shading is used it proves to be quite effective. In starER (Figures 4 & 5) the yellow shading brings clarity to the star structure by showing the boundaries of the different dimensions and how they are separated from each other. This effect can be contrasted with a ME/R (Figure 3) where no shading is applied. It is more difficult to distinguish

the dimensions in this diagram despite very similar constructs and modelling style to starER. In Husemann (Figure 8) and MultiDimER (Figure 15) the use of shading adds emphasis and supports the decomposition of the modelling concepts.

Some of the models include a method for deriving the DW conceptual model from the underlying source systems. This has the advantage of traceability back to source systems. The DFM includes such a methodology. This should allow a more informed verification of the final DW model by allowing the user to compare it to the original source system models.

7.5 Conclusions

DW modelling can be seen as a specialisation of data modelling (Abello et al., 2006). More general data modelling concepts like entities and attributes are further decomposed to add additional meaning to the representation.

Although DW modelling can be seen as a specialisation of data modelling, it may still be necessary to extend the expressiveness of a given formalism, if it does not give appropriate constructs to represent the core concepts in DW.

The basic philosophy of the DW approach is the integration of subject-oriented data in such a way that it can be analysed over time. A DW conceptual modelling technique should therefore be able to model the level of support that the implementation will have for these core concepts. In this way consumers of the DW data can make informed decisions about how to use the DW.

Existing DW data models are relatively strong in their ability to model data in a subject-oriented way. By introducing additional or specialised constructs for facts, dimensions, and levels, they are generally able to represent data from a more analytical perspective. This supports the DW role as a data source for management decisions.

The data models in the survey were not as strong in modelling the realities of data integration in the DW environment. It is almost inevitable that not all the data will integrated perfectly due to differences in implementation date, constraints and business rules, and temporal support.

In traditional database and application design, it is often possible to give a clear separation between the data model on one hand, and the operations on that data as represented in the functional model on the other. In a DW this distinction is not so clear because the data that enters the DW has already been processed by the source systems. We import both a data structure and data that is the result of the functions of the source systems. For this reason, DW conceptual model should be capable of defining this behaviour in terms of constraints so that DW users can understand the meaning of the data. This can be achieved either by defining those constraints in the final model or by a clear mapping from the DW model to the source systems such that constraints can be inferred by association.

Integration and transformation of data also presents the opportunity to represent existing data and relationships more naturally and introduce new data and relationships. Models that explicitly restrict the modelling of derived data place a heavier cognitive load on users who will effectively have to do some of the

integration themselves. Fortunately, this has been recognised in a number of data models in the survey.

With the exception of MultiDimER, the data models were relatively weak in modelling the temporal properties of the DW. A DW is a temporal database but the majority of the models only include explicit temporal support for the modelling of fact measures.

The models gave very limited consideration to cognitive principles when presenting example diagrams. Two of the models in the survey offer an abstraction mechanism but none of the models explicitly discusses why they have chosen a certain layout style over another.

The semantic requirements of the DW reveal the properties of a DW that can be represented more effectively using a graphical notation:

- Decomposition of the domain into analysis criteria (dimensions) and business process or events (facts)
- Representation of hierarchical structures within the data

The models generally use decomposition to separate fact and dimension data. However, though some of the articles acknowledge the existence of generic and recurring hierarchies, they miss the opportunity to model hierarchies as a separate construct. This would have the benefit of simplifying the resulting models by reducing the number of shapes in the diagram. It would also allow the construct to be labelled with properties specific to the hierarchy.

Although the data models do not incorporate many of the semantic properties identified by the survey, the figures in Chapter 6 demonstrate that the diagrams can become quite complex even with a simple domain scenario. The models that did incorporate abstraction mechanisms focused on representing the DW at different levels of detail. There is arguably a case for representing multiple perspectives at each level of detail so as to emphasise different properties rather than trying to incorporate everything into a single diagram.

Whilst proponents of existing models may argue that the modeller has the choice to include or exclude certain properties when using their model, this assumes the modeller will have the time to consider the abstractions that will be most effective and which properties to group together.

Data modellers may find it more helpful to be given a set of DW model templates that incorporate guidance on layout, content and perspective. They can then choose which of these templates are appropriate for their particular scenario. Similar to design patterns used in object-oriented design, the templates would represent an abstraction that gives the modeller a well thought out solution to a particular problem, but without introducing implementation specific detail.

7.6 Limitations of the survey

The survey results rely predominately on my personal observations so there is always the chance that a construct may be misinterpreted or a point missed. The act of creating a diagrammatic representation of the CCL scenario for each of the models was the best way to minimise this risk. This forced a more rigorous consideration of

the nuances of each model. I also studied each article in detail several times before and after the modelling exercise with the aim of identifying any previous omissions.

Although the survey assesses the data models against a predefined set of criteria it is difficult to derive a robust quantitative indication of which is the best model. It is possible to produce a count for each model of the number of properties supported. Although this may be legitimate within a survey category it is less justifiable for the survey as a whole because it assumes that all the properties are of equal weight.

The domain scenario is another potential source of bias. The CCL scenario was designed to encapsulate the main themes of integration, analysis and temporal properties. However, it might by chance happen to emphasise the positive attributes of one model over another.

Despite these limitations, I would argue that it was an appropriate mechanism for assessing the models at this time; DW modelling is still a relative immature field. The variations of modelling styles, notations, properties, layout and emphasis revealed by the survey suggest that more high-level work is required in this area to define or at least narrow the DW problem space.

8. DATA WAREHOUSE CONCEPTUAL MODEL WITH EXPLICIT DIAGRAMMATIC CONVENTIONS

8.1 DWGraph – A data warehouse graphical conceptual model

DWGraph is a set of templates with a new custom graphical notation. The underlying semantics are based on the ER model but the notation includes additional constructs that allow a closer representation of the problem domain.

Each template includes a number of shaded regions that give the modeller explicit guidance on where to place objects. Using the templates, the modeller is able to represent the DW from multiple perspectives. Each perspective uses a layout that results in a readable representation that, where appropriate, facilitates grouping of common objects, hierarchical structures, and relationships between constructs. The set of templates presented below should be capable of modelling most of the semantic properties of the DW. However, this does not mean new templates cannot be added to include new perspectives if this was required for a particular implementation.

DW development is generally an iterative process. The templates are presented in the order they might first be used within a given iteration. Section 8.2 considers each template and how it contributes to a complete DW conceptual model.

8.2 DWGraph templates

After DW requirements have been established the first step is to identify relevant data sources and the high level constraints on those systems. Figure 18 (below) models the relationship between the data sources and DW at a high level of abstraction.

Figure 18 DWGraph - System Perspective

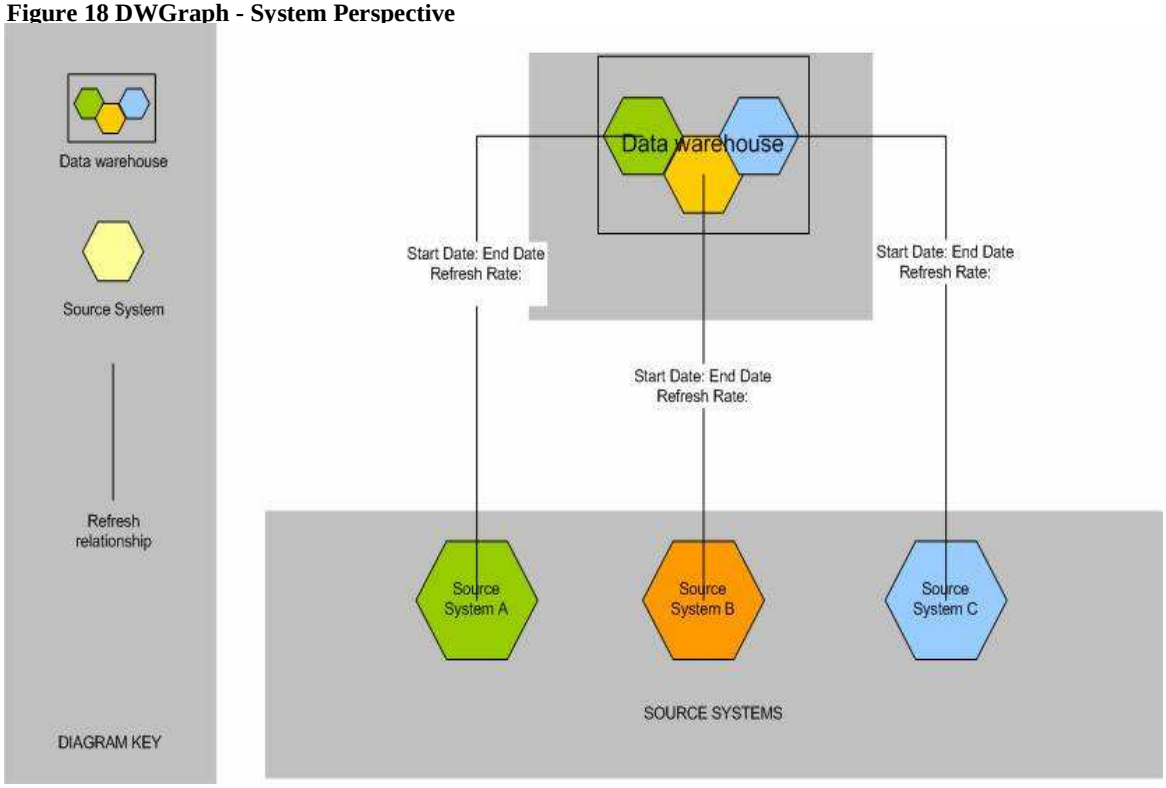
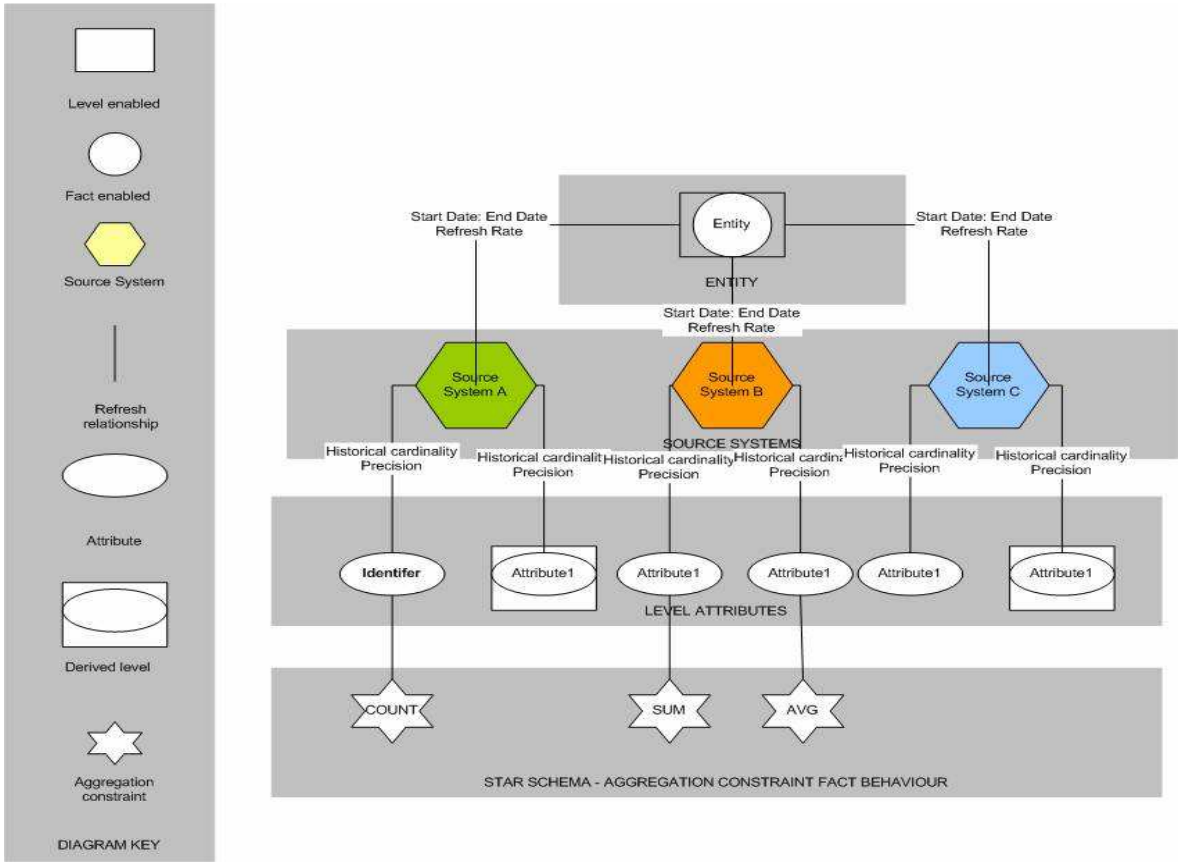


Figure 19 DWGraph - Entity Perspective



Having identified the source systems and their high-level constraints, it is necessary to consider each of the entities in the problem domain in detail. The four regions of the template in Figure 19 show the relationship between the entity, its attributes, and the source systems that will supply the data. An entity can be a candidate for a fact, dimension level, or both depending upon the properties of its attributes. A new level can be derived from an attribute if the attribute assumes a discrete set of values. An attribute value for an entity may change over time. The temporal relationships between the entity, source system and attribute can model this behaviour. An attribute can be a candidate for a fact measure and constraints on these attributes can be modelled in the lower shaded section.

When all the entities identified in the domain have been modelled using the template in Figure 19 (above) it will be possible to construct a high level ER diagram. Figure 20 (below) represents the enterprise data model and includes the entities modelled in the previous steps. This perspective allows a global view of the enterprise data that is used to identify the relationship between facts, dimension levels and potential analysis hierarchies.

Figure 21 (below) shows the template for the hierarchy perspective. The first step is to define any generic hierarchies that recur in the analysis of the enterprise. By encapsulating common hierarchies like Day-Month-Year, Department–Office–Region, or City–State–Country, the model reduces the complexity of the analysis perspectives of the DW (see Figures 22 and 23). Having identified generic hierarchies these can be used to construct user defined hierarchies. Hierarchies can include levels, generic hierarchies and nested user defined hierarchies.

Figure 20 DWGraph - Enterprise Data Model Perspective

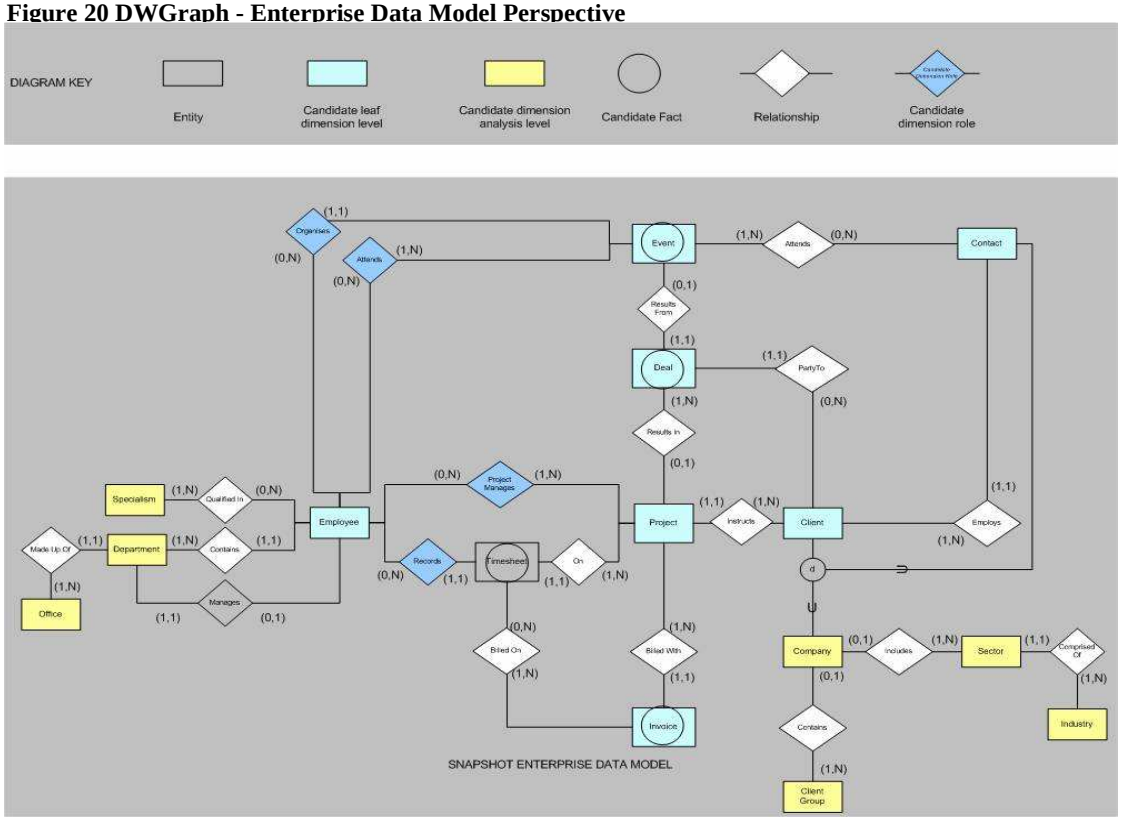


Figure 21 DWGraph - Hierarchy Perspective

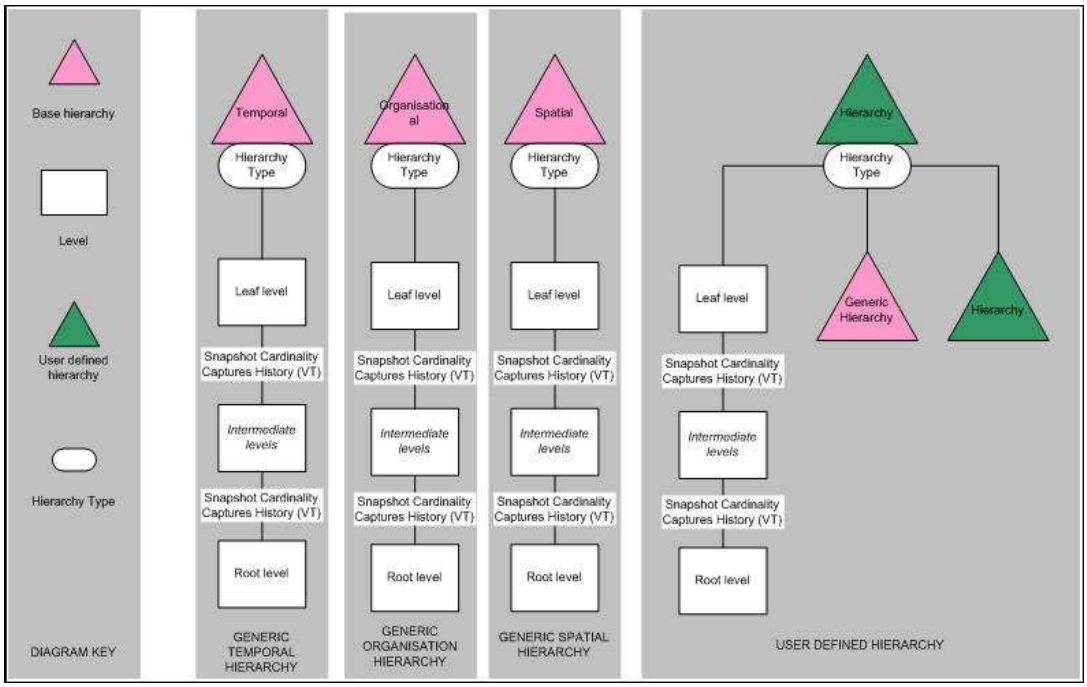


Figure 22 DWGraph - Fact Perspective

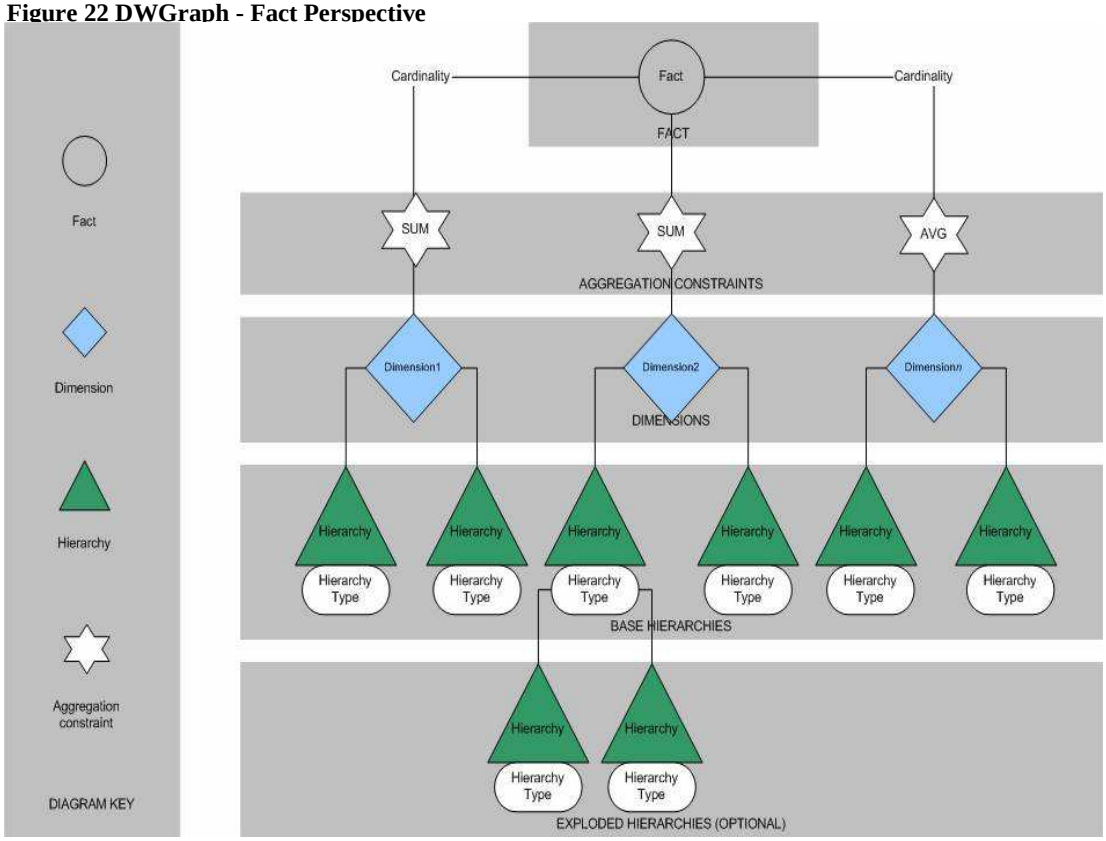
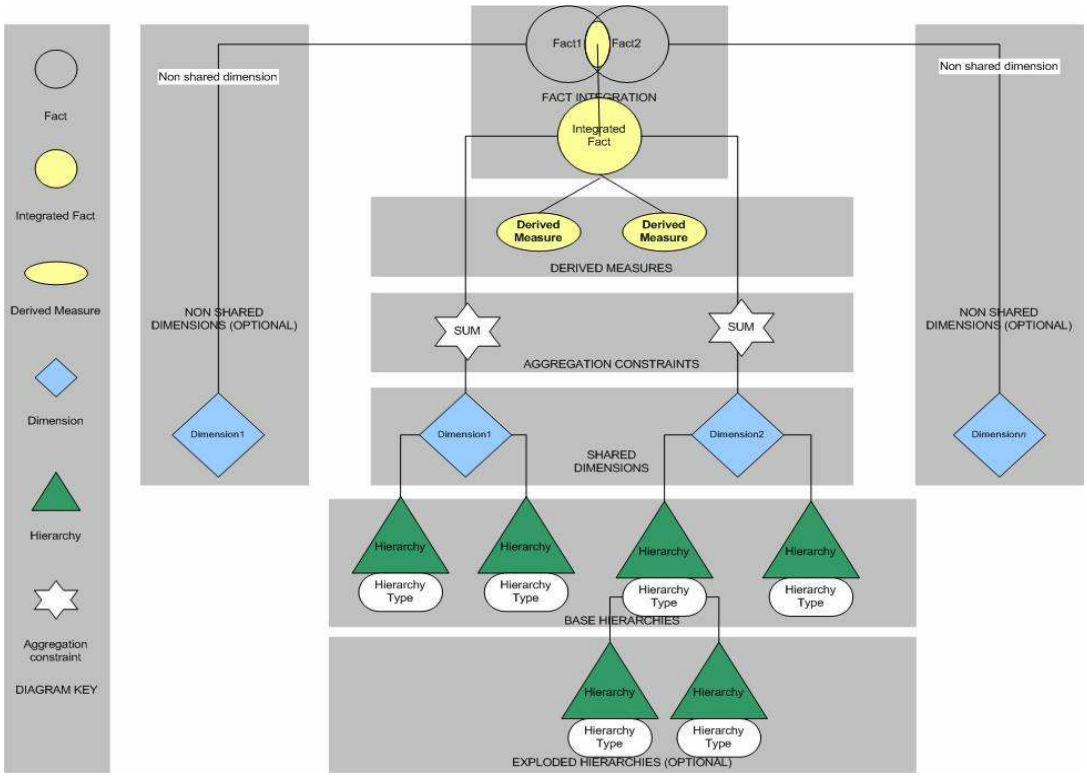


Figure 23 DWGraph - Fact Integration Perspective



Figures 22 and 23 (above) are the templates for presenting user centric analysis views of the organisation. These templates combine the fact entities identified in the entity perspective (Figure 19), with the analysis hierarchies identified in enterprise perspective (Figure 20) and modelled in the hierarchy perspective (Figure 21). This allows the DW to be modelled from multiple analysis perspectives. The template in Figure 22 deals with modelling the valid analysis on single fact. The template in Figure 23 demonstrates the notation for combining facts that share common dimensions. In addition this template allows for the explicit modelling of new measures derived from the integrated facts.

8.3 Future work

DWGraph technique is currently untested in a DW project. A detailed real-world case study would be a valuable method for evaluating the effectiveness of DWGraph. It would also be useful to perform comparative tests on DWGraph against the other DW models to see whether DWGraph did have any benefits for users' understanding.

ER model semantics underlies the DWGraph. However, DWGraph as a stand-alone modelling technique lacks a formal algebra. The graphical notation should be compatible with an existing algebra but more work is required in this area.

Ultimately, no DW modelling technique can claim effectiveness until it is used on real projects, by real developers, and accepted by real business users charged with making management decisions.

APPENDICES

1 Domain scenario

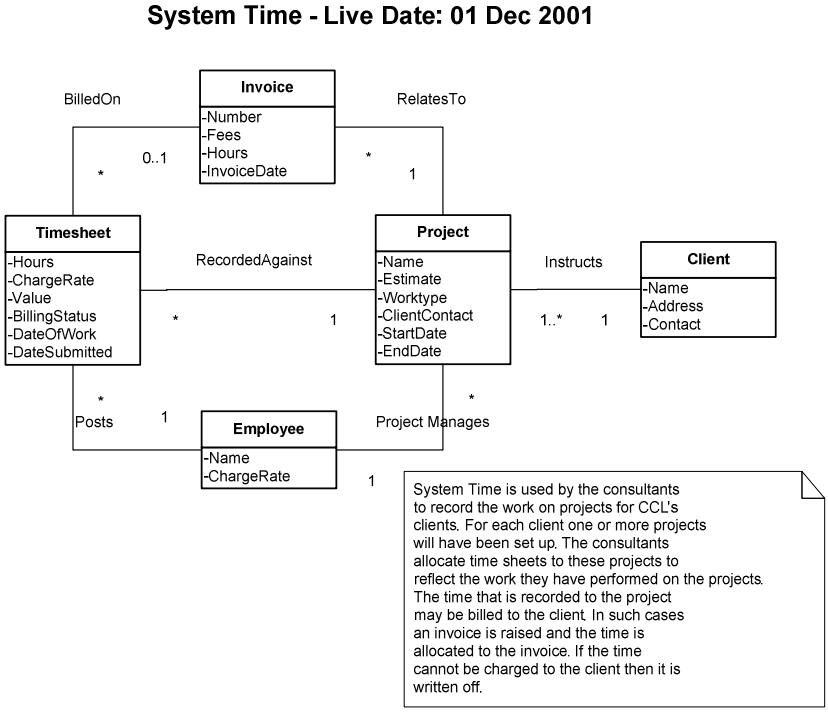
Computer Consultants Ltd (CCL) is a medium sized software consultancy firm. In recent times they have grown rapidly both internally and by acquisition. The firm now finds itself with an array of software systems each designed to meet various requirements of the firm but collectively not well integrated.

CCL decided to build a DW. This allowed them to integrate data relevant to reporting and analysis without having to replace or modify all their existing systems. The data in question resides in four different systems: System Time that records time and billing information on projects; System HR that records details of firm employees; System CRM that records marketing activity and client details; and System Stock Exchange that gives further details of clients.

2 Domain source system data models

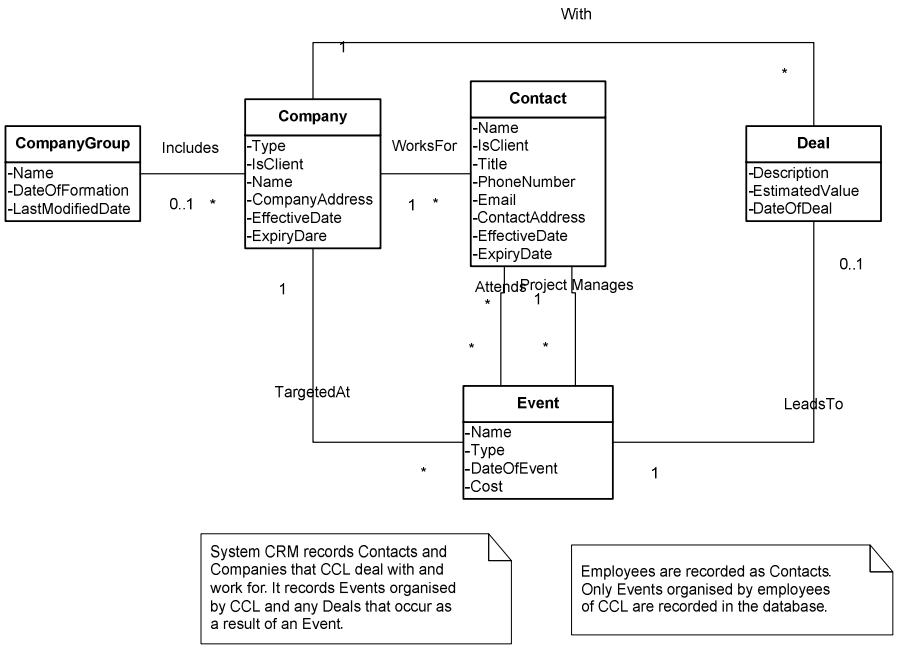
2.1 System Time data model

Figure 24 CCL Scenario - System Time



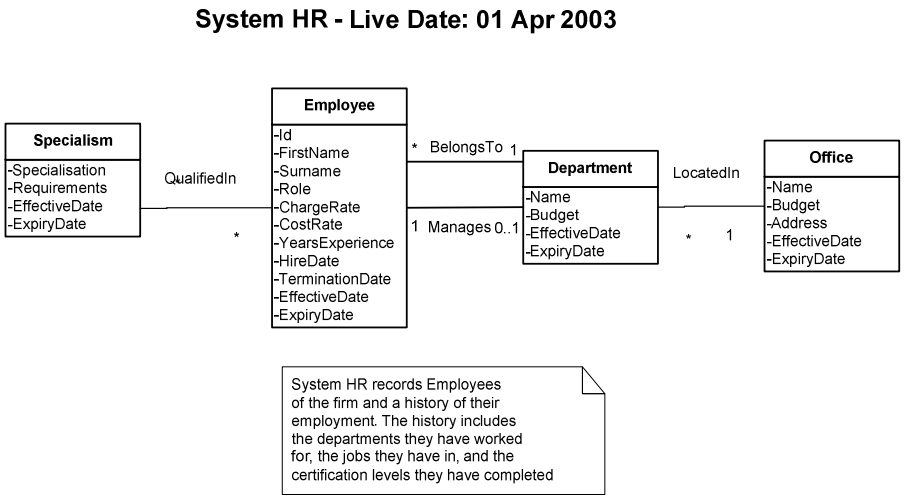
2.2 System CRM data model

Figure 25 CCL Scenario - System CRM



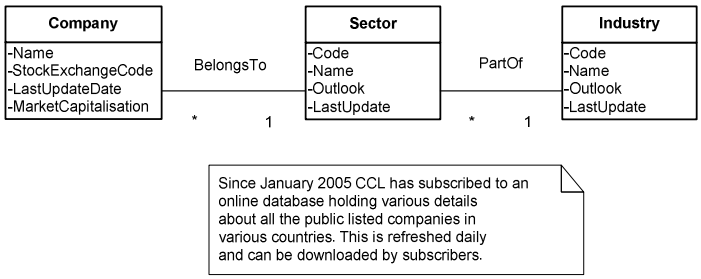
2.3 System HR data model

Figure 26 CCL Scenario - System HR



2.4 System Stock Exchange data model

Figure 27 CCL scenario - System Stock Exchange



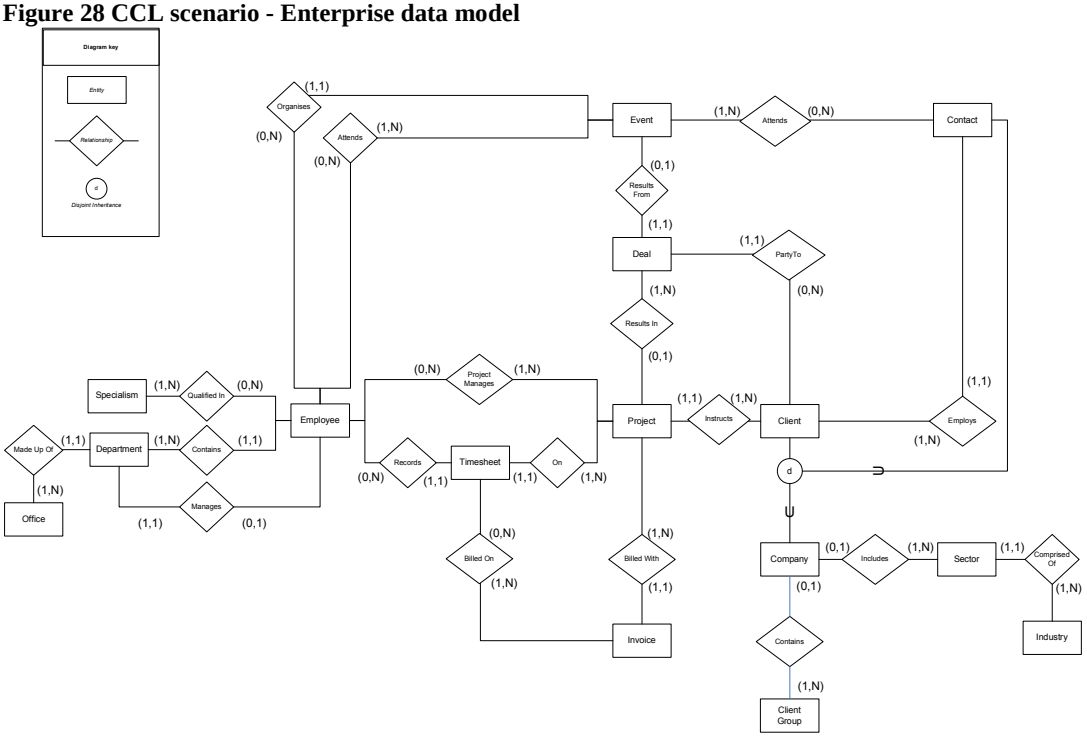
3 Domain scenario data warehouse requirements

The DW is required to track performance of clients, employees, and the firm's marketing strategy. By integrating data from their various systems they hope to gain a better insight into where there revenue is coming from, which employees are generating new business and working on successful projects.

The events they are particularly interested in tracking are billings, staff turnover, and marketing events. They wish to be able to analyse these events from a number of different perspectives including client, employee, project, and over time.

4 Data warehouse scenario enterprise data model

Figure 28 CCL scenario - Enterprise data model



REFERENCES

Abell´o, A., Samos, J., Saltor, F. (2002) YAM^2 (Yet Another Multidimensional Model): An extension of UML, IEEE Proceedings of the International Database Engineering and Applications Symposium, Pages: 172 - 181

Abell´o, A., Samos, J., Saltor, F. (2006) YAM^2: a multidimensional conceptual model extending UML. ELSEVIER, Information Systems 31 (2006) 541 - 567

Akkok, M.N. (2004) Defining Visual Immediacy, the Underused Gift of Diagrammatic Modeling Languages, [online] http://heim.ifi.uio.no/~nacia/DISSERTATION-00.pdf Appendix D Paper #3 [accessed March 2006]

Artz, J.M. (1997) How good is that data in the warehouse? ACM SIGMIS Database archive Volume 28, Issue 3 (Summer 1997)

Artz, J.M. (2006) Data Driven vs. Metric Driven Data Warehouse Design, Idea Group Encyclopedia of Data Warehousing and Mining Vol. 1

Atkins, C. and Patrick, J. (1998) NaLER: A Natural Language Method for Interpreting E-R Models, ACM Press Proceedings of the 1998 International Conference on Software Engineering: Education & Practice

Badia, A. (2004) Entity-Relationship Modeling Revisited, ACM Press ACM SIGMOD Record archive Volume 33, Issue 1

Blair, A., Debenham, J. and Edwards, J. (1995) A Comparative Study of Formal Methodologies for Designing IDSSs, University of Technology, Sydney

Bliujute, R., Saltenis, S., Slivinskas, G., Jensen, C.S. (1998) Systematic Change Management in Dimensional Data Warehousing, [online] www.cs.auc.dk/research/DP/tdb/ TimeCenter/TimeCenterPublications/TR-23.pdf [accessed March 2006]

Bruckner, R.M., Beate, L., Schiefer, J. and Tjoa, A.M. (2001) Modeling temporal consistency in data warehouses, IEEE Database and Expert Systems Applications, 12th International Workshop on 3-7 Sept. 2001 Page(s):901 - 905

Burton-Jones, A. and Meso, P. (2002) How Good are these UML Diagrams? An Empirical Test of the Wand and Weber Good Decomposition Model, In Proceedings of the 23rd International Conference on Information Systems

Burton-Jones, A. and Weber, R. (1999) Understanding relationships with attributes in entity-relationship diagrams, ACM International Conference on Information Systems. Proceedings of the 20th international conference on Information Systems

Calero, C., Piattini, M., Pascual, C., Serrano, M.A.: (2001) Towards data warehouse quality metrics, 3rd International Workshop on Design and Management of Data Warehouses (DMDW 2001), Interlaken, Switzerland

Carlson, R., Chandler, P., Sweller, J. (2003) Learning and Understanding Science Instructional Material, American Psychology Association Journal of Educational Psychology, Vol. 95(3), pp. 629-640.

Chan, H., Siau, K. and Wei, K. (1998) The Effect of Data Model, System and Task Characteristics on User Query Performance - An Empirical Study, ACM Press The DATA BASE for Advance in Information Systems - Vol. 29, No. 1

Chen, P. P. (1976) The Entity-Relationship Model - Toward a Unified View of Data, ACM Press ACM Transactions on Database Systems Vol. 1 No. 1 Pages 9-36

Chen, P. Thalheim, B., Wong, L.Y. (1997) Future Directions of Conceptual Modeling. 287-301, Web resource Conceptual Modeling, Current Issues and Future Directions, Selected Papers from the Symposium on Conceptual Modeling, Los Angeles, California, USA, held before ER'97. Lecture Notes in Computer Science 1565 Springer 1999

Cheng, P., Lowe, R. and Scaife, M. (2001) Cognitive science approaches to understanding diagrammatic representations, Springer Science Artificial Intelligence Review, 15(1-2), 79-94

Chenoweth, T., Corral, K., Demirkan, H. (2006) Seven key interventions for data warehouse success, ACM Press January 2006. Communications of the ACM, Volume 49, Issue 1

Date, C.J. (2004) An Introduction to Database Systems, 8th Edition, Addison-Wesley

Degani, A. (2004) Taming Hal: Designing Interfaces Beyond 2001 , Palgrave Macmillan ISBN: 031229574X

Dori, D., Feldman, R. and Sturm, A. (2005) Transforming an Operational System Model to a Data Warehouse Model: A Survey of Techniques, IEEE. IEEE International Conference on Software - Science, Technology & Engineering (SwSTE'05)

Elmasri, R. and Navathe, S.B. (2004) Fundamentals of Database Systems 4th Edition Reading, MA: Addison-Wesley

Franconi, E. and Kamble, A. (2004a) A Data Warehouse Conceptual Model, IEEE Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04)

Franconi, E. and Kamble, A. (2004b) The GMD Data Model and Algebra for Multidimensional Information, Springer-Verlag CAiSE'04, Pages 446-462.

Franconi, E. and Sattler, U. 1999. A data warehouse conceptual data model for multidimensional aggregation: a preliminary report. Journal of the Italian Association for Artificial Intelligence AI*IA Notizie 9--21. [online] http://citeseer.ist.psu.edu/franconi99data.html [Accessed March 2006]

Galindo, J., Urrutia, A., Carrasco, A. and Piattini, M. (2004) Relaxing Constraints in Enhanced Entity-Relationship Models Using Fuzzy Quantifiers, IEEE Transactions on Fuzzy Systems Vol. 12 No.6

Gemino, A. and Wand, Y. (2003) Evaluating modeling techniques based on models of learning, ACM Press Communications of the ACM Volume 46, Issue 10

Gemino, A. and Wand, Y. (2005) Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties, ELSEVIER Data and Knowledge Engineering 55 p301-326

Golfarelli, M.; Maio, D.; Rizzi, S.; (1998) Conceptual Design of Data Warehouses from E/R Schemes, IEEE System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on Volume 7, 6-9 Jan. 1998 Page(s):334 - 343 vol.7

Golfarelli, M., Rizzi, S. (1999) Designing the Data Warehouse: Key Steps and Crucial Issues, Maximilian Press Publisher Journal of Computer Science and Information Management, Vol. 2, N. 3, 1999

Gregersen, H. and Jensen, C.S. (1999) Temporal Entity-Relationship Models-A Survey , IEEE Educational Activities Department IEEE Transactions on Knowledge and Data Engineering, Volume 11 Issue 3

Gutwenger, C., Jünger, M., Klein, K., Kupke, J., Leipert, S. and Mutzel P (2003) A New Approach for Visualizing UML Class Diagrams, ACM Press Proceedings of the 2003 ACM symposium on Software visualization

Hahn, J. and Kim, J. (1999) Why are some diagrams easier to work with? Effects of diagrammatic representation on the cognitive intergration process of systems analysis and design , ACM Press ACM Transactions on Computer-Human Interaction (TOCHI), Volume 6 Issue 3

Hess, T.J. and Wells, J.D. (2002) Understanding How Metadata and Explanations Can Better Support Data Warehousing and Related Decision Support Systems: An

Exploratory Case Study , IEEE 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 8 p. 223

Husemann, B., Lechtenborger, J., Vossen, G. (2000) Conceptual data warehouse design, In Proceedings of International Workshop on Design and Management of Data Warehouses, Stockholm, 2000

Inmon, W. H. (1996) Building the Data Warehouse, 2nd ed.; John, Wiley & Sons: New York etc. 1996

Khan, K.M., Kapurubandara, M., Chadha, U. (2004) Incorporating Business Requirements and Constraints in Database Conceptual Models , Australian Computer Society Conferences in Research and Practice in Information Technology Vol 31

Kim, Y. and March, S.T. (1995) Comparing Data Modeling Formalisms, ACM Press Communications of the ACM June 1995 Vol 38 No 6

Kimball, R. and Ross, M. (2002) The complete guide to dimensional modelling 2nd Edition, John Wiley & Sons: New York 2002

Koning, H. (2002) Guidelines Readability, version 1.0, [online] http://www.cs.uu.nl/people/koningh/content.php?page=2001 [accessed May 2005]

Koning, H., Dormann, C. and van Vliet, H. (2002) Practical guidelines for the readability of IT-architecture diagrams , ACM Press Proceedings of the 20th annual international conference on Computer documentation

Kulpa, Z. (1994) Diagrammatic representation and reasoning. Machine GRAPHICS & VISION vol. 3, nos. 1/2, 1994, pp. 77-103.

Larkin, J.H. and Simon, H.A. (1987) Why a Diagram is (Sometimes) Worth Ten Thousand Words, Cognitive Science 11 65-99

Liao, C. and Palvia, C.P. (2000) The impact of data models and task complexity on end-user performance: an experimental investigation, Academic Press Inc Int. J. Human-Computer Studies (2000) 52, 831}845

Lujan-Mora, S. (advised Trujillo, J.) (2005) Data warehouse design with UML, PHD Thesis Universitat d'Alacant Department of Software and Computing Systems

Lujan-Mora, S., Trujillo, J. (2003) A Comprehensive Method for Data Warehouse, In: Proc. of the 5th Intl. Workshop on Design and Management of Data Warehouses (DMDW'03), Berlin, Germany (2003) 1.1--1.14

Lujan-Mora, S. and Trujillo, J. (2004) Physical Modelling of Data Warehouses using UML, ACM Press Proceedings of the 7th ACM international workshop on Data warehousing and OLAP (March 2004)

Malinowski, E. & Zimányi, E. (2004), OLAP hierarchies: A conceptual perspective, in 'Proc. of the 16th Int. Conf. on Advanced Information Systems Engineering', pp. 477-491

Malinowski, E. & Zimányi, E. (2006) A Conceptual Solution for Representing Time in Data Warehouse Dimensions, The Third Asia-Pacific Conference on Conceptual Modelling (APCCM 2006)

Moody, D.L. (1997) A Multi-Level Architecture for Representing Enterprise Data Models, Springer-Verlag Lecture Notes In Computer Science; Vol. 1331 archive Proceedings of the 16th International Conference on Conceptual Modeling

Moody, D.L. (2002) Complexity Effects On End User Understanding of Data Models: An Experimental Comparison of Large Data Model Representation Methods, ECIS 2002 • June 6–8, Gdańsk, Poland [online] www.csrc.lse.ac.uk/asp/aspecis/20020016.pdf [accessed March 2006]

Osei-Bryson, K.M. and Ngwenyama, O.K. (2004) Supporting Semantic Diversity in the Relational Data Model: The Case of Multi-Face Attributes, Kluwer Academic Publishers Information Systems Frontiers archive Volume 6 , Issue 3

Parsons, J. (2003) Effects of Local Versus Global Schema Diagrams on Verification and Communication in Conceptual Data Modeling, Journal of Management Information Systems Issue: Volume 19, Number 3

Purchase, H.C., Carrington, D., Allder, J. (2002) Empirical Evaluation of Aesthetics-based Graph Layout, Kluwer Academic Publishers Empirical Software Engineering Volume 7, Issue 3 (September 2002)

Ravat, F., Teste, O. and Zurfluh G. (1999) Towards Data Warehouse Design, ACM Press Proceedings of the eighth international conference on Information and knowledge management

Sampson, J. and Atkins, C. (2002) Semantic Integrity in Data Warehousing: A framework for understanding., IEEE Proceedings of the 35th Annual Hawaii Conference on System Sciences

Sapia, C., Blaschka, M., Höfling, G., Dinter, B. (1998) Extending the E/R Model for the Multidimensional Paradigm , Springer-Verlag Proceedings of the Workshops on Data Warehousing and Data Mining: Advances in Database Technologies Lecture Notes In Computer Science; Vol. 1552

Scaife, M. and Rogers, Y. (1996) External cognition: How do graphical representations work?, International Journal of Human-Computer Studies, 45, 185-213

Sen, A. and Sinha, A. (2005) A Comparison of Data Warehousing Methodologies, ACM Press Communication of the ACM March 2005/Col.48 No. 3

Serrano, M., Calero, C., Trujillo, J. Luján-Mora, S. and Piattini, M. Empirical validation of metrics for conceptual models of data warehouses. In Proc. CAiSE, pages 506--520, 2004.

Shanks, G.G., O'Donnell, P.A. and Arnott, D.R. (2003) Data Warehousing: A Field Study, Web resource vishnu.sims.monash.edu.au:16080/ subjects/ims5026/readings/fieldstudy.pdf Accessed Feb 2006

Siau, K., Chan, H.C., Wei, K.K. (2004) Effects of Query Complexity and Learning on Novice User Query Performance With Conceptual and Logical Database Interfaces,

IEEE. IEEE Transactions on Systems, Man and Cybernetics - Part A Systems and Humans Vol. 34 No. 2

Sinha, A.P., Vessey, I. (1999) An empirical investigation of entity-based and object-oriented data modeling: a development life cycle approach, Association for Information Systems Proceeding of the 20th international conference on Information Systems

Srivastava, J. and Chen, P. (1999) Warehouse creation - A Potential Roadblock to data warehousing, IEEE, IEEE Transaction on knowledge and data engineering Vol 11 No. 1

Sweller, J., Chandler, P., Tierney, P., Cooper, M. (1990) Cognitive Load as a Factor in the Structuring of Technical Material, American Psychology Association Journal of Experimental Psychology: General 1990 Vol. 119 No. 2 176-192

Trujillo, T., Palomar, M.. (1998) An object oriented approach to multidimensional database conceptual modeling (OOMD), Proceedings of the 1st ACM international workshop on Data warehousing and OLAP, p.16-21, November 02-07, 1998, Washington, D.C., United States

Trujillo, T., Palomar, M., Gomez, J., Song, I. (2001) Designing Data Warehouses with OO Conceptual Models , IEEE December 2001 (Vol. 34, No. 12) pp. 66-75

Tryfona, N., Busborg, F., Christiansen, J.G.B. (1999) starER: a conceptual model for data warehouse design, ACM Press Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP

Winter, R. and Strauch, B. (2002) A Method for Demand-Driven Information Requirements Analysis in Data Warehousing Projects, IEEE Computer Society Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03)

Winter, R. and Strauch, B. (2004) Information Requirements Engineering for Data Warehouse Systems, ACM Press 2004 ACM Symposium on Applied Computing

INDEX