Different types of meronymic, or part-whole, relationships have long been recognized as being important, but difficult to distinguish amongst, by researchers in areas outside of the database field. The purpose of the research reported here is to analyze these relationships and show how they apply to database design problems. This includes both providing a clear set of rules for distinguishing amongst different types of meronymic relationships and determining how a design can be affected based upon the analysis.

Database management systems are very good at dealing with the information they have stored on a syntactic basis. They do not, unfortunately, have good capabilities for capturing much of the semantics of a database application. Also, there is a noted need to extend the relational model to accommodate more real world knowledge (Reiter, 1984). In an attempt to overcome such problems, a number of researchers have recently begun to explore certain semantic relationships, known as data abstractions, in order to analyze their impact on database design (for example, Mattos, 1988; Mattos and Michels, 1989; Smith and Smith, 1977; Blaha et al., 1988; Davis and Bonnell, 1989; Potter and Kerschberg, 1988; Goldstein and Storey, 1991a; 1991b; and Ullman, 1986). Data abstractions are semantic relationships, adopted from concepts in semantic networks (Brodie, 1986) with the most common being inclusion (subtype--supertype or is-a), aggregation (part of), and association (membership) relationships. An example of why such relationships are important is easily demonstrated by the well-known class inclusion, or is-a, relationship. A relationship A is-a B implies that something that is true of the entity type B is also true of the entity type A (Brachman, 1983). Any attributes of B, therefore, that also appear as (non-key) attributes of A can be deleted from the entity type A and “inherited” from the B entity type. This helps to minimize the redundancy found in a database. The focus of this research, however, is on meronymic relationships of which the aggregation abstraction is one type.

This paper is divided into five sections. The next section defines semantic relationships and database design concepts. A taxonomy of part-whole relationships based upon the work of Winston et al. (1987) is then analyzed and the analysis applied to a specific database problem. The final section summarizes and concludes the paper, offering suggestions for future research.

Semantic Relationships

Research in linguistics, logic, and cognitive psychology has recognized many semantic relationships (Winston et al., 1987). Semantic relations (hereafter referred to as semantic relationships) describe relationships of meaning among words and are often divided into five categories (Landis et al., 1987): 1) antonyms; 2) synonyms; 3) class inclusion; 4) part-whole; and 5) case relationships. It is the part-whole, or meronymic, relationships that are analyzed in this paper in terms of their impact on database design.

Kuczora and Cosby (1989) suggest that it is difficult,
but necessary, to distinguish amongst meronymic relationships in the knowledge extraction process for expert systems. Within the database area, it is important for database designers to appreciate the more subtle points of these relationships in order to use them to capture some of the semantics of an application. In addition, more and more end-users are becoming the designers of their own databases (Sheng and Higa, 1990). It would also be useful for these end-users to appreciate how meronymic relationships can be properly employed in a design.

The analysis is carried out in terms of the entity-relationship model (Chen, 1976) which is well-known as an effective conceptual modelling tool (Taucovich, 1989), and its translation into a relational model which has become widely accepted as an implementation model.

**Cardinalities**

The relationships discussed in this paper are binary ones of the form $A$ Verb Phrase $B$. The minimum/maximum cardinalities (Tsichritzis and Lochovsky, 1982) represent the minimum and maximum number of occurrences of the entity type, $B$, that can exist for each occurrence of the entity type, $A$, and vice versa.

E.g.:  
Manager manages Employees  
(1,*), (1,1)

This is interpreted as each occurrence of Employee participates in exactly one occurrence of the “manages” relationship, whereas an occurrence of Manager participates in at least one, and possibly many (*), occurrences of “manages”. Informally, every employee has exactly one manager, and a manager manages at least one employee.

**Entity-Relationship Model to Relational Model**

In general, when an entity-relationship model is converted into a relational model, each entity type becomes a separate entity relation (Teorey et al., 1986) that looks exactly the same as the entity (that is, it has the same key and non-key attributes). This entity relation may then be modified if a relationship is represented by adding another entity type’s key to it as a foreign key. Teorey et al. (1986) refer to this modified entity relation as an extended entity relation. Alternatively, a relationship may be represented by creating a separate relationship relation whose key is the concatenation of the keys of the involved entity types and whose non-key attributes are the relationship’s attributes. When the min/max cardinalities of one entity type are (1,1) in a relationship, the key of the other entity type is added, as a foreign key, to the relation of the entity type with the (1,1) cardinalities. All other relationships are many-to-many and are represented as separate relations. Furthermore, it is the many-to-many relationships that can have relationship attributes. (For further discussion on database design, see Storey (1988, 1991a); Storey and Goldstein (1988); Teorey et al. (1986).)

E.g.:  
Relationships:  
Managers manage Employees  
(1,*), (1,1)

Represent by making the key of Manager a foreign key in the Employee relation.

Employees work-on Projects: [length-of-time]  
(0,*), (1,*)

Represent by creating a separate relation whose key is the concatenation of the keys of Employee and Project and whose non-key attribute is the relationship attribute.

**Relational Model (capital letters denote key attributes):**

Manager: [EMP#, ...]  
Employee: [EMP#, ..., manager-emp#]  
Project: [PROJECT-ID, ...]  
Employees_work_on_projects:  
[EMP#, PROJECT-ID, length-of-time]

It is important to appreciate that it is essential to obtain the correct min/max cardinalities for a relationship because they dictate how the relationship will be represented in the final, relational model. Understanding the semantics of a (e.g. meronymic) relationship aids in ensuring that the min/max cardinalities are correct, and, thus, provides some of the motivation for this research.

**Taxonomy of Relationships**

The relationship between wholes and parts has been one of the main issues in philosophy since ancient times (Hai and Klir, 1985). The meronymic, or part-whole relationship is important because it is a transitive, hierarchical, inclusion relationship (Cruse, 1979). (See Chaffin et al. (1988).) Meronymic relationships are expressed by ‘A is part of B’, ‘A is partly B’, and related expressions. There are a large number of such expressions; for example, there are at least 40 fairly general words for different kinds of parts, such as component, member, portion and feature, and a larger number of more specialized terms (Chaffin and Herrmann, 1988). According to Chaffin et al. (1988), the term “part-of” is the most general of the terms that can be used to express different kinds of part-whole relationships and that Roget’s Thesaurus (1962) lists approximately 400 synonyms for part-of. Most part-terms can be used to refer to part-whole relationships, but others cannot.
Related Content

**OO and EER Conceptual Schemas: A Comparison of User Comprehension**
[www.igi-global.com/article/eer-conceptual-schemas/51140?camid=4v1a](www.igi-global.com/article/eer-conceptual-schemas/51140?camid=4v1a)

**Incorporating Spatial Queries into Semantic Sensor Streams on the Internet of Things**
[www.igi-global.com/article/incorporating-spatial-queries-into-semantic-sensor-streams-on-the-internet-of-things/194998?camid=4v1a](www.igi-global.com/article/incorporating-spatial-queries-into-semantic-sensor-streams-on-the-internet-of-things/194998?camid=4v1a)

**Higher-Order Types and Information Modeling**
[www.igi-global.com/chapter/higher-order-types-information-modeling/4376?camid=4v1a](www.igi-global.com/chapter/higher-order-types-information-modeling/4376?camid=4v1a)

**Parallel Skyline Computation Exploiting the Lattice Structure**
[www.igi-global.com/article/parallel-skyline-computation-exploiting-the-lattice-structure/153516?camid=4v1a](www.igi-global.com/article/parallel-skyline-computation-exploiting-the-lattice-structure/153516?camid=4v1a)