Applied Ontology 1 (2005) 117–139
IOS Press

A multi-layered ontology for comparing relationship semantics in conceptual models of databases[1]

Sandeep Purao[a] and Veda C. Storey[b]
[a]*School of Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16802, USA*
Tel.: +1 814 863 0017; E-mail: sandeep-purao@psu.edu
[b]*Department of Computers and Information Systems, Georgia State University, Box 4015 Atlanta, GA 30302, USA*
E-mail: vstorey@gsu.edu

Abstract. Relationships are an integral part of the design of a database. Comparing and integrating relationships from heterogeneous databases requires that the relationships be mapped to each other or to a common classification. Identifying similarities and resolving differences in relationships across large data sources is a resource-intensive task that could benefit greatly from semi-automated approaches. A prerequisite to developing such approaches is a clear understanding of the semantics of relationships used in database design. This research presents a layered ontology for classifying the semantics of relationships. It consists of a core layer that captures the fundamental types of relationships between entities. A middle layer provides the internal context, obtained from entities surrounding the relationship, to interpret the fundamental types. The outer layer allows further interpretation using the external context, that is, the domain in which a relationship is being used. An initial assessment on relationships from a variety of application domains demonstrates that the ontology can be adequate and useful for comparing relationships across databases.

Keywords: Database design, relationships, verb phrase, ontology, semiotics, entity-relationship model, natural language, conceptual modeling, classification scheme

1. Introduction

Understanding the semantics of entities and relationships in a database is an important prerequisite for comparing and integrating data across heterogeneous data sources. This is an increasingly important problem in today's networked world that relies on inter-organizational coordination.[2] The amount of data available in both traditional and non-traditional application domains [Gennari *et al.,* 2000] continues to

[1]This research was supported by the J. Mack College of Business Administration, Georgia State University. The authors wish to thank the editors and reviewers for their helpful comments on an earlier version of this manuscript. The authors also thank Harald Ulrich, Cecil Chua and Yang Chen for their assistance in preparing the input for the empirical analysis. An earlier version of this work was presented at ER'04.
[2]See, for example, the SemDis project aimed at discovering different kinds of associations in the semantic web. http://lsdis.cs.uga.edu/Projects/SemDis/ (accessed, 10 March 2005).

increase, as does the need for new methods to integrate databases. Fully automated integration techniques, however, remain elusive because the integration effort must resolve differences in semantics and context. Integration techniques, therefore, are designed to aid integrators in a manner that requires minimal work on their part [Biskup and Embley, 2003]. A prerequisite to designing such integration techniques is the availability of generic, domain-independent entity and relationship ontologies. Although some research has been carried out to classify entities, very little attention has been devoted to furthering our understanding of the semantics of relationships.

Much research on understanding database relationships has focused on distinguishing entities from attributes and relationships, and defining the innate nature of what a 'relationship' is [Dey *et al.,* 1999; Wand *et al.,* 1999; Weber 1996; Shanks *et al.,* 2002; Weber 2002] In contrast, understanding the *semantics* of relationships requires an explicit characterization of the meaning underlying the descriptive verb phrase that is part of each relationship. Few efforts have been directed towards understanding the semantics of relationships with a view to constructing classifications of the semantics of relationships. Notable exceptions include Storey [2005; 1993] and Siau [2004] which are mostly based on lexicons or the relation element theory proposed by Chaffin and Herrmann [1988]. These efforts are important starting points, but do not all provide comprehensive classifications of relationships that may be useful for comparison of relationships across data sources.

The comparison of relationships is a difficult task. Consider, for example, the following relationships in two separate databases:

Customer (entity) <buys> Product (entity)
Customer (entity) <purchases> Product (entity)

These relationships reflect the same elements in the universe of discourse, and use synonymous verb phrases to describe them. The two relationships may, therefore, be considered *equivalent*. Next, consider:

Customer (entity) <reserves> Car (entity)
Customer (entity) <rents> Car (entity)

It is possible to think of the combination of these as a complex event, which starts with reserving and continues with renting. However, database relationships must take into account individual ingredients of this complex event. These two relationships, therefore, reflect different elements in the universe of discourse. The first captures the declaration of the intent to carry out a transaction that involves acquiring control (though not ownership) of an asset. The second captures the fact that the customer carries out the transaction. Clearly, these may be viewed as consequences of different events that may be temporally connected ('reserve' occurs before 'rent'). Therefore, the relationships cannot be considered equivalent in spite of having the same entities and a similar structure. Finally, consider:

Manager (entity) <negotiates> Agreement (entity)
Manager (entity) <evaluates> Agreement (entity)

Both relationships represent an interaction in the universe of discourse. However, the first relationship, "negotiates," implies changing the entity, whereas the second relationship, "evaluates," simply involves viewing its contents towards a purpose. These two relationships, therefore, are not equivalent in spite of having a similar structure and participation from similar entities. Unlike the previous example, however, these two relationships do not represent events necessarily separated in time. Instead, they represent different alternative interactions that the entity Manager may have with the entity Agreement. These examples illustrate the importance of understanding the meaning of verb phrases for the purpose of classification and integration.

The objective of this research, therefore, is: to *develop a comprehensive ontology for classifying the semantics of relationships; and to provide a preliminary assessment of its adequacy and usefulness for comparing relationships.* We acknowledge and build on the ideas from linguistics, which suggest that events, both static and dynamic, are preconditions for relationships. Our work has the pragmatic goal of facilitating understanding for comparing relationships in databases that occur in practice, and the research goal of providing a systematic classification of events that lead to these relationships. We incorporate prior work that suggests some relationships are "formal" in the sense that they represent structural relationships among things (e.g. part-of), whereas others involve interaction among things. Following this perspective, classifying the universe of relationship verb phrases, therefore, amounts to understanding and classifying events. Our research, however, does not adopt this strategy because of our interest in classifying relationships as they occur in practice, where the dictates of appropriate conceptual modeling may not have been followed fully. Instead, we allow 'not-so-perfect' relationships into the universe of discourse, which gives us the challenge of devising an ontology that can classify relationships occurring in databases for the purpose of comparison and integration. The contribution of the research is to extend prior classifications of data abstractions and verb phrases to provide a useful way to understand and compare relationships as they occur in practice in real world databases.

The paper is divided in five sections. Section 2 reviews related research. Section 3 develops the ontology for classifying relationships. In Section 4, we describe the architecture of a prototype and the results of an initial assessment of the effectiveness of the ontology. Section 5 concludes the paper.

2. Prior work

2.1. Relationships in database design

The design of a database involves representing the universe of discourse in a structure in such a way that it accurately reflects reality [Navathe and Elmsari, 2003]. Conceptual modeling of databases is concerned with things (entities) and associations among things (relationships) [Chen, 1993; Wand *et al.*, 1999; Brodie, 1981]. A relationship models an association between two or more entities [Dahchour 2001, 2003]. The number of entities involved in a relationship is the degree of the relationship. Most database design practices use simple, binary associations that capture these relationships between entities. Although higher-degree relationships are clearly important [Dey *et al*., 1999], much of the real world can, in fact, be represented using binary relationships [Siau *et al*., 1997].

A relationship, R, can be expressed as A <verb phrase> B (A <vp> B), where A and B are entities, and the verb phrase represents the descriptor associated with the relationship [Chen, 1994]. A relationship, thus, derives its meaning from being embedded in the context provided by the entities. This meaning can be inferred, at least partially, by examining the structure and semantics of the relationship. The structure can be observed by considering the manner in which the entities are arranged. This arrangement can include the degree of a relationship as well as its cardinalities. The semantics can be derived from the verb phrases. The verb phrases, in turn, can be interpreted by considering the entity names and the domain in which the relationship is being constructed. Much prior work on understanding relationships in database design has focused on articulating what the relationships 'ought to be' following first principles from Chen's seminal paper [Chen, 1976] or prescriptions such as Bunge's ontology [Bunge 1977, 1979].

One important dimension that distinguishes our research is our desire to create an ontology that can accommodate relationships that occur in real-world databases, i.e. 'not-so-perfect' relationships as opposed to what the relationships 'ought to be' [Wand and Weber 1995; Rosemann and Green 2002]. The research reported in this paper is intended to be an important extension of research in this stream to accommodate relationships in existing databases. These relationships may represent entities and their associations reasonably, but not accurately or parsimoniously. Consider, for example, the following relationships that might represent fragments of a database:

Developer (entity) <conducts> Test (entity)
Test (entity) <performed on> Software (entity)

One could argue that these are the consequences of a faulty conceptual design based upon Bunge's ontology [Rosemann and Green 2002; Rosemann *et al.*, 2004], where an entity must represent 'things,' and relationships between entities must represents 'events' that lead to changes of states for these 'things'. The decision to treat 'Test' as an entity, therefore, would not be correct in the examples above, and may be more accurately represented as:

Developer (entity) <tests> Software (entity).

The universe of relationships we are interested in classifying includes not only this ideal form of the relationship 'Developer tests Software,' but also the imperfect relationships that might reasonably be expected to occur in database designs, i.e. 'Developer conducts Test,' and 'Test performed on Software.' For this universe of "Relationships," we propose to capture the meaning of a relationship that may depend upon: 1) the arrangements of entities and cardinalities[3] of the relationship, and 2) the verb phrase that is part of the relationship. These may be loosely described as *structure* (see Section 2.2) and *semantics* (see Section 2.3). Classifications based on these can be further tempered by the semantic interpretation of the entities that surround the verb phrase, and the domain in which the relationship is constructed (analogous to [Fellbaum, 1998]).

2.2. Data abstractions in conceptual modeling

Conceptual models of databases capture the semantics, as well as the structure, of data [Hammer and Mcleod 1981]. They allow explicit representation of special semantic relationships, or *data abstractions*, that the database modeling community has identified. An abstraction is a simplified description or specification of a system that emphasizes some properties while suppressing others [Theodoulisi *et al.*, 1992; Shaw 1984]. Data abstractions focus on the structure of relationships as a surrogate for understanding what they represent.

The general categories of abstraction mechanisms are: (1) *inclusion* which represents the subtype-supertype relationship [Hull and King 1987]; (2) aggregation which allows a relationship among two or more objects to be thought of as a higher-level object [Smith and Smith 1977]; and (3) *association* where a collection of members is considered as a higher-level set [Brodie 1981; Rundensteiner *et al.*, 1994].

The <is-a> relationship associates two entities where one is general, and the other, specific [Brachman 1983; Goldstein and Storey, 1992, 1999]. The <is-part-of> relationship associates entities of which one

[3]We use the term cardinality to indicate the general nature of connectivity e.g. one-to-many as opposed to the stricter specification of min-max participation from entities.

is the whole, and the others are parts. It represents aggregation, which captures emergent meaning [Smith and Smith 1977; Shank *et al*., 2002]. The <is-member-of> relationship captures the notion of membership [Brodie 1981; Motsechning-Pitrik and Storey 1995]. Other work suggests that the <is-instance-of> relationship captures a similar construct without the notion of a limited number of instances as defining a set [Motsching-Pitrik and Mylopoulous 1992]. The <is-version-of> relationship captures multiple renderings of the same concept represented in multiple entities [Motsching-Pitrik, 2000]. The materialization abstraction [Goldstein and Storey, 1994], describes how an abstract concept may be materialized into concrete instances.

These data abstractions provide a good starting point for classifying relationships. Because they are part of the conceptual design of a database, data abstractions model type-type relationships, even when the underlying relationship instances capture a token-token notion. For example, the part-of relationship captures inclusion between object instances although it is modeled and represented in the conceptual model of the database as a whole-part relationship among types. These distinctions between type-type and type-token or token-token relationships have been addressed in information modeling for object-oriented systems [Davis ans Bonnel 1991; Mathiassen 2000]. Similar observations can be made for the member-of data abstraction.

2.3. Semantics of verb phrases in relationships

A relationship in the design of a database is typically accompanied by a verb phrase that describes the semantics of the relationship. *Semantics*, for the purposes of this research, is defined as the meaning of a term or a mapping from a contruct to the real world. Understanding a relationship, therefore, requires that one understands the semantics of the accompanying verb phrase. Early work on understanding the semantics of verb phrases can be traced to case theories and case grammars [Fillmore, 1977]. This effort from linguistics suggested that verbs may be understood as frames with nouns filling slots created by these frames. Chaffin and Herrmann's [1987, 1988] relation element theory is another example of early work in this regard, which suggested that a semantic relation can be understood as a complex structure with multiple properties, that can be used to compare relationships. This theory has been applied and adapted by database researchers to understand the semantics of verb phrases in database design [Storey, 1993; Siau, 1997, 2004].

More recent work on classifications of verb phrases include Dahchour's [2001] proposal to understand relationships using meta-classes. A final source for understanding the semantics of relationships is the compilation of types of relationships in texts related to conceptual modeling, particularly for object-oriented systems (e.g. Larman [1997]). These works typically do not distinguish between the structural classifications described in the previous sub-section, and the semantic categorization this sub-section discusses. These efforts provide important starting points that can provide further support for the ontology we develop. This research builds on much of this prior research to develop a comprehensive ontology for classifying relationships.

2.4. Characterizing ontologies of relationships

Although there are many different definitions of ontology [Weber 2002; Guarino and Welty 2004], they can generally be thought of as the consequences of capturing, representing, and using surrogates for the meanings of terms. Following Gruber [1993], we define an ontology as the explicit specification of a conceptualization, adopting its refinement by Fensel [2003] as 'a formal, explicit specification of a shared conceptualization.' Our notion of an ontology is similar to that of Dahlgren [1988], who developed an ontology as a classification mechanism for speech understanding and implemented it in an interactive tool. A key component of our approach is the idea of distinctions to create categories, similar to Noy and Hafner [1997], who devised multiple parallel dimensions along with one or more top-level categories or sub-categories. Moench et al. [2001] suggest a continuum to understand ontology building efforts with the key dimension being level of formality. The least formal are pure taxonomies, with the opposite being formal ontologies from research in artificial intelligence. Two mid-points along this are thesauri, and topic maps. These points provide cumulative requirements for building ontologies. Our research falls between the two mid-points of thesaurus and topic map, and includes hierarchy, predefined connections among terms, a taxonomy, and specifications of context and scope.

In the database area, Embley et al. [1999] develop an ontology for understanding web obituaries. Kedad and Metais [1999] use an ontology to manage semantic heterogeneity using a linguistic dictionary. Dullea and Song [1999] propose a taxonomy of recursive relationships. Bergholtz and Johanneson [2001] propose a relationship ontology that incorporates data abstractions with speech acts [Searle, 1979]. These efforts demonstrate the importance of data abstractions in developing an ontology for relationships. They also indicate that efforts to build an ontology should not ignore the intricacies of natural language that is inherent in understanding and classifying relationship verb phrases [Dahlgren, 1995; Kedad and Metais 1999]. Furthermore, these efforts demonstrate that a useful ontology should include multiple levels to capture the complexities inherent in the definition of a relationship [Unrich *et al.*, 2000; Dullea and Song 1999]. Finally, they emphasize the importance of context in classifying relationships, which is realized by the entities surrounding the verb phrase [Fellbaum, 1998].

The research also shows that the development of a useful ontology for understanding relationships continues to be a challenge. Although the need to compare relationships has been recognized by research in heterogeneous databases and data integration, no well-accepted approaches have been proposed for doing so [Biskup and Embley, 2003]. Ontology development in this area is also challenging because it combines problems related to standards development, linguistic analysis and structural analysis.

3. An ontology for classifying relationships

The development of the ontology begins at the core, which consists of fundamental classifications. To develop these fundamental classification, our starting point is Bunge's ontology, as it has been applied previously to information systems [Wand and Weber, 1995; Weber, 1997; Weber and Zhang, 1996].

We use a set of constructs from Bunge's ontology [1977; 1979] that are appropriate for understanding relationships, and refine them to construct a semantic classification. We assume that any domain (i.e. a universe of discourse) can be described by 'concrete things' and 'linkages' between them, which are referred to as 'relationships' in the entity-relationship view of conceptual design or as 'associations' or 'links' in the object-oriented view of conceptual design. Table 1 summarizes the concepts from Bunge's ontology that are useful for understanding the relationship construct.

Our adaptation of these constructs is affected by our desire to devise classifications for 'not-so-perfect' relationships that occur in practice. For example, in Bunge's terms, only concrete things can have history, and therefore, interactions can only occur between them. This may not be followed by the developers, who have designed the databases we may be interested in comparing. The ontology developed in this research restricts the use of Bunge's constructs in order to deal with 'not-so-perfect' relationships.

We start with the observation that many relationships do, indeed, represent 'interactions' among 'things.' 'Properties' of 'things' can be represented as entity attributes.[4] 'Composition' among entities

[4] A different characterization is suggested by [Weber, 1996], where relationships represent mutual properties.

Table 1
Ontological constructs from Bunge (adapted from Bera and Wand 2004)

Construct	Description
Thing	The elementary unit in the ontological model. The world is made of things.
Properties	Things have properties, which may be intrinsic or mutual.
Composition	A composite is a thing that is made up of other things.
State	The attribute value vector of a thing at a given time is the state of the thing.
Event	An event is a change of state of a thing.
Interaction	Interaction is the ability of a thing to change the state of another thing.

can be represented using data abstractions (for example, is-a for type-type; instance-of for token-type; part-of for type-type) [Goldstein and Storey 1999; Davis and Bonnell, 1992]. A relationship, then, could consist of an 'interaction' among 'things' that lead to a change in 'state' of 'things,' which could be caused by 'events.' This restricted definition of 'relationship' however, only covers the ideal case. This research is intended to accommodate 'imperfect' relationships. Thus, it is possible to think of relationships as including 'interactions' between entities, where one entity represents a 'thing', and another an 'action'. The ontological constructs that allow this definition of the ideal relationship are shown in Table 1. They are extended to describe fundamental types of relationships that lead to a classification of relationships that captures the intrinsic differences among relationships. This classification is at the core of our multi-layered ontology.

Two additional factors influence our classification of relationships. The first is the local context provided by the entities surrounding the relationship. This is analogous to the need to understanding nouns surrounding verb phrase in order to effectively classify verb phrases [Fellbaum, 1998]. The second is the external context, which is provided by the domain in which a relationship is being used. For example, the appropriate interpretation of 'screen' (revealing or hiding) depends upon the domain in which it occurs. Figure 1 shows the multi-layered ontology with the fundamental classification at the core, surrounded by the context, which allows further interpretation of the relationship.

The layers in the ontology reflect a 'separation of concerns' among the different perspectives that contribute to defining relationship semantics. They also reflect the fact that we move from concrete to more abstract in the definition of these layers. The first layer, therefore, reflects core categories such as those analyzed rigorously in prior work. The second and third layers allow us to incorporate local and external context, respectively. The local context is provided by entities surrounding the relationship verb phrase. The external context is provided by the domain in which the relationship is being specified. The layers allow us to understand and specify the semantics of a relationship across these layers.

3.1. The core: Fundamental categories

The fundamental categories consist of classes of relationships that provide a categorization of relationships encountered in conceptual database design. These categories cannot be restricted to the ideal (Bunge's) characterization of relationships as 'interactions' among 'things.' Instead, we propose three general classes of relationships that reflect how relationships appear in conceptual database design: *status, change in status,* and *interaction*. These are appropriate because they capture three important orientations displayed by one entity towards another in conceptual database designs.

The first, '*Status*,' represents durable orientations that capture consequences of events, which is one of the ontological constructs from Bunge [1977; 1979]. The notion of durability is likely to be different in different domains, which underscores the need to take into account the external or domain context, the third layer in the ontology. For many databases, it is not only the event that is important but also

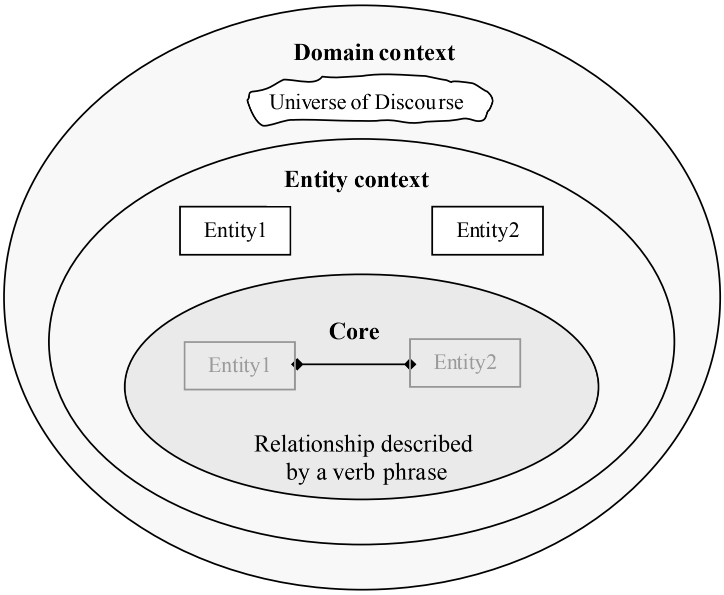


Fig. 1. Classifying a Relationship.

the consequence of the event. The '*Status*' orientation captures this consequence. For example, the event 'acquiring ownership' may lead to a durable orientation of ownership between an owner and an asset. The *Status* category captures this fact as one thing having a durable or permanent orientation with respect to the other. These enduring relationships between two entities express that one entity (A) <is something> with respect to the other entity (B), e.g. A is-owner-of B.

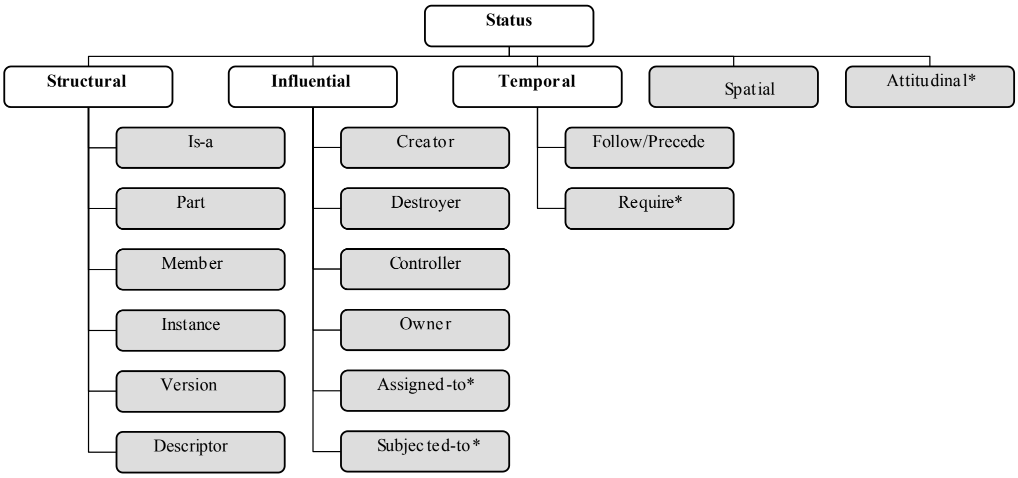
The second, '*Change of Status*' represents the visible consequences of an event as the transition of an entity from one status to another. These transitions are essentially connected to the '*Status*.' This category, thus, captures the event itself, manifested through a change of status. Relationships in the *Change of Status* category, accordingly, express the fact that an entity (A) is transitioning being something with respect to another entity (B) to being something else with respect to that entity, e.g. A <acquires-ownership-of> B.

The third, '*Interaction*' represents a short duration communication or other operation between two entities. The notion of short duration communication (like that of durability) is likely to be different in different domains, which underscores the need to take into account the external or domain context, the third layer in the ontology. Unlike the first two cases where the event is remembered as a change of status or the consequence of an event is remembered as a durable status; this category captures the event as an interaction. This is particularly useful when the event does not lead to a change in status, but is still important to remember. The interaction category, therefore, expresses the fact that an entity (A) <is doing something> with respect to another entity (B) e.g. A <operates> B.

The three categories, thus, respect the relevant constructs from Bunge's ontology (see Table 1). All three categories have a dynamic event as the source of the resulting relationship. The first, *status*, represents a durable consequence of an event. The second, *change of status*, represents a manifestation of an event as a transition from one status to another. The third, *interaction*, represents the event itself such as a communication between two entities or an operation of one entity by another. Table 2 summarizes these fundamental categories.

Table 2
Fundamental Categories

Category	Description	Example
Status	The durable orientation of one entity towards the other entity	A <is-owner-of> B
Change of Status	Transition of status of one entity with respect to the other	A <becomes-owner-of> B
Interaction	Communication between or operation of one entity by the other	A <operates> B



Legend: Leaf nodes (in grey) indicate status primitives, asterisks represent additions to prior work

Fig. 2. Primitives for the Category 'Status'.

These fundamental categories provide a *comprehensive* classification of all relationships in conceptual database design. Together, they provide complete and non-overlapping categories for classification of relationships. However, they are coarse groupings that are likely to be limited for comparing relationships. Each category, therefore, requires finer classes that must distinguish between the large set of relationships likely to fall within it. A hierarchy of classifications is, therefore, necessary to further refine each fundamental category.

3.1.1. Refining the category: Status

The category *Status* has been the subject of prior research on data abstractions (e.g. [Smith and Smith 1977; Brachman 1983; Goldstein and Storey 1999]). Building on this research, and incorporating results from research on patterns [Hay and Barker, 1996] and linguistics [Miller, 1990], we have constructed a hierarchical classification that further refines this category. The compilation from prior research was supplemented and enhanced by additional primitives discovered during iterative analysis of examples. The hierarchy of classifications contains primitives at the lowest level of classification. Figure 2 shows this finer classification of the category *Status*.

The five major sub-categories – *Structural, Influential, Temporal, Spatial and Attitudinal* – are labels that communicate the meaning inherent in that sub-category. Figure 2 shows these as the first level of classification for the category '*Status*.' Of these, two (*spatial* and *attitudinal*) are primitives. The sub-category '*spatial*' refers to relationships that describe the location of entities in space such as City <is on> Longitude>. The sub-category '*attitudinal*' captures the emotional disposition of an entity towards another such as likes and dislikes. An example of this primitive is Customer <likes> Product. The others

Table 3
Primitives for the category ‘Status’

	Primitive	Example	Source
1	A <is a> B	Pilot <is an> Employee	[Brachman 1983]
2	A <is member of> B	Professor <is member of> Department	[Brodie 1981]
3	A <is part of> B	Car <has> Engine	[Smith and Smith 1977]
4	A <is instance of> B	Video Tape <is copy of> Movie	[Motsching-Pitrik and Mylopoulous 1992]
5	A <is version of> B	Draft <is version of> Manuscript	[Motsching-Pitrik, 2000]
6	A <is descriptor of> B	Document <defines> Task	[Larman 1997, p. 156]
7	A <is creator of> B	Author <writes> Book	[Gamma et al. 1995, p. 87]
8	A <is destroyer of> B	Tennant <terminates> Lease	[Gamma et al. 1995, p. 266]
9	A <is owner of> B	Company <owns> Building	[Larman 1997, p. 157]
10	A <is in control of> B	Manager <leads> Team	[Larman 1997, p. 156]
11	A <is assigned to> B	Employee <assigned to> Project	*Added, see above*
12	A <is subjected to> B	Industry <regulated by> Law	*Added, see above*
13	A <follows or precedes> B	Rental <follows> Reservation	[Hay 1996, Chp. 5]
14	A <requires> B	Construction <requires> Approval	*Added, see above*
15	A <is next to> B	Office <is-next-to> Elevator	[Larman 1997, p. 156; Hay 1996, p. 36]
16	A <has attitude towards> B	Customer <likes> Product	Added, see above

(*structural, influential* and *temporal*) are further refined based either on prior research or by adding four new primitives (marked with an asterisk in Fig. 2). The first, ‘*assignment*,’ refers to the allocation of one thing to another. An example of this primitive is Employee <assigned to> Project.

The second and third new primitives represent variations of this idea. The second, ‘*subjection*,’ refers to the situation, when *A* <subjected-to> *B* indicates that A does not have a choice with respect to its assignment relationship with B. An example of this primitive is Industry-is regulated by-Law. The third, ‘requirement,’ captures a similar situation, when *A <requires> B* indicates that A is a prerequisite for B. An example of this primitive is Construction-requires-Approval. It may be possible to extend this construct to entities other than actions i.e. beyond what the example suggests. However, such extensions are likely to be subsumed under other data abstractions e.g. Vehicle requires Engine may be subsumed by Engine part-of Vehicle. This is particularly true for relationships observed in practice. The notion of temporality is also important in characterizing the ‘requires’ semantics. Much prior work exists on this notion (see, e.g. [Allen, 1984]) that can be incorporated to further refine the ‘requires’ primitive. Table 3 details the primitives with the source and an example for each.

3.1.2. Refining the category: Change of status

The change-of-status primitives, in conjunction with the status primitives, capture the lifecycle transitions for each status. Although the idea of a lifecycle has been alluded to previously [Hay and Barker, 1996], research on data abstraction has not systematically recognized the lifecycle concept. Our conceptualization of the ‘Change of Status’ category is based on an extension and understanding of how each primitive in the ‘Status’ category may evolve during the business lifecycle. Each stage in the relationship lifecycle may, therefore, be mapped to different status primitives. Our notion of the lifecycle, thus, explicitly recognizes the often implicit consideration of lifecycle in an ontology.

Consider relationships that deal with acquiring something, as is typical of business transactions and thus, related to the status primitive ‘is-*owner*-of.’ For this example, the lifecycle starts with needing something (‘*has-attitude-towards*’ and ‘*requires*’) which is followed by intending to become an owner (‘*acquire*’ or ‘*create*’), owning (‘*owner*’ or ‘*in-control-of*’) and giving up ownership (‘*seller*’ or ‘*destroyer*’). These sub-classes, then, represent ideas similar to constructing entities at different states such as ‘applicant,’ ‘screened-applicant,’ and ‘ employee’ at different stages of an employment cycle.

Table 4
Primitives for the Category 'Change of Status'

	Primitive		Example
1	A <wants-to-be owner of> B	*intent*	Customer <wants to own> Product
2	A <attempts-to-become owner of> B	*attempt*	Customer <orders> Product
3	A <becomes owner of> B	*transition*	Customer <receives> Product
	Status Primitive: Customer <owns> Product		
4	A <dislikes-being owner of> B	*intent*	Company <wants to sell> Product
5	A <attempts-to-give-up ownership of> B	*attempt*	Company <offers> Product
6	A <gives-up ownership-of> B	*transition*	Company <sells> Product

Table 5
Primitives for the Category 'Interaction'

	Primitive	Example
1	View Status	Analyst <analyses> Requirements
2	Select	Customer <selects> Product
3	Communicate	Modem <negotiates with> Phone Line
4	Perform	Developer <tests> Software
5	Operate	Pilot <flies> Plane
6	Serve	Employee <serves> Customer
7	Manipulate	Instructor <grades> Exam
8	Transmit	Bank <remits> Payment
9	Receive	Warehouse <receives> Shipment

The primitives can, therefore, be used to illustrate a lifecycle that goes through the process of creation or acquisition, ownership, and destruction. The life cycle can be logically divided into: intent, attempt to acquire, transition to acquiring, intent to give up, attempt to give up, and transition to giving up. Table 4 shows this information superimposed on the different states within the lifecycle. The sub-column under the change-of-status primitives shows the meanings captured in each: intent, attempt and the actual transition.

3.1.3. Refining the category: Interaction

'Interaction' describes short duration communication between two entities or an operation of one entity on another. It is the manifestation of an event that may or may not cause a change in one of the entities. For example, an entity may 'manipulate' another entity [Miller *et al.*, 1990], or it may cause movement of the other through time or space (e.g. 'transmit,' 'receive'). Two entities (e.g. subject and object) may interact without causing change to the (object) entity (e.g. 'communicate with,' 'observe'). One entity may interact with another also by way of performance ('operate,' 'serve'). It is possible to argue that some of these interactions, particularly the interactions 'causing change' may lead to a change in status of the entity that is changed. This change of status in the 'entity,' however, should be distinguished from a change of status in the 'relationship,' which is the focus of the previous category. To our knowledge, no prior systematic investigations of interaction primitives are available. The hierarchy we have generated (Fig. 3), along with the primitives (Table 5), therefore, represent the outcome of a top-down and bottom-up analysis that was driven by descriptions of relationships in database design, and examples of relationships from real-world databases.

3.2. The context

The fundamental categories, along with sub-classes of relationship types developed so far provide a first step towards understanding relationships in database design. Important considerations of context

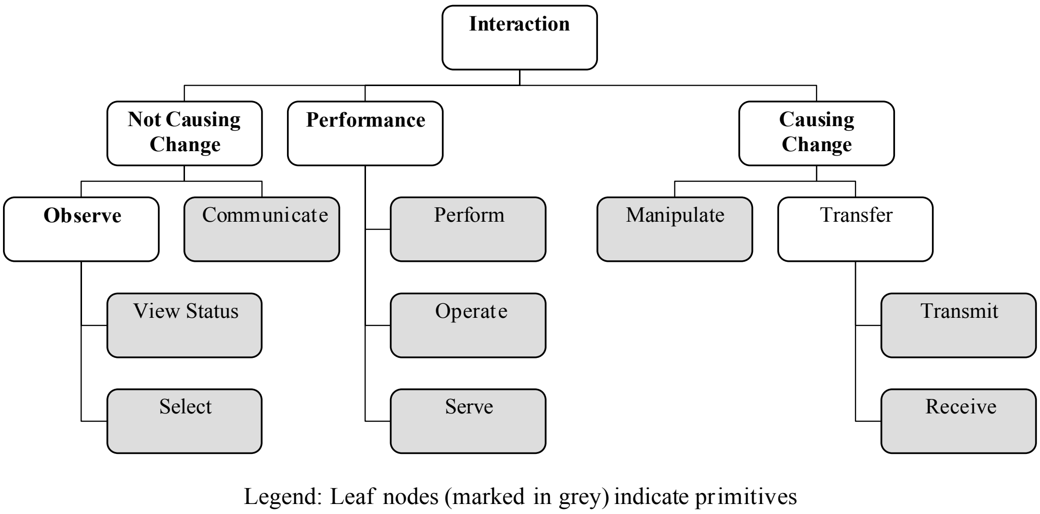


Fig. 3. Primitives for the Category 'Interaction'.

must be incorporated along with these classifications to provide an adequate ontology. Two contexts are necessary for this purpose. The first is the local (entity) context that takes into account the nature of entities that participate in the relationship. For example, the relationships 'Customer <opens> Account' and 'Key <opens> Lock' require very different interpretations that cannot be accurately made unless the participating entities are considered. This is captured by the first, local (entity) context. Consider another example. The relationship 'Customer <opens> Account' may mean one thing in a database being designed for a Bank. A similar relationship 'Theater <opens> Show' may portray something quite different in a database being designed for the arts. This is captured by the second, external (domain) context.

3.2.1. The local (*Entity*) *context*

The interpretation of a verb phrase is heavily dependent upon the nouns that surround it [Fellbaum, 1998]. Analogously, an understanding of entities is necessary for understanding the relationships. Prior research has proposed a way to classify entities by mapping them to a rather large set of common categories [Storey *et al.*, 1998]. We propose a simpler classification, comprised of entity *categories actor, action* and *artifact*.

– Actors are entities which are capable of performing independent action.
– Actions are the performance of an act.
– Artifacts are inanimate objects not capable of independent action, including, for example, notions such as place.

Finer classifications can be mapped to this simple classification but are not necessary for the purposes of this research. In addition to confirming to prior work on entity ontologies, this classification also provides a bridge to constructs from Bunge's ontology. Two classifications, 'Actor' and 'Artifact' map to a 'thing,' whereas the classification 'Action' maps to an 'event.' After the entities have been classified, valid primitives for the verb phrase can be identified. For example, it does not make sense to allow the primitive 'perform' for two entities of the kind 'Actor.' On the other hand, the primitive is appropriate when one of the entities is of the kind 'Actor' and the other is of the kind 'Action.' The argument can be

Table 6
Valid 'Status' Primitives based on Entity Context

Entity 1	Entity 2	Valid Status Primitives (see Table 2)															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Actor	Actor	√	√		√			√	√	√	√	√				√	√
Actor	Action									√	√	√	√				√
Actor	Action						√	√	√	√	√	√					√
Actor	Artifact	√		√	√	√							√	√	√		
Action	Action						√						√				
Action	Artifact	√		√	√	√	√									√	

Table 7
Valid 'Interaction' Primitives based on Entity Context

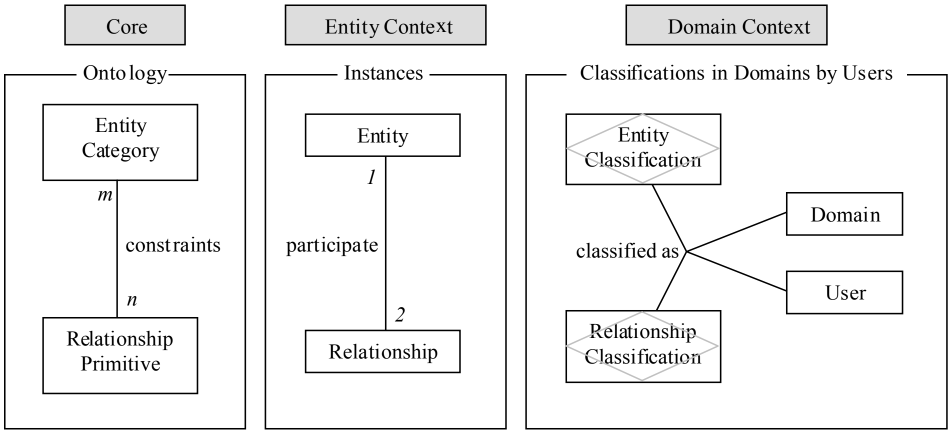
Entity 1	Entity 2	Valid Interaction Primitives								
		1	2	3	4	5	6	7	8	9
Actor	Actor			√			√			
Actor	Action	√	√		√					
Actor	Artifact	√	√	√		√	√	√	√	√
Action	Action									
Action	Artifact	√	√		√					
Artifact	Artifact	√	√	√		√	√	√	√	√

Table 8
Constraints on valid status primitives based on entity context

	Primitive	Rule
1	A <is a> B	both entities should belong to the same category
2	A <is member of> B	both entities should belong to the category 'actor'
3	A <is part of> B	both entities should belong the same category, which should be 'action's or 'artifact's
4	A <is instance of> B	both entities should belong to the same category
5	A <is version of> B	both entities should belong the same category, which should be 'action's or 'artifact's
6	A <is descriptor of> B	one of the entities (descriptor) should belong to the category 'artifact'
7	A <is creator of> B	one of the entities (creator) should belong to the category 'actor,' and the other should be 'actor' or 'artifact.'
8	A <is destroyer of> B	one of the entities (creator) should belong to the category 'actor,' and the other should be 'actor' or 'artifact.'
9	A <is owner of> B	one of the entities (owner) should belong to the category 'actor'
10	A <is in control of> B	one of the entities (controller) should belong to the category 'actor'
11	A <is assigned to> B	one of the entities (assignee) should belong to the category 'actor'
12	A <is subjected to> B	one of the entities (the regulator) should belong to the category 'action'
13	A <follows or precedes> B	both entities should belong to the category 'action'
14	A <requires> B	both entities should belong to the category 'action'
15	A <is next to> B	both entities should belong to the same category, which should be 'action's or 'artifact's
16	A <has attitude towards> B	at least one of the entities should belong to the category 'actor'

applied to *Status* as well as *Interaction* primitives. Because the *Change of Status* primitives capture the lifecycle of *Status* primitives, constraints identified for *Status* primitives apply to the *Change of Status* primitives as well. Tables 6 and 7 show these constraints. They identify the subset of valid primitives for each pair (Actor-Actor, Actor-Action etc.) for valid *status* and *interaction* primitives based upon the context of the entities.

Table 6 shows the permissible *Status* primitives based on Entity context. Because the *Change of Status* primitives represent variations of a given *Status*, these restrictions can also be translated to Change of Status primitives. Table 7 shows the permissible *Interaction* primitives based on entity context.



Legend: The diagram follows a variation of the entity-relationship notation.
The labels at the top, shaded in gray, suggests mapping to the layers (see figure 1)

Fig. 4. Structure of a Repository of Relationship Verb Phrases.

3.2.2. The global (*Domain*) *context*

The meaning of relationships can also vary based on the application domain for which the database is being designed. The same 'relationship' (i.e. labeled with identical verb phrase) may be classified in one manner for one domain and quite differently for a different domain. Consider the verb 'opens' in a theatre database versus a bank database. The relationship Character <opens> Door might occur in the theatre domain, which maps to the interaction primitive <manipulates>. In the bank application, Teller <opens> Account may map to the status primitive <is-creator-of>; Customer <opens> Account may map to <is-owner-of>. Past attempts to capture taxonomies of such context-dependent verbs have required a great deal of manual effort (e.g., CYC [Lenat *et al.*, 1995]; WordNet [Fellbaum, 1998]). We, therefore, take a more pragmatic approach, where a repository of context-dependent verb phrases, classified following the multi-level ontology outlined above, can be constructed over time as the ontology is being used. Such a repository may be developed following the structure in Fig. 4. Each relationship verb phrase stored in the repository will be accompanied by classification from one or more users, along with a classification of the entities that connect that relationship verb phrase.

Such a repository may be built up over time, providing future users readymade classifications of verb phrases that they may use, refine, or change. For example, the verb phrase "screens" may be available in the repository as part of in 'Employee screens Applications' with the classifications 'actor' (for employee), 'view-status' (for 'screens'), and 'artifact' (for applications). It may also be available as part of 'Curtain screens Furniture' with the classifications 'actor' (for curtain), 'manipulate' (for 'screens'), and 'artifact' (for furniture). This would facilitate automated matches (based on domain or entity context) with which a future user may agree or decide to override these. The domain context provided by such a populated repository, however, can be difficult to achieve.

3.3. Implementation

A preliminary implementation of the ontology has been completed in the form of a knowledge base designed using a relational database. Preliminary user interfaces have also been created using Java and HTML that can connect to the back-end knowledge-base. The architecture of this prototype consists of a classification module (facilitating classification of a given relationship), and a comparison module

Table 9
Constraints on valid interaction primitives based on entity context

	Primitive	Rule
1	Observe	one of the entities (who will view status) should be either an ‘actor’ or an ‘artifact,’ and the other should be ‘action’ or ‘artifact’
2	Select	one of the entities (who will select) should be either an ‘actor’ or an ‘artifact’ and the other should be ‘action’ or ‘artifact’
3	Communicate	the entities could be any combination of ‘actor’s and ‘artifact’s
4	Perform	one of the entities should be an ‘actor’ or ‘artifact’ (performer), and the other should be an ‘action’ (that is performed)
5	Operate	one of the entities should be an ‘actor’ or ‘artifact’ (operator), and the other should be an ‘artifact’ (that is operated)
6	Serve	the entities should be any combination of ‘actor’ or ‘artifact’
7	Manipulate	one of the entities (the manipulator) should be an ‘actor’ or an ‘artifact,’ and the other should be an ‘artifact’
8	Transmit	one of the entities (the transmitter) should be an ‘actor’ or an ‘artifact,’ and the other should be an ‘artifact’
9	Receive	one of the entities (the receiver) should be an ‘actor’ or an ‘artifact,’ and the other should be an ‘artifact’

(facilitating comparison of relationships to assess their equivalence or otherwise). The knowledge-base and user interfaces created so far address the intended functionality of the first module, i.e. classification of new relationships. This module, in turn, consists of two sub-modules: the entity classification module, and the relationship classification module, which are invoked in sequence, with the results of the first providing constraints for the second. These constraints can be specified as a rule-base that describes ‘valid’ classifications of relationships based on entity context.

Following the rationale and results provided in Tables 6 and 7, a number of rules can be devised, described in an informal manner in Tables 8 and 9.

Based on these rules, a user interface has been devised that implements an interactive questioning scheme with the objective of minimizing the number of questions that must be asked to the user in order to classify a relationship. It follows the multi-level classification structure of the primitives (Figs 2 and 4) to guide the user to an appropriate classification. The scheme is designed to minimize the number of prompts to the user. Figure 5 illustrates the questioning scheme.

4. Assessment

An empirical analysis was carried out to assess the effectiveness of the ontology. The objective of doing so was to show proof of concept. In general, assessing an ontology is difficult because it can be put to use in a number of different ways. Gruninger and Fox [1995] suggest that ontology assessment should be guided by the overwhelming principle of the *competency* of an ontology, a notion similar to effectiveness of an information system. They define effectiveness as the extent to which the system accomplishes its objectives. By analogy, the *competency* of an ontology corresponds to its ability to satisfy a list of queries that a knowledge-base containing the ontology (and its instantiations) is able to answer (competency queries). The ontology may, therefore, be evaluated by posing questions such as: Does the ontology contain enough information to answer these types of queries? Do the answers require a particular level of detail or representation of a particular area? Noy and McGuiness [2001] extend the arguments provided by Gruninger and Fox [1995], suggesting that competency queries may be representative, but need not be exhaustive.

The primary objective for developing the relationship ontology was to facilitate the comparison and integration of relationships across databases. Accordingly, an assessment plan should address two

Now Classifying the Relationship: Customer bid on items

▼

How would you classify the entity 'customer' - *actor, action* or *artifact*? *User Selection*: **Actor**
How would you classify the entity 'item' - *actor, action* or *artifact*? *User Selection*: **Artifact**

▼

Is the verb phrase capturing:
Status - the orientation of one thing with the other
Change of status - change of orientation of one thing with the other
Interaction - communication or operation between the two things that does not result in a change in orientation
User Selection: **Change of status**

▼

Which best describe this verb phrase?
<< we need to incorporate use of the rule-base here >>
Customer wants - to - bid on item
Customer attempts - to - bid on item
Customer bids on item
Customer dislike - bidding on item
Customer attempts - to – bid on item
Customer give - up- bidding - on item
User Selection: **customer bids on item**

a. Example 1

Now Classifying the Relationship: Pilot flies plane

▼

How would you classify the entity 'pilot' - *actor, action* or *artifact*? *User Selection*: **Actor**
How would you classify the entity 'plane' - *actor, action* or *artifact*? *User Selection*: **Artifact**

▼

Is the verb phrase capturing:
Status - the orientation of one thing with the other
Change of status - change of orientation of one thing with the other
Interaction - communication or operation between the two things that does not result in a change in orientation
User selection: **Interaction**

▼

Which best describe this verb phrase?
Causing change
Not causing change
Performance
User Selection: **performance**

▼

Which best describes it further?
Perform
Operate
Serve
None of the above
User Selection: **operate**

▼

Addition to knowledge base: flies is classified as operate

b. Example 2

Fig. 5.

objectives. First, it should allow classification of relationships encountered in real-world database designs. Second, it should allow use of the instantiated ontology (i.e. a knowledge-base of classified relationships) to compare relationships across databases. The assessment reported in this paper, while carried out on a limited scale, addresses the first objective. For the second objective, we provide examples of how the ontology can be used for comparing relationships across databases. A controlled experiment that assesses effectiveness of the ontology for comparison to relationships (second objective) would

Table 10
Sample test relationships

Example with <verb>	Domain	Meaning(s)
Manufacture <imports> Part	Manufacturing	to buy or bring in (products) from another country
Student <obtains> Degree	Education	to get (something), esp. by asking for it, buying it, working for it or producing it from something else
Agent <collects> Ticket	Airline	to gather together from a variety of places or over time
Supermarket <accepts> CreditCard	Retail	to agree to take (something), or to take (something) as satisfactory, reasonable, true
Traveler <hires> Car	Service	to pay to use (something) for a short period or to pay (someone) to do a job temporarily

require a considerably large knowledge-base of relationship instances as well as additional mechanisms such as computing conditional confidence factors of classifications based on domain and user information. These are beyond the scope of the research reported in this paper.

4.1. Assessing adequacy of the ontology for classifying relationships

To address the first objective, the ontology was used to classify a published set of verb phrases. These were acquired from the SPEDE project [Cottom *et al.*, 2000], the Wordnet project [Fellbaum, 1998], and the online Cambridge dictionary [http://dictionary.cambridge.org]. To ensure that the researchers would be knowledgeable in classifying the relationships, only business-related verbs were retained. Since the SPEDE verbs [Cottam, 2000] were developed for business applications, these automatically became part of the sample set. The researchers independently selected business-related verbs from a set of 700 verbs generated randomly from WordNet [Miller, 1990]. Verbs that were common to the selections made by both researchers were added to the list from SPEDE. The same procedure was carried out for a set of 300 verbs that were randomly generated upon the researchers' request by the support staff at the online dictionary http://dictionary.cambridge.org/. This resulted in a total of 198 business verbs. For each verb in this list of 198 verbs, a definition was obtained from the Cambridge on-line dictionary. Examples provided for understanding the verb phrase at this dictionary was then used to generate the relationships in seven application domains. This resulted in approximately 30 relationships in each of the following domains: education, business management, manufacturing, airline, service, marketing, and retail. Table 10 shows examples generated from this list of verbs. The examples in the table were obtained from the SPEDE project.

The relationships were classified using the relationship ontology. The procedure for the classification phase involved the following steps. First, 30 verbs were classified by the researchers independently and the classifications were assessed for correspondence across these independent classifications. This initial assessment showed that the researches agreed on 24 out of the 30 cases i.e. 80% of the cases. The disagreements were almost entirely due to inconsistent interpretations of the relationships by the researchers, suggesting the possibility that some relationships could be interpreted, and hence, classified differently by different users. During these discussions, the different classifications were justified, when the researchers analyzed different interpretations of the relationship in question.

One relationship was 'Manufacturer imports Part.' While one researcher suggested that this denoted physical movement suggesting the classification 'receive,' another researcher suggested that this denoted the acquiring of ownership, suggesting the classification 'becomes-owner-of.' It is important to emphasize that these disagreements did *not* suggest problems with the classification scheme itself. For example, upon sharing the first interpretation (how 'import' suggests physical movement), the second researcher agreed to a possible classification of 'receive' for this relationship. The researcher, however,

Table 11
Sample classifications of relationships

Entity 1	Verb Phrase	Entity 2	Classification	
Manufacturer	Imports	Part	Interaction	receives
Student	acquires	Textbook	Change of status	becomes-owner-of
Air traffic controller	establishes	Flight path	Change of status	becomes-creator-of
Salesperson	converts	Competitor-customer	Interaction	Manipulates
Customer	enters-into	Sales agreement	Change of status	becomes-subjected-to
Caterer	Delivers-to	Plane	Status	is-assigned-to
Teacher	distributes	Handout	Interaction	Transmits
Airline	Adjusts	Schedule	Interaction	Manipulates

maintained that the alternative interpretation (how 'import' suggests acquiring ownership) was valid, suggesting the classification 'become-owner-of.' A more problematic circumstance would occur if the researchers agreed on the interpretation of the relationship but disagreed about the classification following the ontology. None of the mismatches in classifications exhibited this form of disagreement.

A useful device that allowed the resolution of disagreements was discussing whether the researches interpreted the relationship as 'an event' (suggesting classification as either *Interaction* or *Change of Status*) or an 'outcome of an event' (suggesting classification as *Status*). This framing appeared to resolve many of the disagreements, allowing a researcher to argue for his or her interpretation and also understand the alternative suggested by the co-researcher. The remaining relationships were then classified by the researchers independently following the ontology. For example, the relationship 'Air-traffic-controller establishes Flight-path' was classified as 'Change of status' with the primitive 'becomes-creator-of.' Table 11 shows a sample of classifications performed.

The results were encouraging, especially given our focus on evaluating the competency of the ontology [Gruninger and Fox, 1995] for classifying relationships encountered in databases. First, it facilitated the classification of *all* relationships suggesting that the ontology was adequate for classifying relationships. Because the sample included verbs that represent processes (see, for example, the SPEDE taxonomy), there was a high occurrence of interaction primitives. The biggest difficulty involved arriving at an interpretation of the relationship based on information about the domain, and identifying how to move from one level to the next in the hierarchy of classifications. For example, the relationship 'Student acquires Textbook' was immediately classifiable by one of the primitives. In other cases, the next layer was necessary. The results for relationship classification were recorded as matches, mis-matches or partial matches.

A classification was recorded as a complete match (agreement at all levels of the ontology), partial match (agreement at some of the levels of the ontology) or no match (agreement at none of the levels of the ontology). When a complete match was not found, an attempt was made to understand the reasons for the discrepancy. The mismatches between classifications by the two researchers were often traced to a different interpretation of the relationship, and a focus on the occurrence of an event versus the outcome of an event. The mismatches were, therefore, largely restricted to the first level of classification (i.e. *Status* vs. *Change of Status vs. Interaction*). Once this level of agreement was achieved, the primitive selected for classification by the different researchers matched in most cases. Table 12 shows the results of this independent classification by the two co-researchers with a brief analysis.

The table shows entity classifications, which achieved a 98% match following the simple taxonomy of actor, action and artifact. The lower rows in the table show the number of relationship classifications that were equivalent across the two researchers. These numbers indicate a high degree of fidelity. The row named "Core-Category-level" shows that as many as 181 (91.41%) of the relationships were

Table 12
Analysis of relationship classifications

	Analysis		
	Match		Comments
Entity classifications			
Entity 1	193	97.47%	Resolved by dialog: 8
Entity 2	194	97.98%	Resolved by dialog: 12
Relationship classifications			
Core-Category-level	181	91.41%	Resolved by dialog: 11
Core-level and Primitive	170	85.86%	Resolved by dialog: 14

classified by the two researchers in an equivalent manner at the top level i.e. as *Status*, *Change of Status* or *Interaction*. The final row in the table, labeled "Core-Category-level and Primitive" shows that 170 (85.86%) relationships were classified by the researchers in an equivalent manner at the Core Category level as well as the Primitive level within the category. The comments column shows the classifications that were resolved by dialog. These indicated instances where one researcher's interpretation was obviously flawed (or a typographical error was made), and was corrected immediately on observing the competing classification provided by the other researcher.

The last two rows in the table may be analyzed to understand Complete Match (Core-Category plus Primitive) or Partial Match (Match at the Core Category level but not at the Primitive level). These numbers, therefore, show that classifications for 170 relationships found a complete match, and a further 11 were matched partially. This also suggested that the classifications did not match for 17 relationships. To further understand these problems, the researchers recorded their interpretation of the relationship as either 'occurrence of an event' or 'consequence of an event.'

Consider, for example, one of the relationships the researchers encountered: 'Foreman determines Procedure.' One researcher classified this as an event occurrence, classifying the relationship as Core-category: *Interaction*, Primitive: Select. The other researcher classified this as a consequence of an event, classifying the relationship as Core-category: *Status*, Primitive: in-control-of. Of the 17 relationships classified differently by the researchers, 12 could be reconciled by understanding the researchers' framing of the relationship as either 'occurrence of an event' or 'consequence of an event.' If the alternative framing was taken into account, the classifications could be reconciled. Only five relationships, therefore, represented disagreements among researchers for classifying the relationships.

The results suggest that the breaking point for classification of the relationships is the framing of the relationship, either as the event or its consequence. By allowing multiple classifications that depend upon these framings, a robust implementation of a knowledge-base can overcome this problem.

4.2. Demonstration of use of the ontology for comparing relationships

A classification procedure, such as that outlined in the previous section, can populate a knowledge-base of relationships along with their classifications. Such a knowledge-base can then be used to compare relationships across databases, for the purpose of mapping or integration. Table 13 shows examples of comparing relationships using the fundamental categories (status, change of status, and interaction primitives), and the local context (entities surrounding the relationships). The approach can be further refined with the use of the domain context.

The table shows several examples of how relationships can be compared across databases. A well-populated knowledge-base of relationships along with classifications would be a prerequisite for (semi)automating such comparison. The comparison would require additional mechanisms that leverage

Table 13
Demonstration of relationships comparison

Relationships and Entity Classifications	Possible Relationship Classifications	Comparison of Relationship Classifications
R1=Contractor builds Bridge (Actor–Artifact) R2=Contractor has Employee (Actor–Actor)	No need to compare relationships because entity classifications are different, verb phrase classification not attempted	⇒ *Non-equivalent relationships*
R1=Contractor builds Bridge (Actor-Artifact) R2=Worker does Speeding (Actor-Action)	No need to compare relationships because entity classifications are different, verb phrase classification not attempted	⇒ *Non-equivalent relationships*
R1=Contractor builds Bridge (Actor-Artifact) R2=Builder constructs Tree house (Actor-Artifact)	Entity classifications match. If no matching instances are found in the repository, verb phrase classification can yield possible *status* primitives: 6, 7, 8, 9, 10, 11, 16; possible *change of status* primitives: lifecycle changes for the *status* primitives indicated above; and possible *interaction* primitives: 1, 2, 3, 5, 6, 7, 8, 9 (see Tables 6 and 7)	R1: *status* primitive 7 (is-creator-of) R2: *status* primitive 7 (is-creator-of) ⇒ *Equivalent relationships*
R1=Manager fires Employee (Actor-Actor) R2=Manager employs Worker (Actor-Actor)	Entity classifications match. If no matching instances are found in the repository, verb phrase classification can yield possible *status* primitives: 1, 2, 4, 7, 8, 9, 10, 11, 15, 16; possible *change of status* primitives: lifecycle changes for the *status* primitives indicated above; and possible *interaction* primitives: 3, 6 (see Tables 6 and 7)	R1: *change of status* primitive 'gives-up' along with *status* primitive 10 (is-in-control-of) R2: *change of status* primitive 'acquires' along with *status* primitive 10 (is-in-control-of) ⇒ *Non-Equivalent relationships*

similar and dissimilar classifications of a relationship for the purpose of suggesting appropriate classifications. These mechanisms may consider additional information such as the past history of appropriate classifications by users, domain expertise exhibited by users or even conflicting user classifications provided by different users. These are outside the scope of the research reported in this paper.

5. Discussion and conclusion

The results obtained were very encouraging. Although the researchers participated in the evaluation, the test-suite was carefully developed by identifying relationship verb phrases in various domains from independent sources. The results are also significant because of the strong correlation among researchers during the classification. The empirical analysis, thus, demonstrates an initial validity of the ontology. Clearly, further empirical analysis could be conducted to assess coverage, expressiveness, completeness and soundness, perhaps involving participation from a number of users. One challenge would be to ensure that the users understand the different classifications and levels. Potential avenues to overcome this challenge include training users on a set of examples, describing scenarios appropriate for comparison of relationships across databases, and providing detailed descriptions of domains that the users can draw on to assess the similarity or differences across relationships in the two databases.

In conclusion, we have presented an ontology for classifying relationship verb phrases that would assist in automated comparison of conceptual database designs. The ontology was developed by adapting and extending considerable prior research related to understanding of relationships in databases, linguistics, and ontology development. One caveat for potential uses is that our classification may not be exhaustive although the analysis reported in the paper shows that a significant number of relationships from the universe of interest are, indeed, classifiable using the ontology. A second caveat results from our adaption of Bunge's constructs and use of prior work on data abstraction. These two can sometimes be at odds with one another particularly because of our desire to include 'not-so-perfect' relationships in our classification. The result is a mixing of formal (e.g., part of, member of, instance of) and material relations (e.g., flying to, buying from). These caveats provide avenues for future research.

A pilot test of the ontology illustrates its potential for classifying relationships encountered in real-world database designs. Further research is needed to refine the knowledge acquisition component, complete the implementation, and populate a knowledge-base using this implementation. Further empirical analysis is clearly necessary to further test the adequacy of the ontology for coverage, expressiveness, completeness, and soundness. This should demonstrate that the ontology is effective for its intended purpose of comparing relationships across databases for integration. A more effective test-suite may include comparisons against other approaches proposed for database schema integration such as those drawing on instance populations. Other applications of the ontology are also possible; for example, it may be possible to use the ontology to compare ontology-based relationships. Such additional application would, however, require crafting appropriate scenarios to ensure that the ontology provides coverage, expressiveness, completeness and soundness for these new applications. These concerns remain on our future research agenda.

References

J.F. Allen, (July, 1984). Towards a general theory of action and time, *Artificial Intelligence* **23**(2), 123–154.

M. Bergholtz and P. Johnannesson, (28–29 June 2001). *Classifying the Semantics of Relationships in Conceptual Modelling by Categorization of Roles*, Proceedings of the 6th International Workshop on Applications of Natural Language to Information Systems (NLDB'01), Madrid, Spain.

P. Bera and Y. Wand, (November 4–6, 2004). *Analyzing OWL using a Philosophy-Based Ontology*, Proceedings of International Conference on Formal Ontology in Information Systems (FOIS 2004). Torino (Italy), IOS press, pp. 353–362.

J. Biskup and D.W. Embley, (2003). Extracting Information from Heterogeneous Information Sources using Ontologically Specified Target Terms, *Information Systems* **28**(3), 169–212.

R.J. Brachman, (October, 1983). What IS-A is and Isn't: An Analysis of Taxonomic Links in Semantic Networks, *IEEE Computer*, 30–36.

M.L. Brodie, (1984). On the Development of Data Models, in: *On Conceptual Modeling*, M.L. Brodie, J. Mylopoulos and J.W. Schmidt, eds, Springer-Verlag, New York, NY, pp. 19–47.

M. Brodie, (1981). *Association: A Database Abstraction*, Proceedings of the Entity-Relationship Conference.

M. Bunge, (1977). Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World. D. Reidel Publishing Co., Inc., New York, NY.

M. Bunge, (1979). Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems. D. Reidel Publishing Co., Inc., New York, NY.

R. Chaffin and D.J. Herrmann, (1988). The nature of semantic relations: A comparison of two approaches, in: *RelationalModels of the Lexicon: Representing Knowledge in Semantic Networks*, M. Evens, ed., New York: Cambridge University Press, pp. 289–334.

R. Chaffin and D.J. Herrmann, (1987). Relation elementy theory: A new account of the representation and processing of semantic relations, in: *Memory and Learning: The Ebbinghaus Centennial Conference*, D. Gorfein and R. Hoffman, eds, Hillsdale, N.J.: Lawrence Erlbaum Associates Inc., pp. 221–245.

R. Chaffin and D.J. He∼tmann, (1984). The similarity and diversity of semantic relations, *Memory and Cognition* **13**(2), 134–141.

R. Chaffin, D.J. Herrmann and M. Winston, (1988). An empirical taxonomy of part-whole relations: Effects of part-whole type on relation identification, *Language and Cognitive Processes* **3**(1), 17–48.

P.E. Chen, (1976). The entity-relationship model: Toward a unified view of data, *ACM Transactions On Database Systems* **1**(1), 9–36.

P. Chen, (1993). *The Entity-Relationship Approach*, In Information Technology in Action: Trends and Perspectives, Englewood Cliffs: Prentice Hall, 13–36.

P. Coad et al., (1995). *Object Models: Strategies*, Patterns, & Applications. Prentice Hall.

H. Cottam, (2000). *Ontologies to Assist Process Oriented Knowledge Acquisition,* http://www.spede.co.uk/papers/papers.htm.

H. Cottam, N. Milton and N. Shadbolt, (2000). *The Use of Ontologies in a Decision Support System for Business Process Re-engineering,* http://www.psychology.nottingham.a...search/ai/themes/ka/UseofOnto.html.

Dahchour, (March, 2001). Integrating generic relationships into object models using metaclasses. PhD thesis, Department of Computing Science and Engineering, Université catholique de Louvain, Belgium.

M. Dahchour, (2003). *An advanced metamodel for generic relationships*, Proc. of the workshop on information technologie, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), Rabat, Morocco, March 17–19.

K. Dahlgren, (1995). A Linguistic Ontology, *International Journal of Human-Computer Studies* **43**, 809–818.

K. Dahlgren, (1988). *Naive Semantics for Natural Language Understanding*, Kluwer Academic Publishers, Hingham, MA.

J. Davis and R.D. Bonnell, (March, 1991). A Framework for Constructing Visual Knowledge Specifications in Acquiring Organizational Knowledge, *Knowledge Acquisition An International Journal* **3**(1), 79–113.

D. Dey, V.C. Storey and T.M. Barron, (December, 1999). Improving Database Design through the Analysis of Relationships, *ACM Transactions on Database Systems* **24**(4), 453–486.

J. Dullea and I.-Y. Song, (15–18 November 1999). A Taxonomy of Recursive Relationships and Their Structural Validity in ER Modeling, in: *Conceptual Modeling – ER'99, 18th International Conference on Conceptual Modeling,* J. Akoka et al., eds, Lecture Notes in Computer Science 1728, Paris, France, pp. 384–389.

D. Embley, (18–22 January, 2004). *Toward Semantic Understanding An Approach Based on Information Extraction Ontologies,* Proceedings of the Fourteenth Australian Database Conference, Dunedin, New Zealand, 3–12.

D. Embley, D.M. Campbell, Y.S. Jiang, Y.K. Ng, R.D. Smith, S.W. Liddle and D.W. Quass, (1999). *A Conceptual-modeling Approach to Web Data Extraction, Data & Knowledge Engineering*.

V. Fellbaum, *Introduction,* in Wordnet: An Electronic Lexical Database, The MIT Press, Cambridge, Mass., 1998, 1–19.

D. Fensel, (2003). *Ontologies: A silver bullet for knowledge management and electronic commerce*, Springer.

C.J. Fillmore, (1977). The Case for Case Reopened, *Studies in Syntax and Semantics* **8**, 59–81.

A. Gal, A. Anaby-Tavor, A. Trombetta and D. Montesi, (2003). A Framework for Modeling and Evaluating Automatic Semantic Reconciliation, *VLDB Journal*, 65.

E. Gamma, R. Helm, R. Johnson and J. Vlissides, (1995). *Design patterns: elements of reusable object-oriented software,* Addison-Wesley Longman Publishing Co., Inc., Boston, MA.

J.H. Gennari, A. Silberfein and J.C. Wiley, (2005). *Integrating genomic knowledge sources through an anatomy ontology,* Proceedings of the Pacific Symposium on Biocomputing 2005, 115–126.

R.C. Goldstein and V.C. Storey, (June, 1992). Unravelling IS-A Structures, *Information Systems Research* (*ISR*) **3**(2), 99–126.

R.C. Goldstein and V.C. Storey, (1999). Data Abstractions: Why and How, *Data and Knowledge Engineering* **29**(3), 1–18.

R.C. Goldstein and V.C. Storey, (October, 1994), Materialization, *IEEE Transactions on Knowledge and Data Engineering* **6**(5), 835–842.

T.R. Gruber, (1993), A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition* **5**, 199–220.

M. Gruninger and M.S. Fox, (1995). *Methodology for the Design and Evaluation of Ontologies,* in: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal, Canada.

N. Guarino and C. Welty, (2004). An Overview of OntoClean, in: *Handbook on Ontologies*, S. Staab and R. Studer, eds, Springer Verlag 2004, pp. 151–159.

M. Hammer and D. McLeod, (1981). Database description with SDM: A semantic database model, *ACM Trans. on Database Systems* **6**(3).

D.C. Hay and R. Barker, (1996). *Data Model Patterns: Conventions of Thought*, Dorset House.

R. Hull and R. King, (Sept. 1987). Semantic database modeling: Survey. applications and research issues, *ACM Computing Surveys* **19**(3), 201–260.

Z. Kedad and E. Metais, (15–18 November 1999). Dealing with Semantic Heterogeneity During Data Integration, in: *Conceptual Modeling – ER'99, 18th International Conference on Conceptual Modeling*, J. Akoka et al., eds., Lecture Notes in Computer Science 1728, Paris, France, pp. 325–339.

C. Larman, (1997). *Applying UML and Patterns*, Prentice-Hall.

D.B. Lenat, (1995). *CYC: A Large-scale Investment in Knowledge Infrastructure,* Communications of the ACM.

L. Mathiassen, A. Munk-Madsen, P.A. Nielsen and J. Stage, (2000). *Object-Oriented Analysis & Design*, Marko Publishing House.

G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K.J. Miller, (1990). Introduction to WordNet: An On-line Lexical Database, *International Journal of Lexicography* **3**(4), 235–244.

E. Moench, M. Ullrich, H.-P. Schnurr and J. Angele, Semantic Miner – Ontology-Based Knowledge Retrieval.

R. Motschnig-Pitrik and J. Myloppoulos, (1992). Class and Instances, *International Journal of Intelligent and Cooperative Systems* **1**(1), 61–92.

R. Motschnig-Pitrik and V.C. Storey, (August, 1995), Modelling of Set Membership: the Notion and the Issues, *Data and Knowledge Engineering* **16**, 147–185.

R. Motschnig-Pitrik, (2000), A Generic Framework for the Modeling of Contexts and its Applications, *Data and Knowledge Engineering*, **32** 145–180.

S. Navathe and R. Elmsari, (2003). *Fundamentals of Database Systems*, 3rd Edition. Addison-Wesley.

N.F. Noy and C.D. Hafner, (1997). The state of the art in ontology design: A survey and comparative review, *AI Magazine* **18**(3) 53–74.

N.F. Noy and D.L. McGuinness, (2001), *Ontology Development 101: A Guide to Creating Your First Ontology*, Available at http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html. Accessed 15 March 2004.

M. Rosemann and P. Green, (2002), Developing a meta model for the Bunge-Wand-Weber ontological constructs, *Information Systems* **27**(2), 75–91.

M. Rosemann, P. Green and M. Indulska, (2004), *A Reference Methodology for Conducting Ontological Analyses*, ER 2004, 110–121.

E.A. Rundensteiner, L. Bic, J.P. Gilbert and M.-L. Yin, (April, 1994). Set restrictions tk)r semantic groupings, *IEEE Trans. on Knowledge and Data Engineering* **6**(2), 193–204.

J.R. Searle, (1979), *A taxonomy of Illocutionary Speech Acts*, in Expressions and Meaning: Studies in the Theory of Speech Acts, Cambridge Press, Binghampton, New York, pp. 1–29.

G. Shanks, E. Tansley, J. Nurelini, D. Toblin and R. Weber, (15–18 December, 2002). *Representing Part-Whole Relationships in Conceptual Modeling: An Empirical Evaluation*, International Conference on Information Systems, Barcelona, Spain.

M. Shaw, (1984), The impact of modelling and abstraction concerns on modern programming languages, in: *On Conceptual Modelling*, M.L. Brodie, J. Mylopoulos and J.W. Schmidt, eds, Springer-Verlag, pp. 19–47.

K. Siau, (1996). *Empirical studies in information modeling: Interpretation of the object relationship*, Unpublished dissertation, University of British Columbia, Canada.

K. Siau, (July-Sept, 2004), Relationship Construct in Modeling Information Systems. Editorial Preface, *Journal of Database Management* **15**(3), i–v.

K. Siau, Y. Wand and I. Benbasat, (1997), The Relative Importance of Structural Constraints and Surface Semantics in Information Modeling, *Information Systems* **22**(23), 155–170.

J. Smith and D. Smith, (1977). Database Abstractions: Aggregation and Generalization, *ACM Transactions on Database Systems* **2**(2), 105–133.

V.C. Storey, (2005). Classifying and Comparing Relationships in Conceptual Modeling, *IEEE Transactions on Knowledge and Data Engineering* (forthcoming 2005).

V.C. Storey, D. Dey, H. Ullrich and S. Sundaresan, (1998). An Ontology-Based Expert System for Database Design, *Data and Knowledge Engineering* **28**(1), 31–46.

C. Theodoulidis, B. Wangler and P. Loucopoulos, (1992). The entity-relationship-time model, in: *Conceptual Modeling, Databases, and Case: An Integrated View of Information Systems Development, Wiley,* P. Loucopoulos and R. Zicari, eds, New York, pp. 87–115.

H. Ullrich, S. Purao and V.C. Storey, (28–30 June, 2000), *An Ontology for Classifying the Semantics of Relationships in Database Design*, Proceeding of the 5th International Conference on Applications of Natural Language to Information Systems, (NLDB '00), Versailles, France.

Y. Wand, V.C. Storey and R. Weber, (December, 1999). Analyzing the Meaning of a Relationship, *ACM Transactions on Database Systems* **24**(4), 494–528.

Y. Wand and R. Weber, (1995). On the Deep Structure of Information Systems, *European Journal of Information Systems* **5**, 203–223.

R. Weber, (1997). *Ontological Foundations of Information Systems*, Coopers and Lybrand, Melbourne, Australia.

R. Weber and Y. Zhang, (April, 1996). An Ontological Evaluation of NIAM's Grammar for Conceptual Schema Diagrams, *European. Journal of Information Systems* **6**(2), 147–170.

R. Weber, (June, 1996). Are Attributes Entities? A Study of Database Designers' Memory Structures, *Information Systems Research* **7**(2), 137–162.

R. Weber, (2002). Ontological Issues, in: *Accounting Information Systems*, S. Sutton and V. Arnold, eds, Researching Accounting as an Information Systems Discipline Sarasota, FL: American Accounting Association.