OOIS'94

1994 International Conference on
Object Oriented Information Systems
19–21 December 1994, London

Proceedings

Edited by

D. Patel, Y. Sun and S. Patel

South Bank University, London, UK

Presented by

SOUTH BANK UNIVERSITY
·LONDON·

Sponsored by

DATA ACCESS
U K L I M I T E D

IBM

Springer-Verlag
London Berlin Heidelberg New York
Paris Tokyo Hong Kong
Barcelona Budapest

The OODB Ownership Relationship

Oscar (Ou) Yang†, Michael Halper‡, James Geller†, and Yehoshua Perl†

†CIS Department and CMS
New Jersey Institute of Technology
Newark, NJ 07102 USA
e-mail: oscar@earth.njit.edu, {geller, perl}@vienna.njit.edu
fax: (201) 596-5777

‡Dept. of Math & Computer Science
Kean College of New Jersey
Union, NJ 07083 USA
e-mail: viking@ earth.njit.edu

Abstract. *The notion of "ownership" is prevalent in many social, economic, and political activities. In this paper, we show that an "ownership" semantic relationship can be a powerful addition to the repertoire of object-oriented database (OODB) modeling primitives. The motivation for such a relationship is provided by example OODB schemata. Using these schemata and some (informal) legal definitions, the varied semantics of ownership are identified. A formal ownership relationship is presented as an extension to an OODB data model.*

KEY WORDS: *Semantic Relationships, Ownership Relationship, Semantic Modeling, Commercial Applications, Integrity Constraints*

1. Introduction

Object-Oriented Database Systems have become powerful tools in many advanced application domains due to their data modeling capabilities (Kim et al., 1989; MacKellar et al., 1992; Woelk et al., 1986). While sharing with semantic data models some common notions such as classes and inheritance among classes (Hammer et al., 1978), OODBs possess additional properties like encapsulation and late binding. Their utility is derived in no small part from the semantic relationships — e.g., IS-A (SUBCLASS) (Brachman, 1983; Snyder, 1986) and PART-OF (Halper et al., 1992; Halper, 1993; Kim et al., 1989b; Nguyen et al., 1991) — used in the construction of their schemata. By "semantic relationship" we mean a connection between classes whose interpretation does not lie solely "in its name" (Woods, 1975) but in its constraint-satisfaction, inheritance, and operational mechanisms. Our data model not only captures important static constraints, as most semantic data models do, but also includes as part of the schema many dynamic constraints, i.e., constraints on the state transition of the database (Gray et al., 1992). In other words, such a relationship is a built-in modeling primitive with rich semantics. In this paper, we demonstrate the usefulness of including an "ownership" semantic relationship in an OODB data model. The motivation for this revolves around the following issues:

1. The richness and complexity of "ownership" in real-world applications. There are many different kinds of ownership encompassing a variety of semantics with respect to owner, possession, and their connection.
2. The frequent use of ownership in everyday life and the corporate world.
3. The hierarchical nature of ownership in the corporate world, where, for example, a company can own other companies. This may be represented by one reflexive ownership relationship at the schema level.

For our motivation of an ownership semantic relationship, we shall employ two example schemata. The first is abstracted from the following scenario. Joe owns a manufacturing business that produces an item for which he holds a patent. The business resides in a building which Joe owns and for which a bank, First National Trust, holds a lien. Joe and his wife Jackie rent their house from Tom. Jackie is a professional writer and owns the copyrights for two books. She owns a car, and Joe uses another which is legally owned by his business. Together, they own another car, used by their son John, and a joint bank account. Their individual investment portfolios consist of corporate stocks and government bonds. In addition, each possesses a life insurance policy. Both jointly own the appliances in their home.

The second schema is abstracted from the following. General Motors (GM) owns Chevrolet which in turn owns subsidiaries, manufacturing plants, industrial equipment, etc. GM and Toyota jointly own the Geo Corporation. GM also is a public company and issues stock which is owned by shareholders who are persons or other corporations.

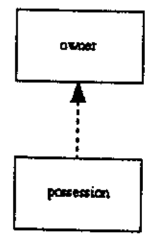

Figure 1. The generic ownership relationship.

We graphically represent "ownership" in an OODB schema (Halper et al., 1993b) by a bold, dotted arrow (Fig. 1). The two above scenarios and their corresponding schemata are shown in Fig. 2 and 3, respectively. As we will discuss, the various ownership relationships which appear in these schemata exhibit a wide range of distinctions. In the next section, we will categorize these different sorts of ownership.

The remainder of this paper is organized as follows. In Section 2, we discuss the legal definition of and terminology relating to ownership. In Section 3, we formally define the ownership relationship as a quintuple comprising a number of "characteristic" dimensions. Section 4 contains concluding remarks.

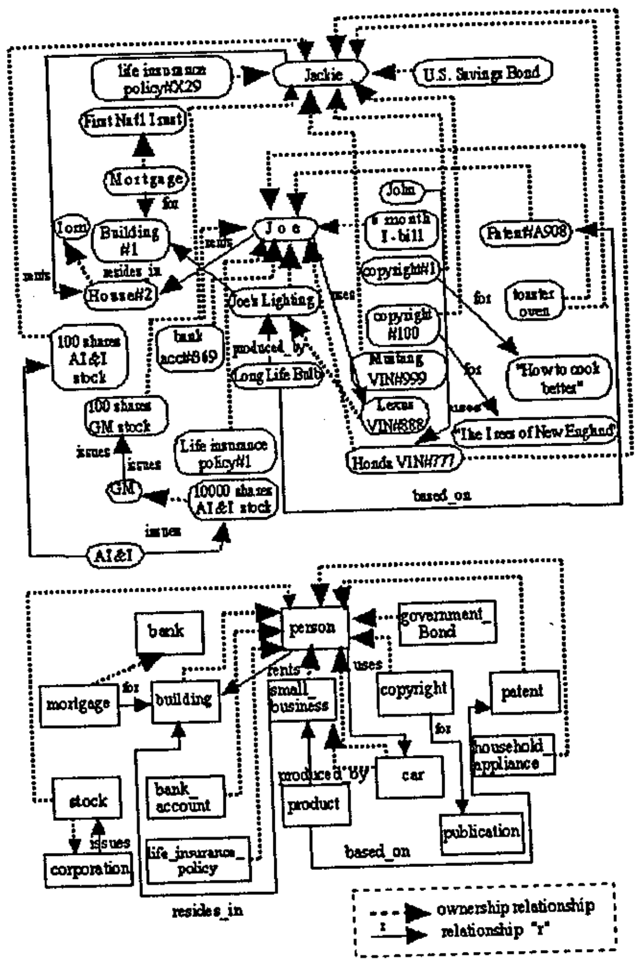

Figure 2. Instance and schema diagrams for first scenario.

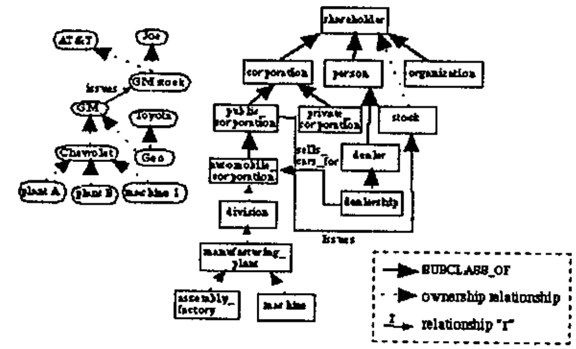

Figure 3: Instance and schema diagrams for second scenario.

2. The Definition of Ownership

When we describe a state of "ownership", we must in general include the following three features: (1) The owner, (2) the possession that is owned, and (3) the characteristics of the relationship between the two (Fig. 1). We are interested in identifying what types of objects can fi2.ll the roles of (1) and (2), and what the characteristics that distinguish the various kinds of ownership are.

According to *Webster's Dictionary*, ownership is defined as follows:

1. The state or fact of being an owner.
2. Proprietorship; Legal right of possession; Legal or just claim or title (to something); in law, the right to use for one's own advantage some possession.[1]

The owner referred to above can, by law, be a natural person, a corporation, or an organization. The latter two are, in general, referred to as *legal entities*. Under the law, legal entities are vested with certain powers, some of which are also held by natural persons. Others, like the power to exist in perpetuity, are unique to legal entities. In our databases, we see that Joe as a natural person owns his business. The GM Corporation as a legal entity owns Chevrolet. In Fig. 2, the following classes can be characterized as legal entities in ownership relationships: *bank, small_business,* and *corporation.* All "owner" classes in Fig. 3 represent legal entities.

Ownership of an item is often distributed among persons and legal entities. For example, Joe and Jackie together own a toaster oven, a bank account, and their son's car. Also, the Geo Corporation is a joint venture of GM and Toyota. We describe such a situation as *joint ownership*. It is perfectly legitimate to have a person and a company jointly own the same thing. In Fig. 2, Jackie and GM hold stock in AT&T and are thus among the group of its joint owners. The ownership need not be divided into equal portions, either. Stock holdings partition the ownership of a public company into various percentages. Joe, for example, owns one thousand shares of GM, making him a small percentage owner.

[1] The technical legal term for possession is *property*; however, due to the widespread use of that term in the database field, we will avoid it here.

In law, *possession* means the rights which one has in anything subject to ownership, whether it be mobile or immobile, tangible or intangible, visible or invisible. Ownership is used synonymously with rights in possession. Thus, a person is said to be the owner of a possession if he has certain rights in it. The term ownership is often used to indicate that one has the "highest rights" (Anderson et al., 1971) in a possession, but it may be used even when one does not have a the rights; thus, we say that a person is an owner of a house even though he has rented it to a tenant who has exclusive rights to the use of the house during the term of the lease (Anderson et al., 1971).

A possession can be classified as *real*, *intellectual*, or *personal*. A real *possession* refers to the rights that one has in land or things closely related to it. An *intellectual possession* is the rights held on an idea (e.g., the design of an invention) or a creative work (such as a musical composition or a novel). For such possessions, the rights apply to a potentiality-no claim is made on any tangible item. Copyrights and patents are the ordinary forms of intellectual possessions. *Personal possessions* encompass everything that is not a real or intellectual possession.

As examples, Joe's business resides in a building which is his real possession. The patent (number A908) for the Long_Life Bulb is his intellectual property. Bank account 369 and a toaster oven are among his and Jackie's personal possessions. In Fig. 2, the class building denotes a real possession. *Copyright* and *patent* are intellectual possessions. The remainder of the "possession" classes represent personal possessions. In Fig. 3, the only real possession is *manufacturing_plant*. The rest are personal possessions.

The major distinguishing characteristic of the ownership relationship itself centers around the existence of a legal document that verifies the owner's rights to a possession. A copyright owner, e.g., is granted a legal certificate giving him exclusive rights to possess, make, publish, and sell copies of his intellectual production, or to authorize others to do so. In contrast, the owner of a household item does not have a legal document to support his ownership, but he has the right to use it as he pleases.[2] We call ownership of the former kind *de jure* and ownership of the latter kind de facto. So, Jackie's copyrights are owned *de jure*, while her toaster oven is owned *de facto*.

In Fig. 2, the following ownership (written as: owner class - possession class) are among those that can be classified as *de jure*: *bank - mortgage*, *person - building*, *person - small_business*, *person - bank_account*, *small_business - car*. The relationship between the classes *person* and *household_appliance* is *de facto*. All ownerships in Fig. 3 are *de jure*.

3. Ownership as an OODB Semantic Relationship

To incorporate ownership into an OODB data model, we need to provide a formal description for it. As discussed earlier, there are different kinds of ownership. Our investigation has revealed characteristics which describe the required functionality to support the different kinds of ownership. Such characteristics are called dimensions, and we will give formal definitions for each dimension of the ownership relationship. We first introduce the definition of a generic ownership relationship, from which all others will be derived. Let $E(C)$ denote the extension of a class C, i.e., the set of all instances of C. The generic ownership relationship between a possession class B and an owner class A (Figure 1) is a relation Ω_B^A, from $E(B)$ to $E(A)$. The pair $(b, a) \in \Omega_B^A$ indicates that the instance b of class B is the possession of (i.e., is owned by) the instance a of class A. We will ordinarily express this fact as $b\Omega_B^A a$.

[2] Yes, he may have kept the sales receipt, but technically that documents the purchase transaction, not the ownership.

3.1 The Formal Definition of the Ownership Relationship

To describe the characteristic dimensions of ownership, we employ the following quintuple.

$$O_{B,A} = \langle \Omega_B^A, \lambda, \chi, \delta, \nu \rangle \quad (1)$$

Ω_B^A is defined as above. The remainder of the quintuple represents the values of four characteristic dimensions. For each dimension we list its name and domain as follows:

$$\begin{aligned}
&\textit{Exclusiveness: } \chi \in X = \{\textit{exclusive, free-joint, percentage-joint}\},\\
&\textit{Dependency: } \delta \in D = \{\textit{owner-to-possession, nil}\},\\
&\textit{Legality: } \lambda \in L = \{\textit{de jure, de facto}\},\\
&\textit{Value Propagation: } \nu \in V = \{\textit{up, down, upTrans, downTrans, up\&down, nil}\}.
\end{aligned} \quad (2)$$

Formal descriptions of each dimension will be given in subsequent sections. For this, we need the following definitions. Assume that there exists an ownership relationship $O_{B,A}$.

Definition 1: $\forall a \in E(A)$, let $Po_B^A(a) = \{b \mid b \in E(B) \wedge b\Omega_B^A a\}$. $Po_B^A(a)$ is called the *possession* set of a with respect to $O_{B,A}$, i.e., the set of instances of B which are possessions of a.

Definition 2: $\forall b \in E(B)$, let $No_B^A(b) = \{a \mid a \in E(A) \wedge b\Omega_B^A a\}$. $No_B^A(b)$ is called *the owner* set of b with respect to the ownership $O_{B,A}$, i.e., the set of instances of A of which b is a possession.

3.2 The Exclusiveness Dimension

Ownership relationships, in general, can be divided along the lines of exclusive and joint. In other words, a possession may be owned by only one owner or jointly owned by several owners. As a basic characteristic of ownership in the real world (Anderson et al., 1971; Moore, 1968), exclusiveness represents an intuitive constraint which may be imposed on objects in an ownership relationship. The formal definition for the exclusive ownership relationship follows:

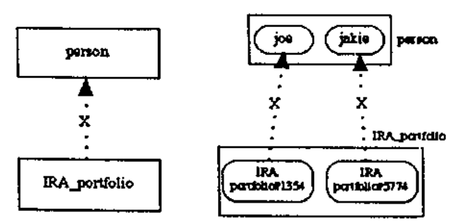

Figure 4: An example of exclusive ownership.

Definition 3: The ownership relationship $O_{B,A}$ is exclusive (i.e., χ = *exclusive*) if $\forall b \in E(B), |N_{\Omega_B^A}(b)| \le 1$. That is, an exclusive possession cannot have more than one owner.

We add in our graphical notation an **X** on the dotted arrow to indicate eXclusive (Fig. 4). Those ownership relationships which are not exclusive are referred to as *joint*, in which case a possession may be either jointly owned freely (e.g., a joint bank account is freely shared by a couple — we call this *free joint*), or jointly owned such that each owner takes a certain percentage of the ownership (e.g., the husband and the wife each owns 50% of their house — we call this percentage joint). We call the case where all owners have the same percentage *equal joint*. Although the exclusiveness dimension has been included in some OODB models (e.g., SHOOD (Nguyen et al., 1991) and OODINI (Halper et al., 1992)), percentage joint is unique to ownership. Percentage joint plays an important role in economic activities. A shareholder has the right to receive his proportion (i.e., percentage) of dividends; or to call a special meeting of the shareholders, if he owns the shares in a stated percentage.

In our graphical notation, a plain dotted arrow indicates free joint (Fig. 5). Percentage joint and equal joint are denoted by labels of **P** and =, respectively (Figs. 6, 7).

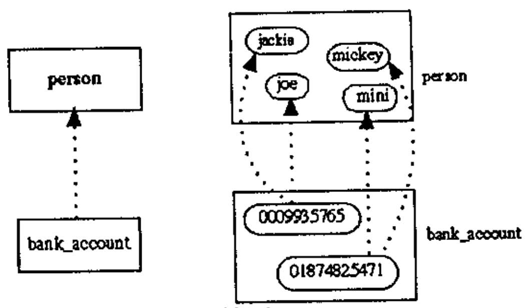

Figure 5. Jointly owned bank accounts.

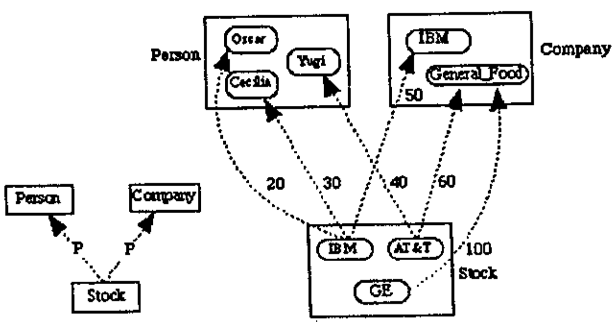

Figure 6. Stocks are owned (percentage) jointly by person and company.

Definition 4: The ownership relationship $O_{B,A}$ is *free joint* (i.e., χ = *free-joint*) if $\forall b \in E(B), |N_{\Omega_B^A}(b)| \ge 0$, i.e., there is no constraint on the owner set of b.

Definition 5a: The ownership relationship $O_{B,A}$ is *percentage joint* if there exists a total function (called the individual share function) $f : E(B) \times E(A) \to [0, 100]$ such that $\forall b, \sum_{a \in N_{\Omega_B^A}(b)} f(b,a) = 100$.

Definition 5a defines the *percentage joint* ownership relationship when the possession class has only one owner class. At times, the ownership may be distributed among owners from different classes. This "multiple ownership" case is defined as follows.

Definition 5b: The ownership relationships $O_{B,A_1}, O_{B,A_2}, \ldots, O_{B,A_n}$ (n is an integer) are percentage joint if there exists a total function $f: E(B) \times \bigcup_{i=1}^{n} E(A_i) \longrightarrow [0,100]$ such that $\forall b, \sum_{a \in M} f(b,a) = 100$, where $M = \bigcup_{i=1}^{n} N_{\Omega_B^{A_i}}(b)$.

To better understand Definition 5b, refer to Figure 6 , where $O_{Stock,Person}$ and $O_{Stock,Company}$ are two percentage joint relationships. For any instance of class Stock, the ownership is distributed among its owners such that each of them takes a certain percentage and that the sum of the percentages is 100 (percent). In Figure 6, the stock IBM's owners are Oscar and Cecilia of class *Person*, and IBM of class *Company*, with 20, 30, and 50 percent of the ownership, respectively.

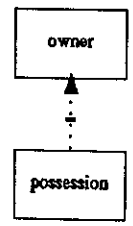

Figure 7: The equal ownership relationship.

3.3 The Dependency Dimension

The dependency dimension of ownership relationship $O_{B,A}$ regulates the semantics of deletion of owner class A or possession class B. It defines when deletion of one should cause deletion of the other. Intuitively, there seems to be no need of a dependency dimension for ownership; neither the owner nor the possession ceases to exist if the other is deleted from the database.

Nevertheless, after looking at some examples, we find that we need such a dimension. Consider the relationship between a person and his car. Neither is dependent on the other. However, suppose that in the application domain, say, of an insurance company, we need to distinguish people who own cars from people who do not. Car owners have some specific properties which are modeled by creating a specialization class *car owner* of the class *person*. The class car owner inherits properties from person and have extra properties. The ownership relationship for the car refers to the car owner class rather than to the person. This way of modeling in OODBs is described in (Kifer et al., 1992) as shifting information from the data to the schema. In this way, if we model the class car owner as a subclass of *person*, then the car owner is dependent on the car. For example, if a car owner *Lisa* owns only one car, and if it is "deleted" (e.g., destroyed in an accident), *Lisa* should be deleted from the class car owner — since she is no longer a car owner — but not from the class person. Now, if the query "list car owners" is executed, *Lisa* will not be returned (Fig. 8). In general, we may have such a dependency for specialization classes which are actually categorized according to the possession.

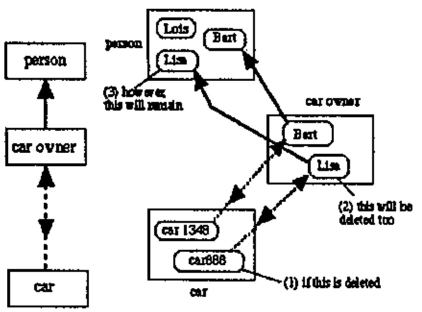

Figure 8: Car owner depends on car.

On the other hand, the possession-to-owner dependency is not justified by our analysis, though there are some examples that seem to require. Consider a corporation that owns several subsidiaries. Usually, if the corporation goes bankrupt, all its subsidiaries will also go out of business. This looks like a case of possession-to-owner dependency, but a closer look reveals that besides the ownership relationship, there is also PART-OF relationship between subsidiary and corporation, i.e., the subsidiaries are not only possessions, but also parts of the corporation. Furthermore, the ownership relationship itself does not cause any possession-to-owner dependency; it is the PART-OF relationship (Halper et al., 1993), that causes this dependency. Therefore, the possession-to-owner dependency does not appear in the dependency dimension of the ownership relationship. This dimension is, therefore, specified by the following set of values:

$\delta \in D =$ {*owner-to-possession, nil*}

The second value indicates that the ownership relationship lacks any dependency semantics. This is desirable in most cases (e.g., a person is not deleted if he sells his car).

In the following, we use the notation *del*(*x*) to denote the application of a method to delete the instance *x*, and use "x => y" to indicate that action *x* implies action *y*.

Definition 6: The ownership relationship $O_{B,A}$ is *owner-to-possession dependent* (i.e., δ = *owner-to-possession*) if $\forall b \in E(B), del(b) \Rightarrow \forall a \in E(A)$ such that $b\Omega_B^A a \wedge P\Omega_B^A(a) = \{b\}, del(a)$.

We draw an extra arrow head pointing to the possession on the dotted line to indicate this dependency (Fig. 8).

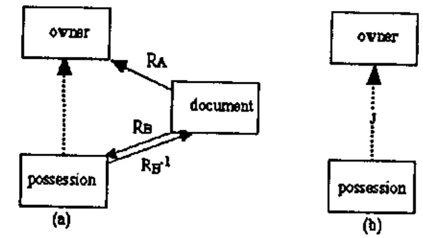

Figure 9: Ownership *de jure*: (a) document relates to both owner and possession (b) the graphical notation

3.4 The Legality Dimension

De jure ownership always has a supporting legal document, while *de facto* ownership does not. To represent *de jure* ownership, we need an extra class document which relates to both owner and possession. We label the relationship between *document* and *owner* as R_A, between *document* and *possession* as R_B. We assume a one-to-one correspondence between the instances document and possession, and define the inverse relationship R_B^{-1} for R_B (Fig. 9(a)).[3] The inverse relationship is necessary to maintain information integrity endowed with ownership *de jure*. For example, if we are given $b\Omega_B^A a$, and wish to find the document that supports it, we need to go from class *possession* to class *document*. Without R_B^{-1}, we would not be able to do so.

To be consistent with the notations for other dimensions and to simplify the graphical representation of the schema, we will denote ownership *de jure* with a **J** on the dotted arrow (Fig. 9(b)). That is, we omit the class *document* and its relationships from the graphical schema, assuming their existence.

In the following, *A* and *B* are the owner class and possession class respectively, *C* is the documentclass, and R_A, R_B and R_B^{-1} are defined as above. In addition, *del*(*x*) is defined as the application of a method to delete the instance *x*, *break*(*x*, *y*) (*con*(*x*, *y*)) to break (establish) the connection between the instances *x* and *y*, and *add*(*x*, *Y*) to add the instance *x* to the class *Y*.

Definition 7: The ownership relationship $O_{B,A} = <\Omega_B^A$, *de jure*, $\chi, \delta, \nu>$, where Ω_B^A, χ, δ and ν are defined as in (1) and (2) above, is called an ownership *de jure*, if $\forall a, b$ such that $a \in E(A), b \in E(B)$, and $b\Omega_B^A a$ holds, there exists $c \in E(C)$ such that $(c, a) \in R_A, (c, b) \in R_B$ (i.e., there exists a legal document verifying the ownership $b\Omega_B^A a$), and if it satisfies the following modification conditions:

(1) **delete owner:** $\forall a \in E(A), del(a) \Rightarrow \forall b \in P\Omega_B^A(a), break(b, a) \wedge \forall c \in E(C)$ such that $(b, c) \in R_B^{-1}, break(b, c), del(c)$.

[3]Note that R_A is not one-to-one as an owner may possess several instances of the same kind of possession. Thus it is not so simple to access the document through the owner class.

(2) delete possession: $\forall b \in E(B), del(b) \Rightarrow \forall c \in E(C)$ such that $(b, c) \in R_B^{-1}$, $break(b, c), del(c)$.

(3) delete ownership relationship: $break(b, a)$ such that $b\Omega_B^A a \Rightarrow \forall c$ such that $(b, c) \in R_B^{-1}$, $break(b, c)$, $del(c)$.

(4) establish ownership relationship: $con(b, a)$ such that $b\Omega_B^A a \Rightarrow add(c, C), con(c, a), con(c, b), con(b, c)$.

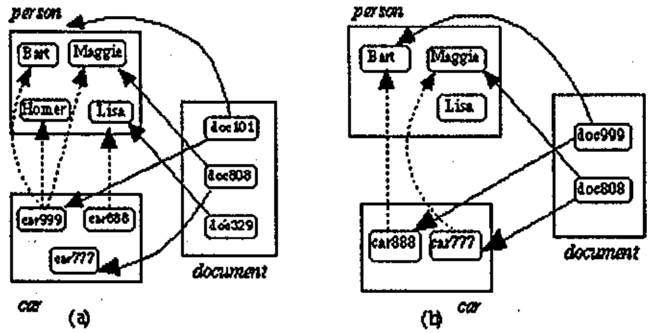

Figure 10: The ownership *de jure*. (a) before operations performed, (b) after operations performed.

The methods in Definition 7 are basic operations. Complex operations can be implemented by combining those methods. An example of such operations is ownership transfer, which is a common transaction in the commercial world. This can be characterized by "the ownership of b is transferred from a_1 to a_2". The method *transfer* (a_1, a_2, b) can be implemented by "*break*(b, a_1)" and "*con*(b, a_2)". In Figure 10, *Bart* lost his car *car999* in an accident and bought a second-hand car *car888* from *Lisa*. In the meantime, *Homer* died of old age, and left his car *car777* which was formally jointly owned with his wife *Maggie*. The changes in the database can be described by the following sequence of operations: *del(car999)*, *transfer(Lisa, Bart, car888)*, and *del(Homer)*. By forcing the database schema to adhere to those constraints, the integrity of the data model is guaranteed.

Definition 8: The ownership relationship $O_{B,A}$ is *de facto* if it is not *de jure*.

3.5 The Value Propagation Dimension

There are times when a certain feature of a possession is naturally assimilated as a feature of its owner, or vice versa. For example, the address of a person may be modeled as the address of his house rather than as an intrinsic attribute of the person. Likewise, the name that appears on the passport can be taken to be the name of its owner. In the former case, the value of *address*, rather than being duplicated, should be stored solely with the house and propagated upward on demand (Figure 11). *Address*, in this sense, is a *derived attribute* of person. In our data model, the value propagation dimension may take on six different values:

V = {*up, down, upTrans, downTrans, up&down, nil*}

Due to space limitations, we will present only the definition for transformational upward propagation, which is expressed by $\delta = upTrans$. For the other cases, refer to (Halper, 1993; Halper et al., 1993), where we use similar definitions for part modeling.

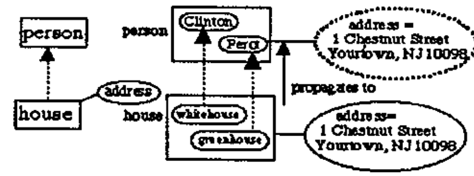

Figure 11: Address propagated from home to person

Definition 9: Let $\pi_B : E(B) \rightarrow \tau$ be a possession of B. The *transformational upward propagating ownership relationship which propagates* π_B is defined as $O_{B,A} = \langle \Omega_B^A, \lambda, \chi, \delta, (upTrans, D_{\pi B}, \{T^{(n)}\}, \nu) \rangle$. Here, τ is any data type, $\{T^{(n)}\}$ is a family of symmetric operators $T^{(n)}: \tau^n \rightarrow \tau$ with $n > 0$, and the function $D_{\pi B}: E(A) \rightarrow \tau$, called a derived attribute, is defined in terms of $\{T^{(n)}\}$ as follows. (Note that the possession set of an instance a of A is taken to be $P\Omega_B^A(a) = \{b_1, b_2, \ldots, b_m\}, m \geq 0$.)

$$D_{\pi_B}(a) = \begin{cases} T^{(m)}[\pi_B(b_1), \pi_B(b_2), \ldots, \pi_B(b_m)], & m \neq 0 \wedge \pi_B(b_i) \text{ is defined}, \\ C, & \text{otherwise.} \end{cases}$$

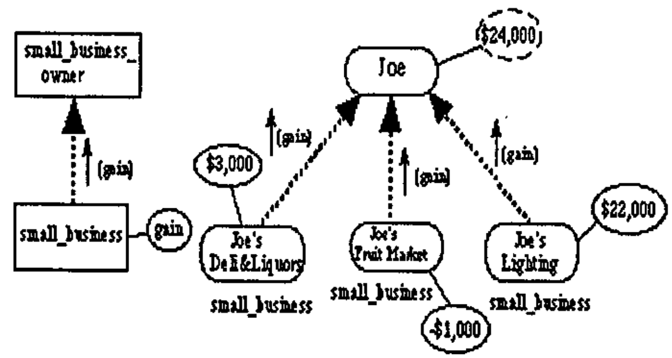

Figure 12: An example of the value propagation.

Here C is a pre-set default value. For example (Fig. 12), Mr. *Joe* owns three businesses, *Joe's Deli & Liquors*, *Joe's Fruit Market*, and *Joe's Lighting*. At the end of a year, he wants to know how much money he has made from his businesses. This can be done by having a transformational upward propagation. That is, the derived attribute D_{gain} of the class *small business owner* can be written in terms of the attribute *gain*: *E(small business)* $\rightarrow$ *REAL* of the class *small Business*, as follows:

$$D_{gain}(a) = \begin{cases} sum[gain(b_1), gain(b_2), \ldots, gain(b_n)], & n \neq 0 \wedge gain(b_i) \text{ is defined for } 1 \leq i \leq n, \\ 0, & \text{otherwise.} \end{cases}$$

4. Conclusion

We have addressed the issue of representing ownership relationships in OODBS, with a model that captures a variety of semantics. In particular, we have distinguished a number of aspects for the roles of the owner and possession in such relationships. These aspects define notions like exclusive and joint owners. Ownership relationships themselves were shown to be either *de jure* or *de facto*, the former being distinguished by the presence of a legal document. Formal definitions for the various ownership relationships were presented. To complement these, we have presented graphical symbols for each of the ownership relationships which expand the graphical schema representation language for OODBs developed in (Halper et al., 1993b). In future work, we will investigate the interaction between ownership transactions and the above ownership dimensions. We plan to integrate the ownership relationship as an integral part of a commercial OODB system supporting the necessary dimensions. Such an addition to OODB systems will increase their appeal for commercial applications.

References

(Anderson et al., 1971): R. Anderson, W. Kumpf, and R. Kendrick. Business Law - *Principles and cases*. South-Western Publishing Co., Cincinnati, OH, 1971.

(Brachman, 1983): Ronald J. Brachman. What IS-A is and isn't: An analysis of taxonomic links in semantic networks. *Computer*, 16(10):30-36, October 1983.

(Gray et al., 1992): P. Gray, K. Kalkarni and N. Paton. *Object-Oriented Databases: A Semantic Data Model Approach.* Prentice Hall, 1992.

(Halper, 1993): M. Halper. *A Comprehensive Part Model and Graphical Schema Representation for Object-Oriented Databases.* PhD thesis, CIS Department, New Jersey Institute of Technology, 1993.

(Halper et al., 1992): M. Halper, J. Geller, and Y. Perl. An OODB "part" relationship model. In *Proceedings of the First International Conference on Information and Knowledge Management*, 602-611, Baltimore, MD, 1992.

(Halper et al., 1993): M. Halper, J. Geller, and Y. Perl. Value propagation in object-oriented database part hierarchies. In *Proceedings of the 2nd Int'l. Conference on Information and Knowledge Management*, 606-614. Washington, DC, 1993.

(Halper et al., 1993b): M. Halper, J. Geller, Y. Perl, and E. J. Neuhold. A graphical schema representation forobject-oriented databases. In R. Cooper, editor, *Interfaces to Database Systems*, 282-307. Springer-Verlag, London, 1993.

(Hammer et al., 1978): M. Hammer and D. McLeod. The Semantic Data Model: a Modeling Mechanism for Database Applications. In Proc. of *ACM SIGMOD*, 1978.

(Kim et al., 1989): W. Kim, N. Ballou, H.-T. Chou, and J. F. Garza. Features of the orion object-orienteddatabase system. In W. Kim and F. H. Lochovsky, editors, *Object-Oriented Concepts, Databases, and Applications*. Addison Wesley, Reading, MA, 1989.

(Kim et al., 1989b): W. Kim, E. Bertino, and J. Garza. Composite objects revisited. In Proc. of the 1989 ACM SIGMOD International Conference on the Management of Data Portland, Oregon, appeared as SIGMOD RECORD, pages 337-347,1989.

(Kifer et al., 1992): M. Kifer, W. Kim, and Y. Sagiv. Querying object-oriented databases. In *Proceedings of the 1992 ACM SIGMOD Int'l Conf. on Management of Data*, 393-402, San Diego, California, June 1992.

(Moore, 1968): Clarence C. Moore. *Business Law*. The Bruce Publishing Company, USA, 1968.

(MacKellar et al., 1992): B. MacKellar and J. Peckham. Representing design objects in sorac: A data model with semantic objects, relationships and constraints. In *Second International Conference on Artificial Intelligence in Design*, Pittsburgh, PA, June 1992.

(Nguyen et al., 1991): G. T. Nguyen and D. Rieu. Representing design objects. In *AI in Design'91*. Butterworth-Heinemann Ltd., 1991.

(Snyder, 1986): Alan Snyder. Encapsulation and inheritance in object-oriented programming languages. In *Proc. OOPSLA-86*, 38-45, 1986.

(Woelk et al., 1986): D. Woelk, W. Kim, and W. Luther. An object-oriented approach to multimedia databases. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, 311-325, Washington, D.C., May 1986.

(Woods, 1975): W. A. Woods. What's in a link: Foundations for semantic networks. In D. G. Bobrow and A. M. Collins, editors, *Representation and Understanding*, 35-82. Academic Press, New York, NY, 1975.