UC Irvine
ICS Technical Reports

Title
Set-related restrictions for semantic groupings

Permalink
https://escholarship.org/uc/item/2qr082fn

Authors
Rundensteiner, Elke
Bic, Lubomir
Gilbert, Jonathan
et al.

Publication Date
1989

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

ARCHIVES
Z
699
C3
no.89-07
c.2

Set-Related Restrictions for Semantic Groupings

Elke Rundensteiner

Lubomir Bic

Jonathan Gilbert

Meng-Lai Yin

Technical Report 89-07

Notice: This Material
may be protected
by Copyright Law
(Title 17 U.S.C.)

Set-Related Restrictions for Semantic Groupings

Elke Rundensteiner
Lubomir Bic
Jonathan Gilbert
Meng-Lai Yin

Information and Computer Science Department
University of California, Irvine
January, 1989

Abstract

Semantic database models utilize several fundamental forms of groupings to increase their expressive power. In this paper we consider four of the most common of these constructs; basic set groupings, is-a related groupings, power set groupings, and Cartesian aggregation groupings. For each, we define a number of useful restrictions that control its structure and composition. This permits each grouping to capture more subtle distinctions of the concepts or situations in the application environment. The resulting set of restrictions forms a framework which increases the expressive power of semantic models and specifies various set-related integrity constraints.

1. Introduction

In the past decade research into the design and implementation of semantic database models has risen to prominence. Semantic models attempt to cope with the demands of new sophisticated database applications by modeling their environment more directly than traditional database models. This trend has been clearly reflected in the literature from the development of the Entity-Relationship model [CHEN76] and the Hierarchical Relational model [SMITH77], to the advent of the first *semantic database models* [CODD79, HAMMER81] and the evolution continues today (see, for example, [BIC86, ABITEBOUL87, HULL87A]). Important features common to most of these models are a property inheritance hierarchy (is-a relationship), a decomposition hierarchy (is-part-of relationship), and a variety of powerful constructs for collecting related data, referred to as *semantic groupings.*

The latter allow new sets of database entities to be constructed from the stored data. The most common groupings are based on extended definitions of sets, power sets, and the Cartesian product.

In this paper, we present several basic types of semantic groupings and a number of restrictions that can be imposed on each of them. A semantic grouping can be specialized to suit the particular concept to be modeled by selecting the appropriate restrictions. We only consider set-related restrictions, i.e., restrictions imposed on the structure of the groupings or their population. Other types of general constraints, such as attribute- or value-based ones, are not considered here. The purpose of developing such a model is twofold: First, it increases the expressiveness of the paradigm and second, it serves as a basis for additional integrity constraints [HAMMER76, LENZERINI87].

The paper is organized as follows. In Section 2 we define the several types of semantic groupings. First an extended notion of a set (called *set grouping*) is defined by allowing elements to repeat and an order to be imposed on its elements. Then the *is-a related set groupings*, *the power set grouping* and the *Cartesian grouping* are defined based on the set grouping definition. The concepts of *identity* and *indistinguishability* are also discussed in the context of these groupings. Sections 3, 4, 5 and 6 then present the restrictions on the four groupings, respectively, each illustrated by a set of examples. Conclusions and some open problems are presented in the final section.

2. Basic Concepts

2.1. Classes, Types, and Sets

In semantic data modeling, the notions of set, class, and type are not always clearly distinguished. In this paper we take the following position. Entities in our data model represent a concept (concrete or abstract) in the application world.

The term entity, the way it is used in this paper, should be interpreted in its most generic form — it may, for example, stand for a row in a relation table [CODD79], an entity (or a relationship) in the entity-relationship model [CHEN76], an entity in a semantic data model [KING84, HULL87A] or an object in an object-oriented model [MYLOP80, HUDSON86]. Entities in the world of the application are grouped into *classes*, based on common properties. The class notion, however, serves a dual purpose. It is a generic description of all entities which belong to that class and hence it imposes a *type* on those entities. At the same time a class represents the *set* of entities which conform to its generic description (type). This distinction is important because certain attributes only apply to a set as a whole (e.g., cardinality) while others are applicable to its individual elements (e.g., color). In this paper we are interested primarily in the set-aspect of a class and, therefore, we will be referring to sets rather than classes.

2.2. Set Groupings

A mathematical set is a collection of elements (entities) taken from a given domain of discourse. The domain for each set depends on the particular application being modeled. We will extend the notion of a set as follows:

(1) elements may have their own *identity*,

(2) multiple *non-distinguishable* copies of an element may occur, and

(3) an order may be imposed on the elements.

Each of these extensions will be described in the following subsections. Note that groupings which inhere to these three extensions are called set groupings.

We distinguish between *simple, base* and *constructed* entities. Both base and simple entities are taken directly from a base domain of the application while constructed entities are built from other entities using Cartesian or power set aggregation abstractions; both of which will be discussed at the end of this section. Note that a simple entity is a special case of a base entity which represents a value

(for example an integer or a string) in the database. Base entities in general may correspond to complex objects in the real world and constructed entities are always complex. We define a *base-set grouping* to be a collection of base entities. All other *set groupings* are either based on these base-set groupings or on other set groupings, each of which may be constructed in the three ways described below.

The first type of extension is concerned with *is-a* related set groupings, in particular, *subset* and *union* groupings. Two sets A and B are said to be *is-a* related if every element from A is described by all the properties which elements in B have. For example, the sets Red Cars and Cars are *is-a* related, since every red car has the properties of a car. This is generally referred to as property inheritance. These new set groupings are constructed by collecting some elements of existing set groupings without, however, changing the shape or identity of these elements. A *subset grouping* is based on the mathematical definition of a subset S which consists of some elements of an existing set. We employ a more general subset definition where a subset grouping S is based on one *or more* existing set groupings $S_1, S_2, ..., S_m$. Each element s of the subset grouping S must exist in all these underlying set groupings S_i for i = 1 to m. More precisely, $S \subseteq (S_1, S_2, ..., S_m)$ if and only if $((\forall i, 1 \leq i \leq m)(s \in S \Rightarrow s \in S_i))$. Note that the above definition of a subset is different from a set intersection. The latter corresponds to the *largest* possible subset, i.e., a special case of a subset. This definition is necessary to maintain the *is-a* relationship between the subset S and all set groupings it is based on. The following implication holds, $S \subseteq (S_1, S_2, ..., S_m)$ implies $((\forall i, 1 \leq i \leq m)(S$ *is-a* $S_i))$.

The *union grouping* is an abstraction which forms a possibly heterogeneous set grouping S from several existing set groupings $S_1, S_2, ..., S_m$. The union grouping is denoted by $S = \biguplus_{i=1}^{m} S_i$. This construct collects the elements from all involved set groupings into one new grouping. The result of a *union grouping* is again a set

grouping. More precisely, S = $\biguplus_{i=1}^{m} S_i$ if and only if (($\forall$ i, $1 \leq i \leq m$) (s $\in$ S_i $\Rightarrow$ s $\in$ S)) and (s $\in$ S $\Rightarrow$ (($\exists$ i, $1 \leq i \leq m$) s $\in$ S_i)). In other words, if and only if an entity s belongs to any of the underlying set groupings S_i, then s must also belong to the union S. Again, the *is-a* relationship is guaranteed, namely, S = $\biguplus_{i=1}^{m} S_i$ implies (($\forall$ i, $1 \leq i \leq m$) (S_i *is-a* S)).

The domain of a subset grouping is the intersection of the domains of all underlying set groupings. The domain of the union grouping is the union of the domains of all underlying set groupings.

A *Cartesian aggregation grouping* ([SMITH77]) is an abstraction which allows a relationship between several database entities to be viewed as a single aggregate or *complex* entity. For example, a relationship between a person, a hotel room and a date can be viewed as a reservation. Each element in the Cartesian grouping is taken from the *cross product* of existing set groupings and a new unique identity is associated with it. A Cartesian grouping C based on the set groupings S_1, S_2, ..., S_n is defined by C $\subseteq$ S_1 $\times$ S_2 $\times$ ...$\times$ S_n. We say that set S_1 fills *position* 1 in the grouping, set S_2 fills *position* 2, etc. The ordering of positions is not essential to the model, it is just a matter of notational convenience. In fact, the positions p_i are usually referred to by labels unique to C which represent the role that the set grouping plays in the Cartesian grouping. If L_i is the label chosen for the position p_i, respectively, then we denote C $\subseteq$ (L_1 : S_1) $\times$ (L_2 : S_2) $\times$ ... $\times$ (L_n : S_n). The previous example would be denoted by reservation $\subseteq$ (who: person) $\times$ (where: hotel-room) $\times$ (when: date).

The notion of a domain for a Cartesian grouping is defined as follows. Let D_i be the domain of S_i for all i. Then, the domain D of C corresponds to the cross products of the domains of the set groupings underlying C, i.e., D = D_1 $\times$ D_2 $\times$ ... $\times$ D_n.

An element t of a Cartesian grouping C is an aggregate entity consisting of n components where the i^{th} component is taken from the set grouping underlying the i^{th} position, S_i. We refer to the i^{th} component of t by $t[L_i]$ where L_i is the label of the i^{th} position, or simply, t[i]. Of course, $t[L_i] \in S_i$. To refer to several positions of an aggregate entity at the same time, we use the notation $t[L_{i1}, L_{i2}, ..., L_{ik}]$ where $L_{i1}, L_{i2}, ..., L_{ik}$ are distinct labels associated with *some* of the set groupings S_i. This is called an aggregate projection. Similarly, $C[L_{i1}, L_{i2}, ..., L_{ik}]$ refers to the collection of all aggregate projections of the Cartesian grouping C using labels $L_{i1}, L_{i2}, ..., L_{ik}$. More formally, $C[L_{i1}, L_{i2}, ..., L_{ik}] := \{ t[L_{i1}, L_{i2}, ..., L_{ik}] \mid t \in C \}$ where $t[L_{i1}, L_{i2}, ..., L_{ik}]$ retains the identity of the aggregate entity t. The latter generalizes the notion of a projection in the relational model.

A *power set grouping* is based on the mathematical concept of a power set 2^S of a set S which consists of all subsets of S. A power set grouping, denoted by G^*, is an abstraction based on a set grouping G. Each element of G^*, which corresponds to a subset grouping of G, has its own unique identity. Hence G^* consists of some (or all) of the possible subsets of G. In general, an element of G^* models a single (complex) entity, usually referred to as a *cover aggregate*. It should be emphasized that the power set abstraction forms a set of aggregates each of which is formed from (contains) a set of elements from G. For example, clubs are cover aggregates, each composed of a set of people. We distinguish between power set groupings whose elements have or do not have a significant order. In power set groupings G^* where the elements have a significant order, each permutation of a subset $g \in G^*$ is considered to be distinct with its unique identity. In a power set grouping without a significant order defined for its elements, all permutations of a subset $g \in G^*$ are considered to be equivalent. In our model, a power set grouping can be defined based on any existing set grouping which is not a power set grouping. In

[HULL87B] Hull shows that this restriction does not limit the 'information capacity' of the model.

The domain of a power set grouping G* based on a set grouping G is defined as follows. If D is the domain of G, then the domain of G* is defined to be D as well.

Note that the four constructs: *set grouping, is-a related grouping, Cartesian grouping, and power set grouping*, are closely related to the following abstractions respectively: classification, generalization/specialization, aggregation, and association. All of these are commonly found in some form in semantic database models [PECKHAM88].

2.3. Identity

The concept of identity has been introduced implicitly in some semantic models (explicitly in object-oriented systems [KHOSHAFIAN86]) to better cope with complex entities. We use it explicitly: an entity (with identity) consists of two parts — an identity and a state. Let us denote the identity and the state of an entity s1 by s1.id and s1.state, respectively. The identity is time-invariant, i.e., when the state of an entity is modified its identity is unchanged. This is typically implemented by a surrogate — a system-assigned global ID invisible to the user. The identity of a given entity is independent of its current state. Hence it is possible for two entities to have exactly the same state and thus be indistinguishable from one another, without actually being the *same* entity. Note that this is in sharp contrast with mathematical sets, which do not have the concept identity associated with their elements. In a mathematical set each element is essentially the string of symbols used to represent it. When the element is modified (i.e., a symbol is changed), it becomes a *different* element. These observations lead us to the following important definitions:

Definition 1: Let s_1 and s_2 be two elements *with identity* from the set groupings S_1 and S_2 respectively. The relationship $s_1 \doteq s_2$ (pronounced *dot-equal* or *identical*) holds if and only if s_1 and s_2 represent the *same entity*, i.e. $s_1.id = s_2.id$.

Note that some simple entities may not have identities. In this case, the predicate $\doteq$ is based on the state of the entities instead of their identities. Then, the relationship $s_1 \doteq s_2$ holds if and only if s_1 and s_2 represent the *same value*, i.e. $s_1.state = s_2.state$.

To capture the concept of indistinguishability, we need to consider Cartesian aggregates and cover aggregates separately.

Definition 2: Let c_1 and c_2 be two Cartesian aggregates in $C \subseteq S_1 \times S_2 \times ... \times S_n$ with $c_1 = s_1 \times s_2 \times ... \times s_n$ and $c_2 = s'_1 \times s'_2 \times ... \times s'_n$. Then, c_1 and c_2 are called *component-identical*, denoted by $c_1 =^c c_2$, if and only if their components are pairwise identical, i.e., the following holds: $(\forall i)$ $(s_i \doteq s'_i.)$

Definition 3:

(a). Let c_1 and c_2 be simple entities. Then, c_1 and c_2 are *value-indistinguishable*, denoted as $c_1 =^v c_2$, if and only if they have the same state, i.e., $c_1.state = c_2.state$.

(b). Let c_1 and c_2 be Cartesians aggregates defined as in definition 2. Then, c_1 and c_2 are *value-indistinguishable*, denoted by $c_1 =^v c_2$, if and only if their components are pairwise *value-indistinguishable*, i.e., the following holds: $(\forall i)$ $s_i =^v s'_i$.

Note that the last definition is recursive; it stops when applied to simple entities.

To illustrate these two definitions which are based on the work of Khoshafian and Copeland [KHOSHAFIAN86], consider a situation where a new car, say car_2, has been built out of the major parts of another (perhaps damaged) car, say car_1. Car_1

and car_2 have different identities but if only the major parts are recorded by the model, they are indistinguishable. According to definition 2, we refer to entities that share the same components as *component-identical.* On the other hand, two cars could be indistinguishable by virtue of having the same type of body, engine, and wheels and being painted with the same color. Their components, however, would be physically distinct entities (i.e., have different identities). According to definition 3, we refer to such entities as *value-indistinguishable* (or just *indistinguishable*, for short). We now present analogous definitions of the relations $=^c$ and $=^v$ for cover aggregation elements (of power set groupings).

Definition 4: Let c_1 and c_2 be two cover aggregation elements with $c_1 = \{s_1, s_2, ..., s_n\}$ and $c_2 = \{s'_1, s'_2, ..., s'_n\}$. Then, c_1 and c_2 are called *component-identical*, denoted by $c_1 =^c c_2$, if and only if they have the same cardinality (denoted by $|c_i|$) and their elements are pairwise identical. In other words, the predicate $=^c$ evaluates to true if and only if the following holds:

(1) $|c_1| = |c_2|$, and

(2) if both c_1 and c_2 have a significant order then $s_i \doteq s'_i$ for all i. If c_1 and c_2 do not have a significant order then there is an ordering of elements of c_1 and c_2 such that $s_i \doteq s'_i$ for all i. If, without loss of generality, c_1 does and c_2 does not have a significant order then there is an ordering of the elements of c_2 such that $s_i \doteq s'_i$ for all i.

Definition 5: Let c_1 and c_2 be defined as in the previous definition. Then, c_1 and c_2 are called *value-indistinguishable*, denoted by $c_1 =^v c_2$, if and only if they have the same number of elements and their components are pairwise *value-indistinguishable.* The predicate $=^v$ evaluates to true if and only if the following holds:

(1) $|c_1| = |c_2|$, and

(2) if both c_1 and c_2 have a significant order then $s_i =^v s'_i$ for all i. If c_1 and c_2 do not have a significant order then there is an ordering of elements of c_1 and c_2 such that $s_i =^v s'_i$ for all i. If, without loss of generality, c_1 does and c_2 does not have a significant order then there is an ordering of the elements of c_2 such that $s_i =^v s'_i$ for all i.

The last definition is again recursive; it stops when applied to simple elements.

To illustrate the above two definitions, imagine two cover aggregate entities $race_1$ and $race_2$ which model the sets of cars participating in a certain car race. Assume that $race_1$ and $race_2$ refer to two different car races, but that exactly the same cars participate in both. Then, elements from $race_1$ may be paired up with elements from $race_2$, such that car_i from $race_1$ is the same car as car_j from $race_2$. By just looking at the participating cars one would not be able to distinguish between these two sets of cars and hence they would be *component-identical* (by definition 4).

On the other hand, there could be two races in which different cars take part, but these cars pairwise look alike. If for each $race_r$ ($1 \leq r \leq 2$) there exist an ordering ($car_{r,1}$, ... $car_{r,n}$) such that $car_{1,i}$ is indistinguishable from $car_{2,i}$ for $1 \leq i \leq n$, for instance, they have the same type of body, engine, and wheels and are painted by the same color. Then, these two cover aggregates (races) are *value-indistinguishable* by definition 5.

Definition 6: For simple entities s_1 and s_2, the two predicates $s_1 =^v s_2$ and $s_i =^c s_2$ are the same and default to the notion of taking on the same value. This is because simple entities have no components. Both predicates evaluate to true if and only if s_1.state = s_2.state.

Simple entities are either both component-identical and value-indistinguishable or neither.

The above definitions capture an important property of the real-world, where many *different* entities may have the same attributes and external appearance. Furthermore, even if two initially distinguishable entities evolve over time so that they become indistinguishable, their identities as two separate individuals will be preserved.

2.4. Multiple Element Occurrences

In the following discussion, we use the term indistinguishability for both *component-identity* and *value-indistinguishability* and we denote it by the symbol $=$. Although multiple non-distinguishable elements may occur within a given set grouping each identity is unique and persistent. In other words, given two simple elements s_1 and s_2, the relation $s_1 = s_2$ may be true or false, depending on whether s_1 and s_2 are distinguishable or not. However, for distinct entities with identity the relation $s_1 \doteq s_2$ is always false within a single set grouping. Every element in a set grouping has an identity which is distinct from all other elements in that grouping, therefore, for any two elements with identity the predicate $\doteq$ is always false within *one* set grouping. This is based on pragmatic grounds. There is nothing fundamental that would prevent us from allowing multiple occurrences of the same element within a set, however, we did not find any concrete application where this would be necessary.

In the remainder of this section, we study interrelationships between these types of predicates within one set grouping.

Lemma 1: Within a given set grouping S the following holds:

(1) $s1 \doteq s2 \Rightarrow s1 =^c s2$, and (2) $s1 =^c s2 \Rightarrow s1 =^v s2$.

And by (1) and (2), (3) $s1 \doteq s2 \Rightarrow s1 =^v s2$.

Lemma 1 establishes that identical entities always consist of exactly the same components and thus look alike (a natural phenomenon of the world). Also, entities which share the same components are indistinguishable.

Next, we present conclusions that can be drawn when allowing/disallowing indistinguishable elements in a set grouping.

Lemma 2: If no component-identical elements are allowed in a set grouping S, then not only (1), (2) and (3) from lemma 1 hold in S but also:

(4) $s1 \neq s2 \Rightarrow s1 \neq^c s2$.

By (1) and (4), we get: (5) $(s1 \neq s2) \iff (s1 \neq^c s2)$.

Nothing can be concluded, however, from the fact $s1 \neq^v s2$.

Lemma 3: If no (value-) indistinguishable elements are allowed in a set grouping S, then not only (1), (2) and (3) from lemma 1 hold in S but also:

(4) $s1 \neq s2 \Rightarrow s1 \neq^v s2$, and, $s1 \neq^c s2 \Rightarrow s1 \neq^v s2$.

Hence, we have: (5) $(s1 \doteq s2) \iff (s1 =^c s2) \iff (s1 =^v s2)$.

All eight combinations of truth values for the predicates $\doteq$, $=^c$ and $=^v$ are possible within the data model. This is because an entity (base entity as well as the component of an entity) may look differently when it is viewed as a member of different set groupings. For instance, a person has different characteristics as student than when viewed as employee. A person element may have a grade attribute when viewed as a student whereas as an employee s/he may have a salary attribute. Therefore, it may be possible for ($element_1$ in employee-class) $\doteq$ ($element_2$ in student-class) to evaluate to true but ($element_1$ in employee-class) $=^v$ ($element_2$ in student-class) and/or ($element_1$ in employee-class) $=^c$ ($element_2$ in student-class) to evaluate to false. However, lemma 1 guarantees that within one set grouping such a situation cannot occur.

2.5. Ordering

The third extension to the standard definition of a set is to allow one or more orders to be imposed on the elements of a set grouping. For each order, this implies the existence of a predicate "$\leq$", which, when applied to any two elements of a set grouping ($s_1 \leq s_2$) returns true if s_1 precedes s_2 in the ordering and false otherwise. The specification of such an ordering takes one of two forms — an explicit enumeration or an evaluable function. An enumerated order requires that the relationship between elements be specified explicitly. For example, one could order the collection of cars by their price by explicitly listing them as "$car_1 \leq car_2 \leq ... \leq car_n$". At execution time, the boolean value for the predicate "$car_i \leq car_j$" (which can be interpreted as "car_i is less expensive than car_j") is obtained by examining the explicit enumeration. In the second case, a function is specified which derives the boolean value for "$\leq$" through some computation on entities or their components at execution time. Examples of such functions are the lexicographical order on words or the less-or-equal function defined on numeric values. For example, if cars were constructed entities with a price component, then one could order a collection of cars by their price. In that case, the less-or-equal function defined on numeric values would be applied to the price component of the respective cars at execution time and the relation *"$car_1 \leq car_2$"* would be evaluated by "car_1.price is less than car_2.price". Then, an explicit enumeration of all cars is no longer needed to specify such an order.

At most one of the orders defined on a set grouping can be designated as being *significant* or primary. If a set grouping has a significant order, then it will always be represented in that order. Furthermore, this order will be used when comparing the set grouping with other set groupings as described in definition 4 and 5. To clarify the distinction between a significant and non-significant order, assume two set groupings A and B which contain the three alphabetic letters 'n',

't', 'o'. Note that A and B have the alphabetic order defined on them by which 'n' comes before 't', etc. This order is, however, not significant since two sets A = {'n', 't', 'o'} and B = {'o', 't', 'n'} are the same even if we do not list their elements in the same order. If, on the other hand, A and B are to represent the words 'not' and 'ton', then there is a significant order, namely, the order of enumeration, defined on them. In this case, A and B are no longer the same, since they represent two different words. Orders - whether significant or not - will be used in the remainder of this paper in the formulation process of constraints.

3. Restrictions on Set Groupings

3.1. Restrictions

We now present a set of restrictions that may be imposed on any set grouping, including Cartesian aggregation and power set groupings. By combining restrictions, distinct special cases of set groupings are produced, which capture a concept in the application more precisely. There are three kinds of user-specified restrictions based on: cardinality, the number of indistinguishable elements in a set, and the range of element values with respect to a user defined order. All three restrictions are defined below. To simplify future discussions, we will use the term indistinguishable to mean *both* component-identical or value-indistinguishable except when explicitly stated otherwise.

1. The *cardinality* of a set grouping S (denoted by |S|) is the number of elements in S. The cardinality of a set grouping is determined by counting its distinct, but not necessarily distinguishable, elements. By lemma 1 an element may not occur more than once in a set grouping. Therefore the cardinality of a set grouping is easily determined. Cardinality can be restricted by giving an integer range $[c_1{:}c_2]$ where $c_1 \leq c_2$ and c_1 and c_2 are the lower and upper limit, respectively. If $c_1 = c_2$ then the set grouping S must have exactly c_1

elements at all times during its existence in the database. If $c_1 < c_2$ then the cardinality can vary within that range. The values $c_1 = 0$ and $c_2 = \infty$ impose no restrictions on the cardinality of the grouping. The latter is assumed to be the default.

2. The *repetition* characteristic of a set grouping S specifies how many indistinguishable copies of an element may exist within S. The restriction has to be parameterized by $=^c$ and $=^v$ to indicate which type of indistinguishability is to be restricted. The repetition count can be limited by specifying a range $[r_1{:}r_2]$ where r_1 and r_2 are integers and $0 \leq r_1 \leq r_2$. This can be interpreted as: for any element s from the domain of S there exist at least r_1 and at most r_2 distinct elements of S which are indistinguishable from s (including s). If $r_1 = r_2$ with parameter $=^c$ or $=^v$ then for every element s_1 in the set grouping there must be exactly $r_1 - 1$ elements s_i (i = 2, ..., r_1) with $s_1 =^c s_i$ or $s_1 =^v s_i$. Recall that $s_1 \doteq s_2$ is always false within a set grouping, and consequently, $s \neq s_i$ for all i. The special case where $r_1 = r_2 = 1$ and the parameter is $=^v$ implies that no two elements in the set grouping are allowed to look alike. By lemma 3, this also implies the implicit restriction of a repetition count of $[r_1, r_2] = [0,1]$ with parameter $=^c$. More general, by lemma 3 any repetition restriction $[r_1, r_2]$ with the parameter $=^v$ enforces the upper bound of the range for parameter $=^c$ never to exceed r_2. Again, the unrestricted case, $r_1 = 0$ and $r_2 = \infty$ with either parameter, is assumed to be the default.

3. For a set grouping S with an order defined over its elements, a sequence of *ranges* $[l_1{:}u_1]$, $[l_2{:}u_2]$, ..., $[l_n{:}u_n]$ may be specified with respect to that order. Only elements which are contained within one of the ranges may appear in S. More precisely, an element s may appear in S if for one of the ranges

$[l_i{:}u_i]$ $(1 \leq i \leq n)$ the following holds: $l_i \leq s \leq u_i$. If no range is specified then the set grouping is unrestricted.

Since not all three restrictions need to be specified for all sets, we use the following labels to designate which restriction is being referred to: (i) set cardinality, (ii) set repetition with $=^c$ or $=^v$, and (iii) ordering.

3.2. Examples of Set Groupings and their Restrictions

In this subsection examples of restricted set groupings are presented which demonstrate the usefulness of various combinations of these three basic restrictions.

Example 1:

A committee S of n people can be characterized by:

1. set cardinality: [n:n].

   Interpretation: The cardinality of S is fixed to n. The fact that any person can occur in S at most once, i.e., the same person cannot act as two or more committee members, is automatically enforced by definition 1. The repetition restriction does not apply, since we do not care whether two people look alike (value-indistinguishability) or possibly even share similar properities, such as, live at the same address (component-identity).

Example 2:

Let S model the collection of words in a dictionary. Its domain is the collection of character strings formed from a given alphabet. This set grouping can be characterized by the following:

1. set cardinality: $[100 : \infty]$
2. set repetition with $=^v$: [0:1],
3. ordering: $[a : z^+]$.

Interpretation: We assume that a dictionary must have at least hundred words; the maximum number is unrestricted. A given string may occur at most once as a word in the dictionary, and since strings are simple entities without identity we may enforce this by the repetition restriction with the parameter $=^v$. Since all words are ordered alphabetically, the "smallest" word is the letter 'a' and the "largest" word is the infinite sequence 'z...z', denoted as z^+. This implies that no special symbols (like quotes or hyphens) would be allowed in our simple dictionary.

Example 3:

The conventional set S may be modeled as a special case of a set grouping by the following restrictions:

1. set repetition with parameter $=^v$: [0 : 1]

Interpretation: The cardinality of S can take any value from zero (empty set) to infinity, and thus no restriction is specified for it. The elements, taken from an underlying domain, are simple entities without identity; each may occur at most once in the set. Elements are unordered and thus a range restriction is not applicable.

While the restrictions presented in this section could be applied to any set grouping, there are certain additional restrictions that may be applied only in the case of Cartesian or power set groupings. Furthermore, even the three basic restrictions may not always be applied freely in the case of non-base set groupings. The various constraints and possible additional restrictions will be presented in the next three sections.

4. Restrictions on IS-A Related Groupings

4.1. Restrictions

In this section, we discuss restrictions that may be imposed on non-base groupings that are part of the is-a hierarchy. We shall refer to these as *IS-A related set groupings.* Base set groupings do not depend on any other set groupings and thus the three restrictions as discussed in the previous section can be freely imposed on them. The most common representatives of IS-A related groupings are subset and union groupings, which were introduced in section 2. Such groupings are based on existing set groupings (through various derivation rules) and, consequently, additional constraints have to be met when applying the three set grouping restrictions to them. This is because there are strong interrelationships between all set groupings which are IS-A related to one another. Below, we discuss the constraints on applying the three kinds of set grouping restrictions to subset and union groupings. For subsets, the following must hold:

1. The *cardinality* of a subset grouping S based on the sets S_i $(1 \leq i \leq m)$ must meet the following additional constraint. If $[c_{il}{:}c_{iu}]$ is the cardinality constraint for the set grouping S_i (for all i), then the cardinality constraint $[c_l{:}c_u]$ for the subset grouping S must satisfy the restriction : $c_u \leq \min_{i=1}^{m} c_{iu}$. This guarantees that the cardinality of S is always less than or equal to the cardinality of the smallest set S_i. The lower bound is not restrained, and can take on any value between 0 and the upper bound. Note that this is a powerful mechanism which may cause the non-base grouping (S) to force a restriction on the is-a hierarchy, i.e., the set groupings underlying S. It permits you to state the minimum number of elements which the underlying set groupings have to have in common. For example, if $|S_1|$ = n and $|S_2|$ = m and the lower cardinality bound of S $\subseteq (S_1, S_2)$ is k, then S_1 and S_2 must share at least k elements.

2. The *repetition* characteristic of a subset grouping S, which specifies how many indistinguishable copies of an element may exist within S, must meet the following constraint. Let $[r_{il}:r_{iu}]$ be the repetition constraint with parameter $=^c$ $(=^v)$ for the set grouping S_i (for all i). Then the repetition count $[r_l:r_u]$ with parameter $=^c$ $(=^v)$ for the subset grouping S has to satisfy the following: $(\forall$ i) ($r_u \leq r_{iu}$). The lower bound can again take any value between 0 and the upper bound.

3. If all underlying set groupings S_i have the same kind of order defined on them, then the subset grouping S *can* be ordered by the same ordering. In this case, the range restriction of the subset grouping consists of a subset of the intersection of the range restrictions of the underlying S_i. It is also possible to define additional orders and to further restrict S by specifying a range on the explicitly defined order.

Let us now consider the characteristics and restrictions on union set groupings. By lemma 1, an entity is only allowed to appear once in any set grouping. Therefore, an entity will occur only once in the union grouping S even if it occurs in more than one of the underlying set groupings. A union grouping may contain indistinguishable elements even if none of the set groupings participating in the union contain duplicates.

1. The cardinality constraint of the resulting union grouping S is determined from the cardinality constraints of the involved set groupings $S_1, S_2, ..., S_m$. Let $[s_{il} : s_{iu}]$ be the cardinality constraint for the set grouping S_i for all i = 1, ..., m. Then the cardinality constraint for S will be [l:u] with:

   $l \geq \max_{i=1}^{m} s_{il}$ and

   $u \leq \sum_{i=1}^{m} s_{iu}$.

   This guarantees that the lower bound of the cardinality of S is no smaller than the cardinality of the set grouping with the largest lower bound and no

smaller than the sum of the maximal cardinalities of all set groupings. Note that this is a very powerful constraint which permits you to state how many elements the underlying set groupings have in common. For example, if $|S_1|$ = n and $|S_2|$ = m and the upper cardinality bound of $S_1 \cup S_2$ is k, then S_1 and S_2 must share n + m - k elements. Hence, this constraint should be used carefully because it may result in a union which is so restricted that it will always be empty.

2. The number of indistinguishable elements that appear in the union depends on the number of indistinguishable elements in the set groupings underlying the union. Let $[r_{il} : r_{iu}]$ be the repetition constraint on the set grouping S_i for all i = 1, ..., m. Then the constraint on S, denoted by [l:u], must be as follows:

   l $\geq \min_{i=1}^{m} r_{il}$ and

   u $\leq \sum_{i=1}^{m} r_{iu}$.

3. A union grouping can be ordered if all underlying set groupings are ordered by the same type of ordering. This means that all elements of the union have at least one attribute in common and the ordering is based on a common attribute. The range associated with the order is the union of the ranges of all set groupings.

4.2. Examples of IS-A Related Set Groupings and their Restrictions

In this subsection examples of IS-A related set groupings are presented.

Example 1:

Let S model the collection of words in a dictionary as described in example 2 of section 3. Then the set grouping Sub = {s | s is a word in a dictionary starting with the letter 'a'} is a subset grouping of S. It can be characterized by the following:

1. set repetition with $=^v$: [0:1],
3. ordering: $[a : b)$.

Interpretation: The fact that a given string may occur at most once as word in the dictionary is a restriction directly inherited from S. Since S is ordered, a range restriction can also be specified on Sub. In this case, all words of Sub must start with the letter 'a'.

Example 2:

Let S_{ICS} and S_{ENG} be the set of all students enrolled in the ICS and engineering departments, respectively. The corresponding cardinalities are $|S_{ICS}|$ = 100 and $|S_{ENG}|$ = 50. Then the set grouping S, which contains all students with a double major, i.e., enrolled in the ICS as well as engineering department is a subset grouping of both. S $\subseteq$ (S_{ICS} , S_{ENG}) could be constrained by:

1. set cardinality: [0 : 50]

Interpretation: The set grouping S contains only students who are in both underlying set groupings, S_{ICS} and S_{ENG}. The upper bound on the cardinality of S is set to the maximum possible value, according to the rules of section 4.1. That is, all 50 engineering students could possibly have a double major with ICS. Neither repetition nor order restrictions are imposed on S.

Example 3:

Let S_{ICS} and S_{ENG} be as in the previous example and assume that students are ordered by their GPA. Let the set grouping S contain all double major students who have a grade point average of 3.5 and better. Then S $\subseteq$ (S_{ICS}, S_{ENG}) is constrained by:

1. set cardinality: [0 : 50]
2. ordering on GPA: [3.5 : 4.0]

Interpretation: S's cardinality is again limited by the cardinality constraints of S_{ICS} and S_{ENG}. Since both set groupings are ordered by the students' GPA, a range restriction can be imposed. Note that, this time, S is a true subset of the two underlying set groupings (assuming that some of the double majors have GPAs of less than 3.5), whereas in the previous example, S corresponds to their intersection - a special case of a subset grouping.

Example 4:

Let S_1 be the catalogoue of books of the Computer Science library, and S_2 be the catalogoue of books of the Mathematics library with $|S_1| \geq 1000$ and $|S_2| \geq 500$. The two libraries wish to combine their catalogues in order to have access to more material, especially since overlapping interest and thus books exist. Then, the combined book catalogue $S = S_1 \biguplus S_2$ can be modeled by:

1. set cardinality: [1300 : ∞]
2. set repetition with parameter $=^v$: [0:15],
3. ordering: ['QA':'QB') & ['Y':'1').

Interpretation: The resulting cardinality of S is constrained to be at least 1300. This implies that S_1 and S_2 together must have referenced at least 1300 distinct books. Repetitions in the combined library should be at most as high as in the individual libraries. So if S_1 never kept more than 15 duplicates and S_2 never more than 5 duplicates of a given book, then S will have at most 15 duplicates of any book. In the case that both libraries had kept the maximally allowable number of copies of a given book, then 5 of these must have been shared. All university books are categorized by the same library code, and hence, there is again a order defined on the unified collection. The range associated with the order has to be extended to encompass both ranges. Assuming that S_1

had the range ['QA':'QB') & ['Z':'1') and S_2 the range ['Y':'Z'] then S is described by the union of these ranges, namely, ['QA':'QB') & ['Y':'1').

5. Restrictions on Cartesian Groupings

5.1. Restrictions

The Cartesian grouping is itself a set grouping, containing complex aggregate entities. Therefore the set grouping restrictions of section 3 also apply to Cartesian groupings. In addition, there are restrictions specific to Cartesian groupings. Each of these may be repeated zero or more times within a Cartesian grouping description. They have the general form [p, r] where p is some combination of positions and r a range restriction. For each possible p, only one restriction of each type may be specified., i.e., if [p, r_1] and [p, r_2] then r_1 has to be equal to r_2.

The first two restrictions limit the number of times an element from a set grouping S_i may appear within some prespecified positions of C. These two restriction types are parameterized by $\doteq$ (sameness) , $=^c$ (component-identity) or $=^v$ (value-indistinguishability). The first applies to the entire Cartesian grouping while the second applies to individual elements of C. The third restriction is concerned with aggregate projections, and the fourth characterizes constraints on relationships that can be modeled by a Cartesian aggregation. This is similar to the idea of "functional dependencies" in the relational model.

Before the restrictions can be presented formally, we need to introduce the notion of *appearance* of an entity within a Cartesian grouping element:

Definition 7: Let L_i with i = 1, ..., m be labels for some of the set groupings S_i underlying C. These set groupings S_i have to be defined on the same domain D. Let the symbol $\diamond$ stand for one of the three predicates $\doteq, =^v$ *or* $=^c$. Let t $\in$ C and t[$L_1, ..., L_m$] be the aggregate projection of t on the labels L_i as defined on

section 2. Then x *appears* in t[$L_1, ..., L_m$] with parameter $\diamond$ if and only if at least one of the following two conditions holds:

1. for some L_i with S_i not a power set grouping:

$$x \diamond t[L_i],$$

2. or for some L_i with S_i a power set grouping:

$$\exists\, e \in t[L_i]\text{: } x \diamond e.$$

Let the aggregate projection of C be C[$L_1, ..., L_m$] = $\{t[L_1, ..., L_m] | t \in C\}$. Then x *appears* in C[$L_1, ..., L_m$] with parameter $\diamond$ if and only if a t exists in C such that x *appears* in t[$L_1, ..., L_m$] with parameter $\diamond$.

For the first two restrictions, let us assume that p_1, p_2, ..., p_m are positions of C whose set groupings have the same domain. The restrictions have the general form [p_1, p_2, ..., p_m, [r_1:r_2]] where the p_i's identify the positions to be restricted and [r_1:r_2] with $0 \leq r_1 \leq r_2$ gives the allowable range.

1. This restriction limits the number of times an entity may *appear* within the designated positions of all aggregate elements of C, i.e., within the projection C[$p_1, ..., p_m$]. If the parameter $\doteq$ is specified, then this 'appearance restriction' refers to the number of occurrences of a single entity across the positions of all occurrences of C. If the parameter $=^c$ ($=^v$) is specified, then it refers to the number of component-identical (value-indistinguishable) elements. The range [r_1:r_2] states that an element of the domain underlying these p_i positions has to *appear* at least r_1 and at most r_2 times within the p_i positions over all aggregate elements of C. When the restriction is applied to a single position a number of interesting special cases can be modeled. In particular, [p_i, [r_1:r_2]] with $r_1 = r_2 = 1$ and parameter $\doteq$ captures the idea that position p_i is a key for elements of C. It states that position p_i is unique for each aggregate element. The same restriction with the parameter $=^v$ provides a mechanism for modeling the concept of a "value-based" key, which means

that all elements of C can be distinguished based on the value of their i^{th} component. Note that a value-based key corresponds to the concept of a key in the relational model.

2. This restriction limits the number of times an entity may *appear* within the designated positions within *a single element* of C. Again, a parameter of the restriction regulates whether this refers to the sameness or the indistinguishability of entities. The specification of the restriction and their meanings are equivalent to those of restriction 1. The major difference is that this restriction imposes limits on the appearance of an element within a single aggregate element while the first restriction limited its appearance in C as a whole.

The third restriction is concerned with combinations of values across the positions within aggregate projections of C. For the following, let X and Y be collections of the set groupings underlying distinct positions of C. X corresponds to the group of positions limited by the component-identity parameter and Y corresponds to the group of positions restricted by the value-indistinguishability parameter. Then, the restriction has the general form [(X) with $=^c$ $\times$ (Y) with $=^v$, [r_1:r_2]] where [r_1:r_2] ($0 \leq r_1 \leq r_2$) is the allowable range. Note that either X or Y may be omitted.

3. Restriction 3 limits the number of elements in C which have a certain combination of values in C[X $\times$ Y]. The user specifies whether the constraint is based on $=^c$, $=^v$, or a combination of both. In the last case, all positions for which $=^c$ is to be applied are listed first in parenthesis with the ones to which $=^v$ applies thereafter. For instance, [(p_1,p_2) with $=^c$ $\times$ (p_3) with $=^v$, [r_1:r_2]]. If a single parameter $=^c$ or $=^v$ is specified, then the restriction ranges over a single collection Z (for example, [(p_4,p_5,p_7) with $=^v$, [r_1:r_2]). This restricts the number of elements t of C which have value-indistinguishable t[Z] components. In this case, the range [r_1:r_2] states that for a given aggregate

projection v of C[Z] there must be at least r_1 and at most r_2 distinct aggregate elements $t_i \in C$ with $t_i[Z] =^v v$ for $1 \leq i \leq r_2$. If a combination of these two parameters is specified, then the number of elements t of C which have component-identical aggregate projections t[X] and value-indistinguishable aggregate projections t[Y] are both restricted. This implies that, for a given aggregate projection value of C[XY] there must be at least r_1 and at most r_2 distinct aggregate entities in C such that $(t_i[X] =^c t_j[X])$ and $(t_i[Y] =^v t_j[Y])$ for all $1 \leq i,j \leq r_2$.

The fourth restriction constrains dependencies between combinations of values across the positions of all aggregate projections of C. Let X and Y again denote collections of labels for set groupings underlying distinct positions of C. Then, the restriction with the general form [X / Y, $[r_1:r_2]$] limits the number of elements of C which have the *same* value combination for X but distinct values for Y. In other words, it limits to how many different Y values an X value can be related to by C. Note that we have two types of *sameness* for aggregate projections t[X], namely, $=^c$ and $=^v$. Similarly there are two interpretations for aggregate projections t[Y] to be considered *distinct*, namely, $\neq^c$ and $\neq^v$. Consequently, these are used as parameters to indicate which type of sameness or distinctness we are referring to.

4. Restriction 4 limits the number of distinguishable (with interpretation $\neq^c$ or $\neq^v$) elements from C[Y] that can occur together with a certain same value combination of C[X] (with interpretation $=^c$ or $=^v$) across all aggregate elements in C. This class of restrictions has the general form [X with parameter / Y with parameter, $[r_1:r_2]$]. For a given parameter pair par_1 for X and par_2 for Y, the range $[r_1:r_2]$ states that for each aggregate projection x of C[X] there must be m (with $r_1 \leq m \leq r_2$) distinct elements t_{i_1}, t_{i_2}, ..., t_{i_m} in C with $(t_{i_j}[X]\ par_1\ x)$ and $(t_{i_j}[Y]\ par_2\ t_{i_k}[Y])$ $(j \neq k)$ for all $1 \leq j,k \leq m$. For instance, if the parameter pair is $=^v$ for X and $\neq^v$ for Y, then for each

aggregate projection x of C[X] there must be m (m at least r_1 and at most r_2) distinct elements in C with ($t_{i_j}[X] =^v x$) but ($t_{i_j}[Y] \neq^v t_{i_k}[Y]$) ($j \neq k$) for all j, k from 1 to m.

In the examples below, we shall use the following labels to denote the four possible Cartesian grouping restrictions: (i) appearance in C (with parameter $\doteq$, $=^v$ or $=^c$), (ii) appearance in one entity (with parameter $\doteq$, $=^v$ or $=^c$), (iii) aggregate projection (with parameter $=^v$, $=^c$ or both), and (iv) dependency (with the parameter pairs ($=^v$ or $=^c$) and ($\neq^v$ or $\neq^c$)).

Note that each position in a Cartesian grouping C may contain elements from other set groupings, including entities from power set groupings. This allows us, for example, to specify the cardinality of a 'multi-valued component' directly at the underlying set grouping level. It is done by defining a power set grouping based on set grouping S_i and then constraining the cardinality for the power set's elements. To clarify this consider the following simple example. Suppose a hotel reservation grouping includes a position for the number of people who will share a room. If a group of up to 3 people is allowed to reserve a single hotel room, then the set grouping underlying the people position of the reservation aggregation would be a power set of the set of persons with the restriction that its elements (subsets of people) contain between 1 and 3 people. The power set grouping is a mechanism powerful enough to determine many other important types of relationships between the elements of a set grouping as will be demonstrated in section 6.

5.2. Examples of Cartesian Groupings and their Restrictions

Different combinations of restrictions specific to Cartesian groupings are discussed below.

Example 1:

Consider a set of committees of company officers represented by Company-Officers $\subseteq$ (President: person) $\times$ (Treasurer: person) $\times$ (Vice-Presidents: person*). Each committee must be composed of exactly 1 president, exactly 1 treasurer, and any number of vice presidents. Then, the Cartesian grouping can be further characterized by the following restrictions:

1. appearance in C with $\doteq$: [president, [0:1]]
2. appearance in C with $\doteq$: [president, Treasurer, Vice-Presidents [0:5]]
3. appearance in an element with $\doteq$: [President, Treasurer, Vice-Presidents [0:1]]

Interpretation: Since people have unique identities only restrictions with the parameter $\doteq$ are applicable. The first constraint limits the number of times an element from the domain person may appear in the president position to a maximum of one. That is, a person can be president of at most one company. The second declares that a person may not belong to more than five different committees (in any position) simultaneously, i.e., a person can not hold more than 5 offices. The third restriction refers to an individual committee, and hence, insures that a person cannot hold more than one position on a single committee. Note that further restrictions could be imposed, for example, on how many different committees exist or whether some of them contain the same people, using the set grouping restrictions of section 3. Similarly, additional restrictions could be placed, for example, on the number of Vice Presidents using power set restrictions, which will be discussed in section 6.

Example 2:

A flight reservation can be viewed as a relationship between a person, a date, a plane, and a seat with the corresponding domains: Reservation $\subseteq$ (Person

: persons) $\times$ (Date : dates) $\times$ (Plane : planes) $\times$ (Seat : seats). Seats are indistinguishable entities half of which are 'business class' (BC) while the remainder are 'tourist class' (TC). This is captured by the following six restrictions:

1. aggregate projection with $=^c$: [Plane , [20,400]]
2. aggregate projection with $=^c$: [Date $\times$ Plane $\times$ Person, [0:1]]
3. aggregate projection: [(Date $\times$ Plane) $(=^c)$ $\times$ (Seat) $(=^v)$, [0:150]]
4. aggregate projection with $=^c$: [Date $\times$ Plane $\times$ Seat, [0:300]]
5. dependency: [(Date) $(=^v)$ $\times$ Plane with $\neq^c$, [0:15]]
6. dependency: [(Date $\times$ Plane) $(=^c)$ / (Person) $(\neq^c)$, [20:300]]
7. dependency: [(Plane) $(=^c)$ / (Seat) $(\neq^v)$, [2:2]]

Interpretation: The first restriction represents the fact that this reservation system monitors between 20 and 400 different planes. The second indicates that a person needs only one seat on a given plane and date. Constraint number 3 states that for a given plane and date there are at most 150 reserved seats of the each type ($0 \leq$ number of BC seats, number of TC seats ≤ 150). The lower bound is set to 0, since this restriction refers to the number of reserved seats rather than their actual count. Restriction number 4 models the fact that each plane has a maximum of 300 seats that can be reserved. The next constraint states that there are at most 15 different flights (planes) scheduled on any date. The sixth restriction indicates that a flight must have a minimum occupancy of 20 passengers and a maximal load of 300. This is because each reservation includes exactly one seat and exactly one person; therefore, the maximum of passengers depends on the number of seats on the plane. Finally, restriction 7 indicates that a plane has exactly two types of seats (namely, TC or BC).

Example 3:

A course can be viewed as a relationship between a professor, a teaching assistant (TA), and a class of students. It has the form Course $\subseteq$ (Professor : person) $\times$ (TA : Students) $\times$ (Class : Students*). Possible restrictions are:

1. appearance in C with $\doteq$: [Prof, [3:5]]
2. appearance in an element with $\doteq$: [TA, Class, [0:1]]
3. aggregate projection with $=^c$: [Prof $\times$ TA, [0:3]]
4. dependency: [Prof ($=^c$) / TA ($\neq^c$), [0:2]]

Interpretation: The first restriction guarantees that a professor will teach at least three and at most five courses. The second constraint ensures that a student is not both a TA and a participant in the same course. Restrictions 3 and 4 regulate the frequency of interactions between professors and TAs. Number 3 states that a professor will have to deal with any given TA at most three times (i.e., in three different courses), and vice versa. Number 4 guarantees that a professor will have to deal with at most two different TA's at any given time.

6. Restrictions on Power Set Groupings

6.1. Restrictions

A power set grouping is a set grouping and therefore the set grouping restrictions described in section 3 may be applied to it as in the case of Cartesian groupings. In addition, a power set grouping's definition can restrict the structure of the cover aggregates which are its elements. These additional restrictions are listed below. To disambiguate the following discussion the term element$_{ps}$ is used for an element of a power set grouping G* and element$_g$ for an instance of the underlying set grouping G.

1. The cardinality of each element$_{ps}$ ps $\in$ G* must be in the range [k_1:k_2]. If k_1=k_2 then only subsets of G with cardinality k_1 are potential valid instances

of G^*. If $k_1 < k_2$ then all subsets of G with cardinality within that range are potential valid elements. The extreme case, where $k_1=0$ and $k_2=\infty$, imposes no cardinality constraints on the subsets of G; this is assumed as default.

2. This restriction limits the number of indistinguishable copies of an element$_g$ $g \in G$ within a given element$_{ps}$ $ps \in G^*$. A range $[l_1,l_2]$ and a parameter to indicate the applicable notion of indistinguishability are specified. If $l_1=l_2$ with parameter $=^c$ $(=^v)$ then for each element $g \in ps$ with $ps \in G^*$ there must be $(l_1 - 1)$ other elements in ps that are component-identical (value-indistinguishable) from g. All other special cases are analogous to those for set groupings (in section 3).

3. This restricts the number of different cover aggregations ps of G^* in which an element g of G may participate. Again, the range restriction $[j_1,j_2]$ with the parameters $\doteq, =^c, or\ =^v$ may be specified. If the parameter is $=^c$ $(=^v)$ then component-identical (value-indistinguishable) copies of an element g of G may participate in n elements ps of G^*, where $j_1 \leq n \leq j_2$. If $j_1=j_2$ and the parameter is $=^c$ or $=^v$ then indistinguishable copies of all elements of G participate in exactly j_1 elements of G^*. If $j_1=j_2=1$ and the parameter is $=^c$ or $=^v$ then a value based intersection of cover aggregations ps is always empty. The parameter $\doteq$ requires that an element $g \in G$ participates in at least j_1 and at most j_2 elements$_{ps}$ $ps \in G^*$. This corresponds to the total number of elements$_g \in G^*$ since, by definition 1, an element with identity can appear only once in any of the cover aggregates $ps \in G^*$. For any parameter, the default is $j_1=0$ and $j_2 = \infty$.

4. This characteristic restricts the total number of occurrences of indistinguishable elements$_g$ $g \in G$ within all elements$_{ps}$ $ps \in G^*$. It is parameterized by $=^c$ and $=^v$. (Conceptually, the $\doteq$ parameter would also make sense, but, due to definition 1, this restriction is identical to restriction 3 with parameter $\doteq$.)

The total occurrence restriction is again specified by a range [m_1:m_2]. If m_1=m_2 then there are exactly m_1 indistinguishable copies of each element g of G which participate in G*, i.e., for each element g $\in$ G, there are g_i ($1 \leq i \leq m_1$) copies of g such that $g_i \in$ ps for some ps $\in G^*$. If m_1<m_2 then the total number of indistinguishable copies of elements g from G in G* may vary between m_1 and m_2. An important special case is m_1=m_2=1; this restriction does not allow any duplicates within any element ps of G* nor within G* as a whole. In this case, G* is a simple partitioning of G, where each partition is one element of G*. Once again the defaults are m_1=0 and m_2=∞.

Note that the four measures are closely interrelated, and hence setting one may influence the others. It is this interplay which allows the model to formulate rather complex semantic concepts, as will be illustrated by the examples which follow.

Most applications seem to utilize restrictions using either the sameness parameter or the indistinguishability parameters. One reason for this is that when representing power sets, it often does not matter whether two or more of the underlying entities look alike. For example, when we model a group of people (let's say a committee) the fact that two people have a similar appearance does not have any bearing on the composition of this group. On the other hand, when we model a situation where the existence of indistinguishable copies of an entity is important, we are usually not be concerned about whether they also have matching identities. For example, consider a set of courses $\{A, B, C\}$ where each course may be taught more than once. In this case, we want to model the fact that different sessions of a particular course are indistinguishable from each other even if they have different identities.

The labels used to specify power set restrictions are (i) element$_{ps}$ cardinality, (ii) element$_g$ repetition within each element$_{ps}$ with $=^v$ or $=^c$, (iii) element$_{ps}$ overlap

with $\doteq$, $=^v$ or $=^c$, and (iv) total participation of elements$_b$ in all elements$_{ps}$ with $=^v$ or $=^c$.

6.2. Examples of Power Set Groupings and their Restrictions

The following examples demonstrate the utility of the characteristics defined above.

Example 1:

The set G* of study groups of up to 3 students over the set G of students can be characterized by:

1. element$_{ps}$ cardinality: [2,3]

   Interpretation: A group ps $\in$ G* has 2 or 3 students. The fact that each student can participate in any group ps at most once is taken care of automatically by the model (definition 1). Also, the fact that a student may participate in any number of study groups (including none) follows from the defaults.

Example 2:

The set G* of convoys of ships over the set G of ships can be characterized by:

1. element$_{ps}$ cardinality: $[2 : |G|]$
2. element$_{ps}$ overlap with $\doteq$: [0:1]

   Interpretation: A convoy ps $\in$ G* has at least two and at most all ships from G in it. Again, the fact that each ship occurs in a convoy at most once is automatically enforced by the model (definition 1). The second restriction specifies that each ship g $\in$ G may participate in at most one convoy ps $\in$ G*.

Example 3:

Let G be the set grouping consisting of all students who take ICS courses. Then, G^*_{grad} is a power set grouping based on the set grouping G defined by G^*_{grad} = { s | s is the group of students in an ICS graduate course }. It can be restricted by:

1. set cardinality: [20:25]
2. set repetition with $=^c$: [0:5]
3. element$_{ps}$ cardinality: [6:30]
4. element$_{ps}$ overlap with $\doteq$: [3:5]

Interpretation: Restrictions 1 and 2 are general set grouping restrictions, while 3 and 4 are specific to power set groupings. Restriction 1 models the fact that there are between 20 and 25 graduate courses offered by the ICS department ($|G^*_{grad}|$). Several courses may have exactly the same group of students enrolled in them and, therefore, some courses may be component-indistinguishable. Consequently, the second restriction allows for indistinguishable courses. The third constraint limits the size of graduate classes to be in the range from 6 to 30 students. Finally, the last constraint states that a graduate student has to take between 3 to 5 courses.

Example 4:

The power set G* defines the teaching assignment of instructors for the duration of three terms over a set G of courses. It can be described by:

1. set cardinality: [15:20]
2. element$_{ps}$ cardinality: [2:5]
3. element$_g$ repetition with $=^v$: [0:4]
4. element$_{ps}$ overlap with $=^v$: [0:6]
5. total participation with $=^v$: [1:6]

Interpretation: We assume that the department has between 15 to 20 instructors (restriction 1). Each has to teach at least 2 and at most 5 courses during a year (restriction 2). An instructor may teach a given course up to four times (restriction 3). By restriction 4, a course can be taught by at most 6 different instructors. A course may be taught up to 6 times within the year but must be taught at least once (restriction 5).

Example 5:

Let G_{word} be the set of all words in a given dictionary defined as the power set over the set G of all characters. The word order, the enumeration of letters within a word, is considered to be a significant order. The following restrictions could be applied:

1. set cardinality: [60000:60000]
2. set repetition with $=^v$: [0:1]
3. element$_{ps}$ cardinality: [1:28]
4. element$_g$ repetition with $=^v$: [0:10]
5. element$_{ps}$ overlap with $=^v$: [50:40000]
6. total participation with $=^v$: [50:400000]

Interpretation: The first restriction assumes that there are exactly 60,000 words in the dictionary. The second states that each word appears only once. The third restriction refers to the cardinality of the subsets, i.e., the minimal and maximal word length. (One of the shortest English words is the article "a" while the the longest English word known to us is "antidisestablishmentarianism" comprising 28 characters). We assume that a given character can be repeated up to 10 times in a word (restriction 4). For the purposes of this example it is further assumed that each letter appears in at least 50 and at most 40,000

words (restriction 5). The total occurrences of a character in the entire dictionary will be at least 50 and at most 400,000 times. This final upper bound was derived by combining the fact that a character can appear in up to 40,000 different words (restriction 5) with the fact that this same character can appear up to 10 times within the same word (restriction 4).

Example 6:

Let G^* represent a tennis tournament over the set of players G. Every player must play against every other player except himself. The following restrictions have to be imposed:

1. set cardinality: $[(|G| * |G| - |G|)/2 : (|G| * |G| - |G|)/2]$
2. set repetition with $=^c$: [0:1]
3. element$_{ps}$ cardinality: [2:2]
4. element$_{ps}$ overlap with $\doteq$: $[|G| - 1 : |G| - 1]$

Interpretation: There will be exactly $|G| * |G| - |G|)/2$ different matches, since every player plays against everybody else (restriction 1). Each player may occur in a match at most once which is guaranteed by definition 1. Furthermore, there is no significant order defined on the elements of the power set, since A playing against B is the same as B playing against A. Consequently, no two matches within the tournament are identical (restriction 2). The cardinality of teams is 2 (restriction 3). Each player will participate in $|G|$ - 1 matches, since s/he does not play against him/herself.

7. Conclusion

The concept of a semantic grouping is a major focus in semantic database modeling research. In this paper we have presented enhancements to several kinds

of groupings which form the basis of most semantic database models. We have enriched the basic groupings by identifying a number of semantic restrictions for each. Combinations of these constraints produce potentially new variations of semantic groupings and provide us with higher-level mechanisms to more accurately model real world concepts. Furthermore, they allow us to place integrity constraints directly into the structure of the database model. Hence, this work represents another step towards overcoming a major disadvantage of conventional database systems, which have to maintain constraints separately from their data and to enforce them explicitly.

The emphasis of our work has been on identifying and formulating these new modeling constructs. We present a general framework which attempts to capture the most important real-world phenomena. In any given application only a, possibly small, subset of the proposed restrictions will be useful. The framework is based on pragmatics and hence examples are provided throughout the paper to show its potential usefulness. Real world knowledge is too unstructured and heterogeneous to hope for a rigorous formal framework that could be proven "correct" or "complete" in any mathematical sense.

There are, of course, issues which still have to be resolved. One of these is the, perhaps intractable, problem of resolving inconsistencies between the various restrictions that can be placed on different parts of a database. For now, we consider it to be the responsibility of the database designer to determine a sensible collection of constraints. Furthermore, we believe that since database applications attempt to captured information as it exists in the real world, inconsistencies will be the exception rather than the rule. An example of an inconsistent specification that can never be satisfied is the following restriction tuple for a set grouping: (1) set cardinality [5:5], (2) set repetition [3:3]. The first restriction requires the cardinality to be exactly five and the second states that each element must repeat exactly

three times which implies that the cardinality is a multiple of three. Violations of constraints could indicate one of two things: either the user has entered some data incorrectly or the specified constraints are indeed inconsistent. The first case is preferable to the user, since this acts as a protection from entering inappropriate data. In the second case, the user will have to reconsider the original specification. Once, the changes have been made, the loading (and checking) phase continues.

Another problem, currently under investigation, is to determine which of these constraints can be enforced *efficiently*. Finally, we will look for heuristics which minimize the possibility of inconsistencies between the restrictions without losing the modeling power gained through specifying them in the first place.

References

[Abiteboul87] Abiteboul, S. and Hull, R. IFO: A Formal Semantic Database Model. *ACM Trans. on Database Systems 12*, 4 (Dec., 1987), 525–565.

[Bic86] Bic L. and Gilbert J.P. Learning from AI: New Trends in Database Technology. *Computer 19*, 3 (Mar., 1986), 44–54.

[Borgida87] Borgida, A. Conceptual Modeling of Information Systems. In *On Knowledge Base Management Systems*, Brodie, M.L. and Mylopoulos, J., Ed., Springer-Verlag, 1987.

[Brodie80] *Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling*, Brodie, M.L. and Zilles, S.N, Ed., Sponsored by the Nat'l. Bureau of Standards, ACM SIGART, SIGMOD and SIGPLAN, Pingree Park, Colordo, 1980.

[Chen76] Chen, P.P. The Entity-Relationship Model — Toward a Unified View of Data. *ACM Trans. on Database Systems 1*, 1 (Mar., 1976), 9–36.

[Codd79] Codd, E.F. Extending the Database Relational Model to Capture More Meaning. *ACM Trans. on Database Systems 4*, 4 (Dec., 1979), 397–434.

[Hammer76] Hammer M. and McLeod D.J. A Framework for Data Base Semantic Integrity. In *Proc. 2nd Int. Conf. on Software Engineering, San Francisco, CA*, IEEE, 1976, pp. 498–504.

[Hammer81] Hammer M. and McLeod D.J. Database Description with SDM: A Semantic Data Model. *ACM Trans. on Database Systems 6*, 3 (Sept., 1981), 351–386.

[Hudson86] Hudson, S.E. and King, R. CACTIS: A database system for specifying functionally-defined data.. In *Proc. of the Workshop on Object-Oriented Databases*, IEEE, 1986.

[Hull87a] Hull, R. and King, R. Semantic Database Modeling: Survey, Applications and Research Issues. *ACM Computing Surveys 19*, 3 (Sept., 1987), 201–260.

[Hull87b] Hull, R., Four Views of Complex Objects: A Sophisticate's Introduction. In *Nested Relations and Complex Objects*, to appear in Springer-Verlag LNCS, 1988.

[KENT79] KENT, W. Limitations of record-based information models.. *Comm. ACM 4*, 1 (Mar., 1979), 107–131.

[KING84] KING, J. Sembase: A Semantic DBMS. In *Proc. Workshop on Expert Database Systems*, 1984, pp. 151–171.

[KHOSHAFIAN86] KHOSHAFIAN, S.N. AND COPELAND G.P. Object Identity. In *Proc. OOPSLA'86*, ACM, 1986, pp. 406–416.

[LENZERINI87] LENZERINI M. Covering and Disjointness in Type Networks. In *Proc. 2nd Int. Conf. on Data Engineering, Los Angeles, CA*, IEEE, 1987, pp. 386–393.

[LISKOV77] LISKOV, B.H. ET AL Abstraction Mechanisms in CLU. *Comm. of the ACM 20*, 8 (Aug., 1977), 564–576.

[MYLOP80] MYLOPOULOS, J., BERNSTEIN, P.A., AND WONG H.K.T. A Language Facility for Designing Database-Intensive Applications. *ACM Trans. on Database Systems 5*, 2 (June, 1980), 185–207.

[PECKHAM88] PECKHAM, J. AND MARYANSKI, F. Semantic Data Models. *ACM Computing Surveys 20*, 3 (Sept., 1988), 153–189.

[SHIPMAN81] SHIPMAN, D.W. The Functional Data Model and the Data Language DAPLEX. *ACM Trans. on Database Systems 6*, 1 (Mar., 1981), 140–173.

[SMITH77] SMITH, J.M. AND SMITH D.C.P. Database Abstractions: Aggregation and Generalization. *ACM Trans. on Database Systems 2*, 2 (June, 1977), 105–133.

[ULLMAN88] ULLMAN, J. D. *Principles of Database and Knowledge-Base Systems — Vol. 1*, Computer Science Press, 1988.