The Semantics of Parts Versus Aggregates in Data/Knowledge Modelling

Renate Motschnig-Pitrik

Department of Information Systems
University Vienna
Liebiggasse 4, 1010 Wien, Austria

Abstract. The incorporation of semantics into conceptual models has for long been a goal of the data/knowledge modelling communities. Equally, conceptual models strive for a high degree of intuitiveness in order be better understood by their human users. This paper aims to go one step in this direction by introducing the part-of relation as a special case of aggregation. To do so we investigate the semantic constraints accompanying this specialization and suggest different ways of incorporating part-of semantics into data/knowledge models. Further, it is demonstrated that, in analogy with IS-A relations, part-of relations form hierarchies (dag's) which constitute an important conceptual aid in understanding complex systems. Finally, we investigate the conditions under which the part-of relation exhibits transitive behavior which can be exploited for automated inferences facilitated by the transitivity property.
keywords: data/knowledge modelling, knowledge representation, conceptual modelling, semantic data models, part-of relations, object-oriented databases, aggregation, transitivity, inference;

"The whole is more than the sum of its parts"

1 Introduction

1.1 Motivation and Related Work

The overall objective of information systems (IS) modelling is to build models which confidently represent parts of the real world. The resulting need to model complex objects for advanced applications has lead to the development of a number of semantic and object-oriented models that attempt to capture more of the meaning as well as the structure and behavior of the data than traditional models [18, 30]. In this context it is important to appreciate why it is useful to incorporate abstractions and additional semantics into information system design methodologies, in particular in the early phases of the development process. Davis and Bonnell [11, p.85] argue that it is important to incorporate appropriate abstraction mechanisms that can be used to identify suitable categories with which to describe phenomena in the real world. This is motivated by the notion that much of what is perceived in the world is generally well-structured information and that a large problem in constructing complex systems capable of intelligent behavior is in clarifying these structures by using appropriate abstraction mechanisms. Furthermore [30], some of the benefits that generally have been identified as being associated with semantic models are: economy of expression, integrity maintenance, modelling flexibility, and simplifying querying.

While by far most research on extending IS design methodologies to capture more meaning has concentrated on the generalization/specialization abstraction and the accompanying mechanism of inheritance (e.g. [4, 22]), this research focuses on the *part-of* relation. The latter is deduced as a subcategory of the aggregation abstraction and enriched with additional semantics. *Aggregation* has been defined as an abstraction in which a relationship between objects is considered as a higher level (aggregate) object [35]. When considering the aggregate, details of the constituent objects are suppressed. An example of aggregation is depicted in figure 1-1a. Note, that the aggregate object (e.g. room) consists of a number of arbitrary constituent objects and/or attributes. This reflects the situation most often encountered in current conceptual modelling and knowledge representation techniques. In particular, it is hardly ever distinguished between parts and other constituents making up some entity in the domain of discourse. In short, in data/knowledge modelling the part-of relation has not (yet) been devoted special attention.

Inspired by research on parts conducted in cognitive psychology [37], we suggest to see a conceptual entity to consist of parts and, in addition, other attributes. At this point there is a strong temptation to wonder whether the distinction is not superfluous, since, attributes **are** parts that constitute the description of some whole. In which way, then, should there exist attributes which are **not** parts? The answer is easy, but only if care is taken to distinguish between real-world concepts and their representation as data structures in some formal notation used to model these concepts: Whereas, obviously, all attributes characterizing some concept are parts of the data structure modelling this concept, not all attributes are parts of the concept as such. Consider, for example, a class (or any similar structure) used to model a room. While ceiling, floor, walls, window, door are parts of the real world concept of a room (compare figure 1-1b), owner, size, and location definitely are important constituents (modelled e.g. as attributes), but they are in no case parts of rooms. Nevertheless, owner, size, and location are part of the class representation of rooms.

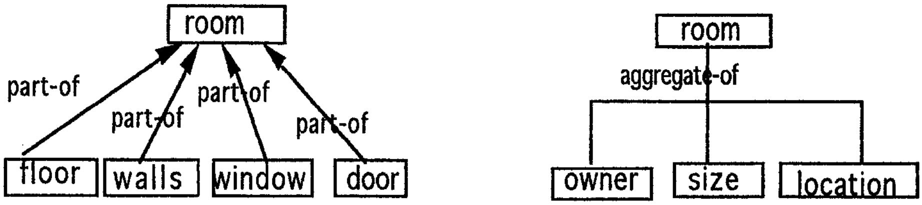


Fig. 1-1:
a) The aggregate `room` as an example of aggregation   b) The composite `room` demonstrating the part-of relation

In this paper it is argued that the *part-of* relation carries specific semantics which can be exploited to enrich conceptual models and to provide for specific inferences. Furthermore, repeated application of the part-of relation results in a hierarchy (more precisely a dag), hereafter referred to as *partonomy*. Partonomies organize concepts in terms of connections between parts or components and wholes or composites.

Within the field of computer science, partonomies play an essential role for modelling systems decomposition in almost all traditional software development methodologies, such as SADT [33] or JSD [19] and, more recently, object-oriented (OO) analysis and design (e.g. [2, 23]). Nevertheless, despite the importance of managing systems decomposed into several thousands of parts and subparts, etc. [39], the specific semantics of part-of relations have hardly ever been investigated in the context of software engineering. In the realm of AI, the object-oriented programming language LOOPS [1] pioneers in providing linguistic means to model composite objects. The mechanisms for processing parts, however, are 'hardwired' into the language with almost no support for incorporating additional semantics.

In particular, recent techniques for OO analysis and design have rediscovered the importance of aggregation. Some techniques (such as e.g. [9, 13, 23]) even offer specific constructs to model aggregation which often is refered to as the part-of relation. No specific semantics, however, are defined for such parts. Moreover, the examples given in the respective documentations (except [23]) indicate that the general case of aggregation is meant. In comparing OO analysis methods, De Champeaux and Faure [12] nicely capture this situation and its ramifications by arguing that 'the notion of the part-of relationship is problematic when its semantics are not clarified. For instance, it is often unstated whether part-of is assumed to be transitive and what its behavioral ramifications are. Should the destruction of A imply the destruction of B when B is part of A? Can an object be a subpart of more than one superpart? ... As a result of the ambiguities surrounding part-of, an analyst should make explicit the intended semantics before using part-of.'

The situation is different in the context of object-oriented databases (OODBs), most prominently ORION [20], where partonomies have been suggested as a means of modelling composite objects and their semantics. Besides discussing the role part-of relations play in schema evolution, Kim et al. go as far as proposing parts as a basis for authorization and locking. The approach taken here is different. While drawing on Kim et al.'s results regarding the semantics of parts (c. f. section 3.1), the emphasis is on a broader discussion of part-of semantics in the context of conceptual modelling, along with the impacts, such as inferences facilitated by the distinction of parts from other attributes and by transitivity. This discussion is motivated by the endeavor to exploit the semantics inherent in part-of relations in order to allow to build models that more closely match human conceptualization. Such models, we claim, are easier to understand and to reason about.

The goals of this research are to

* examine the representational power of partonomies in their role in data/knowledge modelling,
* justify the importance of the part-of relationship from a cognitive point of view,
* introduce semantics of the part-of relationship and show how to incorporate them into data/knowledge models,
* open room for inference mechanisms based on the transitivity of the part-of relation.

Before giving a more detailed account on the points mentioned, let us view the part-of relation as a specific case of a semantic relation and consider its representation in conceptual models. Graphical displays of entities and relationships, such as entity-relationship (ER) diagrams, have been popular since almost two decades for providing high level representations of some parts of the real-world [10]. Their immense usefulness and conceptual aid for database design has been proven in many applications. ER diagrams have also been extended (EER, [36]) to provide specific constructs for modelling generalization/specialization relationships. The (E)ER notation serves to provide a fairly complete picture of the entities and relationships underlying some model. Thus it often coexists with object browsers [15] which were invented by the object oriented programming (OOP) community as a complement to more complete representations such as (E)ER diagrams. Object browsers were designed to display objects (corresponding to entities) with only one or two kinds of relationship, namely subclass (or specialization) and instance-of relations. Specifically, they were not designed to display the part-of structure of objects which, as will be argued shortly, is a further fundamental organizational principle. Therefore, complementing object browsers to depict part-of relations should provide more complete cognitive maps which facilitate the understandability of the underlying models. In any case, the need to visualize complex objects situated at different levels of a partonomy has been recognized in the area of CAD applications. In this context, Udagawa designed an elaborate algorithm for browsing composite objects [38].

1.2 Terminology

Before moving on, let us agree upon the meaning of some fundamental terms which have been used across various fields in computer science with different semantics. Talking about an *object* we mean a symbolic structure denoting some conceptual entity. Each object has a unique object identification (oid), a name, and consists of an aggregation of any number of attributes which collectively characterize the object. Furthermore, an object may be associated additional semantics in which way so ever.

Attributes can either be structural, in which case they are also referred to as semantic relations, or behavioral, most prominently methods. We suggest to distinguish between two categories of semantics relations (see also figure 1-2): ordinary *reference relations*[1], such as works-for or uses, and relations which serve as basic *organizational principles*, such as 'is-a' [6, 4, 3], 'instance-of' [18, 27], 'part-of' [40], and others [25]. To do so is not new! It is just an extension to further organizational concepts, in particular to part-of relations. Strongly simplified, an organizational principle (also called a structuring concept) must be an abstraction useful for understanding the organization or structure of complex systems and abstract from application specific relations among the entities represented in some model.

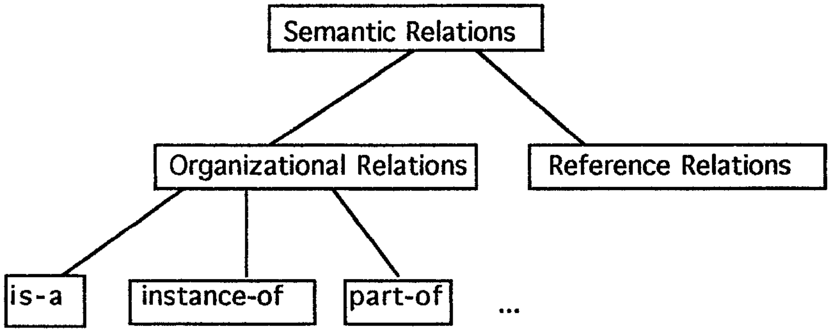


Fig. 1-2: Fragment of a taxonomy of semantic relations

---

[1] Sometimes, for example in the ER model, attributes having elementary ranges, such as age: 1..100, are distinguished from relations, such that only the former are called attributes. We do not draw this distinction and use the term attribute in a generic sense, since the distinction is not relevant in our context.

1.3 Overview of the Paper

The incentive to superimpose part-of semantics on aggregation [35], comes from two perspectives: cognitive psychology and data/knowledge modelling[2]. The psychologically based arguments stem from experiments assessing the importance of attributes being parts: subjects tend to describe concepts primarily in terms of parts. More on this can be found in section 2.1. The incentives from the modelling disciplines center around the incorporation of semantics being peculiar to part-of relations as well as on the exploitation of the transitivity property of the part-of relation. Additional semantics allow to deduce integrity constraints useful, amongst others, in the case of updates and schema evolution. Transitivity, on the other hand, can be applied for drawing inferences of the kind: A part-of B and B part-of C imply A part-of C.

The paper is organized as follows. The next section is aimed to provide cognitive evidence on the role parts play in the human thought process. It thus serves as a justification to consider the part-of relationship an organizational principle. This applies in particular to notations supporting early phases of development since such notations should be oriented toward the human user [29]. Section three explores various issues resulting from distinguishing part-of relations from others taking a computer science oriented perspective. In particular, cardinality constraints and update semantics regarding subcategories of part-of relations are studied. Furthermore, the extensible CM and KR language Telos [28, 21] is used to exemplify the incorporation of part-of semantics into a KR notation. Section four centers on transitivity. The taxonomy of semantic relations suggested in [40] is used to guide a discussion on the distinction of various subcategories of part-of relations with respect to transitivity. In this context, a new break-down into subcategories is suggested, which is based on a more computer science oriented perspective and which establishes transitivity within a core of part-of relations. A summary and indication of issues for further research round up the paper.

2 The Cognitive Perspective on Parts and Composites

2.1 The Role of Parts in the Human Thought Process

There is strong psychological evidence that part-of relations associating parts or components with wholes or composites are one of the most important structuring concepts underlying the organization of human knowledge. In a controlled experiment conducted by Tversky ([37] based on work reported in [32]) subjects were asked to list attributes of both artificial and biological kinds of objects. The attributes obtained were partitioned into two categories: attributes being parts and not-part attributes. To distinguish between these two categories of attributes two criteria were used. The first is a dictionary definition resulting from the consultation of several dictionaries: "A part is one of the segments or portions into which something is regarded as divided; a part is less than a whole; together, parts

---

[2] Since the concepts presented in this paper equally apply to conceptual modelling as well as to knowledge representation, we use the term data/knowledge modelling to subsume the two fields. For a thorough and thoughtful distinction between CM [Brodie84] and KR [Mylopoulos90b] consult [Borgida91].

constitute a whole". The second goes back to Miller and Johnson-Laird [24] who argue that a part-of relation is often expressed in a has-a sentence frame in a similar way as is-a sentences often indicate taxonomic relations.

The most interesting result in the context of this paper is that, in general, part attributes had a significant share: 58% of all attributes listed for artificial objects and 42,7% of attributes ascribed to biological categories were parts. The prevalence of part terms was most significant at the so called basic level of genericity (exemplified by concepts such as bird or table). This can be explained by the fact that objects at that level mainly differ with respect to parts and that parts are associated with--and hence represent--different functionalities. Consider for example a chair consisting of a seat which serves for sitting, a back serving for leaning back, and of legs which serve for moving and support. Consequently, it are the parts of an object that are intimately related to the objects' behavior and that are considered perceptually most salient and functionally most significant.

Interpreting the prevalence of part attributes as shown in Tversky's experiment we conclude that the part-of relation is an important abstraction underlying the organization of human knowledge. The consequence from the above for data/knowledge modelling is straightforward. If the part-of relation has been proved to underlie our internal representation, it will be useful to embody it into formal representations which aim to support understandability. In particular, the close relation between the parts of an object and aspects of this objects behavior underline the importance of parts in OO approaches, which characterize objects via their structure as well as behavior. A complementary issue is to which degree an organizational principle such as the part-of relation proves useful for the representation and reasoning in formal models. This aspect will be subject to investigation in the next section after discussing the relevance of our findings when applied to graphical representations.

2.2 The Representational Account of part-of Relations

In the following let us approach the representational impact of part-of relations from a pragmatic side and, for this purpose, consider IS design notations which have been proved to be useful tools for conceptualization. Perhaps the most broadly used high level graphic notation is the ER diagram [10]. It confidently represents entities (concepts) and relationships holding between these concepts. While it is perfectly helpful in modelling small systems, it has been observed that (the original) ER diagrams lack abstraction mechanisms to make them useful to represent the structure of very large systems. In this respect, concept taxonomies and browsers have been appreciated. In a similar vein, the part-of relation could be used to extract partonomies to be supported by browsers on the part-of relation.

To demonstrate the situation, figures 2-1 and 2-2 show, respectively, an ER diagram and a partonomy extracted from the ER diagram and complemented by further parts. Figure 2-2 displays the partonomy while abstracting from ordinary reference relations. The figures serve to provide a juxtaposition of the semantic relations of aggregation versus part-of. The reader will agree that figure 2-2 is less informative than figure 2-1 but that it shows the part structure of the underlying application much more transparently. Not to be misunderstood, the foregoing discussion in no case should be interpreted as an argument against aggregation which, by the way, is the abstraction underlying the very essential construct of a chunk or module. The juxtaposition should just serve to demonstrate the complementarity of aggregation and part-whole abstraction and to help to argue in favour of viewing partonomics as a useful complement or specialization of aggregates. Even more importantly, the two figures should help to show the importance of distinguishing aggregation from part-whole abstraction. Whereas the former is well recognized across different fields of computer science, the latter has been overshadowed by aggregation, except for a few exceptions. These stem most notably from OODB's [20], CAD applications and, to a certain degree, from the modelling of complex objects in AI programming languages such as LOOPS [1].

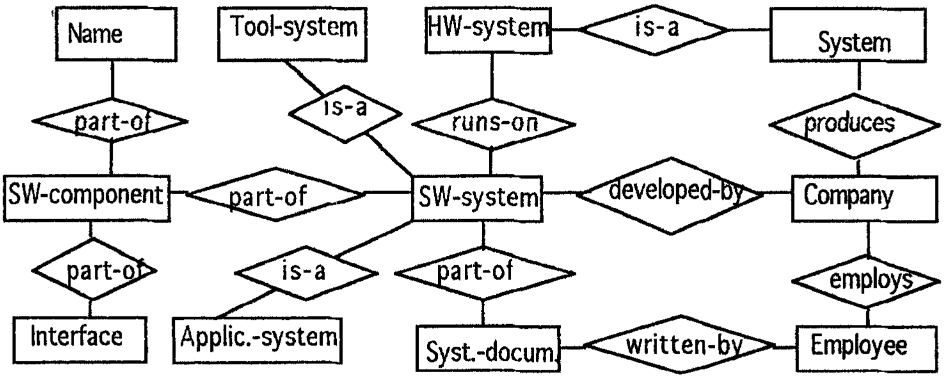


Fig. 2-1: ER diagram displaying entities and relationships in the context of a software system

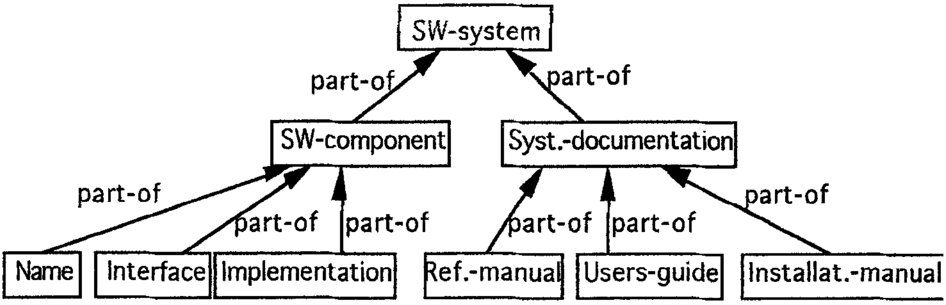


Fig. 2-2: Partonomy of a software system

Summarizing, we suggest that the part-of relation deserves specific support in notations for data/knowledge modelling, in addition to the support of the aggregation abstraction. This is due to the cognitive salience of parts as well as to the representational benefits resulting from partonomies. Further note, that missing support of parts has been experienced and documented to be a serious drawback in the use of hypertext systems such as Notecards [16].

3 Implications for Conceptual Modelling and Knowledge Representation

3.1 Exclusive Versus Shared and Dependent Versus Independent Parts

As computer scientists we are most prominently interested in the feasibility and in the benefits of incorporating semantic concepts into formal models. Therefore, this section is concerned with the question which additional semantics can be attributed to part-whole relations, how these semantics can be incorporated into modelling languages, and what implications this has with respect to updates of the knowledge/database and with respect to the support of inferences.

Subsection 3.1 almost completely borrows from [20]. In particular, the extremely useful--as we believe--categorization of parts into exclusive versus shared and dependent versus independent ones has first been suggested by Kim et al. [20] in the context of modelling composite objects in the OODB ORION. The underlying mechanisms, though, will be summarized in the following in order to make the paper self-contained. Also, to allow for the discussion of consequences, such as the implications on the formulation of cardinality constraints and on inferences facilitated by exploiting transitivity.

Kim et al. observed that while it is common for objects in OODB's to reference any number of other objects, no specific semantics are captured by such reference links. The authors therefore suggest to superimpose the is-part-of relation on nested objects such, that an object may be part of another object. A set of component objects which form a single conceptual entity (a whole) is then referred to as composite object and the links connecting the components with this object are called composite (or, in our terminology part-of) links. Importantly, the model allows to specify for each composite link whether the reference is *exclusive*, i.e. the component exclusively belongs to the composite, or *shared*, meaning that the component may possibly be part-of several composites. Further, a part-of link can be defined to be either *dependent*, which means that the existence of the component depends on the existence of the composite, or *independent*, i.e. having existence irrespectively of the composite. On the whole, four types of composites result from combining the two features.

Consider, for example the reference in figure part-of paper. This reference should be modeled as shared and independent in order to obtain the semantics that each figure may appear in more that one paper and may exist independently of any paper. (This can be implemented, for example, by keeping figures on a separate file.) The situation is different with the semantics of part-of in a sentence such as brain part-of person, in which brain would be characterized as exclusive and dependent. This is because one brain cannot be part of more than one person and its lifetime depends on the lifetime of the person. As another example imagine a situation as expressed in engine part-of car. In this case we may want to model the part-of link as exclusive but independent, to achieve the semantics that, at one point in time, an engine can be part-of at most one car and can exist independently of any car.

Kim et al. formalize the semantics of the different types of part-of references in terms of constraints which must hold if objects are created or deleted. To provide an example, consider the deletion of an object O':

If there exists a dependent and exclusive part-of reference from O' to another object O (i.e., O part-of O'), then it holds that the deletion of O' implies the deletion of O. In case that the reference is independent, however, the deletion of O' does not imply the deletion of O.

In a similar vein, specific conditions on some attribute A must hold, if one wishes to make an object O part-of O' through this attribute. Furthermore, the semantics of composite objects are formalized by stating *topology rules*, such as the following:

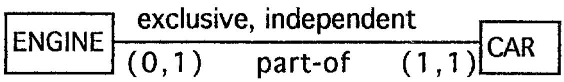
If an object O has an independent exclusive part-of reference to it, then it cannot have a dependent exclusive reference from another object, and vice versa.

The syntax for attribute specification (within class definitions) used in ORION to support the full semantics of part-of links is the following:

syntax for attribute definition:		*example: class Paper with attribute figures:*	
(attributeName	[:init InitialValue]	figures	:init ...
	.		.
	.		.
	[:composite TrueOrNil]		:composite true
	[:exclusive TrueOrNil]		:exclusive nil
	[:dependent TrueOrNil]		:dependent nil

3.2 Cardinality Constraints Associated with part-of Links

In conceptual knowledge/data modelling it is appreciated to annotate relationships (links, slots) with cardinality bounds.Thus figure 3-1 will be interpreted as follows:



(a)

```
CLASS Car
        memberSlot
                engine:EngineType
                        cardMin: 1
                        cardMax: 1
```

(b)

Fig. 3-1: Cardinality constraints expressed in an ER diagram- (a) and a frame based notation (b)

Each engine is part-of at least zero and, at one point in time, at most one car. Similarly, a (ordinary) car has, at one point in time, at least one and at most one (i.e. exactly one) engine. In a frame based, object-oriented notation the former constraint is expressed as shown in figure 3-1b.

Minimality and maximality constraints on the various types of part-of relations can easily be derived from the topology rules and constraints given in [20]. Figure 3-2 depicts the results and furthermore provides examples for each type (combination of features) of part-of relation. While the first and last example in figure 3-2 have already been described above, the second and third phrase remain to be explained. The component paper, in paper part-of journal is specified to be exclusive and independent. This is because we do not want the same paper to appear in more than one journal and wish to grant to each paper an existence which is independent of that of the journal containing the paper. Next, consider the phrase subprogram part-of program-library. In this case the desired semantics are such that on the one hand the subprogram shall be allowed to be shared (reused) among several software products but, on the other hand, should cease to exist upon the deletion of the program-library, given the subprogram is no longer referenced as being part of some software product. Hence we assign the categories shared and dependent. In all the examples note the distinction between the type- and the token (instance) level: all the semantic constraints associated with individual categories of part-of relations are given at the type (or class) level, such that they apply to all objects (tokens) being instances of the corresponding types (classes).

From figure 3-2 it follows that part-of links which are exclusive imply a one as the maximum cardinality on the component object (left hand) side, while, obviously, shared part-of links allow for a maximum cardinality of arbitrary many ('*') on the left hand side. Similarly, dependent part-of links call for a one as minimum cardinality since, by definition, the object they are part-of must exist, whereas the minimum cardinality of independent part-of links is zero. Note, that no cardinality constraints can be deduced for the inverse relation has-part (compare the irregularity of cardinality bounds on the right hand side). This is because, in general, one cannot deduce that some composite object O has at most one component C of one type from the fact that C is an exclusive component of O. If, for example, one engine (at one point in time) can be part of one car only, one might well envisage a car having a second (e.g. spare) engine as its part. Hence the minimum and maximum cardinalities on the right hand sides are intended as examples only, with no claim for generality. Note, however, that also the inverse relationship to part-of, namely *has-part*, can be semantically enriched by using the features exclusive/shared and dependent/independent.

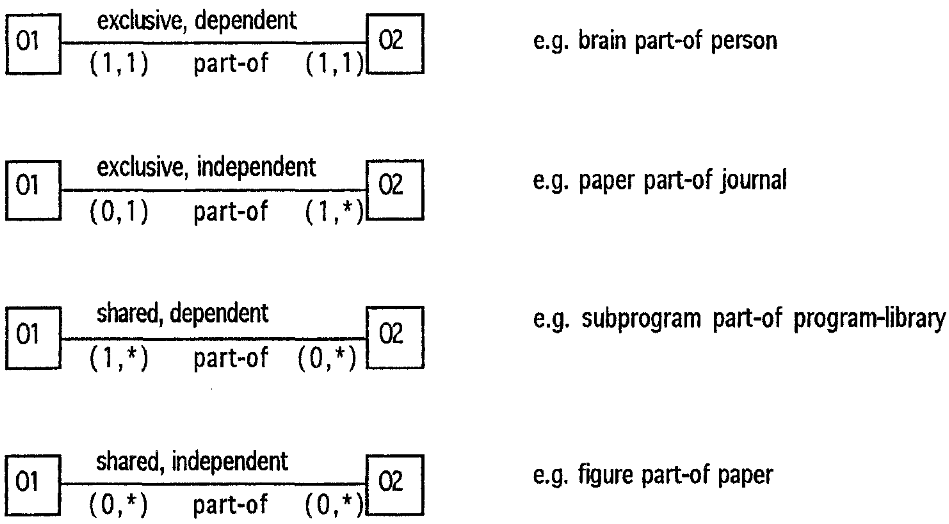


Fig. 3-2: Cardinality constraints depending on the type of part-of relation plus examples

3.3 The Knowledge Representation Perspective

Surprisingly, frame based and object-oriented KR languages, in general, do not support part-of relations, although part-of links used to play an important role in semantic networks (see, for example [5]. Most probably the disappearance of part-of links is attributable to the fact that they were generalized to become slots, which encode all kinds of references including references to parts. While the aggregation of slots to describe concepts or objects has proven to be an extremely useful mechanism, part-of semantics, in general, are not captured in such descriptions. To reconcile the notion of slots (or attributes) with that of additional part-of semantics, let us discuss in which way a slot specification can be extended to capture part-of semantics. Note, that KR languages designed for extensibility are particularly well suited for such an enterprise. For this reason we chose Telos [28] to demonstrate the necessary build-ons. In the sequel, only a simplified version of Telos will be used. Also, only the features necessary to understand the incorporation of part-of semantics will be described. For further details on Telos and on a formal account of its syntax and semantics consult [21, 28].

Telos explicitly supports the following three structuring concepts (compare also figure 3-4):

classification via keyword IN followed by list of classes
generalization via keyword ISA followed by list of classes
aggregation via keyword WITH followed by a list of attributes grouped into attribute categories;

Specifically, the WITH clause serves to associate not only properties but also assertions to classes. One distinguishing feature of Telos is its treatment of attributes as first-class citizens, i.e. objects which have their own identifiers and can be organized along any of the structuring dimensions mentioned. The attribute mechanism of Telos, in particular the handling of attributes as full objects and the capability of defining assertions as attributes, combined with a stratified multi-level classification lattice allow Telos to be adapted to new modelling needs [28, 27]. In the sequel, this will be demonstrated by using Telos to capture the semantics of attributes being parts belonging to any of the four types specified in section 3.1.

To accomplish this, we first specify three attribute metaclasses, namely Part, ExclusivePart and DependentPart, which, respectively, capture the semantics of the various types of parts as depicted in figure 3-3a-c. Interpreting the figure in some more detail, the attribute metaclass Part as such does not include any integrity constraints, reflecting the semantics of shared and independent parts. Nevertheless, the Part metaclass plays an important role in allowing to distinguish and to group part-attributes. Moreover, it serves as the common parent of ExclusivePart and DependentPart in the ISA hierarchy. The integrity constraint in ExclusivePart (compare figure 3-3b) states that, at any given time, an object O being an exclusive part of a composite object O' cannot be part of yet another composite object. (In Telos terms this is expressed such that part-of links having the same destination--the part, denoted by to(x)--must have the same source--from(x).) Similarly, the integrity constraint in DependentPart says that the lifetime of a dependent part, say O, ceases if the lifetime of the composite on which O depends comes to an end, and O is no longer referenced as being part of any other composite object.

As can be seen from the figure, attribute classes strongly resemble ordinary classes. The main difference is that the former more heavily rely on assertions to formulate constraints instead on attributes specifying relationships. In order to associate the semantics defined in attribute metaclasses with an "ordinary" object's attributes, the metaclass of one's choice is instantiated to become an attribute class or, synonymously, category in some class C. Further, an attribute of one's choice is associated with the corresponding attribute category by listing it under that category (compare figure 3-4). As an example consider the attribute figures listed under the category part in figure 3-4. As another example consider the attribute footnotes, which is associated three categories part, exclusive, and dependent, in order to express that footnotes exclusively belong to one paper and cease to be of any interest when the paper is no longer stored. The latter example demonstrates in which way attributes may be associated with more than one category. In this case the constraints stemming from the corresponding attribute metaclasses listed in the header are combined.

Note that, using attribute categories, a part attribute is by default shared and independent such that additional constraints can be imposed on it by listing further attribute categories. Note further, that attribute metaclasses in Telos are highly modular and reusable and not particularly tied to part-of semantics. One could easily envisage to specify objects which depend on other objects and yet are not their parts.

So far we have argued that part-of semantics can be incorporated into KR languages via the specification of attribute metaclasses, as in Telos. But also languages with no support of attribute classes can be extended to capture part-of semantics if they provide for (user definable) annotations. *Annotations* are associated with slots and serve to capture some of the semantics of the slot to which they are associated. Typical examples of built-in annotations are cardinality or type constraints on slot fillers (see, for example, figure 3-1b) [14]. Since most of the more advanced KR languages allow users to specify annotations (most often using some host language such as LISP), part-of semantics can be captured by defining annotations such as part, exclusive, and dependent. Slots then are annotated with any combination of the three annotations in a similar way as attributes are associated attribute classes. Note, however, that the annotation approach does not encourage the textual grouping of slots carrying the same semantics as has been the case with attribute classes.

```
TELL CLASS Part /* shared, independent parts */
        IN M1_Class, AttributeClass
END

TELL CLASS ExclusivePart
        IN M1_Class, AttributeClass
        ISA Part
        WITH
          integrityConstraint
              exclConstr: $ (Forall x, y/Attribute) (Forall z, z'/AttributeClass)
              (Forall t/TimeInterval)
              ( (z in ExclusivePart) and (z' in Part) and (x in z) and (y in z') and
              to(x) = to(y) ==> from(x) = from(y) )[over t] $
END

TELL CLASS DependentPart
        IN M1_Class, AttributeClass
        ISA Part
        WITH
          integrityConstraint
              dpdtConstr: $ (Forall x/Attribute) (Forall z, z'/AttributeClass) (Exists
                      y/Attribute)
              (Forall t/TimeInterval)
              ( ( (z in DependentPart) and (x in z) ==> [when(to(x)) during when(from(x))] )
                      or
              ( (z in Part) and (y in z') and to(x) = to (y) and not (from(x) = from(y)) ) )
              [ over t] $
END
```

Fig. 3-3a-c: Specification of part-semantics via attribute metaclasses in Telos.

```
TELL CLASS Paper
        IN S_Class        /* Simple Class */
        ISA Document
        WITH
          attribute
                  author: Name
          part    /* i.e. shared and independent part */
                  figures: Image
          part, exclusivePart, dependentPart
                  footnotes: String

END Paper
```

Fig. 3-4: Example showing the association of various attribute categories to attributes

3.4 Further Categories of Parts and their Semantics

An analogous approach as that described in the previous subsections can be taken to define yet further categories of part-of relationships and associate with them specific samantics. For example, a part can be *optional*, such as 'mouse part-of PC' or *essential*, such as 'processor part-of PC' carrying the following semantics. Whereas the deletion of an optional part will leave the corresponding composite unaffected, the deletion of an essential part will cause the deletion of 'its' composite.

Although all examples of parts given so far have concerned data entities, the notion of parts equally applies to activities, which are often encountered as separate modelling primitives in requirements modelling techniques. Considering activities, parts can be conceived as partial activities or, in other words, as *phases* of some superordinate activity. Additional semantics can be associated with phases by formalizing the common sense fact that each phase must take place within a time interval which is fully contained in the time interval of the superordinate activity. A sample specification of phase-semantics in Telos is given in figure 3-5. A further specialization of part semantics can be undertaken by defining special phases, such as *initial*, *intermediate* and *final* ones: an initial phase of an activity A is defined to costart with A, an intermediate phase must start later and terminate sooner than A, and a final phase must end simultaneously with A. As an example for the application of phase categories consider a (meta)model of the software process. Such a model could declaratively describe that, at a gross level, software development is split into three phases: an initial phase of requirements analysis, an intermediate phase of design, and a final phase of coding.

```
TELL CLASS Phase
        IN M1_Class, AttributeClass
        ISA Part
        WITH
        integrityConstraint
                phaseConstr: $ (Forall a1,a2/Activity) (Forall x/Attribute) (Forall
z/AttributeClass)
                (Forall t1, t2/TimeInterval)
                ( (z in Phase) and (x in z) and (from(x) = a1) and (to(x) = a2) and
                [t1 during t2] and [from(x) during t1] ) ==> [to(x) over t2] $
END
```

Fig. 3-5: Specification of the semantics of phases in Telos.

4 Transitivity

KR languages strive for powerful inference mechanisms allowing them to deduce new knowledge from existing one without human intervention. One familiar source of inference is the law of transitivity which holds for example for is-a relations and largely contributes to the power of is-a inheritance. Inheritance, in general, does not hold for part-of relations: the attribute numberOfWheels attached to the class Car does not make sense to be attached to the Class Engine which is part-of Car. Nevertheless, intuitively, we expect the part-of relation to be transitive. Knowing, for example, that a processor is part-of a PC and a PC is part of a computer system, it seems plausible to conclude that a processor is part of a computer system. Nevertheless, on investigating part-of relationships Winston et al. [40] have found that it is not in all cases that the part-of relation is transitive. Consider, for example, the syllogism:

the conductor's arm is part-of the conductor,
the conductor is part-of the orchestra,
the conductor's arm is part-of the orchestra.

This transitive combination sounds very odd at best! The strange behavior around part-of transitivity has led Winston et al. to systematically investigate the transitivity of part-of relations. The authors found out that the part-of relation can be partitioned into six semantic categories, which are summarized and demonstrated by examples as follows:

	category of part-of relation:	*example:*
*	component/object	processor part-of computer
*	member/collection	conductor part-of orchestra
*	portion/mass	slice part-of pie
*	stuff/object	steel part-of bike
*	feature/activity	spoon part-of eating, or swallowing part-of eating
*	place/area	Toronto part-of Ontario

It is argued that transitivity always holds when semantic relations of the same category are combined. The authors further present examples to show that the combination of part-of relations stemming from different semantic categories leads to unsensical or at least highly questionable results. Critical examination of these examples, however, leads us to propose that only some of them seem to prove the intransitivity of the part-of relation (in the case that the two constituents of the premise each stem from different categories). Other examples, conversely, at worst sound a bit strange but in no case wrong. This is because, as we conjecture, in natural language one would use a more specialized and hence more suitable term instead of using part-of. As an example consider one of the examples classified as a failure of transitivity in [40]:

The refrigerator is part-of the kitchen,
the kitchen is part-of the house,
--> ? the refrigerator is part-of the house.

If explained as: the refrigerator is part of the equipment of the house, or, in the context of selling the house and leaving the refrigerator in it such that the fridge constitutes one item in assessing the composite value of the house there does not seem anything wrong or even strange about the transitive conclusion.

It is only fair to mention that Winston's taxonomy of part-of relations has been designed not to ensure transitivity, but,in the first place, to distinguish between semantic subcategories of part-of relations on the basis of similarity. Thus relations belonging to the same subcategory are more similar in terms of sharing values for three features called *relational elements*. More precisely, two part-of relations belong to the same category if and only if they share values for the relational elements functionality, separability and homogeneity [40]. Storey [34] gives excellent account on how the similarity among part-of relations belonging to the same subcategory can be exploited in conceptual database design to add semantics to the design and hence to build models which more faithfully capture the intended subject matter. Note, however, that, in general, cardinality constraints do not automatically follow from membership in some subcategory, as we have observed to be the case with distinguishing exclusive/shared and dependent/independent types of part-of relations. We conclude that Winston's subcategories and Kim's features provide complementary means to enrich data/knowledge modelling by supplying additional semantics of part-of relations.

Keeping transitivity in mind, a closer look--and, admittedly, some conceptual modelling bias--at the examples and results in [40] leads us to suggest a different partitioning of part-of relations on top of that proposed by Winston et al. . The strategy thereby is to group those part-of relation categories which, when combined by transitivity, lead to acceptable results, while separating those which lead to erroneous implications. The reader will see soon that, interestingly, the results of this separation process are completely in line with the more computer science based conceptual modelling perspective. The following is intended to document the process of extracting those semantic relations from the taxonomy of Winston et al. which impede transitivity across subcategories.

Many semantic models such as SDM [17] or ACM/PCM [7] provide a specific type constructor, called *grouping* or *association*, to model the relationship between members and a collection. Thus, in semantic modelling, *member/collection* relationships are seen to constitute a specific structuring concept rather than a subcategory of part-of relations, as suggested in [40]. In fact, we observed that transitive combinations including exactly one of the premises from the *member/collection* category are those that sound worst (or most funny, if you like). The reader will remember the example of this combination given with the conductor's arm being part of the orchestra. Clearly, the conductor in more appropriately a member of the orchestra, hence the odd conclusion above.

The situation is somewhat different with object/stuff relations, which are the next candidates to be eliminated as a subcategory from a more restricted the part-of taxonomy, since they destroy transitivity. *Object/stuff* relations, we suggest, shall be dealt with in a way akin to other special purpose relations such as works for in employee works for company. They seem to be situated on the very end of part-of relations anyway since the specific purpose natural language term made-of fits much better than part-of to describe the situation. A bike is clearly made of aluminium or made of steel rather than aluminium being part-of the bike according to the definition of parts given in section 2.1. Furthermore, the substitutability of part-of by the verb phrase made-of is a simple criterion to sort out object/stuff relations from what we are going to call the *core part-of* relations. Finally, a minor constraint on the feature/activity category is necessary to ascertain the transitivity of the remaining core part-of relations. We constrain the feature constituent in the *feature/activity* category to mean only a phase or a subactivity, such as starting is part-of driving (phase), or, breathing is part-of jogging. Specifically, phrases such as running-shoes are part-of jogging, or, a spoon is part-of eating, do not qualify as proper part-of relationships, since shoes or a spoon are neither phases nor subactivities!

Excluding situations as the above from the core part-of relations does little harm, since, firstly, they are easy to be distinguished and, secondly, they can be modelled as any other special purpose reference relation. Again, we give a comparison with natural language: one is more inclined to say, for instance, running-shoes are used for jogging and a spoon is used for eating than to consider the instruments part of the activity itself. Constraining the feature to be a phase or subactivity, or--in syntactic terms--a verb, is particularly useful for assuring transitivity since it implicitly constraints the legal combinations of premises to imply conclusions via transitivity: This is because activities are necessarily verb phrases, as are phases and subactivities. Consequently, it is on syntactic grounds already that the constrained feature/activity category, also referred to as *verb-feature/activity* category, is separated from all the other categories which happen to relate noun constituents.

In brief, if the semantic categories member/collection, stuff/object, and noun-feature/activity are modelled by means other than part semantics, i.e. by reference relations, we have observed that the remaining part-of relation categories, making up the core part-of relations, exhibit transitive behavior if combined in any arbitrary way. Explicitly, the following categories are defined as belonging to the *core part-of relations*:

* component/object
* portion/mass
* verb-feature/activity
* place/area.

In order to argue in favor of the transitivity of the core part-of relations two issues remain to be shown, namely that the verb-feature/activity category is transitive in itself (since it can't be combined with the other categories on syntactic grounds) and that any two-place variation of the remaining three categories from the core part-of relations is transitive. Although we are fully aware of the fact that positive examples can, at their best, test, but never verify a hypothesis, the interested reader is referred to the appendix for examples of all variations. These examples seem to indicate the transitivity of the core part-of relations. In any case, further empirical investigation is necessary to gain confidence in the preliminary results presented above.

Summarizing, the considerations in gaining or preserving transitivity have led us to define core part-of relations. These constitute a more constrained class of part-of relations than the join of the six subcategories as proposed by [40]. In this context we have argued that inferences resulting from transitivity can safely be drawn within the core part-of relations. In particular, we have shown that member/collection relationships are not transitive and hence should be dealt with separately from part of relations. This conclusion smoothly fits the CM perspective, which traditionally has suggested the concept of grouping or association to model member/collection relations.

5 Summary and Issues for Further Research

We have argued on the prominent role parts play in human cognition and pointed to the advantages of providing (formal) representations with a high degree of correspondence with (natural) cognitive maps. Hence, we have investigated the idea of distinguishing parts from other attributes in the field of data/knowledge modelling. In particular, three main benefits were identified to result from the distinction of parts from other attributes:

* the incorporation of additional semantics leads to models that more closely match (some aspects of) the real world;

* the representation of partonomies provides a conceptual aid since partonomies can be seen as partial cognitive maps, much in the flavor of taxonomies but orthogonal to them;

* the exploitation of the transitivity property of part-of relations gives room for powerful automated inferences.

In order to realize these benefits, we have investigated various ways of incorporating the semantics of parts into data/knowledge models. Thereby extensible languages were found to be particularly well suited for this purpose. Although we do not doubt that the specific role parts play in human cognition should be given account in artificial representations, future research is still necessary to

* confirm or to adjust our findings on transitivity;

* to find more precise and yet straightforward (easy to apply) criteria on what parts are, in areas such as social systems, for example organizations, or social events, such as the organization of a conference. In these systems the distinction of parts is by far less obvious than e.g. in CAD models or CASE applications (compare also figure 2-2).

Further, following the research presented in [37], it would be worthwhile to investigate object-oriented analysis design methodologies (for example [2]) which distinguish a base level of classification to contaminate part attributes at the cost of super- and subordinate levels to concentrate of functional features and specializations of parts, respectively. Finally, our research is directed towards the investigation of further organizational principles [26], such as perspectives, in order to examine their cognitive and representational account as well as to determine their semantic properties.

Acknowledgements: Special thanks to Alex Borgida, John Mylopoulos, and Veda Storey for their insightful discussions and comments on an earlier draft of this paper.

References

1. Bobrow, D.G., Stefik, M.J., "The LOOPS Manual"; Xerox Corporation, Palo Alto, December 1983.

2. Booch G.: "Object-Oriented Design with Applications"; Benjamin/Cummings, 1991.

3. Borgida A.: "Knowledge Representation, Semantic Modeling: Similarities and Differences"; H. Kangassalo (editor), Entity-Relation Approach: The Core of Conceptual Modelling, Elsevier Science Publishers B. V. (North Holland), 1991.

4. Borgida A., Mylopoulos J., Wong H.K.T.: "Generalization/Specialization as a Basis for Software Specification"; in: "On Conceptual Modelling"; Brodie M.L., Mylopoulos J., Schmidt J.W., editors, Springer-Verlag 1984.

5. Brachman R.: "On the Epistemological Status of Semantic Networks", in *Associative Networks: Representation and Use of Knowledge by Computers*, Findler, N., V., (ed.), New York, Academic Press, 1979.

6. Brachman R.J.: "What Is-a Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks"; IEEE Computer, October 1983.

7. Brodie, M., L., Silva, E., "Active and Passive Component Modelling: ACM/PCM"; Information Systems Design Methodologies: A Comparative Review, Olle, Sol, Verrijn-Stuart, ed., North Holland Publ. Comp., 1982.

8. Brodie M.L., Mylopoulos J., Schmidt J.W., editors: "On Conceptual Modelling"; Springer, 1984.

9. Coad P., Yourdon E.: "Object-Oriented Analysis", Yourdon Press Prentice Hall, Englewood Cliffs, NJ, 1990.

10. Chen P.P.S., "The Entity-Relationship Model - Towards a Unified View of Data"; ACM TODS, Vol.1, No.1, 9-36, 1976.

11. Davis J., P., Bonnell R., D.: "A Framework for Constructing Visual Knowledge Representation Specifications in Acquiring Organizational Knowledge", in *Knowledge Acquisition*, Vol.3, No 1, p.79-115, March 1991.

12. De Champeaux D., Faure P.: "A Comparative Study of Object-Oriented Analysis Methods", *Journal of Object-Oriented Programming*, p. 21-33, March/April 1992.

13. Embley D., W., Kurtz B., D., Woodfield S., N.: "Object-Oriented Systems Analysis - A Model-Driven Approach", Yourdon Press, Englewood Cliffs, NJ, 1992.

14. Fikes R.E., Kehler T.P.: "The Role of Frame-Based Representation in Reasoning"; in: CACM, September 1985.

15. Goldberg, A. and Robson, D., Smalltalk-80: *The Language and its Implementation*, Addison-Wesley, 1983.

16. Halasz, F., G., "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems"; CACM, Vol.31, No.7, 836-852, July 1988.

17. Hammer, M., Mc Leod, D., "Database Description with SDM: A Semantic Database Model"; ACM TODS, Vol.6, No.3, 351-386, Sept. 1981.

18. Hull, R. and King, R., "Semantic Database Modelling: Survey, Applications and Research Issues", *ACM Computing Surveys 19(3)*, September 1987.

19. Jackson, M., "System Develpoment"; Prentice Hall, 1982.

20. Kim, W., Bertino, E., Garza, J., F., "Composite Objects Revisited"; OOPSLA 89', 337-347, 1989.

21. Koubarakis, M., Mylopoulos, J., Stanley, M., Borgida, A., "Telos: Features and Formalization", KRR-TR-89-4, University of Toronto, Feb. 1989.

22. Lenzerini M., Nardi D., Simi M. (ed.): "Inheritance Hierarchies in Knowledge Representation and Programming Languages", John Wiley & Sons, 1991.

23. Martin J., Odell J.: "Object-Oriented Analysis and Design", Prentice Hall, Englewood Cliffs, NJ, 1992.

24. Miller, G., A., Johnson-Laird P., N., Language and Perception, Cambridge, MA: Harvard University Press, 1976.

25. Motschnig-Pitrik R., "A Framework for the Support of a Common Structural Level for Software, Database-, and Knowledge Based Systems"; The Journal of Systems and Software, North Holland, Vol.12, No.12, 157-165, 1990.

26. Motschnig-Pitrik R., "Toward a Common Structural Level for Software, Database-, and Knowledge Based Systems"; *Applied Artificial Intelligence, An Int.. Journal*, Hemisphere Publ., Vol.3, No.4, 405-426, 1991.

27. Motschnig-Pitrik R., Mylopoulos, J., "Classes and Instances"; Int. Journal on Intelligent and Cooperative Information Systems, Vol.1, No.1, 1992.

28. Mylopoulos, J., Borgida, A., Jarke, M. and Koubarakis, M., "Telos: Representing Knowledge About Information Systems", *ACM Transactions on Information Systems*, (to appear).

29. Mylopoulos, J., "Object-Orientation and Knowledge Representation", in Meersman, R. and Kent W. (eds.) Proceedings of the IFIP-TC 2 Working Conference on Database Semantics: Object-Oriented Databases -- Analysis, Design and Construction, Windermere UK, July 1990.

30. Peckham J., Maryanski F.: "Semantic Data Models", ACM Computing Surveys, Vol.20, No.3, p.153-189, September 1988.

31. Pitrik, R., "Structuring Principles in the Design of Software Systems", in: Advances in Computer Science; G. Lasker, ed., University of Windsor, Canada, 1989..

32. Rosch, E., "On the Internal Structure of Perceptual and Semantic Categories", in T. E. Moore (ed.), *Cognitive Development in the Acquisition of Language*, New York: Academic Press, 1973.

33. Ross, D.T., "Structured Analysis (SA): A Language for Communicating Ideas"; IEEE TSE Vol. SE-3, No.1, 1977.

34. Storey, V., "Understanding Semantic Relationships"; Working Paper, University of Rochester, NY, January 1992; to appear in the Journal on Very Large Data Bases.

35. Smith J., Smith D.: "Data Abstractions: Aggregation and Generalization", TODS, Vol.2, No.2, p. 105-133, June 1986.

36. Teorey, T., J., Yang, D., Fry, J., P., "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model"; ACM Computing Surveys, Vol.18, No.2, 197-222, June 1986.

37. Tversky B., Hemenway K.,: "Objects, Parts, and Categories"; *Journal of Experimental Psychology: General*, Vol.113, No.2, 169-191, June 1984.

38. Ugagawa, Y., "Implementation and Evaluation of a Browsing Algorihm for Design Applications"; Proc. of the 7 th Int. Conf. on Data Engineering, IEEE, 70-78, Kobe, Japan, April 8 - 12, 1991.

39. Wand Y., Weber R.: "A Unified Model of Software and Data Decomposition", Working Paper, Fac. of Commerce and Business Administration, The University of British Columbia, March 1991.

40. Winston M. E., Chaffin R., Herrmann D.: "A Taxonomy of Part-Whole Relations"; *Cognitive Science*, Vol.11, 417-444, 1987.

Appendix

Examples to demonstrate the transitivity of arbitrary two-place variations of the core part-of categories:
component/object portion/mass place/area

1) component/object and portion/mass

variable part-of statement,
statement part-of code
--> variable part-of code;

2) portion/mass and component/object

statement part-of code,
code part-of software product
--> statement part-of software product;

3) component/object and place/area

CN-Tower part-of Toronto;
Toronto part-of Canada
--> CN-Tower part-of Canada;

4) place/area and component/object

Toronto part-of Canada,
Canada part-of world
--> Toronto part-of world;

5) portion/mass and place/ area

South of Everglades part-of Everglades,
Everglades pat-of Florida
--> South of Everglades part-of Florida;

6) place/area and portion/mass

Toronto part-of Canada,
Canada part-of continent
--> Toronto part-of continent;

Example to demonstrate the transitivity of part-of relations belonging to the verb-feature/activity category:

7) testing part-of implementing,
implementing part-of developing software
--> testing part of developing software.