Part-whole relations in Object-Role Models

C. Maria Keet

KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy. keet@inf.unibz.it

Abstract. Representing parthood relations in ORM has received little attention, despite its added-value of the semantics at the conceptual level. We introduce a high-level taxonomy of types of meronymic and mereological relations, use it to construct a decision procedure to determine which type of part-whole role is applicable, and incrementally add mandatory and uniqueness constraints. This enables the conceptual modeller to develop models that are closer to the real-world subject domain semantics, hence improve quality of the software.

1 Introduction

Of all roles one can model in ORM, it is obvious from the set-theoretic formal semantics that subsumption is a first-class citizen as constructor, which is reflected in its graphical representation with a subsumption arrow instead of a role-box. Giving such first-class citizen status to part-whole roles can be less obvious. The $partOf$ relation between object types in ORM has received little attention, apart from Halpin's assessment [9] [10]. He concludes that it is doubtful if it adds any semantics at the conceptual level and that design considerations can 'sneak into' conceptual modelling, because it is said to involve modelling object life cycle semantics (propagating object creation & destruction in the software). So, why bother? First, this conclusion was reached based on the treatment of the aggregation relation in the UML specification v1.3, which is known to be inadequate for representing the semantics of part-whole relations (e.g. [7] [14]). Second, as will be come clear in this paper, part-whole roles *do* enable a modeller to represent the semantics of the subject domain more precisely; hence one can create software that better meets the user's requirements. Third, in the past several years, research into bringing the part-whole relation toward the application stage has gained momentum. The latter entails other advantages such as concept satisfiability checking [3] [5], inferring derived relations and ensuring semantically correct transitivity of relations [4] [5] [20], and achieve (semi-)automatic abstraction and expansion of large conceptual models [12].

We approach part-whole roles for ORM from the perspective of usability and focus on the modeler and user. Many formal ontological aspects of the part-whole role have been discussed (e.g. [1] [4] [6] [16] [17] [18] [19] [21] [22]) and extensions to conceptual modelling languages have been suggested, like [2] [7] [14] for UML. This this is summarised and improved in section 2. Unfortunately, none of the extensions are implemented, as the wide range of modelling options tend to be off-putting. We propose stepwise 'incremental modelling' of part-whole roles, which can be integrated with the customary approach in ORM modelling, thereby structuring and easing the modelling of part-whole roles. This consists of the use of a) a decision procedure to facilitate eliminating the wrong types of part-whole and apply the right one, and b) additional question & answer sessions for uniqueness and mandatory constraints. This is presented in section 3 and applied to an example ORM model in section 4. Last, we draw conclusions and point to further research.

2 Parthood relations and aggregation

Mereology is the formal ontological investigation of the part-whole relation. It has an overlap with meronymy – which concerns part-whole relations in linguistics – but they are not the same, as there are meronymic relations that are not partonomic (see below). Varzi [18] provides an overview of the more and less constrained versions of mereology from the viewpoint of philosophy, and Guizzardi [7] provides a summary from the perspective of conceptual modelling. What mereology lacks, however, is the engineering usefulness for conceptual modelling by being at times more comprehensive (e.g. mereotopology) and limiting regarding other aspects, such as 'horizontal' relations between the parts and the inverse relation $hasPart$. First, we analyse the main aspects of part-whole relations and propose a top-level taxonomy of relations, and subsequently discuss and compare how its main characteristics have been translated to different conceptual modelling languages.

2.1 Mereology and meronymy

The most basic constraints on the parthood relation in mereology, called *Ground Mereology*, are that a partial ordering is always reflexive (1), antisymmetric (2), and transitive (3). All other versions [7] [18] share at least these constraints. Taking $partOf$ as primitive relation, i.e. it does not have a definition, then (1-3) enables one to define *proper* part as (4), from which asymmetry, and irreflexivity follows; thus, x is not part of itself, if x is part of y then y is not part of x, and if x is part of y and y part of z then x is part of z.

$$\forall x(partOf(x,x)) \tag{1}$$

$$\forall x((partOf(x,y) \land partOf(y,x)) \rightarrow x = y) \tag{2}$$

$$\forall x,y,z((partOf(x,y) \land partOf(y,z)) \rightarrow partOf(x,z)) \tag{3}$$

$$properPartOf(x,y) \triangleq (partOf(x,y) \land \neg(partOf(y,x)) \tag{4}$$

Contrary to these straightforward axioms, the transitivity of the $partOf$ relation is regularly discussed and contested (e.g. [11] [16] [20]), including introducing 'types' of part-whole relations to ensure transitivity. On closer inspection, it appears that in case of different types of part-whole relations, different types of universals are related, and, provided one makes the required distinctions, transitivity still holds (see also [19]). For instance, it is common to relate a process to its part-processes as $involvedIn$ to distinguish it from the $partOf$ relation between endurants (object types). Each type of part-whole role then has to be extended with constraints on the participating object types, like

$$\forall x, y(involvedIn(x, y) \triangleq properPartOf(x, y) \wedge Process(x) \wedge Process(y)) \quad (5)$$

Other variants include relating object types spatially through the part-whole relation, denoted as $containedIn$ [4], or $locatedIn$ for relating spatial (geographical) objects. An important distinction exist between mereological $partOf$ relations and meronymic part-whole relations, where the latter is not necessarily transitive. For instance, $memberOf$, also referred to as "member-bunch" [16], is an intransitive meronymic part-whole relation, like players are members of a rugby team, probably member of that team's club, but as player certainly not member of the rugby clubs federation. We illustrate (in-)transitivity of several mereological and meronymic part-whole relations in the following examples, where we have extended or modified the names of the relations in most examples to indicate their ontological type.

- ⋆ - Centimeter part of Decimeter
  - Decimeter part of Meter
  therefore Centimeter part of Meter
  - Meter part of SI
  but *not* Centimeter part of SI, because Meter is actually a *member of* the Système International d'Units.
- ⋆ - Vase constituted of Clay
  - Clay has structural part GrainOfSand
  but *not* Vase constituted of GrainOfSand
- ⋆ - CellMembrane structural part of Cell
  - Cell contained in Blood
  but *not* CellMembrane structural part of Blood
  - Lipid structural part of CellMembrane
  therefore Lipid structural part of Cell
- ⋆ - Politician member of PoliticalParty
  - PoliticalParty located in Bolzano
  therefore Politician located in Bolzano? But *not* Politician member of Bolzano
- ⋆ - ReceptorBindingSite regional part of Receptor
  - Receptor functional part of SecondMessengerSystem
  therefore ReceptorBindingSite functional part of SecondMessengerSystem?

To disambiguate these differences and ensure transitivity, efforts have gone into constructing a taxonomy of part-whole relations. The first proposal, motivated by linguistic use of 'part', i.e. meronymy, was made by Winston, Chaffin and Herrmann (WCH) [22] and several successive articles deal with analysing the WCH taxonomy and modelling considerations (e.g. [1] [6] [7] [16]). For instance, Gerstl and Pribbenow [6] prefer a "common-sense theory of part-whole relations" instead, to allow for "different views on the entities". They reduce the six types of part-whole relations of WCH into three: component-complex, element-collection, and quantity-mass. Conversely, this has been extended and improved upon by Guizzardi [7], who provides criteria for several types, although note that his $subCollectionOf$ is actually a set-subset relation and therefore not included in the taxonomy in Fig.1. In concordance with foundational ontological notions [13], we categorise element-collection as a type of *membership*, which is a meronymic relation. In addition, quantity-mass has to do with object types generally denoted with mass nouns that are not countable, such as water or wine, but one can count *portions* of wine and slices of the pie; thus, a portion is of the same substance (amount of matter) as the whole. Odell's material-object "part-of" relation [16] ontologically corresponds to *constitution*, where a vase is *made of* clay or a bike *constituted of* steel (see also [13]). Taking into account the additional ontological distinctions, we devised a taxonomy of – for conceptual data modelling relevant – types of mereological $partOf$ and meronymic relations, which is depicted in Fig.1. This, however, does not yet deal with other facets of parthood relations, such as existential and mandatory parts, the inverse relation $hasPart$, and if the parts together are *all* parts that make up the whole. These facets are relevant for conceptual data modelling and therefore addressed in §3 and 4.

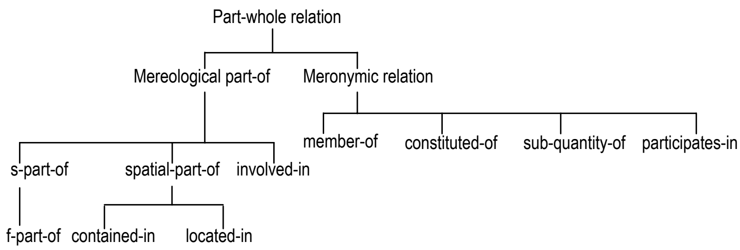

Fig. 1. Taxonomy of basic mereological and meronymic part-of relations. s-part-of = structural part-of; f-part-of = functional part-of.

2.2 ER, UML, ORM, and DL

ER, ORM and Description Logics (DL) do not have special constructors in the language to represent $partOf$, and few are in favour of giving it a first-class citizen status in ER [17] and DL [1] [4]. This does not mean that one is better off with UML. UML implements two modes of the $partOf$ relation: composite and shared aggregation [15]. Composite aggregation, denoted with a filled diamond on the whole-side of the association (Fig.2-A), is defined as

> a strong form of aggregation that requires a part instance be included in at most one composite at a time. If a composite is deleted, all of its parts are normally deleted with it. Note that a part can (where allowed) be removed from a composite before the composite is deleted, and thus not be deleted as part of the composite. Compositions define transitive asymmetric relationships – their links form a directed, acyclic graph. [15]

This ‘implementation behaviour’ of creation/destruction of parts implicitly states that the parts are *existentially dependent* on the whole, and not that when a whole is destroyed its parts can exist independently and become part of another whole. Thus, UML’s implementation behaviour is an implicit *ontological commitment at the conceptual level*. In addition, only binary associations can be aggregations [15], which is peculiar from an ontological perspective, because it suggests that a whole can be made up of one type of part only, except for extending the representation with a {`complete`} in the OCL (e.g. [14]). This difference is not addressed in the UML specification, i.e. it is a “semantic variation point” [15]. Likewise, shared aggregation, denoted with an open diamond on the whole-side, has it that “precise semantics ... varies by application area and modeler” [15], and presumably can be used for any of the $partOf$ types described in Fig.1 and by [11] [14] [16] [22] etc. Unlike composite aggregation, shared aggregation has no constraint on multiplicity with respect to the whole it is part of; thus, the part may be *directly shared* by more than one whole at the same time. Overall, the ambiguous specification and modelling freedom in UML does not enable making implicit semantics explicit in the conceptual model, and rather fosters creation of unintended models.

Halpin’s mapping from UML aggregation to its ORM representation [9], on the other hand, indirectly gives a formal semantics to UML’s aggregation, as, unlike UML, ORM actually has a formal semantics. This mapping is depicted in Fig.2. Using Halpin’s formalisation [8] and setting aside the difference between membership and parthood, the *Club*-*Team* fact has its corresponding first order logic representation as (6-9) and *Team*-*Person* as (10-11).

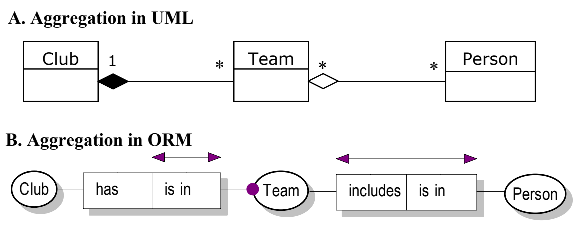

Fig. 2. Graphical representation of “aggregation” in UML and ORM. (Source: [9])

$$\forall x, y, z((isIn(x, y) \wedge isIn(x, z)) \rightarrow y = z) \tag{6}$$

$$\forall x, y(isIn(x, y) \rightarrow Team(x) \wedge Club(y)) \tag{7}$$

$$\forall x(Team(x) \rightarrow \exists y(isIn(x, y))) \tag{8}$$

$$\forall x_1, x_2(isIn(x_1, x_2) \equiv has(x_2, x_1)) \tag{9}$$

$$\forall x, y(isIn(x, y) \rightarrow Person(x) \wedge Team(y)) \tag{10}$$

$$\forall x_1, x_2(isIn(x_1, x_2) \equiv includes(x_2, x_1)) \tag{11}$$

The difference between ORM and UML intended semantics is that with composite aggregation in UML, part x *cannot* exist without that whole y, but ORM semantics of the suggested mapping [10] says that 'if there is a relation between part x and whole y, then x must participate exactly once'. Put differently, x may become part of some other whole y' after y ceases to exist, as long as there is *some* whole of type Y it is part of, but not necessarily *the same* whole. Hence, in contrast with UML, in ORM there is no strict existential dependency of the part on the whole. Both more [2] [7] [14] and less [3] comprehensive formalizations and extensions for aggregation in UML have been given. For ORM, richer representations of the semantics are possible already even without dressing up the ORM diagram with icons and labels.

3 Options to represent part-whole relations in ORM

Advantages to include different parthood relations are automated model verification, transitivity (derived relations), semi-automated abstraction operations, enforcing good modelling practices, and it positions ORM further ahead of other conceptual modelling languages. On the other hand, specifying everything into the finest detail may be too restrictive, results in cluttered diagrams, is confusing to model, and costs additional resources to include in ORM tools. That is, if we include all basic options in the syntax, with the formalization, particular graphical notation, and fixed-syntax sentences, there are at least 63 combinations. Gradual integration of modelling parthood relations will yield better results at this stage. Therefore, we introduce guidelines in the form of a decision procedure, and additional modelling questions to facilitate conceptual modelling process. The major advantages of this approach are its flexibility for both current use and future extensions, it reduces modelling mistakes, and with syntactic and textual analysis, it is still usable for aforementioned reasoning tasks.

The first, and main, step is to decide which role to use. Fig.3 presents a decision procedure, which first assesses – or rules out – all meronymic part-whole relations (up to *participatesIn*) and subsequently goes through the various mereological parthood relations. Although the order of the decision steps can be changed, ordering the two kinds in sequence serves conceptual clarity. Maintaining mereology in the second part of the decision procedure permits non-disruptive extensions to even finer-grained distinctions of parthood relations, if deemed desirable. There are several possibilities to implement the procedure. These options range from a 'no tech' cheat-sheet, 'low tech' drop-down box with the 9 types, to software-support for a decision procedure that asks questions and provides examples corresponding to each decision diamond.

The next step comprises ascertaining existential dependence, mandatoryness and shareability. In addition to the questions that are automatically generated in e.g. VisioModeler, we propose 5 additional questions specific for the parthood roles, which also consider the inverse roles. Looking ahead to the example ORM model in Fig.5, for the fact ShoulderHandle f-part of ConferenceBag / ConferenceBag has f-part ShoulderHandle the default questions generated for selecting 0:1, 0:n, 1, or 1:n are:

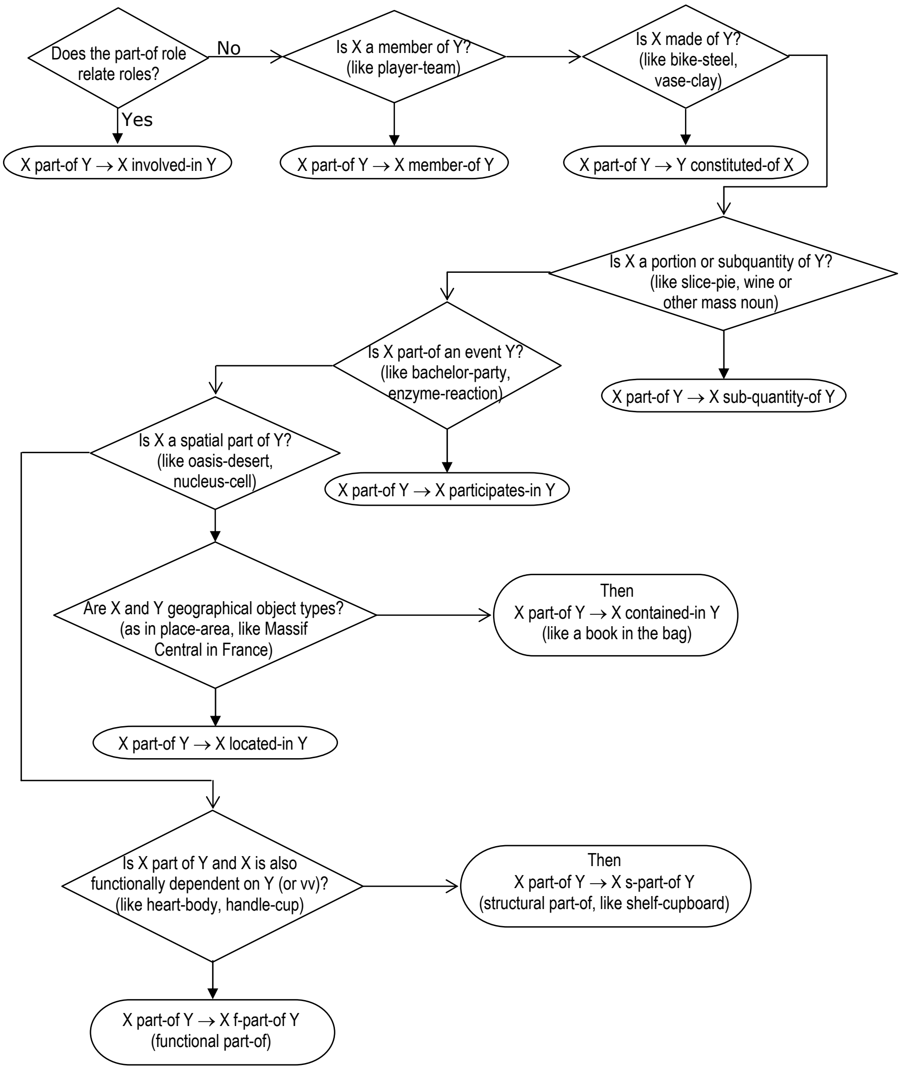

Fig. 3. Decision diagram to ascertain the appropriate parthood relation.

Each **ConferenceBag** has how many **f-part ShoulderHandle**?

How many instances of '**ShoulderHandle**' may be recorded for each instance of '**f-part of ConferenceBag**'?

We reformulate and extend the questions for the part-whole roles to emphasise the properties strict existential dependence, mandatory participation, and shareability. Generalizing for any case of $partOf$ (A1-A3) and $hasPart$ (B1, B2), where the part is of type P and whole of type W, we have:

A1. Can a **P** exist without some **W** it is **part of**?

A2. Can a **P** exist without the same **W** it is **part of**?

A3. Can a **P** be **part of** only one whole? (if not, then **P** can be shared among wholes)

B1. Can a **W** (continue to) exist when it does not **has part** some **P**?

B2. Can a **W** (continue to) exist when it does not **has part** the same **P**?

Fig.4 shows the resultant facts for each answered question; note the different effects of the "some" and "same" in the questions and representation. In a real model, P and W are replaced by their respective object types in the ORM model and part of and has part are replaced by the corresponding type of part-whole roles.

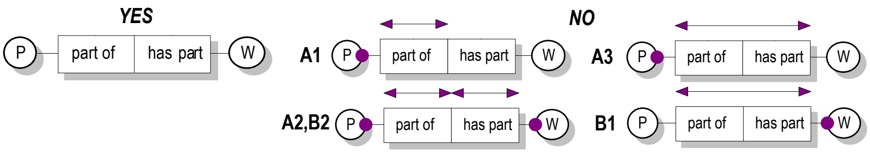

Fig. 4. Representations resulting from the answers to questions A1-B2, with one "yes" result and four distinct representations for "no".

4 Disambiguation of part-whole roles: an example

We demonstrate the results of applying the decision tree and additional questions with a sample ORM model. The top-half of Fig.5 has a model with underspecified part-whole facts, whereas the bottom-half contains the disambiguated version after going through the descision procedure for each role. For instance, Envelope is not involved-in, not a member-of, does not constitute, is not a sub-quantity of, does not participate-in, is not a geographical object, but instead is contained-in the ConferenceBag. Now, with the clear semantics of the part-whole roles, transitivity holds for the mereological relations: derived facts are automatically correct, like RegistrationReceipt contained-in ConferenceBag. Also intransitivity is clear, like that of Linen and ConferenceBag, because a conference bag is not wholly constituted of linen (the model does not say what the Flap is made of). The notion of completeness, i.e. that *all* parts make up the whole, is implied thanks to the closed-world assumption. For instance, ConferenceBag directly contains the ConfProceedings and Envelope *only*, and does not contain, say, the Flap. The structural parts of the whole ConferenceBag are Compartment and Flap. The composite has a functional part, has f-part of Shoulderhandle, which is neither an essential nor a mandatory part of the whole, yet it does not imply shareability either.

5 Conclusions and further research

We have introduced a taxonomy of types of meronymic and mereological relations, and used it to construct a decision procedure to determine which type of parthood relation is applicable. Incrementally, mandatory and uniqueness constraints can be added, which enable the conceptual modeller to develop models

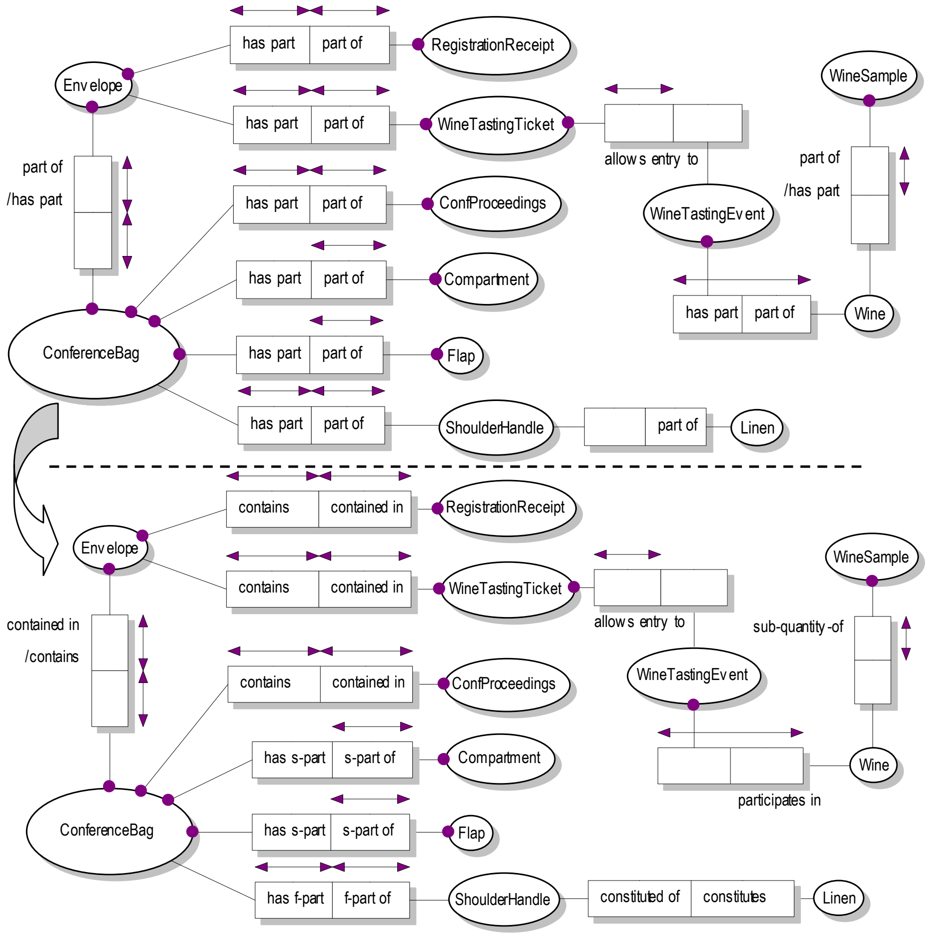

Fig. 5. Example ORM model with all part-of/has-part relations (top) and disambiguated mereological and meronymic parthood relations (bottom).

that are closer to the real-world semantics, hence improve quality of the software. When used more widely, it will be useful to add extensions to the language, e.g. as a separately loadable module in ORM tools for those analysts who need it, analogous to the Description Logics approach with a family of more and less expressive knowledge representation languages.

References

1. Artale, A., Franconi, E., Guarino, N., and Pazzi, L. (1996). Part-Whole Relations in Object-Centered Systems: an Overview. *Data & Knowledge Engineering*, 1996, 20(3):347-383.

2. Barbier, F., Henderson-Sellers, B., Le Parc-Lacayrelle, A., Bruel, J.-M.. Formalization of the whole-part relationship in the Unified Modelling Language. *IEEE Transactions on Software Engineering*, 2003, 29(5):459-470.
3. Berardi, D., Calvanese, D., De Giacomo, G. Reasoning on UML class diagrams. *Artificial Intelligence*, 2005, 168(1-2):70-118.
4. Bittner, T., Donnelly, M. Computational ontologies of parthood, componenthood, and containment, In: *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence 2005 (IJCAI05).* Kaelbling, L. (ed.). pp382-387.
5. Franconi, E., Ng, G. The iCom tool for intelligent conceptual modelling. *7th Intl Workshop on Knowledge Representation meets Databases*. Berlin, Germany. 2000.
6. Gerstl, P., Pribbenow, S. Midwinters, end games, and body parts: a classification of part-whole relations. *Intl. Journal of Human-Computer Studies*, 1995, 43:865-889.
7. Guizzardi, G. *Ontological foundations for structural conceptual models.* PhD Thesis, Telematica Institute, Twente University, Enschede, the Netherlands. 2005.
8. Halpin, T.A. *A logical analysis of information systems: static aspects of the data-oriented perspective.* PhD Thesis, University of Queensland, Australia. 1989.
9. Halpin, T. UML Data Models from an ORM Perspective (Part 8). *Journal of Conceptual Modeling*, Issue 8, April 1999. http://www.inconcept.com/jcm.
10. Halpin, T. *Information Modeling and Relational Databases.* San Francisco: Morgan Kaufmann Publishers, 2001.
11. Johansson, I. On the transitivity of the parthood relation. In: *Relations and predicates*, Hochberg, H., Mulligan, K. (eds.). Ontos Verlag: Frankfurt, 2004. pp161-181.
12. Keet, C.M. Using abstractions to facilitate management of large ORM models and ontologies. *International Workshop on Object-Role Modeling (ORM'05).* In: OTM Workshops 2005. Halpin, T., Meersman, R. (eds.), Lecture Notes in Computer Science 3762. Berlin: Springer-Verlag, 2005. pp603-612.
13. Masolo, C., Borgo, S., Gangemi, A., Guarino, N. and Oltramari, A. *Ontology Library.* WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003). http://wonderweb.semanticweb.org. 2003.
14. Motschnig-Pitrik, R., Kaasbøll, J. Part-Whole Relationship Categories and Their Application in Object-Oriented Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 1999, 11(5):779-797.
15. Object Management Group. *Unified Modeling Language: Superstructure.* v2.0. formal/05-07-04. http://www.omg.org/cgi-bin/doc?formal/05-07-04.
16. Odell, J.J. *Advanced Object-Oriented Analysis & Design using UML.* Cambridge: Cambridge University Press. 1998.
17. Shanks, G., Tansley, E., Weber, R. Representing composites in conceptual modeling. *Communications of the ACM*, 2004, 47(7): 77-80.
18. Varzi, A.C. Mereology. *The Stanford Encyclopedia of Philosophy.* (Fall 2004 Edition), Zalta, E.N. (ed.). http://plato.stanford.edu/archives/fall2004/entries/mereology/.
19. Varzi, A.C. The formal ontology of space: parts, wholes, and locations. In: *The Logic of Space*, Aiello, M., Pratt-Hartmann I., van Benthem J. (eds.). Dordrecht: Kluwer Academic Publishers. Chapter 1, 104p. draft http://www.columbia.edu/∼av72/papers/Space_2006.pdf, date accessed: 16-5-2006.
20. Varzi, A.C. A Note on the Transitivity of Parthood. *Applied Ontology* (to appear).
21. Vieu, L., Aurnague, M. Part-of Relations, Functionality and Dependence. In: M. Aurnague, M. Hickmann, L. Vieu (eds.), *Categorization of Spatial Entities in Language and Cognition.* Amsterdam: John Benjamins. 2005.
22. Winston, M.E., Chaffin, R., Herrmann, D. A taxonomy of partwhole relations. *Cognitive Science*, 1987, 11(4):417-444.