

Positionalism of Relations and Its Consequences for Fact-Oriented Modelling

C. Maria Keet

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

Abstract. Natural language-based conceptual modelling as well as the use of diagrams have been essential components of fact-oriented modelling from its inception. However, transforming natural language to its corresponding object-role modelling diagram, and vv., is not trivial. This is due to the more fundamental problem of the different underlying ontological commitments concerning *positionalism* of the fact types. The natural language-based approach adheres to the *standard view* whereas the diagram-based approach has a *positionalist* commitment, which is, from an ontological perspective, incompatible with the former. This hinders seamless transition between the two approaches and affects interoperability with other conceptual modelling languages. One can adopt either the limited standard view or the positionalist commitment with fact types that may not be easily verbalisable but which facilitates data integration and reusability of conceptual models with ontological foundations.

1 Introduction

Different methodologies and practices to design a fact-oriented or object-role model exist, which commence with a natural language sentence, sample data, or with icons at the type-level in the diagrammatic language. A common aspect of them is the ‘one *fact(type)* at a time’ approach, which follows naturally from the pseudo-natural language starting point, but it is a conscious choice in the diagrammatic interface because with the latter one also could start with drawing several objects and relate them one by one or draw n -ary relationships and then link object types to roles of the relationships. Unlike the (pseudo-)natural language approach, the diagrammatic interface does emphasise the existence and explicit use of *roles* as components of the relationships. For instance, a fact such as **John loves Mary** does not deal with roles, such as the *Lover* role that **John** plays and the *Beloved* role that **Mary** plays. Does this constitute a fundamental, ontological, difference or is it a byproduct of the different interfaces available in the modelling tools and a modeller’s habits and personal preferences? And what if it does constitute a difference? Should a modeller care at all?

In this paper we investigate this problem by availing of recent results from philosophy, and the notion of *positionalism* of relations in particular, with which we isolate and analyse the different ontological commitments underlying the natural language and diagram-based approaches of fact-oriented modelling (FOM).

It is precisely due to the different ontological commitments that there are different formal characterisations for the ‘basic elements’ of the language. In addition, the different parallel commitments affect approaches and practices to conceptual data modelling methodologies, the possibilities for extensions, such as stereotyping relationships, and has consequences for other usage of FOM models, such as data integration, which can be better addressed with the role-based, positionalist, approach than the natural language approach.

The remainder of the paper is structured as follows. Section 2 presents a synthesis of the ontological aspects of relations, which we apply to FOM and analyse its practical consequences in section 3. We conclude in section 4.

2 Positionalism

The ontological status of relations receives much attention in philosophy recently. Early ideas were put forward by Williamson [1] and have been elaborated on and structured in [2,3,4,5]. Here we synthesise the main arguments that will be directly relevant for assessing ORM and adjust terminology and the narrative to conceptual data modelling. We introduce the three different ontological commitments about relations and relationships, which are, in Fine’s [2] terminology, the *standard view*, the *positionalist*, and the *anti-positionalist* commitment. This forms the theoretical basis for analysing the consequences for ORM in the next section.

Let us start with the standard view, which relies on linguistics and the English language in particular. With the fact **John loves Mary**, one could be led to assume that *loves* is the name of the relation and *John* and *Mary* are the objects participating in the relation. Switching the objects, **Mary loves John**, is clearly a different thing and it is not guaranteed to have the same truth value as the former fact; changing the verb from active to passive voice does, i.e., **Mary is loved by John**. In the standard view, we seem to have two relations, *loves* and its inverse *is loved by*. This faces several problems. First, generally, for names *a* and *b*, *a loves b* holds iff what *a* denotes (in the reality we aim to represent) loves what *b* denotes. Names normally denote non-linguistic items, because normally the truth conditions of sentences involving names are not sensitive to the existence of linguistic items: **John loves Mary** is not about language but about John loving Mary, so John and Mary are non-linguistic. Compare this to the fact ‘**cabeza** translates into ‘head’’, which is about the language. [1]. Then, that John loves Mary and Mary is being loved by John refer to only *one* state of affairs between John and Mary—so why should we want, let alone feel the need, to have *two relations* to describe it? A first step toward resolving this, is to designate the two aforementioned facts to be *relational expressions* and not to let the verb used in the fact automatically also denote the name of the *relation*, so that we can have many relational expressions standing in for the single relation that captures the state of affairs between John and Mary. In analogy, we can have many relational expressions for one relationship at the type level.

A second issue is that the relational expression comes in a specific order where changing the order does not mean the same when we consider verbs

that indicate an asymmetric relation (asymmetry of a relationship R being $\forall x(R(x, y) \rightarrow \neg R(y, x))$), such as *loves* as relation for the verb *loves*. Availing of another language illustrates this better. Consider *John kills the dragon*. In Latin, we can convey the same with *Johannus anguigenam caedit* and with *anguigenam caedit Johannus* that both refer to the same state of affairs, but *Johannum anguigena caedit* (the dragon kills John) is a different story altogether. With Latin and other languages such as German and Irish, we implicitly have a linguistic version of *argument places* thanks to the explicit use of the nominative and the accusative that are linguistically clearly indicated with *-us* and *-am*, and therefore the order of the words matters little (in fact, the order of the argument places is not relevant for the relation itself). English, on the other hand, does not have such declensions that change the terms as a means to disambiguate the meaning of a relational expression, so when we change the word order of a fact truthful to its meaning, then either the verb has to be modified or another verb has to be used. But could we have that in reality and descriptions of reality in English language inverses for seemingly asymmetrical relations necessarily exist, but not in other languages even when they represent the same state of affairs? There are relational expressions that are certainly asymmetric, but this does not imply that the relation and fact(type) it verbalises is asymmetric. Nothing prevents us from using a binary relation *killing* and to identify the argument places as *killer* and *deceased*—i.e. two “argument positions” [2] to have “distinguishability of the slots” [5], or, loosely, a place for the nominative and a place for the accusative—, assign *John* to *killer* and *the dragon* to *deceased* and order the three elements in any arrangement we like.

More generally, we then have as ingredients (i) an n -ary relationship R with A_1, \dots, A_m participating object types ($m \leq n$), (ii) n argument places π_1, \dots, π_n , and (iii) n assignments $\alpha_1, \dots, \alpha_n$ that link each object o_1, \dots, o_n (each object instantiating an A_i) to an argument place ($\alpha \mapsto \pi \times o$). Given a, for instance, ternary relationship at the intensional level, R , argument places π_1, π_2 , and π_3 of that relationship, and instances $r \in R, o_1 \in A_1, o_2 \in A_2$, and $o_3 \in A_3$, then any of $\forall x, y, z(R(x, y, z) \rightarrow A_1(x) \wedge A_2(y) \wedge A_3(z))$ —in shorthand $R(A_1, A_2, A_3)$ —and its permutations $R(A_2, A_1, A_3)$ and $A_2 A_3 R A_1$ where each one has its corresponding argument places—i.e., $R[\pi_1, \pi_2, \pi_3]$, $R[\pi_2, \pi_1, \pi_3]$, and $[\pi_2 \pi_3] R [\pi_1]$ —they all denote the same state of affairs under the same assignment o_1 to π_1 , o_2 to π_2 , and o_3 to π_3 for the extension of the relation. Thus, $r(o_1, o_2, o_3)$, $r(o_2, o_1, o_3)$, and $o_2 o_3 r o_1$ are different representations of the same state of affairs where objects o_1, o_2 , and o_3 are related to each other by means of relation r . One can visualise this *positionalist* ontological commitment as shown in Fig. 1-A. We simply have a relation(ship) and several distinguishable ‘holes’ and we put each object in its suitable hole. It may well be the case that not all of the permutations have a nicely readable relational expression in English, but that does not invalidate the permutation and/or even the whole relation.

The positionalist commitment solves the standard view’s problems with seemingly asymmetrical relations because there is just *one* relation for a state of affairs, not two or more as the standard view suggests. According to [1,2,5], there

are no asymmetrical relations, because a relationship R and its inverse R^- , or their instances, say, r and r' , are *identical*, i.e., the same thing. However, also the positionalist commitment may not be perceived to be ideal. From an ontological viewpoint, the positionalist solution to the ontological nature of relations requires identifiable argument positions to be part of the fundamental furniture of the universe. Practically, it requires something to finger-point to, i.e. to reify the argument places, and use it in the signature of the formal language, which is not as clean and simple as without such elements. In addition, there is a problem with symmetric relations and relationships, such as *adjacent to*: when we have the two argument positions, π_a and π_b , of a symmetric binary relation r and assign o_1 to position π_a and o_2 to π_b in state s , we can do a reverse assignment of o_1 to position π_b and o_2 to π_a in state s' , but then o_1 and o_2 do not occupy the same positions as they did in s , so s and s' must be different, which should not be the case. The solution proposed by Fine [2] is the *anti-positionalist* ontological commitment. There are no argument positions, but just a relation and objects that somehow yield states by “combining” into “a single complex”, like a magnet attracts iron things or pins one sticks into a pincushion; this is depicted in Fig. 1-B. This approach eliminates the need for admitting existence of argument positions—hence, avoids the argument position assignment step that is problematic for the positionalist’s symmetric relation. Verbalising such a relation is, ontologically, of no concern; put differently: one can use as many phrases as one sees fit in as many natural languages as desired, or none.

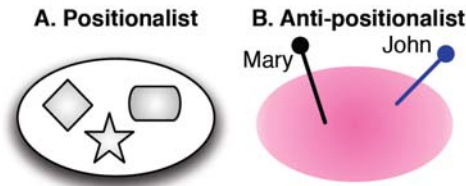


Fig. 1. A: Graphical depiction of the postionalist ontological commitment with relationship R as ellipse and its three argument places as ‘holes’ with shapes square, star, and roundtangle; B: Graphical depiction of the anti-postionalist commitment with a binary relation as shaded pincushion and two participating objects as pins stuck into the cushion in arbitrary places yielding a state

3 Assessing Fact-Oriented Modelling

With the principal aspects of positionalism in place, we can commence examining conceptual modelling and its languages, and FOM in particular. Before doing so, it is worthwhile to recollect that ORM and ORM2 are, roughly, more expressive than UML Class diagrams and EER. Considering their respective ontological commitment, a UML Class diagram with “association ends” [6] has a positionalist commitment and the EER diagram notation adheres to an anti-positionalist commitment. ORM, LISA-D, NIAM, FCO-IM, and FOM superficially refer to

the same kind of thing. However, some use *facts* and *fact types* as the basic unit of information whereas *object-role* modelling uses as basic elements object(type)s and roles, i.e., is more fine-grained. NIAM and FCO-IM have their basis in analysis of natural language text and expect those facts and fact types to be expressed as relational expressions; hence, we arrive at the *standard view for the natural language-based FOM*. Compare this with the diagrammatic objects-and-roles approach, where we have a *positionalist commitment for ORM and LISA-D diagrams*. To illustrate this and the theory of the previous section, let us consider a practical example in NORMA.

The positionalist ternary relationship of Fig. 1-A can be drawn in NORMA as shown in Fig. 2-A, where we have named the ORM roles corresponding to the shapes of the argument places. But how do we name the ternary relationship? As a first step, we could try to make a sentence out of it, e.g., ... **located between** ... **and** ... (shown in Fig. 2-B), so that whatever is assigned to the roles [*square*], [*star*], and [*roundtriangle*] fills the ellipses, *in that order*. Two sample fact types are included in Fig. 2-C, so that we now have the relational expression at the type level **Sea located between Island and Mainland** and another rendering with the roles in a different order in the relational expression **Island is separated from Mainland by Sea**; i.e., in both cases, an object $o \in Sea$ will be assigned to the role [*square*], an $o' \in Island$ to the role [*star*], and $o'' \in Mainland$ to [*roundtriangle*]. While the fact types look different and are verbalised differently, they represent the same state on how the UK, the North Sea and ContinentalEurope relate to each other, and likewise for Jamaica, the Caribbean Sea, and South America. From an RDBMS viewpoint, if we have this data in a GIS database, then the order of the columns—how the data is stored—likely will be different for the two different fact types, but they will produce the same maps because swapping the columns does not change the meaning of the relation that is stored as a table in the database. Generalising this relationship, we have a straightforward topological relation, “*betweenness*”, with three participating roles, being a [*separator*], a [*player1*], and another [*player2*], and we might as well use a verbalisation “[*separator*] **separates** [*player1*] **from** [*player2*]”. If we would have used the relational expression in each sample fact type to name the relationship, i.e., using the standard view ontological commitment, we would have generated four relationships, whereas there really is just one.

In principle, with the diagram-based positionalist approach, one can rearrange the roles of the single relationship (here: *betweenness*) and add as many readings as one likes. This is not possible with a verbalisation-only approach and interface. Moreover, that easily can generate duplicate fact types in larger conceptual data models¹. We analyse this in more detail in the next two sections, with an emphasis on FOM’s formal foundations, modelling methodologies, and consequences in ‘peripheral’ tools and usage.

¹ To mitigate this problem, one could add a script in a CASE tool that checks if there is already a fact type with exactly the same participating object types and to warn the modeller that the new fact type might be represented in the model already.

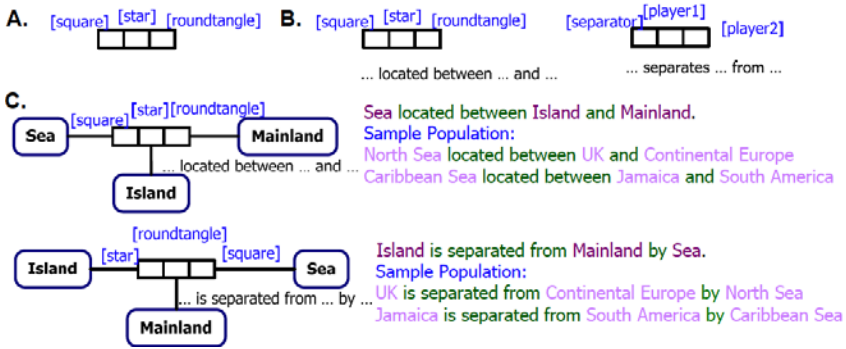


Fig. 2. Positionalist examples in ORM. A: an ORM diagram rendering of Fig. 1-A; B: a reading added and a possible generalization of it; C: sample fact types and populations.

3.1 FOM’s Basic Elements

The different commitments can be observed in the formalizations of FOM.

The first formalisation of ORM/NIAM by Halpin [7] does not include a specific kind of element for the roles: the diagrams do have roles, but the formalisation uses relationships only. That is, without loss of generality, a binary fact type is formalised with one axiom “ $\forall xy(xRy \rightarrow Ax \& By)$ ”. Halpin notes that “[w]e regard our ordered predicate notation and the unordered role notation as different only in their focus... In both approaches we see a linguistic structure in which objects play various roles with respect to the verb or relationship” ([7] p4-3), which, in the light of recent developments in Ontology, is not merely a different focus, but they reflect different ontological commitments.

Ter Hofstede and coauthors introduce a reification with predicators in PSM by introducing a finite set \mathcal{P} of predicators in its signature, which are roles in ORM and are called connectors in LISA-D [8]. In a follow-up paper [9], the authors still reify the argument places and put in the signature of the language a finite set \mathcal{R} of roles, a function $Roles : \mathcal{F} \mapsto \mathcal{R}^+$ that “provide[s] a partition of roles over the relationship types”, relationship types, and several auxiliary functions. They make explicit a distinction between the surface reading and the “deep semantics” that is formalised by taking an unambiguous positionalist stance. This commitment is also taken by [10], given that all and only those languages in the \mathcal{DLR} family of Description Logic languages are positionalist thanks to the explicit “DL role elements” that correspond to ORM roles. That is, as one of the concept constructors in the \mathcal{DLR}_{ifd} language we have $[\$i]\mathbf{R}$ where i denotes a component of the relationship \mathbf{R} ; e.g., a typed relation *loving* between two classes C and D is represented as $loving \sqsubseteq [lover]C \sqcap [beloved]D$. Note that if one were to use, say, OWL-DL as a formal foundation for ORM, then one does not have an equivalent for ORM roles.

Jarrar and Meersman [11] use “lexons” in their ORM-as-ontology-language where “[a] lexon is a binary relationship between context-specific linguistic terms, or in other words, a lexical rendering of a binary conceptual relation.”, denoted

with $\langle \gamma : T_1, r, r', T_2 \rangle$, and have it that “ r and r' are lexicalizations of the pair roles of a binary conceptual relationship R ; the role r' is the inverse of the role r ”. The first order logic axiomatization, however, does not comprise relationships at all, but only roles, and equivalence of the two participating roles is asserted in a way such that the roles are treated as binary relations (“ $\forall x.y r(x, y) \leftrightarrow r'(y, x)$ ”). This duality of roles being treated as relationships at the same time does not aid disambiguation of what the ontological nature of relations and argument places are, although the overall direction in [11] is based on the standard view.

Considering an implementation that has interfaces for both the verbalisation and the diagrams, the ORM CASE tool NORMA [12], then one can observe in the XML serialization of ORM diagrams with their mandatory verbalisations that behind that dashboard, it uses roles explicitly. The XML schema has elements such as `<orm:Facts>` where each fact type has one or more `<orm:ReadingOrders>` and one or more `<orm:Role >` and corresponding `<orm:RolePlayer />`. Thus, from a formal language perspective, it has the positionalist’s flexibility (although the naming of the relationship is based on the verbalisation of the fact type).

Thus, in the light of positionalism and the different commitments taken with FOM approaches, it is not surprising that there is no unanimous agreement on what the basic elements of ‘the’ FOM language are. From a logic perspective, there has been a tendency to move toward the positionalist commitment by giving argument places a proper place in the formalisation.

3.2 Conceptual Analysis Methodologies

Aside from the formal semantics of the languages, the modeller’s and domain expert’s experience and perception that, or if, one can express in a language what one deems necessary for the application domain, is ultimately most important for success of usage of FOM. That some people may prefer pseudo-natural language sentences over diagrams, or vice versa, may have to do with the skills and talents of the person, the to-be-modelled informal source material, or with an unstated ontological commitment. There is a long-standing debate about representing reality versus modelling our understanding of it in natural language. With the standard view, none of the former can go in a conceptual model until one can use its terms in a relational expression. Then, in a conceptual analysis methodology based on the standard view only, in principle, the conceptual analysis stage *halts* until further notice. To the positionalist or anti-positionalist, this would merely mean adding the relationship but just not giving it a specific reading (relational expression) and *continue* with the conceptual analysis; if, or when, a verbalisation is found and agreed upon, it simply can be added. This same approach can be used in the positionalist setting when some of the domain experts want to argue about the ‘best’ terms for the relational expression: one either lets them discuss and the modeller can continue in the meantime, or the modeller lets them each have their preferred verbalisation linked to one and the same relationship. In addition, a consequence of the positionalist FOM diagram is that one should not use the asymmetric ring constraint, because a relationship is never asymmetric even though its natural language readings and the standard

view commitment may give that (false) impression. And, if one were to be an ontological purist, one cannot represent symmetric relations in FOM diagrams, which is only faithfully represented with an anti-positionalist commitment.

A separate topic, which receives comparatively little attention in FOM, is that of reuse of (parts of) conceptual data models. In this context, we mean both the reuse of fragments across conceptual data models and the reuse of relationships within one conceptual model. The latter has its counterpart in UML class diagrams as stereotyped associations, whereas within FOM the reuse of different types of part-whole relationships has been proposed [13]. Forcing this into the standard view means imposing the use of precisely those names for the verbalisation, but this could create artificial debates: when a domain expert insists that something must be **made of** something else and does not like **constituted of**, there really is no debate about the meaning of the relation but only about the terms used. Using the positionalist commitment, there is no such problem and, moreover, simplifies use and reuse of ontologically well-founded relationships in conceptual models, which, in turn, improve the quality of conceptual data models. Concerning reuse of fragments across conceptual models, there are currently no easy to use ways in extant FOM CASE tools to deal with conceptual model design patterns, e.g., a **company** pattern that has several fact types about companies (that have a name, address, status, and so forth). With a methodology based entirely on the standard view, the chances are less that there are actually any design patterns due to the plethora of possible verbalisations that are used for the names of the relationships. The chances are obviously better with the positionalist and anti-positionalist commitments, where such reused fragments can be dressed up with preferred fact type readings afterward anyway. Taking this beyond design patterns and specific relationships, the prospects will be better also for conceptual model-based data integration, including the linking of FOM diagrams with ontologies, because the real relationships will be analysed by the modellers and domain experts during the modelling stage, and not later on attempted to be reconstructed by, often, other people who reuse legacy material.

If one adheres strictly to the two commitments, then FOM tools that support both text-based and diagram-based approaches are ontologically incompatible. Obviously, one can add fact type readings in the diagram to limit the possible orderings of the roles in the relationship, and to somehow force the positionalist stance with the diagram into the straightjacket of the standard view. Or demand as many readings as the domain expert can think of in the standard view-based textual interface, or to add a template sentence to be filled-in, such as “**the role played by X is called ...**” to introduce some notion of the positionalist stance. However, no matter which one is added to the tool, we are hopping on two legs in the overall framework of FOM. One can represent more, and more precisely, the universe of discourse when one takes the positionalist view compared to the standard view. Put differently: the standard view puts constraints on the use of the positionalist way of dealing with relations and forces the order of the roles into those sequence(s) that can be verbalised even though other permutations are logically and ontologically valid. On the other hand, the positionalist comes

with extra representational baggage in the formalization compared to a formalisation only for the standard view, but the formalisation is hidden to the modeller and can be dealt with easily computationally anyway. Moreover, a positionalist representation of the universe of discourse can deal fully with the standard view, but not vice versa. In the same line of thinking, one could argue to also leave behind the positionalist commitment and go for the anti-positionalist one, but the positionalist one has distinct advantages in the representation of role-constraints such as role exclusion, role subset, and multi-role frequencies. For the latter argument, however, it is important to appreciate the difference between what is, according to philosophers, the ontological nature of a relation versus what is a more convenient way of representing relationships and their constraints in a modelling language and the subsequent engineering use of role names to generate software code or a physical database schema.

3.3 Consequences on Other Usage of Conceptual Models

FOM is not necessarily used in isolation. Transformations to other conceptual modelling languages with respect to positionalism is straightforward with the positionalist object-role modelling and UML class diagrams, but results in a subset of possible EER diagrams. If one were to take the standard view for FOM, then transformations result in a subset of possible conceptual models for both UML and EER. When representing an EER diagram in ORM, an issue may arise in ORM to actually verbalise the relation, because putting the relation in a near-natural language sentence is neither part of EER's modelling methodology nor of its modelling tools. Enforcing a standard view representation nevertheless might mean either splitting the relation so that the new fact types are verbalizable (be that conceptually justifiable or not) or not representing it at all. A similar problem arises with reverse engineering a physical database schema.

A more challenging problem is realizing the idea of going from natural language texts, such as public administration documents, to an ORM diagram. There can be many relational expressions for a single relation or relationship, therefore, in principle, one will not have a 1:1 correspondence between the text and the conceptual model. At best, text mining a natural language document yields a set of verbs that are *candidates* for relations and relationships. Further, both the positionalist and anti-positionalist commitments make dealing with multilingual documents and conceptual models easier, because these approaches also take the position that there is a difference between the linguistic relational expression and the relation itself, alike [8,9,10] make a clear distinction between “surface semantics” versus “deep semantics” and in ([14] p36) between “fact type expression” and “elementary fact type”. That is, there is one natural language-independent relationship and many verbalisations in many natural languages. This does not solve multilingual template-based verbalisations of ORM diagrams fully [15], but reduces the problem by enabling to identify better the natural language-independent, reusable, information versus the (near-)natural language sentences the different stakeholders may prefer.

4 Conclusions

The underlying ontological commitments concerning the positionalism of relation(ship)s in FOM differ, being the standard view and positionalist commitments. While the verbalisations are certainly of the former type and the diagrammatic interface of the latter, this is not always consistently used as such for extant formalisations of and tools for FOM. Such incompatibilities affect transparency, hamper transition between the two approaches, and influence the modelling methodologies, and model use and reuse. Arguments for and against the two commitments have been discussed, which then leave the options either to adopt the limited standard view only and remove the flexibility of arbitrary role positions in FOM diagrams, or to accept fact types that may not be verbalisable. The latter has the advantages that it facilitates data integration, conceptual model interoperability, reusability of FOM models, and model quality enhancements through use of ontologically-well founded relationships.

References

1. Williamson, T.: Converse relations. *Philosophical Review* 94(2), 249–262 (1985)
2. Fine, K.: Neutral relations. *Philosophical Review* 109(1), 1–33 (2000)
3. van Inwagen, P.: Names for relations. *Philosophical perspectives* 20(1), 453–477 (2006)
4. Leo, J.: Modeling relations. *Journal of Philosophical Logic* 37, 353–385 (2008)
5. Cross, C.B.: Armstrong and the problem of converse relations. *Erkenntnis* 56, 215–227 (2002)
6. Object Management Group: Superstructure specification. Standard 2.1.2, Object Management Group (2007), <http://www.omg.org/spec/UML/2.1.2/>
7. Halpin, T.: A logical analysis of information systems: static aspects of the data-oriented perspective. PhD thesis, University of Queensland, Australia (1989)
8. ter Hofstede, A., Proper, H., van der Weide, T.: Formal definition of a conceptual language for the description and manipulation of information models. *Information Systems* 18(7), 489–523 (1993)
9. ter Hofstede, A.H.M., Proper, H.A.: How to formalize it? formalization principles for information systems development methods. *Information and Software Technology* 40(10), 519–540 (1998)
10. Keet, C.M.: Mapping the Object-Role Modeling language ORM2 into Description Logic language \mathcal{DLR}_{ifd} . KRDB Research Centre Technical Report KRDB07-2, Free University of Bozen-Bolzano, Italy (2007) arXiv:cs.LO/0702089v1
11. Jarrar, M., Meersman, R.: Ontology Engineering - The DOGMA Approach. In: Dillon, T.S., Chang, E., Meersman, R., Sycara, K. (eds.) *Advances in Web Semantics I*. LNCS, vol. 4891, pp. 7–34. Springer, Heidelberg (2008)
12. Curland, M., Halpin, T.: Model driven development with NORMA. In: Proc. of HICSS-40, p. 286a. IEEE Computer Society Press, Los Alamitos (2007)
13. Keet, C.M.: Part-whole relations in object-role models. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1116–1127. Springer, Heidelberg (2006)
14. Bakema, G., Zwart, J.P., van der Lek, H.: Volledig Communicatiegeoriënteerde Informatiemodellering FCO-IM. Academic Service (2005)
15. Jarrar, M., Keet, C.M., Dongilli, P.: Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Starlab technical report, Vrije Universiteit Brussel, Belgium (2006)