Conceptual Graphs with Relators and Roles
A GFO Coined View onto CG's Relations

Alexander Heußner

Research Group Ontologies in Medicine
IMISE, University of Leipzig
Härtelstrasse 16–18, D-04107 Leipzig

LaBRI, Université Bordeaux 1
351 cours de la Libération
F-33405 Talence cedex

alexander.heussner@labri.fr

Abstract. The importance of relations for conceptual modelling motivates an evaluation of Conceptual Graphs (CG) in this respect. This analysis is presented on the formal ontological basis provided by the General Formal Ontology (GFO). On the basis of a simple example domain, modelling problems are identified and analyzed in connection with more sophisticated relational concepts like roles, relators, and player universals.
This leads to a proposal for enhancing CGs and their diagrammatic modelling framework in order to capture the example domain more adequately. The newly introduced Conceptual Graphs with Relators allow for expressing roles and relators and help to clarify the ambiguous translation of classical CG relations to relators and roles.
From a more general point of view, the overall approach provides an example of applying formal ontological theories in the meta-analysis of modelling language semantics.

The role of formal ontology in today's scientific discourse on conceptual modelling cannot be neglected: it is perceived as both the panacea regarding the future goal of incorporating a more tight semantic basis into modelling as well as an appropriate tool for a large variety of bread-and-butter modelling tasks.

The following approach will focus the application of formal ontology in making expressiveness problems explicit that occur in practical modelling with Conceptual Graphs (CGs). The investigation will center around a simple, but non-trivial modelling example: a concrete act of lending a book and the abstract definition of the underlying trust relation.

After introducing the example domain and applying CGs to achieve a first, simple diagrammatic representation that does not suffice to represent the domains richness, the next steps lead to a formal ontological; this will allow to make the requirements explicit that previously classified these first concept graphs as "unsatisfactory": the absence of relator concepts and roles. As these demands are unsatisfiable in the standard CG modelling paradigm, an enhancement of CG will be proposed that will allow to give a concise model of the example domain.

The following (meta-)investigation will combine two well worked-out fields in order to solve a practical modelling problem. The feedback from formal ontology will prepare the introduction of a novel enhancement of CGs with relators and roles; these will allow for easier application of these graphs in the modelling of relations as well as provide a (semi-)formal semantics which will eliminate misunderstandings regarding the classical CG relations.

Conceptual Graphs

From the large variety of approaches towards Conceptual Graphs (CG), the following discussion will favour the formalization of Frithjof Dau as *Simple Concept Graph with Cuts* [1], especially regarding their semantic foundation in Formal Concept Analysis (FCA). The modelling paradigm will be based on the framework introduced by John Sowa in his classical CG-bible which presented a formalized way to introduce new concepts (*conceptual abstraction* [2, p. 104]) and relations (*relational contraction* [ibid.]) as well as focused the role of the accompanying *ontology*, i.e., the subsumption hierarchy that comes along with each CG.

The existing extension of CGs with *link types* [3] will play an important role when enhancing the basic conceptual graphs to a more fine-grained methodology to model relations.

General Formal Ontology

The General Formal Ontology (GFO) is a top-level formal ontology (also known as upper level or core ontology) and part of the ontological framework which is being developed by the Research Group Onto-Med at the University of Leipzig[1].

GFO is chosen as ontological background for this work because of is subtle modelling of relations and roles [6][7], which makes it stand out from the large variety of other formal ontologies and which will be briefly introduced later in Sect. 2; a general introduction can be found in [4] as well as a meta-theoretical approach towards its underlying layered architecture in [5].

1 Introducing the Practical Example

The following sections will utilize the CG framework to model a practical example domain: the situation of trust as formalized by Coleman and Buskens [8][9]. Initially, this domain will be presented with the help of a prototypical situation (lending a book) and an abstract description mingled with a first – already slightly – formalized approach that is extracted from the above two references[2].

(Semi-formal) Definition

Trust is a quaternary relation $trust(X, Y, S, A_G)$ between two social agents X and Y, which participate together in the contextual situation S. This situation involves an action A that involves a good G belonging to X and which is currently at the disposal of Y.

The relation *trust* reads: "X trusts Y in the situation S to apply action A_G".
Normally, the action lies a certain amount of time in the future which accounts for the risk the trustor must take. The relational roles of X and Y will be labelled `trustor` and `trustee` [3].

Example

This relation holds in the situation of lending a book, i.e., lending is a special case of trust (by adding additional constraints on X, Y, S, A_G and their interrelation). The two agents are the person lending the book (a Mr. Norrell), called lender, and the borrower (Mr. Strange) who is trusted to return the book (A_G) – a book that has the id 314 – after a certain amount of time.

Fig. 1 introduces graphs that try do model the concrete example above with proceeding complexity: starting from trust as a simple dyadic relation between concrete persons ($\mathfrak{G}_1$), the object of trust and its relation to the participating agents is introduced ($\mathfrak{G}_2$ and $\mathfrak{G}_3$). Leaving aside for a moment the modelling of the action and its embedding in time which would require advanced temporal modelling techniques, $\mathfrak{G}_3$ is lacking the assignment of the roles which describe the positioning of the related persons towards the relation.

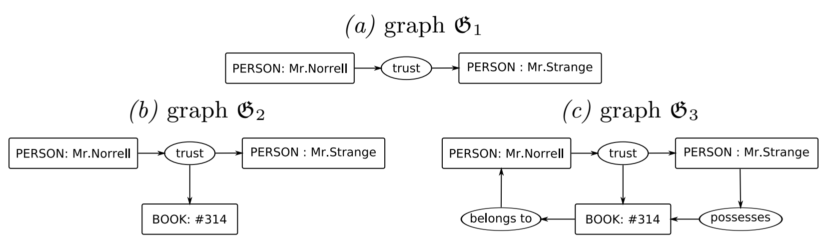

Fig. 1: A first Approach to Modelling the Example Domain

Further, the problem of how to describe (meta-)relations between relations arises when trying to establish the specialization of `trust` to `borrow` beyond simple subsumption.

In order to make these modelling demands and their possible inexpressiveness with CGs explicit, a more detailed view onto relations as entities sui generis is inevitable.

[1] `http://www.onto-med.de`

[2] As worked out more detailed in [10], every conceptualization is based on a set of *pre-conceptualizations* which are often already given in a semi-formal manner.

[3] Ignoring the discussion whether roles and concepts share the same type hierarchy [11], role names will be written like concept names in a monospaced font.

2 Approaching Relations from Formal Ontology

> "*Relations are very peculiar entities; [...] [Many philosophers] have thought that relations are nothing other than the relata and their features or that they are merely appearances. But others have conceived relations as the very stuff from which the world is ultimately constituted.*" [12, p. 58f]

Regarding this quotation of Jorge Garcia, relations are basic entities that heavily depend on the underlying general, ontological paradigm. From the variety of different approaches to formalize relations, GFO's relator model will be introduced in the following as it includes a very subtle approach towards relational roles.

2.1 GFO's Relations and Relators

In brief, GFO relations "*bind [a finite number of] things of the real world together*" [4, p. 33]. These are the *relata* of the relation and their number is the *arity* of the relation. Moreover, the relata can play the same or a different role in the context of the relation. Relations exhibit a categorial character, i.e., they generalize a kind of entities which form the "glue" among other entities. In other words, a relator is the distinct entity that assigns additional capabilities to interrelated entities, these are described by the relator's roles. The crux lies in the modelling of these *(relational) roles* which describe the mediation between the arguments and the relation or relator, respectively. The (meta-)relation between the (categorial) roles of a relation and the corresponding relata is named `plays` which is subsumed by the ontological basic relation `dependent-on` because roles depend on their player and on complementary roles, viz the totality of roles involved in the relator, cf. [4, p. 33f] [6].

As relators can be seen as instantiations of (categorial) relations, the corresponding roles of a relator are instances of a relation's (categorial) roles. Fig. 2 summarizes these new aspects in an UML-style diagram which introduces the classical view[4] on relations as derivable (the entities marked by "/") from the relator or the relation, resp.; the diagram can be read bivalently as either class or object diagram depending on focussing either relations or relators. For simplicity, the following diagrams will only depict the case of dyadic relations but can be extended to arbitrary arity.

The problematic nature of roles resides in the simple fact that they are highly dynamic entities (e.g., roles can change over time, one entity can stand in two different roles to the same relator and needs to be treated differently regarding both roles), whereas the classical conceptual modelling approach prefers the dissection of a domain into more or less static and discrete entities. Therefore, roles prefer to be separated from material entities ("natural kinds") and in the following will be assumed to form a hierarchy of their own. Nevertheless, the connection of the roles's (part-of) hierarchy and the classical material subsumption hierarchy adds additional aspects to the above model.

As one of a role's most important effect is its restriction of the super-type of its player, Fig. 2 includes an abstract universal named `player universal` which can be regarded as a compositum of all the types of the objects that can be in the `plays` relation towards this role, and, hence, serves as a constraint for the type of the relatum.

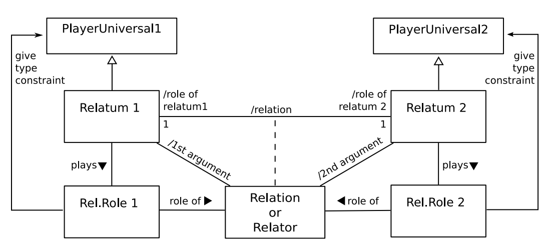

Fig. 2: Extended Diagram of GFO's Relation/Relator

2.2 Requirements to the CG Modelling Language

In the light of the preceding considerations, $\mathfrak{G}_3$ of Fig. 1 still lacks the information of the relational roles of the participants of the trust-relation. Further, as one "*not consider[s] the mere collection of the arguments which respect to a single fact [i.e., the entirety of relator and relata as instance of a relation]*" [4, p. 33], relations tend to resemble CG-concepts instead of CG-relations.

---

[4] Relators/relations are assumed to hold between the material relata and not the roles.

The following requirements would additionally underpin the choice of relational concepts: the demand to model subsumption between relations, e.g., the relation `borrow` as sub-relation of `trust` as well as the composition of relations which is not possible with CGs as only a simple, partial-ordered subsumption hierarchy is admitted [14, p. 481], and the necessity to annex a relation with additional information, like attributive properties.

Another important subject is the difference between relations that include individuals as the relata and the definition of abstract (universal) relations. As a CG-concept is related by default to the existence of an entity of that concept, this distinction does not carry weight in the following CG enhancement as – regarding the terminology of GFO – the entity representing a relation is bound to the instance level, i.e., has to be a relator not a relation.[5] However, a formal way to introduce new relational concepts via abstraction would be necessary to grasp the abstract definition of the trust relator in the example.

2.3 GFO – A more detailed Approach

As elaborated by Frank Loebe [6][7], GFO's modelling of relations has grown more subtle than the above given original approach. The following diagram and discussion is based on a personal discussion with Frank Loebe and uses an enhanced class diagram style. Instantiation is modelled via a general dependency relation ┈┈> tagged with "::" and the instantiating entities are called "individuals"; stereotypes are used to explicate the according categorial type or derived ("/") categorial names which give additional information. For example, the entities instantiating a player universal are often called "players" according to a certain "context".

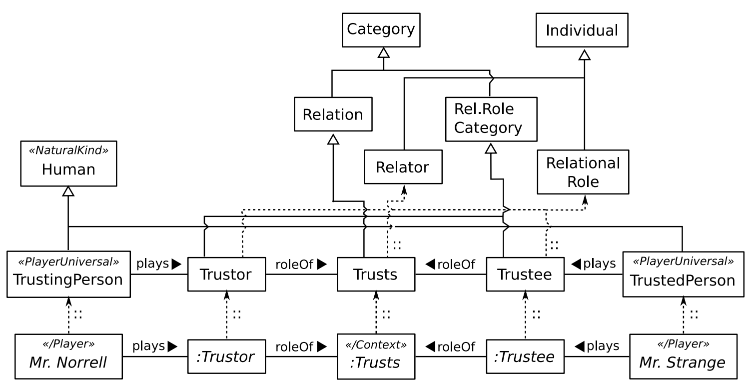

Fig. 3: The Subtleties of GFO's Relation Model

An important change to the previous considerations is the refinement of the definition of `player universal` as the maximal type constraint of a role bearing entity into a class; this step lifts a role-player from the instance level and will be called `(role) player universal`. This class is accompanied by a `natural kind` that constrains the types of the role-bearers.

The prototypical trust relation between two player instances takes place in the lowest row of Fig. 3: `Mr. Norrell`, as the individual entity subsumed under the player universal, plays the individual role (depicted as object) that instantiates the role category `Trustor`. Further, this role individual is in the `roleOf` association towards the relator individual that instantiates the relation `Trusts`. The important feature is the differentiation between instantiation and generalization: `Trusts` is a relation (via generalization) that is simultaneously an instance of the (meta-)category relator.

Another important distinction lies between the similarly named associations of the instance- and the categorial level: the `plays` relations between instances has another semantic grounding than the categorial relation of the same name, nevertheless they depend on each other.

A general, abstract definition of a special relation conforming to the example domain has to give a *role base*, i.e., a relation (`Trusts`) with its relational roles (`Trustor`, `Trustee`) and the natural category which the according player universal specializes (both are `Persons`). The

[5] The different modes of defining the referrent of a concept node would allow to approximate an abstract entity by the general referent $*$, for example in [$\top : *$] ("something").

differentiation between role and class types is hidden behind the demand of a player universal to subsume natural kinds contrary to relational roles.

3 CG with Relators and Roles

By recapitulating the previous excursion into an ontological theory of relations, the following requirements towards the expressiveness of the modelling language can be extracted: it should be able to represent roles distinct from the entities of role-bearers, as well as relators between roles and meta-relations between these relators; further, one needs to introduce new relators in an abstract way (like a role base), and, as player universals are rather complex abstract entities, to express at least their effects as type restrictions on the role-bearers.

3.1 Relators

As already explained above, the mixture of relation and object hierarchies, i.e., relation concepts and classical CG concept, must be avoided. Therefore the approach of Ribière et al. [3], which was originally intended to enhance the reasoning with CGs's to relationships, gives the desired separation and additionally extends CG with the *link formalism* of [15] and a new abstraction for link types.[6] The benefit of this approach becomes obvious if one regards the possibility to use links between links which would allow to deduce new information on a graph due to link-based reasoning.

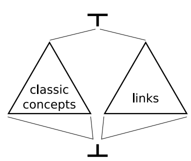

Fig. 4: Type Hierarchy

Ribière et al. proceeded as follows: first, there remains only one CG relation →(relation)→ which connects an element of the link type hierarchy with a classic concept; second, both the link type hierarchy and the concept ontology are disjointly combined into a concept lattice whereas both sub-hierarchies only share ⊤ and ⊥. This leads to the situation depicted in Fig. 4.

As there remains only one CG relation, the corresponding nodes will be omitted in the graphical representation. Further, a new style of vertices (LINKTYPE: referent) is introduced to depict link concepts. Therewith, [Concept1]–(relation)–[Link]–(relation)–[Concept2] can be shortened to [Concept1]–(Link)–[Concept2].

Hence, the approach of [3] enhances the classical CG framework with conceptualized relations, a strict separation of relation concepts and classical concepts and the possibility to express relations between relation concepts.

These improvements will allow to model the relations of the domain more fine grained than with classical CGs.

3.2 Roles

Another requirement is the possibility to name the roles of a certain relator. CG relations were already introduced as roles: "*Conceptual relations specify the role that each percept [or the concept representing this percept, resp.] plays*" [2, p. 70f]. Consequently, the graph [Concept1]→(hasRole)→[Concept2] has to be interpreted as "Concept2 plays the role described by hasRole towards Concept1"[ibid.]. A formal foundation of the approach based on `has<Rolename>` is given in [14, Sect. "Classifying Roles"] and [16].

This application of CG relations overlaps with the approach of utilizing them as conceptual relations itself. Even the original work of John Sowa did not distinguish these clearly: CG relations are applied in both ways – as roles (see above example) and relations (cf. classical "cat (being) on mat" [CAT]→(on)→[MAT] [14, p. 477]).

Besides the problem of expressing complex relations via simple role-names, this approach has the disadvantage of intermingling roles with the relator which were both assumed to be separated due to the general ontological considerations above.

3.3 Conceptual Graphs with Relators

The proposed solution will be a combination of most previously mentioned approaches to model relations: first, relators will be modelled by link types with the appropriate relator taxonomy; second, the relations of conceptual graphs model the relational roles between a (classical CG)

[6] John Sowa already introduced links and a link type hierarchy based on Aristotle's analysis of relational links but without a rigorous foundation [2].

concept and a relator; third, these roles equally form a hierarchy themselves. Therewith the requirements above are satisfied because role and concept types are separated; furthermore, relators allow reified access to the domain's relations. As the semantic foundation will not be laid down formally in detail, these new graphs will be introduced in the more readable graph theoretic way.

Definition: Concept Graphs with Relators

Concept Graphs with Relators are finite, *tri*partite, directed, not necessarily connected multigraphs $\mathfrak{G} = (V, E)$ with vertices $V = \mathfrak{C} \cup \mathfrak{L} \cup \mathfrak{R}$ and edges $E \subseteq V \times V$.

The vertices of the graph are segregated into three types: concepts $\mathfrak{C}$, relators (links) $\mathfrak{L}$, and roles $\mathfrak{R}$. An edge *walk* connects a relator node to either a concept node or a relator node via a single role node[7], hence $E \subseteq \mathfrak{L} \times \mathfrak{R} \cup \mathfrak{R} \times \mathfrak{C} \cup \mathfrak{L} \times \mathfrak{L}$; additionally, there are no other edges than those participating in a walk, and walks do not cross in roles, i.e., the degree of role vertices is always two.

The special role named `hasRelatum` is the maximal element of a lattice-order $\leq_{\mathfrak{R}}$ on the roles. Further, both concepts and relators form a lattice-order $\leq_{\mathfrak{C}}$ / $\leq_{\mathfrak{L}}$ with maximal element $\top_{\mathfrak{C}}$ / $\top_{\mathfrak{L}}$. These two orders are combined into a single lattice with an additional element $\top$ such that $\top \leq_{\mathfrak{R}/\mathfrak{L}} \top_{\mathfrak{R}/\mathfrak{L}}$ serves as new maximal element whereas the bottom elements coincide $\bot = \bot_{\mathfrak{C}} = \bot_{\mathfrak{L}}$.

Fig. 5 depicts the three defined lattices for concepts, relators, and roles. This trisection allows to apply the classical CG procedures of definition: new concepts and relators can be defined via conceptual abstraction, whereas relational contraction is applied to define roles. The maximal element of the (relational) role hierarchy is `hasRelatum` which serves as a default designator for every concept that is attached via a walk to a relator.

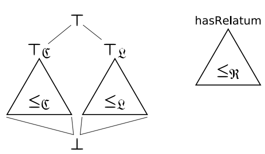

Fig. 5: CG with Relator's Three Type Hierarchies

Regarding the formal semantics of this approach, the only new entities are roles. As with standard CGs, classical concepts and relation concepts are mapped to FCA concepts of $\mathbb{K}_0$ and $\mathbb{K}_{n>0}$. Therefore, the resulting *partial* semantics which ignores roles, i.e., just assumes the top role `hasRelatum` and interprets it as a graphical feature only, embeds into Dau's FCA approach [1]. Advantageous to the mathematizations of Sowa and Dau, concepts and relations now share a common lattice analogous to their underlying semantics structures, i.e., formal power contexts, which did not separate $\mathbb{K}_0$ and $\mathbb{K}_{n>0}$ explicitely either.

The crux resides in the lack of a formal model of roles, which would require further investigative analysis. Nevertheless, reckoning roles as syntactic sugar only, Concept Graphs with Relators allow to describe real world relations more naturally (compared to current conceptual modelling paradigms) than the standard CG approach which does not allow for the presented subtle differences based on the ontological background of relations.

Additionally, the CG framework's notion of conceptual abstraction [2, p. 104] has to be extended to relators; and, hence, will allow to give an abstract definition of a relator, e.g., a general definition of `trust`. The next section will introduce this technique by example while approaching the domain of trust with the new formalism of Concept Graphs with Relators.

4 Resuming the Trust Example

Fig. 6 shows a possible graph with relators that extends $\mathfrak{G}_3$ of Fig. 1. Regarding the abstract approach towards trust of section 1, the exemplary situation needs a generalized foundation, i.e., a definition of the `Trust` relator which is conform to the above general presentation. This generalization – called *relator type abstraction* – will be introduced by example in Fig. 7.

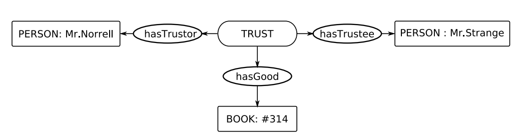

Fig. 6: Example Domain as CG with Relator

[7] Without a formal semantic basis of roles, roles between two relators seem dispensable and will be omitted; nevertheless, these entities could describe a new kind of object which could turn out to be useful in conceptual modelling.

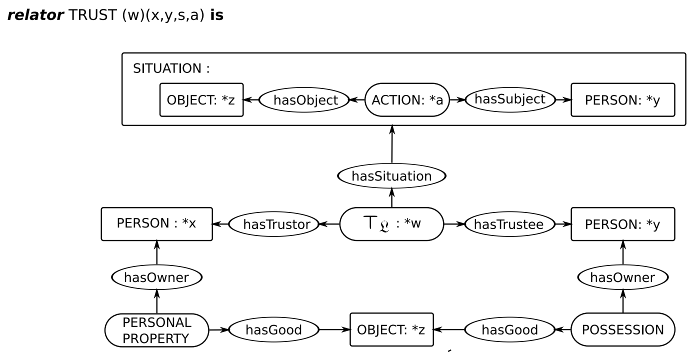

Fig. 7: Defining the Abstract Trust Relator

The heart of the abstraction are two types of coreference: first, w refers to the definiendum but further allows to include subsumption by giving a type more special than $\top_{\mathcal{L}}$ (viz. later Fig. 8 which derives `borrow` from `trust`); second, the (free) variables x, y, s, and a are the relator's arguments whose roles are given by role vertices and whose player universal is given by the type of the corresponding concept node. Thus, the argument x plays the role `hasTrustor` towards the definiendum w and must be an object of type `PERSON`. Therefore, player universals are hidden and only their effect of constraining concept type subsumption is represented. In the spirit of [2] this can be formalized as:

Definition: Relator Type Abstraction

A relator type abstraction written "**relator** $R(r)(a_1, \ldots, a_n)$ **is** $\mathfrak{G}$" declares a new relator $R \in \mathfrak{L}$ of arity n which is given by the $(n+1)$-adic abstraction [2, Def. 3.6.1] of the form $\lambda\, r, a_1, ..., a_n : \mathfrak{G}$ whereas the concept graph $\mathfrak{G}$ includes one relator node r (the definiendum) representing R which is related via roles to concept nodes a_1 to a_n whose type expresses the constraints by the according player universal of the role bearer. The type of r can be used to inherit an already defined relator or set to $\top_{\mathcal{L}}$.

To conclude, the simple `borrow` relation which was mentioned as a prototypical example of a trust relation can be formalized on top of the above relator abstraction as in Fig. 8 whereas the epistemic relators and the (temporal) sequence have to be read "intuitively" without an accompanying, appropriate CG ontology. Thus, this graph highlights the transition from a situation in which the trustee possesses the object to a situation in which the trustor believes that this object has been returned.

Hence, relator type abstraction allows to introduce new, complex relator which derive from already existing ones by giving a role base and additional constraints beyond simple subsumption.

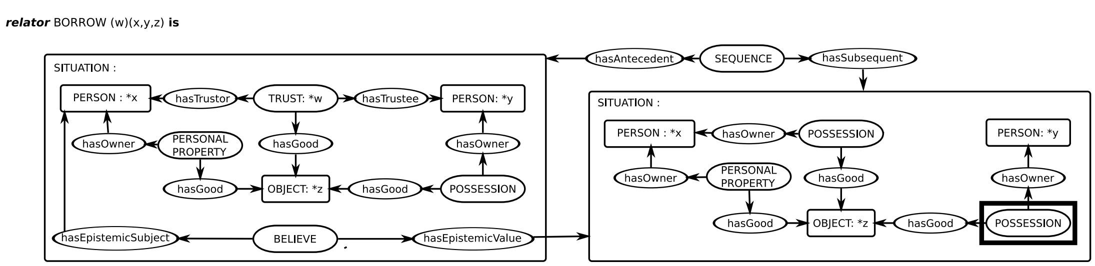

Fig. 8: Additionally Defining the `borrow` Relator

5 Summary

The previous sections are an example for the utilization of a formal ontology to the task of making the differences between a formal semantics and the semantics intended by the engineer explicit, as well as to feed back these results into an appropriate enhancement of the modelling language.

The ontologically coined view allows to express a catalogue of requirements that one would want to express when trying to represent the trust domain (or domains including relations and roles in general). As the standard CG framework does not provide the necessary features to express these demands (particularly relators and roles), an example-tailored enhancement of

CGs is introduced as Conceptual Graphs with Relators which fulfills both the requirements of modelling certain aspects of the domain (instance level description and abstract introduction of relations) as well as the catalogue derived from a closer look onto relations via GFO.

The choice of a particular underlying ontological approach influenced the enhancement as it mirrors the basic distinctions of GFO in the CG framework. Consequently, applying another core ontology could have resulted in another way of enhancing CGs. For example, emphasizing the dynamic aspect of roles could have lead to the field of Dynamic Conceptual Graphs and Actor Models [17] instead of the underlying FCA-based formalism; whereas the latter includes a mathematical rigour close to GFO's own. Another approach could have been to "hide" the representation of relators and roles in the accompanying ontology of the concept graph, whereas the given solution decides to include these directly into the modelling language itself and thereby closer to the concrete task of diagrammatic modelling.

There are several aspects which require additional consideration: first, a complete formal semantic foundation of Conceptual Graphs with Relators by introducing an appropriate (power context based) model for roles; second, the given interplay between formal ontology and the modelling language can only be seen as first cycle of a "circulus creativus"[10, p. 129] and would require additional feedback via modelling further examples with this extended graph formalism; third, a comparison to the large field of other CG based extensions, starting from the above mentioned dynamical extensions.

To conclude, applying GFO to support the semantic meta-language analysis of the (diagrammatic) modelling language of Conceptual Graphs has proven to be another bread-and-butter task that can be facilitated with the help of formal ontology, and proved to be a first step of combining GFO and the CG framework.

Acknowledgements

First and foremost, I am indebted to the Research Group Onto-Med and H. Herre who allowed me to write my thesis about the semantic foundation of diagrammatic modelling languages of which the previous considerations were a minor excerpt [10]; especially, Frank Loebe who spent long discussions to inaugurate me into the subtle distinctions of GFO's relations and roles; further, I am glad for the pointers given by the anonymous reviewers which encouraged me to approach the given solution from different points of departure; last, S. Clarke for the names of the two prototypical bibliophiles used in the main example.

References

1. Dau, F.: The logic system of concept graphs with negations and its relationship to predicate logic. Springer (2003)
2. Sowa, J.F.: Conceptual Structures – Information Processing in Mind and Machine. Addison-Wesley (1984)
3. Ribière, M., Dieng, R., Blay-Fornarino, M., Pinna-Dery, A.M.: Link-based reasoning on conceptual graphs. In Mineau, G.W., Moulin, B., Sowa, J.F., eds.: Proceedings of ICCS. Volume 699 of LNCS., Springer (1993) 1–35
4. Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., Michalek, H.: General Formal Ontology (GFO) – A foundational ontology integrating objects and processes [Version 1.0]. Onto-Med Report 8, Research Group Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig (2006)
5. Herre, H., Loebe, F.: A meta-ontological architecture for foundational ontologies. In Meersman, R., Tari, Z., eds.: Proceedings of CoopIS / DOA / ODBASE 2005. Number 3761 in LNCS, Springer (2005)
6. Loebe, F.: Abstract vs. social roles – Towards a general theoretical account of roles. Applied Ontology **2**(2) (2007) 127–158
7. Loebe, F.: An analysis of roles: Towards ontology-based modelling. Onto-Med Report 6, Research Group Ontologies in Medicine, Leipzig University (2003).
8. Coleman, J.S.: Foundations of Social Theory. Belknap Press of Harvard University Press (1990)
9. Buskens, V.W.: Social Networks and Trust. PhD thesis, University Utrecht (1999)
10. Heußner, A.: Semantic foundation of diagramamatic modelling languages – applying the pictorial turn to conceptual modelling. Diploma thesis, University of Leipzig, Institute for Mathematics and Computer Science (2007) `http://www.onto-med.de/en/publications/diploma-theses/heussner-a-2007--a.pdf`
11. Steimann, F.: On the representation of roles in object-oriented and conceptual modelling. Data Knowledge Engineering **35**(1) (2000) 83–106
12. Gracia, J.J.E.: Metaphysics and Its Tasks: The Search for the Categorial Foundation of Knowledge. SUNY series in Philosophy. State University of New York Press, Albany (1999)
13. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Volume 05-74 of Telematica Instituut Fundamental Research Series. Telematica Instituut, Enschede (Netherlands) (2005)
14. Sowa, J.F.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooks/Cole, Pacific Grove (2000)
15. Fornarino, M., Pinna, A.M.: Expressions des relations et maintien de la cohérence: le concept de lien. Research Report 1346, INRIA (1990)
16. Sowa, J.F.: Roles and relations. `http://www.jfsowa.com/ontology/roles.htm` last visited: 29. 12. 2007, last modified: 04. 07. 2001.
17. Lukose, D., Mineau, G.W.: A Comparative Study of Dynamic Conceptual Graphs. Proceedings of KAW 1998. `http://ksi.cpsc.ucalgary.ca/KAW/KAW98/lukose/` last visited: 15. 05. 2008.