Using abstractions to facilitate management of large ORM models and ontologies

C. Maria Keet

Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bozen-Bolzano, Italy, phone +39 04710 161 28, keet@inf.unibz.it

Abstract. Due to ever larger ORM models and ORM-represented ontologies, information management and its GUI representation is even more important. One useful mechanism is abstraction, which has received some attention in conceptual modelling and implementation, as well as its foundational characteristics. Extant heuristics for ORM abstractions are examined and enriched with several foundational aspects of abstraction. These improvements are applicable to a wider range of types of representations, including conceptual models and ontologies, thereby not only alleviating the Database Comprehension Problem, but also facilitate conceptual model and ontology browsing.

1 Introduction

As a result of ever-increasing company size and complexity, the possibility increases to encounter the Database Comprehension Problem: the difficulty to understand and manage large conceptual models. Similarly, the size of ontologies increases (e.g. the Gene Ontology [28] contains about 18000 entities and the Foundational Model of Anatomy (FMA) [27] 72,000) and with ontology integration and formalisations, an Ontology Comprehension Problem emerges. Yet often only a *section* is of interest: a simplified higher level of granularity such as the GO slims [29], fact&entity finding where the query answer contains only the adjacent and high-level elements, or using a small selection of entities of an ontology when developing an ontology-inspired conceptual model, where zooming and abstraction simplifies the user's actions. Manual contextualisation of ontologies, e.g. with DOGMA [12] or C-OWL [2], can alleviate the problem of manageability and understandability, but they do not provide simplified views of the underlying complex model nor a 'zooming in'. One can apply levels of granularity to organise the conceptualisation, but it has to be pre-defined and is a static structure. In contrast, *abstraction* is the process to go from complex to simpler representations, which we focus on. Section 2 contains theoretical aspects of abstraction, in §3 assessed and compared with ORM abstraction heuristics as developed by Campbell et al [3] using different models and abstraction mechanisms. We discuss results, integrate it with the theoretical aspects of abstraction, and propose improvements – reordering rules, maintaining rules 1-6,12, and replacing 7-11 with the generic abstractions introduced in this article – that are more widely applicable to conceptual models and ontologies for different purposes (§4). We finalise with some concluding remarks.

2 Abstractions

Two meanings of abstraction are common in the literature, which are the duality between abstraction and concretisation and abstraction by the process of ignoring details or the bigger picture. We focus on the second meaning: how to *not* take into account things that are not of interest. This is different from indistinguishability in that with the latter we *cannot* observe a difference at a certain level, whereas with abstraction we *choose to disregard* undesired aspects. What the undesired aspects are and how to ignore them depends on i) the subject domain, ii) user's perspective and context, iii) the type of abstraction, iv) the procedure of (consecutive steps of) abstraction, and v) on what type of representation/model the abstraction is performed. We focus on point iii-v, although i-ii does influence possible usage and solutions proposed in the extant literature.

Manual efforts of abstraction has been, and is being, carried out informally or in somewhat structured fashion with UML modelling, (E)ER (e.g. [11]) and the Abstraction Hierarchy (AH, e.g. [18] [24]), which are laborious and intuitive ad hoc methods. AH does not even have a supporting modelling paradigm such as UML and (E)ER and is akin the 'complete freedom' biologists have with their "black boxes" (e.g. Physiome Project [10], among many). Also ecological modelling tools, such as STELLA/*ithink* [31], still maintain relative freedom to abstract, with the consequential manual effort it requires to do so (e.g. [23] [14]). A different abstraction approach that does not suppress the details, but abstracts away that what is deemed less important because it is *non-functional* [8], is not elaborated on further here. Also, the reductionism versus holism & systems biology where the former ignores the larger systems and simplifies to its smaller (sub-)components, is left for another occasion.

The remainder of this section discusses approaches to abstraction that can be carried out automatically. Abstractions based on heuristics have been developed by [3], who aimed to simplify large ORM models by suppressing roles and objects that, based on the encoded semantics, are less relevant. This is discussed in detail in sections 3 and 4. Ghidini and Giunchiglia [6] formalised abstraction by exploiting Local Model Semantics of context reasoning and formalising abstraction "as a pair of (formal) languages plus a mapping between them", using the *abs* function for syntax mapping. Given the ground language L_0 and the abstract language L_1, symbol abstraction operates on constant symbols (comparable to ORM objects): $c_1, ..., c_n \in L_0, c \in L_1$ and $abs(c_i) = c$, for all $i \in [1, n]$. Idem ditto for function symbols with $abs(f_i) = f$ and predicate symbols $abs(p_i) = p$, where arity abstractions on functions and predicates lower the arity by one; if the arity in L_0 is 1, f_i maps into a constant and p_i into a proposition [6]. What this *abs* function actually does, is, given an entity A, to return the parent entity B that subsumes A. Using specialisation/generalisation as abstraction is a recurring theme across domains and modelling paradigms [5] [4] [21] [13] [9].

Similar to the abstraction mechanism that takes advantage of the isA relationship, is abstraction through the partOf relationship, which is not based on set theory but mereology. While this is an ontological distinction, this can be implemented set-wise by giving in a little to the ontological trade-off. In essence,

through abstraction the parts that make the whole are abstracted exploiting the partOf relation between the entities involved [1] [16] [17]. It is a point of philosophical debate if sub-processes are (a special kind of) partOf of its grander process [20] or involvedIn the grander process [16]. For example that 'I go from Rome to Bolzano' as parent process and take first the bus, then the metro, finally catch the train, and reading a book and listening to music at the same time. For purpose of clarity, we indicate these kind of more detailed sub-processes that can be abstracted away with $involvedIn(x, y)$. In an ontology, x and y are both perdurants, but in a conceptual model they are likely to occur as fact types, relations or methods.

Mani [19] introduced four types of abstraction, in addition to "type shifting operators" for grain size shifts. Type shifting goes from coarse to finer-grained with event to processes, process to states, and process to objects, and three more operations to fold processes and states and to fold events and propositions, preserving compositionality of two logical forms that are abstracted. This, then, is combined with the type shifting operators to create three non-endocentric abstractions and three endocentric (meronymic) abstractions. Pandurang and Levy [21] also emphasise compositionality and use a two-step process 1) abstraction of the intended domain model and 2) define set of formulas that formalises the abstracted domain to make the simplifying assumptions explicit in the base (more detailed) model. Like Ghidini and Giunchiglia [6], Pandurang and Levy [21] exploit the *isA* relation for abstraction. Although Mani's family of abstraction functions is developed for dealing with polysemy and underspecification in linguistics, it is a more promising approach than [6] because it captures the varying semantics of abstraction better than [6]'s pure syntactic approach. However, developing a computational implementation of the folding operations may not be easy. Multiple types of abstraction functions can be useful in particular for abstracting biological complex entities like *Second messenger system* or *MAPK signalling*. The former collapses processes such as *Activation*, *GTP-GDP exchange*, *α-subunit release*, states like *activated*, and components such as *Hormone receptor*, *G_s protein*, and *cAMP*. *MAPK signalling* is already used as a module in systems biology that at a higher level of abstraction is treated as a black box, containing (sub-)processes, input/output behaviour, parameters and their values, etc. [22]. Summarising the different usages, and options, for performing an abstraction, one can identify abstraction based on:

1. Taxonomy: moving 'up' in the specialisation/generalisation hierarchy through the isA relation, abstracting away a distinguishing property.
2. Partonomy (mereology): through the partOf relation.
3. ORM heuristics.
4. UML modularisation.
5. (E)ER abstractions.
6. The black boxes in biology, ecology models, and Abstraction Hierarchy.
7. Folding operations of different types of entities resulting in other types (perdurant into endurant etc), focussed on linguistics.
8. Syntax limited to Local Model Semantics of contextual reasoning.

Grouping these methods of abstraction into *types* of operation, then (1,2) focuses on exploiting primitive relations (relations like causality and aggregation are omitted because it is beyond the current scope or covered with existing rules (see [7]), (1-3, 5, 7) takes into consideration only the 'up' direction to a simplified level. Methods (3-5) are motivated by the database community to keep conceptual models comprehensible and manageable and (6, 7) are UoD-motivated – (1-5, 8) are domain-independent – although (3-7) all are bottom-up driven. Number (8) is syntax-based, where (3, 5) still take into account the coded semantics and thereby seem 'closer' to the domain-centred (informal) abstractions of (6, 7) than the syntax-approach of (8). Last, (1-3, 8) can, in principle, be carried out automatically without user intervention.

3 Experimentation

3.1 Methodology

The aim of the experiments is to assess Campbell et al's [3] abstraction heuristics with two distinct ORM models, and compare this with other abstraction methods.

Bacteriocins is most similar to Campbell et al's case study model because it also is an ORM model for a database (developed and in use; diagram omitted due to space limitations). On the other hand, the *Blood* ORM model (Fig.2) is a resultant of positioning orthogonally the relevant sections of the partonomy and taxonomy of the FMA [27], the Mode of Transmission perspective of infectious diseases [15], and a section of microorganism phylogeny. A similar model to *Blood* can be constructed also with *Bacteriocins* by linking it with the Gene Ontology [28], Agricultural Ontology Services [25], MetaCyc [32], SNOMED-CT [34], Bad Bug Book [26], and microorganism phylogeny. The test procedure is as follows: **A)** Take *Blood*, respectively *Bacteriocins*, and use Campbell's abstraction heuristics. **B)** Compare the result with the result based on a manual semantics analysis. Test where any of the logical and/or ontologically founded, theoretical abstraction mechanisms (points 1, 2, 7, 8 in the previous section) can be useful for improving procedure and outcome. **C)** Test other abstraction mechanisms: **C1** *UML-like modularisation*: group orthogonal sections (Anatomy, InfectiousAgent, ModeOfAction, blood process). **C2** *Modified abstraction heuristics*, taking into account above-mentioned domain-independent abstraction mechanisms: 1) Group anatomy isA: intermediate ones and leaves that are not involved in any role; 2) Group anatomy partOf: remove intermediates and leaves that are not involved in anything else than partonomy; 3) Remove non-mandatory leaves, roles, and 'dead ends' (successive leaves and self-contained subsections). 4) Reapply steps 1-3, if applicable. **D)** Apply C2 to Campbell's case study model.

3.2 Results

Before applying any heuristics, we abstracted *Blood* intuitively, based on semantics alone (Fig.1-D). Table 1 contains the summarised ORM heuristics rules

ORM heuristics	***Blood***	***Bacteriocins***
Rule 1: mandatory roles, 10 points	+	+
Rule 2: unary roles, 10 points	N/A	N/A
Rule 3: non-leaf object types, 9 points (thus, delete leaves)	+	+
Rule 4: smallest maximum frequency, 8 for uniqueness constraint, else lower	+	+
Rule 5: non-value types, 7 points (thus, value types have lower importance)	N/A	+
Rule 6: anchor points, 6 points	+	+
Rule 7: single-role set constraint, 5 points	N/A	N/A
Rule 8: multi-role set constraints, 4 points	N/A	N/A
Rule 9: set constraints and anchor points, 3 points	N/A	N/A
Rule 10: joining roles of set constraints, 2 points	N/A	N/A
Rule 11: first role of set constraint, 1 point	N/A	N/A
Rule 12: first role of internal uniqueness constraint, 1 point	+	+

Table 1. Abstraction rules and applicability to ORM models *Blood* and *Bacteriocins*.

taken from [3], with their applicability to the *Blood* and *Bacteriocins* ORM models. A salient aspect of the rules is the emphasis on treatment of role-set constraints, which, if absent as in *Blood* and *Bacteriocins*, limits the 12 rules to only 7. The weight allocation to the rules [3] were adhered to, to the extend of counting which object type had most mandatory roles and uniqueness constraints, and for other rules used in descending order. Fig.1-A shows the result after 7 iterations of *Blood* with Campbell's heuristics. With the modified rules as listed in C2 in §3.1, then a near identical abstraction emerges after 3 iterative abstractions, with the added advantage that the infection semantics is still present (Fig.1-B); after the 4^{th} iteration it is identical to Fig.1-A. The UML-like abstraction result (Fig.1-C) has the neat aspect that it compartmentalises into the granular perspectives it was originally built up from, thereby correctly closing the circle. Testing *Bacteriocins* with Campbell's rules (see Table 1), it achieved maximum abstraction after 3 iterations. The set constraints rules were left unused, but for *Bacteriocins*, rules for collapsing subsumption and partOf relations would not have been of use either, unless if *Bacteriocins* had been extended analogous to *Blood*. Applying C2 to Campbell et al's case study model, the same final abstracted model was achieved after two iterations of the rules.

4 Discussion

Campbell et al's rules focus on semantic importance *assumed* from syntax and implying that what is not meant in the rules must be unimportant. However, for instance, a mandatory role does not *imply* semantic importance in the UoD, nor does it appear in foundational aspects of abstraction: the *Disease*-related mandatory fact and types in *Bacteriocins* are important to its immediate neighbours, but in the overall semantics play only a secondary role; according to the client,

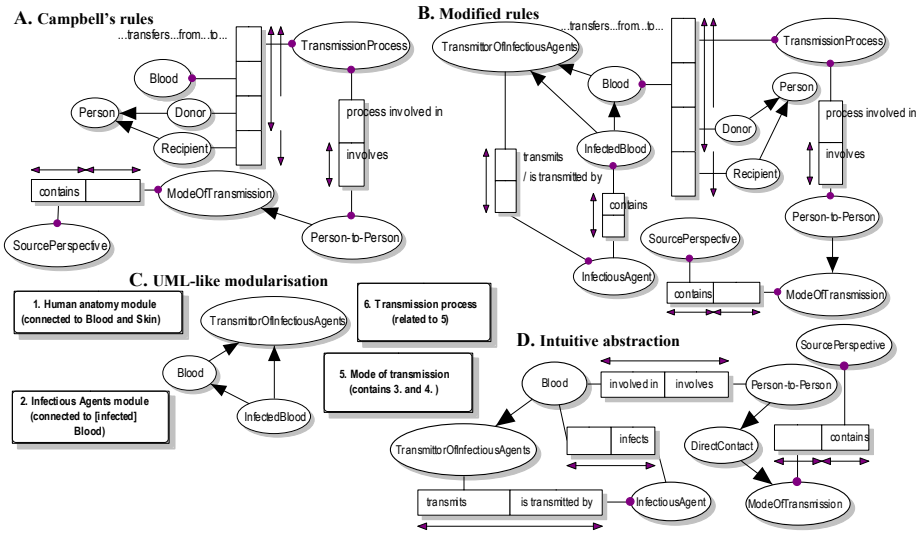

Fig. 1. A: *Blood* with Campbell's rules after 7 steps; B: *Blood* with adapted rules after 3 steps; C Manual UML-like modularisation after 3 steps (modules 3,4 abstracted within 5); D: semantic abstraction.

it had the status of peripheral nice aspect. Likewise, *FoodBorne* is irrelevant for person-to-person transmission in *Blood*. No model captures this and relying on encoded semantics will not address it either. On the contrary, if one knows the model, abstraction heuristics are influenced by such background information. Campbell et al's case study ORM model contains several role set-related rules, which are not necessarily a salient feature in ORM models and certainly not in other representation methods. If *Blood* and *Bacteriocins* would have been used as first case study instead of Campbell et al' case study, neither rules (7-11) nor the unary and mandatory ones would have been included as such, although most aspects of the remaining rules are, heuristically, useful. This illustrates limitations of bottom-up case study based approach: generalising from a bottom-up approach can require tweaking a 'generalised reusable' theory to make it more general. In an effort not to 'pollute' rules so that they only would fit the ORM case study models we brought in, theoretical notions of abstraction are useful and beneficial to improve heuristics and make them applicable to a wider range of models. Taking into account §2, rules, and automation, let x, y, and z be entities (object types, perdurants/endurants), y is at a higher level of abstraction, z related to x, and *abs* a function, then

1. isA and partOf collapsing:

$$\text{if } isA(x,y), \text{ then } abs_{isA}(x) = y \tag{1}$$

$$\text{if } partOf(x,y), \text{ then } abs_{partOf}(x) = y \tag{2}$$

$$\text{if } isA(x,y) \wedge \neg relatedTo(x,z), \text{ then } abs_{isA}(x) = y \tag{3}$$

$$\text{if } partOf(x,y) \wedge \neg relatedTo(x,z), \text{ then } abs_{partOf}(x) = y \tag{4}$$

The reason to include $\neg relatedTo(x,z)$ in (3-4) is to ensure abstracting the hierarchy by removing only intermediate layers that serve no other purpose; this is the same as Campbell's et al's "lowest common identified supertype". Note that for abstracting taxonomies and partonomies, (3-4) is of no use but it certainly is for conceptual models and for (more complex) formal ontologies. In ORM, partOf and its inverse hasPart are merely roles in a fact type, therefore an implementation either will need an additional (text string) analysis of role names, or become a separate element.

2. Additional semi-automation of predicates and role-set constraints inspired by [6]'s syntax approach. Then, in addition to $abs_{isA}(x)$ for taxonomic subsumption, for any predicate p, we have

$$\text{if } p(x,y), \text{ then } abs_{p}(x) = y \tag{5}$$

3. For processes and sub-processes a clear distinction has to be made between *subsumption* of processes for which $abs_{isA}(x)$ suffices, and sub-processes of the $involvedIn(x,y)$ type with the abstraction rule

$$\text{if } involvedIn(x,y), \text{ then } abs_{in}(x) = y \tag{6}$$

4. Complex folding, where w is a *new* object type added to the model

$$(abs_{isA}(x_1) \wedge ... \wedge abs_{isA}(x_n) \wedge abs_{p}(z_1) \wedge ... \wedge abs_{p}(z_n)) = w \tag{7}$$

Note that if one were to allow such operation, compositionality is not maintained, because it requires introduction of a new entity w. It is possible provided a higher abstraction level has been defined in an ontology and computability of the abstraction operation is required (e.g. [21]). It can function as simpler engineering solution than Mani's folding family in order to abstract, for instance, the *Second messenger system*. However, user intervention may be preferred.

5. UML-like modularisation for to modularise-able perspectives, AHs, and black boxes in biology and ecology.

Are (7), point 5, and its envisaged implementations less straightforward because it may be too challenging to capture in simple rules the complex 'flexibility' of human-performed abstraction? Businesses and their models are developed by humans – nature is not. For a biology UoD, one can resort to manual modularisation, combine it with a static structure of pre-defined granularity and use abstraction to move from contents in one level to the other, or firing a combination of rules like (7) to approximate the complexity. With the latter, one can set up separate decision trees, whereby a tool like iCOM [30] may be useful because it already contains a (Description Logics) reasoner. This will also facilitate scaling up experimentation and varying weights given to rules to make these less

dependent on only a few (ORM) models. Regardless automation of abstraction, changing the extant heuristics to better reflect semantic and syntactic richness of (ORM) models facilitates wider applicability. The proposed change in rules, in particular for ORM conceptual models, is to maintain rules 1-6,12 and replacing 7-11 with the generic abstractions introduced in this paragraph. Using the generic abstractions *only* or *before* the ORM rules can reduce the amount of iterations, in particular if (some of) the model uses (sections of) ontologies. In addition, the generic rules are useful for *both* conceptual models *and* ontologies, thereby extending the potential for consistent reuse of abstraction across different types of knowledge representations.

5 Conclusions

Existing ORM abstraction heuristics to simplify large models were examined with two distinct types of ORM models and heuristics augmented with foundational aspects of abstraction. This improvement is applicable to a wider range of types of knowledge representations, thereby has the potential to be of use not only to alleviate the Database Comprehension Problem, but also conceptual model and ontology browsing. Further experimentation with other models, automation, and possible combinations of rules with algorithms are needed if it is to perform effective abstraction of the more complex abstraction mechanisms of 'folding multiple entities' as carried out in the biology domain.

References

1. Bittner, T., Smith, B.: A Theory of Granular Partitions. In: Foundations of Geographic Information Science, Duckham, M, Goodchild, MF, Worboys, MF (eds.), London: Taylor & Francis Books (2003) 117-151
2. Bouquet, P., Giunchiglia, F., van Harmelen, F., Serafini, L., Stuckenschmidt, H.: Contextualizing Ontologies. Journal of Web Semantics (2004) 1(4):24
3. Campbell, L.J., Halpin, T.A. and Proper, H.A.: Conceptual Schemas with Abstractions: Making flat conceptual schemas more comprehensible. Data & Knowledge Engineering (1996) 20(1): 39-85
4. Degtyarenko, K., Contrino, S. COMe: the ontology of bioinorganic proteins. BMC Structural Biology, (2004) 4:3.
5. Fonseca, F., Egenhofer, M., Davis, C., and Camara, G.: Semantic Granularity in Ontology-Driven Geographic Information Systems. Annals of Mathematics and Artificial Intelligence (2002) 36 (1-2): 121-151
6. Ghidini, C., Giunchiglia, F.: A semantics for abstraction. Technical Report DIT-03-082, University of Trento, Italy (2003)
7. Halpin, T.: Information Modeling and Relational Databases. San Francisco: Morgan Kaufmann Publishers (2001)
8. Hanahan, D., Weinberg, R.A.: The Hallmarks of Cancer. Cell (2000) 100: 57-70
9. Hobbs, J.R.: Granularity. International Joint Conference on Artificial Intelligence (IJCAI85) (1985) 432-435
10. Hunter, P.J., Borg, T.: Integration from Proteins to Organs: The Physiome Project. Nature (2003) 4(3): 237-243

11. Jaeschke, P., Oberweis, A., Stucky, W.: Extending ER Model Clustering by relationship clustering. 12th International Conference on Entity Relationship Approach, Arlington, Texas (1993)
12. Jarrar, M., Demy, J., Meersman, R.: On Using Conceptual Data Modeling for Ontology Engineering. Journal on Data Semantics Special issue on 'Best papers from the ER/ODBASE/COOPIS 2002 Conferences' (2003) 1(1): 185-207
13. Kiriyama, T., Tomiyama, T.: Reasoning about Models across Multiple Ontologies. International Qualitative Reasoning Workshop (1993)
14. Keet, C.M.: Factors affecting ontology development in ecology. In: Data Integration in the Life Sciences (DILS2005), Ludäscher, B., Raschid, L. (Eds.), Springer LNBI 3615, (2005) 46-62
15. Keet, C.M., Kumar, A.: Applying partitions to infectious diseases. XIX International Congress of the European Federation for Medical Informatics. Geneva, Switzerland (2005)
16. Kumar, A., Smith, B., Novotny, D.D.: Biomedical Informatics and Granularity. Comparative and Functional Genomics (2005) 5(6-7): 501-508
17. Kumar, A., Yip, L., Smith, B., Grenon P.: Bridging the Gap between Medical and Bioinformatics Using Formal Ontological Principles. Computers in Biology and Medicine (In press)
18. Lind, M.: Making sense of the abstraction heirarchy. Cognitive Science Approaches to Process Control (CSAPC99), Villeneuve d'Ascq, France 21-24 September (1999)
19. Mani, I.: A theory of granularity and its application to problems of polysemy and underspecification of meaning. In: Principles of Knowledge Representation and Reasoning (KR98), A.G. Cohn, L.K. Schubert, and S.C. Shapiro (eds.). San Mateo: Morgan Kaufmann (1998) 245-255
20. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Ontology Library. WonderWeb Deliverable D18 (v1.0). http://wonderweb.semanticweb.org (2003)
21. Pandurang Nayak, P., Levy, A.Y.: A semantic theory of abstractions. In: Proceedings of the International Joint Conference on Artificial Intelligence, Mellish, C. (ed.). San Mateo: Morgan Kaufmann (1995) 196-203
22. Sontag, E.D.: Some new directions in control theory inspired by systems biology. Systems biology (2004) 1(1): 9-18
23. Tett, P., Wilson, H.: From biogeochemical to ecological models of marine microplankton. Journal of Marine Systems, 25 (2000) 431-446
24. Yu, X, Lau, E., Vicente, K.J., Carter, M.W. Toward theory-driven, quantitative performance measurement in ergonomics science: the abstraction hierarchy as a framework for data analysis. Theoretical Issues in Ergonomics Science (2002) 3(2): 124-142
25. Agricultural Ontology Services. http://www.fao.org/agris/aos
26. Bad Bug Book. http://www.cfsan.fda.gov/ mow/intro.html
27. Foundational Model of Anatomy. 2003. http://sig.biostr.washington.edu/projects/fm/index.html
28. Gene Ontology Consortium. http://www.geneontology.org
29. GO-slim. http://www.geneontology.org/GO.slims.shtml
30. iCOM. http://www.inf.unibz.it/∼franconi/icom
31. ISEE Systems. http://www.iseesystems.com
32. MetaCyc & BioCyc. http://BioCyc.org
33. Open Biological Ontologies. http://obo.sourceforge.net
34. Snomed CT. http://www.snomed.org/snomedct/

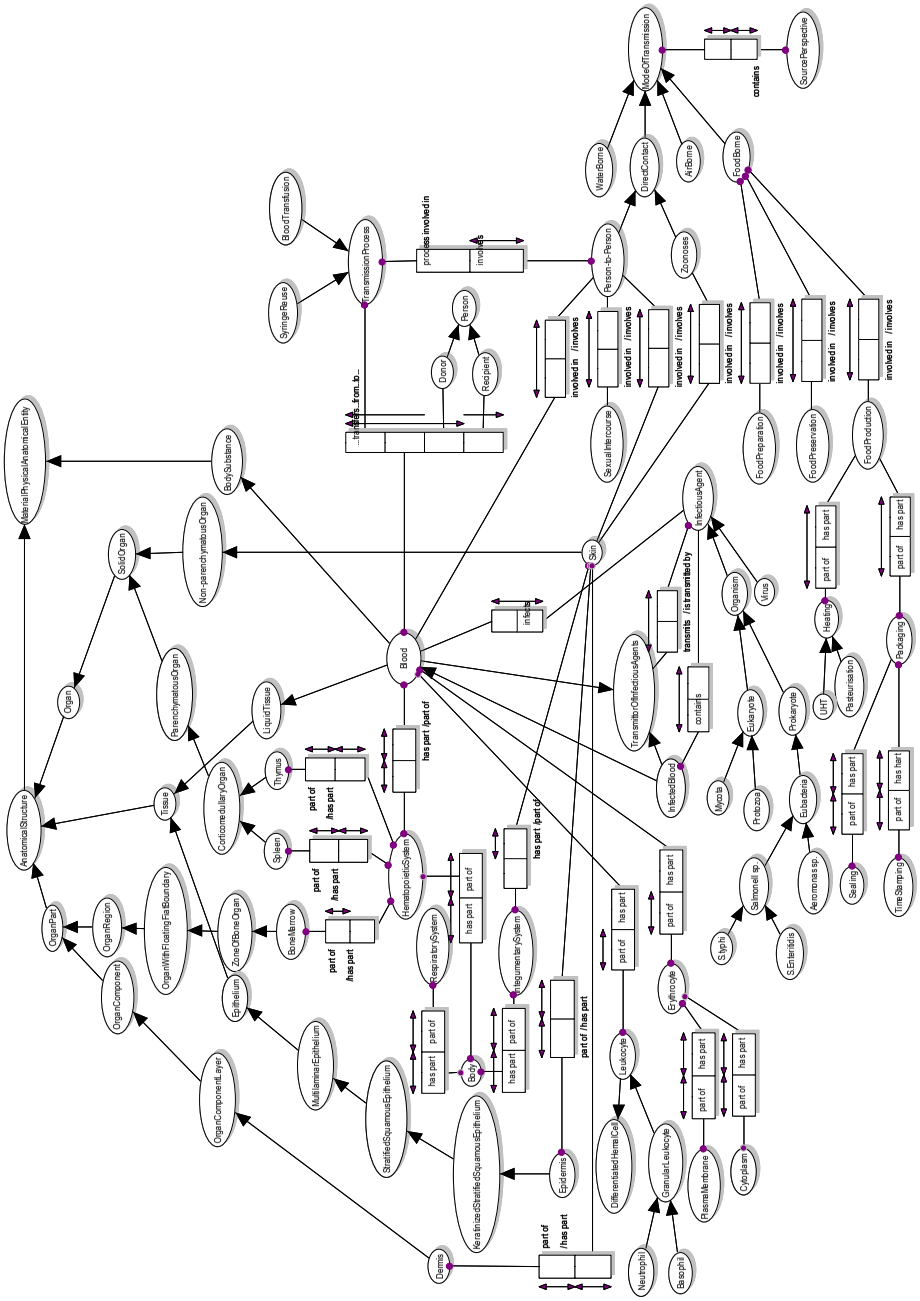

Fig. 2. The *Blood* ORM model showing the intersection of different pespectives (anatomy, phylogeny, mode of transmission) and processes of blood.