Aspects of Ontology Integration

Literature research & background information for the PhD proposal entitled:
Bottom-up development of ontologies and ontology integration
in the subject domain of ecology

C. Maria (Marijke) Keet

February 2004

School of Computing, Napier University
10 Colinton Road, Edinburgh
EH10 5DT, Scotland
Tel: +44 131 455 2773
m.keet@napier.ac.uk

Aspects of Ontology Integration

*Internal report containing literature research and additional background information to the PhD proposal entitled "*Bottom-up development of ontologies and ontology integration in the subject domain of ecology*", submitted in February 2004.*

C. Maria (Marijke) Keet

Director of Studies: Professor Jessie Kennedy
Second Supervisor: Trevor Paterson
Thesis Panel Chair: Professor Elisabeth Davenport
School of Computing
Napier University
10 Colinton Road
Edinburgh EH10 5DT
Scotland

Abstract

There are many aspects to data and domain heterogeneity increasing the possibilities of conflicts and mismatches when combining conceptual data models and ontologies, which will not be resolved easily, if ever. These include schematic differences such as aggregation and data type, diverging semantics with e.g. synonyms, homonyms, and intensional conflicts, which includes the domain heterogeneity. This is augmented with an overview of characteristics of biological data, complicating resolution of heterogeneities when integrating data and subject domains. Domain heterogeneity comprises differences in naming, scope, encoding and attribute scope, as well as modelling paradigm, ontology and content heterogeneities when viewed in a wider context. If one extends this view to ontologies, there can be identified different types of ontologies according to the level of formalism used and categorise them according to subject, such as top-level and domain ontologies, decreasing potential for interoperability. This is exacerbated by the methodological differences in constructing models (empirical or theory-based) and development phases from informal to formal ontologies.

Several of these theoretical factors and its effect on computing were tested in a pilot experiment with the modelling software for ecology, STELLA. Comparing the elements of the ecological model with computing terminology, formalising the identified correspondences between the elements in the ecological model and computing terminology is within reach, facilitating further possibilities for guided bottom-up development of ontologies. The methodology of using extended semantic representations to organise ecosystem equations in a placeholder objects model proved to be an approach useful for computing science and had a positive effect on ontology development.

A literature review of combining ontologies was carried out and analysed. Multiple terms, definitions and practices that refer to ontology 'integration' were structured, loosely categorised on a scale of increasing levels of integration and a list of factors and properties that contribute to distinguishing these multiple methods of integrating ontologies was created. It revealed that although ontologists demand from the subject matter experts to reach consensus, there is no agreement among themselves concerning the multiple interpretations as to what constitutes 'ontology integration' and its related concepts.

Both positive and negative expectations on integrating ontologies of the same, similar and orthogonal subject domains were formulated. Semantic versus structural integration was highlighted with an example of the polder ecological niche, so were the potential positive effects and complications of facilitating multilingualism for ontology development and integration. It revealed that a strict separation between semantic and structural integration is not as obvious as the definitions might suggest. Another example involved ontology construction exploiting the approach of an ontology base and commitment layers with relation to microbiology, which may improve reuse of knowledge even more and may assist in clarifying the multiple understandings of the Defined Terms Ontology. Further, effects of varying modelling paradigm heterogeneity was highlighted and an analysis of the model / ontology of Defined Terms of plant taxonomy is provided, which might benefit from a higher level of formalism and clear definitions and justifications for the taken methodology.

Ontology integration software was briefly addressed. Each application provides a partially automated solution to a specific aspect of ontology integration within their chosen implementation language. Compared to the automation of the heuristics of integrating ontologies on the semantic level, automation on the system and syntactic level is relatively straightforward and achieved; semi-automation of semantic integration is still a hot research topic.

The report is concluded with a discussion on outstanding issues and how they may be investigated.

Table of Contents

Table of Figures and Tables

List of Abbreviations

Abbreviation	**Description**
AOS	Agricultural Ontology Service
CG	Conceptual Graph
DL	Description Logics
DOGMA	Developing Ontology-Guided Mediation for Agents
DTO	Descriptive Term Ontology
ER	Entity-Relationship
IDE	Integrated Development Environment
KBS	Knowledge-Based Systems
KIF	Knowledge Interchange Format
KR	Knowledge Representation
LEEDS	Lake Eutrophication, Effect, Dose, Sensitivity model
ODE	Ontology Development Environment
OIS	Ontology-driven Information Systems
OO	Object-Oriented
ORM	Object Role Modelling
ORM-ML	Object Role Modelling Markup Language
SEEK	Science Environment for Ecological Knowledge
SME	Subject Matter Expert
UML	Unified Modelling Language
XML	eXtensible Markup Language

1. Introduction

The fields of bio- and ecoinformatics are relatively new and experience multiple hurdles in their interdisciplinary approach of combining different specialisations within biology and computing. Biologists from different sub-disciplines require access to the same data, but for different purposes, hence can attach distinct structure and meaning to the 'same' data and underlying concepts and relations, or vice versa. Therefore capturing the semantics is a major problem and combining these models into larger (software) systems is even more challenging. One approach to achieve consensus between divergent views in the subject domain of biology are the efforts to create ontologies to facilitate communication about the concepts and their relationships that in turn can be reused for various purposes. Over the past decade many resources have gone into researching and developing ontologies, including several ontologies with subject domains such as plant taxonomy, genes, cell function and agriculture. However, with the recent proliferation of ontologies, the aspect of *integrating* ontologies will become important if one wants to avoid re-inventing the wheel by re-analysing a subject domain, and save development time of both creation of ontologies as well as (re)use of ontologies to aid the analysis phase of the software development process.

In order to determine the most appropriate way to integrate multiple ontologies, one first needs to know what is to be integrated, i.e. considering the possible types of ontologies and heterogeneity of the data and subject domain, which will be addressed in chapter 2. Chapter 3 contains a pilot experiment illustrating several of the issues analysed in the preceding chapter and looks forward to development of ontology/ies in the subject domain of ecology. Subsequently, the attention is directed towards ontology 'integration' in chapter 4, where the concept of 'integration' is used to denote a plethora of actions with lowest common denominator "something happens and it involves more than one ontology", and a (partial) categorisation of integration concepts and methodologies is proposed. Chapter 5 summarizes some of the efforts on the available integration software, implementation difficulties of combining ontologies encountered by other researchers, and addresses software features, and some integration software requirements. The last chapter provides conclusions and outlines several outstanding issues and research suggestions regarding ontologies.

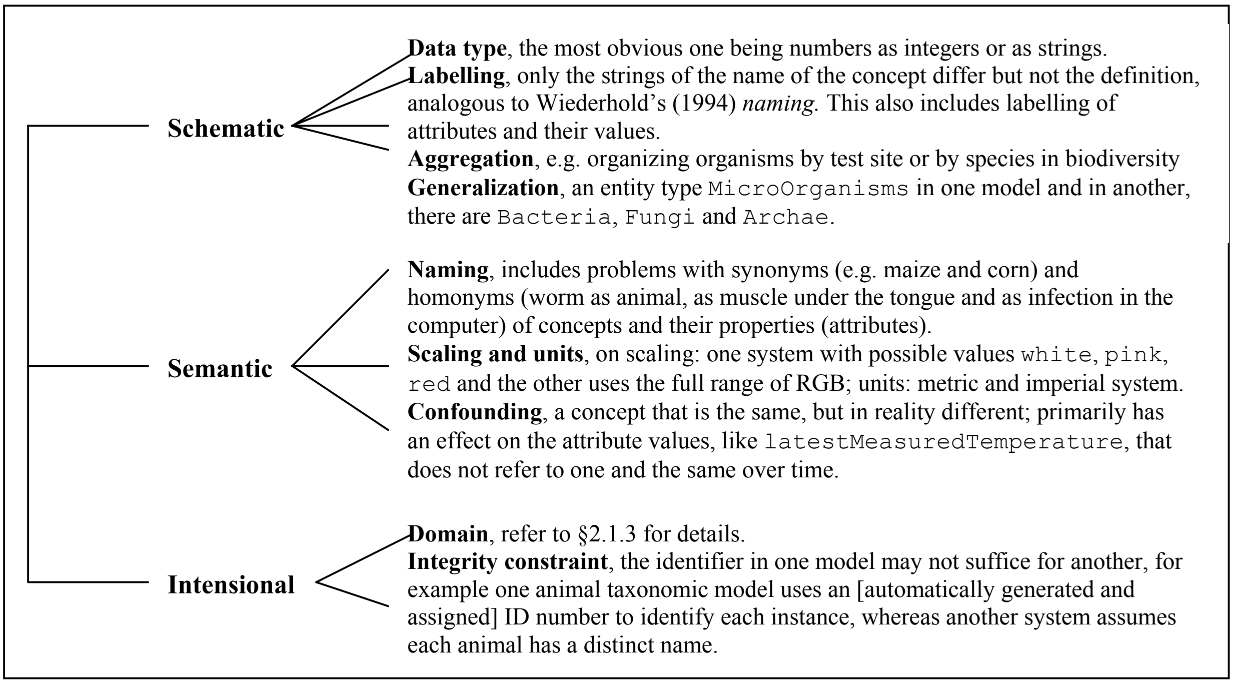
2. Domain heterogeneity and types of ontologies

2.1 Heterogeneities

Data heterogeneity and specific data characteristics concerning the subject domain of biology are examined, and subsequently expanded with other types of heterogeneities.

2.1.1 Data heterogeneity

Irrespective of the subject domain one desires to model for one purpose or another, it is possible to identify a number of factors contributing to data heterogeneity. According to Goh (1996), there are three types, each with further subdivisions, which are schematic, semantic and intensional heterogeneities (that can result in data conflicts). Although the distinction between semantics and structure is not always agreed upon (see also *Example 5* in §4.2.1), generally the idea is that structural refers to how concepts are organised, thereby assuming there is agreement on the meaning (semantics) of these concepts. *Figure 2.1* contains an overview of the different types of data heterogeneity, each with an example to illustrate the aspect.



Schematic

Data type, the most obvious one being numbers as integers or as strings.
Labelling, only the strings of the name of the concept differ but not the definition, analogous to Wiederhold's (1994) *naming*. This also includes labelling of attributes and their values.
Aggregation, e.g. organizing organisms by test site or by species in biodiversity
Generalization, an entity type `MicroOrganisms` in one model and in another, there are `Bacteria`, `Fungi` and `Archae`.

Semantic

Naming, includes problems with synonyms (e.g. maize and corn) and homonyms (worm as animal, as muscle under the tongue and as infection in the computer) of concepts and their properties (attributes).
Scaling and units, on scaling: one system with possible values `white`, `pink`, `red` and the other uses the full range of RGB; units: metric and imperial system.
Confounding, a concept that is the same, but in reality different; primarily has an effect on the attribute values, like `latestMeasuredTemperature`, that does not refer to one and the same over time.

Intensional

Domain, refer to §2.1.3 for details.
Integrity constraint, the identifier in one model may not suffice for another, for example one animal taxonomic model uses an [automatically generated and assigned] ID number to identify each instance, whereas another system assumes each animal has a distinct name.

Figure 2.1. Factors of data heterogeneity. Categorisation based on Goh (1996:17-22), explanation & examples by author.

2.1.2 Characteristics of biological data

A brief treatise on the characteristics of biological data is in place considering that the aspects discussed in this report – heterogeneity/mismatches, ontology types and, in chapter 4, ontology integration – will be applied to biology, and ecology in particular. In an abstract sense, one can think of 'data is data' and the principles are the same regardless the subject domain. It has been argued, though rarely, that analysing and modelling biological data does not differ from human-generated systems such as business practices, but that data in e.g. financial domains stick to 'raw' data, where analysis and interpretation of the results is conducted by application software and human intervention adding more semantics to the data, whereas the requirements of biologists are not only to capture data but also must include (a subset of) semantic mediation. Conversely, one can consider this 'semantic mediation' not separately, but as an intricate part, a characteristic, of biological data. This author is convinced there are distinct features of biological data influencing conceptual modelling, ontology development and ontology integration, because it involves complex data and categorisation & reductionism in biology is a guideline, not a certainty – or: the whole is can be more than its parts. These are of lesser importance or absent from the standard examples predominantly found in the research literature, which are often human-created and 'common sense' subject domains where the modeller is also subject matter expert, such as (integration of) ontologies of universities, modelling a travel ticket system or some of the Semantic Web[1] examples. First, general characteristics of biological data are addressed, followed by some additions for ecological data in particular.

General biology

What makes biological data different from the more 'standard' type of data that it merits special attention? Aside from aspects specific to the (sub-)domain, there are five general characteristics distinguishing it from the more common and human-created subject domains like businesses and university structures.

First, neither is there a legacy to rely upon, nor can one expect a modeller to have full knowledge of the Universe of Discourse (UoD). Although there are hundreds of biological databases[2] and application software packages, the scope of the topics covered varies widely, hence the chance that data analysts/ontologists pondered about the same questions and have concluded to represent it one way or another throughout is relatively small, whereas this is the case with, say, [models of] financial systems capturing business processes. Although over time this difficulty may be alleviated when more data is modelled, the hurdle of computing scientists having to learn extensively about a knowledge-based subject domain and the subject matter experts (SMEs) 'forced' to become cognizant of the structure and categorisations of computing will not come naturally, even if the analyst/researcher is trained in both disciplines. In addition, this does not merely involve taking an 'introduction course' in biology, but philosophically one can distinguish the processes and knowledge-based approach of scientific enquiry characteristic for the life science as distinct from the engineering practices of informatics.

Second, take for example production of a metabolite (a molecule produced by an organism) or strength of inhibition by an antibiotic to kill the bacteria causing an infection that can have 'higher' production and 'stronger' effects in some environments and less/weaker under other circumstances, or change morphology due to changes in the environment[3]. This poses two questions, which would need to be analysed and modelled somehow: first, how much weaker or stronger, i.e. how to represent gradations, non-discrete data, in relationships? There is no such equivalent in, say, hockey club membership: either you are a member, or you are not. The second question relates to the 'some environments'. What environment, what are the determining factors and, more importantly, what is their effect on occasional relationships? It would require a model capturing "if parameter x is above threshold a, parameter y 'somewhat warm' and a 'low level' of z" and so forth, then there is a relationship – only to note that the exact parameters and their possible values involved to determine the existence of a relationship are often not fully known or understood even by the domain experts themselves. How then, can a computer

[1] http://www.w3.org/2001/sw/.

[2] Infobiogen maintains a database of biological databases, online at http://www.infobiogen.fr/services/dbcat.

[3] For example the effect of starvation on the *Vibrio* S14, which changes shape from a rod-like shape with one large flagellate to 'ultramicrocells' that are much smaller, spherical and with several mini-flagellates; the changes depend on the duration of starvation too (Kjelleberg *et al.*, 1990).

scientist represent the semantics correctly and comprehensively? How ought one to represent environmental conditionality, heterogeneous information and fluctuating data quality? Alternatively, for example an address from a company: one knows the components (attributes), *all* of them *and* modelled numerous times before. On the contrary with biological data: in addition to aforementioned uncertainties, functionality can be 'confirmed' as well as 'postulated', i.e. there can be a requirement to document a plethora of conjectures; how can one anticipate attributes and entity types if researchers do not precisely know the parameters? These 'informed guesstimates' may not only be valid in hindsight, but may be of such importance, that what at present suffice as an attribute has to be 'upgraded' to become an entity type or object with its own related parameters – an example of this may be the realisation that 'junk' DNA, long dismissed as an evolutionary leftover, has some function after all. However, this aspect is only of higher relevance if one were to use OO or ER for modelling, whereas the modelling paradigm as for example ORM is attribute-type free.

The third difference is the lack of versus the abundance of data in a certain subject area. In itself, this is not necessarily a problem of modelling concepts and their relations. However, if it is an exception or rarity of complex data, it might not be worthwhile to spend an excessive amount of time to create an elaborate model or section in an ontology to cater for this anomaly. Although one may argue that the necessity to be entirely '100%' correct is not per definition a characteristic of biological data but a design decision, this problem occurs considerably more often when modelling biological data. Compare for example the available data on *Zerna inermis* and *Zingiber mioga* [4].

Fourth, there are definitional problems and a general lack of standardization in nomenclature in biological data (Wittig and De Beuckelaer, 2001; Frishman *et al.*, 1998; Macauley *et al.*, 1998; Laser and Roest Crollius, 1998, among many others): "anarchy" according to Drysdale (2001), although the FlyBase[5] she describes adds to this problem because she devised her own keyword system. The MBGD[6] elevates this to a feature: the user can create his/her own classification table (Uchiyama, 2003). A similar problem exists in the domain of ecology: there is an overabundance of (semi-standard) models, but a *common* declarative standard is absent (Villa, 2001). There are few coordinated attempts to unify data formats via Abstract Syntax Notation I (Frishman *et al.*, 1998; Bader *et al.*, 2001), the NEXUS file format (Maddison *et al.*, 1997), the Ecological Metadata Language[7], and the establishment of the Gene Ontology Consortium[8]. When viewed from the perspective of conceptual modelling for databases and software development, the latter approach with ontologies might be criticised for 'dumping' semantic and conceptual disagreements of research groups into the lap of ontologists; using more abstract methods does not imply consensus and interoperability is easier to achieve and, more importantly, ontology efforts use different approaches.

The fifth, and last general aspect, is related to the previous one: the definitional problems and lack of standardisation is not just due to the complexity of biological data, but there are disagreements between (sub-)disciplines and even within disciplines amongst research groups[9] as well as *within* research groups.

[4] Both at the Centre for New Crop and Plant Products, Purdue university, *Zanthoxylum americanum* at http://www.hort.purdue.edu/newcrop/herbhunters/pricklyash.html or *Zerna inermis*: http://www.hort.purdue.edu/newcrop/nexus/Bromus_inermis_nex.html and *Zingiber mioga*: http://www.hort.purdue.edu/newcrop/proceedings1993/V2-051.html#Myoga

[5] Database for the *Drosophila* genome: http://flybase.bio.indiana.edu/

[6] MicroBial Genome Database: http://mbgd.genome.ad.jp/

[7] By the Knowledge Network for Biocomplexity: http://knb.ecoinformatics.org/software/eml/.

[8] More information on the Gene Ontology Consortium is online available via: http://www.geneontology.org/, GOC (2001) and for an example of its use with pathway databases, see Krishnamurthy *et al.* (2003). There are longer established nomenclature attempts in naming enzymes and coordinated bacterial nomenclature (the latter subject to re-classifications resulting from molecular biology, analogous to the "New Drude" in plant taxonomy (Graham *et al.*, 2002)).

[9] An interesting case study was carried out by Miall and Miall (2001) with relation to stratigraphy, and considers conflicting paradigms, paradigm shift and influences in minority/majority views within a discipline.

Ecology

In addition to these 5 aspects, ecology often comprises interdisciplinary (so-called Mode 2) science, hence having to resolve ontological differences between these (sub-)disciplines, *and*, more importantly, taking into account "management and broader social views of the natural environment" (Argent, 2003 *in press*) and between 'basic' science, e.g. climate, and applied science like agriculture (Mineter *et al.*, 2003). Villa (2001) adds that one *has to* use different modeling paradigms to be able to capture such complex ecological system: e.g. a model that tries to capture the understanding of land use under different management scenarios, involving a process-based model for the landscape dynamics which has to interact with individual-based "multiple resolution models representing the stakeholder community".

Further, there is an 'embeddedness' of mathematical formulas within ecological concepts and their use which draws multiple concepts together. For example the canopy photosynthesis draws together several properties of a photosynthesis process, leaf with a CO_2 diffusion gradient, canopy, site with surface area and daylength and additional relations between these "placeholder objects" (Keller and Dungan, 1999), the Monod kinetics of (microbial) growth under nutrient limitation or the equations and objects forming the LEEDS model[10].

The second characteristic mentioned in the 'general biology' section, is exacerbated with ecological data: uncertainties exist in many intertwined system parameters, but still need to be included and it has to be able to cope with unavailable information by using *estimates* in attributes and their values as well as the relations among concepts (Huang and Chang, 2003; Ceccaroni *et al.*, 2000; Ceccaroni *et al.*, 2004 *in press*), or even base a whole model on one or more assumptions (Brilhante and Robertson, 2001). Such rough estimates in e.g. ecological efficiency, growth/mortality rates and C/N-ratio are then used for further calculations of nitrogen mineralisation in a food web such as the Lovinkhoeve farm lands (De Ruiter *et al.* (1994a) as discussed in Akkermans *et al.* (1996b)), its results in turn fed into larger ecosystem models. (Note that in this document no distinction is being made between ecological [semi-freely dynamically interacting] and biogeochemical [element-conserving] models, or any other that has a specific meaning in the life sciences [e.g. physical-biochemical], and all are referred to as 'ecological model' – i.e. 'a model with as subject some knowledge in the discipline of ecology' – at this stage). If one were to consider to include this in an ontology, which contains in the strictest sense *what is* and regularly also *what can be*, modelling such estimated, assumed or hypothesized ecological data would introduce the oddity to capture *what might be*.

Applied bioscience

Capturing the subject domain semantics of an applied bioscience faces slightly different problems compared to conceptual modelling for the 'core' life sciences. Whereas the former requires an emphasis on practical problems and solutions conceptually representing the integration of various disciplines, the latter stresses conceptual and ontological 'all-inclusive' comprehensive models within their primary specialisations, such as biochemistry and genetics, which can be analysed separately from other (sub-)disciplines when developing a conceptual model or ontology. Following are two examples for illustration.

The breakdown of penicillin: research from the perspective of organic chemistry focuses on factors such as pH, temperature, molecular structure of the substrate and enzymes involved (penicillinase [β-lactamase]), as with any other molecule. On the other hand, the interdisciplinary approach in applied bioscience will investigate the environment where penicillin is to be used and include a larger range of compounds that may interfere with the effectivity of penicillin as antibiotic, how the body disposes it and its effect when released in the environment. With this broader scope, it has found that cycloamyloses, produced by e.g. the bacterium *Bacillus macerans* via transglycosylation of monosaccharides (to produce cyclodextrin), can speed up the breakdown (cleavage of the amides) of penicillin 89-fold compared to the

[10] LEEDS = Lake Eutrophication, Effect, Dose, Sensitivity model, see e.g. Malmaeus and Håkanson (2004) and §3.3 for an example on the extended semantics of equations via the methodology of a placeholder objects model.

uncatalysed reaction rate due to the covalent catalysis caused by the circular/cone-shaped cyclodextrin that functions as an enclosure complex[11]. A researcher in an applied science or technology would want to have this type of information included in a model, not just the characteristics of the penicillin molecule.

On a much larger scale: there are e.g. climatological factors such as rainfall and temperature affecting primary produce, resulting in a change in characteristics and/or composition of fruit, vegetables and so forth, which in turn influences harvest and storage physiology and subsequent product quality (like taste, colour, nutrients) of either the original or processed product, with a knock-on effect on its marketability and consumer behaviour. There are ample examples of both positive and negative outcomes of such chains of events, respectively e.g. wine (Mason, 2003) and tomatoes (Keet and Van Lune, 1997). Such a chain of investigations involves at least the disciplines of earth sciences, (plant) biology, food sciences, engineering, human nutrition and the social sciences. Each vertical production column is unique, yet there may be similarities when one cuts horizontally through it to combine a section across production columns. This, whereas in the cores life sciences some degree of reductionism serves the scientists, with applied bioscience both detail and holistic views are essential.

Last, note that for each specific UoD there are additional (practical) data type specific problems to be resolved on top of these general aspects of biological [including ecological] data; for example classification systems in plant taxonomy (Raguenaud *et al.*, 2002; Priss, 2003; among others) or the loosely defined groups of microorganisms (Keet, 2003b).
Example 2 further below illustrates some of the aspects examined in this paragraph in conjunction with modelling paradigm heterogeneity considered in the next section.

2.1.3 Other heterogeneities

Apart from heterogeneity in the data, a distinction in types of heterogeneity is made between the semantic level and the representational level (Visser *et al.*, 1997), where 'representational' is imprecise; hence clearer distinctions are, apart form the semantic level, structure, syntax and system. Semantics and structural heterogeneity will be addressed later in this report. System heterogeneity comprises platform heterogeneity, including the operating system, file system and the hardware and information system heterogeneity, such as DBMS software (Sheth, 1999). Syntax heterogeneity actually covers more than merely differences in syntax, clarified by e.g. Sowa (2000) and summarised here. Apart from standard classical first order logic (FOL), there are variations of FOL, hence other ways of formalizing knowledge to capture the concepts and their relations in an ontology – and a source for conflicts and mismatches – organised according to six different characteristics:

- *Syntax*: for example Peirce's ∑ versus Peano's ∃ as existential quantifier.
- *Subsets*: the constraints on the allowed operators. For example, Prolog does not allow disjunctions in the inclusion of an implication and propositional logic includes Boolean operators (but not quantifiers).
- *Proof theory*: restricting or extending the permissible proofs. Linear logic (use a certain proposition only once), nonmonotonic logics (introducing default assumptions) and so forth.
- *Model theory*: modifies the denotation or truth value of a statement, as is with the multi-valued certainty factors in fuzzy logic.
- *Ontology*: uninterpreted logic; some versions of logic supplement FOL with an ontology of built-in predicates and axioms and mathematicians generally use the ontology of set theory as a basis for defining the foundations of mathematics.
- *Metalanguage*: the language for defining, modifying or extending a/any version of logic.

[11] Some information for this example was taken from Engbersen (1994) and Schlegel (1995).

To make this even more interesting, these six distinct characteristics can be combined in any blend, providing many opportunities to 'translate' between the syntaxes. For example some version of fuzzy Prolog defined as a "restriction of FOL to the Horn-clause subset with a modified proof theory and model theory and with metalanguage for expressing certainty factors." (Sowa, 2000:20).

Visser *et al.* (1997) add three[12] more types of heterogeneity:

- *Paradigm* heterogeneity, which the authors consider as different *modelling* paradigms, such as ER and OO; *Example 1* illustrates and discusses some differences and consequences of modelling choice.

  However, based on the name, this could equally well mean paradigms within the *subject domain* of the ontology, as is a major problem in, for example, developing an ontology for plant taxonomy (PrometheusDB[13]). It is also important to note the difference between *ecological* modelling and the modelling paradigms within the discipline of informatics. With ecological modelling, there is a plethora of methodologies and graphical representations that does not bear any apparent relation to informatics models whatsoever, e.g. Odum's conventions (see Heemskerk *et al.* (2003) for an example), and are more focussed on ecology and simulations than on modelling for its own sake: there are over 529 terrestrial ecological models documented in the Register of Ecological Models, each one with a unique approach and structure (Liu *et al.*, 2002), and over 160 agricultural models in the UK alone[14]. Further, the content of an ecological model is often *associative* knowledge-oriented as opposed to structural; see also *Example 2*. From a software perspective, the ecological software has been gradually moving from procedural [currently mostly legacy] systems to OO software[15] and ER for RDBMS databases, and subsequently the very recent attempts to create the Ecological Metadata Language and ontologies (e.g. AOS[16], SEEK[17]) to annotate and model ecological and agricultural knowledge. To date, the computing heterogeneity is better organised and structured than the 'free-form' modelling in ecology.
- *Ontology* heterogeneity refers to "different ontological assumptions", like the components of a farm as farm (fence, house, livestock) and the farm as a niche market for tourism (cuddly animals, kitchen appliances).
- *Content* heterogeneity, when two systems represent different knowledge. For example, one can model a flower being composed of a petal, leaves and so forth, but also from a utilitarian perspective (sellable, the related logistics system).

With Visser's second and third heterogeneity types, one enters the areas of domain and semantic heterogeneity, encompassing multiple factors. Whereas logic/language and 'paradigm' heterogeneity can be restricted to development of a single ontology, although a potential source of conflict when combining multiple ontologies, the effects of domain and semantic heterogeneity mostly have an effect on combining ontologies and only 'in hindsight' on representation/creation of a single ontology, because it is not always known what is going to happen with the ontology once it is created, hence it cannot be anticipated what representation is the best one.

Example 1. Paradigm heterogeneity within computing

A small experiment is discussed to illustrate effects of paradigm heterogeneity. The concepts and relationships under test represent an overview of the organisation of 'descriptive terms' used to describe characteristics of plants for constructing (a section of) a plant taxonomy, as modelled by the

[12] Visser added "language heterogeneity", as indicated already by Sowa in this section.

[13] http://www.prometheusDB.org

[14] Silsoe Research Institute (SRI), http://www.sri.bbsrc.ac.uk/science/bmag/itagr.htm

[15] Consult e.g. Mineter *et al.* (2003), Baskent *et al.* (2001) and the Analest and Reciclado de Nutrientes of the ICA (http://www.ica.inf.cu).

[16] http://www.fao.org/agris/aos/

[17] http://seek.ecoinformatics.org

PrometheusDB project (Paterson *et al.*, 2004 *in press*). They modelled the concepts with a version of OO (*Figure 2.2*) taking the diagram at face value, albeit not containing 'complete' information as one would find in a UML class diagram, which will be discussed further below.

Analysing this model and related literature, this author formulated 12 questions (included in *Appendix A-1*) with as main aim to verify interpretations and receive clarification on ambiguous factors. These questions were answered by the relevant informatician before the model was translated into ORM. Recreating the first version of the model, this author took the liberty to make several assumptions and left certain aspects, like the rules (commitments) between `State Group` and `Structure` empty. Subsequently, these assumptions and further questions raised during this 'translation' activity, as included in *Appendix A-2* (points 13-27), were confirmed/answered; the result based on the provided information is shown in *Figure 2.3*[18].

With the additional information gathered, an ER version of the domain was created as well, because the intention is to implement the descriptive term model in a relational database – although VisioModeler allows one to automatically generate a relational database. The ER model, in *Figure 2.4a and 2.4b*, also does contain more information, rules and constraints than the OO version but less than the ORM version.

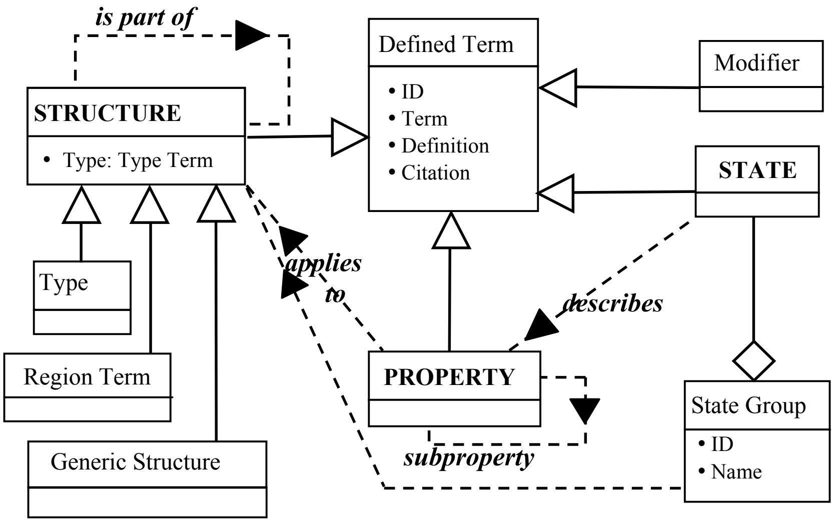


Fig 4. Concepts and relationships in the descriptive term ontology. All terms are types of Defined Term. Structures can be 'part-of' other structures recursively, and may have attribute: Type. States are composed into groups, which may be restricted to ('applies-to') certain structures. Therefore these state groups may represent 'de facto' properties, which may include a structural context. States describe a given property, which may be applicable to only certain structures.

Figure 2.2.The main concepts and relationships of the Descriptive Term Ontology.
(Reproduced from Paterson *et al.* (2004 *in press*))

[18] Note that this ORM exercise is different from *Example 3* in §2.2.1 [with relation to the DOGMA approach]. In both examples ORM figures are included, but here no distinction is made (yet) between ontology base and commitment layer, merely to represent the knowledge on a more abstract level with a more expressive modelling paradigm. See *Example 4* for a discussion of this Defined Terms Ontology in the context of ontologies.

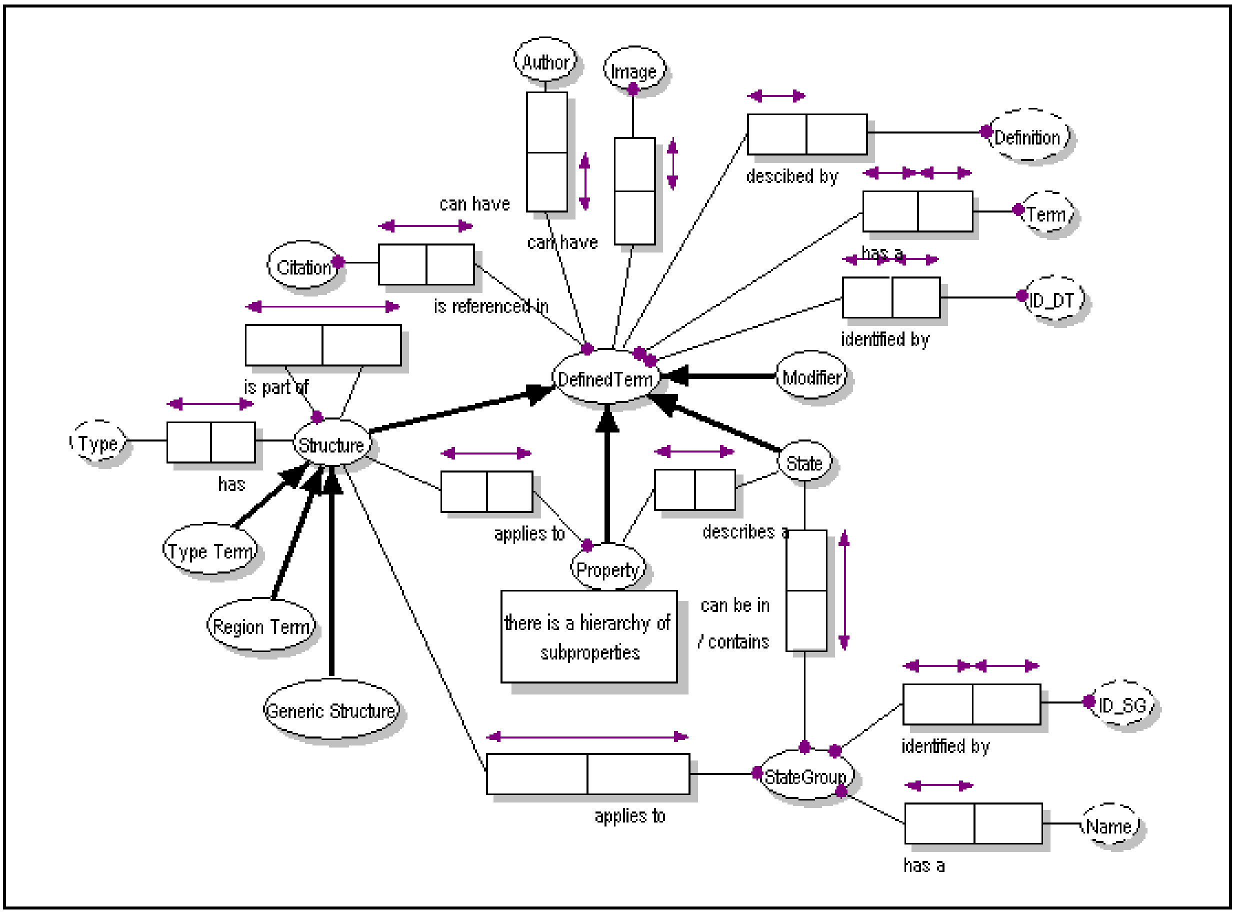
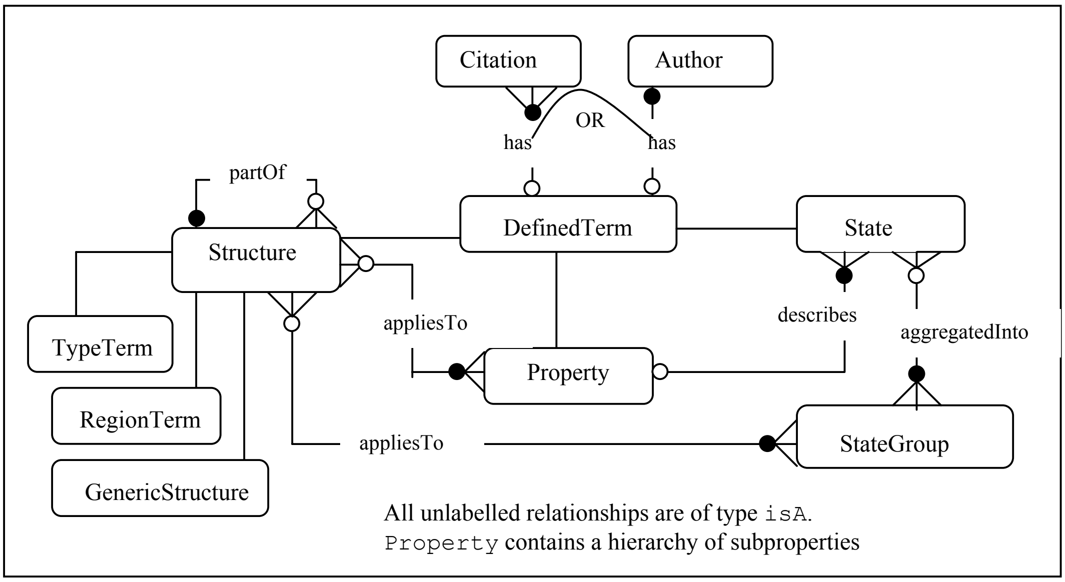


Figure 2.3. 'Translation' plus additional knowledge not captured in the OO model. The Objects Type Term*,* Region Term *and* Generic Structure *all have the same attributes as the* Defined Term *object, though omitted from the diagram to avoid a too cluttered view.*



All unlabelled relationships are of type isA.
Property contains a hierarchy of subproperties

Figure 2.4a. ER diagram, including the later provided information.

```
DefinedTerm(ID, Term, Definition, Citation, Author, Image)
Modifier(ID, Term, Definition, Citation, Author, Image)
TypeTerm(ID, Term, Definition, Citation, Author, Image)
RegionTerm(ID, Term, Definition, Citation, Author, Image)
GenericStructure(ID, Term, Definition, Citation, Author, Image)
Property(ID, Term, Definition, Citation, Author, Image)
Structure(ID, Term, Definition, Citation, Author, Image, Type)
State(ID, Term, Definition, Citation, Author, Image)
StateGroup(ID, Name)
```

Figure 2.4b. Entity types of the ER model.

The most obvious aspect is the amount of detail that is captured in the three types of diagrams: it was *after* seeking answers on targeted questions that more detail about the relationships, and the constraints in particular, surfaced. Though this might have been present 'implicitly' in the OO model of *Figure 2.2* and/or possibly might be added at a later stage, if the present Prometheus project member(s) would not have been contactable, this knowledge would have been absent or even lost and would have required yet another assessment of the subject domain. Further discussion with the same and another informatician involved in the project revealed that the *emphasis* of this particular model should be on the fact that this diagram is a "summary" and a "work in progress" where there is "no consensus on all aspects" yet, and the graphical representation is "with no particular intention" whatsoever an OO/UML diagram, merely used for purposes that "these symbols are familiar and known in the modelling domain and discipline". However, exactly because they are known and used under specific circumstances, using them for other purposes easily can lead to misinterpretations of the subject matter modelled and generate higher expectations than are in fact realised. Secondly, it is only implicit in the diagram that it is a summary: one needs to know more of the context to be sure, of which some is mentioned in the accompanying text in the article.

The analysis as part of the ORM exercise clarified some of the vagaries of the relationships, the constraints/rules in particular between concepts `State-Property-Structure` and `State-StateGroup-Structure`: as modelled, they are not different representations of the same semantics, but actually *are different*, most notably:

a) There can be properties, hence structures, that are not defined by one or more states, whereas via the second 'route' with `StateGroup` this is not accommodated for in the present model.

b) There is a strict hierarchy of (sub)`Properties`, but this is neither included in the model as such nor indicated that there might/can be a hierarchy of (sub)`StateGroups` analogous to, or the same as, the sub-properties hierarchy.

c) `State`s are aggregated 'on demand' into `StateGroups`, which is indeed represented as such with the diamond shape. However, the article also states, "The composition of such a set of states (a 'State Group') could be considered to circumscribe an implicit, *de facto* 'property'", but this is not shown as such [with a diamond shape in the `describes` relation] in the model. This may, or may not be an unintentional omission.

d) Moreover, regarding this `describes` relation, the article mentions that "a state can include aspects of *several* properties" (emphasis added) in the view of taxonomists, which did not find its way in the model, but such statement was not made about the state groups, i.e. there was no confirmation that several state groups can make up a state.

The intention, according to the computing scientists, is to "solve" this ambiguity in due course by "implementing both and see what works best" – however, this is certainly different from pretending they are the same: they are different views and what actually is meant is that each of the parties involved in the construction of this model is convinced that at a later stage one's own interpretation is proven correct. Seemingly in contradiction was the mentioning of an 'imagined' relationship `isCalled` between `Property` and `StateGroup`, omitted from the diagram "for political reasons": if they really are the same, why bother with some labels attached to the concepts? Because then another starting point, apart from communicative improvements, could have been to implement the taxonomist's interpretations and use an alias table to store the different labels for the same concepts. However, from a modelling perspective*, it actually can be interpreted as advantageous to use a less expressive modelling technique to avoid confrontation or to hide irreconcilable differences; likewise, spelling out the knowledge may actually allow one to specify the gritty details and help solving the different views* as opposed to postponing to resolve it. Thus, utilizing a particular modelling paradigm does not only depend on the expressiveness of a certain modelling technique and other scientific argumentations, but also – or maybe even more so – on the sociological factors of cooperation between individuals and teams. Less expressive models can provide a tool of power for the benefit of the modeller and detriment of the domain expert, in the sense of always being able to tell more than what is actually represented in a diagram, thereby the modeller is one or more steps ahead in processing this information. When such extra detail is un- or loosely documented, these facts/knowledge have, and remain to have, the characteristic to become a source of disagreement regarding the 'who said what when where and how' sphere of communications. It is outside the scope of this document to digress further into the realms of social informatics and the scientific cooperation across disciplines.

Other factors that were included in the ORM model, were additional attributes (value types) *unintentionally* omitted from the OO model (e.g. `Author` and `Image`), and their relationship to the entity types were defined, although these aspects are relatively of lower importance. Further, the `subproperty` recursive relationship of `Property` in the OO model is not represented correctly: what actually is meant is that there is a hierarchy of properties 'underneath' `Property`, omitted from the OO diagram and noted in the ORM diagram. A recursive relationship is when e.g. an entity type `Nurse` has several instances of nurses, say {`a, b, c, d`}, and one of these nurses, `a`, is also team leader, but still they are all nurses. In *Figure 2.2*, this is where one instance of a `Structure` is a structure of it self, but can also form a part of a structure. On the other hand, underneath the type `Property` is a hierarchy of sub properties, where a sub property γ can only be part of another property β or α if β and α are in the same branch higher up in the hierarchy, but γ cannot be a sub property of ю that is categorised in another branch of the `Property` hierarchy (and obviously, γ never can be a sub property of δ that is further down in the same branch of the `Property` hierarchy). These limitations do not make it equal to a standard recursive relationship.

Analysts have their own preferences for data modelling, and one may be more suitable than another considering the particular function of the model and the subject matter to be modelled, but from a data analysis perspective, richer modelling techniques, such as ORM or Conceptual Graphs[19], are more beneficial for representing the knowledge as accurately as possible, in turn improving understandability, hence user requirements and software quality.

□

Another aspect related to plant taxonomy is the different representations of plant taxonomic data (that from an outsider's view it would be exceedingly suitable for hierarchical modelling), and the limitations of conceptual modelling facilities built into the modelling techniques. While *Example 1* suggests a more expressive – and more formal – modelling paradigm may be beneficial for that particular case, this is not true per definition and it is not established that all biological data can be captured in formalisms.

Within plant taxonomy, there is not one, but three principle hierarchies (classification, name and rank), each with varying definitions of their actual instances as used by taxonomists. Within the ranking hierarchy, data has the ability to acquire roles or change behaviour according to context; further, the

[19] An excellent textbook on ORM is written my Halpin (2001); Publications by e.g. Mineau (Mineau *et al.*, 2000) on conceptual graphs.

intended conceptual model should support recursive behaviour and composite entity types (Raguenaud *et al.*, 2002). ER does not allow for such complex data types, except when one would implement this in the application layer, which is not the intention when devising a conceptual model. Raguenaud (2002) created his own version of conceptual modelling, based on the Extended OO model, called POOM (Prometheus Object Oriented Model), to allow for taxonomic complexities. On the other hand, Priss (2003) modelled overlapping hierarchies, especially the taxonomic ranking (variety, species, genus, and so forth), and devised a mathematical formalization utilizing Formal Concept Analysis[20]. In principle, FCA facilitates reuse of software instead of having to write *ad hoc* solutions, like POOM, and it emphasizes the use of logic to make the implicit explicit. Albeit providing a convincing model, the prime aspect is the assumption that one can capture biological semantics in formalizations. Can one formalize everything mathematically?[21] Without digressing in philosophical matters if at some point in the future understanding of biology has advanced to such an extent that humans may be able to capture all aspects of the life sciences in mathematical formulae, or if this would be impossible, at the time of writing, there is, from the viewpoint of a computer scientist, a considerable lack of structure, abundance of uncertainties and apparent inconsistencies of biological data and disagreements on biological concepts that would make conceptual modelling with FCA, or e.g. Description Logic, a difficult undertaking. Additionally, it may not be reasonable to expect researchers to be fully trained in some biological domain, be an expert in conceptual modelling/ontologies, competent in mathematics and adequately capable of teamwork and communication. The more realistic situation is an interdisciplinary approach where people from different disciplines need to find common ground and overcome differences in research methodologies and practices. This is demonstrated in *Example 2* with regards to interpretations and assumptions as to what constitutes 'modelling' in ecology and the hierarchical perspective of an ontology.

Example 2. Paradigm heterogeneity between ecological and computing models

This example illustrates the differences between an associative knowledge-oriented ecological model and how this 'translates' to representations in computing. The interrelations and dependencies between fungi, mainly *Leucoagaricus* or *Lepiota*, and leaf-cutting ants such as *Atta Sexdens*[22] who cultivate and eat the fungi, are shown in *Figure 2.5*. The fungi grow in the 'fungus gardens' and live mainly from leaves, other non-decomposed wooden debris and ant secretions; ants feed themselves with the fungi and regulate the microbial population via various secretions.

Remodelling into an ontology, the first aspect is to distinguish three distinct areas: the organisms involved, chemical compounds and activities, included in *Figure 2.6* (which is incomplete). Intriguingly, the categorisations in *Figure 2.6* do contain *more* structure and concepts, such as `Prokaryotes`, `Eukaryotes`, `Inhibition` and so forth, but at the same time *less* than the ecological model of *Figure 2.5* by not [yet] having accommodated for the fact that e.g. the mycelium of a fungus functions as food for ants (indicated in *Figure 2.6* with a dashed line) nor have the instances been addressed. One would need to expand the relationship types from `isA` and `partOf` to 'freestyle' to connect types both within a categorisation as well as between the categorisations. This is allowed in conceptual models such as ER and ORM, but not always the case with ontologies (e.g. Guarino and Welty, 2000), although one can think of those inter-relationship types as a second layer over the main `isA` and `partOf` relationship types.

[20] Refer to http://www.upriss.org.uk/fca/fca.html and Ganter and Wille (1999) for details on FCA.

[21] This author has no straight answer on this at the time of writing (1-2004): sometimes I think it should be possible, other times, primarily when indulging in biology/ecology [research] literature, I think it probably is not possible, or at least not with the current state of knowledge of nature.

[22] The species names and some introductory information on the subject matter can be read online at e.g. http://www.botany.hawaii.edu/faculty/wong/BOT135/Lect24.htm

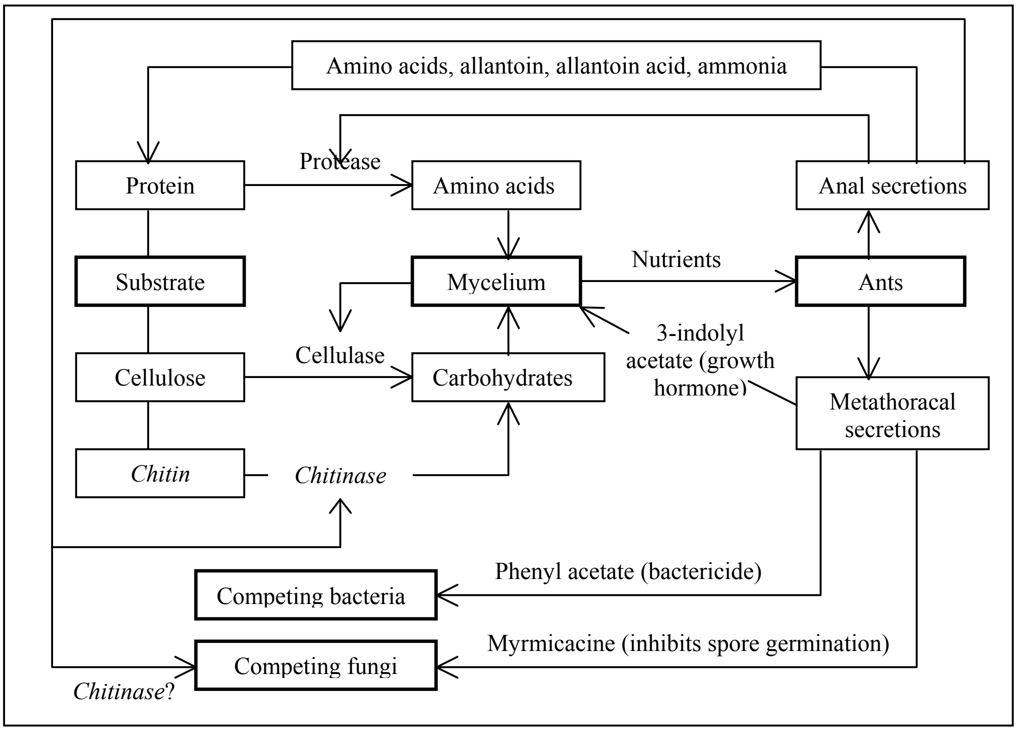


Figure 2.5 Ecological model of the chemical interactions of the symbiotic relationship between leaf-cutting ants and fungi (Source: translated from Akkermans *et al.*, 1996b).

Further, the very basic categorisation of `Chemical compounds` and `Activity` likely may benefit from a more rigorous categorisation, by e.g. using (part of) the Gene Ontology and the organisms could be taken from a top-ontology of organisms – both approaches essentially providing much more 'extra baggage' whilst still not modelling the relationships between the different types in a non-hierarchical manner. However, one can think of adding a second layer to the three areas consisting of the freestyle relationships such as the one indicated with the dashed line, read as "leaf-cutting ants consume mycelium". This implicitly also should be interpreted as "mycelium provides food for leaf-cutting ants", but the latter is semantically not exactly the same as the "mycelium providing nutrients for the ants" in *Figure 2.5*: an organism can eat things that do not provide nutrients for itself; nor is 'providing food' the right-to-left reading of 'consume': an organism can not only consume food but also liquids and air.

Although the ecological model makes sense and provides sufficient information for a biologist, it does not from the perspective of an informatician. For example, the representation of `Secretion` and `Substrate` may not be accurate: the ecological model does *seem to suggest* that (some or all?) compounds of the anal secretion is substrate as well, but one cannot be sure with the limited given information – and maybe this detail does not matter from the perspective of the subject matter expert. More generally, *none* of the relationships in the ecological model is expressive in its meaning, apart from indicating that a thing in one text box has *something* to do with what is written in another text box. For example, that `Mycelium` provides `Nutrients` to `Ants`, contains the implicit information that mycelium is the (main) part of a fungus, the ants eat this mycelium, which contains (essential) chemical compounds required by (the metabolism of) ants to stay alive. Further, the italics and question mark associated with `Chitin` suggests this was at the time of creating the model a hypothesis, which ideally should not occur in conceptual models or ontologies for the fact that it implies potential unreliability and would require possibly extensive facilities for (database) schema maintenance. On the other hand, e.g. De Ruiter *et al.*'s (1994b) ecosystem model of the soil food web in farmland (not included in this document), does *not* contain any names or functions of the relationships as in the fungus-ant symbiosis of *Figure 2.5*, but does indicate the *importance* of a relation: the thicker the shaft of the arrow, the higher the percentage. To accommodate this in a computing model, facilities of artificial intelligence are required.

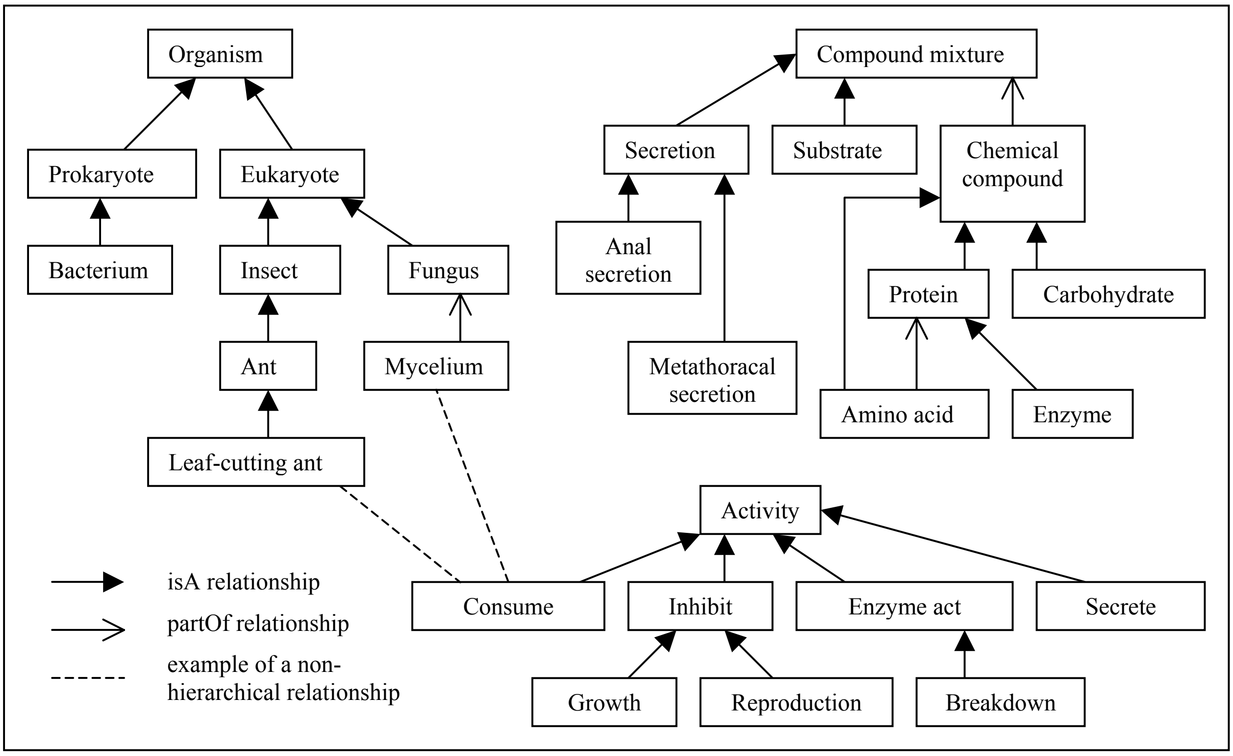


Figure 2.6. Organisms, chemical compounds and activities of Figure 2.5 *separated from each other.*

These factors, and other literature, suggest there is a somewhat chaotic situation of approaches for capturing knowledge, with the range of associated problems well known in computing science. Secondly, mycelium of what kind of fungus? At one place using a generalisation, e.g. `Ants` and `Mycelium`, with others more specific, such as `Cellulose`, is mildly inconsistent. Third, a homonym was initially present in a draft of *Figure 2.6*: `Secretion` as an activity and `Secretion` as the material that is secreted. Renaming the latter into e.g. `SecretionStuff` might have been an option, but sounds ugly and there probably are better names (which would need to be suggested by / confirmed with the domain expert), therefore the `Activity` subtypes were all renamed into a verb, which is a subjective design decision of the modeller (this author). Fourth, although to this author it is clear that the labels `protease`, `cellulase` and `chitinase` are enzymes that breakdown the macromolecules protein, cellulose and chitin respectively, this may not be obvious to people who do not have a background in biology.

Aside from analysing the actual differences between the models, Todorovski and Džeroski (2001) shed light on the process of creating models, and identified there are differences in methodology between ecological modelling and a theoretical approach (including computing). The former has a basis in an empirical, trial-and-error, tactics to tweak with the model until it fits observed data and only a limited set of parameters of the subject domain is used; this is valid for equation discovery in particular. In contrast the theoretical approach, where one starts with identifying the basic processes and objects involved, subsequently the details are filled in and fine-tuned together with domain experts, for only afterwards the values of the constant parameters are adjusted with the real data. (Todorovski and Džeroski (2001)). From this perspective, it is quite understandable that ecological models end up being different than theoretical models, because an effect of this divergent approach is that model (/ontology) development from the empirical perspective starts small and evolves by *growing and spreading out* once more research is conducted and the discipline better understood, whereas the second develops from a framework and will gradually be *filled up directed inwardly*. Different methodologies facilitate making divergent design decisions; therefore increase the likelihood of ending up with a different structure/semantics of an ontology.

The computing perspective complicates the already relatively complex symbiotic relation between fungi and ants and the ecological model contains a lot of implicit knowledge (and assumptions?) and under-specified relationships. Even though within ecology the practitioners are familiar with the idea of modelling their domain knowledge, this does not mean it will be easier for the computing scientist to elicitate and translate this knowledge into a model that could be used in a computing environment.

□

Continuing with characteristics of domain and semantic heterogeneity, naming, scope, encoding and attribute scope can be identified[23]:

- *Naming* heterogeneity means that concepts or their attributes are considered to be the same, but merely have another label attached to it, with, say, `EndoplasmaticReticulum` in one ontology and `EndoRet` in another. However, Wiederhold's description does not seem to include the problem of homonyms, which are more difficult to identify than synonyms. The concept that involves homonyms, where the same word has different meanings, is called polysemy, which, unlike synonym heterogeneity, always requires human intervention to identify and solve.
- *Scope* heterogeneity is somewhat more difficult to identify and requires content analysis: the intersection domains do not match precisely or the rules are not basic to the domain intersection. For example one conceptual model with a concept `MicroOrganism` and attributes or related concepts `LatinName` and `CausesDisease` and the other has `MicroOrganism` with related `LatinName` and `Designate`.
- *Encoding* differences in attribute values, a common one being conversions between the SI system and other units of measurements or one can think of the Munsell colour coding system versus (the arbitrary coding of) the Pantone system.
- Subjectivity of *attribute scopes*. For example, the term `old` in an anthropology domain has a considerably smaller time span than `old` in a climatologic domain. If one were to combine the two, for example to look for parallels in anthropological data and changes in the climate eras, knowledge of the domain experts is required to resolve this.

Note here that naming and encoding regarding domain heterogeneity is like Goh's semantic data conflicts (*Figure 2.1*), but then in a larger framework. On the other hand, one can argue that the more comprehensive categorisation proposed by Goh encompass the naming and coding (although 'missing' scope/context heterogeneity).

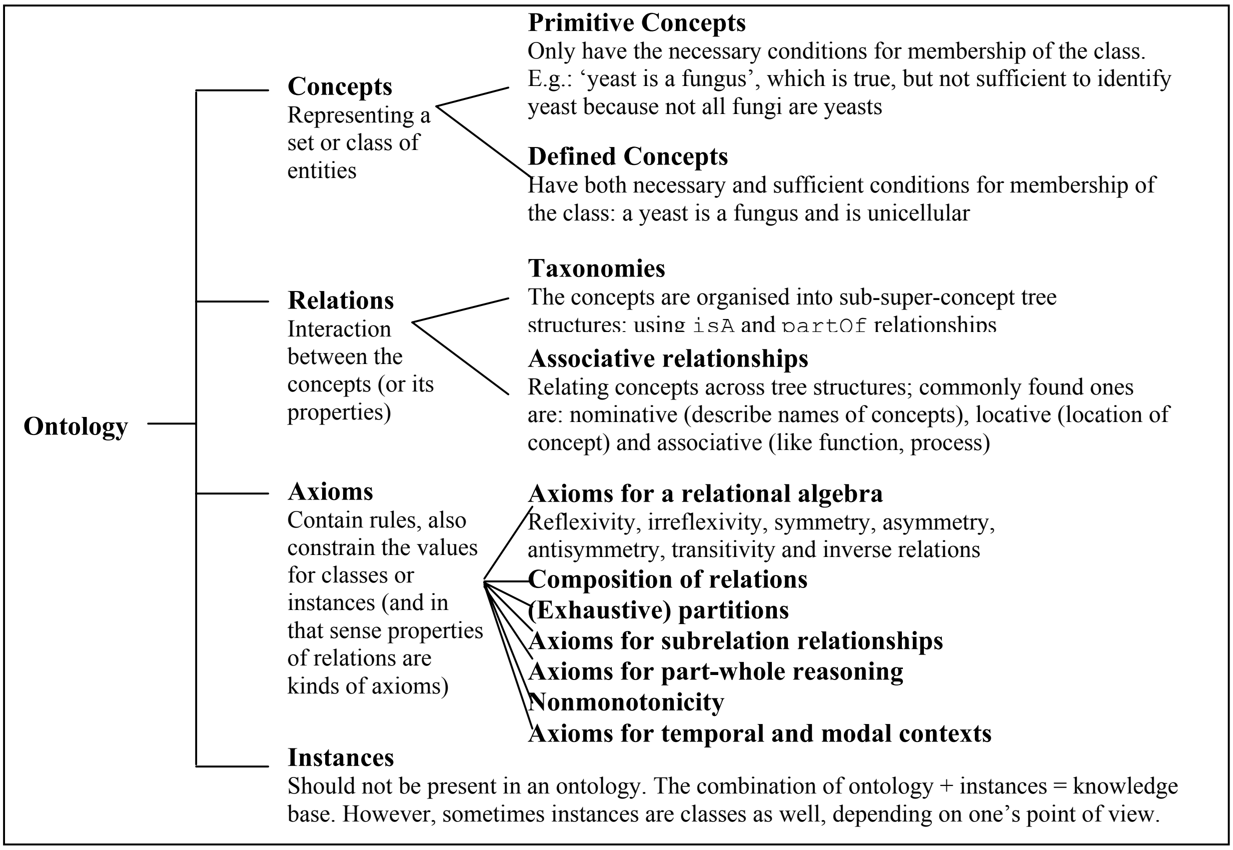
In summary, there are multiple factors that characterise data heterogeneity, like aggregation and naming, added with FOL and modelling paradigm heterogeneity in representing the data, which extend itself to the domain-level heterogeneity, including scope and attribute values, that all have an effect on interoperability and a source for conflicts and mismatches. On top of these aspects are the difficulties of biological data itself that add to challenges in resolving heterogeneity when integrating data and models.

[23] As discussed by Wiederhold (1994), which were by Hefflin and Hendler (2000) paraphrased and renamed into terminology, scope, encoding and context respectively.

2.2 Types of ontologies

Ideas of what an ontology is precisely, how one develops ontologies, the characteristics of an ontology, the level of formalism used to represent the ontology and so forth vary considerably, but there is agreement that an ontology captures consensus about the concepts of the UoD from the perspective of the subject domain experts. However, an ontology is not a specification of a conceptualization as Gruber initially phrased it in 1993 or the rather long definition by Van Heijst *et al.* (1997), but could be summarized as "a (possibly incomplete) agreement *about* a conceptualization" Guarino (1997a), which is the output of the "study of the categories of things that exist or may exist in some domain" (Sowa, 1997). *Figure 2.7* presents the components of an ontology.

Note there are differences between a conceptual model and an ontology, which has a knock-on effect on types of ontologies and the approach for ontology development. Although Andreasen and Nilson's (2004) observation that "formal ontology specification overlaps with conceptual modelling using formalisms such as entity-relationship diagrams or conceptual graphs, or proper logics such as description logic", is correct, the notion that they *overlap* does not indicate what each has as unique characteristics of itself. Jarrar *et al.* (2003) consider the distinguishing factors to be 1) the *consensus level* about ontological content, 2) a conceptual model is *static* offline, whereas an ontology is for direct use 'online' and *dynamic*, e.g. ontology querying software, and 3) an ontology is independent of an application that is developed, whereas a conceptual model is tailored toward the to be developed application. However, these factors only address the (perceived) differences in *usage* of ontologies and conceptual models.



Ontology

Concepts
Representing a set or class of entities

Primitive Concepts
Only have the necessary conditions for membership of the class. E.g.: 'yeast is a fungus', which is true, but not sufficient to identify yeast because not all fungi are yeasts

Defined Concepts
Have both necessary and sufficient conditions for membership of the class: a yeast is a fungus and is unicellular

Relations
Interaction between the concepts (or its properties)

Taxonomies
The concepts are organised into sub-super-concept tree structures: using `isA` and `partOf` relationships

Associative relationships
Relating concepts across tree structures; commonly found ones are: nominative (describe names of concepts), locative (location of concept) and associative (like function, process)

Axioms
Contain rules, also constrain the values for classes or instances (and in that sense properties of relations are kinds of axioms)

Axioms for a relational algebra
Reflexivity, irreflexivity, symmetry, asymmetry, antisymmetry, transitivity and inverse relations

Composition of relations

(Exhaustive) partitions

Axioms for subrelation relationships

Axioms for part-whole reasoning

Nonmonotonicity

Axioms for temporal and modal contexts

Instances
Should not be present in an ontology. The combination of ontology + instances = knowledge base. However, sometimes instances are classes as well, depending on one's point of view.

Figure 2.7. Components of an ontology. (based on Stevens et al.*'s (2000) descriptions; with the addition of axiom categories as discussed by Staab and Mädche (2000)).*

Bench-Capon *et al.*'s (2000) viewpoint of difference between the two is that "a schema is an ontology with little *extra structure* consisting of keys and the division of classes into entities and relationships" (emphasis added), elaborated on in §2.2.1 below. However, the presence of keys *ought* not to be a differentia, because a 'true' conceptual model is supposedly application-independent, whereas keys are specific for database development. One interpretation of Bench-Capon's 'extra structure' is, in my view, the specification of the relationships: not just that there *is* a relationship, but *how* these entity types or classes relate to one another, in particular the participation constraints and multiplicities. Further, a conceptual model captures only what is necessary in that *instance* of the analysis phase of the software development process, whereas an ontology includes, from the perspective of the application, 'non-essential' concepts, because it comprises what exists, or can exist, and thus will include more concepts, relations and axioms than a conceptual model. This is in line with Bowers and Ludäscher's (2003) interpretation, who see a conceptual data model simply as an instance of an ontology: as one particular combination of a subset of the larger ontology and is to be used for application development (see *Figure 2.8*). OntologyWorks[24] provides software that uses an ontology to automatically generate databases.

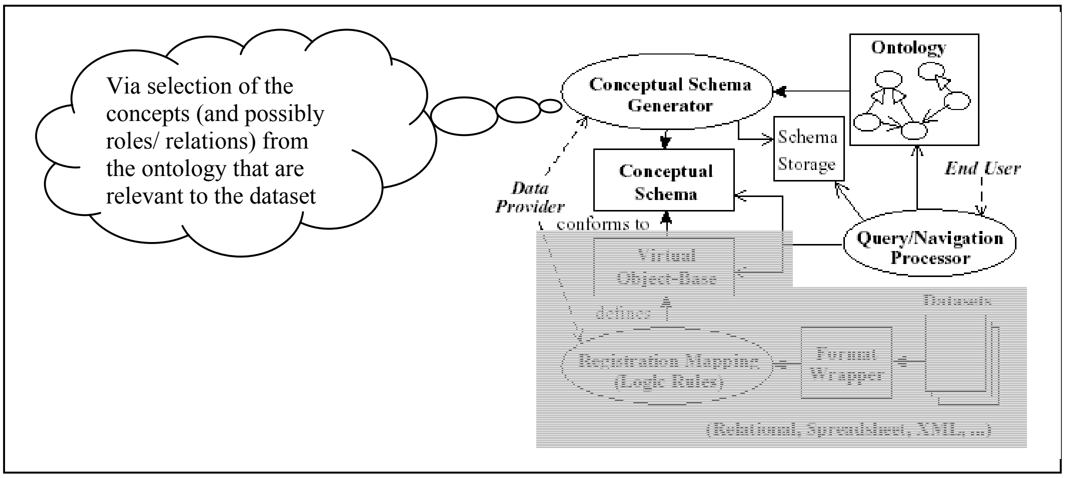

Via selection of the concepts (and possibly roles/ relations) from the ontology that are relevant to the dataset
Conceptual Schema Generator
Ontology
Schema Storage
Conceptual Schema
Data Provider
End User
conforms to
Query/Navigation Processor
Virtual Object-Base
defines
Datasets
Registration Mapping (Logic Rules)
Format Wrapper
(Relational, Spreadsheet, XML, ...)

Figure 2.8. Generation of a conceptual model based on concepts defined in an ontology.
(Source: modified from Bowers and Ludäscher (2003)).

These interpretations are not in contradiction with each other: Bowers and Ludäscher consider a conceptual model as a direct precursor for an application, by which is *not* meant an ontology querying tool, but for example database application like Keet's (2003c) bacteriocin database. Hence, one can think of having an ontology and using (a subset of) an ontology as a basis to create a conceptual model containing extra information on the type of relations between the concepts/entity types/classes that adhere to the meaning of the concepts as defined in the ontology. For example, Köhler *et al.*'s (2003) approach to map tables and attributes of a model of a database to concepts of an ontology, as pursued in their SEMEDA[25]. This also meets the idea of the static offline model mentioned above: an ER model is not used as a database[26], but translated to a computational model (logical model) and then to a physical model, each of which may be slightly modified compared to the original conceptual model, i.e. tailored for a particular purpose and thereby requiring a lower level of consensus than an ontology.

[24] http://www.ontologyworks.com

[25] SEmantic MEta DAtabase: http://www-bm.ipk-gatersleben.de/semeda/login.jsp. Observe though that they do the reverse as a starting point to develop ontologies.

[26] A conceptual model is generally a 'paper exercise' or created with a software tool like VisioModeler, in contrast with and separated from the actual DBMS like Oracle, MySQL etcetera.

2.2.1 Composition of ontologies

The most well-know divisions to categorise types of ontologies, is by their level of 'formal-ness': ranging from a list of terms, to concepts having relations and axioms. *Figure 2.9* summarises these distinctions, including other terminology for these differences as used by for example Corcho *et al.* (2003), discussing lightweight and heavyweight ontologies.

An interesting note, and for the subject matter of the research highly relevant, on prototype-based ontologies was made by Sowa (1997):

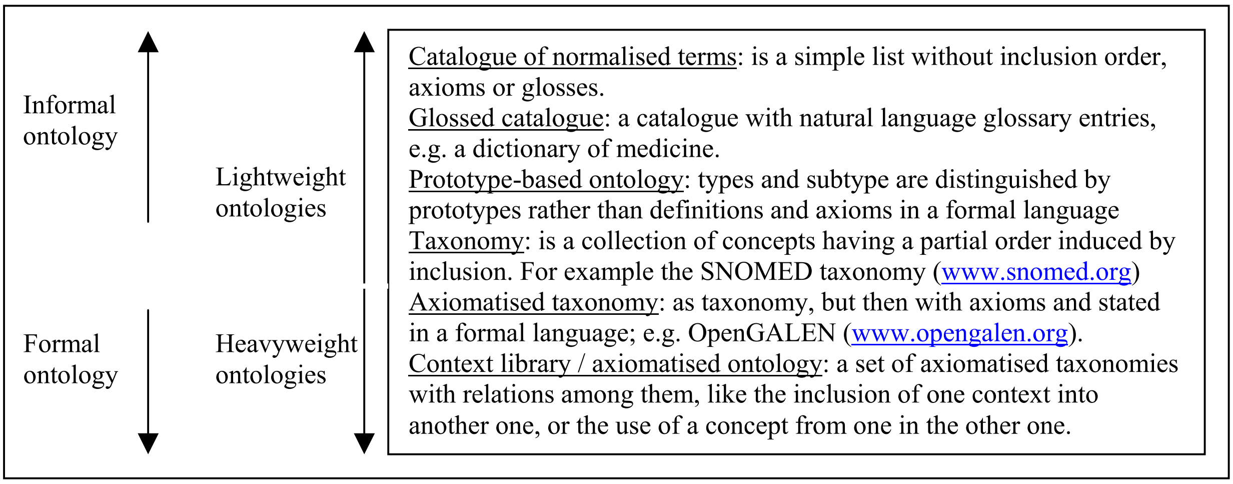
> Large ontologies often use a mixture of definitional methods: formal axioms and definitions are used for the terms in mathematics, physics, and engineering; and *prototypes are used for plants, animals*, and common household items. (emphasis added)

However, this is not the only method of categorising ontologies. One such variation is the *ontology base* and a *commitment layer*, as developed by Meersman and Jarrar (2002) and Jarrar *et al.* (2003) and graphically expressed in *Figure 2.10*. The advantage of this approach is that it separates the most general (most reusable) knowledge and places this in an ontology base, whereas a layer of ontological commitments is positioned between the ontology base and the applications.

The ontology base is composed of a set of context specific binary conceptual relations, called lexons. A lexon is represented as <γ: $\texttt{Term}_1$, `Role`, $\texttt{Term}_2$>, with context identifier γ, defining (γ, T) as a concept. The layer of commitments contains the ontology *view*, which refers to the lexons of the ontology base that are relevant and the ontology *rules*, where

> [e]ach ontological commitment corresponds to an explicit *instance* of an (intensional) first order *interpretation* of the domain knowledge in the ontology base. In other words, it is the role of commitments to provide the formal interpretation(s) of the lexons. (Jarrar *et al.*, 2003).

Resulting from this organisation of an ontology into two sublayers, a "conceptual schema can be seen as an ontological commitment defined in terms of the domain knowledge". The following example illustrates the use of the ontology base and commitment layers.



Catalogue of normalised terms: is a simple list without inclusion order, axioms or glosses.
Glossed catalogue: a catalogue with natural language glossary entries, e.g. a dictionary of medicine.
Prototype-based ontology: types and subtype are distinguished by prototypes rather than definitions and axioms in a formal language
Taxonomy: is a collection of concepts having a partial order induced by inclusion. For example the SNOMED taxonomy (www.snomed.org)
Axiomatised taxonomy: as taxonomy, but then with axioms and stated in a formal language; e.g. OpenGALEN (www.opengalen.org).
Context library / axiomatised ontology: a set of axiomatised taxonomies with relations among them, like the inclusion of one context into another one, or the use of a concept from one in the other one.

Figure 2.9. Classification of kinds of ontologies, based on the level of formalism utilised.
(The diagram combines information from Gangemi *et al.* (1998), Corcho *et al.* (2003) and Sowa (1997)).

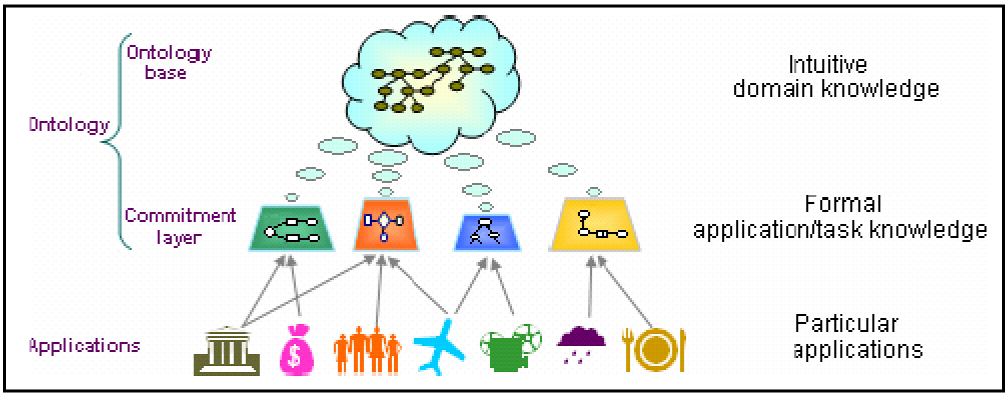


Figure 2.10. Subdivisions of ontologies, as used in the DOGMA approach. (Source: Jarrar *et al.*, 2003)

Example 3. Ontology base and commitment layers

Table 2.1 shows an example of (a fragment of) an ontology base with the contexts `Microorganisms` and `Diseases`, where a lexon reads for example `{< Microorganisms: Microorganism, StoredAt, CultureCollection >}`. These lexons can be used to create ontological commitments, for example for a microbiology department who maybe wants to use the ontology base to create a teaching aid (*Figure 2.11*), and a commercial enterprise like the American Type Culture Collection[27] selling freeze-dried inoculants (*Figure 2.12*). Alternatively, someone else may be more interested in microorganisms that cause diseases (*Figure 2.13*). Comparing the table entries with the three figures, one can see that an entry in the `Role` column corresponds to a fact type (a rectangle in the figure) and a term to an object (ellipse). However, the figures are not just simple graphical representations of the rows in *Table 2.1*, but contain additional rules (constraints), graphically represented as arrows and blobs in this example. An example of these extra semantics captured in the commitments is displayed in *Figure 2.14*, containing a screenshot of the verbalizer output from VisioModeler v4.1, with which the figures were created: the fact type between `MicroOrganism` and `CCNumber`. In another setting, there may be another commitment (rule/constraint) between the two, or be absent as in *Figure 2.11*.

Note that in this interpretation, these three ontological commitments may be (part of a) conceptual model. Analogous to Bowers and Ludäscher's view shown in *Figure 2.8*, this author performed the task of the Conceptual Schema Generator when creating the models as included here in *Figures 2.11-13*.

Table 2.1. Example of an ontology base.

Context	Term_1	Role	Term_2
Microorganisms	Microorganism	IsAn	Organism
Microorganisms	Microorganism	Has	LatinName
Microorganisms	Microorganism	Has	CCNumber
Microorganisms	Microorganism	StoredAt	CultureCollection
Microorganisms	Microorganism	PurchaseCost	Price
Microorganisms	Price	Has	Value
Microorganisms	Price	Has	Currency
Microorganisms	LatinName	Has	LatinNameFamily
Microorganisms	LatinName	Has	LatinNameSub
Microorganisms	LatinName	Has	LatinNameSubSub
Microorganisms	LatinName	Has	Designate
Microorganisms	Microorganism	SupertypeOf	Bacterium
Microorganisms	Microorganism	SupertypeOf	Fungus

[27] http://www.atcc.org

Microorganisms	Microorganism	SupertypeOf	Archae
Microorganisms	Microorganism	Has	Morphology
Diseases	Disease	Has	DiseaseName
Diseases	Disease	IdentifiedBy	WHO_ID
Diseases	Disease	CausedBy	Cause
Diseases	CausativeAgent	SupertypeOf	Infection
Diseases	CausativeAgent	SupertypeOf	Poisoning
Diseases	Disease	Has	Symptoms
Diseases	Infection	By	Microorganism
Diseases	Infection	By	Virus
Diseases	Infection	By	Worm
Diseases	Poisoning	By	Microorganism

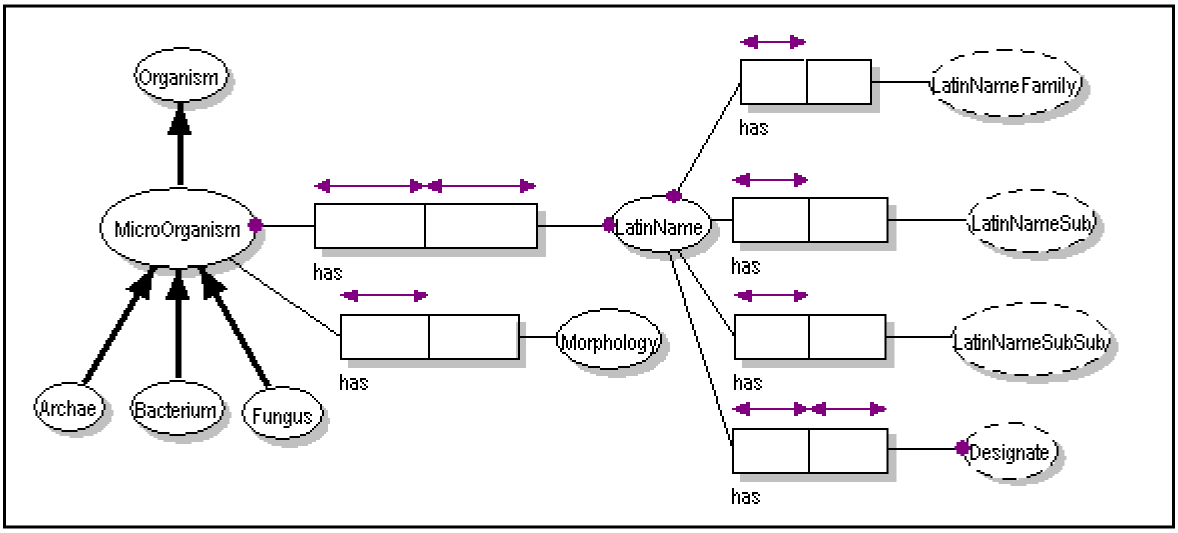


Figure 2.11. 'MicroBio department' Ontological Commitment.

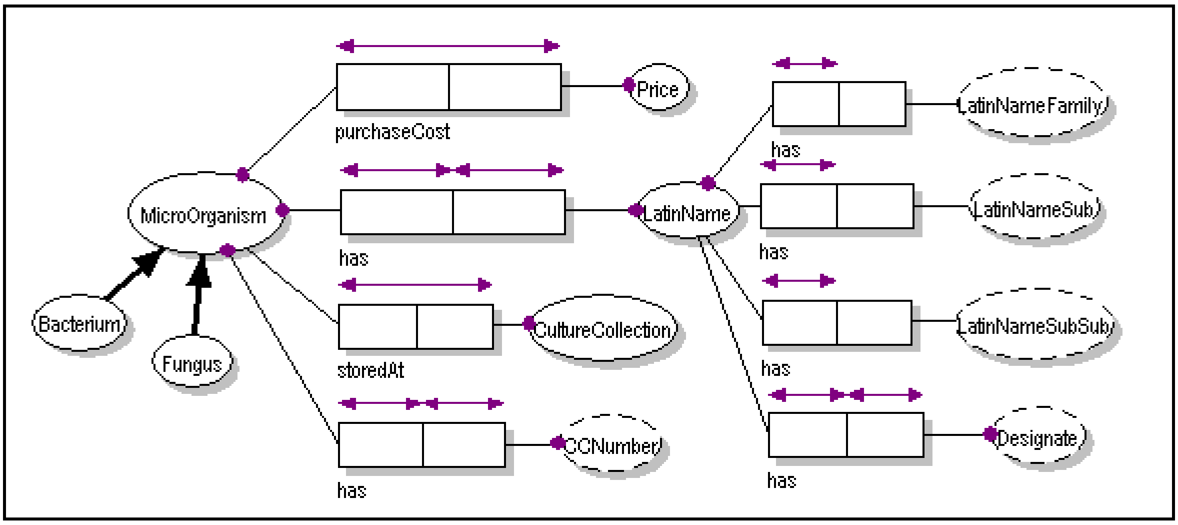


Figure 2.12. 'Culture collection' Ontological Commitment.

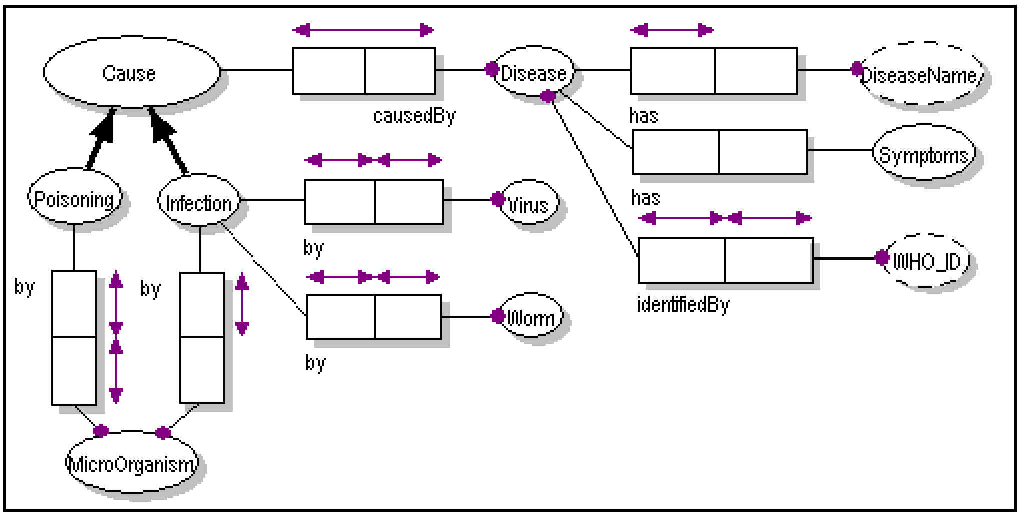
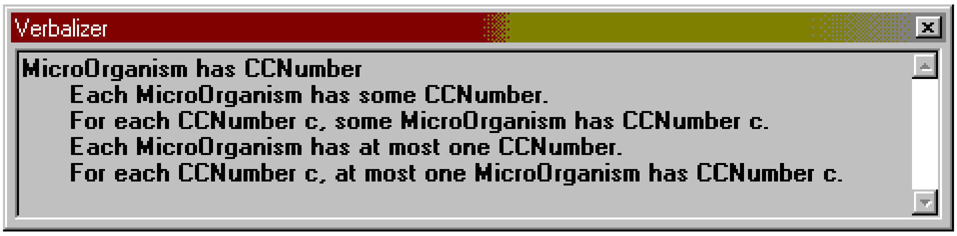


Figure 2.13. 'Diseases' Ontological Commitment.



```
MicroOrganism has CCNumber
    Each MicroOrganism has some CCNumber.
    For each CCNumber c, some MicroOrganism has CCNumber c.
    Each MicroOrganism has at most one CCNumber.
    For each CCNumber c, at most one MicroOrganism has CCNumber c.
```

Figure 2.14 (Near) natural language of the ontological commitment rules between `MicroOrganism` *and* `CCNumber`.

Bench-Capon *et al.* (2000) have a very similar interpretation on how ontologies and schemas are related to one another. They start with an ontology *O* and construct a [conceptual] schema *S* with part, or all, concepts of the ontology, from which a new ontology may be created – bottom-up – O_S: because of creating a schema (conceptual model / ontological commitments) from an ontology, new relationships and entity types were introduced that need a 'place' in the ontology O_S, thereby providing more structure than *O*, but because some axioms of *O* were lost when creating *S*, there is also less structure (or knowledge) captured. The manner in which *O* and O_S are related to each other "through their 'common denominator' – an ontology without the axioms of *O* and without the extra classes of O_S", which is remarkably similar to ontology base.

Pinto *et al.* (1999) describe not so much a framework of possibilities on how ontologies are composed, as discuss it in the context of methodologies of ontology development in relation to ontology integration (see also chapter 4). One of these considered is Grüninger (1996): ontologies are built based on building blocks and foundation theories, somewhat resembling the DOGMA[28] ideas with the ontology base and commitments explained in *Example 3*. A foundation theory is a "set of distinguished predicates and functions together with some axiomatization" and the building blocks are the "classes of theories using the predicates and functions in the foundation theories". METHONTOLOGY uses incremental ontology creation (Fernandez) and focus on the *knowledge* level, thus *neither* at the symbol level *nor* the implementation level (Pinto *et al.*, 1999). This is also carried out by Guarino (1997b), who first focussed on theories of parts, wholes, identity, dependence and universals, from which to create top-level ontological concepts into a backbone ontology to subsequently expand it with other basic categories

[28] Description is available on the DOGMA website of the VUB: http://www.starlab.vub.ac.be/research/index.htm

(functional, biological etc.). A third example is the Geographic Ontology[29], consisting of `isA` subsumption relationships between the concepts. However, several of these concepts contain an alternative description for the name of the concept, such as `Geographic Objects [Independent Geographic Entities]` and mini-explanations, e.g. `Processual Entities [Exists in space and time, unfolds in time phase by phase]`, as well as examples of the kind of things that belong to a particular concept, e.g. `Physical Attributes` has a separation line in the box and underneath written "soil types, elevation at a point" to clarify what is meant with the modelled concepts. This provides clarity for both the ontologists what they call something and additional information to make it unambiguous for the geography discipline. At the same time, the divisions in the concepts are of a high level, can be separated further and eventually may, or might not, integrate the three 'views' (snap, field and span) with other types of relationships and axioms, i.e. move up in the hierarchy of formalness of an ontology.

There are more ontology development processes, which can lead to different interpretations and representations of the consensus of 'what is' – even though this may sound as a contradiction in terms – which does affect the composition, and resulting from that, integration, of ontologies elaborated on in chapter 4.

2.2.2 Topics/kinds of ontologies

Similar to the classification of types of ontologies, one can categorise the different purposes of ontologies, shown here in *Figure 2.15.*

Representation ontologies: contain specifications of the conceptualisations that underlie knowledge representation formalisms.
Top-level ontologies: to describe generic and intermediate ontology concepts. This can be on top of a domain ontology or as stand-alone effort; main aspect is domain independence. See e.g. Sowa (2001) for an example of a top-level ontology.
Generic ontologies consist of the general, foundational aspects of a conceptualisation
Intermediate ontologies are slightly more tailored towards a conceptualisation of a specific domain. There may not be references to generic ontologies, less/loose axiomatization.
Domain ontologies specialize in a subset of generic ontologies in a domain or subdomain.

Figure 2.15 Kinds of ontology modules. (Source: paraphrased from Gangemi *et al.*, 1998)

In addition, different amounts of detail in the ontologies and levels of logic in the representation are more (or less) relevant depending on the application, like information retrieval, machine translation, problem solving, database question answering and automatic programming (Sowa, 2000). However, this seems mildly in contradiction with the idea of an ontology, supposedly 'entirely' independent of the intended use, although one can interpret the independency in this context as being independent of computational models. I have excluded the 'application ontology' from *Figure 2.15*, because I do not consider them ontologies as such, but more alike conceptual models renamed as 'ontology' whilst having a lower degree of consensus amongst the SMEs and thereby less reusable.

Another avenue of looking at the 'different kinds of ontologies' is when designing large ontologies to partition this into smaller topics encompassing specific (sub-)disciplines in order to be able to create the ontology/ies. An example of dividing an interdisciplinary UoD is to make use of topic spaces (Pepper, 2000), as experimented with developing the fisheries ontology (Gangemi *et al.*, 2002a), reprinted in *Figure 4.8*. Gramene[30] and The Plant Ontology[31] have divided their ontology efforts into the

[29] http://ontology.buffalo.edu/bfo/GeO.pdf
[30] http://www.gramene.org and Jaiswal *et al.* (2002).
[31] http://www.plantontology.org and The Plant Ontology Consortium (2002).

following categories, whereby the Gene Ontology is taken from the Gene Ontology Consortium[32]; the intention of the trait ontology is to facilitate phenotypic comparison within and between crop genera:

- **Plant Ontology**
  - Plant Anatomy (morphology, organs, tissue and cell types)
  - Growth stages (plant growth and developmental stages)
- **Trait Ontology**
  - Plant traits and phenotypes (agronomic, mutant phenotypes, quantitative trait loci)
- **Gene Ontology**
  - Molecular function
  - Biological process

Figure 2.16. Distinct areas within Gramene and the Plant Ontology.

There is, however, a caveat with such an approach. The Gene Ontology forms an intricate part of the Trait Ontology, hence when the Gene Ontology is updated, this *will* have a knock-on effect on the understanding, use, or even the categorisation, of the Trait Ontology. Take for example Kumar and Smith's (2003) analysis on the defects of the Gene Ontology, highlighting incorrect use of continuant, occurent, dependent and independent entities, the "confusion in the distinctions between *functions* and their *functioning*" and between function and activity, which will require changes to be made to the Gene Ontology. Ceusters *et al.* (2003) use *endurant* instead of continuant, and add "confusion" problems in ontologies between *instantiation* and *subsumption* and distinguish *entity* from *term* which are important factors when 'cleaning up' ontologies in order to meet semantic correctness and stricter requirements on formalisms Guarino and Welty (2002) provide a methodology for validating taxonomies with OntoClean. However, one also can interpret this as a 'moving up' in the kind of ontology hierarchy: from lightweight to heavyweight and from domain to generic or even top-level ontology. But different groups approach ontology creation from different angles, e.g. emphasising semantic correctness from the SME perspective or approaching the task from the ontologist's point of view, and/or may be at a different 'stage' in the ontology development process (see also *Example 4* below). Breaking up the domain into sub-domains will result in different kinds of ontologies, hence be a potential source for conflicts and mismatches due to loss of oversight of the whole and potential obfuscation of internal/external dependencies so that a minor revision may result in a large ripple-effect that may, or may not, be detected in due time.

SEEK plans to create ecological ontologies, though it is not yet specified which (sub-)domain will be covered in what ontology, apart from the ontology on plant taxonomy and an ontology of units and measurements. Thus, instead of one monolithical ontology, the drive is towards multiple smaller ontologies, increasing the likelihood of encountering conflicts and mismatches on multiple levels as indicated above, hence increasing the challenge of ontology integration. The amount of detail in e.g. organism taxonomy is not always desired (read: not known or not important in some context) in ecological systems; e.g. "omnivorous sucking mites" and "fungivorous nematodes" in the soil food web (De Ruiter *et al.* (1994a) as discussed in Akkermans *et al.* (1996b)) do not necessarily fit in the same taxonomic family as determined by either the Linnaean or cladistic system (or both). The next example takes some of the topics raised in this paragraph and applies it to the Descriptive Term Ontology (Paterson *et al.,* 2004 *in press*).

Example 4. The Defined Terms Ontology

With this information on ontologies, the model of *Example 1* is revisited here. Although in *Example 1* the diagram of *Figure 2.2* was treated simply as an OO model, it is meant to represent the *main aspects* of the Descriptive Term Ontology (DTO). There are three general factors to analyse: type of ontology, the kind of ontology module and the modelling paradigm used to represent the ontology.

[32] http://www.geneontology.org

- The DTO is a taxonomy, i.e. a collection of terms, their definitions and organised primarily via the partial order `isA` relationships and occasionally `partOf` relationships (the `StateGroup` with `States`, `Structure` with itself). However, three other relationships, 2x `appliesTo` and `describes`, are identified as existing, but in Paterson *et al.*'s (2004 *in press*) model neither formalised, nor even have any indication or structured information about these relationships between the concepts. The ORM exercise did clarify certain aspects, which can aid development of the ontology from the level of lightweight/informal taxonomy toward a heavyweight ontology (of the type axiomatised taxonomy). At present, it seems to be a mixture of the two ontology types, "semi-formal" according to one's own definition.
- Recollecting *Figure 2.15*, the subject domain of the DTO is tailored specifically for the plant taxonomists – even a subset thereof, hence one can categorise the kind of ontology as a *domain ontology* or maybe even an 'application ontology' (the authors refer to it as a prototype ontology). On the other hand, it is to contain top-level aspects such as measurements and other quantitative properties (e.g. `length`) as well as so-called modifiers, which include spatial structures (`sphere`) and temporal aspects. Do the researchers attempt to reinvent the wheel concerning top-level ontological concepts? Do they (intend) reuse (a section of) an existing top-level ontology, such as the maths ontology? This seems to be a mixing of two kinds of ontologies.
- Another aspect relevant to customizing the ontology for (plant) taxonomists is that some of the vocabulary used is incompatible with the more formal approaches such as the Semantic Web, its ontology infrastructure WonderWeb[33] and DOLCE which forms part of this effort. DOLCE is an ontology of particulars. Particulars differ from universals, in that the former has no instances whereas the latter are entities that do have instances; further, properties are universals as opposed to *qualities*, which are particulars (Gangemi *et al.*, 2002b). Qualities are "the basic entities we can perceive or measure: shapes, colors, sizes…lengths" and have a 'value' called *quale* that "describes the position of an individual quality within a certain *conceptual space*". Consequently, `Property` in the DTO is a *quality* according to the WonderWeb. Secondly, DOLCE's *state* is subsumed by stative perdurants and comprise e.g. 'being open', not a `State` with example 'oval' as mentioned in the explanation of the DTO. Consequently, DTO's `State` matches WonderWeb's *quale*. A third example of the difference is the `Structure`s in the DTO, which mention 'hairs' and pores' as example, but qualify as *features* in DOLCE, which is categorised under Entity – Endurant – Substantial – Physical-Substantial and, as formulated in Gangemi *et al.* (2002b), means that each *feature* is an essential whole "but no common unity criterion may exist for all of them… [they] have a topological unity, as they are singular entities…may be *relevant parts* of their hosts", such as the hairs on a leaf are part of that leaf. A complete analysis falls outside the scope of this document, but considering the intention to expand the DTO to a wider subject domain and maybe integrate it in larger ontology systems, it is important to note that the DTO as is may need to be adjusted to match those conventions. Even if there is no desire to have any link with ontology developments related to the Semantic Web, one still may want to make the model consistent via e.g. the OntoClean (Guarino and Welty, 2002) methodology.
- The critique on the modelling paradigm used, which was initially interpreted to be OO but meant to be "no paradigm in particular", does not limit itself to this particular instance of modelling an ontology, but is of a more general nature: OO/UML class diagrams are limited in their capability to represent semantics fully and are closer to being a computational model than a conceptual model, let alone have sufficient expressive power for a formal ontology. One can argue that such a model suffice when developing a lightweight (informal) ontology, but ideally one would want to be able to formalise the subject matter to improve

[33] http://wonderweb.semanticweb.org/

> interoperability and reuse of the knowledge captured in the ontology, eventually. If one were indeed to move on towards formalizing the knowledge, as seems to be the direction, OO/UML is not expressive enough to capture all rules and one would need to ‘translate’ the existing model into whichever formal representation one chooses. In this perspective, it would be more advantageous to use a more expressive conceptual modelling methodology from the start.

Last, the ‘routes’ `State-Property-Structure` and `State-StateGroup-Structure` are, depending on phrasing of the question, “alternative drawings of the same thing” or “our interpretation is better structured and more comprehensive and includes quantitative properties as well” (see *Example 1*). If the former, then it would be incorrect to include both routes in an ontology because the ‘what is’ should be included only once. If the latter, this means that there is no consensus on the matter, and therefore should not yet be incorporated into one ontology; or at least mentioning separately that there are dissimilar interpretations on how `State`s relate to `Structure`s and that further research will assist in identifying the appropriate ontological relations of the knowledge. It is here that the approach of an ontology base and a separate commitment layer may be helpful by separating the terms/concepts involved from their rules. However, at the time of writing there does not seem to be an agreement on the concepts that would go in the ontology base, which makes defining one commitment layer very difficult, if not impossible. On the other hand, the DOGMA methodology facilitates the two interpretations to be represented – each in its own commitment – hence can be recognised, tested and analysed as such.

An interesting aspect that shines through the DTO development is the conflicting demands between the call for more structure, rigour and categorisations (and some sort of standardisation) by the computing scientists, the taxonomists who are used to the “natural language/free text descriptions” and the attempts by the informaticians to accommodate this lack of standardisation to some extent. Considering the ‘free text freedom’, one might consider to investigate if the advances in natural language processing have potentially useful technology to offer.

The Descriptive Term Ontology is in the process of being developed and is in need of a clarification on the type and kind of ontology pursued – or a justification why not to follow such a route – and may benefit from adhering to the idea of achieving consensus before suggesting there is one ontology to capture the descriptive terms.

□

2.3 Conflicts and mismatches

Conflicts and mismatches can occur on various levels, as the reader may have inferred from the previous paragraphs on data and domain heterogeneity. Here, the effects of different kinds of logic, domain differences and types of mismatches are briefly discussed. Visser *et al.* (1997) distinguish two main types of mismatches: conceptualisation and explication. The conceptualisation mismatches cover largely the heterogeneity addressed above, where either the concepts or its relations mismatch with their counterparts in other ontologies; explication mismatches deals with the manner the conceptualisation is specified, using any combination of mismatches between term (*T*), the definiens (*D*) defining the concept and ontological concept (*C*). For example, a *CD* mismatch occurs when T_1 and T_2 are the same, but differ in definiens and concept, hence *T* is a homonym. Mitra *et al.* (1999) identified them slightly different: apart from semantic mismatches [of ontologies], there are structural mismatches (Goh’s ‘generalization’ in *Figure 2.1*), instance (something is an instance of `class1` and in the other ontology it is of `class2`, where `class1` ≠ `class2`) and granularity (more or less comprehensive hierarchies). One can add to this instance-class mismatch, where in one model the element is defined as a class(/entity) and in another as an instance. Klein (2001) categorised the problems of ontology combination for any subject domain; *Figure 2.17*, contains an adapted version of Klein's diagram, which is likely still incomplete, but does provide a useful overview. The numbers in the figure indicate a clarification is provided beneath the figure, while the next chapter addresses some of the issues raised here with relation to ontology integration efforts.

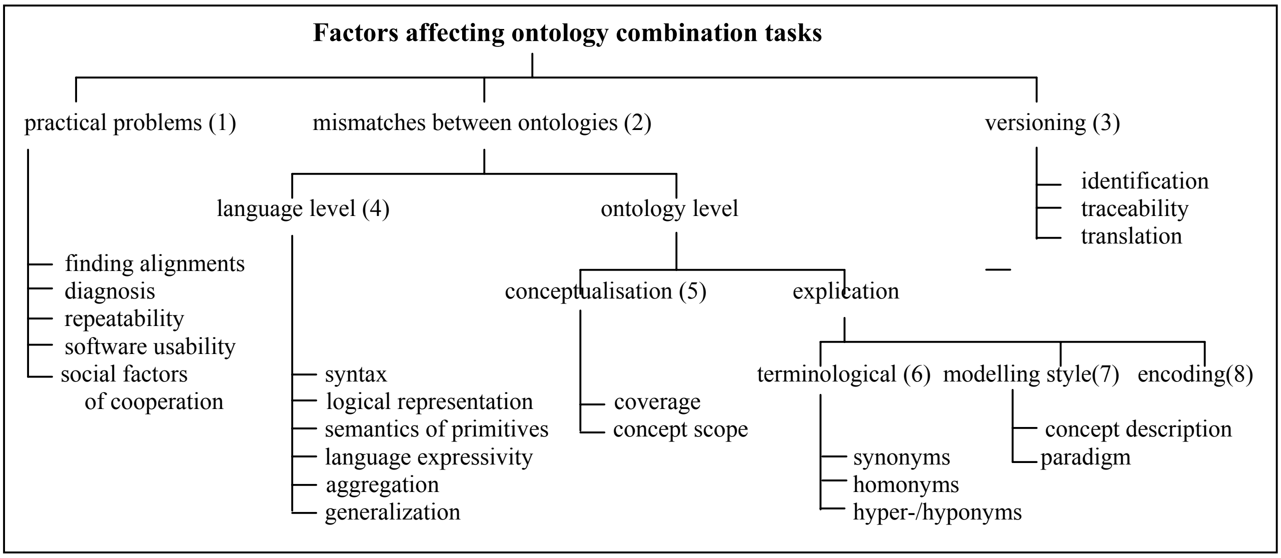


Figure 2.17. Schema of problems in ontology combination tasks.
(Source: based on Klein (2001), augmented by this author)

(1): 'Finding alignments' is not only a practical problem in the sense of manual and (semi-)automatic processes (see also chapter 5) based on computerisation of the heuristics otherwise characteristic for the manual procedure, but also can be considered as a difficulty of matching, or addressing the mismatches, of ontologies as organised under (2).

(2): Refer to chapter 4, ontology integration, and aspects addressed in this chapter such as kind and type of ontologies.

(3): This covers management of ontology revisions and tracing these changes. This author interprets 'translation' as for example the polder example in §4.2.1 as well as the straightforward translation of the labels of the concepts between different natural languages, like the concept with English name `Tree` as `Baum` in German.

(4): Think of CGs, RDF, KIF and various description logics. See also §2.4.1 and Sowa (2000) in particular; an example of the use of description logics for bioinformatics ontology is the TAMBIS Project[34], which used the GRAIL concept modelling language (Baker *et al.*, 1999).

(5): These include Wiederhold's scopes as mentioned in §2.1.3, and the marking out of the domain.

(6): In addition to the synonyms and homonyms, this author added hypernyms and hyponyms, matching one concept of ontology O_1 with another concept in O_2 that has a slightly broader respectively narrower definition than the concept in O_1.

(7): Recollect Visser *et al.*'s paradigms of ER versus OO in §2.1.3 and *Example 1*.

(8): For example using different natural language (Italian or Finnish).

[34] http://imgproj.cs.man.ac.uk/tambis/index.html

3. Pilot experiment

The brief investigation contained in this chapter draws together multiple aspects discussed and analysed in chapter 2. These include, but are not limited to, differences between computing models and modelling software used by ecologists, biological and ecological data characteristics, and the methodology of using placeholder objects to capture an extended semantic representation of equations is assessed. It also serves the purpose to prepare for a larger research project, such as the possibility of analysing and (re)modelling the LEEDS model[35], so that formulation of more comprehensive research questions and/or hypotheses may be achieved. Another aim was to investigate the prospective for a bottom-up approach to create computing models, such as conceptual models and ontologies, based on existing ecological models that are already captured in modern modelling software such as STELLA. This software is used by ecologists in both research and education for systems analysis and creation of simulations of phenomena such as predator-prey interaction, effects of contamination and food chains.

3.1 Methodology

A preliminary pilot experiment was carried out with STELLA software v8 for Windows from High Performance Computing and a demonstration model provided, called *Amalgamated Industries*.

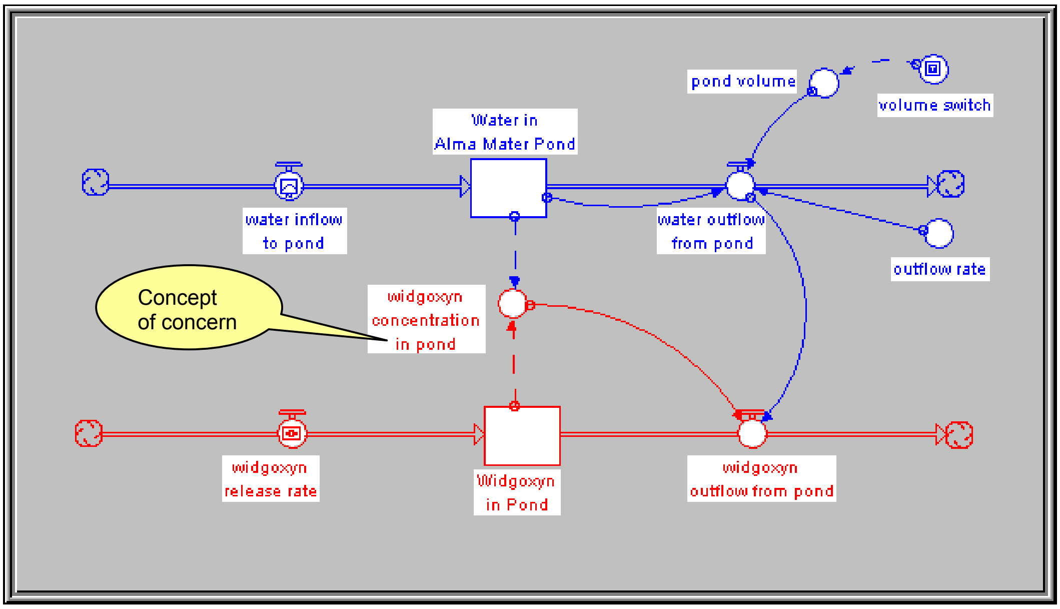


Figure 3.1. Model of the Amalgamated Industries demo.

This representative ecological model with *Amalgamated Industries* captures a scenario where a factory disposes toxic waste (*widgoxyn*) in the river, which flows into the pond downstream on the university complex (*Alma Mater*) that may in turn kill organisms living in the pond depending on the pollutant concentration (*Figure 3.1*). The concept of concern is the concentration of the pollutant in the pond at the centre of the diagram, which has the influencing factors modelled 'around' it, such as the released amount of pollutant by the chemical plant and the amount of water entering and leaving the system. The analysis

[35] Lake Eutrophication Event, Dose, Sensitivity-model, see e.g. Malmaeus and Håkanson (2004) for an example.

procedure in this pilot experiment involves the abstraction of this model and matching computing jargon with the modelling elements of STELLA, the reorganisation of the abstraction into a lightweight ontology (a taxonomy) and the accommodation of the extended semantics of the formulae into a model containing placeholder objects that matches the taxonomy. Paragraph 3.4 contains a discussion of the results and §3.5 some concluding remarks.

3.2 Model abstractions

3.2.1 Ecological and computing concepts

Before analysing the *Amalgamated Industries* model, *Figure 3.1* was simplified and generalised into an abstract pollution scenario as shown in *Figure 3.2*, which conveys three main aspects:

1) Water and pollutant flow in and out of the bounded system,
2) The combination of water volume and amount of pollutant determine the actual pollutant concentration in the pond, and
3) The combination of water outflow and pollutant concentration determines the amount of pollutant outflow.

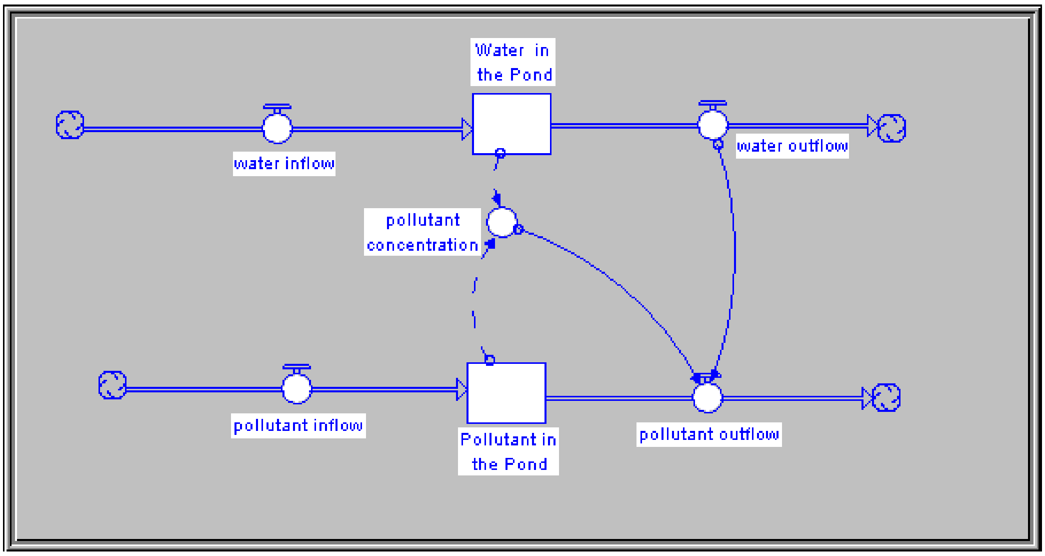


Figure 3.2. Abstraction of the pollution example.

There are two factors of interest in comparing this type of model with its variants in computing, such as class diagrams and ontologies:

- ∗ The ecological model is *event centred*, hence contains the representation of the concept of time, diagrammatically represented with the horizontal thick arrows with an open shaft, or phrased as the *route* taken by energy or a nutrient (in e.g. a food web). This is in stark contrast with most computing models, which centre around objects and their relations.
- ∗ Key aspects in the ecological model are the flow, stock, converter and action connector; *Figure 3.3* contains the comparison with computing verbiage (top half). `Object` is a candidate for a class or (entity) type, `event_or_activity` in OO terms a candidate for a method and in an ontology categorised under a function or activity hierarchy and converter maps to `attribute_or_property`, which says something about the object, such as the outflow rate.

The action connectors (thin lines with arrows) may be candidates for binary (ternary?) relationships between any two of flow, stock and converter.

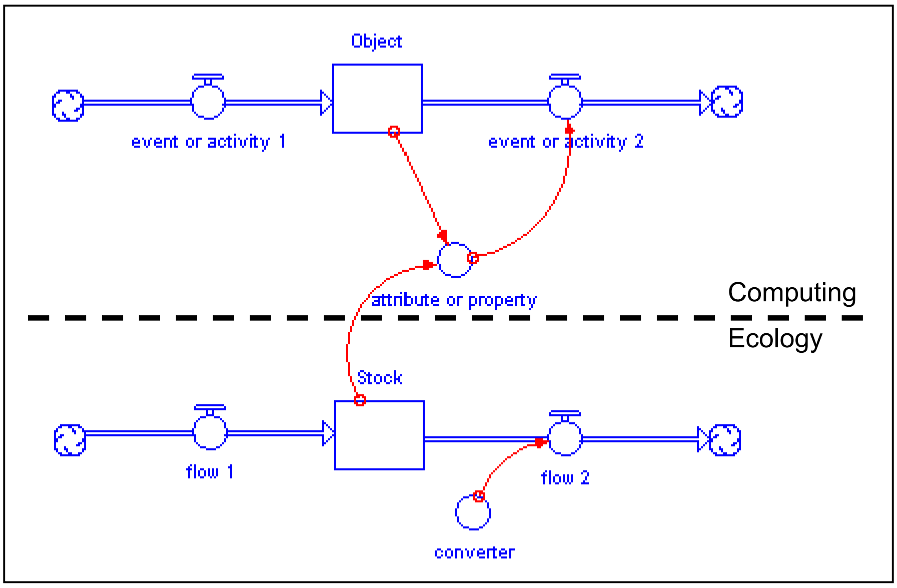


Figure 3.3. Comparing the ecological model components with its analogous element in a computing model.

What consequences do these differences have on 'translating' an ecological model into a conceptual model or ontology? Primarily, temporality and the movement of energy or nutrients are normally not considered in the conceptual models or ontologies, hence is unlikely to be represented exactly as is captured in the ecological model. However, what is possible is incorporating the fact that types of events, such as inflow and outflow, exist. Together with the almost one-to-one translation of stock to object/class/entity type and converter to attribute/property/value type, the original ecological model can be remodelled into a computing model consisting of three features:

- Entity types with attributes,
- Categorisation of events, and
- The relationships between them.

The expectation is that the latter, which includes the accommodation of action connectors, likely will be the most challenging. Key task then is how one can formalise this correspondence, and consequently the formalisation of the ecological knowledge itself. If this is possible, it should also be within reach to automatically generate an ontology base and populate this by loading several of the STELLA ecological models, such as *Amalgamated Industries* and the LEEDS model of Malmaeus and Håkanson (2004). In essence, the models as depicted in Figure 3.1 and 3.2 are readily available intermediate models analogous to Aguado *et al.*'s (1998) conception of an "intermediate model", thereby functioning as common ground for communication between the disciplines computing and ecology.

3.2.2 A lightweight ontology

However, before exploring the possibilities of formalisation and automation, a quick manual exercise was conducted to provide a sketch with an indication of a possible taxonomic representation, restricted to concepts and the `isA` and `partOf` relationships (*Figure 3.4*). Before analysing the figure, a few explanatory notes are in order. The `isA` relationship between `Water` and `Molecule` is grey, because although `Water` is indeed a molecule, `Water` *in the context of some ecological site* is not meant as pure H_2O, but water containing dissolved molecules and suspended particles. *Figure 3.4* is easy to draw and, from the perspective of computing, straightforward to understand, but the methodology of ontology base &

commitment layers would have been more advantageous (see also §2.2.1 and *Example 3*). This, because "water is a molecule" can be included in the ontology base and omitted from a commitment layer in the context of the ecological site, whereas it would be included in a commitment layer of a chemicals ontology (which would omit the "water is a liquid mixture"). The three concepts `Volume`, `Rate` and `Concentration` are grey as well, because they capture a characteristic of their respective concept they are attached to, alike an attribute, and are not strictly a part of the concept each is related to.

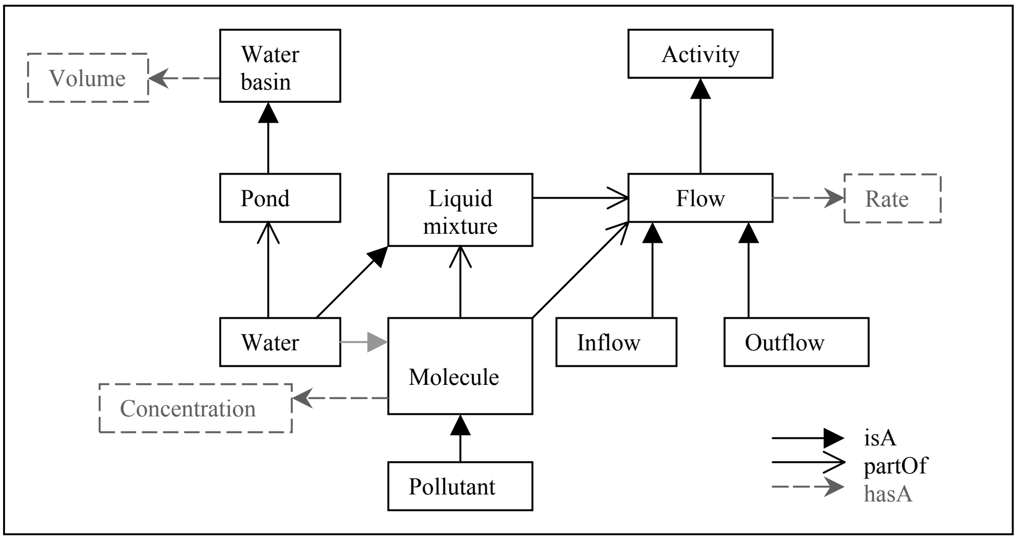


Figure 3.4. Taxonomy of directly *relevant concepts.*

Of course, water can occur in the frozen state, but then the water flow and change in pollutant concentration is assumed zero, and can be considered a state that water can be in, not a characteristic of water (a characteristic would be expansion of water during freezing). The current STELLA simulation demonstration includes drought and flooding, but not the effect of a frozen river and pond; analogous is the design decision on in/exclusion of rainfall and vaporisation. Considering possible future expansion and reusability, one may wish to include it in the taxonomy. This notion is also valid for concepts such as the `WaterBasin` with hitherto omitted subtypes such as `River`, `Sea` and `Lake`, where inclusion of `WaterBasin` in *Figure 3.4* may appear inconsistent, but serves the purpose of illustrating the abstraction and potential for reuse.

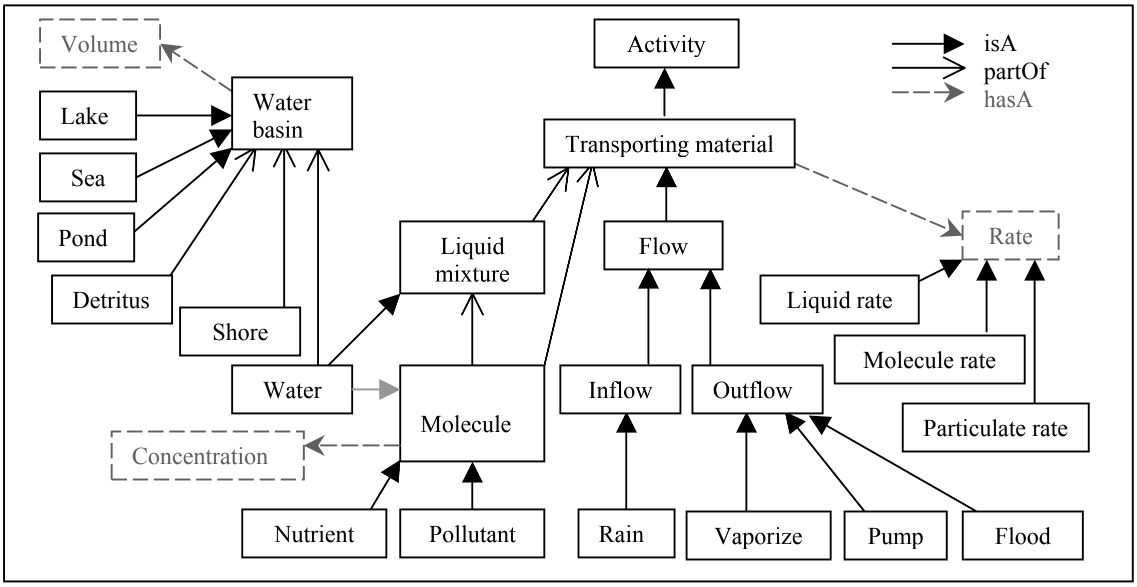


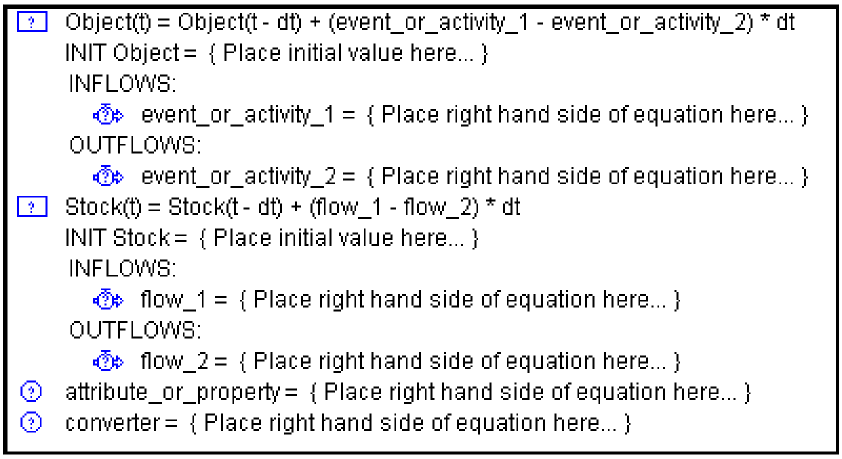
Figure 3.5. Enlarged taxonomy of the pollution scenario

A more comprehensive taxonomy is drawn in *Figure 3.5*, containing more than double the amount of concepts relative to the directly relevant concepts in *Figure 3.4*, hence has the capability of higher occurrences of reuse for a wider variety of pollution simulations.

A striking difference is the apparent loss of coherence of the subject matter due to the move from ecological to computing model: the so-called *concept of concern* used as the start to develop all elements in the ecological model is now merely one part of many and not included in a taxonomy-as-ontology that is restricted to `isA` and `partOf` relationships.

3.3 Formulae and placeholder objects

The STELLA (and related iThink) software captures formulae in the background, and are more obscure than the placeholder objects of Keller and Dungan (1999). *Figure 3.6* is a screenshot of the software's automatically generated framework to finish the calculations that are assumed relevant. Albeit using the labels of the corresponding figure (here *Figure 3.3*), with larger models, the generated text-based framework may be prohibitive to wade through to untangle the connections between the variables in the formulae. Analysing the pollution diagram in *Figure 3.2*, there are two core concepts involved in the calculations: water and pollutant. To create placeholder objects, these and their attributes are positioned as in *Figure 3.7*.



```
Object(t) = Object(t - dt) + (event_or_activity_1 - event_or_activity_2) * dt
INIT Object = { Place initial value here... }
 INFLOWS:
   event_or_activity_1 = { Place right hand side of equation here... }
 OUTFLOWS:
   event_or_activity_2 = { Place right hand side of equation here... }
Stock(t) = Stock(t - dt) + (flow_1 - flow_2) * dt
INIT Stock = { Place initial value here... }
 INFLOWS:
   flow_1 = { Place right hand side of equation here... }
 OUTFLOWS:
   flow_2 = { Place right hand side of equation here... }
attribute_or_property = { Place right hand side of equation here... }
converter = { Place right hand side of equation here... }
```

Figure 3.6. Formulae underpinning the model in Figure 3.3.

Secondly, one determines the parameters that can be measured and the ones that need to be calculated, hence somehow accommodated in the placeholder objects. Measurements of the following parameters can be taken on site: the inflow and outflow of water in the pond and its volume (respectively W_{in} and W_{out} and W_{vol}), the amount of pollutant dumped by the production plant (P_{in}) and the concentration of the pollutant in the pond (C_{pol}). This leaves two variables to be calculated: the amount of pollutant in the pond (P_{amount}) and the amount leaving the pond (P_{out}); accumulation of pollutant in the pond (P_{acc}) can be derived from the placeholder object model, but is not included in the STELLA model.

Is the concentration of the pollutant in the pond an attribute of the water in the pond or of the pollutant itself? The taxonomy in *Figure 3.4* states that "a molecule has a concentration", but it can only have a concentration dissolved or suspended *in something* and not *of itself* (of itself are properties like melting temperature and structure of a molecule). However, to conclude it is an attribute (i.e. `hasA`) of the water in the pond is premature: if modelled as such, the model will not be able to accommodate for other pollutants unless one labels them as C_{pol1}, C_{pol2} and so forth. Hence, there can be multiple instances of pollutant concentrations, even of the *same* pollutant if there are multiple strata in the pond, therefore entitling it to have its own placeholder object. This is included in *Figure 3.7*, thus now containing all required parameters. As example of the model's usability of representing the semantics of the formulae, the amount of pollutant leaving the pond P_{out} is included in *Figure 3.8*; the two remaining calculations are in *Appendix B*, which make use the *exact same configuration* of the placeholder objects. In extreme, it is possible to represent any calculation deriving the value of one variable from others and to create new variables that may be of interest, such as the accumulation of pollutant.

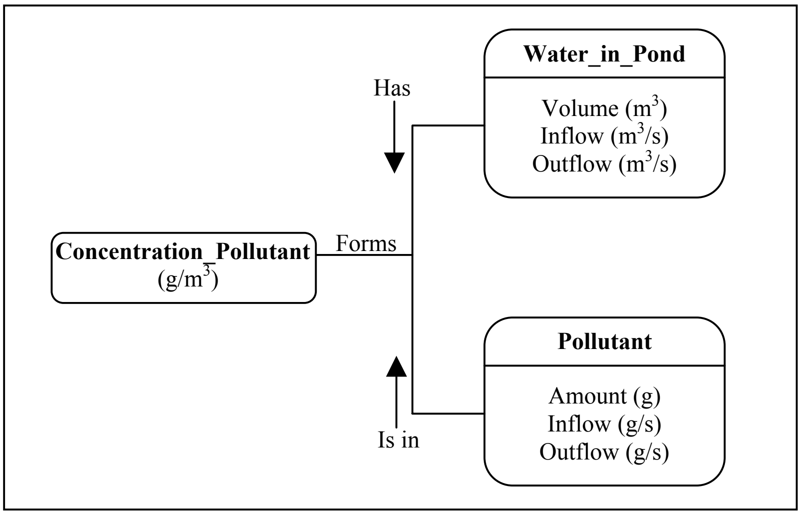


Figure 3.7. Placeholder objects model.

Considering the taxonomy of *Figure 3.4* (or *3.5*) and the model of the placeholder objects in *Figure 3.7*, the concept of flows and the sense of temporality of the ecological model is eliminated, yet represented in a for the computing scientist utilisable manner because *Water_in_Pond*, *Pollutant* and *Concentration_Pollutant* have a direct correspondence to the concepts `Water`, `Pollutant` and `Concentration` in the taxonomy. In this view, the placeholder objects with their formulae can be interpreted as an extra 'layer' on top of the taxonomy.

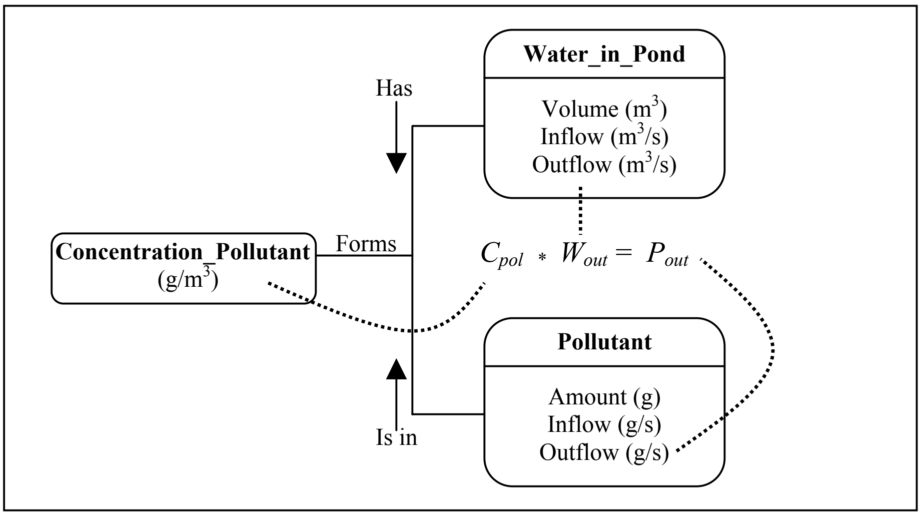


Figure 3.8. Placeholder objects and the formula to calculate the amount of pollutant leaving the pond.

3.4 Discussion

Even though the type of ecological model utilised in this pilot experiment was different from the object-focussed fungus-ant symbiosis in *Example 2*, similar issues surfaced when attempting to remodel it into an (informal) ontology: the abstraction introduced several extra concepts, such as `Activity` and `Molecule`, and the associative knowledge of the subject matter is seemingly 'lost' and replaced with a more rigid, structural organisation of the concepts. However, there are two factors ameliorating this potential problem:

i. Ecology already divides concepts into three categories: natural, functional and integrative concepts. The first two types of ecological concepts can be identified in the taxonomies of both this pollution experiment and *Example 2*: the functional concepts are categorised under `Activity` and the natural concepts on the left-hand side of the respective figures, as e.g. under `Organism`. Imposing a separation and categorisation may actually benefit ecology. Ford (2000) presents the interdependencies between the types of ecological concepts, and, for example, indicates that

   > [n]ew functional concepts arise to describe newly understood structures or interactions in natural concepts and research into functional concepts is constantly used to refine the definition of existing natural concepts and their classifications.

   and "[d]evelopments in measurement lead to refinements of functional concepts". Hence, by trying to define the concepts more clearly with the aid of ontology development, the discipline of ecology may advance at a faster pace. However, realise that it is considered that the *change in definition of concepts* and *how they may be classified* is the very essence of scientific advance (Ford, 2000), which has the consequence that *any* development (software) of an ecological ontology *must* facilitate for extensive features for ontology evolution. A full treatise on the actual and potential impact of this and more detailed knowledge on the differences in the scientific enterprise will be discussed at a later date; practical concerns on managing changes in ontologies is addressed in chapter 5 and the interested reader may to refer to Klein and Noy (2003).

   In addition to this requirement, there still exist the challenge of representing the integrative concepts that make ecological models, which are sometimes established and captured in axioms, but also still be conjectures or in the process of being refined, where the second and third stage include alternative views of some ecological domain. This indicates that the chosen ontology development process *must* be capable of representing alternative views of a domain. These uncertainties, assumptions and occasional sub-optimal definitions of ecological concepts make creation of an ecological ontology a daunting task. 'Normal' ontology development entertains itself with *how* to represent what is [known], whereas an ontologist for ecology will have to cooperate in the process that otherwise logically occurs before ontology creation, i.e. the *what* and *why*. This should be modelled not only in a for a computing science usable way (the ontology), but also usable for an ecologist, who will be pushing the boundaries of his/her discipline by clarifying relevant concepts, thereby may be able to formulate research questions, and consequently their theories, better. Therefore, the seeming loss of associative knowledge when translating an ecological model into an ontology, as carried out in this pilot experiment, is not an actual loss and more likely to be an advantage because, provided alternative views of combinations of concept (into integrative concepts and theories) can be accommodated for, it will aid the advance in ecological science.

ii. Some coherence might be regained by remodelling the pollution model into an ontology base and commitment layer, where the latter is of prime importance because of its expressiveness and closeness to a (more flexible) conceptual model. Downside of such an approach, and for this reason not pursued in this pilot experiment, is that in order to create the commitment layer, more detail about the rules of the concepts and relationships is required, which is not available to the desired extent at the time of writing. If one were to engage in this regardless, being it with DOGMA or maybe Conceptual Graphs, some assumptions will need to be made. Acknowledged, the pollution model *Amalgamated Industries* already contains two assumptions: P_{in} equals $P_{dumped\_by\_plant}$, even though adsorption and absorption to particulates in the river and subsequent sedimentation is possible[36], and a uniform concentration of pollutant throughout the pond, although variations in concentration is incorporated in the LEEDS model and facilitated for in the placeholder objects model, hence not impossible.

It must be noted however, that inclusion of `Concentration` in the taxonomy is not to this author's satisfaction. It was interesting to discover the near-correctness of `Concentration` in the taxonomy through the formulae and placeholder objects exercise. Apart from the analysis in §3.3 on how to represent this in the placeholder objects model, one can take a closer look at the ontological connections to consider if a change to the taxonomy as in *Figure 3.4* is warranted. It can be considered a political decision to view `Concentration` of a pollutant as a `partOf Water` (instead of `Molecule hasA Concentration`), because the release of pollutant – hence its resulting concentration in the environment, including the water – is ontologically related to human activity and not a 'natural' case of *what is*. Secondly, the *molecule* (pollutant) is *in* the water, not the concentration of the molecule. Third, `Water hasA Concentration` faces a similar problem as the `partOf` and is incomplete: in the more expressive languages used for constructing ontologies, one can contemplate a relationship between `Water` and `Molecule` called `forms` that is also related to `Concentration` (or a ternary relationship that reads `Molecule hasA Concentration in Water`). In this instance, analysing the variables of the equations not only can, but actually does, benefit development of the lightweight ontology. Further, it serves as a crosscheck that placeholder objects required for the calculations are indeed represented in some manner in the taxonomy to avoid (or at least minimise) loss of knowledge. Even though some of the semantics of the ecological model might seem 'lost', this is not the case; on the contrary, the placeholder objects methodology provides a richer semantic representation of the knowledge that was 'hidden' in the original ecological model.

However, before getting too excited, the taxonomy was built manually and the identification of direct correlations was achieved with implicit knowledge and informed assumptions. That the similarities between e.g. stock and object/type are deduced by exploiting the author's knowledge of both disciplines *does not imply* these correspondences will *always* be applicable. It requires further investigation on this matter to determine its validity, but it is encouraging that these correspondences could be identified in some larger STELLA models inspected, such as the marine microbial loop (ML) and its sub-models (Tett and Wilson, 2000) and the Vollenweider models[37] (the latter forms the basis for eutrophication control). Secondly, it is the expectation that accommodating the action connectors of the ecological model will be most challenging and might not have a one-to-one relation with one of the elements of an ontology. The use and meaning of the action connectors found its way into the placeholder objects model, in turn bearing a relation with the suggested taxonomy, hence may aid in determining a formal representation of the 'translation' from ecological to computing model.

The pilot experiment exceeded the author's expectation concerning the prospect of having to deal extensively with uncertainties in relationships between concepts and the accommodation of assumptions. The choice of using the abstracted pollution prototype for this experiment was relatively random (based on interest in the subject matter) and not handpicked to ease the exercise. The relative absence of such difficulties may be due to the size of the pollution prototype, because accommodating all elements of the *Amalgamated Industries* demo demands extending the taxonomy in the direction as

[36] Else, one can calculate P_{in} as opposed to measuring it, or consider the river as a separate system that can be placed 'before' the pollution-in-the-pond example, where P_{out_river} then equals P_{in} into the pond, which would involve including concepts such as ad/absorption, sediment accumulation and consumption by phytoplankton.

[37] http://tejo.dcea.fct.unl.pt/resources.asp, made by the IMAR - Centro de Modelação Ecológica.

indicated in *Figure 3.5*. Secondly, the present taxonomy ideally will include more concepts that subsume the existing structure, or are subsumed by existing concepts, in order to increase its potential for reusability, but conversely will increase the likelihood of having to resolve ambiguities and assumptions. However, an 'avoidance' approach might be feasible, or taken for practical reasons. For example, both the ML and SeaWeed[38] model are composed of smaller sub-models; the former contains Riley+, MicroPlankton and Autotroph-Heterotroph, the latter Vollenweider and a tide & light simulation. One can create a 'mini-ontology' for each small ecological model separately and develop a library where the user can choose the desired sections to create larger models. Caveat may be the prospect of integrating such ontologies, especially because there exist e.g. Vollenweider models of *increasing complexity*. An analysis of the differences between such 'simple' and 'complex' versions revealed that the more complex models of the same topic contain both *additional sections* with influencing elements added to the model as well as *filling* the existing structure with more details. Tett and Wilson (2000) indicate that this may be the case with multiple models, because there is a desire to keep the amount of Stock elements (state variables / instances of concepts) to a minimum for reasons of computational power and practical as well as theoretical challenges of estimating parameters. Smith (1974) added that a good simulations should include as much detail as possible, a good model should include as little as possible. These perceptions and actual knowledge change over time, having not [yet] achieved consensus, and have the potential unstable effects of cascading such uncertainties in larger simulation models, which are, according to Nihoul (1998), neither possible nor desirable to include in one model. A design decision on larger ontology versus multiple mini-ontologies will need to be made.

3.5 Conclusions

The pilot experiment revealed that with ecological modelling software such as STELLA, guided bottom-up development of ontologies might be within reach by formalising the identified correspondences. The methodology of placeholder objects proved to be a useful approach to include semantics of ecological formulae in a manner useful for computing science. The relative absence of known challenges of ecological data such as uncertainties and assumptions may be due to the size of the pollution prototype. Improvements in formulating research questions and hypotheses to scale-up this experiment were made and are included in the PhD research proposal.

[38] http://tejo.dcea.fct.unl.pt/resources.asp, made by the IMAR - Centro de Modelação Ecológica.

4. Ontology integration

Bearing in mind the factors of data heterogeneity, potential mismatches, the composition and types of ontologies, this chapter looks into the myriad of ways to combine ontologies, clarifies some of the related terminology and analyses aspects such as the clouding between semantic and structural integration and differences and expectations when integrating ontologies with the same, a similar, and complementary subject domains.

4.1 What is integration?

In the ontology-related research literature, the concept of 'integration' means anything ranging from integration, merges, use, mapping, extending, approximation, unified views and more; sometimes interchanging the words as if all are synonyms, although these concepts are used as homonyms as well. *Appendix C* contains a summary of this paragraph in table format and *Figure 4.6* in §4.1.3 contains a graphical representation of the level of integration of a concept relative to the other concepts of integration, whereas this and the next paragraph provide more insight in these different notions of ontology integration and include some examples to illustrate the meanings.

4.1.1 Overlapping and contrasting concepts of ontology integration

Pinto *et al.* (1999) conceptualised integration by untangling the various activities generally categorised as integration into clearer distinctions of *integration*, *merge* and *use*.[39] Real integration applies "when building a new ontology reusing other available ontologies" whereby the integrated concepts can be used like sub-modules, adapted, specialised or augmented by new concepts and assumes the ontologies involved are each of a different domain, thus limiting the scope of their interpretation. Pinto's merging refers to combining different ontologies with the same subject domain and creating a unified ontology, though noting that the process of merging is "very unclear", which has not changed much since then (see also §4.2). The third category involves the use of ontologies to build software applications, further examined in §4.1.2.

In contrast, Sowa (1997) distinguishes between *different levels* of integration: alignment, partial compatibility and unification (in order of increased integration). His *unification* is synonymous with Pinto's merging. The *alignment* means a mapping of concepts and relations between multiple ontologies based on preservation of the partial ordering and synonyms, as well as the possible introduction of *new* concepts that will function as sub- or supertypes. This is in contrast with Mena *et al.* (1996), who use existing concepts in the ontology by traversing the tree for hyponyms and hypernyms to 'link' concepts between the ontologies (depicted in *Figure 4.1*).

This aspect of allowing hypo- and hypernyms, plus using sound and complete mappings, is by Akahani *et al.* (2002) referred to as "approximate ontology translation", as a 'best fit' for a mapping exercise (*Figure 4.2*). Their translation can be considered as an extra ontology that is positioned between the two ontologies that are being approximated, which does not lend itself well for scaling up the ontology integration where $n>2$.

[39] Refer to *Appendix D* for diagrams on integration, merging and using ontologies according to Pinto's definitions.

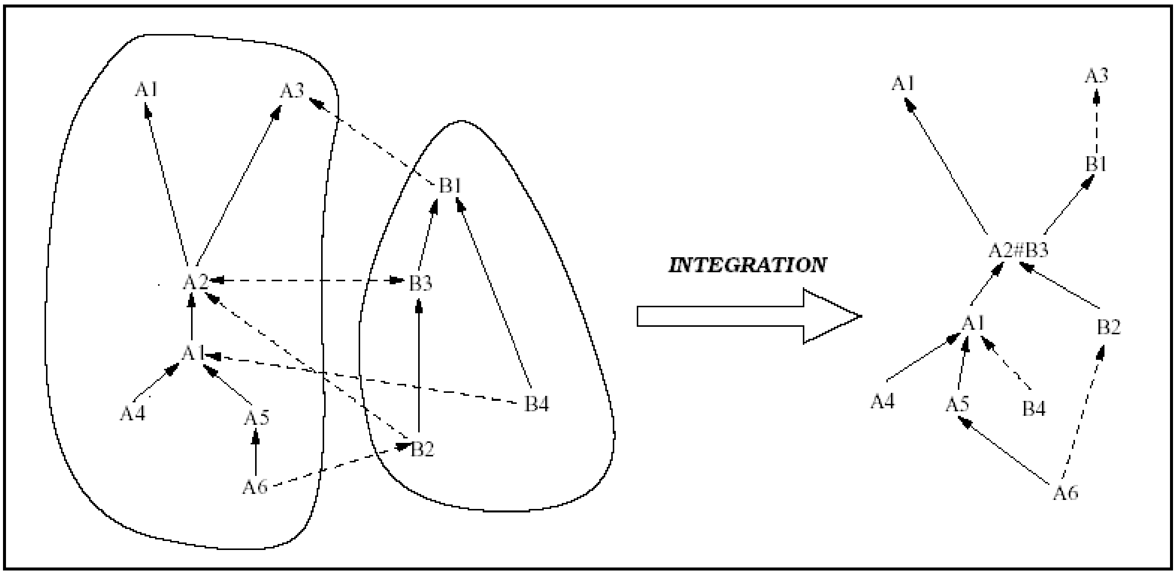


Figure 4.1. Integrating two ontologies. (Source: Mena *et al.*, 1996)

For example, if there were a third ontology to be integrated, will they be in sequence, as in $O_1 - O_{mapA} - O_2 - O_{mapB} - O_3$, and transitive, or would it require another translation ontology between O_1 and O_3? For example if one were to add a third ontology to *Figure 4.2a* like the hypothetical ontology shown in *Figure 4.2c*, there are two methods to map this:

1. `WineMenu` <translation ontology A> `WineList` <translation ontology B> `DrinkMenu`. Should one be capable of querying the `DrinkMenu` and have a `Merlot` returned, i.e. the connections are transitive?
2. `WineMenu` <translation ontology A> `WineList` <translation ontology B> `DrinkMenu` <translation ontology C> `WineMenu`, hence a triangular relation, with multiple ontologies resulting in a mesh structure.

Madhavan *et al.* (2002) consider a 'helper model' similar to the O_{mapA} of Akahani, in order to accommodate additional requirements to achieve the mapping between two ontologies. Further, use of hypo or hypernyms / approximation will result in a loss of information that may or may not outweigh the benefit of integration, depending on the specific task and UoD.

The author's referral of alignment to *a mapping* needs clarification. Mapping can involve an extending (Marjomaa, 2002), as in 'plugging in', of a second ontology into the main ontology (see example in *Figure 4.3*). This is sometimes called backbone ontology with a mapped refinement (Guarino and Welty, 2000), a local ontology integrated into a global or reference ontology[40], or the mapping can connect two entirely different ontologies. Gangemi *et al.* (2002a) use the former technique, though called *incremental loading*, to construct the fisheries ontology; another would be, say, an ontology of tuberous plants and one can later map an ontology of *Ipomoea batatas* (sweet potato) into the main ontology as a process of incremental ontology design. The latter, mapping of entirely disjoint ontologies, was carried out during the ONTOGENERATION project (Aguado *et al.*, 1998), which combines the CHEMICALS ontology with the linguistic ontology GUM (among others) and by Goertz *et al.* (2003) who did the same with EMBASSI (multi-media equipment ontology). In line with Sowa's levels of integration, these examples can be considered to be weaker than his definition of alignment, primarily because these mappings have much less aligned concepts and relations than in the alignment of ontologies.

[40] Depending on the context of an article, this can refer to quite different aspects of integration as well. See also next section.

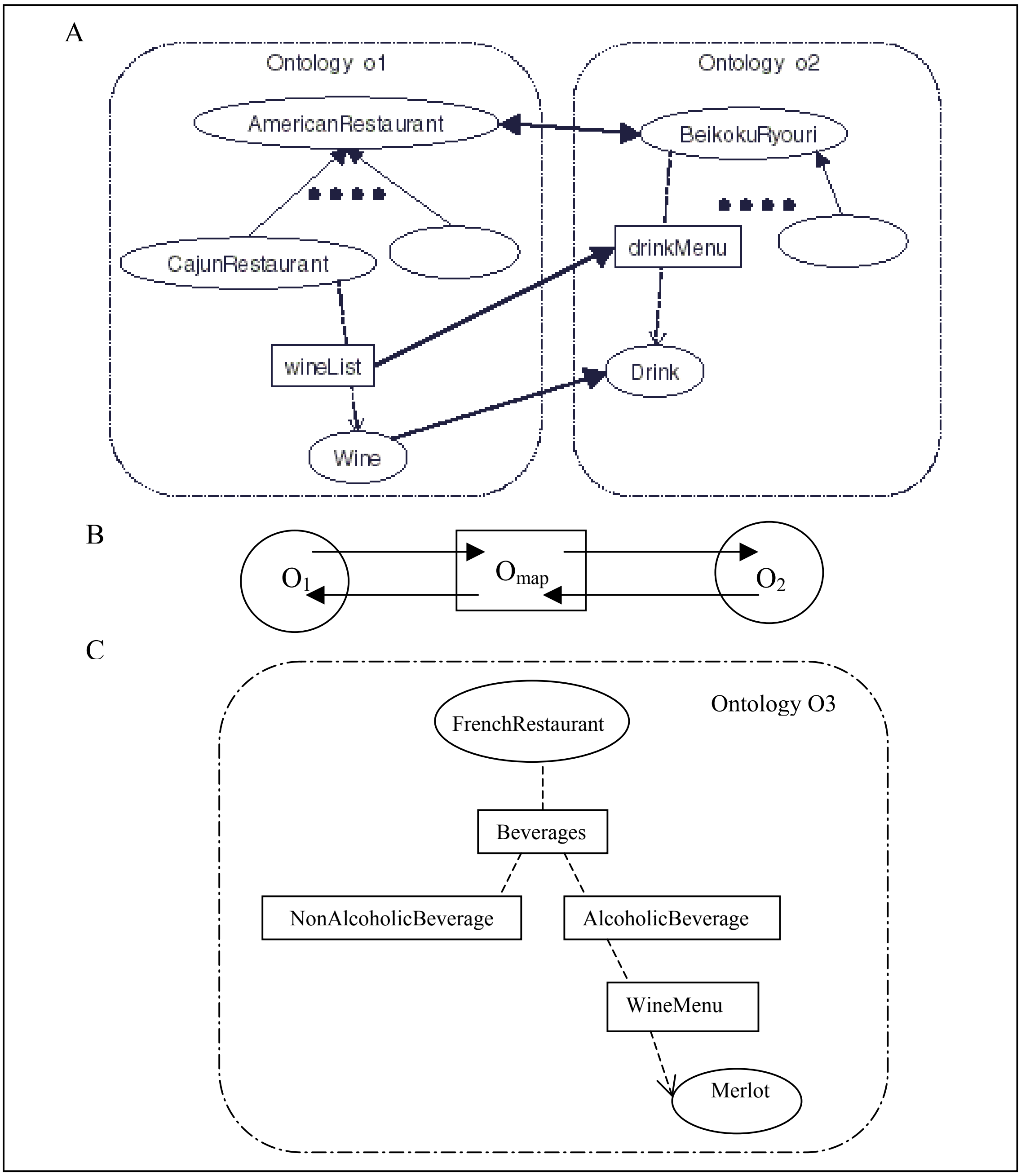


Figure 4.2. 2a: Ideal mapping (Source: Akahani *et al.*, 2002), *2b: 'approximate ontology translation' and 2c: a hypothetical third ontology.*

Sowa's (1997) *partial compatibility* refers to the capacity that "[A]ny inference or computation that can be expressed in one ontology using only the aligned concepts and relations can be translated to an equivalent inference or computation in the other ontology". However, one can interpret this as one of a more practical matter (chapter 5), or as a structural integration, but does not necessarily deal specifically with semantic integration.

Hefflin and Hendler (2000) divide integration methods into *mapping ontology*, where an ontology O_M contains the rules that map concepts between ontologies O_1 and O_2, *mapping revisions*, where O_1 contains rules that map O_2 objects to O_1 terminology and vice versa, and an *intersection ontology*, where ontology O_N is created containing the intersection of concepts between O_1 and O_2 and they rename terms where necessary; see *Figure 4.4*.

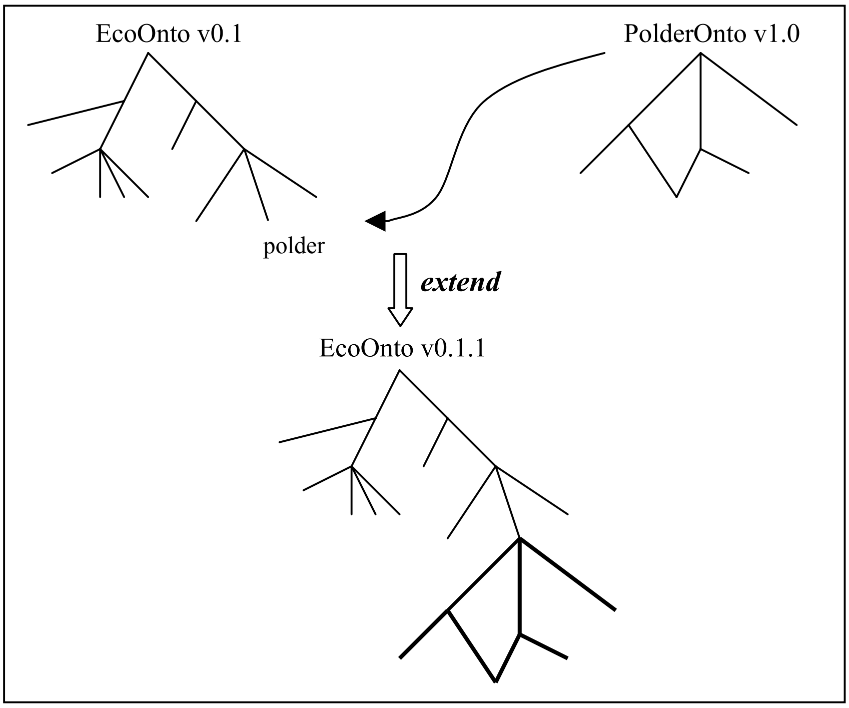
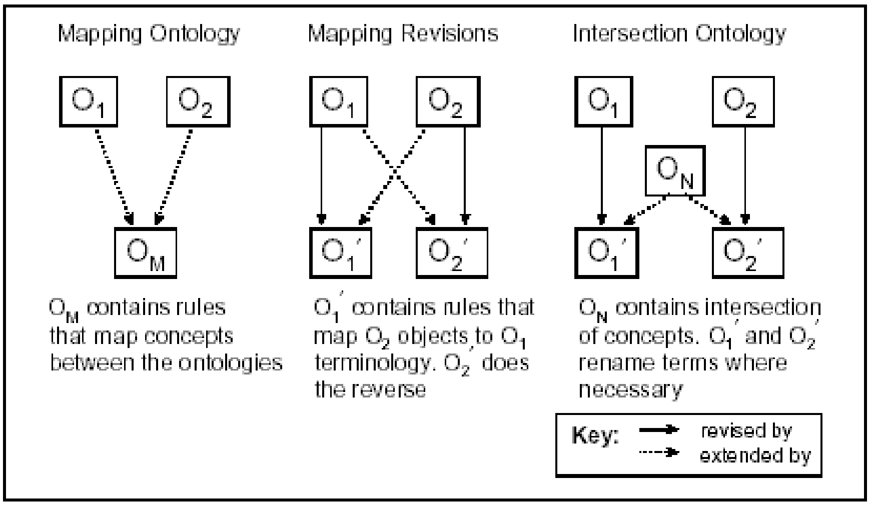


Figure 4.3. The author's interpretation of Marjomaa's extending of an ontology.



O_M contains rules that map concepts between the ontologies

O_1' contains rules that map O_2 objects to O_1 terminology. O_2' does the reverse

O_N contains intersection of concepts. O_1' and O_2' rename terms where necessary

Figure 4.4. Integration methods according to Hefflin and Hendler (2000).

4.1.2 Non-disruptive integration and (re)use of ontologies

All above-mentioned integration, unification, mapping, alignment, approximation and partial compatibility are more or less intrusive in that *when combined* one way or another *they form a new ontology emerging from or added to the original ontologies.* However, Calvanese *et al.* (2001a) consider mapping between one global and several local ontologies leaving the local ontologies intact by querying the local ontology/ies and converting the query result into a concept in the global ontology (or vice versa – outline included in *Appendix E*). Kalfoglou and Schorlemmer (2002) exploit a similar idea with their IF-Map, using an empty reference ontology with local ontologies populated with instances, but place the results, obtained via logic infomorphisms instead of database queries, in a new global ontology to be created on the fly while not disrupting the local nor the reference ontologies (*Appendix F*)[41]. Wache *et al.* (2001) divide these non-disruptive ontology integrations into three categories: single, multiple and hybrid approaches (*Figure 4.5*). The *single ontology* uses a global ontology with shared semantics. With *multiple ontologies* there must be [non-disruptive] inter-ontology mappings, but there does not exist a global ontology, although, like many others, Wache *et al.* (2001) do not, or cannot, specify how this is to be accomplished, which led them to propose the hybrid approach. The hybrid approach does use one shared global vocabulary, but unlike the single ontology approach, contains only the basic terms of a subject domain like Guarino and Welty's (2000) backbone ontology.

One could argue if these non-disruptive versions of ontology integration is 'real' integration, because the main aspect is creating an extra ontology what could be interpreted as 'ontology middleware' of some sort instead of actually combining (mapping, extending, unifying) existing ontologies into a new one, where only the newly created ontology is to be used. On the other hand, maintaining the original ontologies with the loose coupling allows for more flexibility and may be easier to reuse than a large monolithic ontology that has been developed by incrementally mapping new ontologies into the main ontology (as is the case with CYC (Reed and Lenat, 2002)).

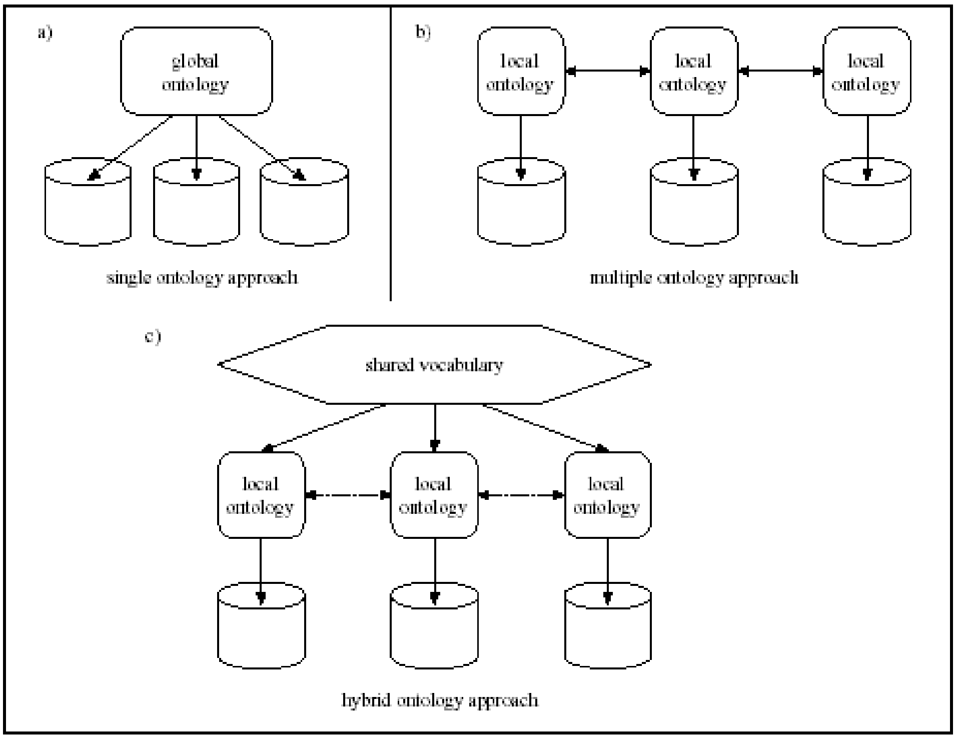


Figure 4.5. Non-disruptive integration methods. (Source: Wache *et al.*, 2001)

The *use* of more than one ontology in an application setting could be considered as an even 'lower' level of integration, because this type of integration does not intend to modify the affected ontologies in any way, merely using (some of) the concepts as is. However, this also depends on one's point of view: using certain concepts across ontologies in some conceptual model as predecessor for computational models to create an application, based on object-oriented software or some (relational) database, may actually induce bottom-up creation of a new ontology for the particular subject domain[42].

[41] Observe that this approach is slightly different from the 'original' description by Kent (2000), who summarises that "Sharing interoperability occurs through generic ontologies viewed as theories (types and constraints) with no instance collections".

[42] See for example Pazzaglia and Embury (1998), but note that they use the concepts of 'translation' and 'mapping' differently due to the minimal shift of emphasis of their research, not elaborated on further here.

This is unlike software programs to query ontologies, but also depend on the implementation: does this mean querying ontologies from a unified global ontology, or leaving the concepts and their labels unchanged and provide an alias lookup if necessary? Since Pinto *et al.*'s survey, there have been only limited developments in this area of using ontologies in/for applications; examples for ontology use are On-To-Knowledge[43] and Haystack[44] and as starting point for database design, the Ontology Management Portal (Sugumaran and Storey, 2002). Furthermore, it is difficult to predict what software can and may be developed and how this would affect ontology integration (Pinto *et al.*, 1999). For example, Stumme and Mädche (2001a) provide a 'grander' interpretation with relation to the Semantic Web and combine ontology use and merging[45], consisting of local (federated) ontologies with the purpose of merging them at some stage; these "federated ontologies" are analogous to federated databases (see also chapter 5 and Stumme and Mädche (2001b)). Another interpretation of reusing existing ontologies, in combination with formal integration, is the architecture of FAO's Fisheries ontology[46]. One can consider the DAI-DEPUR+ (Ceccaroni *et al.*, 2000) and its successor OntoWEDDS (Ceccaroni *et al.*, 2004) as really *using* and *re*using an ontology to enhance their software system. These systems combine two knowledge-based systems (KBS), one case-based reasoning and the other a rule based system, with WaWO, an ontology of a wastewater treatment plant focussed on the microbial aspects, to enhance reasoning and deduction of the overall software system[47]; WaWO itself reuses *part* of the UpperCyc ontology in addition to the wastewater treatment concepts. Ecolingua reused sections of Ontolingua and Cyc (Correa da Silva *et al.*, 2002).

4.1.3 Overview and comparative characteristics

Sowa (1997) considers the distinctive criterion to be *interoperability*: interoperability may be viewed as a computational property only, but it has to be based on a conceptual integration at the ontological level too. However, can one draw lines on the gradient of increasing interconnectedness of two or more ontologies? *Figure 4.6* provides a first attempt to graphically categorise the levels of integration, using definitions provided by the consulted references (for a summary of these, please refer to *Appendix C*). Likewise, *Table 4.1* is an (incomplete) list of factors perceived to be distinguishing characteristics between the various interpretations of integrating ontologies, functioning as a step towards clearer distinctions between the terms and processes involved in 'combining' ontologies.

It is a rather curious situation where ontologists managed to create a sub-discipline, ontology integration, which is riddled with ill-specified definitions that function as synonyms and homonyms, the very problem that the use of ontologies in information integration tries to resolve. This paragraph serves only as a summarised survey; with further (literature) research (refinements), other variations likely will emerge. At present, the approach to this problem seems to be that each author indicates what type of 'integration' s/he refers to before digressing into its details of the one particular type chosen. However, it would be more appropriate to define each concept to clarify these matters, which may well be carried out by devising an ontology of ontology integration.

[43] http://www.cs.vu.n0l/~ontoknow/index.shtml. The On-To-Knowledge Project resulted in a software toolkit for ontology development, maintenance and (re)use of ontologies; refer to Fensel *et al.* (2002) for a description of the software modules. The language, OIL, was integrated with DAML (http://www.daml.org), which formed the basis for OWL (http://www.w3.org/2001/sw/WebOnt/) for the Semantic Web.

[44] http://haystack.lcs.mit.edu/

[45] A diagram of their 5-layered architecture is included in *Appendix G*.

[46] http://www.fao.org/agris/aos/ and further discussed in e.g. Gangemi *et al.* (2002). A summarizing figure is included in *Appendix G -1* for convenience.

[47] The architecture of the complete software is included in *Appendix G -2*.

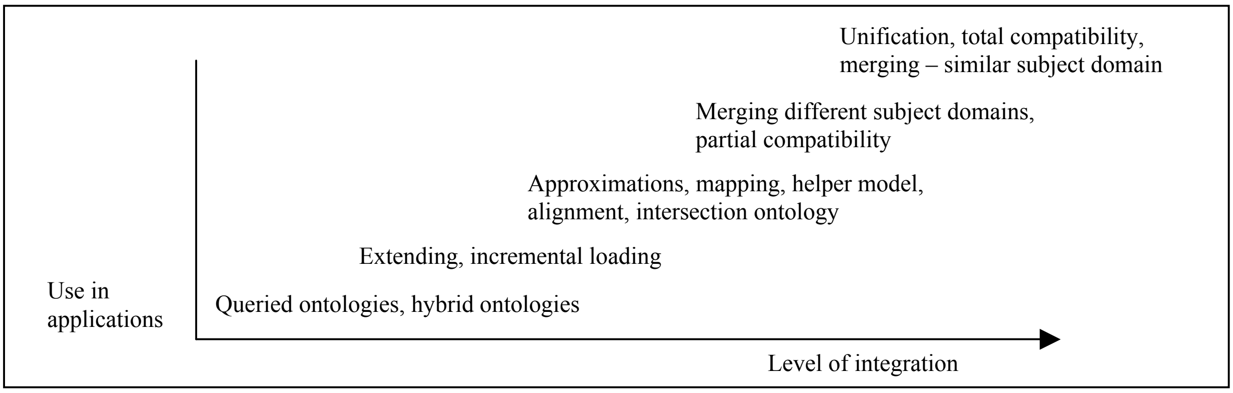

Unification, total compatibility, merging – similar subject domain
Merging different subject domains, partial compatibility
Approximations, mapping, helper model, alignment, intersection ontology
Extending, incremental loading
Queried ontologies, hybrid ontologies
Use in applications
Level of integration

Figure 4.6. Stylized graphical representation of the (perceived) level of integration.
Note: this is a first sketch, and subject to modification when the concepts are more clearly defined.

Table 4.1. List of factors and properties that contribute to distinguishing the multiple methods of 'integrating' ontologies. Refer to Table C-1 for the definitions of the concepts listed in the "Where used?" column.

Aspect	Where used?	Compared to what?
Automatically finding similarities, synonyms	Matching	Merging, which involves 'human intervention'
Combining similar or the same subject domains	Unification, total compatibility, merging	Partial alignment of less-well related ontologies
Combining different subject domains	Merging, mapping	Unification
Aligning two similar concepts	Approximation, mapping (e.g. w.r.t. hyper- and hyponyms)	Exact matches of concepts that are the same
Sections of the ontologies have an 'overlap' in their respective ontologies	Partial alignment, partial compatibility	Total compatibility, unification
Generate a new ontology that 'sits in-between' the two that are combined	Helper model, intersection ontology	Mapping and merging where the two ontologies have a 'direct link'
Adding new sections to an ontology like 'dressing up a Christmas tree'	Extending, incremental loading, hybrid ontology	
Adding new sections to an ontology like 'dressing up a Christmas tree', non-disruptive	Hybrid ontology	Versus 'disruptive' techniques extending and incremental loading
Leave the original ontologies as is and use them as such	Helper model, intersection ontology, queried ontologies, hybrid ontology, multiple ontology, use	Mapping, merging, unification, single ontology, i.e. compared to process that changes something of the original ontologies to create a new one
Obtaining concepts and their properties by querying the local ontologies	Queried ontologies	Generating a backbone ontology and using the local ontologies for details
A 'translation layer' between two ontologies	Helper model, intersection ontology	
No intention to change anything in either ontology	Use	Merge, unify, i.e. the content of at least ontology will change, whereas with using it does not
An aspect is dependent on other aspects originally (ontologically, 'by nature')	Generic integration	Coincidental integration, where a relationship is established/concept created as a result of 'something new', the Creative Design process
New inventions generate new relations between previously unrelated concepts, hence between different ontologies.	Coincidental integration	Generic integration
Some form of autonomy at the 'local' site	Federated ontologies	Ontology under centralized

4.2 Integrating ontologies

Having touched upon the myriad of possibilities to integrate ontologies, nothing has yet been mentioned about *what* is to be integrated and to what extent this has an effect on the choice of the type of integration that is most applicable to achieve a formulated goal. This can be divided into semantic, structure and syntax integration; system integration has no effect on the integration of ontologies, but on the implementation, and therefore omitted from the scope here.

Semantic integration focuses on the intended meaning of the concepts, i.e. if concept C_1 in ontology I is synonymous (or, if one settles for an approximation, a hypo- or hypernym) with concept C_2 in ontology II. Structural integration addresses the aspect that while the semantics is agreed to be identical, the organisation of the concepts (categorisation, schema) is not and needs to be aligned and integrated. Note though, that the distinction between semantics and structure is not as clear as it may seem, because the structure conveys a semantic interpretation of the conceptualisation (Goh, 1996). Methodologically, syntax integration comes after semantic and structural integration in methodology, as it covers the 'translation' between the formalisms from source to target ontology (as there is not much point in matching formalisms if the meaning of what is being integrated does not make sense), e.g. from Description Logic to KIF. However, these translations, such as the syntactic representation of *a* concept in two formal languages, can be researched independent of ontologies.

4.2.1 Semantics and structure

A major difference between the semantic versus the structural and syntactic integration in practice is that the former relies heavily on the input by the SMEs to extract knowledge to make assumptions and thought processes explicit in order to determine the meaning of the related concepts between the ontologies, especially when the ontologies need to be merged and are covering a closely related, or the same, subject matter. Merging disjoint or orthogonal ontologies provide fewer problems than merging ontologies of the same type and subject.

As Aguado *et al.* (1998) succinctly point out, SMEs are not experienced in formalizing their knowledge and may require intermediate steps in the ontology development process to bring this knowledge in a usable manner to the surface from the viewpoint of the ontologist. On the other hand, in structural integration of ontologies, shared concepts are known facts; setting aside *how* this may be the case. Having established synonyms and similarities, design decisions include the name of the concept: must one name be replaced by the name of the concept in the other ontology or use different labels appropriate for each domain? The latter approach is taken with the SHOE implementation (Heflin and Hendler, 2000), which allows a `def-rename` to be specified to allow the local ontology to keep its preferred vocabulary while at the same time being interoperable with the ontology it is mapped to. However, if one were to choose to map to a *similar* concept, one may need to either find a concept that subsumes the concept of the local ontology or introduce one or more new sub/supertypes to minimize information loss. Although loss can be determined with closeness metrics such as precision and recall (Akahani *et al.* (2002) and Mena *et al.* (2000)) and formalised extended DL for 'loosely-sound', 'loosely-complete' and 'loosely-exact' mappings (Calvanese *et al.*, 2001b), the loss is, or may be, still there. Further, with such changes made, the integrated ontologies result into a new ontology, or, as in Akahani's model, do the 'translation' axioms become a separate ontology that is positioned in-between the two ontologies? When, with which ontologies and type of integration, is one preferable over the other? It may be interesting to explore this avenue by identifying and investigating which factors and properties determine success or failure, or to investigate the more practical aspects of performance querying the ontologies, analogous to database performance metrics.

Overall, structural integration is not necessarily as straightforward as it might seem.

Example 5. Structure versus Semantic integration

A difference between semantics and structure is not easily identified because the structure may convey semantics. (Note that this suggests that in order to carry out a structural integration, there must be agreement on the semantics). This example, based on two hypothetical polder ontologies, demonstrates the problem of this ambiguity. *Figure 4.7* shows sections of two ontologies with a similar domain: one could envisage the `EcoOnto` defined as part of the SEEK programme covering ecological niches and the `PolderOnto` created some time ago by Dutch researchers, who kindly have translated the Dutch labels of the concepts for their English counterpart.

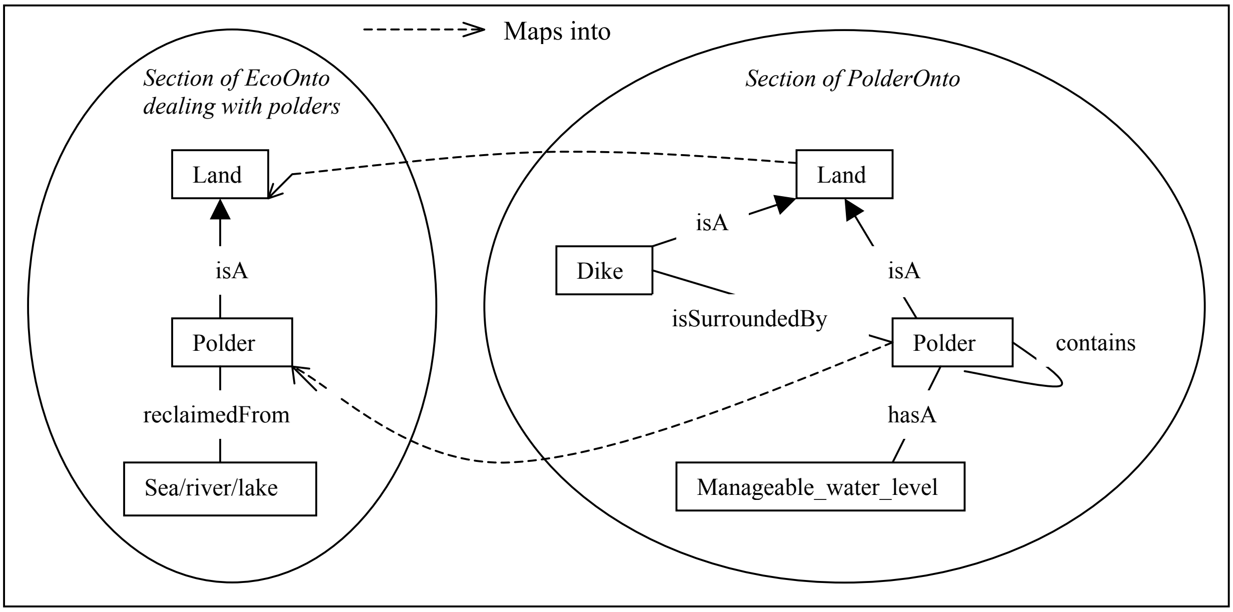


Figure 4.7. Semantic versus structural integration: polder example.

The differentiation between types of dikes is omitted from the example for the sake of clarity and to avoid a lengthy discussion on representations of compound nouns such as *zeedijk*[48]. At first sight, this looks like a straightforward structural integration: `land` maps to `land`, `polder` to `polder`, and the `isA` relationship is the same as well. The extra relationships and related concepts in the `PolderOnto` seem to provide a little more detail, in that one may extend ('load incrementally') the `EcoOnto` with the details from the `PolderOnto`. However, on closer inspection, it appears that the (English-language-based) `EcoOnto` concept of a polder,

> An area of low-lying land which has been reclaimed from the sea, a river or lake, especially in the Netherlands.[49]

is different from the Dutch concept. The Dutch-English translation of the definition of 'polder' is:

> 1) A piece of land surrounded by dikes, having a manageable water level.
>
> 2) Region with polders.[50]

[48] A *zeedijk* (sea-dike) is a dike on the seacoast or on the riverbanks where the river enters the sea, in order to prevent flooding.

[49] Definition from: http://www.allwords.com/query.php?SearchType=3&Keyword=polder.

[50] Original in the Van Dale [Dutch equivalent of the OED]: "1) door dijken omgeven stuk land, met beheersbare waterstand; 2) landstreek met polders." http://www.vandale.nl/opzoeken/woordenboek/?zoekwoord=polder.

In other words: *the semantics is different*! The Dutch interpretation is semantically not only richer than the English concept (e.g. the recursiveness of polder), but has as main criterion a ‘manageable water level’, which bears no relation to ‘reclaiming land’ in the English definition. The author considers the English-language version (i.e. as represented in the hypothetical `EcoOnto`) to be *incorrect*. There is nothing to be reclaimed: you never had the land for use to begin with. Further, people do not create a polder out of a river but divert the river instead (‘cut’ the river at two places, build a canal between the two points), in order to use the land previously containing the meanders. Last, because the manageable water level is the defining characteristic, a polder does not have to be on ‘low-lying land’: it can be anywhere, as long as the water level is manageable. Just because the Dutch made extensive use of the techniques since the 14th century, does not mean it *must* be situated on low-lying land. Therefore, what appeared to be a pure structural affair turned out to be of a semantic nature.

This example of combining different models of polders has shown that there may not necessarily be a clear distinction between semantic and structural integration. Secondly, it did highlight an additional difficulty when combining ontologies across linguistic barriers.

□

It might not suffice to ask ‘some’ SMEs to analyse and determine a correct method of integration, but one might need language experts as well, especially in the field of ecology or agriculture where concepts are localised and culturally embedded in society. This opens up another assortment of problems: does the (main) ontology need to be in one language, and one only – as is with the SEEK Project – or interoperable multilingual ontologies (e.g. the AOS Project)? Does the latter represent the same structure with other labels, or maybe captures different semantics ‘hidden’ behind translations? Can one determine, and if yes how, the loss of semantics when adhering to one ‘global’ or a ’backbone’ language, and what effects does this loss have on the comprehensiveness and chance of success of the SEEK Project? Do advantages of a single language counterbalance the loss? Is it reasonable to demand or expect that SMEs are not only experts in their discipline, but also fluent in at least one other language, and, stretching it further, be able to formalise their knowledge? After all, that is what one would require when determining if semantic integration is required, or if structural integration suffices.

There are no, most certainly less, potential difficulties with ‘hidden semantics’ when integrating more familiar domains, like the university ontologies of Doan *et al.* (2002) and Noy and Musen (2003): all have professors, departments, emails etc, though one might call a course a course and the other ontology refers to this as a module, or use profesor, departemento and correo electrónico instead. These semantics are the same, whereas the structure, categorisation of the concepts, may differ to reflect peculiar details of a particular university (e.g. the PhD-er as student or as employee).

However, and this might contribute to the relative ease of integration, subject domains such as universities, businesses, finance etc. are all constructions, inventions, of the human imagination, whereas the former example, ecology, or any area in biology, chemistry and so forth, is not per definition human created – more often than not, the available knowledge is in the research phase, contains hiatuses, and may not suit strict formalisms. In this context then, it is not important if the difficulty of formalising biological knowledge is due to the nature of the knowledge or the lack of training in formal thinking of life science researchers – fact is that it is not as straightforward as other subject domains (see also §2.1.2). Furthermore, it may be easier to identify the requirement for semantic and/or structural integration with, say, university ontologies, because the ontology engineer is cognizant of the subject matter.

4.2.2 (In)formal ontologies

Another discrepancy may exist between the to-be-integrated ontologies is a distinction between lightweight and heavyweight ontologies (and every gradation in between). According to Corcho *et al.*

(2003), taxonomies are considered full lightweight ontologies, e.g. the Yahoo!Directory. On the other hand, heavyweight ontology imposes more restrictions on modelling the domain in a "deeper way", adding axioms and constraints to lightweight ontologies. Gangemi *et al.* (1998) provide a finer grained categorisation, as discussed in §2.2.1 and included in *Figure 2.9* on the right-hand side of the arrow. The effects and difficulties of trying to integrate ontologies of a different formal level may be encountered with the Semantic Web. There is a desire for increasing relaxation of rigour of the construction of the Semantic Web in order to popularise it (Hendler (2002); van Harmelen (2002)), but is counterbalanced by the call to keep it formal (e.g. Rousset, 2002). Obviously, the latter facilitates integration, whereas the flexible and lightweight ontologies will be exponentially more difficult to integrate in any manner once the Semantic Web really catches on. However, one may argue that consistency is an impossible task to aim for due to the nature of the Internet anyway (Horrocks, 2002); but at the same time, this would defeat the purpose of ontologies – defining *consensual* knowledge about a subject to facilitate sharing and reuse of information. Integrating a lightweight into a heavyweight ontology poses several design decisions, depending on the type of integration. A comprehensive treatise is outside the scope of this report, but obvious aspects are: formalising a lightweight ontology to bring it on a par with a heavyweight ontology, as carried out for example with the ONIONS project (Gangemi *et al.*, 1998) and the Fisheries Ontology (Gangemi *et al.*, 2002a) or allowing lightweight 'side branches' in a heavyweight, formalised, backbone ontology[51].

4.2.3 Similar, complementary and orthogonal ontologies

Integrating orthogonal ontologies, for example combining the radiation section within the SEEK measurement ontology with e.g. a French grammar ontology and linking it to a English-French dictionary in order to retrieve query results in (near) natural French, is relatively not a difficult problem because of the distinct subject domains of the ontologies. One could even argue if this is 'real integration' or an interesting way of reusing existing ontologies.

Difficulties of integration arise with the same, similar and complementary ontologies, especially if the subject domain(s) is/are interdisciplinary.

* Same domain: fish taxonomy and FIGIS species[52], or two ontologies of plant taxonomy.
* Similar domains: plant taxonomy ontology and an animal taxonomy ontology.
* Complementary domains: Closely related, but in principle different, views on some domain, such as a forest from the perspectives of biodiversity and as production entity. *Figure 4.8* is an example of complementary ontologies for the aquaculture subject domain and its sub-domains ('topic spaces').

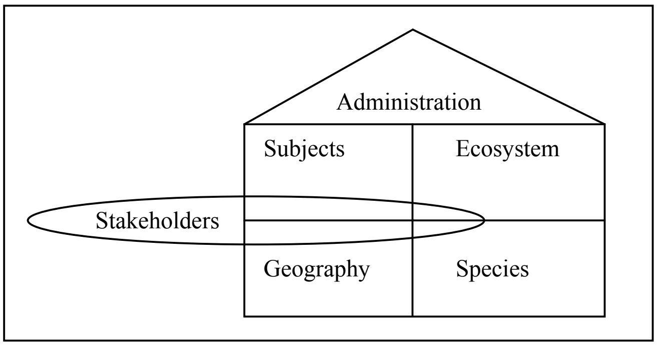


Figure 4.8. Topic spaces (complementary ontologies) for FAO's Fisheries.
(Reproduced from Gangemi *et al.*, 2002a. Refer to Pepper (2000) for more detail on topic maps)

[51] The latter is comparable to Sowa's (1997) "mixed ontology".
[52] FIGIS contains organisms used with aquaculture: http://www.fao.org/figis

It is possible to devise reasons why one type of subject domain combination is 'easier' to match, map, merge and so forth than the other two, but also argumentations why this may be more challenging to do:
Same domain:

- Pro: the ontologies likely will have many concepts in common already, which also would facilitate automated heuristics (see chapter 5);
- Con: if there are essential differences, there may be good (scientific) reasons to have these, i.e. there might be a need to facilitate competing subject domain views.

Similar domains:

- Pro: there will be a partial overlap of concepts, hence at least a mapping or 'partial compatibility' is within reach;
- Con: what on the surface may seem as identical concepts, might mean something entirely different in the other related discipline, which, if not addressed carefully could result in incorrect representations (from the perspective of one of the disciplines). Further, there would be the need to address resolution of synonyms and homonyms in some way, as it may not be realistic for a whole discipline to force them to change their vocabulary.

Complementary domains:

- Pro: one may extend the other and might be useful when building large ontologies in an incremental fashion. Potential intra-discipline disagreements (as mentioned in the 'same domain' section above) are [assumed to be] solved and there is less conflict of interest (e.g.: "let the engineers come up with their technical interpretation, because they know best about that, and we'll do the sociological aspects of forestry");
- Con: faces familiar problems of interdisciplinary work, especially because the knowledge has to be conceptualised. One has not only different vocabularies but also different perspectives and emphases on the subject(s). This could be a pretext for disaster: imagine an ontology of biochemical compounds categorised according to the chemical structure and integrating it with the usage in medicine: one most certainly would not want to map the concept containing instances like morphine and its structurally (on a molecular level) closely related antagonist naloxone into one group of 'painkillers'. Would a biochemist know the function of each instance in a group, or the medic the exact biochemical structure of each drug?

It would be simplistic to conclude that ease of integration of the same, similar or complementary ontologies depends entirely on the task at hand, i.e. what is the point of departure and what is the goal to achieve with the particular integration, but at the time of writing, a clear answer cannot be provided, because there is no overview of on successes and failures of ontology integration efforts, their subject domains, the type of integration and why the project was (un)successful.

4.2.4 Other aspects

Separately from semantic and structural integration of (in)formal ontologies of the same, similar, complementary or orthogonal subject domains, there are data characteristics specific to biology (see §2.1.2) increasing the level of difficulty in ontology integration. Whereas in §4.1 the suggestion of best matches, or approximations, of concepts was raised as a reasonable alternative because exact matches are frequently not available, this may not be suitable for certain disciplines within the biological domain. For example, Kalfoglou and Schorlemmer (2002) 'lose' information with their integration example on the differentiation between houses and cottages, which may be mildly annoying when house hunting, but neither disastrous nor a factual incorrectness on which a research hypothesis has to be built or verified (and possibly subsequent policies endorsed). Further, 'odd data' in biology could well hint to interesting and potentially useful exceptions to the rule, therefore minor data loss due to generalisation and the use of hypernyms should not be thought of lightly just because of the drive towards using automated integration. However, simultaneously categorisations within the biological sciences are not always as rigid as computing scientist would like it to be, which suggests that approximations in some instances of ontology integration still would fit within the boundaries of a less strictly defined concept. Although such a design decision may be regretted at a later stage when results of more research can more clearly identify some difference (e.g. the 'split' between Archae and Bacteria); having to reassess the concepts and/or its instances is boring, tedious and error prone. When an approximation is allowable and when it is not, depends on the human intervention of the SME.

In summary, there are multiple possibilities to combine ontologies, whose approaches can be distinguished by factors such as the level of integration, subject domain(s) and level of formalism of the ontologies.

5. Implementations of ontology integration

Whereas the idea and related efforts to integrate ontologies are not new, there remains limited and fragmented information on *how* this is achieved. However, there does exist some literature on *what* has been achieved. For example Chimaera (McGuinness *et al.*, 2000), a web-based browser ontology environment that accepts more than 15 designated input format choices (KIF, Protégé and so forth[53]) and some level of merging via a "taxonomic resolution mode", by looking for syntactic term relationships, "taxonomic analysis, and semantic checks", leaving the reader pondering about how the authors actually achieve this. Their focus is more on syntactic integration (matching, translation) than structure or semantic integration. OntoMorph is another example (Chalupsky, 2000)[54]. However, Reed and Lenat (2002) do mention they deemed it necessary to develop a variant of predicate calculus, CycL, to represent the assertions for "mapping/merging/integrating ontologies", albeit without providing exact details. Strikingly, the researchers who do seem to have integration software running and scaled up, do not provide implementation details, whereas details on formalisms, description logic etc. in order to carry out an integration of some sort are provided only in recently published articles. To these belong Calvanese *et al.* (2001a) with their query response integration, Kalfoglou and Schorlemmer (2002) with IF-Map and Akahani *et al.* (2002) with their approximation translation. Primarily, integration efforts are theoretical exercises with a mini-ontology example in a 'common sense' subject domain like universities, housing or travel arrangements, or laborious and time-consuming manual efforts to integrate large ontologies.

The remainder of the chapter looks into the available (semi-)automated mappings of ontologies and their differences in methodologies and ends with the integration software applicability and usability.

5.1 Semi-automatic ontology matching and integration

With the mushrooming of ontologies, it is highly desirable to have some sort of *automatic integration* of ontologies, not merely on the syntactic level, but also on the structure and semantic level. Few efforts have been published: at the 'low end' of automation are SKAT's rule entry by experts (Mitra *et al.*, 1999), then the automatically generated questions to be answered by SMEs in ONTOGENERATION (Aguado *et al.*, 1998) and the (albeit failed) integration of OpenDirectory into Cyc via a workflow application that ought to have guided the knowledge workers through the process of matching terms between Cyc and OpenDirectory. Wiederhold (1994) considers these SME-generated specific instructions to merge ontologies, to be encoded as "matching rules"[55], to form a new, "second layer abstract ontology", which is a different emphasis from Aguado and Cyc. On the other hand, Bernstein *et al.* (2000) use matching as "educated guess made by the system" to aid the engineer in the decision making process (and 'merging' when the contents of one ontology is moved into a source ontology). Hence, in this interpretation, the automatic part of the ontology integration is called matching, and *after human intervention,* one can achieve a merging of ontologies. To obtain more insight in the effectiveness of such an approach, it may be advantageous to categorise the type of matches that are suggested by the software and accepted by the

[53] This may suggest that automatic translation between syntactic representations is always carried out as intended, which is not exactly true. Consult e.g. Correa da Silva *et al.* (2002) for a syntactic translation test and the resulting difficulties they observed with the Ecolingua ontology.

[54] As an aside, Chalupsky refers to 'syntactic mapping' as "translation".

[55] The matching rules are the binary operations intersection, union and difference.

user. Weinstein and Birmingham (1999) identified three kinds of matches, inherited, specialised and serendipitous, analysed here:

- *Inherited* from shared concepts of a previously agreed upon global ontology, O_g, with a diverged local ontology, O_l, referred to by Weinstein and Birmingham (1999) as "request" and "recommendation" ontology, where the shared concepts subsume the matching request *and* the recommendation. This, of course, results in highest matching compatibility. However, it is arguable if this really is 'matching' because the two ontologies involved were one and the same before they were separated into O_l/O_g, presumably each with its own autonomy, and the divergent development of the two ontologies is either:
    a. A surfacing of the latent disagreement or absence of consensus when the original O_g was established. If this is the case, then matching – manually or (semi-)automatic, with or without guiding questions – may not be successful.
    b. The people responsible for O_l did change and model the ontology as part of a compartmentalisation of the subject domain or 'filling' the details of a more generic ontology O_g; such an 'inherited matching' actually is an update of O_g with the new concepts of O_l and may be considered as ontology development or maintenance instead.
- One can deduce from the next two matching groups, that Weinstein and Birmingham must mean the first option: an exact match implies that that particular section of O_l and O_g is unaltered over time and records a 'not exact' match when a change did occur over time. This author considers this type of matching part of ontology development and maintenance, or ontology evolution[56], which is different from matching two ontologies hitherto separated and who never have shared a 'common ancestor'.
- *Specialised* matches, were a local ontology has added more detail to some branch in the hierarchy, such as the 'incremental loading' in *Figure 4.3* or the local/global ontologies as in *Figure 4.5* and *at least one* of the elements has been specialised in the ontology. This covers point b above.
- *Serendipitous* matches: match by 'chance' and not inherited from any shared concepts between he two ontologies, for example the local ontology was updated with extra concepts, which may have been inherited from O_g, but not necessarily so. Essentially, each community O_l and O_g has (re)invented the wheel independently and at least partially have the same view on (a part of) the subject domain.

Hence, one can distinguish different contexts for (semi-)automatic matching, which will bear a direct relation to the actual process of matching, more or less useful strategies, and the output/results of a matching operation:

I. Ontology development, maintenance, revision, evolution. Divergent ontologies share a common ancestor with agreed-upon concepts and relations (and axioms, if applicable), thus in matching exercises, the *difference* between the ontologies are of interest and it is reasonable to assume that the majority of ontology elements do match and have the same structure and semantics as inherited from the original ontology. From a software functionality point of view, automated heuristics such as string comparisons with a thesaurus or 'near matches' of the labels of the elements are not its primary requirement because the unmodified inherited elements are the same, but tools such as logging the changes that are made or a facility to perform a structural diff are useful and/or essential. If the software is focussed on finding matching elements, it ought to ignore, or move to the background, these matches and bring to the fore the differences, i.e. the 'negative' of the matching result to highlight the changes between the ontology versions. The target users of this kind of software are likely ontology developers with a computing background.

[56] Refer to Klein and Noy (2003) for a framework on ontology evolution, covering aspects such as transformations sets, change logs, ontology of change operations, structural diff and their ontology of change operations (there are more than 80 basic operations).

II. Automated guidance to combine two ontologies that were developed independently. In this scenario, one desires to find *commonalities* (intersection[s]) among the concepts and relationships of the two ontologies. Subsequently, on has to establish with the domain expert if the match is indeed correct and have the same meaning, whereas with a mach as under I, one may assume such a match is valid. This higher dependency on the input of domain experts likely affects the procedure of carrying out matching: e.g. the phraseology of questioning a comparison of the two elements of the two ontologies, the GUI, the possible need for an intermediate representation model of the knowledge that is understandable to the domain expert (e.g. one will not achieve much by subjecting biologists to description logics), and usability in general.

The software and ontology combining approaches mentioned in this chapter are primarily focussed on achieving matching type II, not I; ontology development and maintenance is a large enough research sub-discipline that it would require a separate thesis to be able to address it appropriately and comprehensively.

Continuing software engineering efforts in ontology integration, Stumme and Mädche's (2001b) approach of federated ontologies have mainly automated the steps *before* the actual merging, i.e. as a 'pre-processing' stage to facilitate the actual merging of the ontologies. They use documents such as web pages (Semantic Web) to find instances via a linguistic analysis for the two (or more) ontologies that are to be merged. The output of this first step, two (or more) formal contexts, is subsequently merged using FCA-MERGE and pruned to remove too specific formal concepts, in order to provide concept lattices to be taken as starting point for deciding how to create concepts or relations with them. Even though the last step is largely manual, examples of instances (in this case actual instances), can help domain experts to understand concepts better. What this author considers a major advantage of the pre-processing and FCA-MERGE procedures is that it is easy to scale up to merging of multiple (>2) ontologies without having to change the functionality of the procedure, not merely in theory but already built in into the processes. Other software approaches indicate that for multiple ontology integration the procedures have to be adapted, such as IF-Map, whereas some practically cannot be scaled up, such as the use of intersection ontologies as discussed in chapter 4 and *Figure 4.2*.

A higher level of 'semi-automatic' ontology construction and integration was achieved with SoftOnt (Mena *et al.*, 2000), whereas IF-Map accomplished integration largely automatic with only minor additional manual refinements, with the cost being loss of information (Kalfoglou and Schorlemmer, 2002). The agent-based approach from Akahani *et al.* (2002) seems fully automatic, yet also mention the information loss. This 'best match mapping', as well as the aforementioned semi-automatic integration, is largely based on automation of heuristics. These include the structural information like concept names, relations, concept types, comparisons of the labelled-graph structures and the use of data instances; see e.g. Madhavan *et al.* (2002) and Noy and Musen (2002 and 2003 *in press*) for a brief overview of the software packages. Doan *et al.* (2002) compare the effectiveness in percentage matching accuracy of automated name learners, content learners, meta learner and relaxation labeller, and combinations thereof, with several ontologies. Two interesting aspects are that one simple name learner is least effective and that there is considerable difference in outcome when reversing the matching (O_1 to O_2 versus O_2 to O_1). The latter is of particular interest, especially if one were to analyse this on the structural and semantic level (i.e. why is there a difference and what caused it), not addressed by its authors. Despite this lacuna, the approach of combining several automated heuristics is a step forward compared to the 'single-issue' heuristic (semi-)automated ontology integration.

5.2 Software applicability and usability

Most articles are focussed purely on software functionality and performance. It might be argued if the integration software systems are sufficiently mature to be subjected to user and usability testing, or if this should have been considered when developing the integration software. Nevertheless, Lambrix and Edberg (2003) took software packages Protégé-2000 with PROMPT[57] and Chimaera[58] to the test with computing scientist and biologists. A main difference between the two appeared to be that Chimaera deals with *where* something should happen whereas PROMPT suggests *what* actions can/should be taken; Chimaera's merging is faster because it incorporates the non-matched concepts into the other ontology automatically. The tool is less user-friendly because there are [too] many menu options – this also may be interpreted as having more functionality than PROMPT and thereby a flatter learning curve. Contrary to previously reported research that 'biologists just cannot formalise their knowledge well', hence are not well equipped to work directly with ontologies, Lambrix and Edberg did not find a significant difference between the two types of users when they were integrating the Gene Ontology with the Signal Ontology. The reasons why these researchers did not find a difference may be multiple, and worthwhile investigating further. For example: did the prior lecture on ontologies made it clearer, or was it how the merging suggestions were phrased by the software, or the user guide and/or online help system of the software? Were the volunteers not as random as suggested, or the researchers better teachers / communicators than the average (stereotype) computer scientist?

One can take the development of the ontology integration software one step 'further' by not focussing purely on the technical features and automation of integration heuristics, but to use for example Constructive Technology Assessment (Schot, 1999) between biologists and computer scientists. Although CTA is primarily designed for 'technology and society', a similar approach based on the consensus conference model to "shape anticipation, reflexivity and learning" in order to improve the design process of the integration software could be useful and may increase acceptability of both the development and (re)use of ontologies.

However, equally important for designing ontology integration software is to create clarity in types of integration, such as indicated in *Figure 4.6* and *Appendix C*, and when clearer distinctions and definitions are formulated it is also possible to identify more precisely the most appropriate integration operations for certain goals based on given input. For example, merging ontologies of the same domain will benefit from a linguistic analysis of concept names and relations by label comparisons and use of a thesaurus or content learners. When the subject domain is highly specialised but interdisciplinary, such as a wastewater treatment plant, the former would be a relatively minor function running in the background, but emphasis put on presentability factors such as the user interface and the feedback on matches/conflicts, and extra features such as (a workflow) guidance with automatically created questions on the candidate elements. On the other hand, if one wants to 'integrate' ontologies by using sections of multiple ontologies for a conceptual model, ontology commitment layer or a new domain ontology, a select and drag-'n-drop feature will be beneficial. There are multiple such if-then suggestions, based on heuristics and theory, which suits a separate research effort to cover it comprehensively.

Summarizing the efforts on ontology integration software, each provides a, partially automated, solution to a specific aspect of ontology integration within their chosen implementation language. It will benefit from a combination of such features and (re)structuring them in accordance with their use for the specific integration tasks and content that will be integrated.

[57] Available online at: http://protege.stanford.edu/.

[58] From KSL at Stanford: http://www.ksl.stanford.edu/software/chimaera/.

6. Conclusions and further research

6.1 Conclusions

There are many aspects to data and domain heterogeneity increasing the possibilities of conflicts and mismatches when combining conceptual data models and ontologies, which will not be resolved easily, if ever. On top of these aspects are the difficulties of inherent in biological data adding to challenges in resolving heterogeneity when integrating data and subject domains. If one extends this view to ontologies, there can be identified different types of ontologies according to the level of formalism used and categorise them according to subject, decreasing potential for interoperability. This is exacerbated by the methodological differences in constructing models (empirical or theory-based) and development phases from informal to formal ontologies.

The pilot experiment with the ecological modelling software STELLA, guided bottom-up development of ontologies might be within reach by formalising the identified correspondences between the elements in the ecological model and computing terminology. The methodology of using extended semantic representations to organise equations in a placeholder objects model proved an approach useful for computing science.

Subsequent research into ontology integration revealed that although ontologists demand from the subject matter experts to reach consensus, there is no such thing concerning the multiple interpretations as to what constitutes 'ontology integration' and its related concepts such as merging, matching and so forth. Terms, definitions and practices found in a representative sample of the extant literature were structured and loosely categorised on a scale of combining ontologies. In addition, expectations on integrating ontologies of the same, similar and orthogonal subject domains were formulated, where each combination has both positive and negative factors. Semantic versus structural integration was highlighted with an example of the polder ecological niche, so were the potential positive effects and complications of facilitating multilingualism for ontology development and integration. It revealed that a strict separation between semantic and structural integration is not as obvious as the definitions might suggest. Other examples include ontology construction via the DOGMA approach with relation to microbiology that may improve reuse of knowledge even more and may assist in clarifying the multiple understandings of the Defined Terms Ontology, highlighting modelling paradigm heterogeneity and providing an analysis of the model / ontology of Defined Terms of plant taxonomy, which may benefit from a higher level of formalism and clear definitions and justifications for the taken methodology.

Ontology integration software was briefly addressed. Each application provides a partially automated solution to a specific aspect of ontology integration within their chosen implementation language. Compared to the automation of the heuristics of integrating ontologies on the semantic level, automation on the system and syntactic level is relatively straightforward and achieved; semi-automation of semantic integration is still a hot research topic.

6.2 Open issues

6.2.1 Areas of interest

This report directly and indirectly indicated several facets related to ontology integration that are still unclear, or at the time of writing either not well known to the research community or still in the early stages of research. This chapter highlights some of the areas that are in need of further research.

The sub-discipline of ontology integration would benefit from efforts to achieve an agreement on the various activities that fall within the liberally used concept 'integration'. This may well result in a simple vocabulary to define each type of integration (mapping, merging and so forth) and clarify which ones are synonyms and which ones homonyms, or, in a more formal and structured fashion, to develop an ontology of ontology integration.

With these clear definitions, it should be easier to identify which type of integration would suit what combination of ontologies, bearing in mind the reason for the integration. For example, with two ontologies, an ecological and an agricultural one, say, the production process of rice that needs part, but not all, of the concepts of the ecological ontology and may have some of the ecological concepts defined within the agricultural ontology, one may prefer to create a separate intersection ontology for the particular domain. However, one may decide to merge a forest ontology with the ecological ontology because of its closer relevance of the subject domain of ecology. Via extensive reasoning and testing one might be able to extract patterns and define a decision tree to guide integrators to identify the optimum type of integration strategy for a given requirement. Here it is important to work with parameters, properties, of the different kinds of integration that, with a certain combination, might say 'map' or 'match', together with the given requirements: this still leaves open opportunities to refine, or maybe even redefine, one or more (sub-)concepts of integration.

A following step could be the identification of factors or properties affecting integration and subsequent (partial) automation of the actual integration process; for example guided by concept similarity searches on labels and their definitions (e.g. by using a thesaurus) and/or on the structure of their formal representations, searching the ontology for hypernyms, avoiding cycles, terms with contradictory ranges etc. Note here, that to achieve the semantic integration of multiple ontologies, the implementation of integration on the syntactic and structural level is a prerequisite. Two factors in semi-automatic integration are of particular importance. First, how should one 'guide' the SME, e.g. using directed questions to answer by the SME or providing a model framework to be validated? Secondly, what are the effects of loosening ontology integration when settling for approximations of concept mappings, primarily how to measure the loss of semantics and data, and 'what if' scenarios of accepting integration of lightweight ontologies into formal ontologies.

Relevant for all three levels (semantic, structural and syntactic) integration, is the development of the Semantic Web: if the wider public does accept it, as it seems to be the case, this will, in the author's opinion, have a negative impact on ontology development, because integrating less well-defined ontologies is more difficult, if not impossible due to the lack of rigour, to achieve. This also may have an effect on the SEEK ontologies: for it to be used relatively widely, it must be (made) compatible with other ontology efforts. Likewise, if one wishes to extend (one of the formal) SEEK ontologies with one developed less formally: should one allow the dumbing down (or vice versa)? Aside from the (in)formalness of the Semantic Web, popularisation of ontologies likely will entail localisation of ontologies, analogous to software localisation. This does not necessarily pose a huge problem to be taken into account if it were to be limited to simple dictionary translations. Depending on one's view, it is (un)fortunate that especially in the disciplines of ecology and the agricultural sector many concepts do not translate well, or are ontologically seen as from a different origin, which can affect ontology integration in various ways.

Though not specifically addressed in this report, is the integration on the system level, and the implementation of ontologies, ontology editors and their interoperability. Ontology development environments (ODEs) are of relevance to define the ontology development process; for example revision control during/after the integration procedures, as well as the general development framework, which may be analogous to one of the more commonly used software development processes like prototyping and incremental development. In the subject domain of biology, bottom-up development of ontologies likely will be of particular importance, both because there are already a multitude of (software) models that may be 'pulled' onto the higher abstraction layer and secondly, because it will be extremely difficult to start with modelling top-level biological concepts in FOL – and might not be possible at all. With a bottom-up approach one can 'move' up the hierarchy from informal, lightweight ontologies to wherever it is possible to achieve in the region of formal ontologies.

Existing manual and (semi-)automatic integration efforts involve database query mechanisms, agent-based systems, infomorphisms, among others – but an ontology stored in a database cannot easily query/converse with an agent. To what extent do such variations in implementations have an effect on the possibilities to implement ontology integration? Are all types of integration possible with each type of implementation so that one freely can choose one's own preference of representation of an ontology on the system level? If not, what is not possible, where and why is it not implementable?

If the identification of type of integration and the actual integration process are within reach, or even possible at all, a natural next step would be to combine and automate these two processes in an ODE. With the help of ontology libraries to achieve the highest level of reuse, a developer could select the ontologies of relevance and have them automatically integrated in the appropriate manner. However, considering the present status of research in ontology integration, this may take a while.

Another aspect related to this automation and the ODEs, is the effectivity and usability of the software for both the computer scientists and the SMEs, and the understandability of the ontologies with their representations. In the majority of case studies, it has been observed that SMEs do not formalise their knowledge easily. This may be of various reasons, such as the knowledge-based life sciences, lack of training, lack of communication skills of computer scientists, for which there may be solutions or different approaches to alleviate the problems; one can think of CTA and the intermediate representation models that can be explored further.

Most of the areas of interest outlined in this paragraph emphasise the technical aspects involved with development and integration of ontologies, but only sparingly the social aspects such as what the differences are in methodology/ies between computer scientists and the SMEs in biology/ecology, how this affect development, integration and reuse of ontologies, what the effects of the approaches to pursue ontologies and computing have on the life sciences and if ontologies are indeed the panacea it claims to be. Using a methodology and technology just because it is there does not imply it is the right path to follow, but depends on a multitude of factors outside the realms of computing science, such as the sociology of cooperation and science/art of communication.

6.2.2 Research questions and approaches

The abovementioned 'areas of interest' can be reformulated into the following research questions and brief approaches how these might be answered, which may, or may not, be pursued. There are three categories: ontology development, ontology integration and interdisciplinary approaches to ontologies and social informatics. *It will not be feasible to address all aspects raised here within one research project; that is not the intention*: this paragraph comprises a range of research subjects and questions to provide a flavour of the myriad opportunities and challenges within ontology research – although it still is only a small selection of possible research topics.

Ontology integration

- *Can one categorise ontology integration and its sub-concepts, like merging and mapping, into a taxonomy, or even an ontology, of ontology integration?* After a comprehensive survey of the literature to elicitate the (mis)use of integration-related concepts, these various uses could be categorised according occurrence of usage across the reference literature and operation-oriented factors like 'integration' of concepts and/or instances, creation of a new (intersection) ontology, exact correspondences between concepts of the integrating ontologies or use of approximation heuristics and so forth. When successful, this should be communicated to the community of ontology researchers as a draft version up for discussion, and, like ontologists ask from SMEs, try to reach an agreement on the terminology.

- *Is it possible to find patterns in optimum integration strategies?* I.e., given ontologies O_1 and O_2 of subject domain(s) *a* (and/or *b*) and ontology type(s) *x* (and/or *y*) that need to be 'integrated' to achieve goal Γ, suggest the type of integration that likely will provide the best results. This builds upon the previous question, and in addition requires an overview of ontology types, including the effects of lightweight and heavyweight ontologies on integration, as well as insight in the effect of the subject domain on ontologies themselves and their integration (including a treatise on the distinctions of biological data from other subject domains). Regardless if extensive knowledge is available in the literature on domain, integration type and successfulness, this research would benefit from experimentation with various integration software: as a subsection, one could investigate especially the workflow/questions and intermediate model-guided integration software, because they might be best suitable for the subject domain experts of the SEEK project. The former involves software with different levels and features of automated heuristics to create the questions that are to aid the SME, as opposed to finding the differences and similarities manually. The second, an intermediate 'helper' representation, is to bridge the gap between SMEs and informaticians to represent the formal knowledge in a manner the SME can understand and can be used by them to formulate their knowledge an a for the computing scientist useable manner (i.e. to convert and include it into the ontology). However, a potential drawback of developing such an intermediate representation is that it may be a 'one-off' exercise for each collaboration and not readily be useful for others.

  One can take SEEK, as well as SEEK-related ontologies that are readily available on the Internet to investigate the potential use of the SEEK ontologies by different research communities ('marketability' from a technical viewpoint), and test a same pre-defined goal via various methods in order to determine characteristics and the pattern(s) for (un)successful integration efforts. With this, one may to be able to make for example a decision support or workflow system to simplify the 'preparation' stage of ontology integration (i.e. how one can, or cannot, go about doing the actual implementation of the integration of the ontologies.

- *If heuristics can be identified, can the process of proposing an integration method be automated, if yes, how?* This builds upon the previous point. Determine what questions in what sequence would need to be asked of an ontology engineer in order to be able to propose the best-chance integration method, alike a decision support system. Further, it may include suggestions for software available that would suit the particular integration task analogous to the WonderTools[59] suggestions for ontology creation, and if not available list the requirements that the 'ideal software' would need to have to achieve the integration

[59] http://www.swi.psy.uva.nl/wondertools/

- *How and where can these (semi-)automated integration heuristics be incorporated in the overall ontology development process?* To answer this, various sub-questions can be formulated: *Would this involve providing a separate module on 'integration suggestions', or tightly integrated with existing software, and with what software? To what extent limits the choice to be made for the latter, the possibilities of integrating ontologies? Are all types of integration possible with each type of implementation so that one freely can choose one's own preference of representation of an ontology on the syntax and the system level? If not, what is not possible, where and why is it not implementable?* There are ODEs, editors, revision managers, integrators and libraries that are all relevant for suggesting the type of integration and actually carrying out the manual, semi- or automated integration. Other researchers reported experiments and conducted some surveys on the use of the myriad of ontology(-related) software, which could potentially facilitate answering the main question in this section and its range of sub-questions. It is likely that it is not possible to achieve everything with one piece of software at the time of writing, which will require a careful analysis on all the advantages and disadvantages before implementing the (semi-)automated integration heuristics, including formulating requirements for this 'ideal integration software'.

- *Knowing integration parameters, patterns, implementations etc, are there lessons to be drawn for reusing ontologies?* For example, one type of ontology might be exceedingly difficult to integrate, but ideally one would want to use and reuse ontologies either in part or complete. With the proliferation of ontology creation, a set of best practices for ontology construction, with ontology integration in mind, should be created, as well as organised versioning and ontology library management. At present, there is no such organisation with relation to ecological ontologies, although the AOS Project for agriculture is somewhat related. One way to make available the ecological ontologies for direct use is via an ontology server, but also will need to include 'extraction' services where users can download an ontology in their preferred syntax for local reuse for example.

Ontology development

- Opposite to Ontology-driven Information Systems (OISs) advocated by e.g. Guarino (1998), is the approach advocated by e.g. Meersman (2001) and used in the PrometheusDB Project. There are two kinds of starting point: first, there are multiple conceptual models of applications and there is a desire to establish consensus in the subject domain, taking the conceptual model as a beginning. Second, during the analysis stage of software development, it 'appears' that an ontology needs to be developed in order to find solutions to one or more problems observed. Several facets can be investigated in these bottom-up approaches. Factors that may be more, or less, important are: *does having the existing models help or hinder ontology development? If the former, what aspects contribute most, if the latter, why does it hinder? Does it constrain the thought process? Could it end up in conceptual schema integration instead of ontology creation? Can the division into an ontology base and ontological commitments* (in Jarrar *et al.* (2003) and an example is included in the 'aspects of ontology integration' file) *alleviate this problem? Any other approach? What is the influence of the conceptual modelling method on this methodology?* Addressing these questions will require cooperation with at least 2-3 analysts who conducted an analysis in the same application domain to obtain reliable information, which may not be practicable. Else, it would take the researcher to 'play' the various roles given 2-3 existing conceptual models. Further, one may expect that a modelling method like ORM will be more effective than OO or ER due to the implementation restrictions embedded in these modelling methods.

- On the second bottom-up approach mentioned in the previous point: what caused the decision to look into ontologies, *what are the reasons to pursue with ontology before 'finalising' analysis/design of the* *intended application? Did a change of approach to the project occur? If yes, what and why? If not, why not (understanding of the ontology process/proper life cycle model or random start/etc)? Did the ontology approach answer questions raised during the conceptual modelling, or solved the problems observed during the conceptual modelling? If yes, where and what was changed in the conceptual model? What were the causes to start implementation, sufficiently large ontology or non-scientific reasons; or did it not solve problems/answer (some of) the questions, if so, why not? Was pursuing the area of ontologies 'kicking the problem upstairs', i.e. there are still questions/problems, but then on a more abstract level?* From answers on these questions, e.g. by investigating the process encountered with the PrometheusDB Project as a case study and a new analysis project to be conducted (related to a SEEK-related discipline), one may be able to extract indications for a 'principle' of working procedures for bottom-up ontology development and identify what an ontology can and cannot answer/solve.

* Opinions vary, but some researchers (e.g. Meersman and Jarrar (2002); Liu (personal communication); Bowers and Ludäscher (2003); Kendall *et al.* (2002); Sugumaran and Storey, 2002) consider it possible to 'convert' an ontology into a conceptual model, or even a computational model (refer to "Aspects of ontology integration" for more detail on ontology versus conceptual model). For example, Kendall *et al.* (2002) developed Visual Ontology Modeler, with an add-in into Rational Rose, extending UML to enable modelling of frame-based KR concepts. A problem with this is, that UML is closer to being a computational model than a conceptual model (Juristo and Moreno, 2000), hence posing some restrictions on reusability, however their approach may be more generally applicable than the envisaged component ontology of Liu. Of a more general approach is the DOGMAModeler that uses an extension of ORM, ORM-ML, which 'translates' ORM into XML to represent a machine-readable, hence exchangeable, version of the ontology (Jarrar *et al.*, 2003). The potential advantage of using a tool such as the online Ontology Management Portal or DOGMAModeler is that it requires relatively small changes to create the conceptual model from its ontological representation, the latter is graphical and provides (near) natural language descriptions that aid subject matter experts considerably (Keet, 2003a; Halpin, 2001), the latter of particular importance when modelling biological/ecological data. Further, ORM allows for both object-oriented and relational modelling (Halpin, 2001), providing the software developer with more flexibility and possibilities for concept reuse. The benefits of generating a framework for each particular conceptual model designed for an application (as opposed to application ontologies) are potentially huge, especially in conjunction with an ontology integrator or ontology library. Related research question to be answered are: *What has to go into such an ontology library? Should one divide these up into several 'topical packages', including a 'base library component', analogous to OO IDEs? Would it be more beneficial to aggregate certain subject domains, e.g. that may be easier to work with? What are the requirements of such software? How can one connect e.g. DOGMAModeler, or similar, to a program like VisioModeler?* (Which in turn does generate a relational database automatically) *Can one, and should one, allow reverse engineering form a conceptual model into a 'proposed' ontology? If yes, how to organize versioning and how/where to integrate that with an ODE? To what extend would this facilitate round-trip ontology engineering and use?* (See also section 'ontology development process') *Should this envisioned software be functioning as an application, or maybe having a base partially or entirely on an Ontology Server?* Most of these aspects are in the early stages of research, and any endeavours will require a strict narrowing down of activities to undertake, for example to investigate this aspect from the UML/OO perspective, or ORM. Others are to decide if one were to design a 'translator' that can provide a framework for generating a conceptual model from an ontology ourselves, or to use a third party software application like DOGMAModeler. If the latter, it may be advantageous to look into versioning of ontologies, tracking changes and so forth, and a looking into the potential of reverse engineering (semi-automatic bottom-up generation of ontologies) as well as the practical usability for developing

applications based on ontologies created as part of the SEEK project. Further, *do you loose information when 'translating', converting, between the definitions/relations etc. of the components in an ontology and a conceptual model? If yes, what? Does XML suffice to represent the ontologies (compared to other DL, KIF formats)? Like with top-down ontology-driven information systems, is this approach too limited for application development, and if yes, why (if not, what range of tasks does it meet)?* Answering this set of questions will likely involve a more mathematical approach with relation to the 'conversions' – see Bench-Capon *et al.* (2000) and §2.2.1 for further information.

* An aspect briefly mentioned in *Example 5* is the use of language of the SEEK ontologies, which is English. Considering a multilingual approach may, or may not, be a useful avenue, not only for the problems it raises, but it may not be a solution to knowledge sharing, or it might even be a strategy of avoiding to reach consensus on concepts. On the other hand, it could increase the user base. Phrased into research questions: *Does the (main) ontology need to be in one language, and one only – as is with the SEEK Project –, or interoperable multilingual ontologies (e.g. the AOS Project)? Does the latter represent the same structure with other labels, or maybe captures different semantics 'hidden' behind translations? Can one determine, and if yes how, the loss of semantics when adhering to one 'global' or a 'backbone' language, and what effects does this loss have on the comprehensiveness and chance of success of the SEEK Project? Do advantages of a single language counterbalance the loss? Is it reasonable to demand or expect that subject matter experts are not only experts in their discipline, but also fluent in at least one other language, and, stretching it a bit further, be able to formalise their knowledge?* One example of potential misunderstandings caused by a multi-lingual approach was provided in *Example 5* in this report; however, in order to answer these questions, one may need to carry out a comparative study between people from the same discipline creating on ontology of several domains (or subsections thereof). Another avenue could be in line with ONTOGENERATION (Aguado *et al.*, 1998), where the English-language ontology CHEMICALS is integrated with Spanish language related ontologies, so that users can query an ontology in their native language and receive responses likewise; providing such systems for other languages on top of (integrated with?) the 'core' SEEK ontologies could alleviate the language problem to some extend, and increase the potential user base.

Interdisciplinary approaches to ontologies and social informatics

* It is well known that many subject matter experts cannot formalise their knowledge well (see e.g. Aguado *et al.* (1998) or the experiences of the Prometheus Project). One approach is to teach them how to create ontologies, another, pursued by the ONTOGENERATION project, to a lesser extend by Keller and Dungan (1999), and embedded in a tool like VisioModeler, is to create an intermediate representation model with can be understood by subject matter experts, but is also usable from an ontology engineer's perspective, in order to speed up correct creation, maintenance and use of ontologies. *What is an 'understandable intermediate model'?* Ecologists are familiar with the process of creating models, albeit different from a computing perspective and more focussed on the practical use of the model, but an idea of categorisations and so forth nevertheless. Figuring out an appropriate intermediate model partially depends on the adherence of an ontology development process: of this is top-down, one might as well start creating such a model from scratch, on the other hand, were one to start from a bottom-up process of ontology development, one of the existing (graphical) modelling methods could be explored first, ORM in particular. Subsequently, this intermediate model needs user testing of such a model, and assess if it is indeed effective in speeding up ontology development and improving its quality.

* Thee lion's share of ontology development, versioning, integration and so forth is focussed on common sense subject areas like universities (Noy and Musen, 2003 *in press*) or the government (Mitra *et al.*, 1999), but *what effect does the complexity of biological data have on ontology creation as well as maintenance and usability? Can one formalise all biological data? If not, why and what level of lightweight ontology would be best suitable for development and maintenance of biological ontologies? Is this in line with the current aims and practices of the SEEK project?* General characteristics of biological data and its interplay with core and applied science are addressed elsewhere (Keet, 2003a, 2003b), but may need to be extended to include specifics on ecological data. Formalising this biological data has happened sparingly, is primarily focussed on procedural, OO or ER modelling, but may be possible with richer modelling techniques like ORM, FCA or CGs. Proof of concept may be achieved by modelling some of the hitherto 'unformalizable' biological knowledge and identifying the conceptual modelling features that make this possible. Else, one should be able to identify the hiatus(es) in conceptual modelling methods and propose changes to meet these 'impossible requirements'.

* Somewhat related to ontology integration but more focused on the social dimension, is the use of SEEK type of ontologies primarily in the subject domain of agriculture, which relies on ecological data for its management in the primary production sector and in relation to simulation software. Development decisions will need to be taken on, for example, *can one, and should one, integrate ontologies of agriculture, such as the AOS, with SEEK ontologies, or extend plant taxonomy ontology with more details per plant* (like the rice/grasses [Gramene] and maize [Plant Ontology] approaches)*? Might it be better to take sections of an ecological ontology and create separate ad-hoc ontologies, only reusing part of the SEEK ontologies?* Answering such questions requires an investigation into the motives of agriculturalists on what they would expect, where they would see the potential benefit, from ontologies of ecology. Following this exercise in requirements elicitation and specification, one can devise a reasoned approach to either decide to meet such demands, and if yes how, or what cannot be accommodated for, and why not.

* There are implications caused by the enforcement of IT/computing methodologies of research [engineering] onto life sciences, e.g. the categorisations of ontologies (plus requirement to formalise their knowledge) and increased use of structured software tools. This type of controlled approach is different from working methods in biology (including ecology); not that biology is unstructured, but arranges knowledge alike a 'mental network of information' as starting point – i.e. associative and incremental. Successful development of bioinformatics software depends on input of biologists, which in turn may alleviate some of the mind numbing work allowing the researcher to pursue the more theoretical aspects of his/her subject domain. However, for the by biologists provided input to be useful to computer scientists, it has to meet certain requirements, in effect demanding from the life science researcher another approach of looking at, and using, knowledge and information. *Is there a change in methodology and does it have an effect on the research methods of life science researchers?* For example, the differences in model creation based on empirical data and expanding this or with its starting point the theory and a framework of key factors/object. Secondly, it used to be that the primary sciences 'dictated', or at least set the boundaries of possibilities, for the engineering disciplines (apart from the engineer's creativity of course), *could it be that with the huge expansion of technology over the last 50 years, engineers/computer scientists are about to direct the possibilities and methodologies of research for the biologists? Does the influence of IT/Computing have a (profound) effect on the type of research questions asked, and hypotheses formulated, by the life science researchers? Or could it be, that there is no such change, but new sub-disciplines are being created in parallel with the existing methodologies? Do IT specialists, as is quite common, simply create a new need to keep themselves busy?*

* Following from the previous point, *what effect does bio/ecoinformatics have on the curricula offered at universities?* A certain 'mushrooming' of bioinformatics top-up courses have been developed, i.e. a biologist studies IT implementations to improve skills for a year and vice versa, where an IT specialists can take a crash-course in biology (primarily genetics and proteomics) for a year or two, *but do they really deliver? Do undergrad curricula change to meet the demand of the bioinformatics industry? Are the top-up courses sufficient or would one need people with two full studies in order to really exploit the potential of bioinformatics in order to live up to its promises?* It is my experience that the two disciplines teach another way of thinking, how to analyse, which implicit assumptions one can make consciously and which are after some time of study carried out virtually unconsciously, i.e. how to do science. For example, the positivist approach towards conducting the scientific enterprise is reductionism, but this is differently used within the life science and computing/engineering. In the former, one does so because to be able to investigate some the vast amount of (hidden) knowledge, it *has to* be divided to make it comprehensible to the human mind, whereas the scientist knows the subject under investigation forms part of a larger system, hence reductionism as a necessity and with the knowledge the compartmentalisation is a simplification. In contrast engineering (including IT and computing), which is designed so one actually conveniently *can* chop up large subject domains into virtually independent sub-disciplines (compared to the life sciences) and do what you like as long as the interface matches the human-defined standards (API, SCSI and so forth).

To investigate this further and bring to the fore such fundamental differences may improve understanding for each method of working.

* From various sides of research has emerged the claim that computer scientist think that the life sciences specify "impossible requirements" that cannot be met; conversely, there exist multiple "unanswerable questions" phrased by computer scientist/engineers towards biologists. *What is impossible and what unanswerable? Why? Are there options for computer scientist to work on the 'impossible requirements' so that they may be met some time in the near future? Any prioritisation?* This may be combined with the second point in this subsection.

* Despite the claims of improvements in reuse of knowledge and interoperability, ontologies have not yet quite proven these claims convincingly even within 'common sense' research areas, let alone biology. It might be that there are not yet sufficient ontologies to take advantage of reuse, by e.g. to develop ontology libraries analogous to software libraries in C++ and Java builders, and interoperability of software. The less patient scientist might contemplate the validity of the 'marketing claims' in favour of ontologies. *Are ontologies a "necessary evil" or do the ecologists consider it as "time well spent"? If the former, is it considered by them to be for their own benefit, and do they realise that? If the latter, could it be, that viewing their subject matter from another perspective actually generated new insight and knowledge hitherto hidden? If it is seen as a waste of time, why, and what can be done to convince them it's not such a bad idea – or is it that ontologists with their zeal actually didn't realize it is not practicable?*

References

Aguado, G., Bañón, A., Bateman, J., Bernardos, S., Fernández, M., Gómez-Pérez, A., Nieto, E., Olalla, A., Plaza, R. and Sánchez, A. (1998). ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation. *Proceedings of the ECAI'98 Workshop on Applications of Ontologies and Problem Solving Methods*, Brighton, U.K.

Akahani, J., Hiramatsu, K., and Kogure, K. (2002). Coordinating Heterogeneous Information Services based On Approximate Ontology Translation. *First International Joint Conference on Autonomous Agents & Multiagent Systems (AA MAS 2002)*, Bologna, Italy.

Akkermans, A.D.L., Brussaard, L., Didden, W.A.M., Kammenga, J. and Marinissen, J.C.Y. (1996a). *Functioneren van ongestoorde bodemoecosystemen* [Functioning of non-disrupted soil ecosystems]. In: Course reader Bodembiologie H300-219, Wageningen Agricultural University, the Netherlands. 24p.

Akkermans, A.D.L., Brussaard, L., Didden, W.A.M., Kammenga, J. and Marinissen, J.C.Y. (1996b). *Interacties tussen bodemorganismen* [Interactions between soil organisms]. In: Course reader Bodembiologie H300-219, Wageningen Agricultural University, the Netherlands. 15p.

Andreasen, T. and Nilson, J.F. (2004). "Grammatical specification of domain ontologies." *Data & Knowledge Engineering*, **48**(1): 221-230.

Argent, R. M. (2003 *in press*). "An overview of model integration for environmental applications—components, frameworks and semantics." *Environmental Modelling & Software*, **xx**(xx): xx-xx.

Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F.F., Pawson, T. and Hogue, C.W.V., (2001), "BIND — The Biomolecular Interaction Network Database". *Nucleic Acids Research*, **29**(1): 242-245.

Baker, P.G., Goble, C.A., Bechhofer, S., Paton, N.W., Stevens, R. and Brass, A. (1999). "An ontology for bioinformatics applications." *Bioinformatics*, **15**(6): 510-520.

Baskent, E.Z., Wightman, R.A., Jordan, G.A. and Zhai, Y. (2001). "Object-oriented abstraction of contemporary forest management design." *Ecological Modelling*, **143**: 147-164.

Bench-Capon, T., Malcolm, G. and Shave, M. (2000). *Semantics for Interoperability: relating ontologies and schemata.* Department of Computer Science, University of Liverpool. 19p. http://www.cs.ucsd.edu/groups/tatami/seek/. Date accessed: 2-1-2004.

Bernstein, P.A., Halevy, A.Y., Pottinger, R.A. (2000). "A Vision for Management of Complex Models". *SIGMOD Record*, **29**(4): 55-63.

Bowers, S. and Ludäscher, B. (2003). Towards a Generic Framework for Semantic Registration of Scientific Data. *Semantic Web Technologies for Searching and Retrieving Scientific Data*, Sanibel Island, Florida, USA.

Brilhante, V. and Robertson, D. (2001). Metadata-Supported Automated Ecological Modelling. In: *Environmental Information Systems in Industry and Public Administration.* C. Rautenstrauch (ed.). Hershey, Pennsylvania: Idea Group Publishing.

Calvanese, D., De Giacomo, G. and Lenzerini, M. (2001a). A Framework for Ontology Integration. *Proceedings of the First Semantic Web Working Symposium.*

Calvanese, D., De Giacomo, G. and Lenzerini, M. (2001b). Ontology of integration and integration of ontologies. *Proceedings of the International Workshop on Description Logics (DL2001).*

Ceccaroni, L., Cortés, U. and Sànchez-Marrè, M. (2004 *in press*). "OntoWEDSS: augmenting environmental decision-support systems with ontologies." *Environmental Modelling & Software*, **xx**(xx): xx-xx.

Ceccaroni, L., Cortés, U. and Sànchez-Marrè, M. (2000). WaWO - An ontology embedded into an environmental decision-support system for wastewater treatment plant management. *Proceedings of ECAI2000 - Wo9: Applications of ontologies and problem-solving methods*, Berlin, Germany.

Ceusters, W., Smith, B. and Flanagan, J. (2003). Ontology and Medical Terminology: Why Description

Logics Are Not Enough. *Proceedings of TEPR 2003 — Towards an Electronic Patient Record*, San Antonio, USA.

Chalupsky, H. (2000). OntoMorph: A Translation System for Symbolic Knowledge. *7th International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann. pp 471-482.

Corcho, O., Fernandez-Lopez, M. and Gomez-Perez, A. (2003). "Methodologies, tools and languages for building ontologies. Where is their meeting point?" *Data & Knowledge Engineering* **46**(1): 41-64.

Correa da Silva, F., Vasconcelos, W.W., Robertson, D.S., Brilhante, V., de Melo, A., Finger, M. and Agustí, J. (2002). "On the Insufficiency of Ontologies: Problems in Knowledge Sharing and Alternative Solutions". *Knowledge-Based Systems Journal*, **15**(3): 147-167.

Doan, A., Madhavan, J., Domingos, P. and Halevy, A. (2002). Learning to Map between Ontologies on the Semantic Web. *11th International World Wide Web Conference*, Honolulu, Hawaii, USA.

Drysdale, R., (2001), "Phenotypic data in FlyBase". *Briefings in Bioinformatics*, **2**(1): 68-80.

Engbersen, J.F. J. (1994). *Biokatalyse I*. Course reader A400-209, Department of Organic Chemistry, Wageningen Agricultural University, the Netherlands.

Fensel, D., van Harmelen, F., Ding, Y., Klein, M., Akkermans, H., Broekstra, J., Kampman, A., van der Meer, J., Sure, Y., Studer, R., Krohn, U., Davies, J., Engels, R., Iosif, V., Kiryakov, A., Lau, T., Reimer, U. and Horrocks, I. (2002). "On-To-Knowledge in a Nutshell". *IEEE Computer*, ?. http://www.cs.vu.nl/%7Efrankh/postscript/IEEE-Computer02.pdf, Date accessed: 19-1-2004.

Fernandez-Lopez, M. Description of methodologies. *Ontology .org*: http://www.ontology.org/main/presentations/madrid/descriptions.pdf. Date accessed: 9-12-2003.

Ford, E.D. (2000). *Scientific methods for Ecological Research*. Cambridge: Cambridge University Press,. 564p.

Frishman, D., Heurmann, K., Lesk, A. and Mewes, H.-W., (1998), "Comprehensive, comprehensible, distributed and intelligent databases: current status". *Bioinformatics*, **14**(7): 551-561.

Gangemi, A., Pisanelli, D.M. and Steve, G. (1998). Ontology Integration: Experiences with Medical Terminologies. *Proceedings of the Conference: Formal Ontology in Information Systems (FOIS '98).*

Gangemi, A., Fisseha, F., Pettman, I., Pisanelli, D.M., Taconet, M., Keizer, J. (2002a). A Formal Ontological Framework for Semantic Interoperability in the Fishery Domain. *Proceedings of the ECAI-02 Workshop on Ontologies and Semantic Interoperability*, Lyon, France.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. and Schneider, L. (2002b). Sweetening Ontologies with DOLCE. *Proceedings of EKAW 2002*, Siguenza, Spain.

Ganter, B. and Wille, R. (1999). *Formal Concept Analysis – Mathematical foundations*. Berlin-Heidelberg: Springer-Verlag. 284p.

Gene Ontology Consortium. (2001). 'Creating the Gene Ontology Resource: design and implementation'. *Genome Research*, 11(8): 1425-1433.

Goerz, G., Buechner, K., Ludwig, B. (2003). Combining a lexical taxonomy with domain ontologies in the Erlangen Dialogue System. In: *Reference Ontologies and Application Ontologies Pre-Workshop Notes*. Grenon, P. (ed.). Faculty of Medicine, Institute for Formal Ontology and Medical Information Science (IFOMIS).

Goh, C.H. (1996). *Representing and reasoning about semantic conflicts in heterogeneous information sources*. PhD, MIT.

Graham, M., Watson, M.F. and Kennedy, J.B. (2003). 'Novel visualisation techniques for working with multiple, overlapping classification hierarchies'. *Taxon*, **51**: 351-358.

Grüninger, M. (1996). Designing and evaluating generic ontologies. *ECAI96's Workshop on Ontological Engineering.*

Guarino, N. (1997a). "Understanding, building and using ontologies." *International Journal of Human-Computer Studies*, **46**(213): 293-310.

Guarino, N. (1997b). Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In: *Information extraction: A multidisciplinary approach to an engineering information technology*. Pazienza, M.T. (ed.). Springer Verlag.

Guarino, N. (1998). Formal Ontology and Information Systems. *Formal Ontology in Information Systems,* *Proceedings of FIOS'98*, Trento, Italy, Amsterdam: IOS Press.

Guarino, N. and Welty, C. (2000). A formal ontology of properties. *Proceedings of 12th Int. Conf. on Knowledge Engineering and Knowledge Management*, Lecture Notes in Computer Science, Springer Verlag.

Guarino, N. and Welty, C. (2002). "Evaluating ontological decisions with OntoClean." *Communications of the ACM*, **45**(2): 61-65.

Halpin, T., (2001), *Information Modeling and Relational Databases*. San Francisco: Morgan Kaufmann Publishers. 761p.

Harmelen, F. van (2002). "The complexity of the Web Ontology Language". *IEEE Intelligent Systems* (March/April): 71-72.

Hefflin, J. and Hendler, J. (2000). Dynamic ontologies on the Web. *Proceedings of 17th National Conference on Artificial Intelligence (AAAI-2000).*

Heijst, G. van, Schreiber, A.Th. and Wielinga, B.J. (1997). "Roles are not classes: a reply to Nicola Guarino." *International Journal of Human-Computer Studies,* **46**(213): 311-318.

Heemskerk, M., Wilson, K. and Pavao-Zuckerman, M. (2003). "Conceptual Models as Tools for Communication Across Disciplines". *Conservation Ecology*, **7**(3). http://www.consecol.org/vol7/iss3/art8.

Hendler, J. (2002). "Ontologies on the Semantic Web". *IEEE Intelligent Systems* (March/April): 73-74.

Horrocks, I. (2002). "An ontology language for the Semantic Web". *IEEE Intelligent Systems* (March/April): 74-75.

Huang, G. H. and Chang, N.B. (2003). "Perspectives of environmental informatics and systems analysis." *Journal of Environmental Informatics*, **1**(1): 1-6.

Jaiswal, P., Ware, D., Ni, J., Chang, K., Zhao, W., Schmidt, S., Pan, X., Clark, K., Teytelman, L., Cartinhour, S., Stein, L. and McCouch, S. (2002). "Gramene: development and integration of trait and gene ontologies for rice". *Comparative and Functional Genomics*, **3**: 132-136.

Jarrar, M., Demy, J. and Meersman, R. (2003). "On Using Conceptual Data Modeling for Ontology Engineering." *Journal on Data Semantics Special issue on "Best papers from the ER/ODBASE/COOPIS 2002 Conferences"*, **1**(1): 185-207.

Kalfoglou, Y. and Schorlemmer, M. (2002). Information-Flow-based Ontology Mapping. *Proceedings of the 1st International Conference on Ontologies, Databases and Application of Semantics (ODBASE'02)*, Irvine, CA, USA.

Keet, C.M. (2003a). "Biological data and conceptual modelling methods". *Journal of Conceptual Modeling*, **29**, http://www.inconcept.com/jcm.

Keet, C.M., (2003b), 'Conceptual Modelling for Applied Bioscience: The Bacteriocin Database'. *CSPS: Computational intelligence/0310001*. Available online: http://www.compscipreprints.com/comp/Preprint/mkeet/20031008/1

Keet, C.M. (2003c). *The use of bacteria and bacteriocins in the food industry – modelled and documented in a relational database*. BSc Final Year Project, Open University, UK.

Keet, C.M. and Van Lune, F.S. (1997). *Invloed van de teeltwijze op de produktkwaliteit van tomaten*. [The influence of cultivation on the product quality of tomatoes]. Departments of Communication Science and Food Science, Wageningen Agricultural University, the Netherlands. 47p.

Keller, R.M. and Dungan, J.L. (1999). "Meta-modeling: a knowledge-based approach to facilitating process model construction and reuse." *Ecological Modelling*, **119**: 89-116.

Kendall, E.F., Dutra, M.E., McGuinness, D.L. (2002). Towards A Commercial Ontology Development Environment. *First International Semantic Web Conference ISWC'02*, Springer, Berlin.

Kent, R. E. (2000). The Information Flow Foundation for Conceptual Knowledge Organization. *Proceedings of the 6th International Conference of the International Society for Knowledge Organization (ISKO),* Toronto, Canada.

Kjelleberg, S., Nystrom, T., Albertson, N. and Flardh, K. (1990). "Papers presented at the Symposium on Nutrient Limitation: Global responses and Prokaryotic Development." *FEMS Microbiol. Ecol*, **74**: 91-

239.

Klein, M. (2001). Combining and Relating Ontologies: Problems and Solutions. *IJCAI Workshop on Ontologies*, Seattle.

Klein, M. and Noy, N.F. (2003). A Component-Based Framework for Ontology Evolution. *Workshop on Ontologies and Distributed Systems at IJCAI-2003*, Acapulco, Mexico.

Köhler, J., Philippi, S. and Lange, M. (2003). "SEMEDA: ontology based semantic integration of biological databases". *Bioinformatics*, **19**(18): 2420-2427.

Krishnamurthy, L., Nadeau, J., Ozsoyoglu, G., Ozsoyoglu, M., Schaeffer, G., Tasan, M. and Xu, W. (2003). "Pathways database system: an integrated system for biological pathways". *Bioinformatics*, **19**(8): 930-937.

Kumar, A. and Smith, B. (2003). "The Unified Medical Language System and the Gene Ontology: Some Critical Reflections". In: *KI 2003: Advances in Artificial Intelligence (Lecture Notes in Artificial Intelligence 2821).* Günter, A., Kruse, R. and Neumann, B. (eds.). Springer Verlag: Berlin. pp135-148.

Lambrix, P. and Edberg, A. (2003). Evaluation of ontology merging tools in bioinformatics. *Pacific Symposium on Biocomputing.*

Laser, U., Lehrach, H. and Roest Crollius, H. (1998). "Issues in developing integrated genomic databases and application to the human X chromosome". *Bioinformatics*, **14**(7): 583-90.

Liu, J., Peng, C., Dang, Q., Apps, M. and Jiang, H. (2002). "A component object model strategy for reusing ecosystem models". *Computers and Electronics in Agriculture*, **35**: 17-33.

Macauley, J., Wang, H. and Goodman, N. (1998). "A model system for studying the integration of molecular biology databases". *Bioinformatics*, **14**(7): 575-582.

Maddison, D.R., Swofford, D.L and Maddison, W.P. (1997). "NEXUS: an extensible file format for systematic information". *Systems Biology*, **46**(4): 59-621.

Madhavan, J., Bernstein, P.A., Domingos, P. and Halevy, A.Y. (2002). Representing and Reasoning about Mappings between Domain Models. *Eighteenth National Conference on Artificial Intelligence (AAAI'02)*, Edmonton, Canada.

Malmaeus, J. M., Håkanson, L. (2004). "Development of a lake eutrophication model". *Ecological Modelling*, **171**(2): 35-63.

Marjomaa, E. (2002). "Necessary Conditions for High Quality Conceptual Schemata: Two Wicked Problems." *Journal of Conceptual Modeling.* **27**.

Mason, B. (2003). "Wines bask in rising temperatures." *Nature Science Update*, 4 November 2003. http://www.nature.com/nsu/031103/031103-3.html. Date accessed: 23-1-2004.

McGuinness, D.L., Fikes, R., Rice, J. and Wilder, S. (2000). The Chimaera Ontology Environment. *17th National Conference on Artificial Intelligence, AAAI'00*, Austin.

Meersman, R. (2001). *Reusing certain database design principles, methods and techniques for ontology theory, construction and methodology.* STARLab, Vrije Univeristeit Brussel, Belgium. 17p.

Meersman, R. and Jarrar, M. (2002). *An Architecture For Practical Ontology Engineering and Deployment: the DOGMA Approach.* http://lsirwww.epfl.ch/courses/dip/2002ws/doogma.pdf. Date accessed: 26-11-2003.

Mena, E., Kashyap, V., Illarramendi, A. and Sheth, A. (1996). Managing multiple information sources through ontologies: relationship between vocabulary heterogeneity and loss of information. *Knowledge Representation Meets Databases, Proceedings of the 3rd Workshop KRDB'96*, Budapest, Hungary, Technical University of Aachen.

Mena, E., Illarramendi, A., Kashyap, V. and Sheth, A.P. (2000). "OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies". *Distributed and Parallel Databases*, **8**(2): 223-271.

Mena, E., Illarramendi, A. and Goñi, A. (2000). Automatic Ontology Construction for a Multiagent-based Software Gathering Service. *Cooperative Information Agents, The Future of Information Agents in Cyberspace, 4th International Workshop*, Boston, MA, USA, Springer.

Miall, A.D. and Miall, C.E. (2001). "Sequence stratigraphy as a scientific enterprise: the evolution and persistence of conflicting paradigms." *Earth-Science Reviews*, **54**(4): 321-348.

Mineau, G.W., Missaoui, R. and Godinx, R. (2000). "Conceptual modeling for data and knowledge management." *Data & Knowledge Engineering*, **143**(2): 147-164.

Mineter, M.J., Jarvis, C.H. and Dowers, S. (2003). "From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data." *Environmental Modelling & Software*, **18**(4): 379-391.

Mitra, P., Wiederhold, G. and Jannink, J. (1999). Semi-automatic integration of knowledge. *Proceedings of Fusion '99*, Sunnyvale, USA.

Nihoul, J.C.J. (1998). "Modelling marine ecosystems as a discipline in Earth Science." *Earth-Science Reviews*, **44**(1): 1-13.

Noy, N.F. and Musen, M.A. (2003 *in press*). "The PROMPT suite: interactive tools for ontology merging and mapping." *International Journal of Human-Computer Studies*, xx(x):xx-xx.

Noy, N.F. and Musen, M.A. (2002). Evaluating Ontology-Mapping Tools: Requirements and Experience. *Workshop on Evaluation of Ontology Tools at EKAW'02 (EON2002).*

Paterson, T., Kennedy, J.B., Pullan, M.R., Cannon, A., Armstrong, K., Watson, M.F., Raguenaud, C., McDonald, S.M. and Russell, G. (2004). A Universal Character Model and Ontology of Defined Terms for Taxonomic Description. *Data Integration In Life Sciences*, Leipzig, Germany.

Pazzaglia, J.C.R. and Embury, S.M. (1998). Bottom-up integration of ontologies in a database context. *5th KRDB Workshop*, Seattle, WA.

Pepper, S. (2000). The TAO of Topic Maps - finding the way in the age of infoglut. http://www.gca.org/papers/xmleurope2000/papers/s11-01.html. Date accessed: 26-11-2003.

Pinto, H.S., Gomez-Perez, A., and Martins, J.P. (1999). Some issues on ontology integration. In: *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving methods (KRR5)*, Stockholm, Sweden.

Plant Ontology Consortium. (2002). "The Plant Ontology Consortium and Plant Ontologies". *Comparative and Functional Genomics*, **3**: 137-142.

Priss, U. (2003). 'Formalizing Botanical Taxonomies'. *Proceedings of the 11th International Conference on Conceptual Structures, 2003*. Online preprint: http://www.upriss.org.uk/papers/iccs03.pdf. Date accessed: 30-9-2003.

Raguenaud, C. (2002). *Managing complex taxonomic data in an object-oriented database*. PhD Thesis, Napier University, Edinburgh. 196p.

Raguenaud, C., Pullan, M.R., Watson, M.F., Kennedy, J.B., Newman, M.F. and Barclay, P.J. (2002). "Implementation of the Prometheus Taxonomic Model: a comparison of database models and query languages and an introduction to the Prometheus Object-Oriented Model". *Taxon*, **51**: 131-142.

Reed, S. L. and Lenat, D.B. (2002). *Mapping Ontologies into Cyc*. CYCORP White Paper. http://www.cyc.com/publications.html. Date accessed: 2-11-2003.

De Ruiter, P.C., Bloem, J., Bouwman, L.A., Didden, W.A.M., Lebbink, G., Marinissen, J.C.Y., de Vos, J.A., Vreeken-Buijs, M.J., Zwart, K.B. and Brussaard, L. (1994a). "Simulation of dynamics in nitrogen mineralisation in the belowground food webs of two arable farming systems". *Agriculture, Ecosystems and Environment*, **51**, 199-208.

De Ruiter, P.C., Neutel, A.-M. and Moore, J.C. (1994b). "Modelling food webs and nutrient cycling in agro-ecosystems". *Trends in Ecology and Evolution*, **9**: 378-383.

Schot, J. (1999). Constructive Technology Assessment Comes of Age. The birth of a new politics of technology. *Proceedings of the International Summer Academy on Technology Studies*, Deutschlandberg, Austria.

Rousset, M.C. (2002). "The Semantic Web needs languages for representing (complex) mappings between (simple) ontologies." *IEEE Intelligent Systems* (March/April): 76-77.

Schlegel, H. G. (1995). *General Microbiology*. Cambridge: Cambridge University Press, 7th ed. 655p.

Sheth, A. P. (1999). “Changing focus on interoperability in information systems: from system, syntax, structure to semantics.” In: *Interoperating Geographic Information Systems*. Goodchild, M.F., Egenhofer, M.J., Fegeas, R. and Kottman, C.A. (eds.). Boston, Kluwer Academic Publishers.

Smith, J.M. (1974). *Models in ecology*. Cambridge: Cambridge University Press, (1979 ed.).

Sowa, J.F. (1997). *Principles of ontology*. onto-std.archive, Knowledge Systems Laboratory Stanford University. http://www-ksl.stanford.edu/onto-std/mailarchive/0136.html. Date accessed: 12-12-2003.

Sowa, J.F. (2000). *Knowledge representation*. China Machine Press (Thomson).

Sowa, J.F. (2001). Top-level categories. http://www.jfsowa.com/ontology/toplevel.htm. Date accessed: 15-10-2003.

Staab, S. and Mädche, A. (2000). Ontology Engineering beyond the Modeling of Concepts and Relations. *14th European Conference on Artificial Intelligence Workshop op Applications of Ontologies and Problem-Solving Methods*.

Stevens, R., Goble, C.A. and Bechhofer, S. (2000). Ontology-based Knowledge Representation for Bioinformatics. *Proceedings of ECAI-2000: The European Conference on Artificial Intelligence*, Amsterdam, IOS Press.

Stumme, G. and Mädche, A. (2001a). *FCA-Merge: Bottom-Up Merging of Ontologies*. Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Universität Karlsruhe. http://www.aifb.uni-karlsruhe.de/WBS/gst/presentations/2001-05-04-DBFusion.pdf. Date accessed: 17-11-2003.

Stumme, G. and Mädche, A. (2001b). Ontology Merging for Federated Ontologies on the Semantic Web. *IJCAI'01 Workshop on Ontologies and Information Sharing*.

Sugumaran, V. and Storey, V.C. (2002). “Ontologies for conceptual modeling: their creation, use, and management”. *Data & Knowledge Engineering* **42**: 251-271.

Takeda, H. and Nishida, T. (1998). *Some theoretical considerations on integration of ontologies*. Nara Institute of Science and Technology, Japan. 17p.

Tett, P. and Wilson, H. (2000). “From biogeochemical to ecological models of marine microplankton.” *Journal of Marine Systems*, **25**: 431-446.

Todorovski, L. and Džeroski, S. (2001). Using Domain Knowledge on Population Dynamics Modeling for Equation Discovery. *Proceedings of the Twelfth European Conference on Machine Learning*, Lecture Notes in Computer Science, Springer Verlag. pp 478-490.

Uchiyama, I., (2003), **“**MBGD: microbial genome database for comparative analysis”. *Nucleic Acids Research*, **31**(1): 58-62.

Villa, F. (2001). “Integrating modelling architecture: a declarative framework for multi-paradigm, multi-scale ecological modelling.” *Ecological Modelling*, **137**(1): 23-42.

Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M. and Shave, M.J.R. (1997). An analysis of ontology mismatches; heterogeneity versus interoperability. *AAAI Spring Symposium on Ontological Engineering*, Stanford University, California, USA.

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hübner, S. (2001). Ontology-Based Integration of Information A Survey of Existing Approaches. *Proceedings of IJCAI 2001 Workshop on Ontologies and Information Sharing*.

Weinstein, P.C. and Birmingham, W.P. (1999). Comparing concepts in differentiated ontologies. *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)*, Banff, Alberta, Canada.

Wiederhold, G. (1994). An algebra for ontology composition. *1994 Monterey Workshop on Formal Methods*, Monterey, CA.

Wittig, U. and De Beuckelaer, A. (2001). “Analysis and comparison of metabolic pathway databases”. *Briefings in Bioinformatics*, **2**(2): 126-142.

Internet resources

AllWords.com: http://www.allwords.com
American Type Culture Collection: http://www.atcc.org
AOS: http://www.fao.org/agris/aos/
Botany, University of Hawaii: http://www.botany.hawaii.edu/faculty/wong/BOT135/Lect24.htm
Centre for New Crop and Plant Products: http://www.hort.purdue.edu/newcrop/default.html
Chimaera: http://www.ksl.stanford.edu/software/chimaera/
DAML: http://www.daml.org
DOGMA: http://www.starlab.vub.ac.be/research/index.htm
Drosophila genome: http://flybase.bio.indiana.edu/
FIGIS: http://www.fao.org/figis
Gene Ontology Consortium: http://www.geneontology.org/
Geographic Ontology: http://ontology.buffalo.edu/bfo/GeO.pdf
Gramene: http://www.gramene.org
IMAR - Centro de Modelação Ecológica: http://tejo.dcea.fct.unl.pt/
Infobiogen: http://www.infobiogen.fr/services/dbcat
Instituto Ciencia de Animal: http://www.ica.inf.cu
Knowledge Network for Biocomplexity: http://knb.ecoinformatics.org/software/eml/
KSL: http://www.ksl.stanford.edu/software/chimaera/
MicroBial Genome Database: http://mbgd.genome.ad.jp/
OpenGALEN: http://www.opengalen.org
On-To-Knowledge: http://www.ontoknowledge.org
OntologyWorks: http://www.ontologyworks.com
OWL – Web Ontology Language: http://www.w3.org/2001/sw/WebOnt/
Plant Ontology Consortium: http://www.plantontology.org
PrometheusDB: http://www.prometheusDB.org
Protégé (and PROMPT): http://protege.stanford.edu/
SEEK: http://seek.ecoinformatics.org
SEmantic MEta DAtabase: http://www-bm.ipk-gatersleben.de/semeda/login.jsp
Semantic Web: http://www.w3.org/2001/sw/
Silsoe Research Institute (SRI): http://www.sri.bbsrc.ac.uk/science/bmag/itagr.htm
SNOMED: http://www.snomed.org
TAMBIS: http://imgproj.cs.man.ac.uk/tambis/index.html
Van Dale Woordenboeken: http://www.vandale.nl
WonderTools: http://www.swi.psy.uva.nl/wondertools/
WonderWeb: http://wonderweb.semanticweb.org/

Appendix A – Modeling paradigms

A-1: Questions before 'translating' to ORM

1. To confirm: are the open triangles `isA` (/subtype of) relationships?
   A: yes.
2. To confirm: The diamond is an aggregate?
   A: yes.
3. To confirm: In addition, what is the dotted line with filled triangle/arrow, standard relationship?
   A: yes.
4. What is the relation between `State Group` and `Structure`?
   A: Call that `Applies to` as well.
5. Why are structure, property and state in capitals and bold, because of the Description Element information preceding the paragraph?
   A: yes, because they are the most important.
6. So Type is an attribute of `Structure`, but what exactly do you mean with "Type: Type Term"? That the `Type` attribute is of data type `Type Term`? If so, what is that supposed to mean? Does it have to do with the subclass `Type` that is drawn below `Structure`, or has it to do with the `Defined Term`, or some combination of the two?
   A: yes, data type of `Type` is `Type Term`, and the subtype `Type` of `Structure` may be better named as `Type Term`. This `Type` cannot be part of the part of relationship of the `Structure` object/class, therefore separate in an `isA` hierarchy.
7. If states are "composed into groups", that means that the diamond has to be on the other side of the line – that is, if 2 is yes.
   A: Whatever.
8. Groups of states doe not affect some property? Ever?
   A: not according to the plant taxonomists.
9. The text in the figure description mentions "these state groups may represent 'de facto' properties". What is exactly the difference? If they are 'de facto' the same, then representing them as different is incorrect one way or another: either they are different after all or they are the same and the modelers & SMEs could not agree. May be both. When the former: the representations means something different, because the upper 'route' says it can be that '*one [or more?] state describes one [or more?] property and applies to one [or more] structure*'. Whereas the other [bottom] route says, taking the aggregate sign into account,: '*one [or more] state is grouped and this group of states has some relationship with one [or more] structure* '
   A: The two semantic 'routes' could be the same thing, but not always. Computer scientist's view: if the plant taxonomists would use the Properties accurately, then it would exactly be the same. Ultimately, one of the two representations will be removed, after testing with real data.
10. How do you get a `State Group`? What defines a `State Group` and what are/can be the differentiae?
    A: The taxonomists define it, but there is no regularity in that according to the computer scientist.
11. This relationship with `Property` subproperty `Property` is nowhere discussed in the article. What is it about exactly?
    A: that is how the computer scientist thinks how one can represent the `State Groups` the taxonomists see, but then to view them as [better organized] `Properties`. However, this is not tested yet. Further, there is a hierarchy of properties, but do not represent them separately, only as one list of properties that have something to do with each other.

12. Where are the three types of modifiers (`Relative`, `Spatial` and `Temporal`) in the figure? Nowhere! Why absent?
    A: It's a summary diagram, and probably will not be used later, or as a very minor aspect, in the database anyway. Will be stored as free text.

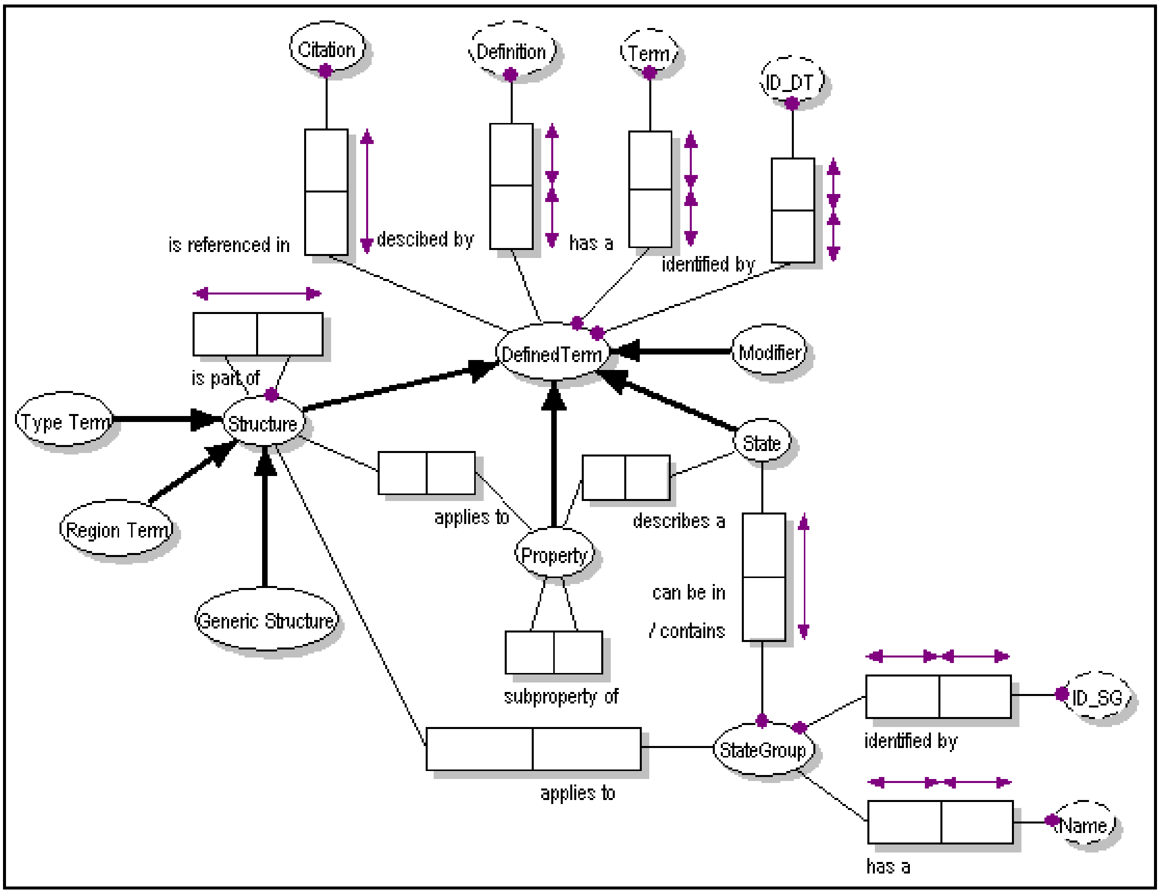


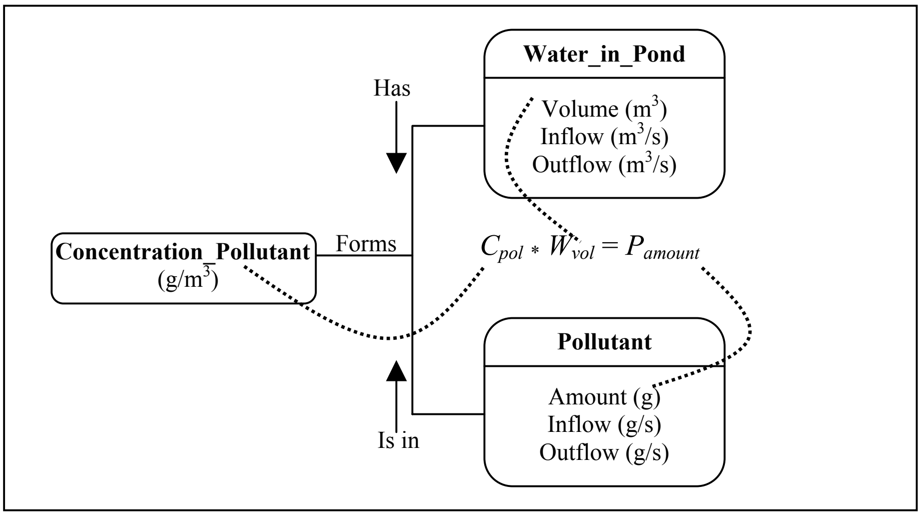
Figure A-1. First ORM model of the descriptive term ontology. Note the ambiguity of the relationships between State*,* Property*,* Structure *and* StateGroup *and added clarity surrounding the attributes.*

A-2: Questions that surfaced when drawing ORM

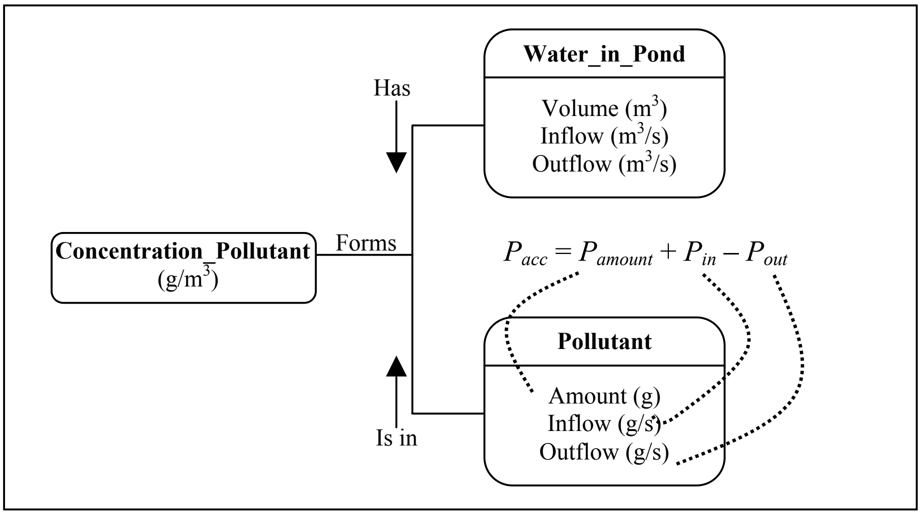
13. Does every `Defined Term` have a `Term` name? This is represented like that now. **A**: Yes.
14. And a full `Definition`? Also with only one description per term (ID)? Now represented as that there can be zero or one description for each defined term (in case there are terms but no info, or too lazy to enter the values), and for each defined term, one may record only on `Defined Term`, i.e. homonyms are ruled out.
    A: Yes, ideally, they would have full definition, but is not always available, nor known. You can have the same definition, but have another `Term` name. Each combination of `Term` and `Definition` is unique, but not formally, as this is identified by the ID.
15. And per definition linked to 'zero or more', 'exactly one' or 'one or more' `Citation`? Currently represented as 'zero or more', and that for each citation, there may be one or more defined terms recorded. Is that right?
    A: Each `DefinedTerm` can have only one `Definition`; there should be exactly one, but it is allowed to have none. There's either a `Citation`, or an `Author` or both for each `DefinedTerm`.
16. I assume that `Citation` is a concept in itself, and further defined with types like author, title and so forth.
    A: Yes. For things that you cannot find citations for, the `DefinedTerm` can have an `Author`, but no publication information related to it, so `Author` has to be added as an attribute to `DefinedTerm`. Some can have an `Image` as well.

17. How can a "Generic Structure" be a *subtype* of a structure? That sounds like if it should be the *supertype* or something.
    A: No, they are generic in the sense that they are occurring in different structural contexts, such as pores and hairs (the ones that do not make sense in the hierarchical network structure), but a leaf is always structurally related to a stem.
18. The recursive relationship of `Structure`: I presume that is optional and that if a structure is part of another structure, this is one or more, so overall 'zero or more'? Or is a `Structure` *always* composed of other structures? I represented the first version now. It reads "For each `Structure` s1 there may be zero or more `Structure` s2 recorded" and "For the fact 'Structure s1 is part of Structure s2' How many instances of 'Structure s1' may be recorded for each instance of 'Structure s2'?", which I answered with 'one or more'. Don't know if this is right.
    A: Any `Structure` is always part of at least one other `Structure`, except the entire plant, which is called 'root' here. The `Structure` relationship part of are all optional until you instantiate them. One is *always* part of another `Structure`, but not always the same `Structure` that hierarchically precedes it one 'step' in the hierarchy, but may NOT 'skip' one, but have to create a new part of relationship. Currently it is assumed to be have included all these kind of part of relationships already, and not newly created anymore.
19. On the `State Group` aspects: each state group has 'zero or one' name and there is only one name per state group, is that right?
    A: Has *exactly one* name at the moment. – later questioning it again: lots of different state groups can have the same name
20. Each `State Group` has one or more `States` (would be weird if there were no states in the state group) and each `State` can be in zero or more `State Group`. Right?
    A: Yes regarding the first. On the second: the computer scientist thinks so. `State Group` have some sort of subtyping according to the taxonomists, but not according to the computer scientist; they tried to convince the taxonomists that the subtyping with the `Properties` is 'the same' (read: better structured) than the types of `State Group`.
21. I left out the three types of modifiers. **A**: ok
22. There are no rules relating the `State Group` to `Structure`. What is it? Does each `State Group` apply to *one* `Structure`? Left empty in the figure.
    A: Each state group can apply to more than one structure and more than one structure can apply to one state group. There is at least on structure for each state group, a structure not necessarily has a state group.
23. Do `Property`, `Type Term`, `Region` and `Generic Structure` not have some attributes, even if it were a mere string or something, or an ID to identify them?
    A: Each one has the same attributes as the `DefinedTerm` one, obtained via inheritance. `Structure` has the attribute `Type`, which should be in the picture as well – somehow.
24. The `Property` subproperty of `Property`, I draw a blank here: can a property be a sub property of more than one property? Is there a definite hierarchy in properties, of all of them? And what are their differentiae? The fact is left empty for the time being.
    A: Yes there is a definite hierarchy in properties, all are organized like that. They can only have one parent.
25. The `Property` applies to `Structure` has pretty much the same vagaries as mentioned under point 22, and left empty. Each `Property` applies to how many `Structure`? For each `Structure` how many `Property` may be recorded?
    A: Multiple either way, m:n. There are quantitative and qualitative properties; it is the qualitative that cause the problem. The taxonomists' interpretation is that the quantitative 'properties' do not have any relation with structure, whereas the computer scientist says there is.
26. I assume it is the `State` that describes a `Property`, say, a one-to-one relationship? Nothing represented either way at the moment though.
    A: There can be lots of states (combined) describing one property. You can have properties that do not have states but values when you use them (like length – the quantitative ones).
27. The relationships with their constraints between state-property-structure, stategroup-structure and property-property really need to be cleared up.
    A: ok, done, see previous answers.

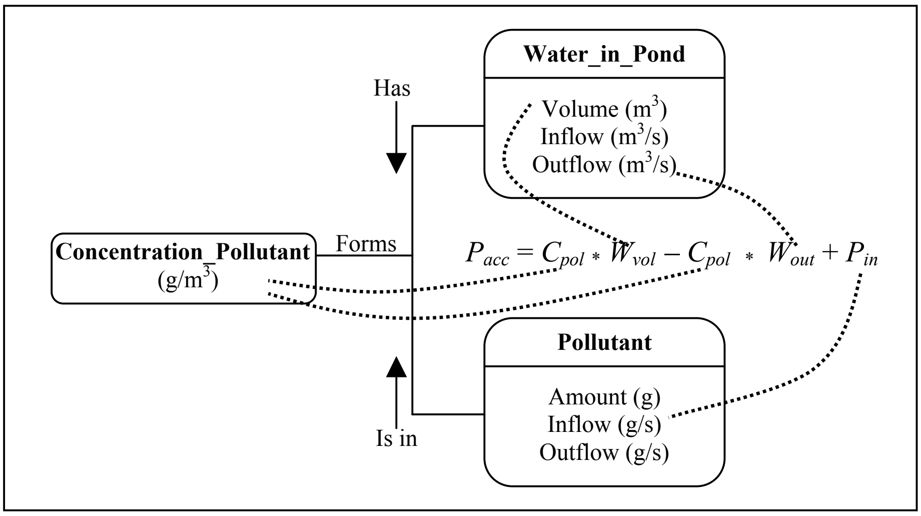
Appendix B – Extended semantics for equations



B-1. Calculation for the amount of pollutant in the pond.



B-2. Calculation for the accumulation of pollutant in the pond.



B-3. Substitution of P_{out} from Figure 4.8 and P_{amount} from B-1 into B-2.

Appendix C – Definitions of ontology integration

Table C-1. Summary of different interpretations on 'integrating' ontologies.

Type of 'Integration'	Description	References	Comments
Unification	"Everything that can be done with one can be done in an exactly equivalent way with the other"	Sowa, paraphrased in Gangemi *et al* (1998) and quoted in Pinto *et al* (1999)	Sounds like the same as Pinto's merging. Also called 'total compatibility' according to Gangemi *et al* (1998)
		Mitra *et al* (1999)	See their 'merging' entry
Merging	"Combining different ontologies with the same subject domain and creating a unified ontology"	Pinto *et al* (1999)	Is like Sowa's unification; see *Appendix D-2*
	"…product of this merge will be, at the very least, the intersection of the two given ontologies" and "…the engineer is in charge of making decisions that will affect the merging."	Kalfoglou and Schorlemmer (2002)	The author's impression is, that the reference distinguishes 'merging' from mapping in that the former has human intervention
	Seems to have much more 'matches', hence larger intersection and the intention is to create 'monolithic knowledge base'	Mitra *et al* (1999)	Uses it synonymously with unification, and the authors consider the 'monolithic knowledge base' as unattainable
	"The process …takes as input two (or more) source ontologies and returns a merged ontology based on the give source ontologies"	Stumme and Mädche (2001b)	It does not mention if the ontologies have the same/similar/complementary/orthogonal subject
Mapping	"sound and complete"	Akahani *et al* (2002)	See *Figure 4.2a*
	"Set of formulae that provide the semantic relationships between the concepts in the models"	Madhavan *et al* (2002)	See also 'helper model'
	Finding the corresponding semantic concepts in the ontologies that are to be integrated	Doan *et al* (2002)	See also Doan's 'matching'
	Local-centric	Calvanese *et al* (2001a)	See *Appendix E*
	Global-centric	Calvanese *et al* (2001a)	See *Appendix E*
	Global- and local-centric, but then "loosely" –sound, complete and exact	Calvanese *et al* (2001b)	See *Appendix E*
	With logic infomorphisms. Take two local ontologies populated with instances, have a reference ontology and 'place' the local one onto the reference one to create a global ontology	Kalfoglou and Schorlemmer (2002)	See also *Appendix F-1*
Matching	The correspondence between individual concepts of the two ontologies, found automatically (without human intervention)	Bernstein *et al* (2000)	Sounds like the same as Doan *et al* (2002)

	" the ontology-matching problem is to find semantic mappings between them"	Doan *et al* (2002)	The authors use mapping and matching almost interchangeably, though this author has the impression that matching is used for the automated part and mapping the overall process, including the human intervention.
	Finding the correspondence of terms, based on a set of rules	Mitra *et al* (1999)	Uses/creates the intersection
	"the identification of maximal one-to-one correspondences between elements [concept or relation oredge] of the compared definitions [between request and recommendation ontology]… Matchings enable analysis of similarities and differences between the concepts to predict their semantic compatibility."	Weinstein and Birmingham (1999)	There are subdivisions according to the types of matches: inherited, specialized and serendipitous. See reference for more detail.
Approximate Ontology Translation	'sound and complete' mapping operators, but also 'sound and complete' when using specialization and generalization operators	Akahani *et al* (2002)	See *Figure 4.2b*, the 'specialization and generalization' is what Mena (1996) calls hyponyms and hypernyms
Translation		Akahani *et al* (2002)	See 'approximate ontology translation'
		Kalfoglou and Schorlemmer (2002), Chalupsky (2000), some others	Also used in relation to converting between different syntactical representations of an ontology – which others consider 'preprocessing'.
Partial compatibility	"[A]ny inference or computation that can be expressed in one ontology using only the aligned concepts and relations can be translated to an equivalent inference or computation in the other ontology."	Sowa quoted in Pinto *et al* (1999)	Gangemi *et al* (1998) adds that there may be difficulties to prevent full unification
Alignment	"Mapping of concepts and relations between multiple ontologies based on preservation of the partial ordering and synonyms, as well as the possible introduction of new concepts that will function as sub- or supertypes"	Sowa in Pinto *et al* (1999)	Gangemi *et al* (1998) adds: "it is useful for classifications and information retrieval, but it does not support deep inferences and computations"
Mapping ontology	Ontology O_M contains the rules that map concepts between ontologies O_1 and O_2,	Hefflin and Hendler (2000)	See *Figure 4.4*
Mapping revisions	Where O_1 contains rules that map O_2 objects to O_1 terminology and vice versa	Hefflin and Hendler (2000)	See *Figure 4.4*
Intersection ontology	Ontology O_N is created containing the intersection of concepts between O_1 and O_2 and rename terms where necessary	Hefflin and Hendler (2000)	See *Figure 4.4*
Single ontology	One global ontology with all concepts of the local ontologies	Wache *et al* (2001)	See *Figure 4.5*
Multiple ontologies	No global ontology, only local ontologies with each other	Wache *et al* (2001)	See *Figure 4.5*
Hybrid ontology	Global backbone ontology with the main concepts, local ones with the details	Wache *et al* (2001)	See *Figure 4.5*
Integration	"When building a new ontology reusing other available ontologies of different subject domains"	Pinto *et al* (1999)	See main text and *Appendix D-1*

	"The process of finding commonalities between two different ontologies A and B and deriving a new ontology C that facilitates interoperability between computer systems that are based on the A and B ontologies. The new ontology C may replace A or B, or it may be used only as an intermediary between a system based on A and a system based on B. Depending on the amount of change necessary to derive C from A and B, different levels of integration can be distinguished: alignment, partial compatibility and unification"	Sowa in Gangemi *et al* (1998)	See the entries 'alignment', 'partial compatibility' and 'unification'. The definition itself is rather broadly formulated.
		Mena *et al* (1996)	See *Figure 4.1*, this is alike Pinto's merging and Sowa's unification. Note also that this is the same as Akahani's ideal way of mapping
	"Combination aspect connects heterogeneous aspects in which aspect theories are simply merged"	Takeda and Nishida (1998)	The authors refer to orthogonal ontologies similar to Pinto's merging, but also add it is a *constructive* approach
	"Category aspect connects homogeneous aspects in which aspect theories are connected with possibility modality"	Takeda and Nishida (1998)	Relating to combining ontologies of the same subject domain like the unification above, adding that it is the *teleological* approach (integration arisen as hypothesis)
Generic integration	Based on generic relations that "an aspect is dependent on other aspects originally"	Takeda and Nishida (1998)	The more general interpretation of combining ontology
Coincidental integration	Results form engineering/design activities: new inventions ("Creative design") generate new relations between previously unrelated concepts, hence between different ontologies.	Takeda and Nishida (1998)	E.g. a 'screw' from form a structural ontology that also may function as a stopper related to the linear movement from kinematics concepts
Federated ontologies	Distributed, 'connected' ontologies, somewhat analogous to federated databases, although the intention is to merge	Stumme and Mädche (2001a, 2001b)	See *Figure H-1* and the discussion in chapter 5.
Use of multiple ontologies	To build software applications	Pinto *et al* (1999)	The idea is to not change anything, see *Appendix D-3*
		Gangemi *et al* (2002a)	See *Figure G-1*
Total compatibility		Gangemi *et al* (1998)	See 'unification'
Helper model	For the mapping, in order to accommodate additional requirements to achieve the mapping between two ontologies. "…needed in cases where it is not possible to map directly between a pair of models"	Madhavan *et al* (2002)	See also mapping
Extending	Adding the second ontology as an extra 'branch' to the main ontology	Marjomaa (2002)	Marjomaa did not provide a clear definition. Interpreted as represented in *Figure 4.3*
Incremental loading		Gangemi *et al* (2002a)	See extending and *Figure 4.3*
Ontology sharing		Kent (2000)	Using the infomorphisms. See reference and *Appendix F-3*

Appendix D – Integrating ontologies

Graphical representations of Pinto *et al* (1999)'s definitions on integration, merging and using ontologies.

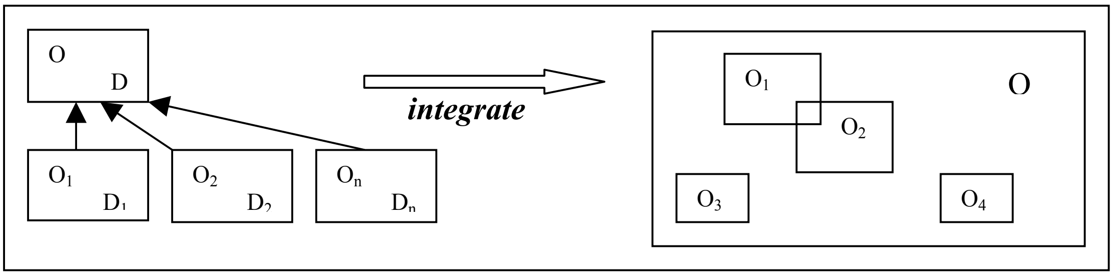


Figure D-1. Ontology integration. D= domain; O= ontology.

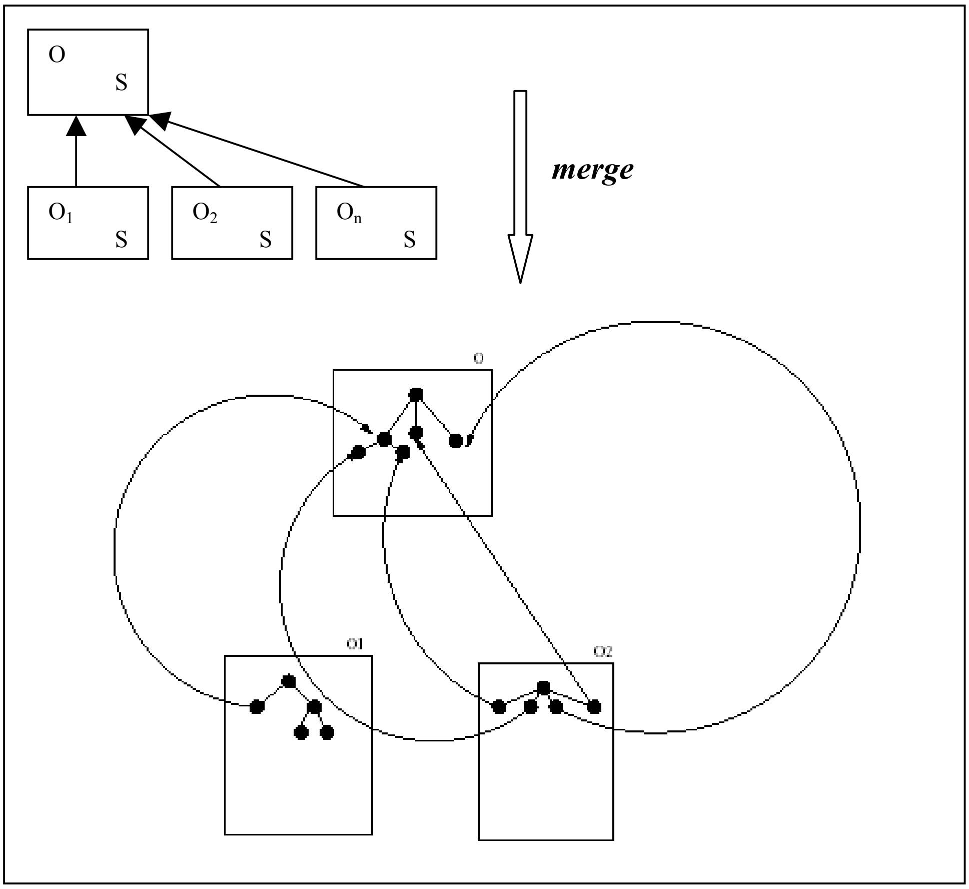


Figure D-2. Merging. S=subject domain (which are the same); O= ontology.

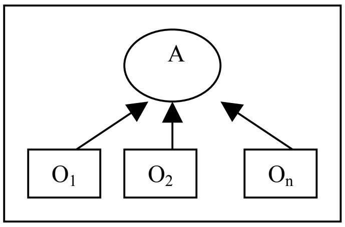
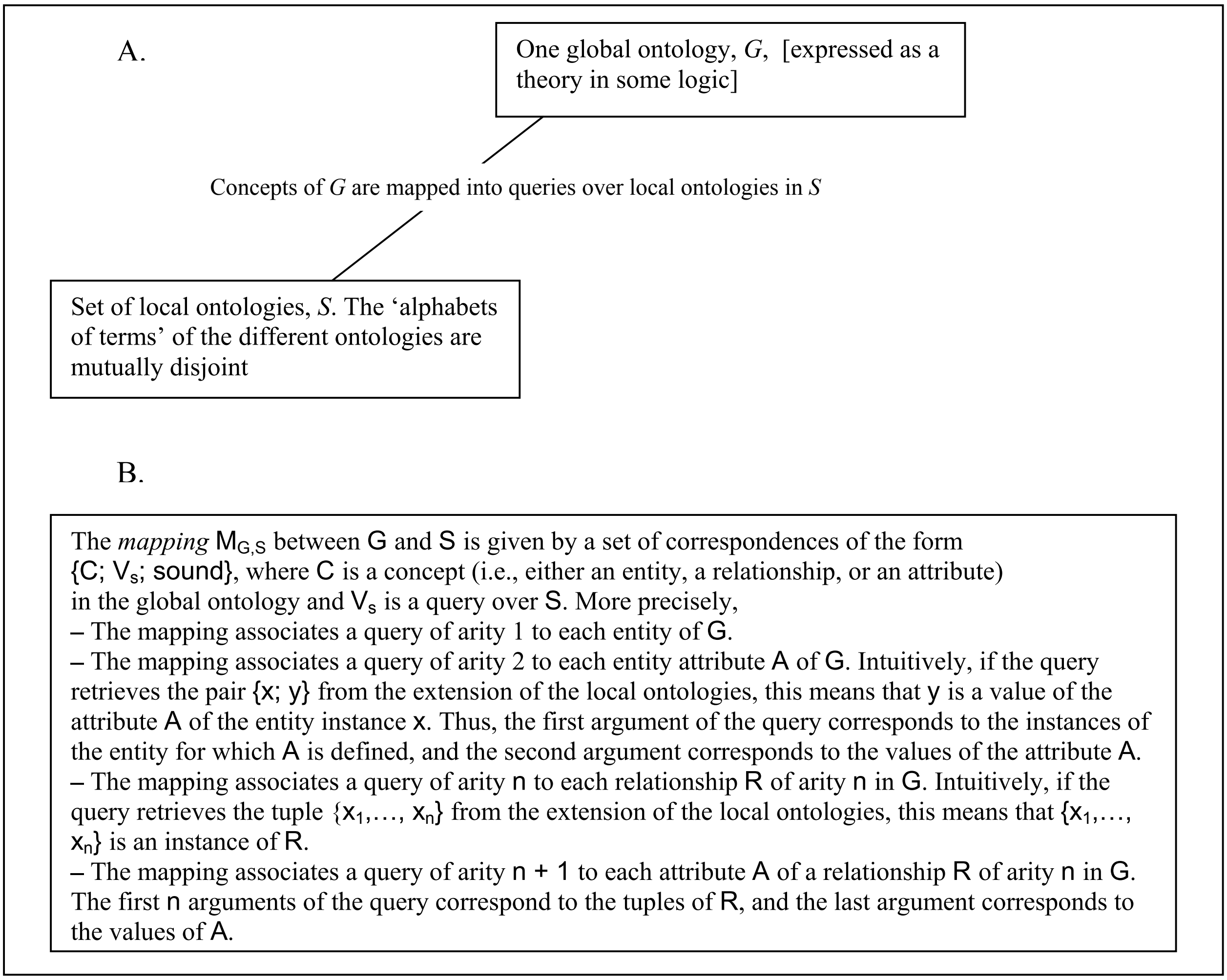


Figure D-3. Using an ontology for an application.

Appendix E – Integration via queries

Calvanese *et al* (2001)'s local and global query integration.



A.

One global ontology, *G*, [expressed as a theory in some logic]

Concepts of *G* are mapped into queries over local ontologies in *S*

Set of local ontologies, *S*. The 'alphabets of terms' of the different ontologies are mutually disjoint

B.

The *mapping* $M_{G,S}$ between G and S is given by a set of correspondences of the form {C; V_s; sound}, where C is a concept (i.e., either an entity, a relationship, or an attribute) in the global ontology and V_s is a query over S. More precisely,

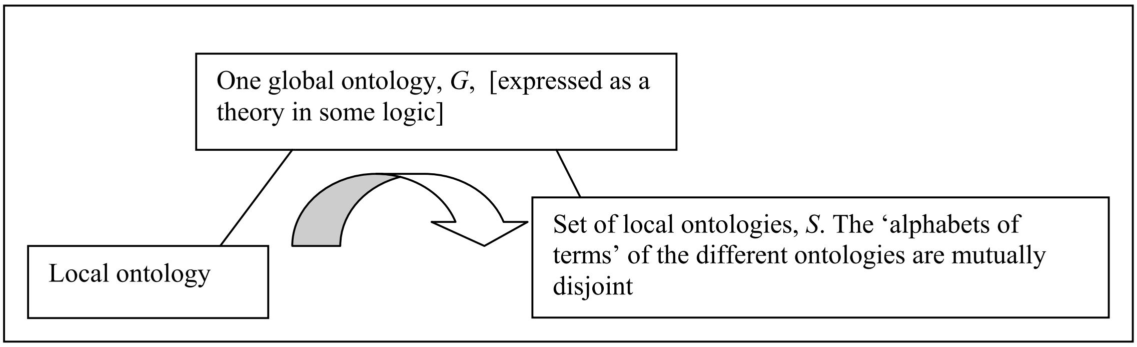
– The mapping associates a query of arity 1 to each entity of G.

– The mapping associates a query of arity 2 to each entity attribute A of G. Intuitively, if the query retrieves the pair {x; y} from the extension of the local ontologies, this means that y is a value of the attribute A of the entity instance x. Thus, the first argument of the query corresponds to the instances of the entity for which A is defined, and the second argument corresponds to the values of the attribute A.

– The mapping associates a query of arity n to each relationship R of arity n in G. Intuitively, if the query retrieves the tuple {x_1,…, x_n} from the extension of the local ontologies, this means that {x_1,…, x_n} is an instance of R.

– The mapping associates a query of arity n + 1 to each attribute A of a relationship R of arity n in G. The first n arguments of the query correspond to the tuples of R, and the last argument corresponds to the values of A.

Figure E-1. Global-centric integration. C-1a: graphically; C-1b the rules for the actual mapping



One global ontology, *G*, [expressed as a theory in some logic]

Local ontology

Set of local ontologies, *S*. The 'alphabets of terms' of the different ontologies are mutually disjoint

Figure E-2. Local-centric integration (rules omitted): querying the other local ontologies through the global ontology

Appendix F – Mapping with infomorphisms

Mapping with infomorphisms, according to Kalfoglou and Schorlemmer (2002).

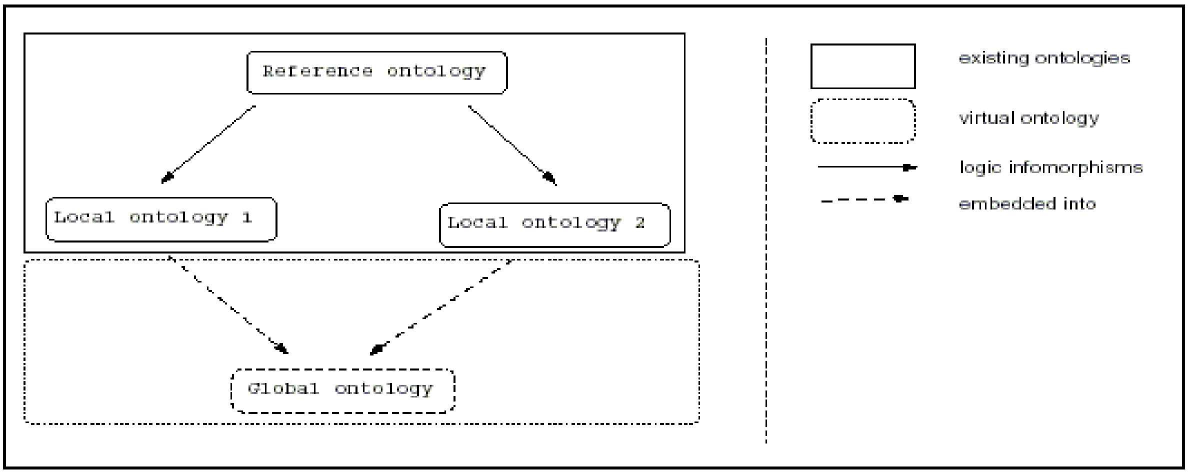
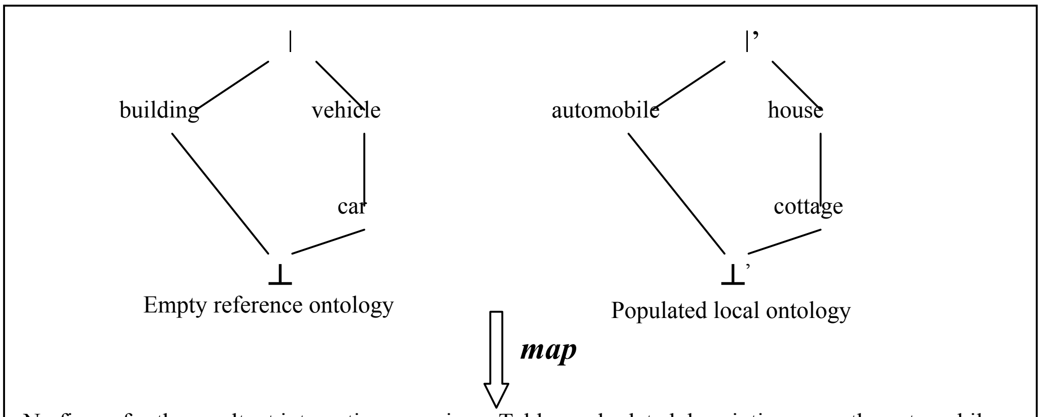


Figure F-1. Scenario for ontology mapping.



No figure for the resultant integration was given. Tables and related descriptions map the automobile instances from the local ontology into both vehicle and car and the houses (and implicitly cottages) into building. The example does not indicate how to integrate a second local ontology: is that sequential, hence with a non-empty reference ontology, or does this occur concurrently?

Figure F-2. Example of a mapping via information flow.

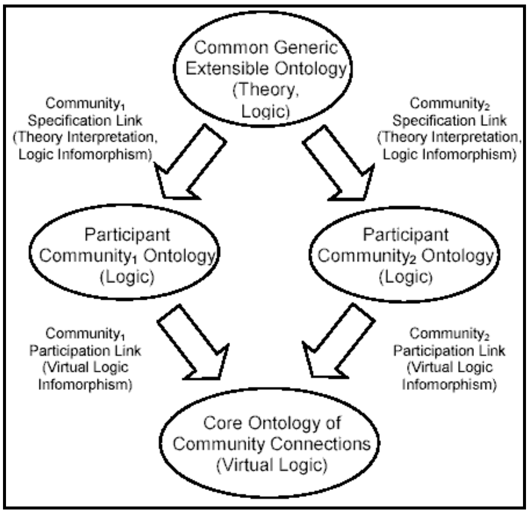

Common Generic Extensible Ontology (Theory, Logic)
Community₁ Specification Link (Theory Interpretation, Logic Infomorphism)
Community₂ Specification Link (Theory Interpretation, Logic Infomorphism)
Participant Community₁ Ontology (Logic)
Participant Community₂ Ontology (Logic)
Community₁ Participation Link (Virtual Logic Infomorphism)
Community₂ Participation Link (Virtual Logic Infomorphism)
Core Ontology of Community Connections (Virtual Logic)

Figure F-3. Ontology sharing between communities (Source: Kent, 2000).

Appendix G – (Re)Use of ontologies

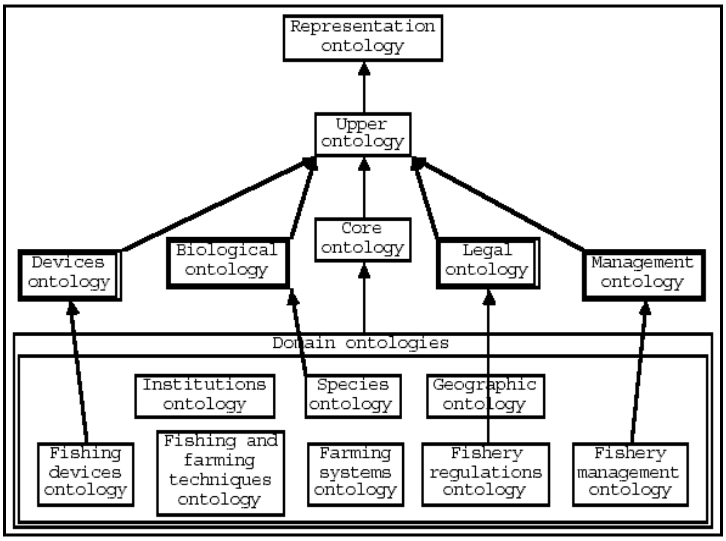


Figure G-1. Architecture of the fishery ontology library; double frames mean use of external ontologies. (Source: Gangemi *et al*, 2002a)

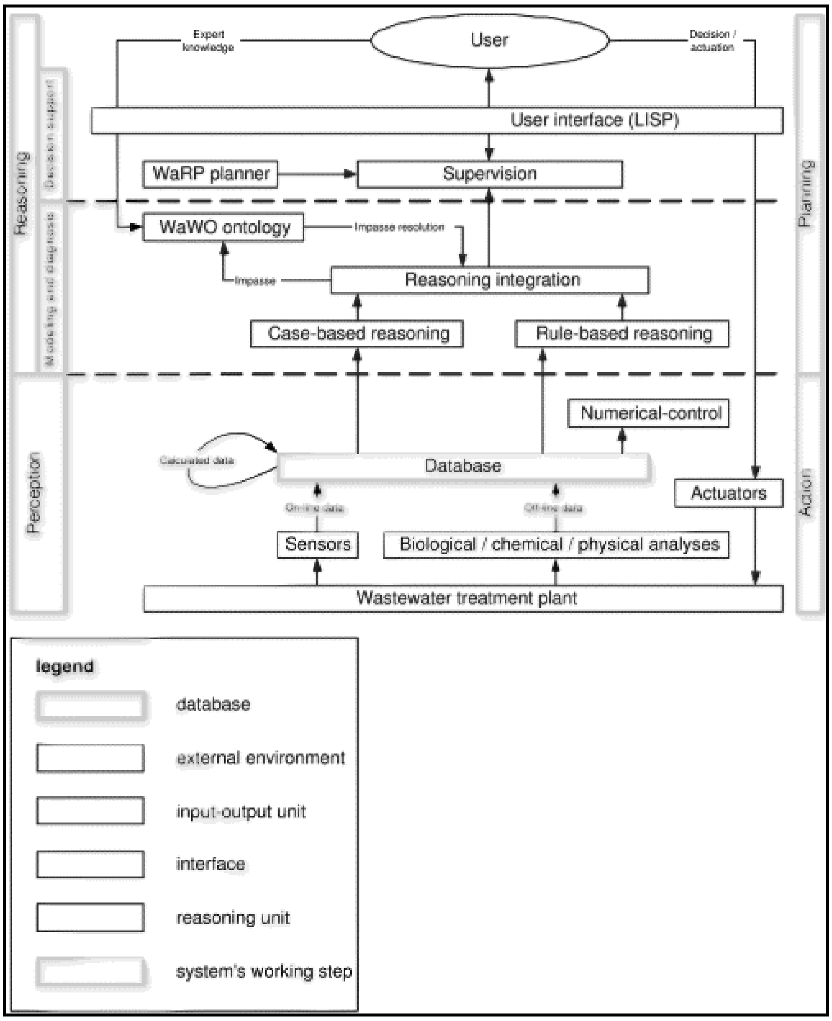


Figure G-2. The OntoWEDSS architecture with the WaWO ontology tightly integrated with the overall structure of the software: AI's case and rule based reasoning systems, database, input devices, and user interface. (Source: Ceccaroni *et al*, 2004)

Appendix H – Federated ontologies

A combination of reuse and merging of ontologies.

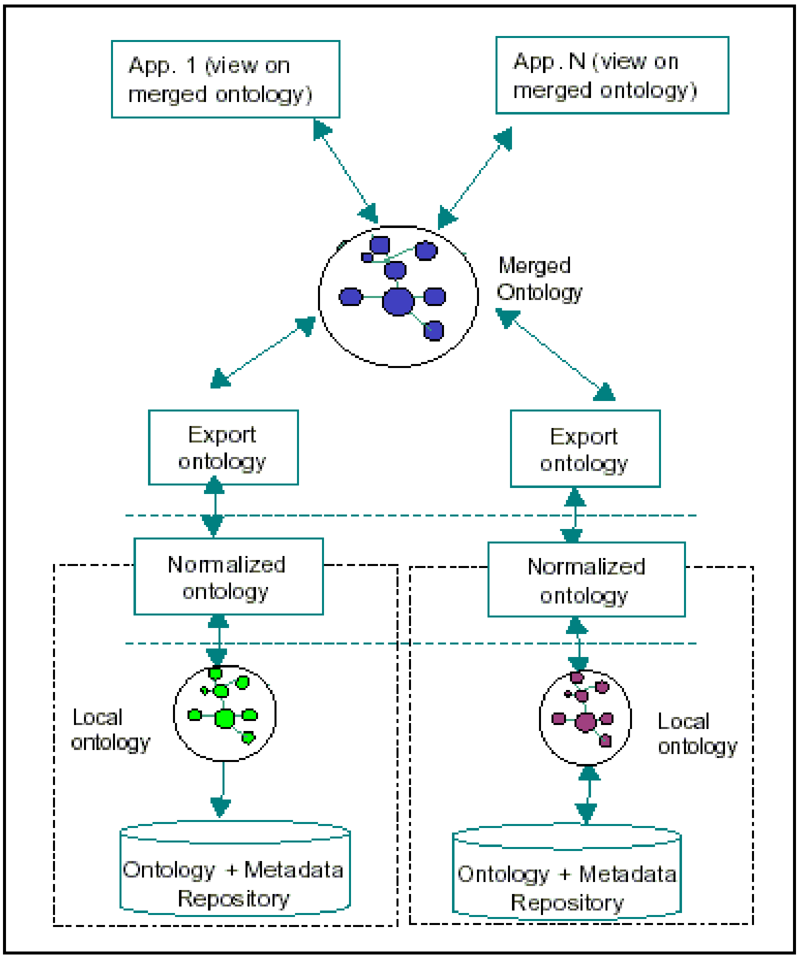

App. 1 (view on merged ontology)
App. N (view on merged ontology)
Merged Ontology
Export ontology
Export ontology
Normalized ontology
Normalized ontology
Local ontology
Local ontology
Ontology + Metadata Repository
Ontology + Metadata Repository

Figure H-1. Federated ontologies. (Source: Stumme and Mädche, 2001a)