FRANK LOEBE

ORGANIZATION AND MANAGEMENT OF LARGE CATEGORICAL SYSTEMS

Abstract. This chapter surveys approaches to handling categorical systems of extensive size, spanning from semi-formal systems in terminology and classification sciences to formal logical approaches. In particular, we briefly review the transition from terminologies to ontologies that are formalized in logics, exemplified in the medical domain. The main part presents the state of the art of the modularization of logical theories with an ontology-related background. Since the field is still very young and active, an evaluation of these approaches results in a heterogeneous landscape of proposals and leaves perspectives for future research.

1 INTRODUCTION

Ontologies grow large in size if systems are considered that cover a very complex domain like medicine, or a number of domains possibly together with upper levels of categories. Especially for philosophical investigations of basic categories, the ontological systems studied in the past typically comprise only a handful of basic categories, as in the works of Aristotle, Kant, Brentano, Husserl, and many others, see [Sow 2000, sect. 2.2] for an overview in the context of computer science. A proper analogy holds for logic, considering the limited size of axiomatic systems formerly under examination. Manual logical investigations were usually concerned with studying the consequences of a few axioms, a task hard enough itself. Trying to model real-world problems with logic, even when equipped with automated reasoning as currently available, one easily faces tremendously large signatures and theories which require novel solutions. A few numbers may provide some intuition on the current scale of the problems. As a self-contained system, the leading clinical healthcare terminology system SNOMED CT is implemented in a logical formalism and comprises more than 311,000 concepts (with formal definitions) as of January 2008[1]. On this scale of even a single system, new methods are required for applying formalisms and using available technologies. The situation is even more complex in the context of the Semantic Web effort [Ber 2001]. [Hen 2007, p. 823] qualifies its major representation languages RDF, RDFS [W3C 2004a] and OWL [W3C 2004b] as “the most widely used KR [knowledge representation] languages in history” and states that “[a] web search performed around the beginning of 2007 finds millions of RDF and RDFS documents, and tens of thousands of OWL ontologies.” Many of those OWL ontologies complement one another regarding the domains which they cover. Consequently, the use of several such ontologies as components of a joint, larger system is an immediate idea. However, the first detailed attempts to elaborate this idea have discovered many unresolved issues and choices for solutions.

In this chapter we survey approaches to tackle the complexity of huge categorical systems and describe the state of the art from two perspectives: (1) the internal organization and use of categorical systems, and (2) modularization issues of logical formalisms. Put more generally, (1) focuses on a single-category perspective, i.e., it is concerned with means for specifying or searching single categories in the system. In contrast, (2) involves a perspective of the system as a whole, as typically held during the construction and maintenance of categorical systems. For (1), we start from precursors of current formal ontologies by reviewing work in terminology and classification sciences, describing the move toward the use of logical approaches and their benefits. The main part is then concerned with aspect (2) and discusses modularization universally as well as applied to formal logical languages. Since modularization as a field is still emerging, this part sets up a terminology of modules and module characteristics in order to describe and evaluate the current branches of research to some extent uniformly.

The present chapter assumes a very broad notion of ontology. It includes all kinds of concept systems, i.e., systems aiming at specifying a collection of concepts or notions, which themselves are referred to in other information systems. Hence, natural language texts serve as presentations of ontologies just as other types of systems, ranging by increasing degree of formality from vocabularies, glossaries, and thesauri over terminologies or conceptual models up to formal axiomatizations, to name just a few types [Gom 2004, ch. 1]. In a sense, our scope is similar to that in terminology management [Wri 1997], but with less consideration of representational or linguistic aspects like the naming of concepts, cf. also [Kei 2000b].

Terminological and ontological aspects are relevant to all domains, and terminological efforts in the form of determining a common language arise immediately in most fields of science and engineering. In the next section, we concentrate on the development of concept systems in medicine as a representative of domains with high complexity.

2 TERMINOLOGICAL SYSTEMS IN MEDICINE

The development of common vocabularies and classification systems in medicine started very early, exemplified by the introduction of the International Statistical Classification of Diseases (ICD) in 1893 [WHO 2004]. Concept systems in medicine grow very large as soon as a broad coverage of subdomains is to be combined with even a medium depth of subsumption hierarchies.[2] Medical terminology management has thus developed from mere collections of terms (and in some cases definitions) in the form of vocabularies, glossaries, classifications, code lists, nomenclatures, and others via semi-formal models to logic-based approaches, cf. [Her 2009b] in this volume or [Kei 2000a,b]. We briefly review major aspects in the course of that development in order to identify solved and unsolved problems concerning the complexity of these systems. Nevertheless, this will ignore most of the overwhelming number of problems relating to clinical terminology [Rec 1999, Cim 1998], many of which still remain open.

The move toward formal systems is primarily driven by two partially interrelated aspects, which are often intermingled in the literature: (1) *compositionality* and (2) the *structured arrangement of concepts*.[3] Both are motivated by mitigating the combinatorially exploding number of concepts in medicine. For instance, given the notions of fracture as well as of bone and bone subtypes, there are fractures of many kinds, like fractures of the arm, the forearm, the ulna, the radius, the femur, etc., which need to be available e.g. for electronic health records.

2.1 Compositionality

Looking at aspect (1), semi-formal compositional concept systems try to reduce the representational complexity by providing a number of atomic concepts, e.g., fracture and arm, forearm, ulna, and radius. *Compositionality* means to express certain concepts in terms of combinations of others, starting from atomic concepts. Expressions of the form “(fracture, ulna)” are used for representing the notion of “fracture of the ulna”. The number of explicitly managed concepts in a compositional setting can be greatly reduced to the number of atomic concepts. Compositionality is in this context also called *post-coordination* and contrasted with *pre-coordination*, cf. [Rec 1999, p. 246]. Pre-coordinated terminological systems are constructed as pre-established enumerations of concepts, listing all available concepts explicitly, whereas post-coordinated systems allow for forming implicit concepts from explicitly given constituents. Both approaches usually aim at a degree of coverage as high as possible, and at internal consistency of the resulting concept system.

Two new problems arise for compositional systems: (a) meaningless combinations of concepts and (b) the need to detect equivalent combinations. An example for (a) is (fracture, oral cavity), which is not reasonable because cavities cannot break. The pairs (fracture, ulna) and (ulna, fracture) illustrate problem (b), if they are understood to express the same concept. Both problems were tackled by the use of description logics [Baa 2003b] starting in the 1990s, for instance, in GALEN [Rog 2001, Rec 2005] and SNOMED CT [Igg 2001]. In particular, equivalence and subsumption checking of concepts are common reasoning problems in description logic [Baa 2003b, sect. 2.2.4] which address (b). Problem (a) of meaningless concept definitions can only partially be supported by a description logic formalism. For this purpose, concepts and relations among them must be expressed appropriately and need to be augmented by restrictions. For example, (fracture, ulna) could be represented adequately as $\mathsf{fracture} \sqcap \exists \mathsf{has\text{-}location.ulna}$. Description logics can further be used to detect inconsistencies among logical restrictions, i.e., they support ensuring consistency. Moreover, less common reasoning tasks like determining the least common subsumer of a given set of concepts or the most specific concept of an instance can help in defining new concepts, cf. [Baa 2003b, sect. 1.6 and 6.3]. Recently, rather weakly expressive description logics – called the EL family – are gaining much attention, including their theoretical foundations [Baa 2003a, Bra 2004, Baa 2005, Kri 2007] as well as efficient reasoners [Baa 2006]. One major reason for this development is the (re)discovery that the limited expressivity is already beneficial for large medical terminologies like GALEN and SNOMED CT, while more expressive logics do not yet scale to systems of this size.[4]

Although meanwhile superseded by description logics, it is informative to look at earlier intermediate solutions to the problem of arbitrary combinations, namely the use of *multi-dimensional* or multi-axial concept hierarchies introduced in the late seventies, cf. [Spa 1998]. The restriction to combine only concepts from different dimensions like anatomic site, morphology, or etiology avoids insensible combinations along one and the same axis, e.g. (ulna, femur). Since such restrictions remain insufficient, the approach has been refined to *multi-focal* models, cf. [Str 2002]. Here, not all dimensions are treated equally, but certain dimensions appear – possibly constrained in their values – attached to specific values at another dimension, where a value spanning its own field of dimensions is called a *focus*. For instance, the value “fracture” on a “disease” dimension may be a focus with a “location” dimension constrained to “bones”.[5]

2.2 Navigation

Although a number of systems based on multi-dimensional or multi-focal models are currently found in practice (e.g. LOINC [For 1996, McD 2003]), description logics prevail with regard to their combinatorial capabilities and the validation of the consistency of restrictions.

Nevertheless, those former approaches have an interesting effect concerning the organization of subsumption hierarchies, aspect (2) from above. Subsumption reasoning in description logics allows for inferring formally consistent subsumption lattices based on concept definitions. However, the number of explicitly introduced elements in these lattices usually becomes very large, due to naming composed concepts in order to reuse them without recomposing them in each case of use. In spite of their inferential potential, current logical approaches and tools do not yet offer support to tackle the *navigation problem*, i.e., the problem of comprehending and orienting creators and users of such lattices.[6] The availability of corresponding methodological advices is equally limited. Alan Rector is among the few authors addressing this question in the field of computer science [Rec 2003]. His approach to organizing concepts is similar to the main idea of multi-dimensional systems: for the construction of large concept systems one starts with a number of initially unrelated, self-contained, even mono-hierarchical taxonomies, called *primitive skeletons*. Those hierarchies should further be homogeneous, i.e., the classification within a primitive skeleton follows a single criterion (or several progressively more restrictive criteria). Once primitive skeletons are established, all remaining concepts are defined by means of logical definitions using skeleton concepts, which yields the effect that any concept with multiple parents is a defined concept.

Rector's approach is not only reasonable for the construction phase of an ontology, but having access to these distinctions in the overall system can further be exploited for navigational purposes. Over the last decade, methods corresponding to the multi-dimensional organization of concepts are studied in information retrieval as *faceted browsing / search* or *dynamic taxonomies* [Sac 2000, 2006]. A major focus in this area is the usability of systems. For instance, [Hea 2006] suggests design recommendations for faceted search interfaces and reports that the use of multiple hierarchies does not confuse users. Faceted browsing is further gaining popularity in the context of the Semantic Web [Hil 2006, Spe 2007], and it relates closely to *faceted classification* in library and information sciences. One of the initial proponents of the latter is Shiyali R. Ranganathan, who elaborately provides methodological guidance on the construction of faceted systems in this field [Ran 1967, Ran 1962], cf. [Spi 1998] for simplifications. From a very general point of view, characteristics of facets may be found in many representation approaches, as argued in [Pri 2000], which makes a first, integrative attempt to abstractly describe and formalize facets. However, an in-depth elaboration of facets, neither formally nor ontologically, does not seem to be available yet, albeit it appears desirable due to its ascribed potential of enhancing formal approaches, including description logics.

Intermediately summing up, the complexity of concept systems in terms of the number of their concepts has primarily been addressed by description logics. This solves the larger parts of the compositionality issues, including support for ensuring consistency and to some extent for avoiding insensible concepts. But with respect to search and navigation within large systems further improvements are desirable and may be transferred from other fields.

3 COMPLEX SYSTEMS AND MODULARIZATION IN GENERAL

The previous section is primarily concerned with a single-concept perspective on the overall system, which is a common case from a user point of view. There is another aspect of modularity which is concerned with the overall structure and architecture of an ontology. This trespasses the view of ontologies as a huge set of interrelated concepts or, more formally, as a very large set of formal sentences, which becomes highly relevant for the construction, maintenance and evolution of large ontologies. In addition, modularity contributes to the comprehension by maintainers and users, possibly to training the latter in using the system, and it might correspond to some extent to mental knowledge organization principles. Many systems in nature are very large and complex, which requires major theoretical efforts in order to comprehend and describe them. Artificially created systems have reached similar degrees of complexity. *Modular design* is a universal approach to alleviate complexity in terms of artefactual systems, which is employed in every engineering discipline, e.g. from construction to machine to software engineering. Independent of the reduction of complexity and better comprehensibility, further functional advantages of modular design are the facilitation of change and the encouragement of the parallel development of different parts of a system.

Focusing on ontologies, *modularity* is a very young research area in their respect. Most (medical) terminologies are sparsely structured in their models [Gan 1999, p. 190].[7] The recency of modularity for ontologies applies particularly to ontologies as they appear in currently popular application fields. In the Semantic Web [Ber 2001], for instance, there is a growing need and discussion on modularization of web ontologies, exemplified by a new series of workshops on modular ontologies [Haa 2006, Cue 2008a, Sat 2008]. Much work there is devoted to providing modularity regarding the Web Ontology Language (OWL) [W3C 2004b]. Bio-ontologies such as the Gene Ontology [Ash 2000] or the Open Biomedical Ontologies (OBO)[8] form another active area of application [Kel 2009].

There are a number of domains providing ideas and initial approaches for modularization of ontologies. An active field highly intertwined with it is ontology integration or ontology matching, cf. also [Kal 2009] in this volume. Ontology integration is to some extent more mature than modularization, with [Kal 2003] and [Noy 2004] presenting first review articles of the subject, and a comprehensive book [Euz 2007] being recently available. Moreover, a couple of approaches in knowledge representation offer accounts which may be reused for modularization. The area of knowledge-based systems has also developed solutions – typically added on top of the basic representation formalisms employed. Looking at implemented approaches, an often-mentioned parallel for managing large formal systems, including concept systems, is software engineering [Loe 2006, Ami 2005, Dia 1993]. In connection with formalisms, modularization is frequently studied with respect to formal structures rather than the contents expressed. We follow this route in reviewing and discussing logical theories and modularization in the remainder of this chapter.[9]

Apart from high-level functional desiderata for logical modules, such as facilitating reuse or maintainability, a clear notion of module has not yet been established. In the context of OWL, the primary ontology language of the Semantic Web which is based on rather expressive description logics [Baa 2003b], only a simplistic means is available to tie formulas together which are syntactically distributed over several files. The owl:imports statement is the only element available for a syntax-level splitting of OWL ontologies. That corresponds to the union of the axioms of the importing ontology and the imported ones, which is insufficient from the functional perspective outlined above. Meanwhile, first approaches have been developed in this context, with differing aims and outcomes, e.g. [Bou 2003, Cue 2006c, 2007a, Kut 2008]. In most cases they are based on earlier work which proves adaptable to modularity issues, and usually they fall into one of two types, distinguished by the question of how a resulting modular system is constructed. On the one hand, a *compositional* route can be taken by defining modules and general ways of how these combine into systems. On the other hand, several *decompositional* approaches consider the problem of finding appropriate ways of partitioning a given, typically large ontology. In order to interrelate approaches of either type via a uniform terminology, we formulate a self-contained, general framework in the next section.

4 ABSTRACT FRAMEWORK FOR MODULES

4.1 Overview

The purpose of this framework is to achieve a degree of comparability among the families of current and relevant earlier approaches that are reviewed in section 6. Modules are only introduced on a very general level in this section. Section 5 presents a cross-cutting view on the literature by collecting a number of specific characteristics that are applied in section 6.[10]

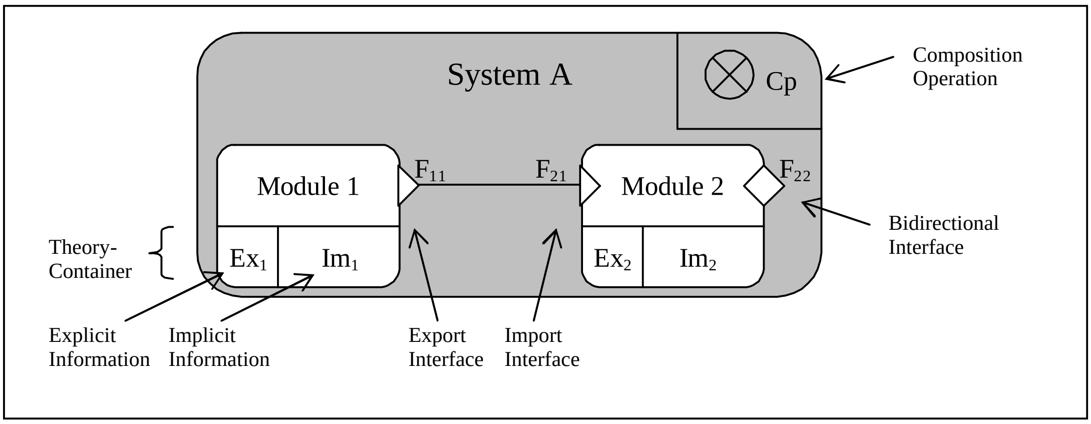

Figure 1: Illustration of major conceptual components of the abstract module definition.

Basically, modules are understood as components of systems which contain information and are interconnected at interfaces (as known from software engineering), where they can exchange information. The primary elements of this model are illustrated in Fig. 1. More logically speaking, such system components contain logical theories, i.e., sets of sentences, which they exchange through their interfaces. Moreover, the system itself can influence this exchange among modules by means of a composition operation.

In the subsequent sections, we refine these intuitions and capture them more precisely in conventional mathematical style, for general comprehensibility.[11] As an example domain, the modular construction of a top-level ontology is utilized, primarily focusing on the combination of a module for time with a module for processes.

4.2 Formal Preliminaries

In the sequel, a very general understanding of a logic is used in order to cover a range of approaches, along the lines of [Bar 1985, Her 1995]. A logic $\underline{L}$ is understood as a triple $\underline{L} = (L, MS, \vDash)$ of a language L, considered a set of sentences[12], a set of model structures MS, and a satisfiability relation $\vDash \subseteq MS \times L$ between model structures and sentences. The latter generalizes in the standard way[13] to model and consequence relations between models, model classes, and theories, also denoted by $\vDash$. The powerset of an arbitrary set S is denoted by *Pow*(S).

Grammar-based definitions of a language L usually distinguish between sentence constructions and a signature/vocabulary V. *Lg*(V) designates the language over V, i.e., all sentences which can be constructed from V in a given grammatical framework. Sometimes it is useful to speak of the same language even for different vocabularies. Any subset $T \subseteq Lg(V)$ is called a theory. *Voc*(T) denotes the vocabulary on which T is based, the language of T is $Lg(T) =_{df} Lg(Voc(T))$. For $T \subseteq Lg(V)$ and $V' \subseteq V$, the restriction of T to V' is defined as $T|_{V'} =_{df} T \cap Lg(V')$. *Cn*(T) denotes the deductive closure of a theory T regarding a fixed consequence relation $\vDash$. *Cc*(T) refers to the set of classical consequences of T, in contrast to certain non-monotonic consequence relations, for instance. The use of *Cc* assumes that T is formulated in a language which can be interpreted classically. The set of all tautologies of a logic is referred to as *Taut*(V) for a signature V and *Taut*(L) for a language L. This formal setting is broad enough to cover at least classical propositional logic (PL) and first-order logic (FOL), as well as description logic (DL) and a number of rule-based approaches with a model-theoretic semantics. Moreover, many non-monotonic formalisms likewise fit under this umbrella.

4.3 Defining Modules

Common to all modules under consideration is that they contain information, i.e., formally, they "contain" a theory. Everything from which a theory may be extracted is called a *theory container*. $Th(\mathrm{C})$ denotes the theory within a theory container C. Further, two kinds of sentences in a theory are distinguished, motivated by the difference between axiom sets and deductively closed theories: sentences may belong *explicitly* or *implicitly* to a theory, where implicit sentences are in some sense derived. For instance, an axiomatization of time may form the explicit part of a theory container, whereas consequences of that axiomatization which are not axioms themselves pertain to the overall theory implicitly.

The set of explicit sentences within a theory container C is denoted by $Ex(\mathrm{C})$, implicit sentences by $Im(\mathrm{C})$, which do not overlap and together yield $Th(\mathrm{C})$, i.e., $Ex(\mathrm{C}) \cap Im(\mathrm{C}) = \emptyset$ and $Th(\mathrm{C}) = Ex(\mathrm{C}) \cup Im(\mathrm{C})$. Note that a theory container C may distinguish explicit and implicit parts of its theory without the implication that the latter is deductively closed; hence $Th(\mathrm{C}) \neq Cn(Th(\mathrm{C}))$ remains possible.

An essential aspect of modules is that they are combined and used together, resulting in a larger system and module intercommunication. Adapted from software engineering, the notion of interfaces is very relevant in this respect. [Loe 2006] discusses a number of options for interpreting the notion of interface for the logical case, among them the view of interfaces as theories (which should be extracted from the module theory). Here we refine this view and distinguish interfaces and their specifications, where the former arise from applying the latter to a theory container. To illustrate the intentions for these notions, consider a time theory which may comprise an elaborate axiomatization of time points, intervals and various relations among them. An interface may be intended to provide restricted access to that theory in terms of a language restriction, e.g. to a theory of intervals and the part-of relationship between intervals only. The interface specification defines that restriction abstractly, and an interface is created by applying the restriction to the time theory, i.e., to a particular theory container.

More formally, an *interface specification* $\mathrm{FS} = (\mathrm{L_M}, \mathrm{L_F}, \mathrm{Op_F})$ is defined by two interfaced languages, $\mathrm{L_M}$ internal to the module and $\mathrm{L_F}$ as available externally at the interface, and a transformation operation $\mathrm{Op_F}$. This operation may serve purposes of adapting internal formulas of the module for external combinations, e.g. by additional relativization of quantifiers, cf. [Ebb 1994, ch. VIII, p. 119 ff.]. It may likewise not change anything in the special case that $\mathrm{Op_F}$ is the identity operation.

Interfaces are connected with information flow and its direction, which can be realized by an exchange of formulas among two theory containers via a connection among their interfaces. We distinguish three types of interface specifications: *import* and *export* interface specifications as well as *bidirectional* ones. Import (export) interface specifications are connoted with the expectation that formulas in the interface language (a) are primarily provided outside (inside) the interfaced theory container and (b) they flow only to (away from) the module. In the top-level ontology example, the time theory may provide an export interface for communicating formulas to the process theory at an appropriate import interface. In contrast to strict export and import, bidirectional interfaces allow for the exchange of formulas in both directions, such that assumptions on the mutual competence are less adequate. This requires different operations for these types: import specifications supply $\mathrm{Op_F} : Pow(\mathrm{L_F}) \rightarrow Pow(\mathrm{L_M})$ and export specifications $\mathrm{Op_F} : Pow(\mathrm{L_M}) \rightarrow \mathrm{Pow}(\mathrm{L_F})$. In the bidirectional case, $\mathrm{Op_F} : Pow(\mathrm{L}) \rightarrow Pow(\mathrm{L})$ with $\mathrm{L} =_{\mathrm{df}} \mathrm{L_M} \cup \mathrm{L_F}$ and $\mathrm{Op_F}(\mathrm{T}) = \mathrm{Op_F}(\mathrm{Op_F}(\mathrm{T}))$ for every $\mathrm{T} \subseteq \mathrm{L}$.[14]

If an interface specification $\mathrm{FS} = (\mathrm{L_M}, \mathrm{L_F}, \mathrm{Op_F})$ is applied to a theory container C this yields an *interface* F, which is represented as $\mathrm{F} = (\mathrm{FS}, \mathrm{C})$, where $Cont(\mathrm{F}) = \mathrm{C}$ and $type(\mathrm{F}) \in \{im, ex, bi\}$ as derived from the type of $\mathrm{Op_F}$ in FS. The language internal to the module is denoted by

Lgm(F), hence *Lgm*(F) = L_M. *Oprn*(F) designates the operation of the interface specification of F. If an input G is provided to Op_F this yields the theory of the interface, $\mathit{Th}(F) =_{df} Op_F(G)$. For export interfaces, the input is the theory of the container, hence $\mathit{Th}(F) = Op_F(\mathit{Th}(C))$. Resuming the time example, an export interface F defined on intervals and their part-of relationships could result in a subtheory of the elaborated time theory *Th*(C), possibly with additionally relativized quantification in *Th*(F).

Transforming a theory container C into a module M means to add a number of interfaces to C, and to provide an operation describing how information imported via interfaces together with explicit / local information of C yield the implicit theory of the module. E.g., in the case of the process module, this operation defines how the process axiomatization combines with imported formulas in the language of time and possibly others. Hence, a module $M = (L_M, C, (F_j)_{j\in J}, Op_M)$ must define $Op_M : \mathit{Pow}(L_M) \times \prod_{j\in J} (\mathit{Pow}(\mathit{Lgm}(F_j))) \rightarrow \mathit{Pow}(L_M)$, which further is the basis to explain the consequences of changes in modules from which a given module imports formulas. The theory of the local container C of M is considered the explicit part of M, if M is viewed as a theory container itself, denoted by $\mathit{Ex}(M) =_{df} Th(C)$, thus requiring $\mathit{Lg}(C) \subseteq L_M$. $\mathit{Ifc}(M) =_{df} (F_j)_{j\in J}$ denotes all module interfaces. Due to Op_M, *Th*(M) is dependent on the inputs to import and bidirectional interfaces of M, hence using *Th*(M) must assume a fixed environment/system in which the module is employed, as introduced next. The theory resulting from importing only empty theories is designated as $\mathit{Th}^{\varnothing}(M)$.

The final step is to compose systems from modules, like forming a top-level ontology from the time and the process theory and others. First of all, such a system S will also contain a theory T in a language L. T is usually derived from the modules $(M_k)_{k\in K}$ and the way they are interconnected at their interfaces; below $\mathit{Ifm}((M_k)_{k\in K}) =_{df} \cup_{k\in K} \mathit{Ifc}(M_k)$ refers to all interfaces in a family of modules, which for a system S over that family of modules is abbreviated as *Ifm*(S). We identify two aspects involved in linking modules and the composed system: (1) a structure U of the intended (possibly mutual) use among the modules, meeting at their interfaces, and (2) a composition operation Cp capturing system-driven transformations between two connected interfaces based on the module usage structure. Cp allows the system to influence the communication between interfaces independently from its modules. For instance, if multiple modules of a top-level ontology provide time formulas, Cp may be used for conflict resolution, taking into account the overall system. Much more commonly, however, systems are formed from axiomatized modules by their set-theoretical union.

A *modular theory* or *system* S is described as $S = (L, (M_k)_{k\in K}, U, Cp)$, where $U \subseteq \mathit{Ifm}(S) \times \mathit{Ifm}(S)$ such that interfaces are connected with suitable antagonists, i.e., for every $(x,y) \in U$: either (*type*(x) = *bi* or *type*(*y*) = *bi*) or (*type*(x) = *im* [*ex*] iff (*type*(*y*) = *ex* [*im*]). The condition that for every $k\in K$: $\mathit{Lg}(\mathit{Th}(M_k)) \subseteq L$ is not required for generality, but we expect it to be relevant in many scenarios. Cp should be a generic operation (i.e., it does not depend on specific modules) and it should respect the distinction of import and export interfaces and the structure U. The theory of S is defined as $\mathit{Th}(S) =_{df} Cp((M_k)_{k\in K}, U)$. A module interface in S is said to be *open* iff it is not connected with any other interface in U. A system is called *saturated* if it does not require external input, i.e., iff there are no open import interfaces; bidirectional interfaces may remain open. Like modules, modular theories are theory containers and may divide their theories into explicit and implicit parts, and they can also be equipped with their own interfaces, independent of those of their modules.

4.4 Example Module Types

4.4.1 Basic Modules

Two formalisms may provide further illustration of how this framework can be used to understand approaches involving logical modules. The first defines a very common view based on classical logical theories (e.g., in FOL or any DL), considering the union of these theories plus its deductive closure as the composition operation for the system.[15] In the above framework, this corresponds to forming a trivial module out of a theory $T \subseteq L$ by setting $M = (L, T, F, \cup)$ with $F = ((L, L', id), M)$, where *id* denotes the identity operation, and $L' \subseteq L$ forms the interface language. That means, the theory itself (or parts of it) serves as a bidirectional interface, and obviously there may be several such interfaces for different sublanguages. Modules of this type are called *basic modules*. They also cover cases where parts of the signature are marked as "external" while the information flow remains unrestricted when joining theories [Cue 2007a]. Possibly this corresponds to interface languages which are a proper subset of L. Given a number of basic modules $(M_i)_{i\in I}$ and their corresponding family of interfaces $Ifm((M_i)_{i\in I})$, the use of the set-theoretical union as composition produces the system $S = (Lg(\cup_{i\in I} Voc(M_i)), (M_i)_{i\in I}, Ifm((M_i)_{i\in I}) \times Ifm((M_i)_{i\in I}), \cup)$.[16] Consequently, $\cup_{i\in I}(Th(M_i)) \subseteq Th(S) = Cc(\cup_{i\in I}(Ex(M_i)))$. For a logic with Craig interpolation such as FOL [Cha 1990, Cra 1957], $Th(S)$ is consistent if and only if each pair of connected interfaces is consistent, cf. also [Ami 2005].

4.4.2 Modules in Distributed First Order Logic

Aiming at an uncommon composition operation, another example is provided by Distributed First Order Logic (DFOL) [Ghi 2000], see also section 6.5. DFOL theories are defined in disjoint first order languages, e.g. $L_1 \cap L_2 = \emptyset$. Syntactically, they can be linked with "interpretation constraints" $\varphi \rightarrow \psi$, where φ and ψ belong to different languages, e.g. $\varphi \in L_1$, $\psi \in L_2$. The intuition is that φ in the L_1-module yields that ψ holds in the L_2-module.[17] Accordingly, a DFOL interpretation constraint implicitly defines two contributions: one to an export interface F_{ex} such that $\varphi \in Th(F_{ex})$ and the other to an import interface F_{im} with $\psi \in Th(F_{im})$, where it is reasonable to let the "union" of all equally directed interpretation constraints between the same modules jointly define those interfaces. Further, the above constraint influences the composition operation of the combined system S by adding the pair (F_{ex}, F_{im}) to its usage structure, thus contributing to deriving the theory $Th(S)$. Notably, the language of S is restricted compared to a system composed of basic modules. DFOL does not allow for composing formulas with constituents from the different module languages.

Most of the approaches to be studied in section 6 deal with basic modules. However, in general, composition within a system may depart from set-theoretical union. For instance, a system might even provide a weaker theory than each or some of its modules, i.e., $Th(S) \subset Th(M_i)$ for every or some $i \in I$. A selection of general characteristics is presented in the next section, as a foundation for comparison.

5 CHARACTERISTICS OF MODULE NOTIONS

All of the subsequent characteristics have been collected within the literature or have been devised from work on the top-level ontology General Formal Ontology [Her 2006, Her 2009a]. Accordingly, they represent a collection of features or desiderata to which particular notions of module may adhere, instead of a set of necessary requirements for every module definition.

5.1 Informal Characteristics

First, we formulate a number of requirements which are hard to state on a sole formal basis.

CI-1 *Comprehensibility*: In order to support maintainability, a module should remain "comprehensible". Two options in order to achieve this for basic modules are (a) the restriction to a rather small vocabulary and a small set of axioms of arbitrary form, or (b) to have a possibly large, structured vocabulary equipped with only simple axioms, which furthermore follow one or a few common schemes. Regarding systems, comprehensibility should derive from the way in which they are composed of basic modules.

CI-2 *Stability*: For a system, changes to a single module or the addition of loosely related modules should not exhibit a strong influence on the structure of the system, i.e., the previous system structure should remain stable. For example, adding a module should not alter connections among other modules. Furthermore, side effects of evolution should be reduced to a minimum. This criterion has also been proposed in [Rec 2003].

CI-3 *Compositionality*: For a number of logical properties of module theories it is desirable to be able to reason over the properties of the theory of the resulting system. For example, a composition operation could be devised such that the consistency of a system immediately results from the consistency of its modules. Compositionality would be very valuable especially for intractable logics such as FOL, because one could then concentrate on proving properties of much smaller theories. For more tractable logics, compositionality would still be helpful in practice, because large ontologies can take current reasoners far beyond their capabilities, cf. [Pan 2006].

CI-4 *Directionality* Composition should allow for a directed information flow among modules, such that a module B may use another, A, without B having an impact on A, cf. [Ghi 2000, Bao 2006a]. Formally this is possible in the framework by using import and export interfaces. Let F_i and F_k be interfaces with which two components M_i and M_k within a system S are connected via an export-import-connection. The exporting module cannot receive formulas over that connection. However, in order to enforce directionality more generally and also on the system level, composition must additionally assure that the combination of $Th(M_i)$ and $Th(M_k)$ to $Th(S)$ respects this limited information flow.

5.2 Formal Characteristics

The remaining features can be formally captured in the framework introduced above. Note that any pre-established constraints with respect to feasibility or computability are not imposed. We will frequently refer to a system S composed of a family of modules $(M_i)_{i\in I}$, for an arbitrary index set I.

5.2.1 Characteristics Primarily Based on either Interfaces, Modules or Systems

CF-1 *Basic Interface and Identity Interface*: Naturally, interfaces should only provide access to and from the module contents. Often, one would even require that they do not change im- or exported formulas. Hence, an interface F is called *basic* iff, for arbitrary input G and output language L, its operation satisfies that $Th(F) \subseteq G|_L$. It is called an *identity interface* iff $Th(F) = G|_L$.

CF-2 *Black-box Interfaces*: The next characteristics aim at the potential of a module at an interface to interconnect with other modules. It is derived from [Cue 2007a] and refers to the assumptions a module holds at an interface it imports from. Given a module M with $Ifc(M) = (F_n)_{n\in N}$, an import or bidirectional interface $F_m = (FS, M)$, $m \in N$, has a (*simple*) *black-box property* iff for an arbitrary input to F_m such that its theory $Th(F_m)$ is consistent, $Th(F_m)$ can be consistently combined with the theory of otherwise only empty imports to M. In general, that means that for an arbitrary theory T and the family

$(T_n)_{n\in N}$ with T_n = *Oprn*(F_n)(∅) for n≠m and T_m = *Oprn*(F_m)(T) such that T_m is consistent, it follows that *Th*(M) = Op_M(*Ex*(M), $(T_n)_{n\in N}$) is consistent. For basic modules, this simplifies to enforcing that *Th*(F_m) ∪ $Th^{\varnothing}$(M) is consistent for arbitrary consistent inputs to F_m. A much stricter version is defined by the analogous assumption for arbitrary consistent imports at all other interfaces of M, which we call a *strict black-box interface* of M.

CF-3 *Module Language Overlap*: This feature refers to the question of whether module signatures or languages may overlap, thus analyzing *Voc*(M_i) ∩ *Voc*(M_j) and *Lg*(M_i) ∩ *Lg*(M_j). There is a variety of options, ranging from empty intersections in both cases[18] over sharing restricted vocabulary elements (e.g. special relations only) to arbitrary overlap among modules.

CF-4 *Classically Closed Composition*: Due to the important role of classical logics, it is often convenient if systems are closed under classical logic: *Th*(S) = *Cc*(*Th*(S)). Basic modules and their compositions are classically closed by definition. The rationale behind this closure is to assure correct reuse of information from S within classical reasoning systems, i.e., when using S itself as a module in another system.

5.2.2 Characteristics with respect to the Interplay of Modules and Systems

For the following characteristics, we assume that modules are equipped with identity interfaces only.

CF-5 *Inclusion of Explicit Module Information*: It appears natural that a system S should provide the information contained explicitly in its modules: $\cup_{i\in I}$ (*Ex*(M_i)) ⊆ *Th*(S).

CF-6 *Inclusion of Complete Module Information*: This is a strengthening of CF-5 to additionally include implicit information of the modules: $\cup_{i\in I}$ (*Th*(M_i)) ⊆ *Th*(S). For basic modules this is called local correctness in [Cue 2006c], which is traced back to [Gar 1989].

CF-7 *Deductive Conservativity*: This criterion is a kind of "inverse" to CF-6, because it requires a system to be a (deductive) conservative extension of its modules: for all i∈I: $Th(S)|_{Lg(M_i)} = Th(M_i)$. Put differently, every module must cover every formula of the system which can be expressed in its language. This is referred to as local completeness in [Cue 2006c]. The criterion of deductive conservativity is recently strongly advocated and analyzed with respect to its application in novel reasoning services for description logics, cf. among others [Lut 2007a,b, Kon 2007, Ghi 2006a, Cue 2008b]; cf. [Kon 2008] for extensions of the notion of conservativity.

CF-8 *Deductive Conservativity over Sublanguages*: A weakening of deductive conservativity yields another form of interrelating a system S and its modules M_i. Here it is not the case that every sentence of S expressible in terms of a module language *Lg*(M_i) must be part of the module theory *Th*(M_i), but this is only required for expressions of a sublanguage K ⊆ *Lg*(S). One option for determining K is the use of sentence schemata. For instance, in a DL setting, [Cue 2006c] defines modules to be closed under subsumption, i.e., for a sentence φ ∈ *Th*(S) of the form C ⊑ D such that C or D ∈ *Voc*(M_i), it is required that φ ∈ *Th*(M_i). This definition requires atomic DL concepts to belong to one and the same module if they stand in a subsumption relation. That can produce severe effects on system structures in case of changes to the system or its modules, especially in decompositional settings if module signatures are not fixed by other means, cf. section 6.3.

CF-9 *Transitivity of Information Flow among Modules*: In general, given a system S in which the modules M_i and M_j are connected such that M_i has an impact on M_j, then composition in S should allow M_i to have an (indirect) impact on any M_k which is connected to M_j, cf. [Bao 2006a]. Different kinds of influence lead to different variants of this criterion. One way of understanding this intuitive statement in the framework is the following. Remember that the composition operation of S should be given independently of specific arguments, i.e., in a way which can be applied to arbitrary theories (e.g., set-theoretical union). The criterion can mean that there are cases of M_i, M_j, and M_k such that $Th(M_k)$ in S differs from $Th(M_k)$ in a copy of S in which M_j ignores imports from M_i, i.e., where all (x,y) from U in S with $x \in Ifc(M_i)$ and $y \in Ifc(M_j)$ are removed.

5.3 Discussion of Characteristics

The above criteria have been identified as those of major relevance for the subsequent comparison of works related to modularization. The selection is further influenced by prevalent discussions in the literature. Some more advanced but less frequently stated criteria have been omitted, like “robustness under joins of signatures” [Kon 2007], similarly for formal, but irrelevant ones, e.g. refraining from the inclusion of only implicit information as (an analog to the explicit case, CF-5). Hence, the list is not a complete compilation, neither with respect to the possibilities in the framework nor to the coverage of the literature.

Concerning interrelations among these characteristics, one can already notice some interplay which is likely to prevent attempts to satisfy all of them as requirements for a single notion of module. For example, it appears non-trivial to satisfy (a) directionality together with (b) the inclusion of module information in the system, (c) deductive conservativity of the system as well as (d) classically closed composition. The naïve approach to directionality fails easily, i.e., a composition operation which assigns competences to modules based on their languages. To see this in a FOL setting, let S be composed of M_1 and M_2 such that M_2 uses M_1, and $Th(M_1) = Cn(Ex(M_1))$, $Th(M_2) = Cn(Ex(M_2) \cup Ex(M_1))$. S may answer queries in $Lg(M_1)$ only by means of the first module, thus by $Th(M_1)$, but queries in $Lg(S) \setminus Lg(M_1)$ in terms of $Th(M_2)$. This leads quickly to a situation where S loses classically closed composition, e.g., by no longer satisfying the deduction theorem of FOL.

The properties of the inclusion of module information and of deductive conservativity (CF-5 to CF-7) appear very restrictive even without the interplay with other characteristics. The question arises whether a non-trivial composition operation can be found which differs from set-theoretical union but satisfies both requirements, all the more since these characteristics originate from settings of basic modules. For the latter, black-box interfaces also exhibit some peculiarities. For modules with a single, possibly language-restricted bidirectional interface, the simple and the strict black-box variants are equivalent because of the restriction to a single source of import. Moreover, a black-box interface with output language L at a basic module M is given iff $Th^{\varnothing}(M)|_L = Taut(L)$. Therefore, this property corresponds to the fact that $Th(M)$ is a deductive conservative extension over every consistent interface theory $Th(F) \subseteq L$. From this follows further that linking two modules at black-box interfaces yields no interaction.

After those considerations on an abstract level, more concrete approaches can be discussed.

6 ANALYTIC OVERVIEW OF LOGICAL APPROACHES

As stated earlier, modularization of logical theories is a rather young research interest. The following families of approaches are reviewed below.

- “conservativity and disjoint languages”
  [Cue 2006a, 2007a, 2008b, Ghi 2006a,b, Kon 2007, 2008, Lut 2007a,b] and
  [Pon 2006a,b, 2007, 2008]
- “partition-based reasoning” [Ami 2000, 2005, Mac 2003]
- “semantic encapsulation” [Cue 2004, 2005, 2006b,c]
- “package-based description logics” [Bao 2006a,b,c,d, Cue 2007b]
- “distributed logic” [Bor 2003, Bou 2003, Ghi 2000, 2001, Ser 2004, 2005]

The first four are recent approaches tackling modularization explicitly, whereas the “distributed logic” approach originates from different motivations, but applies to modularization as well. In this section, we provide a short contextual placement for each of these accounts and sketch their main ideas. Moreover, an interpretation in our framework is discussed to improve their mutual comparability, where the last subsection surveys all features in tabular form.

6.1 Conservativity and Disjoint Languages

The rediscovery of the logical property of conservativity and its potential use in connection with modularization issues is among the latest developments in the fields of description logics and Semantic Web ontologies. Conservativity in its deductive variant means the following (cf. also CF-7, CF-8): given two theories $T \subseteq T'$, T' is a *deductive conservative extension* of T iff T' does not entail consequences that can be expressed in the language of T, but are not already consequences of T. That means, T and T' have the same *Lg*(T)-consequences, $Th(T)|_{Lg(T)} = Th(T')|_{Lg(T)}$. A formal-semantic and usually stronger notion is *model-conservativity*, which requires that every model of T can be extended to a model of T' without changing the interpretation of symbols in the language of T, i.e., if $M \models T$, there is an $M' \models T'$ with $M'|_{Voc(T)} = M$; cf. also e.g. [Lut 2007a, Cue 2008b] for definitional variants.

The application of conservativity to description logics is mainly studied by Frank Wolter and colleagues [Ghi 2006a,b, Kon 2007, Lut 2007a,b]. Their work focuses on decision and complexity problems, as well as procedures for corresponding reasoning tasks like deciding whether one DL theory is conservative over another, or computing explanations for non-conservativity among two theories. More recently, also types of conservativity and properties of such relations among theories are analyzed [Kon 2007, Kon 2008].

In terms of a more explicit approach to defining modularity for DL-based ontologies, conservativity has also been adopted as a foundation in [Cue 2007a]. In this work theories are formulated in languages with their signatures being partitioned into local and external symbols. Moreover, the basic assumption is made that a modular use of the external symbols V_{ext} requires that a theory T should produce conservative extensions $T \cup T'$ over V_{ext} for every extension T' using symbols from V_{ext}. Accordingly, these theories form basic modules with bidirectional identity interfaces as they appear due to language overlap. In addition, the requirement means that the interface defined over the external signature V_{ext} is a simple black-box interface. Note that connections among two theories at these interfaces yield no information exchange among them, cf. section 5.3. [Cue 2008b] is a comprehensive successor of [Cue 2007a] which adopts a weaker, relative definition of “module”: a subtheory $T' \subseteq T$ of a given DL theory T is a module of T for another DL theory S iff $S \cup T$ is a (deductively) conservative extension of $S \cup T'$ with respect to *Voc*(S).

As a special case of conservativity one may consider the decomposition of theories into disjoint language components. The uniqueness (modulo equality formulas) of such decomposition in the first-order case has been proved by Denis Ponomaryov [Pon 2006a,b]. It appears instructive to include this case in Table 1 for comparison. Moreover, the approach is being developed further, recently including relative decompositions of theories, i.e., studying the relation between theories and decomposable subtheories [Pon 2007, Pon 2008]. Accordingly, this work evolves into a partition-based reasoning approach, which is introduced next.

6.2 Partition-Based Reasoning

This approach is motivated with a scenario of reasoning over multiple theories with overlapping content and vocabularies in propositional or first-order logic. A subsidiary aim is the improvement of the efficiency of reasoning by means of partitions. [Ami 2005] presents the latest introduction to the theoretical framework, which is based on earlier publications [Mac 2003, Ami 2000]. The main idea is to use a message passing metaphor from the object-oriented paradigm in software engineering for reasoning over a partitioning of some theory. More precisely, given a theory T and a partitioning $(T_i)_{1 \leq i \leq n}$ of T, message passing algorithms are specified which employ "standard" reasoning within each T_i, but use message passing between certain T_i and T_k if their languages overlap. The specified algorithms are proved to be sound and complete with respect to classical reasoning over T, with these results being heavily based on Craig interpolation [Cra 1957, Lemma 1].

Moreover, [Ami 2005] presents a decomposition algorithm for theories which aims at minimizing three parameters, ranked by importance: for a single partition T_i, it minimizes primarily the number of symbols exchanged with other partitions, secondarily the number of unshared symbols within T_i. For the overall partitioning, the number of partitions should be kept to a minimum, which is the third and least enforced parameter.

In terms of the above framework, the theories in a partitioning are basic modules equipped with bidirectional identity interfaces over $Lg(T_i) \cap Lg(T_k)$, which are composed by set-theoretical union. Therefore, neither directionality nor deductive conservativity is satisfied in general. It remains to be studied whether the decomposition algorithm produces modules over which the system is deductively conservative. Apart from that, the decomposition approach minimizes coupling among modules, module size, and their number.

6.3 Semantic Encapsulation

The term "semantic encapsulation" is borrowed from [Cue 2005], and we use it to cover work on ε-connections [Kut 2004] as well as its continuation and extension especially for DLs by Bernardo Cuenca Grau and colleagues in [Cue 2004, 2005, 2006b,c]. *ε-connections* provide a method of combining certain kinds of logics, hence this approach belongs to the more mature field of combining, fibring, and fusing logics[19]. There is the general result that the combination of decidable logics which can be expressed in terms of ε-connections yields a possibly more expressive, but still decidable formalism. One of the central ideas behind ε-connections is that each logic maintains its own interpretations in terms of separate model structures, but certain restrictions among the distinct model structures can be expressed when combining logics, by means of *link relations*.

Cuenca Grau et al. [Cue 2004, 2006b] apply ε-connections to the combining of ontologies based on description logics, motivated by the idea of an integrated use of independently developed OWL ontologies on the web. This work has later been extended in a way which departs from ε-connections to some extent [Cue 2006c]. Nevertheless, it transfers the above-mentioned feature of the ε-connections method – distinct model structures connected by link relations – to a certain class of SHOIQ theories[20], namely to such that allow for a partitioning of the domain of their models. More precisely, [Cue 2006c] introduces a decomposition approach for such SHOIQ theories by presenting a partitioning algorithm for a theory T, in which the resulting set P of partitions containing subtheories is correlated with a specific group of models of T. These models exhibit a domain partitioning D, and each single partition p of T (i.e., $p \in P$) can be evaluated in a single partition $d \in D$ of the domain of the model. It is this property which may justify the name "semantic encapsulation". Each partition has the property that any two concepts such that one entails the other belong to the same partition. Modules in the sense of [Cue 2006c] are computed based on unions of such partitions, which by the computation inherit this property.[21]

With respect to our model, this account is also concerned with basic modules with identity interfaces, composed by set-theoretical union, yet in contrast to partition-based reasoning here in a DL setting. In its decompositional form, the stability of modular systems arising from such decompositions is not ensured if the initial theory is modified.

6.4 Package-based Description Logics

In [Bao 2006d], Jie Bao and colleagues define a package-based approach for description logics, P-DL, with intended features such as a localized semantics of modules, directional semantic relations among them, and partial reuse or information hiding. Similarly to [Cue 2007a], the symbols of a theory in P-DL are divided into *foreign terms* and *home package terms*. Semantically, the disjointness of model domains as discussed above is alleviated for P-DL, i.e., importing among theories occurs by means of *domain relations*, which are specific one-to-one mappings among the domains of local models required as soon as one theory imports a foreign term.[22]

In general, basic modules form the basis of this account, as well.[23] Directionality in the sense of CI-4 – though intended – is not fully supported yet, because it is possible that one module adds assumptions on foreign terms only, which propagate back to the modules these terms originate from, even if the latter do not import anything from the former, see [Bao 2006a, p. 627]. Note further that [Cue 2007b] presents a self-contained formal definition of P-DL together with a simplified, but equivalent variant which employs identity as the one-to-one mapping among model domains.

6.5 Distributed Logics

The notion of "distributed logics" covers several approaches rooted in the contextual reasoning community, specifically based on Local Model Semantics (LMS) and Multi-Context Systems (MCS), cf. [Ghi 2001, Ser 2004]. More precisely, it refers to works of Luciano Serafini et al. on developing Distributed First Order Logic (DFOL) [Ghi 2000], Distributed Description Logics (DDL) [Bor 2003] and a contextualized form of OWL, called C-OWL [Bou 2003].

The major idea of LMS/MCS, which conveys to the more recent approaches, is to have a collection of theories (possibly in different logical systems) each of which is first of all interpreted locally, i.e., with respect to the semantics of its associated logic (all of which are assumed to be model-theoretically defined). *Domain relations* define (arbitrary) interconnections among the domains of such local models, which can be restricted by *compatibility constraints*, specific syntactic expressions involving different local languages. This yields information exchange among the local theories. *Bridge rules* provide a proof-theoretic counterpart for compatibility constraints: given two theories T_1 and T_2, a bridge rule is a pair of formulas (φ, ψ) such that $\varphi \in Lg(T_1)$, $\psi \in Lg(T_2)$, which states that $\varphi \in Th(T_1)$ allows one to conclude ψ in T_2.

The distributed logic approach is compositional rather than decompositional. Actually, it does not aim at creating an integrated system and has not been invented as an approach to tackle modularization originally. A view of many partially interrelated, coexistent theories is advocated instead. One rationale for this is the motivation to describe communicating agents with possibly different views. Despite of this, the list of properties required for DFOL [Ghi 2000] documents quite some overlap with our collection of characteristics.

Local theories in DFOL can be considered as modules equipped with identity interfaces. Bridge rules provide a non-trivial composition operation, which fully supports directionality in DFOL. Further, due to the peer-to-peer semantics, the language extension at the system level is restricted to compatibility constraints. They cannot be iterated, i.e., only local formulas are permitted as subexpressions of a compatibility constraint, such that arbitrary formulas over the union of the vocabularies are not included. Put differently, the (virtual) system language is severely restricted to the union of the languages of its modules plus compatibility constraints. On a more technical level, the fact that all languages are local and mutually independent creates some inconvenience, because the representation of overlapping vocabularies implies the addition of many bridge rules.

DDL [Bor 2003] transfers the approach to a description logic setting, where only restricted forms of bridge rules have been studied yet, which express forms of inter-module subsumption. However, the composition operation remains fairly unconstrained, as in the basic formalism. The effects of this freedom have been criticized not to meet expectations for modules (named the *subsumption propagation problem* and *inter-module unsatisfiability problem*, cf. [Cue 2004, Bao 2006a]). Although these can be partially avoided by appropriate modeling [Ser 2005], we maintain that this mismatch originates from the motivation of linking systems with different views, which should not be intermingled with modular systems (that are assumed to be, at least potentially, globally consistent and integrated). Due to the named problems [Bao 2006a] denies DDL transitive reuse, which Bao et al. define regarding subsumptions within a module. In general, however, certain effects from one module may flow to another one which is not directly connected to the first, but reachable via an intermediate module, cf. [Ser 2005], thus satisfying transitivity of information flow as defined above (CF-9). Moreover, DDL satisfies the inclusion of implicit information [Bor 2003, p. 170]. Bridge rules provide directionality, except for some special cases where DDL may show effects along the reverse direction of a bridge rule, cf. [Ser 2005].

6.6 Summarizing Overview

Table 1 shows an evaluation against the presented characteristics of modules and modular systems. The order of presentation from left to right is geared to the deviation from classical logic, rather than the chronological occurrence of the approaches. Apparently, the majority of approaches refers to basic modules composed by means of set-theoretical union. To some extent this commonality is due to our selection of works which centers on common, purely logical settings, cf. also the remarks about further relevant work in the next section. It also explains the remarkable acceptance of the inclusion of explicit module information (CF-5).

<<TABLE 1 ABOUT HERE>>

In general, the table entries provide strong indications regarding the criteria, but they should not be seen as assignments with a unique and irrevocable interpretation. For instance, in case of directionality, “No” should be read as such that it is possible to create directional compositions for basic modules as a special case, but it is not enforced in general. Similarly, compositionality with respect to consistency is only satisfied in the case of disjoint signatures if equational theories do not conflict.

Concerning modularization vs. additional motivations of the formalisms discussed, in our opinion distributed logics and P-DL intermingle different views with knowledge organization from a single point of view. We believe that these should be more clearly separated and tackled independently. Furthermore, first comparisons are available, focusing mainly on ε-connections, DDL, and P-DL. [Wan 2007] provides an evaluation of these approaches against a set of primarily technological criteria, clustered into five dimensions: networking, dynamics, distribution, reasoning, and expressivity. The overlap with the characteristics above is rather limited. [Wan 2007] follows [Bao 2006c] which discusses all three approaches from a DFOL perspective, showing their mutual interrelationships in this setting. In contrast, [Cue 2007b] provides a comparison from a description logic point of view. In addition, the authors discuss the option to tackle modularization by novel reasoning services for existing languages, in contrast to developing languages with new, non-standard semantics. It remains open whether one of these lines offers more advantages than the other. From our perspective, reasoning services like determining the conservativity of one theory over another are definitely beneficial for managing theories. As services, however, they exhibit a dynamic character, similarly to decompositional approaches in general – which conflicts with criteria like stability (CI-2). A potential resort for this may be to distinguish “designed modularity” when building large ontologies from “dynamic modularity support” when using or analyzing ontologies.

Altogether, it seems that logical modularization currently produces diversified results and finds itself still in a phase where, based on common, high-level goals like reuse and comprehensibility, the main directions of research require further clarification.

7 CONCLUDING REMARKS

7.1 *Further Related Areas*

The comparison section above concentrates on recent approaches in purely logical settings, most of which explicitly relate themselves to their application for formalizing ontologies. However, there are many other fields which may provide valuable input to theories of logical modularization.

Certainly the most prominent large-scale system involving reasoning and structuring theories has not been presented here, namely the solution provided for structuring the CYC knowledge base [Len 1990] in terms of microtheories. The theory for this has primarily been developed by Rahmanathan Guha [Guh 1991], placed in the field of contextual reasoning and strongly inspired by John McCarthy, who later continued work on contexts with a related formalism [McC 1998]. [Obr 2009, sect. 4] in this volume summarizes both microtheories and reasoning about contexts. Contextual reasoning is at the borderline of our selection. We just note that covering this approach as well appears possible and would clearly transcend basic modules, e.g., by the need for actual transformations in interfaces.

In the Semantic Web area there is a lot of research which either extends logical formalisms more radically, adds extra-logical features, or may also provide less logic-oriented approaches, e.g. for decomposing ontologies in RDF(S) based on properties of RDF graphs. Examples for this overall group are [Stu 2003, 2004, Sei 2006].

Another very relevant line of research refers to formal approaches in software engineering and in the semantics of programs and programming languages. Cursorily, a number of authors including ourselves already draw inspiration upon software engineering, e.g. by referring to notions like interfaces, information hiding, etc. However, established formal results in these fields should be studied more closely, e.g., approaches such as [Ber 1990, Dia 1993]. The latter is founded on category theory, cf. [Ada 1990, Hea 2009]. On a category-theoretic basis several related approaches have been developed. One of them involves the Common Algebraic Specification Language (CASL) [Mos 2004], which was originally designed for algebraic software specification. CASL and its extension HetCASL [Mos 2005], respectively, allow for logical specifications in a variety of systems. They are recently employed for ontologies [Lüt 2006] by means of the Heterogeneous Toolset (Hets) [Mos 2005]. Concerning modularity, that represents an interesting approach insofar as concepts for structuring in CASL are only very loosely dependent on the specific logic in use. [Kut 2008] discusses theoretical foundations for this direction, suggesting the category-theoretic notions of diagram, colimits, and of the institutions of Goguen and Burstall [Gog 1992] as a proper basis for modularity in ontologies. This approach appears promising, despite and due to its high level of abstraction, e.g. [Kut 2008] can directly align itself with the conservativity approaches in section 6, re-interprets ε-connections [Kut 2004] in the semantic encapsulation approach and DDL [Bor 2003] in the distributed logics family, and supports implementations with Hets [Mos 2005].

Finally, the Information Flow Framework (IFF)[24] developed by Robert Kent [Ken 2005] remains to be included. On a strict category-theoretic foundation Kent defines a metatheory intended to serve for the structuring of the Standard Upper Ontology[25], see also [Obr 2009, sect. 4.5]. Similarly to HetCASL and [Kut 2008], a major aim of the IFF is to achieve independence of particular logics in which ontologies may be expressed. Accordingly, both approaches are also closely related to the field of combining logics, while the motivation of the IFF is intimately tied to the organization of ontologies.

7.2 Conclusions

First, let us briefly summarize this chapter. Tracing the route of terminologies and ontologies in information systems, we outline the move of (medical) terminological systems from simple term lists to formal description logic theories. From this perspective, reducing the complexity of systems was first tackled by composing complex concepts from atomic concepts, as well as by a structured arrangement of concepts. Ensuring correct concept composition has ultimately been achieved by the use of description logics, which can be used to prove formal consistency and to avoid meaningless concepts to some extent. Nevertheless, the problems of how to comprehensibly arrange, search, and navigate large structures of categories like polyhierarchies or graphs remain open, despite recent research and some advances.

Due to the status of (description) logics as the contemporary formalism to express ontologies, e.g. in the medical domain or the Semantic Web, we concentrate on logical modularization approaches as a means to tackle the complexity of large categorical systems to the extent of facilitating their comprehensibility, construction, maintainability, evolution and reuse. Regarding work concentrating on ontologies, a corresponding field is currently being established, with a major focus on description logics for the Semantic Web. In order to compare its approaches and discussions, we present an abstract framework to describe characteristics of modules and related notions. On the one hand, this framework is tuned to a high level of generality with sufficient degrees of freedom in order to cover many proposals. On the other hand, it attempts to remain conceptually minimal, by a model established on top of theories mainly by the notions of interface, module, and system. Basically, modules are understood as components of systems which are interconnected at interfaces, exchanging logical sentences. Notably, there are model-based approaches to modularization which seem to deviate from the idea of exchanging sentences.

In the previous section, a number of approaches are introduced and considered regarding this framework. That section indicates the fairly heterogeneous landscape of modularization proposals in terms of the characteristics introduced in section 5. Other characteristics were not covered, like the application of the approaches for top-level ontologies as discussed in [Loe 2006], or more generally the utility and adequacy with respect to specialized purposes. It remains to be seen how those proposals behave along other dimensions.

It may appear questionable to discuss modularization for logical systems rather than genuinely for ontologies. However, we are not aware of any particular ontological theory of modularization, nor elaborate methodological guidance which concerns the structuring of categorical systems. Currently, we see [Rec 2003] as the closest work to attempting such guidance. In connection with logical approaches it is noteworthy that the core of all approaches covers a partitioning / division of the logical *languages*. Ontologically, this may be interpreted to refer to different domains, which leads one to the theory of levels of reality [Gno 2004] and of domains in general. Those may become an initial step to ontological principles of organization. However, this requires further studies, see also chapters [Her 2009b] and [Sym 2009] in this volume. To provide an example, a distinction with logical impact and potential benefit by means of an ontological approach refers to special cases of partitions of categories. In [Loe 2006], taxonomic interrelations of such partitions are distinguished: *horizontal* partitions comprising categories on a similar level of generality, yet in different domains, which are contrasted with *vertical* partitions organized in such a way that all categories in one partition are more general than one or more members in another. [Cue 2006a] adheres to a similar difference and defines diverse characteristics for modularization in correspondence with these cases. Pursuing such routes further may transcend purely formal solutions, possibly founded on more elaborate ontological theories about categories and domains than available at present.

8 ACKNOWLEDGEMENTS

We would like to thank Heinrich Herre, Frank Wolter, Denis Ponomaryov, Alexander Heußner, and Roberto Poli for valuable comments on earlier versions of the manuscript, and we gratefully acknowledge the members of the research group Onto-Med at the University of Leipzig for maintaining a lively and inspiring atmosphere.

9 REFERENCES

[Ada 1990] Adámek J, Herrlich H, Strecker GE, 1990. Abstract and Concrete Categories: The Joy of Cats.
Vol. of Pure and applied mathematics. New York: John Wiley & Sons.
Improved online edition (2004): http://katmat. math.uni-bremen.de/acc/

[Ami 2000] Amir E, McIlraith S, 2000. Partition-based Logical Reasoning.
In: Cohn AG, Giunchiglia F, Selman B, (editors), Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR 2000), Breckenridge, Colorado, USA, Apr 11-15. San Francisco: Morgan Kaufmann, pp. 389–400.

[Ami 2005] Amir E, McIlraith S, 2005. Partition-based Logical Reasoning for First-Order and Propositional Theories.
Artificial Intelligence 162(1-2):49–88.

[Ash 2000] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G, 2000. Gene ontology: tool for the unification of biology.
Nature Genetics 25(1):25–29.

[Baa 2003a] Baader F, 2003. Terminological cycles in a description logic with existential restrictions. In: [Got 2003], pp. 325–330.

[Baa 2003b] Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P, (editors), 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge, UK: Cambridge University Press.

[Baa 2005] Baader F, Brandt S, Lutz C, 2005. Pushing the EL Envelope. In: [Pac 2005], pp. 364–369.

[Baa 2006] Baader F, Lutz C, Suntisrivaraporn B, 2006. Efficient Reasoning in EL+. In: [Par 2006], pp. 15–26.

[Bao 2006a] Bao J, Caragea D, Honavar VG, 2006a. Modular Ontologies – A Formal Investigation of Semantics and Expressivity. In: Mizoguchi R, Shi Z, Giunchiglia F, (editors), The Semantic Web – ASWC 2006: Proceedings of the First Asian Semantic Web Conference, Beijing, China, Sep 3-7. LNCS 4185. Berlin: Springer, pp. 616–631.

[Bao 2006b] Bao J, Caragea D, Honavar VG, 2006b. On the Semantics of Linking and Importing in Modular Ontologies. In: [Cru 2006], pp. 72–86.

[Bao 2006c] Bao J, Caragea D, Honavar VG, 2006c. On the Semantics of Linking and Importing in Modular Ontologies. Computer Science Technical Reports Nr. 465, Department of Computer Science, Iowa State University, Ames, Iowa, USA.

[Bao 2006d] Bao J, Caragea D, Honavar VG, 2006d. Towards Collaborative Environments for Ontology Construction and Sharing. In: Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS 2006), Las Vegas, Nevada, USA, May 14-17. IEEE Press, pp. 99–108.

[Bar 1985] Barwise J, Feferman S, (editors), 1985. *Model-Theoretic Logics*. Vol. of Perspectives in Mathematical Logic. Berlin: Springer.

[Ber 1990] Bergstra JA, Heering J, Klint P, 1990. Module algebra. *Journal of the ACM* 37(2):335–372.

[Ber 2001] Berners-Lee T, Hendler J, Lassila O, 2001. The Semantic Web. *Scientific American* 284(5):34–43.

[Bor 2003] Borgida A, Serafini L, 2003. Distributed Description Logics: Assimilating Information from Peer Sources. *Journal on Data Semantics* 1:153–184.

[Bou 2003] Bouquet P, Giunchiglia F, van Harmelen F, Serafini L, Stuckenschmidt H, 2003. C-OWL: Contextualizing Ontologies. In: Fensel D, Sycara K, Mylopoulos J, (editors), The SemanticWeb – ISWC 2003: Proceedings of the Second International Semantic Web Conference, Sanibel Island, Florida, USA, Oct 20-23. LNCS 2870. Berlin: Springer, pp. 164–179.

[Bra 2004] Brandt S, 2004. Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and – What Else? In: de Mántaras RL, Saitta L, (editors), Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia, Spain, Aug 22-27. Vol. 110 of Frontiers in Artificial Intelligence and Engineering. Amsterdam: IOS Press, pp. 298–302.

[Cal 2005] Caleiro C, Sernadas A, Sernadas C, 2005. Fibring logics: Past, present and future. In: Artemov S, Barringer H, d'Avila Garcez AS, Lamb LC, Woods J, (editors), *We Will Show Them: Essays in Honour of Dov Gabbay*, Vol. 1. King's College Publications, pp. 363–388.

[Cha 1990] Chang CC, Keisler HJ, 1990. *Model Theory*, Vol. 73 of Studies in Logic and the Foundations of Mathematics, 3. ed. Amsterdam: Elsevier.

[Cim 1998] Cimino JJ, 1998. Desiderata for Controlled Medical Vocabularies in the Twenty-First Century. *Methods of Information in Medicine* 37(4/5):394–403.

[Cra 1957] Craig W, 1957. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory.

Journal of Symbolic Logic 22(3):269–285.

[Cru 2006] Cruz I, Decker S, Allemang D, Preist C, Schwabe D, Mika P, Uschold M, Aroyo L, (editors), 2006. The Semantic Web – ISWC 2006: Proceedings of the 5th International Semantic Web Conference, Athens, Georgia, USA, Nov 5-9. LNCS 4273. Berlin: Springer.

[Cue 2004] Cuenca Grau B, Parsia B, Sirin E, 2004. Working with Multiple Ontologies on the Semantic Web. In: [McI 2004], pp. 620–634.

[Cue 2005] Cuenca Grau B, 2005. Combination and Integration of Semantic Web Ontologies. PhD thesis, Department of Informatics, Valencia University, Valencia, Spain.

[Cue 2006a] Cuenca Grau B, Horrocks I, Kutz O, Sattler U, 2006. Will my ontologies fit together? In: [Par 2006], pp. 175–182.

[Cue 2006b] Cuenca Grau B, Parsia B, Sirin E, 2006. Combining OWL ontologies using ε-Connections. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 4(1):40–59.

[Cue 2006c] Cuenca Grau B, Parsia B, Sirin E, Kalyanpur A, 2006. Modularity and Web Ontologies. In: [Doh 2006], pp. 198–209.

[Cue 2007a] Cuenca Grau B, Horrocks I, Kazakov Y, Sattler U, 2007. A Logical Framework for Modularity of Ontologies. In: [Vel 2007], pp. 298–303.

[Cue 2007b] Cuenca Grau B, Kutz O, Honavar V, 2007. Modular Ontology Languages Revisited. In: Finin T, Caragea D, Mladenic D, Sure Y (editors), Proceedings of the Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa) held at IJCAI 2007, Hyderabad, India, Jan 7.

[Cue 2008a] Cuenca Grau B, Honavar V, Schlicht A, Wolter F, (editors), 2008. Proceedings of the 2nd International Workshop on Modular Ontologies (WoMO 2007), Whistler, Canada, Oct 28. CEUR 315. Aachen, Germany: CEUR-WS.org.

[Cue 2008b] Cuenca Grau B, Horrocks I, Kazakov Y, Sattler U, 2008. Modular Reuse of Ontologies: Theory and Practice. *Journal of Artificial Intelligence Research* 31:273–318.

[Dia 1993] Diaconescu R, Goguen J, Stefaneas P, 1993. Logical support for modularisation. In: Huet G, Plotkin G, (editors), Logical Environments: Proceedings of the Second Annual Workshop on Logical Environments, Edinburgh, Scotland. New York: Cambridge University Press, ch. 4, pp. 83–130.

[Doh 2006] Doherty P, Mylopoulos J, Welty CA, (editors), 2006. Principles of Knowledge Representation and Reasoning: Proceedings of the Tenth International Conference (KR 2006), Lake District, UK, Jun 2-5. Menlo Park (California): AAAI Press.

[Dzb 2008] Dzbor M, Motta E, 2008. Engineering and Customizing Ontologies: The Human-Computer Challenge in Ontology Engineering. In: Hepp M, De Leenheer P, de Moor A, Sure Y, (editors), *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*. Vol. 7 of Semantic Web and Beyond: Computing for Human Experience. Berlin: Springer, ch. 2, pp. 25–57.

[Ebb 1994] Ebbinghaus HD, Flum J, Thomas W, 1994. *Mathematical Logic*. Vol. of Undergraduate Texts in Mathematics, 2. ed. Berlin: Springer.

[Euz 2007] Euzenat J, Shvaiko P, 2007. *Handbook of Ontology Matching*. Berlin: Springer.

[For 1996] Forrey AW, McDonald CJ, DeMoor G, Huff SM, Leavelle D, Leland D, Fiers T, Charles L, Griffin B, Stalling F, Tullis A, Hutchins K, Baenzinger

J, 1996. Logical observation identifier names and codes (LOINC) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. *Clinical Chemistry* 42(1):81–90.

[Gan 1999] Gangemi A, Pisanelli DM, Steve G, 1999. An overview of the ONIONS project: Applying ontologies to the integration of medical terminologies. *Data & Knowledge Engineering*, 31(2):183–220

[Gar 1989] Garson J, 1989. Modularity and relevant logic. *Notre Dame Journal of Formal Logic* 30(2):207–223.

[Ghi 2000] Ghidini C, Serafini L, 2000. Distributed First Order Logic. In: Gabbay D, de Rijke M, (editors), Frontiers of Combining Systems 2: Papers presented at FroCoS'98, Amsterdam, 1998, Baldock (UK): Research Studies Press Ltd., No. 7 in Studies in Logic and Computation, pp. 121–139.

[Ghi 2001] Ghidini C, Giunchiglia F, 2001. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. *Artificial Intelligence*, 127(2):221–259.

[Ghi 2006a] Ghilardi S, Lutz C, Wolter F, 2006. Did I damage my ontology: A Case for Conservative Extensions of Description Logics. In: [Doh 2006], pp. 187–197.

[Ghi 2006b] Ghilardi S, Lutz C, Wolter F, Zakharyaschev M, 2006. Conservative extensions in modal logic. In: Governatori G, Hodkinson IM, Venema Y, (editors), Advances in Modal Logic, Volume 6 (AiML-2006), Noosa, Queensland, Australia, Sep 25-28. College Publications, pp. 187–207.

[Gno 2004] Gnoli C, Poli R, 2004. Levels of reality and levels of representation. *Knowledge Organization*, 31(3):151–160.

[Gog 1992] Goguen J, Burstall R, 1992. Institutions: Abstract model theory for specification and programming. *Journal of the ACM* 39(1):95–146.

[Gom 2004] Gómez-Pérez A, Fernández-López M, Corcho O, 2004. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Vol. of Advanced Information and Knowledge Processing. London: Springer.

[Got 2003] Gottlob G,Walsh T, (editors), 2003. Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, Aug 9-15, 2003. San Francisco: Morgan Kaufmann.

[Guh 1991] Guha RV, 1991. Contexts: A Formalization and some Applications. PhD Thesis / Technical Report STAN-CS-91-1399, Computer Science Department, Stanford University.

[Haa 2006] Haase P, Honavar VG, Kutz O, Sure Y, Tamilin A, (editors), 2006. Proceedings of the First International Workshop on Modular Ontologies, (WoMO 2006), Athens, Georgia, USA, Nov 5, CEUR 232. Aachen, Germany: CEUR-WS.org.

[Hea 2006] Hearst MA, 2006. Design Recommendations for Hierarchical Faceted Search Interfaces. In: Broder AZ, Maarek YS, (editors), Proceedings of the First Workshop on Faceted Search held at SIGIR 2006, Seattle, Washington, USA, Aug 10. [Paper available from:] http://flamenco.berkeley.edu/papers/faceted-workshop06.pdf

[Hea 2009] Healy M, 2009, Ontology and Category Theory. In: Healy M, Kameas A, Poli R (editors), *TAO – Theory and Applications of Ontology*, Vol. 2: The information science stance. Part 3, ch. 3

[Hen 2007] Hendler J, van Harmelen F, 2007. The Semantic Web: Webizing Knowledge Representation. In: van Harmelen F, Lifschitz V, Porter B, (editors), *Handbook of Knowledge Representation*, Amsterdam: Elsevier, Foundations of Artificial Intelligence, ch. 21, pp. 821–839.

[Her 1995] Herre H, 1995. Generalized compactness of nonmonotonic inference operations. *Journal of Applied Non-Classical Logics* 5(1):121–135.

[Her 2006] Herre H, Heller B, Burek P, Hoehndorf R, Loebe F, Michalek H, 2006. General Formal Ontology (GFO) – A Foundational Ontology Integrating Objects and Processes [Version 1.0]. Onto-Med Report 8, Research Group Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig, Germany.

[Her 2009a] Herre H, 2009. General Formal Ontology: A Foundational Ontology for Conceptual Modeling. In: Healy M, Kameas A, Poli R (editors), *TAO – Theory and Applications of Ontology*, Vol. 2: The information science stance. Part 2, ch. 4

[Her 2009b] Herre H, 2009. The Ontology of Medical Terminological Systems: Towards the Next Generation of Biomedical Ontologies. In: Healy M, Kameas A, Poli R (editors), *TAO – Theory and Applications of Ontology*, Vol. 2: The information science stance. Part 3, ch. 1

[Hil 2006] Hildebrand M, van Ossenbruggen J, Hardman L, 2006. /facet: A Browser for Heterogeneous Semantic Web Repositories. In: [Cru 2006], pp. 272–285.

[Igg 2001] Iggulden P, Price C, 2001. SNOMED Clinical Terms – looking forward to the next phase. *British Journal of Healthcare Computing and Information Management* 18(10):20–22.

[Kal 2003] Kalfoglou Y, Schorlemmer M, 2003. Ontology mapping: the state of the art. *The Knowledge Engineering Review* 18(1):1–31.

[Kal 2009] Kalfoglou Y, Schorlemmer M, 2009, Ontology Alignment. In: Healy M, Kameas A, Poli R (editors), *TAO – Theory and Applications of Ontology*, Vol. 2: The information science stance. Part 1, ch. 6

[Kei 2000a] de Keizer NF, Abu-Hanna A, 2000. Understanding Terminological Systems II: Experience with Conceptual and Formal Representation of Structure. *Methods of Information in Medicine* 39(1):22–29.

[Kei 2000b] de Keizer NF, Abu-Hanna A, Zwetsloot-Schonk JHM, 2000. Understanding Terminological Systems I: Terminology and Typology. *Methods of Information in Medicine* 39(1):16–21.

[Kel 2009] Kelso J, Prüfer, K, Hoehndorf R 2009, Ontologies in Biology. In: Healy M, Kameas A, Poli R (editors), *TAO – Theory and Applications of Ontology*, Vol. 2: The information science stance. Part 3, ch. 2

[Ken 2005] Kent RE, 2005. Semantic Integration in the Information Flow Framework. In: Kalfoglou Y, Schorlemmer WM, Sheth AP, Staab S, Uschold M, (editors), Semantic Interoperability and Integration. Dagstuhl Seminar Proceedings 04391. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

[Kon 2007] Kontchakov R, Wolter F, Zakharyaschev M, 2007. Modularity in DL-Lite. In: Calvanese D, Franconi E, Haarslev V, Lembo D, Motik B, Turhan A, Tessaris S, (editors), Proceedings of the 20th International Workshop on Description Logics (DL-2007), Brixen-Bressanone, Italy, Jun 8-10. CEUR 250. Aachen, Germany: CEUR-WS.org, pp. 76–87.

[Kon 2008] Kontchakov R, Wolter F, Zakharyaschev M, 2008. Can You Tell the Difference between DL-Lite Ontologies? In: Brewka G, Lang J, (editors), Principles of Knowledge Representation and Reasoning: Proceedings of the 11th International Conference (KR 2008), Sydney, Australia, Sep

16-19. Menlo Park, California: AAAI Press, pp. 285–295.

[Kri 2007] Krisnadhi A, Lutz C, 2007. Data Complexity in the EL Family of Description Logics. In: Dershowitz N, Voronkov A, (editors), Logic for Programming, Artificial Intelligence, and Reasoning: Proceedings of the 14th International Conference (LPAR 2007), Yerevan, Armenia, Oct 15-19. LNCS 4790. Berlin: Springer, pp. 333–347.

[Kut 2004] Kutz O, Lutz C, Wolter F, Zakharyaschev M, 2004. ε-Connections of Abstract Description Systems. *Artificial Intelligence* 156(1):1–73.

[Kut 2008] Kutz O, Mossakowski T, 2008. Modules in Transition: Conservativity, Composition, and Colimits. In: [Cue 2008a].

[Len 1990] Lenat DB, Guha RV, 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.*
Reading (Massachusetts): Addison-Wesley.

[Loe 2006] Loebe F, 2006. Requirements for Logical Modules. In: [Haa 2006].

[Lüt 2006] Lüttich K, Masolo C, Borgo S, 2006. Development of Modular Ontologies in CASL. In: [Haa 2006].

[Lut 2007a] Lutz C, Walther D, Wolter F, 2007. Conservative Extensions in Expressive Description Logics. In: [Vel 2007], pp. 453–458.

[Lut 2007b] Lutz C, Wolter F, 2007. Conservative Extensions in the Lightweight Description Logic EL. In: Pfenning F, (editor), Automated Deduction – CADE-21: Proceedings of the 21st International Conference on Automated Deduction, Bremen, Germany, Jul 17-20.
LNCS 4603. Berlin: Springer, pp. 84–99.

[Mac 2003] MacCartney B, McIlraith SA, Amir E, Uribe TE, 2003. Practical Partition-Based Theorem Proving for Large Knowledge Bases.
In: [Got 2003], pp. 89–98.

[McC 1998] McCarthy J, Buvac S, 1998. Formalizing Context (Expanded Notes).
In: Aliseda A, van Glabbeek RJ, Westerståhl D, (editors), Computing Natural Language, Vol. 81 of CSLI Lecture Notes. Center for the Study of Language and Information (CSLI), Stanford University, pp. 13–50.

[McD 2003] McDonald CJ, Huff SM, Suico JG, Hill G, Leavelle D, Aller R, Forrey A, Mercer K, DeMoor G, Hook J, Williams W, Case J, Maloney P, 2003. LOINC, a Universal Standard for Identifying Laboratory Observations: A 5-Year Update. *Clinical Chemistry* 49(4):624–633.

[McI 2004] McIlraith SA, Plexousakis D, van Harmelen F, (editors), 2004. The Semantic Web – ISWC 2004: Proceedings of the Third International Semantic Web Conference, Hiroshima, Japan, Nov 7-11.
LNCS 3298. Berlin: Springer

[Mos 2004] Mosses PD (editor), 2004. *CASL Reference Manual: The Complete Documentation of the Common Algebraic Specification Language*.
LNCS 2960. Berlin: Springer.

[Mos 2005] Mossakowski T, 2005. Heterogeneous Specification and the Heterogeneous Tool Set. Habilitation Thesis. University of Bremen, Germany.

[Noy 2004] Noy NF, 2004. Semantic integration: a survey of ontology-based approaches. *SIGMOD Record* 33(4):65–70.

[Obr 2009] Obrst L, 2009, Ontological Architectures
In: Healy M, Kameas A, Poli R (editors), *TAO – Theory and Applications of Ontology*, Vol. 2: The information science stance. Part 1, ch. 2

[Pac 2005] Pack Kaelbling L, Saffiotti A, (editors), 2005. Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI 2005),

Edinburgh, Scotland, UK, Jul 30-Aug 5.
Denver, Colorado: Professional Book Center.

[Pan 2006] Pan Z, Qasem A, Heflin J, 2006. An Investigation into the Feasibility of the Semantic Web. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006) and the 18th Innovative Applications of Artificial Intelligence Conference (IAAI 2006), Boston, Massachusetts, USA, Jul 16-20. Menlo Park, California: AAAI Press, pp. 1394–1399.

[Par 2006] Parsia B, Sattler U, Toman D, (editors), 2006. Proceedings of the 2006 International Workshop on Description Logics (DL 2006), Lake District, United Kingdom, May 30-Jun 1.
CEUR 189. Aachen, Germany: CEUR-WS.org.

[Pon 2006a] Ponomaryov D, 2006. Formal Knowledge Representation and the Decomposability Problem. Technical Report 135 (translated version), Institute of Informatics Systems, Siberian Division of the Russian Academy of Sciences, Novosibirsk, Russia.

[Pon 2006b] Ponomaryov D, 2006. Semantic Web Basics in Logical Consideration.
In: Hitzler P, Sure Y, (editors), Proceedings of the First International Workshop on Applications of Semantic Technologies (AST 2006), Dresden, Germany, Oct 6.

[Pon 2007] Ponomaryov D, 2007. Generalized decomposability notions for first-order theories. *Bulletin of the Novosibirsk Computing Center* 26:103–110.

[Pon 2008] Ponomaryov D, 2008. A Decomposability Criterion for Elementary Theories. *Siberian Mathematical Journal* 49(1):152–154.

[Pri 2000] Priss U, 2000. Faceted Knowledge Representation.
Electronic Transactions on Artificial Intelligence 4(Section C):21–33.

[Ran 1962] Ranganathan SR, 1962. *Elements of library classification*.
3. ed. Bombay: Asia Publishing House.

[Ran 1967] Ranganathan SR, 1967. *Prolegomena of library classification*.
Bombay: Asia Publishing House.

[Rec 1997] Rector AL, Bechhofer S, Goble CA, Horrocks I, Nowlan WA, Solomon WD, 1996. The GRAIL Concept Modelling Language for Medical Terminology. *Artificial Intelligence in Medicine*, 9(2):139–171.

[Rec 1999] Rector AL, 1999. Clinical Terminology: Why Is it so Hard?
Methods of Information in Medicine 38(4/5):239–252.

[Rec 2003] Rector AL, 2003. Modularisation of Domain Ontologies Implemented in Description Logics and related formalisms including OWL.
In: Gennari J, Porter B, Gil Y, (editors), Proceedings of the Second International Conference on Knowledge Capture (K-CAP 2003), Sanibel Island, Florida, USA, Oct 23-25. New York: ACM Press, pp. 121–128.

[Rec 2005] Rector AL, Rogers J, 2005. Ontological & Practical Issues in using a Description Logic to Represent Medical Concepts: Experience from GALEN. Preprint Series CSPP-35, School of Computer Science, The University of Manchester.

[Rog 2001] Rogers J, Roberts A, Solomon D, van der Haring E, Wroe C, Zanstra P, Rector AL, 2001. GALEN Ten Years On: Tasks and Supporting Tools. In: Patel V, Rogers R, Haux R, (editors), MedInfo 2001: Towards global health: The informatics route to knowledge: Proceedings of the Tenth World Congress on Health and Medical Informatics of the International Medical Informatics Association, London, UK, Sep 2-5.
Vol. 84 of Studies in Health Technology and Informatics. Amsterdam: IOS Press, pp. 256–260.

[Ros 1997]	Rossi Mori A, 1997. A second generation of terminological systems is coming. In: Pappas C, (editor), Proceedings of the 13th Medical Informatics Europe (MIE 1997), Porto Carras, Greece, May 25-29. Vol. 43 of Studies in Health Technology and Informatics. Amsterdam: IOS Press, pp. 436–440.
[Sac 2000]	Sacco GM, 2000. Dynamic Taxonomies: A Model for Large Information Bases. *IEEE Transactions on Knowledge and Data Engineering* 12(3):468–479.
[Sac 2006]	Sacco GM, 2006. Some Research Results in Dynamic Taxonomy and Faceted Search Systems. In: Broder AZ, Maarek YS, (editors), Proceedings of the First Workshop on Faceted Search held at SIGIR 2006, Seattle, Washington, USA, Aug 10.
[Sat 2008]	Sattler U, Tamilin A, (editors), 2008. Proceedings of the Workshop on Ontologies: Reasoning and Modularity (WORM 2008), Tenerife, Spain, Jun 2. CEUR 348. Aachen, Germany: CEUR-WS.org.
[Sei 2006]	Seidenberg J, Rector A, 2006. Web Ontology Segmentation: Analysis, Classication and Use. In: Carr L, De Roure D, Iyengar A, Goble C, Dahlin M, (editors), Proceedings of the 15th International Conference on World Wide Web (WWW 2006), Edinburgh, Scotland, May 23-26. New York: ACM Press, pp. 13–22.
[Ser 2004]	Serafini L, Bouquet P, 2004. Comparing formal theories of context in AI. *Artificial Intelligence* 155(1-2):41–67.
[Ser 2005]	Serafini L, Borgida A, Tamilin A, 2005. Aspects of Distributed and Modular Ontology Reasoning. In: [Pac 2005], pp. 570–575.
[Sow 2000]	Sowa JF, 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove: Brooks/Cole.
[Spa 1998]	Spackman KA, Campbell KE, 1998. Compositional Concept Representation Using SNOMED: Towards Further Convergence of Clinical Terminologies. In: Chute CG, (editor), A Paradigm Shift in Health Care Information Systems: Clinical Infrastructures for the 21st Century: Proceedings of the 1998 AMIA Annual Symposium, Orlando, Florida, USA, Nov 7-11. Philadelphia: Hanley and Belfus, pp. 740–744.
[Spa 2001]	Spackman KA, 2001. Normal forms for Description Logic Expressions of Clinical Concepts in SNOMED RT. In: Bakken S, (editor), A Medical Informatics Odyssey: Visions of the Future and Lessons from the Past: Proceedings of the 25th AMIA Annual Symposium 2001, Washington DC, USA, Nov 3-7. Bethesda, Maryland, USA: American Medical Informatics Association, pp. 627–631.
[Spe 2007]	Specia L, Motta E, 2007. Integrating Folksonomies with the Semantic Web. In: Franconi E, Kifer M, May W, (editors), The Semantic Web: Research and Applications: Proceedings of the 4th European Semantic Web Conference (ESWC 2007), Innsbruck, Austria, Jun 3-7. LNCS 4519. Berlin: Springer, pp. 624–639.
[Spi 1998]	Spiteri L, 1998. A simplified model for facet analysis: Ranganathan 101. *Canadian Journal of Information and Library Science* 23(1-2):1–30.
[Str 2002]	Straub HR, 2002. Four different types of classification models. In: Grütter R, (editor), *Knowledge Media in Healthcare: Opportunities and Challenges*, Hershey, Pennsylvania: Idea Group Publishing, pp. 57–82.
[Stu 2003]	Stuckenschmidt H, Klein M, 2003. Modularization of Ontologies. Deliverable D21, IST Project 2001-33052 Wonderweb, Vrije Universiteit, Amsterdam.

[Stu 2004] Stuckenschmidt H, Klein M, 2004. Structure-Based Partitioning of Large Concept Hierarchies. In: [McI 2004], pp. 289–303.

[Sym 2009] Symons J, 2009, Levels of Reality
In: Poli R, Seibt J, Symons J (editors), *TAO – Theory and Applications of Ontology*, Vol. 1: The philosophical stance. Part 1, ch. 6

[Vel 2007] Veloso MM, (editor), 2007. Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, India, Jan 6-12. Menlo Park, California: IJCAI.

[W3C 2004a] W3C, 2004. Resource Description Framework (RDF) and RDF Schema (RDFS) Specifications. World Wide Web Consortium (W3C).
http://www.w3.org/RDF/

[W3C 2004b] W3C, 2004. Web Ontology Language (OWL) Specifications. World Wide Web Consortium (W3C). http://www.w3.org/2004/OWL/

[Wan 2007] Wang Y, Bao J, Haase P, Qi G, 2007. Evaluating Formalisms for Modular Ontologies in Distributed Information Systems. In: Marchiori M, Pan JZ, de Sainte Marie C, (editors), Web Reasoning and Rule Systems: Proceedings of the First International Conference (RR 2007), Innsbruck, Austria, Jun 7-8. LNCS 4524. Berlin: Springer, pp. 178–193.

[WHO 2004] WHO, 2004. History of the development of the ICD. World Health Organization (WHO).
http://www.who.int/classifications/icd/en/HistoryOfICD.pdf

[Wri 1997] Wright SE, Budin G, (editors), 1997. *Handbook of Terminology Management*. Amsterdam: John Benjamins.

---

[1] Since 2007, SNOMED CT, the Systematized Nomenclature of Medicine – Clinical Terms, is maintained and further developed by the International Health Terminology Standards Development Organization (IHTSDO), see http://www.ihtsdo.org. The number originates from the page http://www.ihtsdo.org/snomed-ct/snomed-ct0/, accessed on 13.09.2008.

[2] Most of the current terminological systems arrange concepts by subsumption relations, i.e., along their degree of generality. The broad use of the term “subsumption hierarchy” includes trees and directed acyclic graphs, also called polyhierarchies. The depth of such hierarchies refers to the path lengths between roots and leaves.

[3] This mixture appears, e.g., in the description of first-, second- and third-generation terminologies; cf. [Ros 1997, Spa 1998, Str 2002].

[4] GALEN and SNOMED CT have adopted weak description logics very early. [Spa 1998, Spa 2001] report the use of a very restricted and therefore computationally well tractable description logic for an earlier version of SNOMED CT. The only concept constructors referred to are conjunction and existential restrictions, plus top and bottom symbols (apart from bottom, these constructors form nowadays the EL description logic [Baa 2003a]). Such usage is also claimed for GRAIL, the language used for GALEN. However, according to [Rec 1997], the structure of GRAIL appears to be related to, but somewhat different from standard description logics, cf. also [Rec 2005, p. 12 f.].

[5] In [Str 2002], Straub further proposes so-called *multi-point models* in order to allow for multiple values on one and the same dimension on a regulated basis, motivated by examples like a double fracture and given an analysis why extensions by another dimension would not solve that representation problem. Though we agree on the examples, some skepticism about this solution remains on our side, but cannot be elaborated here.

[6] See also [Dzb 2008] for the navigation problem and related challenges from the perspective of human-computer interaction applied to ontological engineering.

[7] On the technical level, they are most often delivered in the form of huge data files, with a technically oriented structure which is rather independent of the conceptual architecture.

[8] Open Biomedical Ontologies: http://obo.sourceforge.net/

[9] Terminologically, we separate the use of “ontology” from that of logical “theory”, generally adhering to more formal, logical vocabulary in this part of the chapter. In the same line the term “semantically” should now be read as referring to a formal, model-theoretic semantics.

---

[10] Note that section 6 provides a review of the selected approaches which should to a large extent be readable without detailed knowledge of the framework. However, the latter supports a unified view and collects recurrent properties required for modules in section 5.

[11] Note that partially similar issues are treated in [Kut 2008] at a comparable level of generality, exposed in more sophisticated terms on the basis of category theory; cf. section 7.1 for more details about this and related works.

[12] We consider only closed formulas, i.e., formulas without free variables (in languages with variables).

[13] One may consider first-order logic as a prototypical case for those notions, cf. [Ebb 1994, ch. III].

[14] This requirement is a real restriction compared to replacing a bidirectional interface with an arbitrary pair of an import and an export interface.

[15] In the sequel, following common conventions and despite actually distinguishing theories and deductively closed theories, we abbreviate this composition operation as “union” or “set-theoretical union”, denoting it as $\cup$.

[16] Concerning the usage structure among basic modules, there is some arbitrariness between two options: either to consider all interfaces connected to each other, or to see connections only if there are non-empty intersection languages among the M_i. We chose $\mathit{Ifm}((M_i)_{i\in I}) \times \mathit{Ifm}((M_i)_{i\in I})$ because even with empty intersection languages some interaction may occur among the M_i, e.g., creating inconsistency based on logical sentences restricting the universe to different cardinalities. However, for FOL the second option is in effect equivalent to this [Ami 2005].

[17] Interpretation constraints are a special kind of “compatibility constraints” as mentioned in section 6.5. Their proof-theoretic counterparts share the same intuition and are called “bridge rules”, a new type of inference rule in DFOL.

[18] In FOL with equality, the equals symbol must be noticed and can be seen as commonly shared. For example, contradictory equality sentences may produce contradictions when joining seemingly signature-disjoint theories.

[19] The field emerged around the mid of the 1990s, exemplified by dedicated publications and events such as the workshop series “Frontiers of Combining Systems”, cf. [Cal 2005].

[20] SHOIQ is a slightly more expressive description logic than SHOIN, the description logic underlying OWL-DL. See [Cue 2005, sect. 2] for an in-depth discussion.

[21] This property has an effect which may be problematic in some cases. Given a domain ontology D modularizable according to [Cue 2006c], the use of an ontology with more general categories G to integrate D-categories by means of subsumption causes all modules to collapse into one. This has practically been observed in tests with the GALEN ontology [Cue 2005, p. 150 f.], and will prevent the use of foundational ontologies in modular fashion according to this approach.

[22] Actually, [Bao 2006d] discusses three types of semantics: a local semantics per module, a global semantics, in which all model domains are united and domain correspondences are merged, and a distributed semantics which includes a central package and all of its imports.

[23] It is tempting to view foreign terms as defining import interfaces instead of the bidirectional identity interfaces of basic modules. This assumption is also supported by the propagation of subsumption relations among terms to modules importing all atomic term components. However, conservativity properties are not generally satisfied by P-DL, and accordingly, bidirectional interfaces appear more appropriate than import interfaces.

[24] Information Flow Framework: http://suo.ieee.org/IFF/

[25] Standard Upper Ontology: http://suo.ieee.org/

Table 1: Overview of logical approaches discussed in section 6 with respect to the characteristics defined in section 5.

Characteristic		*Disjoint Signatures*	*Basic Modules With Conservativity*	*Partition-based Reasoning*	*Semantic Encapsulation*	*Package-based Description Logic*	*Distributed Logics*
Logic		FOL	FOL, DL	FOL	DL, (ε-Connections)	DL	FOL, PL, DL
Way of Creating Modular Systems		Decompositional (in [Pon 2006a,b])	Primarily Compositional	Compositional and Decompositional	Compositional and Decompositional	Primarily Compositional	Compositional
CI-1	Comprehensibility	Not addressed	Not addressed	Not addressed	Not addressed	Not addressed	Not addressed
CI-2	Stability	No	Not addressed	Rather No	No	Rather Yes	Rather Yes
CI-3	Compositionality wrt Consistency	Yes	No	No	No	No	No
CI-4	Directionality	No	No	No	No	Partial	Yes
CF-1 CF-2	Interface Types	None (no interfaces)	Identity, some Black-box	Identity	Identity	Identity	Identity
CF-3	Module Language Overlap	Only Equality	Arbitrary	Arbitrary	Arbitrary	Arbitrary	None
CF-4	Classically Closed Composition	Yes	Yes	Yes	Yes	Yes	No
CF-5	Inclusion of Explicit Module Information	Yes	Yes	Yes	Yes	Locally: Yes Globally: No	Yes
CF-6	Inclusion of Implicit Module Information	Yes	Yes	Yes	Yes	Locally: Yes Globally: No	DFOL: Not appl. DDL: Yes
CF-7	Deductive Conservativity	Yes	Yes	No	No	No	No
CF-8	Deductive Conservativity over Sublanguages	Yes (implied)	Yes (implied)	No	No	No	No
CF-9	Transitivity of Information Flow	No (implied)	Yes	Yes	Yes	Yes	Partially
Additional Remarks			Ded. Conservativity over sublanguages is discussed in [Kon 2007]			Cf. [Bao 2006d] for the distinction of local and global semantics.	Restricted system language in DFOL