Knowledge Driven Software and “Fractal Tailoring”: Ontologies in development environments for clinical systems

Alan RECTOR
School of Computer Science, University of Manchester, UK

Abstract. Ontologies have been highly successful in applications involving annotation and data fusion. However, ontologies as the core of “Knowledge Driven Architectures” have not achieved the same influence as “Model Driven Architectures”, despite the fact that many biomedical applications require features that seem achievable only via ontological technologies – composition of descriptions, automatic classification and inference, and management of combinatorial explosions in many contexts. Our group adopted Knowledge Driven Architectures based on ontologies to address these problems in the early 1990s. In this paper we discuss first the use cases and requirements and then some of the requirements for more effective use of Knowledge Driven Architectures today: clearer separation of language and formal ontology, integration with contingent knowledge, richer and better distinguished annotations, higher order representations, integration with data models, and improved auxiliary structures to allow easy access and browsing by users.

Keywords. Ontology based systems, Model driven architecture, OWL

Citation. Rector A; Knowledge driven software and “fractal tailoring”: Ontologies in development environments for clinical systems. 2010; *Formal Ontology in Information Systems (FOIS 2010)*: IOS Press; 17-30.

1. Introduction

In the past fifteen years, the notion of “ontology” has gone from being an obscure “O” word to almost a synonym for “good”. Everybody thinks, and is often told, that they need ontologies. At the same time ontologies have, on the one hand, become dissociated from other forms of knowledge representation and, on the other hand, been taken by some to be all of knowledge representation. Specialist ontology languages have been developed, and have, to a considerable degree, displaced work on broader knowledge environments. The characteristics of ontology languages, particularly OWL, are often taken as defining “ontologies”.

These trends have served some applications well but others less well. Ontologies have been widely used for annotation, as the basis for terminologies, and sometimes for schema and data fusion, document enrichment and semantic discovery.

Our requirements are different. We wish to build “Knowledge Driven Architectures” analogous to “model driven architectures” (MDA) but based around core “ontologies”. Our group shares the interest of the Model Driven Architectures community in software based on a foundation of declarative knowledge. However, to avoid combinatorial explosions and achieve indefinite levels of "fractal tailoring," it requires logical composition, automatic classification, and a framework for managing defaults and exceptions. This was the motivation for our group's successful early use of ontologies in PEN&PAD [1, 2] and GALEN [3]. We have still only been able to achieve this combination of features using description logics and the related "ontological" methods.

However, perhaps because our requirements are different, we still often find ourselves too often building bespoke environments rather than working with off-the-shelf, preferably open-source, tools, and frequently find ourselves working around, rather than with, standards for ontologies and ontology languages. In the remainder of this paper, we first set out key aspects of our use cases and then identify a number of the issues that limit the usefulness of current systems for our purposes.

2. Our use cases: Building Complex Clinical Systems

Developing successful clinical information systems has been notoriously difficult. Scaling up isolated successes to sustainable wide scale use has rarely proved successful. We believe there are five key problems:

1. *Combinatorial explosions* – of terms, codes, guidelines, protocols, and patients' problems. Any one condition may occur in many locations, grades, types, and be caused by many different etiologies; patients come with many different combinations of conditions; health care institutions with many different combinations of resources.
2. *Fine-grained specialization in healthcare delivery* – so that, combined with doctors' resistance to "one-size-fits-all" solutions, systems are required for myriad specialist niches. This leads to the need for *"fractal tailoring"* of almost every aspect of systems, a requirement that conventional software engineering techniques have been unable to meet.
3. *Variability and legacy of medical language* – so that information is communicated telegraphically, and words' use in practice often does not conform to their literal meanings.
4. *Rapid evolution and high variability of medical knowledge and practice* – in both place and time, so that any fixed solution will be have only limited impact.
5. *Myriad independently developed systems*– to meet the myriad specialist niches so that standards are essential for interworking and intercommunication.

It has not proved possible to build systems flexible enough to address these issues without declarative knowledge representations and knowledge driven approaches in which ontologies play a crucial role.

Our fundamental approach is based on

1. *Compositional systems of descriptions* – so that not all possible entities need be enumerated
2. *Fractal tailoring* – so that information can be expressed in its most general form to "fail safe" and yet be specialized to any degree required by specialty, subspecialty, user, task, and patient situation.
3. *Language independenc*e – so that the same system can be used with many different natural languages and cultures with a minimum of adaptation and translation.

We shall focus on five issues and show later how they combine to achieve the above three aims.

1. The separation of language and the symbolic system that constitutes the logical ontology.
2. *Representation of the background knowledge* in clinical systems, only the core of which is “ontological”
3. *Binding of structural and ontological knowledge*
4. *Representation of higher order and meta knowledge*
5. *Human comprehensibility*

3. Our Approaches

3.1 Ontology and language in information systems

We draw a sharp distinction between the symbolic systems – often described as “models’ – and the language used to label those symbols. The validity for the information systems as we build them depends solely on the symbol system. Communication with the humans about the interpretation of that behaviour depends on the language for labeling, or more generally for interpreting, those symbols. For example, whether we label the main bone in the lower extremity the “femur”, “thigh-bone”, “fémur,” or “combcsontra” – it will be a long bone, part of the lower extremity, have an anatomical neck, and fractures of the anatomical neck will typically be fixated by pins, etc. (using the same symbols labeled in whatever language).

This seems trivial, but Separating linguistic information from the symbolic system / logical model has four important consequences:

1. It allows for separate description of linguistic and logical/ontological phenomena.
2. It allows the language and ontology to evolve independently. Minor spelling and other errors are common, and even major changes over time and regional differences in language not infrequent. The software and symbolic ontology need to be insulated from such purely linguistic changes.
3. It allows language generation techniques to produce human understandable language from the formal descriptions in the ontology language. GALEN showed this to be particularly important for quality assurance and acceptance by those less familiar with the formal system.
4. It allows arguments about language and terms to be separated from arguments about the logical / ontological substance. For example, whether “neoplasm” implies “malignant tumour” or just “tumour”[1] has changed over the last two decades. Arguments over the words have been fierce, but nobody has questioned the need for a way to refer to categories for each of “benign tumour”, “malignant tumour”, and “benign or malignant tumour” nor about their organization in a subsumption hierarchy, nor about their clinical significance. The argument has been almost purely about words. Similarly, few question the need for some top root entity in our representations, The

[1] more properly “proliferation”, since one must include diffuse proliferations such as leukemias.

effect of the labeling of this node in the symbol system on the behaviour of the rest of the system is almost always nil. However, whether it should be called "Top", "Thing", "TopThing", "Concept" or "Category" has occasioned not a little heated debate and occasional disparagement (*e.g.* Smith 2004 [4]), most of it irrelevant to our use of "ontologies" in information systems.

A "Terminology" in our usage is the association of a symbolic system of descriptions and a language layer for communicating those descriptions. The separation of language and content is an old requirement most dramatically stated in Cimino's desiderata [5] for "non-semantic identifiers" – i.e. meaningless symbols as their prime identifiers – a practice now followed by virtually all ontologies or terminologies in widespread use in biomedicine.

Work on the formal relationship between formal symbolic ontologies and language seems to us surprisingly sparse. There is some work on the relation of ontologies and language *e.g.* Hahn [6, 7] and Bateman [8] and some work on the generation of language from description logics [9, 10, 11], but the field seems surprisingly limited. There is now an urgent demand to be able to generate pseudo natural language from expressions in representations such as SNOMED[2] and the NCI Thesaurus and to use natural language techniques to enrich and even compile ontologies.

Finally, logical ontologies combined with literal interpretations of language can lead to errors. For example, in early versions of GALEN, "heart valve" was simply defined as a valve in the heart, and logically subsumed prosthetic valves and various fetal structures that act as valves as well as the four great valves, which are normally the intended clinical meaning of "heart valves". More seriously, in general usage, "Heart disease" includes not only diseases of the heart itself, but also diseases of the covering of the heart ("pericardium"), which is a separate organ clearly distinguished embryologically. To get the clinically relevant answers to "All heart diseases" requires notion of functional association, which is lacking in both the Foundational Model of Anatomy [3] and SNOMED [12]. To match clinical usage, the system must represent that, in standard clinical usage, the English language phrase "heart disease" covers more than the logical category "diseases of the heart."

3.2 Background knowledge: What can be said, what is necessarily true, what can be assumed, and what must be taken into account as fact.

One way of describing the fundamental goal of knowledge driven systems is that their behaviour should depend on declaratively represented background knowledge rather than procedural coding to meet each possible situation. Another is to say that they should interact with their human users on the basis of similar background assumptions. The system should be able to make inferences from "obvious" facts such as that the heart is part of the circulatory system, extremities exist in roughly mirror-imaged pairs, and that pneumonia occurs in the lung, etc.

Either definition requires the representation of extensive background knowledge. However, background knowledge comes in several different forms which are too often conflated, in no small measure because no single environment provides a common framework and syntax that makes it easy to use each appropriately.

---

[2] http://www.ihtsdo.org/

[3] http://sig.biostr.washington.edu/projects/fm/FME/

3.2.1 Ontological or necessary knowledge and logical composition.

The ontological, or necessary, information forms the skeleton of the background knowledge in the form of a logical system of descriptions. It serves four functions:

1. To specify the entities and relations about which there is information to be conveyed.
2. To organize the entities and descriptions into "kind-of" or "subsumption" lattice.
3. To represent the universal characteristics of those entities and relations– those things that are either true by definition or by necessity.
4. To specify what can, and cannot, be sensibly said about those entities and relations.

From these four functions, we gain two key components for our architecture:

1. *An index* of the entities and relations inferred from their to their logical relations
2. *Compositional rules* i.e. how descriptions of new entities can be composed, what descriptions are consistent, which tautologous, and where new description fit in the hierarchical index of kinds. For example, from the fact that lungs are organs we can derive that they can be punctured, and that a "punctured lung" is sensible. From the fact that pneumonia necessarily occurs in lungs, we can derive that "Pneumonia of the lung" is tautologous. It is simply impossible to enumerate everything that might be of interest in biomedical applications, and attempts to do so rapidly lead to absurdities such as the "exploding bicycle". [4]

Ontological knowledge today is most commonly captured in description logics, most often the W3C standard otology language, OWL-DL[5]. The systems we work with all come equipped with one or more reasoners that infer the subsumption hierarchy and check consistency. Such reasoners may be thought of as "terminology compilers" and are essential to our applications: it is simply not possible to manage manually the massively interconnected networks of descriptions and relations needed by clinical systems.

Such description logic based systems serve the functions 1) - 3) above well. They perform function 4) less well. In fact, as standard, they perform the converse – they specify what *cannot* be sensibly said – i.e. what relations amongst entities are inconsistent or "unsatisfiable." What can be said can only be determined as that which is not unsatisfiable. Unless the constraints in the ontology are fully complete, this will be over inclusive. Furthermore, standard reasoners support no simple query for 4), so that the result must be obtained by exhaustive testing to eliminate the impossible, which can be massively inefficient. In short, ontologies as we currently express them are good at ruling out complete nonsense such as "friendly thyroids" but poor at helping us say what might actually be said of thyroids. This is sometimes called the

[4] The number of codes for bicycle accident in related Read ICD terminologies went from eight in 1980, to 80 in the early 1990s, to 587 in the 1999 Australian Clinical modification, including such items as *"V31.22 Occupant of three-wheeled motor vehicle injured in collision with pedal cycle, person on outside of vehicle, nontraffic accident, while working for income"* Jeremy Rogers, personal communication 1999.

[5] Now OWL-DL-2, see http://www.w3.org/TR/owl2-overview/

problem of "sanctioning" from the construct used to deal with this problem in the early description logic GRAIL from the GALEN project [13].

Note that, contrary to the way in which some realist philosophers formulate their criteria for "ontologies" in their sense for their purposes, it is essential that "ontologies" in "knowledge driven information systems" be able to describe entities whether or not they exist in the "real world". Indeed, it is often necessary to describe entities precisely to say that they do not exist. For example, an ontology can describe "metastatic basal cell carcinoma", and may need to do so as part of an information system, precisely to say that, by contrast to all other carcinomas, they have never been observed to occur. Equally, the information system must be able to express that, despite all previous experience, one such has been discovered and to describe it.

This is completely in keeping with the standard logical form used to represent ontologies. In all formal logical languages for ontologies, all statements in the ontology are of the form "All Xs…". It is a well known feature of logic that expressions of the form "All Xs…" do not imply the existence of any Xs, only that if there are Xs, then they must satisfy the given condition.

Confusion might have been less had the common name for the artifacts in information systems now called "ontologies" been taken to be something like "system of descriptions". However, as in the medical domain itself, language evolves for historical reasons, and its historical evolution is a fact we have to live with. We should not allow the evolution of the use of the world "ontology" as it has come to be used in information systems to obscure the fundamental requirements for the artifacts in our information systems now called "ontologies."

3.2.2 *Contingent and typical knowledge*

Whatever definition is used, ontological knowledge is universal. Its characteristic is that all statements are of the form "All Xs …". This is true whether we are dealing with binary relational languages such as OWL or larger fragments of first order logic, or even higher order logic.

Most background knowledge is contingent: i.e. of the form "Most Xs…" "Some Xs…", "There are Xs that…", etc. Consider a seemingly simple question such as "What are the causes of pneumonia" which might appear equally on an undergraduate medical school exam or as part of the process of determining the value set for causes of pneumonia on a data entry form or decision support system. While it is difficult to rule out, absolutely, any micro-organism from causing pneumonia, a standard list of headings "bacterium, virus, yeast, parasite" with a similar short list under each subheading is what is expected for most purposes. Furthermore, even if we take the archetypical cause – the bacterium pneumococcus – it is true neither that all pneumococci cause pneumonia nor the converse. It is not true even true that all pneumococci have the "disposition" to cause pneumonia; many are "non-pathogenic." The best we can say in standard ontology languages is that there is a class of pneumococci that cause (or have the disposition to cause) pneumonia – which is true if and only if there is also a class of pneumonia caused by pneumococcus (although none of the standard reasoners deliver this inference automatically or simply).

The ontology can provide a framework for contingent knowledge, but the contingent knowledge itself can be expressed at best indirectly within the ontology itself. The authors have argued elsewhere [14] that, if we are going to use ontological forms, then the best formal answer to what causes pneumonia is a list of the form

"Bacteria that cause some Pneumonia", "Virus that causes some Pneumonia" etc. Such expressions can be used safely in further inference, and suitable user interaction software can strip off the parts that appear redundant to the user.

However, is even this appropriate? If the ontology is about what is universal, then what is contingent ought to be expressed in some other way, using the ontology but going beyond it. A major problem is that we lack a convenient formalism and environment that covers both contingent and universal knowledge.

Note that using first order logic does not address this problem adequately. Intrinsically, first order logic provides only a simple existential qualification to which is usually taken to capture the notions of "some", although even this is at variance with common linguistic usage. However, the information we need to answer the question "What causes pneumonia" is really, "What commonly causes pneumonia?" A commonly used alternative to using existential quantifiers to get the effect of "generally" or "may" in ontology languages is to create pairs of properties systematically– *e.g.* "*may_cause"* and "*causes"*. All variants of this method suffer from various defects. In particular they are not reciprocal; even if the relation "*may_cause*" is symmetric, we cannot conclude that if "(all) pneumococcus *may_cause* some pneumonia" that "(all) pneumonia *may_be_caused_by* some pneumococcus". Yet, our expectation is that "If A *may_cause* B, then "B *may_be_caused_by* A".

We distinguish three cases:

1. *Categorical facts about what generally happens* – the case of causes of pneumonia
2. *Defaults with exceptions* – how to express the general knowledge that most birds fly, or most patients with cardiovascular disease should have their data collected with a given protocol
3. *Probabilistic or uncertain knowledge* – more specific knowledge about the probabilities, typically in medicine, that one condition is caused by another, e.g. that the link between a positive HIV test and HIV infection is X%, or that 70% of male patients with acute abdominal pain have appendicitis.

All three are important to biomedical applications. As discussed above, *categorical facts* about what generally happens form the basis for many question answering and user input systems.

Defaults and exceptions apply wherever there are complex networks of facts that form interacting contexts. In the decision support systems, they are important for assigning actions and treatments according to context, *e.g.* that beta-blockers are generally strongly contraindicated in asthma but certain subclasses of beta-blockers may be used "with caution." A safe default value is required, even if it is often overridden in specific contexts. In data capture systems they important for assigning "value sets" to fields according to context. For example, the default value set for "loudness" may be "loud, medium, soft, inaudible", but when applied to heart murmurs, the value set is "Grade_I…Grade_VI". Requirements from users include complete replacement of value sets, addition to existing value sets, and deleting of specific values – all while maximizing re-use.

The fundamental requirement for defaults and exceptions is that they allow systems to be built so that they "fail soft" – i.e. work safely with a minimum of information, but can be tailored to any degree through any number of stages to meet any degree of sub sub …sub specialization. This is what we refer to as "*fractal tailoring*".

There is no general logically complete and computationally tractable solution to reasoning with defaults. However, collecting the set of most specific values for a given property according to the standard Touretzky criterion [15] provides a set of candidate solutions. If the ontology is well normalized [16], this set is almost always a singleton. Although provision must always be made for resolving the conflicts that cannot be ruled out, this method provides a quick, easily implemented, and well tested mechanism that, in practice, covers the majority of cases. In the author's opinion, ignoring or abandoning this heuristic, has been a major step backwards in the use of knowledge based systems and ontologies in clinical medicine, and perhaps other fields.

Probabilistic knowledge is very different and required for decision support or suggestions as well as aspects of "intelligent interfaces". Exact probabilities are rarely critical in our applications (and the statistics to support them are rarely available). However, approximations for updating belief that behave consistently with probability theory are important. Most obviously, we need to be able to attach "updating strengths" to causal or evidential links – roughly likelihood ratios and the likelihood ratio of the inverse – so as to be able to say that "X is good evidence for Y if present" or "X is good evidence against Y if absent" or both. Bayesian networks have made enormous strides in the past two decades, and there are several proposals for combining probabilities with ontologies [17-20]. However, given that probabilistic knowledge is clearly not universal, and clearly uses numerical calculations rather than logical inference in its implementation, is integration the most appropriate path? Or would hybrid systems exploiting the complementary relation between two forms of reasoning be more promising? For example, one might imagine systems in which the ontology indexed updating strengths and Bayesian calculations (or approximations to them) performed the actual updating of beliefs.

3.2.3 Facts and databases

Facts ought to be simple: relations between two or more individuals, *e.g.* that a particular drug is licensed for use in a particular disease, that the ID number for this product is NNNN, that the topic of this trial is breast cancer. The difficulty for ontology languages and systems is that most facts of importance are compiled in closed world databases, where negation as failure is intrinsic to correct interpretation. By contrast, the "facts" or "A-Boxes" of most ontology languages are open world and take negation to mean proven falsehood. Because most large fact bases in biomedicine come from closed world databases, large open world description logic "A-boxes" are rarely appropriate. For example, if there is no product with ID NNNN in a drug database, then there is no such product. If it is there, but there is no entry that says it is licensed for the treatment of asthma, then it is not licensed for the treatment of asthma. No further inference is needed; a demand for further proof leads to wrong, possibly dangerous, answers.

A common pattern is of an open world schema or ontology describing the categories for a closed world database of facts. There is much work on large A-Boxes for OWL. There appears to be much less work and few off-the-shelf solutions for systems combining OWL T-boxes with standard closed world databases even though this combination frequently appears to be appropriate.

3.3 Meta data and annotation

It comes as something of a shock to many ontology implementers that, in many applications, users regard the meta-data and annotations as the core of the system. To the users, the ontology is often little more than a skeleton for those annotations. It is regrettable that most knowledge representation and ontology systems provide a single mechanism for annotation and meta data, even though there are numerous different categories of meta data and annotation that play very different roles in systems:

1. *Comments and guidelines* as to what the entities in the ontology are supposed to mean and how they are expected to be used. Simple names or "descriptions"[6] are rarely enough, and in any complex ontology, further information is required.
2. *Editorial information and provenance* on which both the credibility and the maintenance of the ontology depend.
3. *"Schemas" for the ontology*. Curiously, while databases have long emphasized the schema, ontologies have not. There is much work on universal upper ontologies, but little work on describing the specific domain level ontologies in ways that make it easy for users to comprehend them.
4. *Information to software* about how different entities in the ontology are to be used, additional information to support "sanctioning" and other issues of use, information on how entities are combined into patterns, access to "attached procedures" for calculations, etc. At its richest, such metadata forms a major extension to the reasoning mechanisms as proposed by Parsia.[7]

3.4 Higher order statements

Metadata and annotations should not be confused with higher order statements, even though in many of our ontology languages, higher order statements can only be expressed as annotations.

Consider the statement: "Whales are an endangered species." It is neither a statement about "All whales", nor an annotation on the information about whales, nor an annotation on the symbols in the knowledge representation itself. It is a statement about "Whales" qua class or species. Likewise, it is difficult to deal with aggregate notions such as the average value of a test or the prevalence of a condition – *e.g.* "Swine Flu" - except by treating that entity "Swine Flu" as a first class object in its own right. That is, we need to be able to express higher order information about "Swine Flue" itself as opposed to first order information about "All instances of swine flu". There are tricks, *e.g.* creating a class for "cases of swine flu" of or a "dependent continuant" for "Swine Flu", but none has been demonstrated to be satisfactory.

OWL and related reasoners support only first order reasoning and reject any ontology with higher order statements. Therefore, we are faced with a trilemma. We either give up reasoner support completely or give up higher order statements or resort to tricks. "Punning" is a compromise in the new OWL-2 standard but still limited. There are but proposals for greater integration either through "rich annotations" layered implementations [21], but none are widely available. This needs to be rectified. To

[6] what typically appears in the Dublin Core dc:description. (See http://dublincore.org/)
[7] http://code.google.com/p/owl1-1/wiki/RichAnnotations

meet requirements, systems need to allow higher order knowledge to be represented and queried while still allowing full inference with the first order component.

3.5 Binding of Ontology to Data Models

Most information systems are specified in terms of "data models" that are really templates of fields and value sets. Such template-based data models' behaviour is precisely opposite of that of ontologies and description logics. Description logics are sets of axioms that restrict what can be said. The more that is known, the more the domain is restricted and the less can be said. By contrast, in information systems, whether specified by UML (at least in practice) or object oriented models, are templates. Nothing can be said until there is a place for it in the template. Hence, the more you know, the more can be said – *i.e.* in UML parlance, the more associations and attributes apply.

A special requirement for meta knowledge is in using ontologies to provide the titles and value sets for fields in information systems. Space does not permit repeating the arguments made elsewhere [22], that when entities from ontologies are used in data models the classes from the ontology should be treated as individuals in the data model – *i.e.* that the information about how to use them in data model is metadata about the class in the ontology artifact rather than first order information about the members of that class in the ontology domain. Treating data structures and ontology classes at the same level leads, inevitably, to absurdities such as a "person with a missing blood pressure." Rigorously synergistic methodologies and environments for binding ontologies and data structures remains a major challenge.

4. Human navigable hierarchies: Thesauri, classifications, and other artifacts that are not ontologies.

4.1 SKOS and knowledge organization systems

Ontologies are frequently unintuitive to navigate for domain experts. Upper level entities are regarded as just "noise". Doctors are interested in disorders, anatomic structures, findings, etc. These categories have long histories even if often the topic of much dispute. Furthermore, even distinctions that they would agree with in theory they do not use in practice – *e.g.* the difference between "breast cancer" and "breast tumor".

Each domain will differ, but two key distinctions are pervasive in clinical systems:

1. Between entities that can be present or absent and those that can be normal or abnormal
2. Between those than can be directly observed or measured and those that must be inferred.

The first distinction might be derived from the ontology – those entities that can be normal or abnormal are those that are always present – mainly qualities or parts of the body or its parts – *e.g.* heart rate, heart valve, skin colour, etc. Those that can be present or absent are disorders or abnormal structures – fractures, bruises, tumors, diabetes, dementia, etc. – roughly parts and dependent continuants in BFO's language.

The second distinction is more subject to arguments. Most clinicians would agree that the "blood pressure" was an observable finding and "hypertension" an "disorder" arrived at by judgment. However, whether "elevated blood pressure" should be a judgment or an "observable" is far from clear.

Equally awkward, most clinicians expect clinical entities to be organized according to how they are traditionally collected and reported in clinical histories and physicals, organized according to nested sections rather than any logical notion of subsumption. Similarly, most information systems expect entities to be defined by their position in a data model, which is typically a containment structure – often today an XML schema – which again depends on containment rather than subsumption.

We can imagine two solutions to these problems:

1. Formal tricks, so that the primary organization is along the doctors expected lines and the ontological categories are provided as additional characteristics (restrictions in OWL). This can work up to a point, although it always risks creating inconsistencies.
2. An entirely separate Knowledge Organisation System over the clinically significant entities in the ontology. The most obvious candidate for such organisation is the new W3 standard SKOS[8]. This seems a safer solution.

In fact, providing a separate "browsing view" of the ontology based on non-logical principles seems critical to user acceptance in many situations. Ideally, the non-logical view would be systematically linked to the ontology – *e.g.* by combining subsumption, partonomy, and causality with annotations as to which entities in the ontology should be "hidden" in the browsing view. However, in other situations, it may be necessary to construct such a view entirely independently just because that is the way users expect it.

1. Summary: Requirements for Environments for Ontology Driven Applications

The only means we know to meet the range of requirements for clinical information systems require Knowledge Based Architectures with "ontologies" at their core. The key criteria are that the ontology support the information to be conveyed and the computational operations that need to be performed on that information. The key values of using ontologies are in supporting:

1. *Composition of descriptions* so that not everything that has to be described need be enumerated.
2. Automatic Indexing and "Fractal tailoring" with defaults with exceptions.

Essential to the effective functioning of ontologies for these purposes are:

1. Clear separation of language and symbolic knowledge
2. Clear understanding of the binding between the axiom-based ontology and the template-based data models that use the symbols from that ontology.
3. Auxiliary information to make the ontology comprehensible and usable by human authors, software engineers and other users.

Ultimate solutions to formal integration of all relevant forms of reasoning are beyond current technology, probably beyond possibility. However, heuristic integration has proved possible in many cases. We suggest it as a "grand challenge" to the "ontologies" community to develop, test, and refine the generic heuristics needed

[8] http://www.w3.org/TR/2009/CR-skos-reference-20090317/

for hybrid systems to realize the potential of ontology driven software to achieve the flexibility, maintainability, and usability needed by future information systems, whether in biomedicine or elsewhere.

References

[1] W. A. Nowlan, Kay, S., Rector, A. L., Horan, B., and Wilson, A. 1991. PEN&PAD: A multi-lingual patient care workstation based on a unified representation of the medical record and medical terminology. Medical Informatics Europe (MIE-91). 1043-1048.

[2] A. L. Rector, Glowinski, A. J., Nowlan, W. A., and Rossi-Mori, A. 1995. Medical concept models and medical records: An approach based on GALEN and PEN&PAD. *Journal of the American Medical Informatics Association* 2:19-35. (reprinted in JH van Bemmel and A T McCray, *Medical Informatics Yearbook 1996* pp 225-241).

[3] A. L. Rector and Rogers, J. E. 2006 Ontological and practical issues in using a description logic to represent medical concept systems: Experience from GALEN. In *Reasoning Web*, P. Barahona, F. Bry, E. Franconi, N. Henze, and U. Sattler,Eds Springer-Verlag.

[4] B. Smith 2004. Beyond concepts: Ontology as reality representation. Formal Ontology in Information Systems (FOIS 2004).

[5] J. J. Cimino 1998. Desiderata for controlled medical vocabularies in the twenty-first century. *Methods of Information in Medicine.* 37, 4-5, 394-403.

[6] U. Hahn and Romacker, M. 1999. An integrated model of semantic and conceptual interpretation from dependency structures. COLING 2000 - Proceedings of the 18th International Conference on Computational Linguistics. 1, 271-277.

[7] U. Hahn, Romacker, M., and Schulz, S. 2000. medSynDiKATe: Design considerations for an ontology-based medical text understanding system. AMIA 2000 - Proc Annual Symposium of the American Medical Informatics Association. . 330-334.

[8] J. A. Bateman 1993. Ontology construction and natural language. Intl Workshop on Formal Ontology. 83-93.

[9] R. H. Baud, Rassinoux, A.-M., and J.-R, S. 1991. Knowledge representation of discharge summaries. Third European Conference on Artificial Intelligence in Medicine. 173-184.

[10] J. C. Wagner, Rassinoux, A.-M., Baud, R. H., and Scherrer, J.-R. 1994. Generating noun phrases from a medical knowledge representation. Twelfth International Congress of the European Federation for Medical Informatics, MIE-94. 218-223.

[11] D. Scott and Power, R. 2001 Generating textual diagrams and diagrammatic texts,. In Cooperative Multimodal Communication, H. Bunt and R.-J. Beun,Eds Springer-Verlag.

[12] C. Rosse and Mejino, J. L. V. J. 2003. A reference ontology for biomedical informatics: the Fondational Model of Anatomy. *Journal of Biomedical Informatics*. 36, 478-500.

[13] A. L. Rector, Bechhofer, S., Goble, C. A., Horrocks, I., Nowlan, W. A., and Solomon, W. D. 1997. The GRAIL concept modelling language for medical terminology. *Artificial Intelligence in Medicine*. 9, 139-171.

[14] A. Rector, Stevens, R., and Drummond, N. 2008. What causes pneumonia? The case for a standard semantics for "may" in OWL. Owl Experiences and Directions (OWLEd 2008). www.webont.org/owled/2008/papers/owled2008eu_submission_14.pdf.

[15] D. S. Touretzky. 1986 *The Mathematics of Inheritance Systems*. Morgan Kaufmann.

[16] A. Rector 2003. Modularisation of domain ontologies Implemented in description logics and related formalisms including OWL. Knowledge Capture 2003. 121-128.

[17] Z. Ding and Yun, P. 2004. A probabilistic extension to ontology language OWL. 37th Hawaii International Conference on System Sciences (2004).

[18] P. Klinov. 2008 Pronto: A non-monotonic probabilistic description logic reasoner. In The Semantic Web: Research and Applications, Springer Verlag.

[19] P. Klinov and Parsia, B. 2009. On improving the scalability of checking satisfiability in probabilistic description logics. International Conference on Scalable Uncertainty Management (SUM-2009). Springer LNCS 5785/2009, 138-149.

[20] D. Koller, Levy, A., and Pfeiffer, A. 1997. P-Classic: a tractable probabilistic description logic. Fourteenth National Conference on Artificial Intelligence(AAAI-97). 390-397.

[21] J. Z. Pan, Horrocks, I., and Schreiber, G. 2005. OWL FA: A metamodeling extension of OWL DL. Proc International workshop on OWL: Experiences and Directions (OWL-ED2005).

[22] A. Rector, Qamar, R., and Marley, T. 2009. Binding ontologies and coding systems to electronic health records and messages. *Applied Ontology*. 4, 1, 51-69.