Knowledge Integration in Open Worlds: Utilizing the Mathematics of Hierarchical Structure [*]

Cliff A Joslyn, Karin M Verspoor
Information Sciences
Los Alamos National Laboratory
Los Alamos, NM, 87545, USA
{joslyn,verspoor}@lanl.gov

Damian DG Gessler
National Center for Genome Resources
Santa Fe, NM 87505 USA
ddg@ncgr.org

Abstract

Semantic web services present a new challenge in distributed knowledge integration. Under the Simple Semantic Web Architecture and Protocol (SSWAP, http://sswap.info), web resources publish information about their data and services in terms of OWL ontologies. With semantic tagging, the output of one service can drive the input of another, presenting the alluring prospect of machines autonomously assessing the suitability of web resources for distributed workflows. Ontological reasoning under OWL greatly expands the potential for semantic web service interoperability by guaranteeing the compatibility of resources using such techniques as inferencing, typing, and class subsumption. Formal methods for managing such structures are urgently needed. Recent advances in the mathematics of hierarchies can formalize the construction of synthetic hierarchies from unstructured information, identifying mutual subsumption architectures. Here we examine the problem space and describe the promise of mathematical order theory (formal concept analysis and lattice metrics) for advancing knowledge integration in open worlds.

1. Semantic Web Services in Open Worlds

1.1. The Limitations of Web Services

In high-throughput, web-based integration, the sheer size of the data or problem complexity demands technologies to empower computers to assist humans in finding, discerning, invoking, and integrating disparate data and services from across the web. Concomitant with the size and complexity issue, there are few standards for how data should be structured on a human-readable web page. For example, the HTML `<table>` tag may be used as a pseudo-data structure for two-dimensional data, metadata, formatting, or simply not used at all. Thus data acquisition from web pages is reduced to page "scraping", or parsing web pages for data based on heuristics and page formatting conventions. Page scrapping is error prone because web pages are inherently idiosyncratic in their content; it is also labor intensive since it is difficult and inefficient to refactor page scraping code across web sites or even pages within a site.

The problem is not restricted to parsing HTML-based web pages; for example, despite a plethora of domain-specific XML-based "markup languages", there is no reliable high-throughput method to integrate data even if it is all encoded in XML. DTDs which often accompany XML documents are limited to specifying the structural and syntactical integrity of the XML document, but are insufficient for instructing computers to discern the suitability of the content for a particular purpose. HTML was never designed for machine-machine integration of data and services.

This has precipitated web services as an approach distinct from browser-based technologies. Here providers, such as web sites offering data or algorithmic services, agree to a common set of rules that allow clients to engage their services, for example using the Simple Object Access Protocol (SOAP, http://www.w3.org/2000/xp/Group) for remote procedure calls, the Web Services Description Language (WSDL, http://www.w3.org/TR/wsdl) for service description, and the Universal Description Discovery and Integration (UDDI, http://www.uddi.org) specification for service registration. These standards have been adopted or ignored to varying degrees: where XML is ubiquitous, UDDI has been nearly abandoned. If adopted fully, web service standards would allow for robust interoperability.

But these protocols define only a *syntactic* framework for clients to make requests and providers can return responses. They do not provide an extensible *semantic* framework (a standard for inferring meaning) within which clients and providers can describe their data and services using domain-specific annotations for engaging in semantic negotiation.

[*] Work supported in part by the National Science Foundation under Grant No. 0516487.

0-7695-2997-6 2007
U.S. Government Work Not Protected by U.S. Copyright
DOI 10.1109/ICSC.2007.82

IEEE computer society

In web services, semantics (the meaning of tokens and the relationships among entities) are largely implied, and the consequences of their intended use are not formalized in a machine-computable logic. For example, if you want to retrieve a DNA sequence (or a stock quote), you may be intrigued by the SOAP services and WSDL descriptions for a "DNASequenceRetrieval" that requires a "GeneSymbol" as input, or a "FreeStockQuote" service that requires a "StockSymbol" as input. But in the absence of any explicit semantics about the services, one could only have inferred suitability of these services based on *ex situ* knowledge or heuristics. Without human intervention, there is often no way to know that these services are any more or less appropriate than services "Foo" and "Bar". And this, of course, is a critical problem for machines charged with assessing data and services for discerning high throughput integration. It is also a problem for humans when they are presented with service names such as (the hypothetical and pedagogical example) "NucAcdSeqRet_Ver321_Txn_SCrv_RevB": just exactly what are you getting if you invoke that service? (Nucleic Acid Sequence Retrieval; Version 3.21 for taxon Saccharomyces cerevisiae, Revision B.)

1.2. Semantic Representations

The W3C, the sanctioning body of the World Wide Web, has a standard for encoding semantics as class and property axioms in a first order description logic in RDF/XML. This is done with the W3C standard OWL (Web Ontology Language, www.w3.org/2004/OWL) built upon RDF (Resource Description Framework, www.w3.org/RDF), RDFS (RDF Schema, www.w3.org/TR/rdf-schema), and XML Schema (www.w3.org/XML/Schema). A description logic is a logical system specifying instances (individuals), their relationships (properties or predicates), and the sets (classes) to which they belong. A first-order logic allows quantification (e.g., there exists; for all) over individuals (in contrast to second- or higher-order logics allowing quantification over properties and classes). Further restrictions are that the sets of all individuals, properties, and classes are disjoint (an entity cannot be both an individual and a class), thereby restricting the types of statements that can be made (one cannot make general statements about classes of classes).

These and other restrictions are important, since first order description logics can be proven to be consistent, complete, and decidable. Using OWL, one can describe individuals (such as web resources of data and services represented by hyperlinks) and their logical relationships to each other. In the biology domain, one can use ontologies, such as the Gene Ontology [6] (GO, http://www.geneontology.org) and Sequence Ontology (www.sequenceontology.org) to tag input and output data with publicly accessible, community standards on domain-specific concepts such as "DNA repair" (GO:0006281) or "region" (SO:0000001). In this manner, one can formalize the semantics of data and services by formalizing the context using publicly available, extensible ontologies and logical relationships of those concepts to each other, and to individual resources.

Semantic web standards allow for interoperability, but without service protocols, they are not powerful enough for web-based integration. OWL, by itself, offers no standards on service discovery or invocation, nor any of the web service capabilities of SOAP, WSDL, and UDDI. It is thus not sufficient for high throughput integration: what is needed is a hybrid semantic web services architecture and protocol that brings semantics to the web service problem space.

1.3. The Need for Distributed Ontologies for Open World Web Services

For high throughput integration, web services and semantic representations are both necessary, but not sufficient, either alone, or taken together in a naive manner. One reason is the manner in which semantic operations like generalization and subsumption are handled in web services.

Ontologies in well delineated domains typically represent subsumption as nested subclass relations to organize concepts hierarchically. In fact, virtually all ontologies used today in biology and biomedicine (including the GO, and others being standardized through the Open Biomedical Ontologies (OBO, http://obo.sourceforge.net)) are essentially static subsumption hierarchies, with topological complexities rarely exceeding that of a Directed Acyclic Graph (DAG). These explicitly assert subclass relationships axiomatically (e.g. by using the `rdfs:subClassOf` predicate), rather than deriving subsumption dynamically (e.g. by inferring classes and subclasses from sets of individuals based on shared properties).

If the problem at hand is well defined, and if ontology maintenance, extension, and deprecation are centrally controlled, then such axiomatic subsumption can yield substantial value in semantically organizing information. But because they tend to be built from the top down, foundational concepts near the root are established early in the ontology's life cycle when there is the least amount of experience and real-world feedback on the utility of the ontological model. Changes to those terms established earliest in the project are unfortunately likely to cause the most extensive repercussions later on, thereby working against evolvability.

Moreover, there is antagonism between static subsumption and the need to allow distributed nodes to change without affecting or requiring change in other parts. Because static subsumption classes are defined in transitive subclass definitions, it is difficult for third parties to extend concepts unless the entire subsumption hierarchy mathces the concepts relevant for their problem at hand; i.e., the users' problem space has to match the ontology creator's world view up the chain of the hierarchy. Monolithic ontologies, residing as one ontology per file, create great difficulties for dynamic web services environments that seek access across ontologies on the per-term or per-concept basis, and it is inherently dangerous to claim axiomatic class subsumption to super-concepts outside of one's domain or control.

Thus static subsumption hierarchies fuel both rigidity (the inability to change to meet new demands) and fragility (the propensity to fail, often in multiple and seemingly unrelated places, under changes). We seek an ontological model that is robust to extension and re-use in an *open* world, and specifically one that encompasses, but does not limit, the pre-conceived subsumption relationships of the ontologist.

In an open world, statements not known to be either true or false are not assumed to be either. In a traditional closed world database, given the entry 'Jane:phoneNumber = 555-1212', the query "Is Jane's phoneNumber = 555-1234?" would return False. But an open world reasoner would conclude that the answer is unknown, allowing the possibility that Jane may have two telephones, and even that one may be 555-1234. In fact, if the reasoner was given the statements that Jane lives at 123 Any Lane, and that everyone who lives at 123 Any Lane has the telephone number 555-1234, then the reasoner would conclude, exactly contrary to the closed world database, that Jane's number is 555-1234, even without such an explicit statement.

1.4. The Simple Semantic Web Architecture and Protocol (SSWAP)

The preceeding shows that a hybrid semantic web services model is needed to address high-throughput integration on the web. To address this need, the Simple Semantic Web Architecture and Protocol[1] (SSWAP) was developed by one of the authors (Gessler) as part of an NSF-funded project for the Virtual Plant Information Network (http://vpin.ncgr.org). SSWAP uses OWL in a semantic web services framework to allow resources (data and services) to describe themselves in terms of publicly available, third-party ontologies. SSWAP specifies OWL classes, predicates, and their semantics to allow parties to discover, invoke, and pipeline web resources. Thus SSWAP provides the protocol that allows clients and providers to use semantics in a web services model.

As an OWL-based technology, SSWAP integrates knowledge in an open world. The use of SSWAP and OWL describing a resource creates a semantic network, a non-hierarchical set of RDF statements relating individuals, properties, and classes across web sites. SSWAP allows for a shared semantics of data and services by allowing resources to extend that network using third-party ontologies. For example, SSWAP allows the use of refactored OBO ontologies in semantic web services, while also allowing third-party extension of those ontologies or *de novo* introduction of new ontologies. Anyone can host an ontology, and to aid in finding and re-using them, a portal for these ontologies is provided at http://sswapmeet.sswap.info. As discussed above, these third-party ontologies are invariably static subsumption hierarchies, recalling the problem of open world reasoning in a semantic web services model as reasoning over a subset of statements taken from distributed, third-party subsumption hierarchies.

To give an example of the application and size of ontologies in biology, the GO contains nearly 20,000 terms grouped into three separate ontologies (Biological Process, Molecular Function, and Cellular Component) organized as taxonomies of nodes within the three major category headings. Once a gene is sufficiently characterized, it can be annotated to the appropriate node, as shown in Fig. 1 [6].

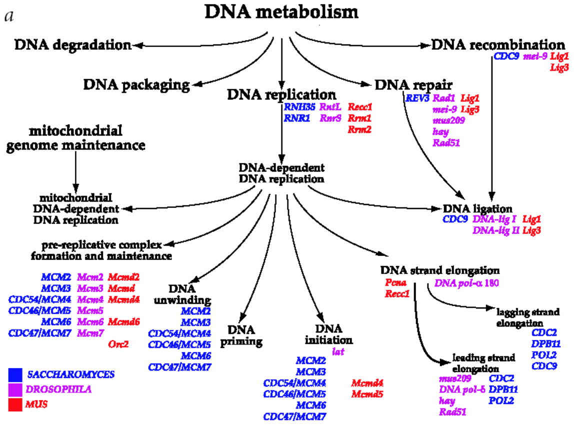

Figure 1. A portion of the GO [6], with genes annotated below the nodes.

Ontologies such as the GO were not designed for semantic web services, but they represent an important knowledge classification for the domain. To enable their use in web services, SSWAP is designed such that service discovery, requests, and responses reference ontological terms atomically, so that terms from disparate ontologies are referenced via their own URLs, similar to hyperlinks on a web page. Thus individual terms (concepts) can be used to classify the input and output data classes of the web resources. In this manner, SSWAP allows clients and providers to reuse legacy ontologies in a semantic web services framework.

Such semantic tagging enables semantic searching and pipelining. Consider an ontology that states that `classB` is a subclass of `classA`. Now consider a semantic web service `resourceA` that publishes that it accepts data of type `classA` upon which it performs some operation. The user then asks for all resources that operate on his/her data of type `classB`. Because `classA` is a superclass of `classB`, a semantically enabled search engine such as that

[1] http://sswap.info, http://ontologies.ncgr.org/sswap

running at http://sswap.info should return `resourceA`, since that resource is guaranteed to operate on data of type `classB` by virtue of the subsumption relationship between `classB` and `classA`. This is a distinguishing characteristic of semantic vs. pure lexical search engines.

SSWAP allows `resourceA` to accept data typed by one ontology, and return data typed by another. This is very common, e.g. one may use the concept of accession IDs from one ontology as keys into a database returning data semantically tagged by another ontology. Thus `resourceA` is operationally performing ontology alignment, mapping instances of classes from one ontology to those from another. Ontology alignment is a challenging field and few people have examined implications of using semantic web services as alignment agents, producing data-dependent, non-universal, mappings from one ontology to another.

Where ontologies provide typing information for *data* of web services, they can similarly provide typing information for the web services *themselves.* Consider the over 1600 database and web server entries of the annual Nucleic Acids Research (NAR) issues [14]. These web resources are classified into categories and subcategories, creating a static subsumption hierarchy of services. For example, the NAR subcategory "Java" is subsumed by the category "Computer Related" (www.oxfordjournals.org/nar/webserver/cat/1). SSWAP can use this classification to enable semantic searching and discovery: given a search for the key word "computer", the discovery server will find all "Computer Related" web resources, including the Java resources, even if none of them used the token "computer" in its definition.

Thus there are at least two classes of ontological relations relevant in open world semantic web services. The first is the application and deployment of traditional static subsumption heirarchies used to classify data. This is relevant for finding the appropriate services and joining the output of one service into the input of another. The second is the classification of the web resources *themselves*, either by low-throughput manual curation as done by NAR, or automated classification as done by a reasoner.

2. Order Theoretical Representations of Distributed Semantic Hierarchies

We need a formal grounding for knowledge integration over *distributed* semantic hierarchies, and thus turn to the *mathematics* of hierarchy, that is order theory, or the theory of lattices and ordered sets (posets) [1, 3]. Our research program [7, 8, 9, 10, 11, 12, 17, 18] is built on the representation of taxonomic ontologies as ordered sets. This technology encompasses not only existing ontology analysis mechanisms such as semantic similarities [2, 13], but also provides methods for ontology integration which are especially appropriate for the open-world modeling of semantic web services and SSWAP.

2.1. Order Theoretical Ontologies

A finite poset [3] is a mathematical structure $\mathcal{P} = \langle P, \leq \rangle$ where P is a finite set and $\leq \subseteq P^2$ is a reflexive, anti-symmetric, transitive binary relation on P. Posets are the most general mathematical structures admitting to description in terms of *levels* such as semantic generality. They are more specific than directed graphs or networks, since every poset is a DAG, and every DAG determines a unique poset by transitive closure. But they are also more general than trees (which most people normally take for general hierarchies) or lattices (which are used extensively in mathematics and computer science), in that a lattice is a poset where each pair of nodes has a unique parent, and a tree is a lattice where every single node has a unique parent.

The GO has the particular structure of being an **annotated multi-poset** [12]. It is a multi-poset because of the presence of both `is-a` subsumption links and `has-part` composition links, yielding on a single set of nodes P the two ordered sets $\langle P, \leq_{\texttt{is-a}} \rangle, \langle P, \leq_{\texttt{has-part}} \rangle$. Annotations then arise from a function $F \colon X \rightarrow 2^P$ mapping a gene $x \in X$ to a set of GO nodes $F(x) \subseteq P$. Fig. 1 shows this explicitly, although only for `is-a` links.

Some nodes $a, b \in P$ are **comparable** in that one is above or below the other, for example Leading Strand Elongation $\leq$ DNA Strand Elongration in Fig. 1. Others are **noncomparable**, for example Leading vs. Lagging Strand Elongation. Collections of comparable and noncomparable nodes form chains and antichains respectively, whose sizes and compositions comprise the "vertical" and "horizontal" structure of the poset. In general, the "chain decomposition" of a (bounded) poset, or a sub-interval of a poset $[a, b] = \{c : a \leq c \leq b\}$ determined by two comparable nodes $a \leq b \in P$, provides information about its structure.

2.2. Ontology Measurement and Display

As an illustrative example of the order theoretical approach, we have been working on the **categorization** task in the GO, where a biomedical researcher wants to take the names of hundreds or thousands of genes which have been annotated to the GO and gain an understanding of their overall function by examining their distribution of their annotations: are they localized, grouped in distinct areas, or spread uniformly? Our POSet Ontology Categorizer [10] (POSOC, http://www.c3.lanl.gov/posoc) takes a list of genes of interest, and calculates a score for each node in the GO to represent how well that node best captures the overall distribution and location of the query in the poset (see Fig. 2). This score depends on a vertical pseudo-distances $\delta(a, b)$ from a node b to the comparable nodes a below it on chains where annotations sit, determined as a function of the properties of the collection of lengths of the chains in the chain decomposition of intervals $[a, b]$.

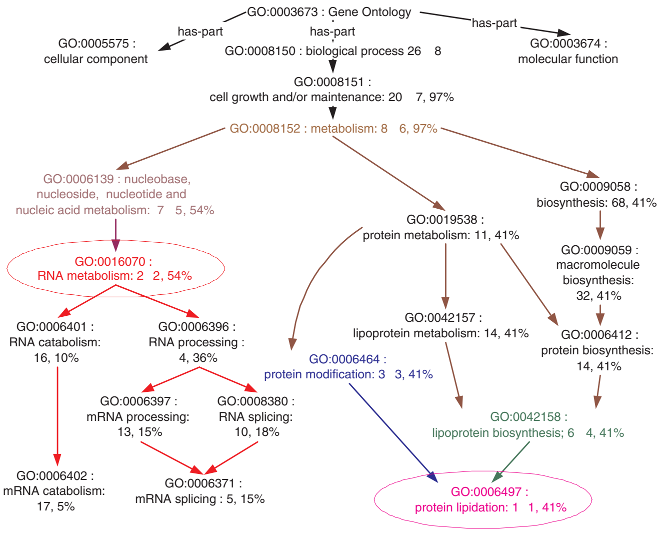

Figure 2. Partial output from POSOC for a sample query [10].

We have also been using chain decompositions to generalize notions of vertical distance in posets and to optimize layout and display [11]. Fig. 3 shows a small upper-bounded poset on the left, and on the right the same structure with a lower bound $0 \in P$ added, and laid out according to the chain decomposition. Vertical rank is assigned as an interval-valued concept (e.g. the rank of F is $[2, 4]$, that of E is $[3, 3]$) based on the maximum chain lengths from *both* the top $1 \in P$ *and* the bottom $0 \in P$, relative to the total height (size of largest chain), in this case e.g. $0 \prec D \prec E \prec I \prec B \prec 1$, yielding 6. Horizontal layout of a node a is then a function of the chain lengths in the intervals $[a, 1]$ and $[0, a]$.

Traditional approaches to ontology analysis assign weights (relative frequencies) to the nodes of a semantic hierarchy, and calculate mutual information between nodes as a measure of "semantic similarity" [2, 13]. We are generalizing these approaches to develop metrics within hierarchies measuring distances between nodes. For example, in Fig. 2, we can determine that the distance between the GO nodes 16070 and 6497 is $d(16070, 6497) = |\uparrow 16070| + |\uparrow 6497| - 2|\uparrow 16070 \cap \uparrow 6497| = 6 + 12 - 2 \times 4 = 10$, where $\uparrow a = [a, 1] = \{b \in P : \ b \geq a\}$ is the "up-set" of a.

2.3. Concept Lattices

Another key technology in the order theoretical approach is Formal Concept Analysis (FCA) [5], a set of mathematical methods to represent the hierarchical relationships and implications present among relational data represented as sets of objects and their properties. FCA defines a **formal context** as a mathematical structure $K = \langle G, M, I \rangle$, where G is a set of objects (individuals), M is a set of their attributes (properties), and $I \subseteq G \times M$ is a relation such that saying $\langle g, m \rangle \in I$ means that "object g has attribute m".

	Breast	Colorectal	CaMP < 1	CaMP ≥ 1	CaMP ≥ 1.5
SKIV2L	√	√	√		
C6orf29		√		√	
SLC29A1		√		√	√

Table 1. A small formal context.

Consider the small example formal context K shown in Table 1, summarizing five attributes $m \in M$ of three genes $g \in G$ [15]. "Breast" and "Colorectal" indicate if the gene was associated with tumors from those tissues, while "CaMP" is a numerical Cancer Mutation Prevalence score determined by the authors of [15], where CaMP > 1 indicates a likelihood of it contributing to the cancer.

Identifying relevant patterns such as which collections of genes are either highly, somewhat, or not implicated in either breast, colorectal, or both forms of cancer involves sorting the table by rows and columns multiple times to "move checkboxes together". Some patterns are obvious: all genes are associated with colorectal cancer; others are more subtle: every gene associated with breast tumors are also associated with colorectal tumors, yet do not induce disease. This trivial example is pedagogical, but as the number of objects and attributes increases (the real dataset involves thousands of genes), the complexity of permuting and sorting such tables to find all patterns becomes combinatorically overwhelming, and the ability to scan them to identify relevant patterns approaches very real limitations.

FCA automatically constructs all possible permutations of rows and columns to identify maximal rectangles of checkboxes, and then organizes these into a **concept lattice**, a semantic hierarchy which dually catalogs both the collections of objects which have certain attributes and collections of attributes which hold for certain objects. The example concept lattice is shown in Fig. 4. Nodes contain information about "precise" facts in the table, such as the fact that SKIV2L is the only gene which is both breast and not disease implicated. Following the links obtains other information, for example, going up from the object SKIV2L shows that it is also colorectal, and this exhausts the information about it. Similarly, one can examine attributes such as CaMP ≥ 1, to show that C6orf29 is disease-implicated, but traversing downward we see that so is SLC29A1 (so that C6orf29 has only CaMP > 1, and not 1.5). Going back, each of these two genes is colorectal-associated, and only colorectal-associated.

These are facts obtained by examining the relations between objects and attributes. Other information is available by looking at the relations between objects and objects or attributes and attributes. For example, among the genes

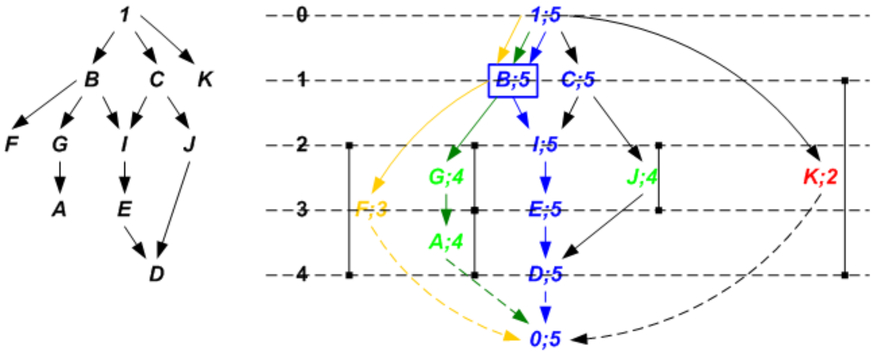

Figure 3. (Left) A small poset. (Right) Its chain decomposition layout.

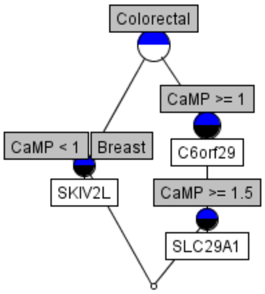

Figure 4. Concept lattice of K in Table 1.

present in this pedagogical formal context "breast implies colorectal" since the breast attribute is below the colorectal attribute (this is not true for the full data set). Finally, information is available about what pairs or groups of objects and attributes are in common. For example, to see what SKIV2L and SLC29A1 have in common, we traverse upwards from both until we arrive at their join at "colorectal": not only are they both implicated in colorectal tumors, but also that is the only thing they have in common.

Note that all the information in the lattice is also available from the table, and indeed, one can derive either from the other exactly. The advantages of the lattice are that it is a visual representation, it is a summary representation of all the information in the table, and it is unbiased with respect to the default ordering of the rows and columns in the table.

But most significantly, while there is not necessarily any explicit hierarchical structure in the context K, the concept lattice derives the hierarchical relations among the attributes implicit in the structure of their objects. The resulting concept lattices represent ontological subsumption hierarchies derived from data. In our small example, Colorectal is a more general category than Breast. As in annotations to hand-made ontologies like the GO, genes are grouped by virtue of their shared properties, and thus those collections of shared properties represent ontological categories, growing more general (encompassing more objects) as one ascends the hierarchies. But unlike GO, in an FCA representation the classes are derived automatically from experimental properties of the data; and furthermore, all classes can be proven to be logically consistent and complete.

3. FCA for Integration and Induction

Note that each concept lattice is itself a poset of the form $\langle P, \leq \rangle$ described above, and is thus ammenable to analysis in terms of pseudo-distances and lattice metrics as described in Section 2.2. But an even more significant feature of FCA is its ability to *integrate* both hierarchical and non-hierarchical information from multiple sources to create a common ontological representation. If one of those information sources is already a semantic hierarchy, the result is a concept lattice that preserves the ordering of the hierarchical concepts while simultaneously inducing a compatible order on the non-hierarchical concepts. FCA is thus becoming widely recognized for the value it brings to tasks in ontology induction, management, and interoperability [16], as well as for FCA-derived semantic similarity measures [4].

We will illustrate this with an example which first shows ontology integration, and furthermore does so in the context of integrating semantic information about web resources *themselves*, as discussed in Sec. 1.4. Consider a web service ontology using NAR categories and subcategories, in particular a class of services "Human Genes and Diseases" (HGD) with two sub-classes "Cancer Gene Databases" and "General Polymorphism Databases". SSWAP records that the Protein Mutant Database (PMD, http://pmd.ddbj.nig.ac.jp), the Human Potential Tumor Associated Antigen database (HPtaa, http://www.bioinfo.org.cn/hptaa), and the Human Genome Mutant Database (HGMD, http://www.hgmd.cf.ac.uk/ac/index.php) are known to be appropriate for each of these categories respectively.

This results in the Web Services Ontology (WSO) fragment shown in Fig. 5, with HPtaa and HGMD also being

	Cancer Genes	General Polymorphim	HGD
HGMD		√	√
HPtaa	√		√
PMD			√

Table 2. Context K_{WSO}.

	Tumor	Mutation	Antigen
HPtaa	√		√
PMD		√	
MTB	√	√	

Table 3. Context K_{KW}.

annotated to HGD by inheritance. This can then in turn be represented in the small formal context K_{WSO} shown in Table 2. In this simple example, the resulting concept lattice shown on the left side of Fig. 6 is pretty much the original ontology fragment back again, although in general the concept lattice can be very different, and in particular new annotations can be induced through the combination of the annotation structure and the inheritance structure of $\mathcal{P}$ [12].

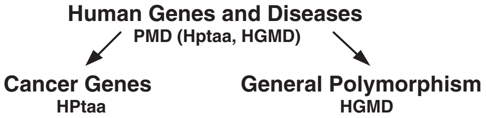

Figure 5. Web Services Ontology fragment

But consider that I have *additional* information about *some* of the same resources from another source without an inheritance structure. For example, SSWAP metadata resources records the keywords "tumor" and "antigen" associated with HPtaa; and "mutation", but neither "tumor" nor "antigen", with PMD. Additionally, the Mouse Tumor Database (MTB, http://tumor.informatics.jax.org) is associated with the keywords "tumor" and "mutation", but not "antigen". The formal context is shown as the context K_{KW} in Table 3, similar to our previous cancer example from Table 1 and Fig. 4. Its concept lattice is shown in the center for Fig. 6, and note now the *induction* of a hierarchical, ontological structure on these resources in virtue of their combinations of keywords.

The fact that web resources are shared between these contexts allows the construction of the integrated formal context K shown in Table 4, whose concept lattice is shown on the right of Fig. 6. The extraction of the hierarchical relations among the attributes of the joint lattice in turn allows the construction of a joint ontology, shown in Fig. 7.

	CG	GP	HGD	Tumor	Mutation	Antigen
HGMD		√	√			
HPtaa	√		√	√		√
PMD			√		√	
MTB				√	√	

Table 4. Joint context K.

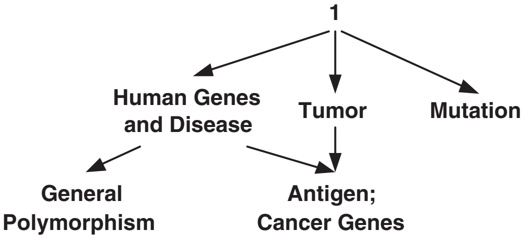

Figure 7. Resultant joint ontology.

Both of the hierarchical structures in the left and center of Fig. 6 are reflected in the joint concept lattice on the right side. For example, on the right we have Cancer Genes $\leq$ Human Genes and Disease; and General Polymorphism $\leq$ Human Genes and Disease, as on the left; and Antigen $\leq$ Tumor as in the center. In this way, joint concept lattices are effective methods for merging both distinct taxonomies, and taxonomies with other relational structures. This is because the different information sources reflected in K_{WSO} and K_{KW} have no overlap of their columns.

But in some cases, parts of the original orders are violated. In the center of Fig. 6 we have MTB $\leq$ PMD, whereas on the right these are non-comparable. That's because PMD is used in both K_{WSO} and K_{KW}, and in different ways: in K_{KW}, MTB is both Tumor and Mutation, while PMD is only Mutation, yielding MTB strictly more specific than PMD in the center. But in K_{WSO} we are told something more about PMD, namely, that it is annotated to the category Human Genes and Disease, but we have no such information about MTB. Thus MTB is no longer strictly more specific than PMD, so on the right, it is moved distinctly away from MTB into it's own atomic "object concept".

Beyond that, the joint lattice does identify additional information amongst constituent members. For example, the category Cancer Genes is both identified with the keyword Antigen, and also placed below the Tumor keyword in a hierarchical relationship. Thus this method both respects the hierarchical structures present in constituents, induces new hierarchical relations from relational data sources, and integrates these into a new, general hierarchical structure.

4. Conclusion

The challenges to high-throughput integration occur at the level of semantics, logic, authentication, and trust; with these abstract issues being implemented on top of an established technology stack including HTTP. Thus knowledge integration in open worlds is unlikely to be solved using tra-

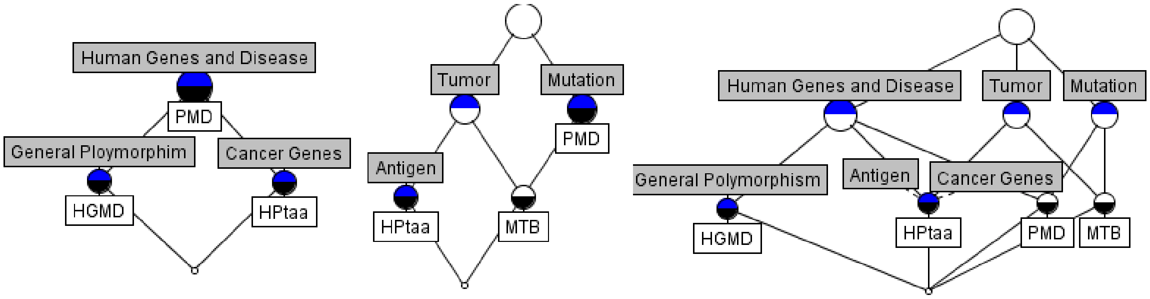

Figure 6. Concept lattices for (left) K_{WSO}, (center) K_{KW}, and (right) K.

ditional, top-down technologies, where, for example, shared semantics are clearly defined and universally applied. It is more likely that we will be challenged to reason on an inconsistent, non-decidable semantics that describes web resources – data and services – that span the gamut of the ephemeral to the established. These resources are presented to us not on an implementation expressly designed for this problem, but on the proven infrastructure and under the best practices commonly recognized as the World Wide Web.

But also, solving the "integration equation" in the context of web resources will require mathematical and computational models not yet well developed. It is this latter observation that motivates our discussion on the mathematics of hierarchical structure.

References

[1] Birkhoff, Garrett: (1940) *Lattice Theory*, v. **25**, Am. Math. Soc., Providence RI, 3rd edition

[2] A Budanitsky and G Hirst: (2006) "Evaluating WordNet-based Measures of Lexical Semantic Relatedness", *Computational Linguistics* , v. **32**:1, pp. 13-47

[3] BA Davey and HA Priestly: (1990) *Introduction to Lattices and Order*, Cambridge UP, 2nd edition

[4] Formica, Anna: (2006) "Ontology-Based Concept Similarity in Formal Concept Analysis", *Information Sciences*, v. **176**, pp. 2624-2641

[5] Ganter, Bernhard and Wille, Rudolf: (1999) *Formal Concept Analysis*, Springer-Verlag

[6] Gene Ontology Consortium: (2000) "Gene Ontology: Tool For the Unification of Biology", *Nature Genetics*, v. **25**:1, pp. 25-29

[7] CA Joslyn: (2004) "Poset Ontologies and Concept Lattices as Semantic Hierarchies", in: *Conceptual Structures at Work, LNAI*, v. **3127**, ed. Wolff, Pfeiffer and Delugach, pp. 287-302, Springer-Verlag, Berlin

[8] CA Joslyn and WJ Bruno: (2005) "Weighted Pseudo-Distances for Categorization in Semantic Hierarchies", in: *Conceptual Structures: Common Semantics for Sharing Knowledge, LNAI*, v. **3596**, ed. F Dau, M-L Mugnier, G Stum, pp. 381-395

[9] Joslyn, Cliff; Gessler, DDG; Schmidt, SE; and Verspoor, KM: (2006) "Distributed Representations of Bio-Ontologies for Semantic Web Services", in: *Joint BioLINK and 9th Bio-Ontologies Meeting (JBB 06)*

[10] Joslyn, Cliff; Mniszewski, Susan; Fulmer, Andy; and Heaton, G: (2004) "The Gene Ontology Categorizer", *Bioinformatics*, v. **20**:s1, pp. 169-177

[11] Joslyn, Cliff; Mniszewski, SM; Smith, SA; and Weber, PM: (2006) "SpindleViz: A Three Dimensional, Order Theoretical Visualization Environment for the Gene Ontology", in: *Joint BioLINK and 9th Bio-Ontologies Meeting (JBB 06)*

[12] Kaiser, Tim; Schmidt, Stefan; and Joslyn, Cliff: (2006) "Concept Lattice Representations of Annotated Taxonomies", to appear in LNAI, CLA 06

[13] PW Lord, R Stevens, A Brass, C Goble: (2003) "Investigating Semantic Similarity Measures Across the Gene Ontology: the Relationship Between Sequence and Annotation", *Bioinformatics*, v. **10**, pp. 1275-1283

[14] *Nucleic Acids Research*, 2006, Vol. 34, Web Server issue W1, http://nar.oxfordjournals.org/content/vol34/suppl_2/index.dtl

[15] T Sjöblom; S Jones; LD Wood; DW Parsons et al.: (2006) "Consensus Coding Sequences of Human Breast and Colorectal Cancers", *Science*, v. **10.1126**, `science.1133427`

[16] Stumme, G; Studer, Rudi; and Sure, York: (2000) "Towards an Order-Theoretical Foundation for Maintaining and Merging Ontologies", in: *Verbundtagung Wirtschaftsinformatik*, pp. 136-149, Singapore

[17] KM Verspoor, J Cohn, CA Joslyn, SM Mniszewski, A Rechtsteiner, LM Rocha, T Simas: (2004) "Protein Annotation as Term Categorization in the Gene Ontology Using Word Proximity Networks", in: *BMC Bioinformatics*, v. **6**:s1

[18] Verspoor, KM; Cohn, JD; Mniszewski, SM; and Joslyn, CA: (2006) "A Categorization Approach to Automated Ontological Function Annotation", *Protein Science*, v. **15**, pp. 1544-1549