Joslyn, Cliff A: (2000) "Hypergraph-Based Representations for Portable Knowledge Management Environments: A White Paper", Los Alamos Technical Report LAUR = 00-5660

# Hypergraph-Based Representations for Portable Knowledge Management Environments: A White Paper

**Dr. Cliff Joslyn**

Distribtued Knowledge Systems and Modeling Team
Modeling, Algorithms, and Informatics Group (CCS-3)
MS B265, Los Alamos National Laboratory
Los Alamos, NM 87545
`joslyn@lanl.gov, http://www.c3.lanl.gov/~joslyn`

November, 2000

## DRAFT
## DISTRIBUTED FOR COMMENT ONLY NOT FOR ATTRIBUTION

### Abstract

In this brief white paper we define **semantic hyperwebs** as labeled, weighted, directed hypergraph-theoretical structures similar to weighted conceptual graphs; propose them as a canonical framework for representation of knowledge and semantic information in Distributed Knowledge Systems; and discuss technologies to support implementation. We are hopeful that such structures and implementations will be useful in the development of robust, portable environments for knowledge management, development, elicitation, and exchange. This paper is being distributed for comment, and not intended to be comprehensive, nor even especially novel. Rather we hope to solicit guidance from the community and help point the way to hopefully help develop and/or integrate off-the-shelf technologies which can be brought to bear on the problem at hand.

## Contents

# 1 Introduction: Towards Portable Knowledge Management Environments

In this document we briefly describe some preliminary ideas for the development of hypergraph-theoretical representations, and their implementations, for portable Knowledge Management (KM) environments.

## 1.1 Desiderata and Tasks for KM Environments

Many groups are currently focused on the problem of being able to develop knowledge management systems for Distributed Knowledge Systems (DKS) which are portable from one organization to another. This is obviously especially difficult because different organizations have different data types and content, let alone different needs, goals, and organizational cultures. As noted by Johan de Kleer at the Xerox PARC KM workshop in April, 2000, there is a high specificity in each system, coupled with an overall lack of theory. As noted by Daniel Bobrow also at that time, we should attempt to adapt technology to the actual communities, seeding knowledge bases with initial content and structure, while granting public recognition to community members involved in specifying the environment to the organization.

Instead of moving a KM *system* from one organization to another, we want to consider construction of a KM *environment* which can be introduced into multiple organizations, and then semi-automatically instantiated to that organization. Such an environment would contain a variety of components, from standard database and computer-science tools useful for many purposes, to more sophisticated representations and methods which might be appropriate for some particular organization's needs. Moreover, this environment should support communities of knowledge agents (human and/or computational) with tasks such as knowledge representation, elicitation, exchange, and negotiation.

Desirable characteristics of such environments thus include:

**Portability:** An environment to be instantiated, not a system to be installed.

**Flexibility:** The ability to support multiple tasks and representations, depending on the local need.

**Theoretical Soundness:** Rooted in a solid mathematical basis, or at least describable in formal representations.

**Balance Between Generality and Richness:** The environment should have a minimally sufficient degree of complexity to accomplish its tasks without over-design, both for theory and implementation.

**Interpretation Independence:** KM systems can be focused on single representations or interpretations, such as ontological representations of semantic relations among concepts. A formally-rooted system will support multiple interpretations by allowing the representation of arbitrary relations among any components in a DKS (e.g. attribution or historical relations among documents, generic meta-data based linking, author-based "guides" through a high dimensional corpus, etc.).

**Small, Variable Structures:** Technologies should support not just large, monolithic representations of huge corpora, but also small structures used in collaborative knowledge sharing and agent communication and ontology negotiation.

**Design and Discovery:** Ideally, there should be support not just for human authoring of knowledge structures, but also for knowledge *discovery* (bottom-up, induced relations).

The kinds of tasks which can be supported by such a KM environment include:

**Hyperdocument Construction:** Authoring and analysis of individual semantically enhanced, nonlinearly structured hybrid hyper-documents and document collections.

**Corpus Analysis, Management, and Representation:** Representation, manipulation, and visualization of corpora as collections of documents connected by structural links (citation, web links) and annotated by semantic keywords. Standard tasks include organization and retrieval, and also customization and recommendation to users and communities of users [45].

**Ontological Representation:** Facilities for Communities Of Practice (COPs) to construct and manipulate representations of their specific ontological structures, and support for sociological methodologies for the self-elicitation of same [34, 35, 40].

**Natural Language Processing:** Representation of syntactic structures of and semantic relations in text (e.g. document abstracts).

**Agent Negotiation and Collaboration:** Facilities for agents (human or computational) to share not just information structures through a common protocol, but also *knowledge structures* which provide the semantic basis for the *interpretation* of those structures [44].

**Data Mining:** Graph clustering and other data mining methods (e.g. sequence analysis, network analysis, latent semantic analysis, and other statistical tools) to uncover hidden patterns and structural relations.

## 1.2 Representation of Semantic Information

Of course, KR frameworks have been pursued for decades. What is new today is the combination of new formal frameworks and available computational power. Beyond that is the understanding that we can move towards the representation of *semantic* information in DKS, that is, further information about the content and interpretation of other pieces of information in our knowledge systems. An example is Berners-Lee's call for a "semantic web" [5, 7, 46] (although Joslyn introduced that term earlier to denote a particular formal structure useful in this context [23]), referring to the next level of internet-like DKS.

The discussion of **semantic information theory** in general is very deep, and we won't go into it too much here [2, 14, 21, 24, 25]. However, we will note that the movement to semantic information involves at a minimum a meta-syntactic labeling to distinguish information tokens as to their meaning, use, or interpretation. Examples include:

- Keywords to distinguish subjects and topics of documents, as used in digital libraries and other corpora.
- Meta-data to distinguish data format, provenance, and content, as is being widely used in modern knowledge exchange environments [42].
- Labeled or typed relations to distinguish semantic categories and relations among concepts, as used in ontology representations environments.

## 1.3 Syntheses: Formal and Representational

Another aspect of the modern environment appears to be the ability to draw from the multiple fundamental formalisms which underly modern information systems, including:

**Relational:** Mathematical relations of the form $R \subseteq \bigtimes_i X_i$, where the $X_i$ are multiple dimensions (sets). Such structures provide a canonical meta-language for mathematical systems [30], and are the foundation for relational databases.

**Logical:** The traditional frameworks of AI, including frame semantics, rule bases, and description logics [41, 48].

**Graph Theoretical:** Including traditional semantic networks [47] and other diagrammatic representations such as entity-relation and dataflow diagrams [17, 19].

**Object-Oriented (OO):** Mirroring modern software engineering methods, including inheritance hierarchies diagrams and DAGs, and the UML approach to meta-meta-modeling.

It should be noted that many of these representational frameworks are homomorphic or isomorphic under certain conditions. For example, mathematical relations, systems of logical propositions, and OO hierarchies have standard graph theoretical representations [48]; and logical systems, OO hierarchies, and graphical structures are reducible to certain kinds of mathematical relations $R$.

Building on the ability to integrate these mathematical bases, there also appears to be a coming synthesis in the community in terms of the recognition of the needed properties in these environments and possible mathematical representations, merging data objects and logic in a variety of distributed environments. This can be seen in a summary of ontology (or knowledge) representation languages which has been provided by Corcho and Gómez-Pérez [10] in Fig. 1. Note the great similarities in capabilities, and while logical features are emphasized, OO concepts are also prominent.

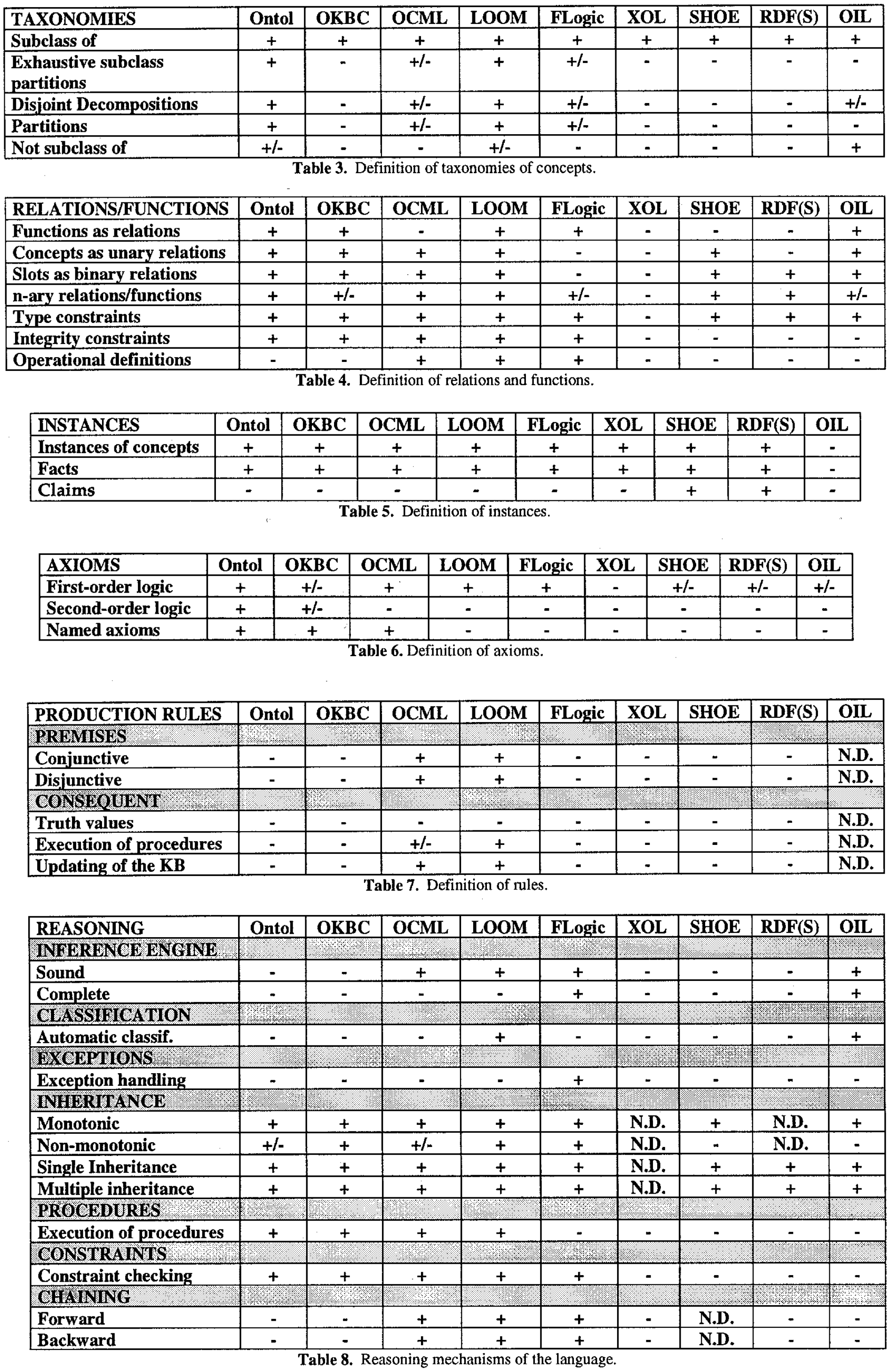
### 1.4 Notes on This White Paper

As an aside, let me here note a couple of things about the context of this paper.

- This document is one of exploration, soliciting comment and input. I am aware that a vast array of KM tools, methods, and formalisms are available and being brought to bear from both the academic and corporate communities (e.g. [6, 10, 27, 41, 48]). Indeed, we are becoming more aware of these every day, and this document should in no way be seen as attempting to be comprehensive, either about existing KM tools or about underlying graph-theoretical methods. As I learn more about the current state of the art, I hope I will discover to what extent I am recapitulating existing work, and update this document accordingly.
- Similarly, it is my sincere hope that much of what I'm discussing and proposing here is, in fact, *not* novel, but rather that we would be able to buy or download facilities such as envisioned here off the shelf. Barring that, we would like to be able to acquire a more general environment which could be instantiated to arrive at the ideas described here. Barring *that*, we would like to team with like-minded developers within the KM community to help construct such an environment.
- I should also stress that while much of what is discussed here is abstract, most of these ideas have developed over a number of years of considering what would be useful for a personal tool for my own KM needs. In other words, I really want to have what I'm proposing here for its own sake, and for my own purposes.
- Finally, we here address almost entirely issues of *representational* environments, over and above mechanisms for *inferencing*, or the production of new knowledge relative to an existing corpus of knowledge, within such environments. We are, however, aware of this issue and existing inference methods, and simply note the possible role of:
  - Production systems for logical representations
  - Projection, extension, join [26] and closure for relational representations
  - Graph matching and manipulation for graphical representations

## 2 A "Semantic Hyperweb" Approach

As mentioned, it is well known that there are a number of distinct formalisms available for representing knowledge structures, including mathematical relations, frames and rules, object-oriented



| TAXONOMIES | Ontol | OKBC | OCML | LOOM | FLogic | XOL | SHOE | RDF(S) | OIL |
|---|---|---|---|---|---|---|---|---|---|
| Subclass of | + | + | + | + | + | + | + | + | + |
| Exhaustive subclass partitions | + | - | +/- | + | +/- | - | - | - | - |
| Disjoint Decompositions | + | - | +/- | + | +/- | - | - | - | +/- |
| Partitions | + | - | +/- | + | +/- | - | - | - | - |
| Not subclass of | +/- | - | - | +/- | - | - | - | - | + |

**Table 3.** Definition of taxonomies of concepts.

| RELATIONS/FUNCTIONS | Ontol | OKBC | OCML | LOOM | FLogic | XOL | SHOE | RDF(S) | OIL |
|---|---|---|---|---|---|---|---|---|---|
| Functions as relations | + | + | - | + | + | - | - | - | + |
| Concepts as unary relations | + | + | + | + | - | - | + | - | + |
| Slots as binary relations | + | + | + | + | - | - | + | + | + |
| n-ary relations/functions | + | +/- | + | + | +/- | - | + | + | +/- |
| Type constraints | + | + | + | + | + | - | + | + | + |
| Integrity constraints | + | + | + | + | + | - | - | - | - |
| Operational definitions | - | - | + | + | + | - | - | - | - |

**Table 4.** Definition of relations and functions.

| INSTANCES | Ontol | OKBC | OCML | LOOM | FLogic | XOL | SHOE | RDF(S) | OIL |
|---|---|---|---|---|---|---|---|---|---|
| Instances of concepts | + | + | + | + | + | + | + | + | - |
| Facts | + | + | + | + | + | + | + | + | - |
| Claims | - | - | - | - | - | - | + | + | - |

**Table 5.** Definition of instances.

| AXIOMS | Ontol | OKBC | OCML | LOOM | FLogic | XOL | SHOE | RDF(S) | OIL |
|---|---|---|---|---|---|---|---|---|---|
| First-order logic | + | +/- | + | + | + | - | +/- | +/- | +/- |
| Second-order logic | + | +/- | - | - | - | - | - | - | - |
| Named axioms | + | + | + | - | - | - | - | - | - |

**Table 6.** Definition of axioms.

| PRODUCTION RULES | Ontol | OKBC | OCML | LOOM | FLogic | XOL | SHOE | RDF(S) | OIL |
|---|---|---|---|---|---|---|---|---|---|
| PREMISES | | | | | | | | | |
| Conjunctive | - | - | + | + | - | - | - | - | N.D. |
| Disjunctive | - | - | + | + | - | - | - | - | N.D. |
| CONSEQUENT | | | | | | | | | |
| Truth values | - | - | - | - | - | - | - | - | N.D. |
| Execution of procedures | - | - | +/- | + | - | - | - | - | N.D. |
| Updating of the KB | - | - | + | + | - | - | - | - | N.D. |

**Table 7.** Definition of rules.

| REASONING | Ontol | OKBC | OCML | LOOM | FLogic | XOL | SHOE | RDF(S) | OIL |
|---|---|---|---|---|---|---|---|---|---|
| INFERENCE ENGINE | | | | | | | | | |
| Sound | - | - | + | + | + | - | - | - | + |
| Complete | - | - | - | - | + | - | - | - | + |
| CLASSIFICATION | | | | | | | | | |
| Automatic classif. | - | - | - | + | - | - | - | - | + |
| EXCEPTIONS | | | | | | | | | |
| Exception handling | - | - | - | - | + | - | - | - | - |
| INHERITANCE | | | | | | | | | |
| Monotonic | + | + | + | + | + | N.D. | + | N.D. | + |
| Non-monotonic | +/- | + | +/- | + | + | N.D. | - | N.D. | - |
| Single Inheritance | + | + | + | + | + | N.D. | + | + | + |
| Multiple inheritance | + | + | + | + | + | N.D. | + | + | + |
| PROCEDURES | | | | | | | | | |
| Execution of procedures | + | + | + | + | - | - | - | - | - |
| CONSTRAINTS | | | | | | | | | |
| Constraint checking | + | + | + | + | + | - | - | - | - |
| CHAINING | | | | | | | | | |
| Forward | - | - | + | + | + | - | N.D. | - | - |
| Backward | - | - | + | + | + | - | N.D. | - | - |

**Table 8.** Reasoning mechanisms of the language.

Figure 1: Comparison of ontology markup environments [10].

type hierarchies, semantic networks and conceptual graphs, and many others [48]. To a large extent, isomorphic and homomorphic transformations are available mapping these into each other, and so the choice of a particular methodology depends on the relative tradeoffs of ones purposes. It is wise to choose methods which are general enough to be flexible and portable to serve many purposes, but specialized enough to be actually *useful* for those purposes.

## 2.1 The Value of Graph-Theoretical Representations

Graph theoretical representations have been standard and crucial for knowledge systems and computer science in general for decades. Graphs as collections of nodes and relations are used to represent a variety of concepts: frames and slots, entities and relations, nodes and arcs, concepts and semantic relations, arguments and predicates, objects and attributes, types and properties, or classes and states.

Formalisms which use graph theoretical representations are especially important for our purposes, as they both capture important inherent properties of DKS, and provide a valuable human interface. Their long legacy goes back to semantic networks and before [20, 48]. The birth of hypertext moved graph theoretical representations into individual documents. Entity-relation diagrams, dataflow diagrams, and others are central to information science in general [17]. Our colleagues [34, 35, 40] work with such graphical structures as scratch nets and factor complexes as representations for COPs to self-elicit their knowledge structures. And the growth of the Web, merged with bibliometrics, have placed an emphasis on graph formalisms for representing DKS in general [1, 16, 28, 29, 50].

The software development community is also definitely moving in this direction. The current HTTP protocol for the Web is decidedly graph-theoretical, but primitively so, mapping to the most basic of directed graphs. Other attempted hypertext standards (e.g. HyTime [38]) had more sophisticated approaches, with typed-link hypertexts mapping to labeled graphs. Other developing standards support yet more sophisticated graphical models (see Sec. 3.2). Graph theoretical tools are also being developed for corpus and web management [9, 37].

Joslyn has proposed semantic webs [23] as directed, acyclic multigraphs. This structure has value in striking the balance between generality and usefulness, being minimally sufficiently complex to represent multiple, interacting loosely hierarchical structures within one corpus.

## 2.2 Towards Semantic Hyperwebs

Here we suggest a particular hypergraph-theoretical knowledge representation and software tool to support a common set of activities within a portable KM environment. To reiterate some of the comments from Sec. 1.1:

- Our goal is to invoke a general enough formal structure so as not to commit to any particular interpretation or highly specified existing system, for example Cyc's upper ontology [11, 32, 33], or to any particular KM task, such as ontology or natural language representation or meta-data editing [42].
- Similarly, we wish to not commit to any particular logic or inference system, since the task of representation is at least prior to inference, if not sufficient for some of our tasks.

- We wish to have the ability to construct small, specialized knowledge structures which can be shared amongst agents in collaborative knowledge exchange environments.
- Nevertheless, the structures should have sufficient richness and potential complexity to be able to represent both ontological and non-ontological structures in the same framework, e.g. citation networks and the Web, entity-relation diagrams, natural language parse trees, or any other diagrammatic structure (e.g. scratch nets or factor complexes).

We would like to offer **labeled, weighted, directed hypergraphs** as this foundational structure for KR in DKS. In such structures:

- Nodes represent concepts, documents, keywords, agents, or any other entities in DKS.
- Directed hyperedges of order $n$ represent whatever $n$-ary semantic relations exist among nodes.
- Labels, where present, indicate the kind of semantic relation.
- Weights (numeric attributes in $[0, 1]$), where present, indicate the strength of the semantic relation.

When the nodes and links are included in inheritance hierarchies, we will call these structures **semantic hyperwebs**. We postulate that semantic hyperwebs as equivalent to **weighted conceptual graphs** [48].

We motivate the move towards semantic hyperwebs by building up structures with the following characteristics:

**Nodes and Relations:** First establish the basic graph structure by defining a universe of discourse $X$ with $a, b \in X$, a relation $R \subseteq X^2$, and a directed graph $G$ on $X$. A link directly represents an element of the relation:

$$a \longrightarrow b \qquad \Longleftrightarrow \qquad \langle a, b \rangle \in R$$

where $a \longrightarrow b$ is an edge in $G$.

**Labeled Edges as Binary Propositions:** Advancing to labeled graphs [3], the labeled links (edges) represent propositions, where the label on the edge represents the predicate and the nodes the arguments:

$$a \xrightarrow{f} b \qquad \Longleftrightarrow \qquad f(a, b)$$

for some predicate $f$.

**Labeled Hyperedges as $n$-ary Propositions:** Representing higher-order relations or propositions requires a directed hypergraph, a structure $\mathcal{H} = \langle X, E \rangle$ where $E$ is a collection of directed hyperedges $\vec{e} = \langle x_1, x_2, \ldots, x_n \rangle$, where $x_i \in X$ and $n$ is the order of $\vec{e}$ [4, 8, 36, 49]. The order of $\mathcal{H}$ is the maximal order of the $e$. Thus a directed graph is a directed hypergraph of order two. And finally, a directed hyperedge with label $f$ of order $n$ thus represents the $n$-ary proposition $f$.

Diagramming directed hypergraphs can be difficult [49]. The left side of Fig. 2 shows the unordered hyperedge $\{a, b, c\}$ simply as the label of the subset. A directed hyperedge has the form of a vector $\langle a, b, c\rangle$. The center of Fig. 2 shows its diagrammatic representation, where the numbers indicate the position of each argument in the proposition $f(a, b, c)$. The "tentacle" representation is commonly used, as shown in the right of Fig. 2, basically transforming it into a higher-order bipartate graph where the predicates and arguments are distinguished, and the new edges are labeled by the argument position.

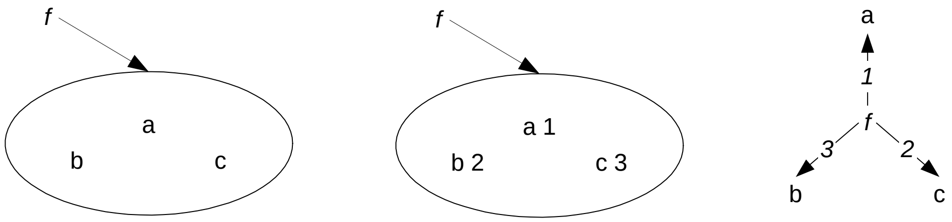


Figure 2: (Left) The unordered hyperedge $\{a, b, c\}$. (Center) The ordered hyperedge $\langle a, b, c\rangle$. (Right) The "tentacle" representation of $\langle a, b, c\rangle$.

**Inheritance Hierarchies:** Link types (labeled edges) are of central importance, and correspond to a data typing mechanism. Individual link types can represent semantic categories, ontological or semantic relations, predicates relating arguments (their nodes), or any other relation typing needs. As data types, it is natural that they exist in a multiple inheritance hierarchy, and available through an OO database. Thus the nodes and relations need to be related in lattices. Node hierarchies represent sub-typing, whereas relation hierarchies represent inheritance of relational properties.

Link types should be parameterized by their relational properties, including at least reflexivity, transitivity, symmetry, and the "antis" of those. These should be available both for identification purposes ("how transitive is this sub-graph?"), and for construction of closures ("save the transitive closure of this graph").

An example of such a link-type hierarchy is shown in Fig. 3, from Sowa [48]. Here he shows the possible classes of formal relations. Similarly, Fig. 4 shows the semantic relations possible among a class of verb-types in the mono-stratal linguistic theory of Davis and Koenig [12, 13].

Such hierarchies are the backbone of ontological representations, as in the Cyc project [11, 33]. Another example drawn from Sowa [48] is shown in Fig. 5, demonstrating how it is possible in complex hierarchies to represent both abstract relation types and concrete statements of knowledge.

## 2.3 A Labeled, Directed Graph Example

Fig. 6 shows an example knowledge representation. This diagram is of a labeled directed graph (neither weighted nor a hypergraph) where both the nodes and edges exist in type hierarchies. It is derived from Fensel *et al.* [15] as an example of a structure representable in the Ontology Interface Layer (OIL). Source code is shown in Tab. 1 on p. 16.

Some features of the diagram include:

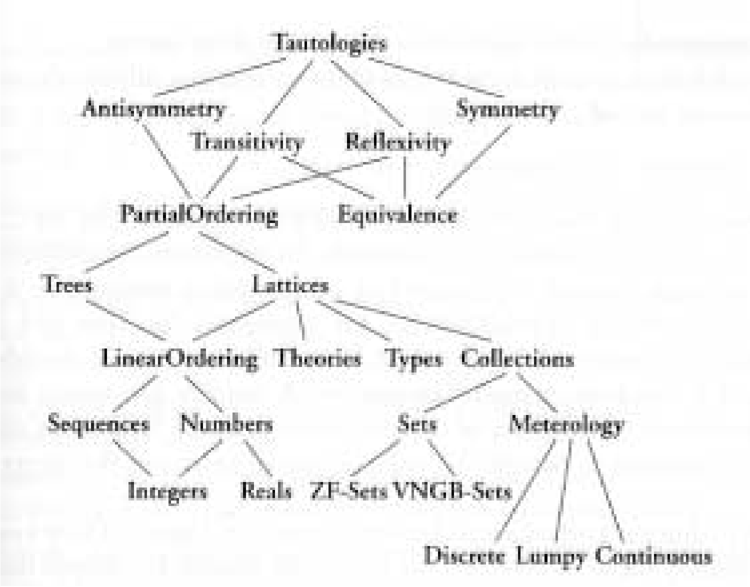


Figure 3: Hierarchies of theories [48].

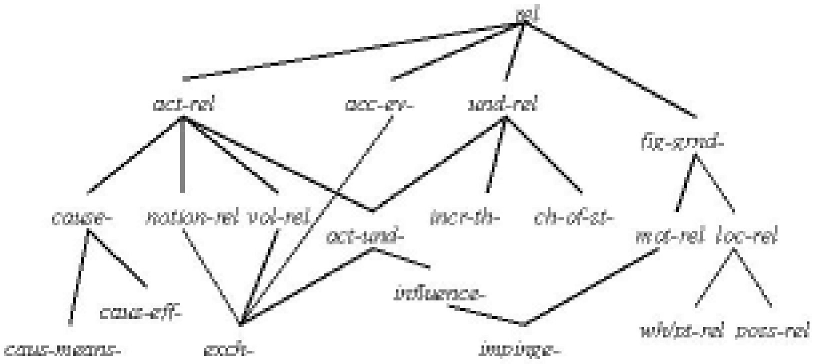


Figure 4: Inheritance relations among verb types [12].

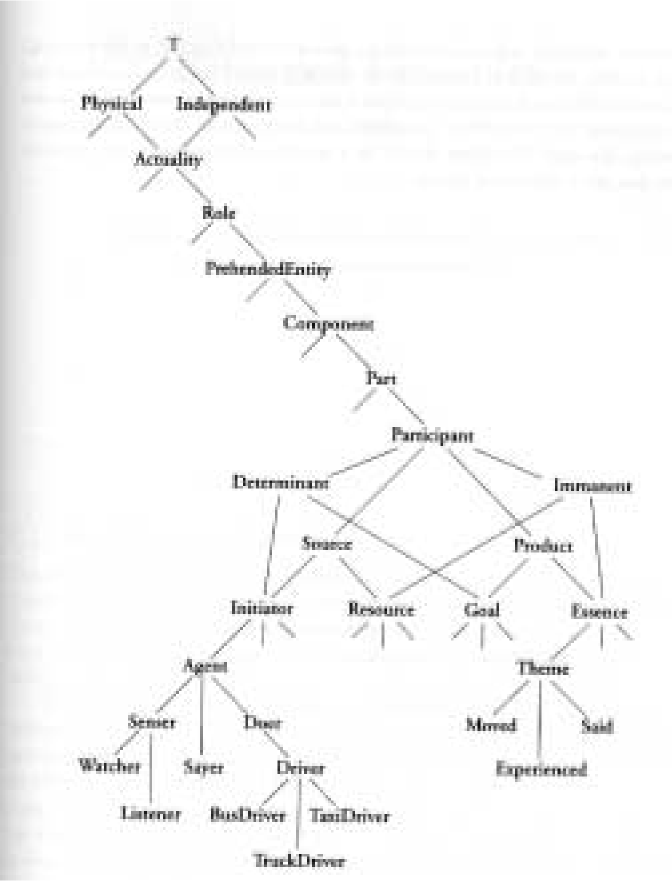


Figure 5: From abstract to specific semantic relations [48].

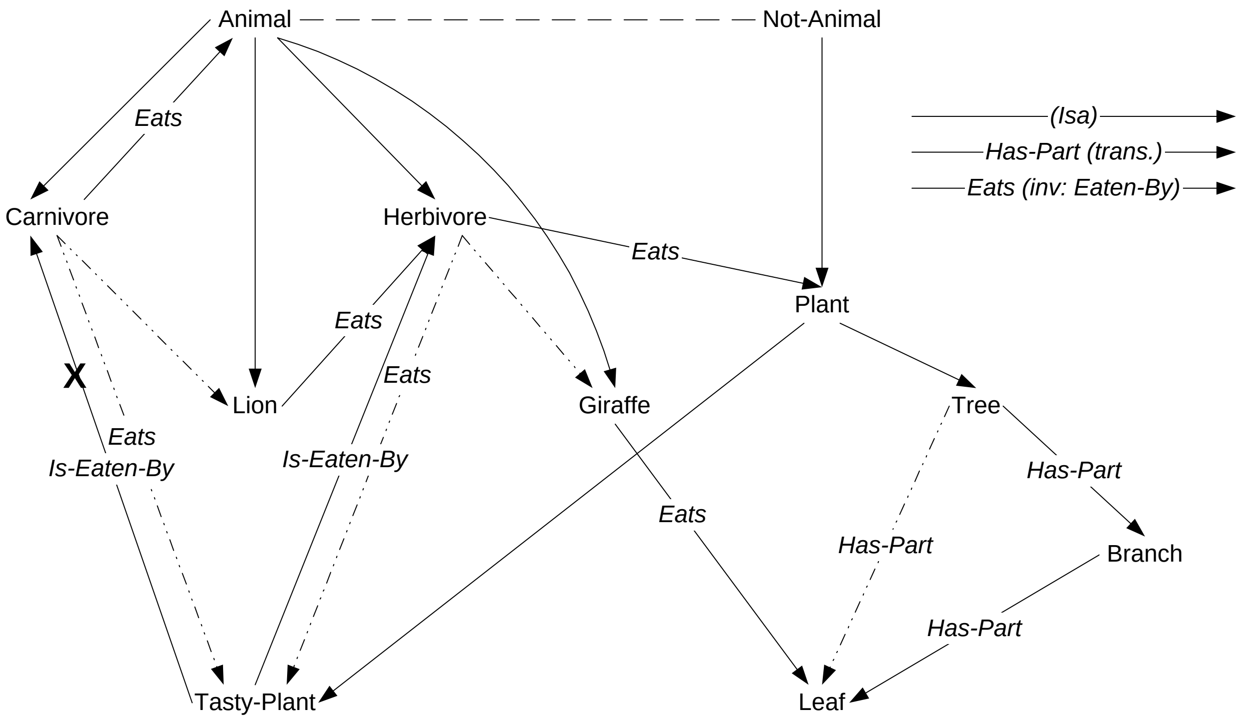


Figure 6: Labeled-graph knowledge representation example, derived from [15].

- Inferred links are indicated by dashed edges.
- Negation is also supported, as in the specification of the class of non-animals.
- The node hierarchy is indicated by the unlabeled links (links of "type null"), and indicated in the key by (Isa). Composition of the relevant isa and eats relations allows for inference of isa(lion,carnivore) and isa(giraffe,herbivore).
- "Has-part" is a standard ontological category, and a high-level link type in Sowa's hierarchy in Fig. 3. This property allows inference of has-part(tree,leaf).
- "Eats" is a specific ontological category, and would exist deep within either of the hierarchies of Fig. 3 or Fig. 5. Its possession of an inverse "is-eaten-by" allows other inferences, for example that is-eaten-by(tasty-plant,carnivore) is inconsistent, because eats(carnivore,animal) and isa(tasty-plant,plant) and isa(plant,not-animal).

## 2.4 Conceptual Graphs and Semantic Hyperwebs

We can define semantic hyperwebs formally:

**Definition 1 (Semantic Hyperweb)** A semantic hyperweb of order $n$ is a structure $\mathcal{S} = \langle \mathcal{C}, \mathcal{L} \rangle$, where $\mathcal{C} = \{C\}$ is a lattice of nodes and $\mathcal{L} = \{\mathcal{H}\}$ a lattice of directed, weighted hypergraphs of maximal order $n$ on $\mathcal{C}$.

The abstract nodes $C \in \mathcal{C}$ live in an inheritance hierarchy. The hypergraphs $\mathcal{H} \in \mathcal{L}$ also live in an inheritance hierarchy, and each represents a different semantic relation whose arity is that of the order of $\mathcal{H}$.

**Note 2** By itself, $\mathcal{L}$ is a labeled, weighted, directed hypergraph constrained by the lattice ordering relations among the $\mathcal{H} \in \mathcal{L}$.

**Definition 3 (Semantic Web)** A semantic hyperweb of order 2.

**Note 4** Many simpler structures are recovered for special cases, for example if there are no weights, or no labels, or if the structure is of order 2. For example semantic networks are recovered for an order 2 structure with no weights, a fuzzy relation or graph for an order 2 structure with no labels, and a Bayes net for additively constrained fuzzy relations. And of course, simple, unlabeled, and unweighted graphs are supported by default.

However, conceptual graphs (CGs) have already been defined by Sowa as structures very similar to what we're proposing. The following definition is an interpretation and to some degree a simplification from [48].

**Definition 5 (Conceptual Graph (Simplified) [48])** A CG is a structure $\mathcal{G} = \langle \mathcal{C}, \mathcal{R}, G \rangle$, where $\mathcal{C}$ is a lattice of concepts, $\mathcal{R}$ a lattice of relations, and $G \subseteq \mathcal{C} \times \mathcal{R}$ is a bipartate graph on $\mathcal{C}$ and $\mathcal{R}$.

As with semantic hyperwebs, the concepts $C \in \mathcal{C}$ live in an inheritance hierarchy. Each relation $(R_i \in \mathcal{R}) \subseteq C^{j_i}$, where $j_i \in \{1, 2, \ldots\}$ is the dimensionality of $R_i$, represents a different semantic relation.

CGs capture most of the key characteristics we require: a lattice of node types and a lattice of relation types which are defined on them. Indeed, we believe that semantic hyperwebs capture all the properties of CGs and then some, all within a more elegant framework.

**Conjecture 6** Each semantic hyperweb determines a unique conceptual graph. Conversely, each conceptual graph determines a class of semantic hyperwebs equivalent to weights.

The conjecture basically states that a CG is equivalent to an unweighted semantic hyperweb. The path to a proof involves mapping the conceptual relations $R \in \mathcal{R}$ to the labels of the hyperedges of $\mathcal{L}$.

The issues where we part company with CGs include:

- Semantic webs are weighted structures.
- The bipartate representation of a CG echoes the "tentacle" representation of a hypergraph. The difference is that all the tentacles together comprise a single hyperedge, whereas in the bipartate graph these have to be called out distinctly.
- Similarly, hypergraphs are closer to the typed-link hypertext model. Here now only concepts are represented as vertices, and relations of different types are assigned edges of different types (labels), resulting in a labeled or multigraph mathematical structure.

This can be seen in two partial examples. A CG for the statement "a person is between a rock and a hard place" is shown in Fig. 7 [48]. The equivalent directed, unweighted hypergraph form is shown in Fig. 8 in its non-tentacle form. The tentacle form is essentially identical to Fig. 7, where the circular conceptual relation nodes become the labels of the directed hyperedges.

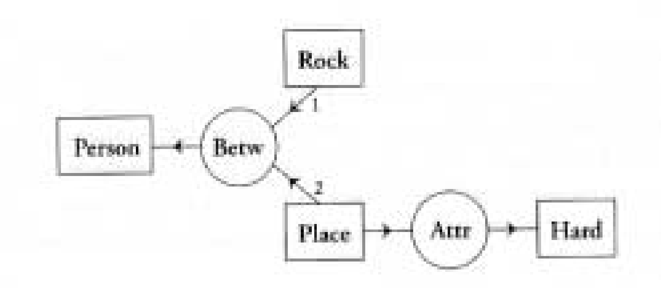


Figure 7: A conceptual graph [48].

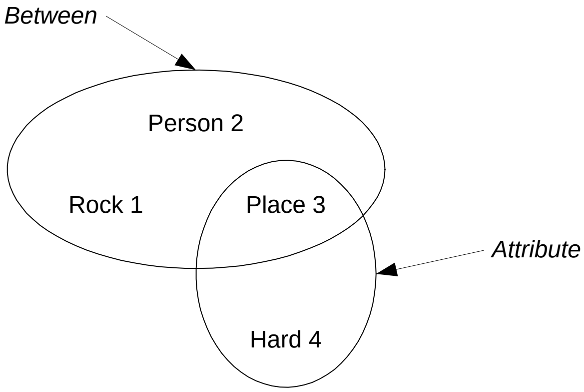


Figure 8: A non-tentacle form of a directed hypergraph representation of the CG from Fig. 7.

# 3 Implementation Design

We have in mind coupling these mathematical structures with a particular, relatively quickly realizable implementation.

## 3.1 Front End

First, there is a need for a general, robust front end, not something highly specialized like the Java Ontology Editor (JOE) [22], or Protege [42]. The following features are desired:

- A full drawing/visualization environment for both simpler node-arc diagrams and more complex hypergraph form.

- Rubberbanding
- A professional GUI: full undo/redo, a full audit trail (time stamping, authoring, versioning), graphical and textual annotation, etc.
- Building on commercial platforms for graph editing and on supported protocols both at the intersection of graph markup and KR.

Recent work at LANL collaboratively with the University of Liverpool has begun development building from the Visio platform for Windows. Visio is a superb environment for working with reasonably sized labeled graphs, and can be adapted to the tentacle representation of hypergraphs and CGs.

### 3.2 Back Ends

We envision multiple back ends. Two necessary but subsidiary ones are:

- A relational or OO database, as is provided in the TSA-1/Liverpool Visio work.
- Specialized graph theoretical representations, for example for Matlab or for specialized graph theoretical algorithms.

But the most important back end is a full read/write markup language facility. The community has developed a series of markup environments over the years, in chronological order:

**Hypertext:** HTTP/HTML, representing directed graphs.

**Typed-link Hypertext:** HyTime [38], representing labeled, directed graphs.

**Graph Markup Language (GML):** Representing general graphs [18].

**XML/XML-Schema:** This is the current environment of choice for serial representations of graphical structures [6].

**RDF/RDF-Schema:** This is the current hot standard, representing labeled directed graphs, and recovering typed-link hypertext through URL markup, as shown in Fig. 9 [31, 43].

**Ontology Interface Layer (OIL):** OIL [39] is a very exciting standard being developed which appears to capture unweighted semantic webs (unweighted second order semantic webs) completely, in addition to being a generic markup environment for graphs in general. We therefore propose that our development be done in this context. OIL code for the example in Fig. 6 is shown in Tab. 1 [15].

```
ontology-container
  title "African animals"
  creator "Ian Horrocks"
  subject "animal, food, vegetarians"
  description "A didactic example ontology describing African animals"
  description.release "1.01"
  publisher "I. Horrocks"
  type "ontology"
  format "pseudo-xml"
  format "pdf"
  identifier "http://www.cs.vu.nl/ dieter/oil/TR/oil.pdf"
  source "http://www.africa.com/nature/animals.html"
  language "OIL"
  language "en-uk"
  relation.hasPart "http://www.ontosRus.com/animals/jungle.onto"
ontology-definitions
  slot-def eats
    inverse is-eaten-by
  slot-def has-part
    inverse is-part-of
    properties transitive
  class-def animal
  class-def plant
    subclass-of NOT animal
  class-def tree
    subclass-of plant
  class-def branch
    slot-constraint is-part-of
      has-value tree
  class-def leaf
    slot-constraint is-part-of
      has-value branch
  class-def defined carnivore
    subclass-of animal
    slot-constraint eats
      value-type animal
  class-def defined herbivore
    subclass-of animal
    slot-constraint eats
      value-type
        plant OR
        (slot-constraint is-part-of has-value plant)
  class-def giraffe
    subclass-of animal
    slot-constraint eats
      value-type leaf
  class-def lion
    subclass-of animal
    slot-constraint eats
      value-type herbivore
  class-def tasty-plant
    subclass-of plant
    slot-constraint eaten-by
      has-value herbivore, carnivore
```

Table 1: OIL code for the example in Fig. 6 [15].

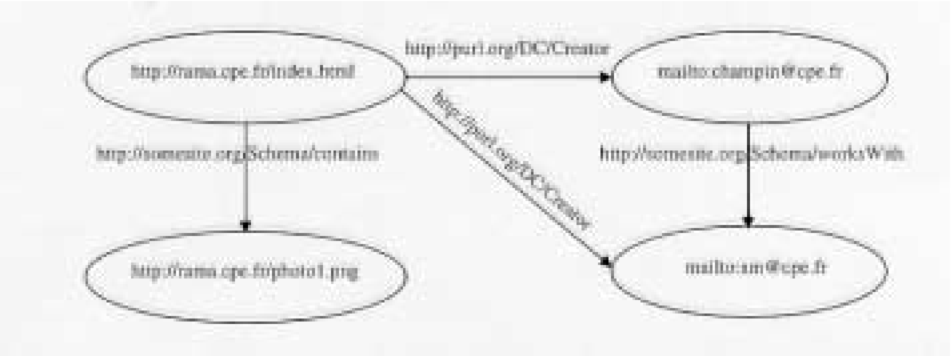


Figure 9: Labeled graph representation in RDF [43].

### 3.3 Graph Theory Support

The tool or environment envisioned should provide ample support for graph theoretical techniques. I am currently becoming more familiar with hypergraph theoretical methods, but can list a number of standard techniques for analyzing and manipulating regular graphs:

- Subgraph extraction: by link type, node type, $n$-neighborhood, or a cut-level of weight.
- Dual graph construction
- Principle component analysis
- Cycle-finding ("Is this graph cyclic? Show me the cycles. Reduce the cycles to new meta-nodes")
- Chain-finding ("Show the linear chains from $A$ to $B$")
- Shortest path
- General and standard graph statistics
- Metricity calculations
- Planarity optimization
- Clustering
- Root and leaf finding
- Level-finding (when the graph is DAG)
- Morphisms, equality testing, distance measures ("how similar are these graphs? can I twist this one into that one?")

Some examples of this kind of activity are shown. In Fig. 10 from Paton [40] we see transformation of graphs into various structures. For example, $A$ transforms the graph at the bottom of the

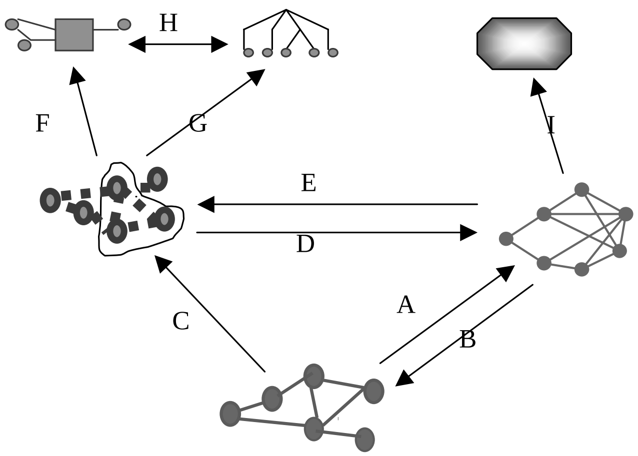


Figure 10: Transformations among graphs [40].

figure to its line graph form, indicating connections among the edges (now considered as nodes themselves), and $C$ is the identification of a clique.

Two other examples are drawn from an earlier paper of ours [23]. Fig. 11 shows cycle finding and "cyclic reduction". The cycle $b \longrightarrow c \longrightarrow f \longrightarrow d \longrightarrow b$ in a graph $R$ is first identified and then replaced with the single meta-node $Y$ and meta-link $a \longrightarrow Y$, recovering a new DAG with the level sets identified.

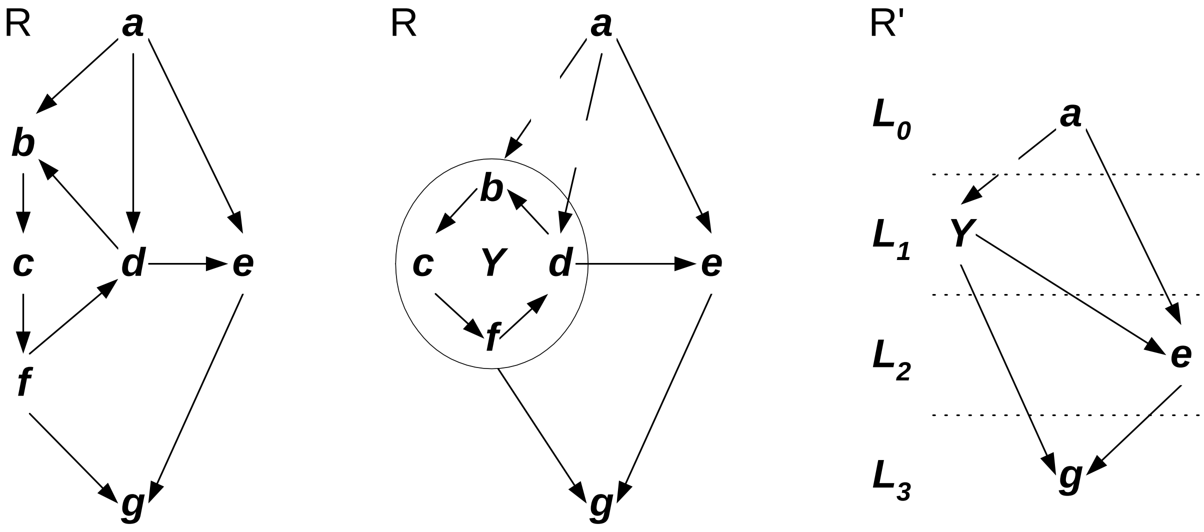


Figure 11: (Left) A directed graph. (Center) Cycle identification. (Right) Cyclic reduction to a DAG [23].

In Fig. 12 [23] $M$ is an acyclic multirelation: a directed labeled graph (multigraph) with two colored edges and no cycles in any one relation type. In [23] we proposed these structures for representing multiple interleaved hierarchies in DKS. $R(M)$ is the reduction of the two relations to a single (unlabeled) directed graph. On the bottom of Fig. 12 the two interleaved relations are separated.

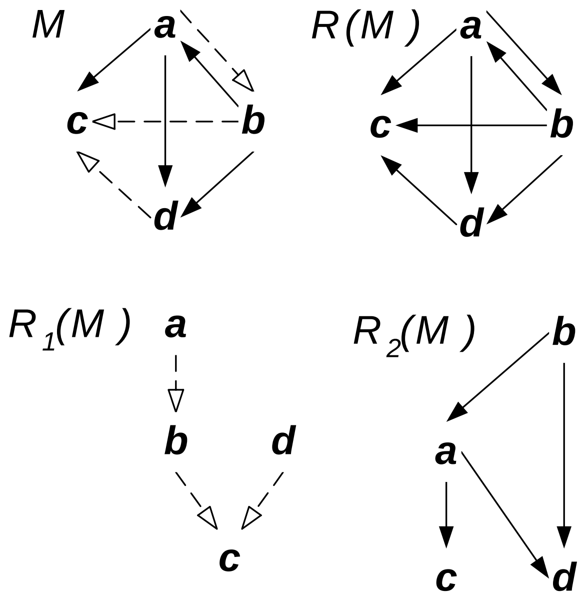


Figure 12: (Top left) A multigraph (labeled graph). (Top right) Reduction to a directed graph. (Bottom) Separation of relations [23].

## References

[1] Adamic, Lada A: (1999) "The Small World Web", in: *Proc. 3rd European Conf. on Research and Advanced Technology for Digital Libraries*, Paris, `http://www.parc.xerox.com/iea/www/smallworld.html`

[2] Bar-Hillel, Y and Carnap, Rudolph: (1952) "Semantic Information", *British J. for the Philosophy of Science*, v. **4**, pp. 147-157

[3] Barrett, Chris; Jacob, Riko; and Marathe, Madhav: (2000) "Formal Language Constrained Path Problems", *SIAM J. of Computing*, v. **30**:3, pp. 809-837

[4] Berge, C: (1973) *Graphs and Hypergraphs*, North-Holland, London

[5] Berners-Lee, Tim: (1998) "Semantic Web Road Map", WW3 Consortium, `http://www.c3.org/DesignIssues/Semantic.html`

[6] Berners-Lee, Tim; Connolly, Dan; and Swick, Ralph R: (1999) "Web Architecture: Describing and Exchanging Data", WW3 Consortium, `http://www.w3.org/1999/04/WebData`

[7] Berners-Lee, Tim and Fischett, Mark: (1999) *Weaving the Web*, Harpers, New York

[8] Boros, Endre; Gurvich, Vladimir; and Khachiyan, L et al.: (2000) "Generating Partial and Multiple Traversals of a Hypergraph", *Lecture Notes in Computer Science*, v. **1853**, pp. 588-599

[9] *The Brain*, `http://www.thebrain.com`

[10] Corcho, Oscar and Gomez-Pere, Asuncion: (2000) "Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages", in: *2000 Workshop on Applications of Ontologies and Problem-Solving Methodologies*, `http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/3.ps`

[11] *Cycorp: Creators of the Cyc Knowledge Base*, `http://www.cyc.com`

[12] Davis, Anthony R: (2000) *Types and Constraints for Lexical Semantics and Linking*, Cambridge UP

[13] Davis, Anthony R and Koenig, J: (2000) "Linking as Constraints on Word Classes in a Hierarchical Lexicon", *Language*, v. **76**:1, pp. 56-91

[14] Dretske, Fred: (1982) *Knowledge and the Flow of Information*, MIT Press, Cambridge

[15] Fensel, D; Horrocks, J; and van Harmelen, F et al: (2000) "OIL in a Nutshell", in: *2000 Workshop on Applications of Ontologies and Problem-Solving Methodologies*, `http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/4.pdf`

[16] Gibson, David; Kleinberg, Jon; and Raghavan, Prabhakar: (1998) "Inferring Web Communities from Link Topology", in: *Proc. 9th ACM Conf. on Hypertext and Hypermedia*

[17] Glasgow, Janice; Narayanan, NHA; and Chandrasekar, B, eds.: (1995) *Diagrammatic Reasoning: Cognitive and Computational Perspectives*, MIT Press

[18] *Graph Markup Language (GML)*, `http://www.infosun.fmi.uni-passau.de/Graphlet/GML`

[19] Harel, David: (1988) "On Visual Formalisms", *Communications of the ACM*, v. **31**:5

[20] Heylighen, Francis: (1998) "Bootstrapping Knowledge Representations: From Entailment Meshes via Semantic Nets to Learning Webs", *Int. J. Human-Computer Studies*, submitted,

[21] Hoffmeyer, Jesper: (1996) *Signs of Meaning in the Universe*, Indiana UP, Bloomington IN

[22] *JOE: The Java Ontology Editor*, `http://www.engr.sc.edu/research/CIT/demos/java/joe/joeBeta.html`

[23] Joslyn, Cliff: (1996) "Semantic Webs: A Cyberspatial Representational Form for Cybernetics", in: *Proc. 1996 European Conf. on Cybernetics and Systems Research*, v. **2**, pp. 905-910, `ftp://wwwc3.lanl.gov/pub/users/joslyn/emcsr96.pdf`

[24] Joslyn, Cliff: (2000) "Levels of Control and Closure in Complex Semiotic Systems", in: *Annals of the New York Academy of Sciences*, v. **901**, ed. J. Chandler, G. van de Vijver, pp. 67-74, NY Academy ofScience, New York

[25] Joslyn, Cliff: (2001) "The Semiotics of Control and Modeling Relations in Complex Systems", *Biosystems*, `ftp://ftp.c3.lanl.gov/pub/joslyn/biosystems.pdf`, in press

[26] Joslyn, Cliff and Mniszeiski, Susan: (2001) "DEEP: Data Exploration through Extension and Projection", *Knowledge Discovery and Data Mining*, in preparation

[27] *Ongoing KBS/Ontology Projects and Groups*, `http://www.cs.utexas.edu/users/mfkb/related.html`

[28] Kleinberg, Jon M: (1998) "Authoritative Sources in a Hyperlinked Environment", in: *Proc. 9th ACM-SIAM Symp. on Discrete Algorithms*, `http://simon.cs.cornell.edu/home/kleinber/auth.pdf`

[29] Kleinberg, Jon M: (2000) "Navigation in a Small World", *Nature*, v. **406**

[30] Klir, George: (1991) *Facets of Systems Science*, Plenum, New York

[31] Lassila, Ora: (1997) "Introduction to RDF Metadata", W3 Consortium, `http://www.w3.org/TR/Notes-rdf-simple-intro`

[32] Lenat, Douglas B: (1998) "The Dimensions of Context Space", Cycorp, `http://www.cyc.com/context-space.doc`

[33] Lenat, Douglas B; Ramanathan, V Guha; and Pittman, K et al.: (1990) "Cyc: Towards Programming with Common Sense", *Communications of the ACM*, v. **33**:8, pp. 30-49,

[34] Meyer, MA; Butterfield, KB; and Murray, WS et al.: (2001) "Guidelines for Eliciting Expert Judgment as Probabilities or Fuzzy Logic", in: *Fuzzy Logic and Probability Applications*, ed. Tim Ross, ASA, forthcoming

[35] Meyer, MA and Paton, RC: (2001) "Interpreting, Representing and Integrating Scientific Knowledge from Interdisciplinary Projects", *Int. J. Interdisciplinary Studies*, forthcoming

[36] Minas, Mark: (2000) “Hypergraphs as a Unifrom Diagram Representation Model”, in: *Proc. 6th Int. Workshop on Theory and Applications of Graph Transformations*, `ftp://ftp.informatik.uni-erlangen.de/local/inf2/Papers/tagt98.pdf`

[37] *Nestor: The Web Browser and Cartographer*, `http://www.irpeacs.fr/~zeiliger/nestor.htm`

[38] Newcomb, Steven R; Kipp, Neil A; and Newcomb, Victoria T: (1991) “HyTime: Hypermedia/Time-Based Document Structuring Language”, *Communications of the ACM*, v. **34**:11, pp. 67-83

[39] *Ontology Interface Laryer*, `http://www.ontoknowledge.org/oil/`

[40] Paton, Ray: (1999) “Collections, Systems and Mathematical Metaphors”, in: *Proc. Computing Anticipatory Systems 1999*

[41] Paton, Ray: (2000) “Ontologies and Representations: An Interim Summary”, *LANL Technical Report*,

[42] *The Protege Project*, `http://smi-web.stanford.edu/projects/protege`

[43] *Resource Description Framework*, `http://www.w3.org/RDF`

[44] Richards, Diana; McKay, Brendan D; and Richards, Whitman A: (1998) “Collective Choice and Mutual Knowledge Structures”, *Advances in Complex Systems*, v. **1**:2-3, pp. 221-236

[45] Rocha, Luis M and Bollen, Johan: (2000) “Biologically Motivated Distributed Designs for Adaptive Knowledge Management”, in: *Design Principles for the Immune System and Other Distributed Autonomous Systems*, ed. I Cohen and L Segel

[46] *The Semantic Web*, `http://www.semanticweb.org/`

[47] Shastri, Lokendra: (1988) *Semantic Networks*, Morgan Kaufman, Los Angeles

[48] Sowa, John F: (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole, Pacific Grove

[49] Wang, Yang and Wong, Andrew KC: (1996) “Representing Discovered Patterns Using Attributed Hypergraph”, in: *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, ed. E Simondies, J Han, U Fayyad, pp. 283-286, AAAI Press, Menlo Park CA

[50] Watts, Duncan and Strogatz, Steven H: (1998) “Collective Dynamics of ’Small World’ Problems”, *Nature*, v. **393**, pp. 440-442