Predicting How Ontologies for the Semantic Web Will Evolve

Henry M. Kim

Schulich School of Business, York University, 4700 Keele St., Toronto, Ontario Canada M3J 1P3,

hkim@schulich.yorku.ca | (416) 736-2100 x77952 | (416) 736-5687 [fax]

I. Introduction

According to Tim Berners-Lee, the WWW will evolve towards the *Semantic* Web:

> "To date, the World Wide Web has developed most rapidly as a medium of documents for people rather than of information that can be manipulated automatically. By augmenting Web pages with data targeted at computers and by adding documents solely for computers, we will transform the Web into the Semantic Web.
>
> Computers will find the meaning of semantic data by following hyperlinks to definitions of key terms and rules for reasoning about them logically. The resulting infrastructure will spur the development of automated Web services such as highly functional agents." [1]

However, what if not enough people represent machine process-able information at all, or not richly enough, or not in numbers sufficient, to make these services viable? The "information" that appears to be the bottleneck for the adoption of the Semantic Web is not data; it is not '7' or 'cat.' It is the rules and meanings about data defined precisely enough so that machines, not slow, error prone humans, can correctly interpret and quickly process that data; it is information like 'sabbaticals occur every 7 years,' or 'cats and dogs are mammals.' For the Semantic Web, ontologies from the AI field are envisaged to codify this information [1].

Therefore, the future of the Semantic Web is linked with the future of ontologies on the Semantic Web. In order to predict the future of these ontologies, why not look at the history of something similar. As opposed to models to codify information on the WWW, what about going far back and examining models that codified information on paper? The aim of this paper is to examine evolution of paper-based systems, explain it using a conceptual model of system evolution, apply the model to web-based systems analogs to paper-based systems, and finally project what happened with paper-based systems to make predictions about how ontologies will evolve, if at all, to make Berners-Lee's vision of the Semantic Web viable.

II. Evolution of Business Forms

In simple paper (e.g. memos) and HTML use, the author is responsible for authoring, and the reader, interpretation and processing. Paper dissemination requires a mechanically enabled physical infrastructure symbolized by the printing press; HTML dissemination, an electronically enabled virtual infrastructure symbolized by the Internet.

However information is disseminated, the human mind's processing capacity is small relative to the size of the problems requiring processing for an objective solution. Simon [2] calls this *bounded rationality*. Fox [3] states that bounded rationality compels humans or processors to seek techniques to reduce complexity in information, task, and coordination. His model of evolution of organizational structures to reduce complexity can be applied to explain the evolution of paper based information manipulation.

Information is too complex when it requires more processing than available in order to be properly analyzed and understood [3]. This complexity is reduced by omission and abstraction. When numerous simple documents need to be examined, requiring them all to be interpreted and processed by the reader is too taxing, especially when they are poorly written, or contain unnecessary or incomplete information. An omission strategy forces the author to only submit sets of information required for processing. An abstraction strategy allows sets of information to be abstracted from one document so that processing can be performed on a set rather than the whole document. For paper documents, this strategy is executed using business forms, which delineate the structure of the document from its contents.

According to Barnett [4], the first business form was a form letter for dispensation of sins, developed by Gutenberg himself in 1454. What used to be the responsibility of a simple paper document author was decomposed into forms design, and forms data entry. Designers were unlikely to be entering data, so they developed standard operating procedures that data entry clerks could use.

When volume of actions necessary to accomplish a task becomes too great, the complexity of the task must be reduced [3] through *division of labor*. What used to be the responsibility of the reader of the simple paper document was decomposed into design of forms processing tasks, and task execution. Forms and task design were centralized and performed by professionals; data entry and task execution, de-centralized and performed by clerks. Innovations (circa 1890-1930) [4] enabled further division of labor: Counting machines for punch cards and register machines sped processing, and one-write systems and carbon paper eliminated unnecessary task steps.

One way to guide division of labor to reduce complexity of coordinating different tasks is *near decomposability of a system* [5]: Construct units within which tasks are performed such that interactions between units are minimal. Strategies for reducing coordination complexity are predicated upon this principle [3]. One is *contracting*, wherein informational complexity is reduced to a price, and task complexity, to contractual terms with a near decomposable unit, the contractor. Many businesses outsourced forms design and production to specialized printing houses such as *Moore Business Forms*, because large-scale forms production was prohibitively expensive. Low-cost office typesetting (circa 1950) changed this. A near decomposable unit—the organizational systems department (often subsuming a forms department)—was created by many businesses, equipped with typesetters, and staffed by forms and task designers. Hence, a specialized functional division arose, whose birth can be explained by the following: Organizational sub-structuring towards *functional or product orientation*—depending on characteristics of problems faced by an organization—also reduces coordination complexity.

Last significant electro-mechanical innovations (circa 1960-70) were electrostatic and xerographic photocopying, which enabled inexpensive, high quality, large volume replication. As photocopiers became available outside the organizational systems department, forms users reduced their dependency on the department by photocopying extra legitimate, as well as customized, "bootleg" forms. The use of *slack resources* to de-couple dependent tasks is a third coordination complexity reduction strategy. However, these "bootleg" forms also introduced uncertainty:

> "The ease with which forms can be reproduced has resulted in the proliferation of "bootleg" forms—those forms which can be produced outside of the control of the forms department… I'm not suggesting that forms should never be photocopied or that "bootleg" forms should never exist: sometimes the cost of control just isn't worth the effort. However, the real cost lies in the clerical processing and from my experience in dealing with forms for almost 30 years, I have found few designers of "bootleg" forms who give processing efficiency much consideration" [4]

For example, a data processing clerk could not process a “bootleg” form that seemingly contained required information, though expressed ambiguously; or, a system tuned to process a certain volume of completed forms could not cope with additional volumes of user-photocopied forms. Uncertainty introduced by “bootleg” forms to an efficient forms processing system led to efficiency loss.

With the advent of widespread computerized data processing, systems based on paper-based business forms were transformed to those manipulating digitized data; organizational systems departments of forms and process designers gave way to MIS departments of database and programming analysts. One aim of process re-engineering (circa 1990’s) was to re-design computerized systems that had gradually evolved from forms-based systems, and hence still predicated upon some mechanistic and manual restrictions of forms use that no longer applied.

III. Implications for Evolution of Ontologies for the Semantic Web

If re-engineers understood how adoption of innovations led to changes in an organization’s forms-based systems, they would have been able to systematically identify components of the evolved system most amenable for re-design as ones developed to implement outdated innovations. Moreover, if they could explain changes to forms-based systems using a model such as Fox’s, they may have been able to make some predictions about how their re-designed system would evolve as vanguard innovations were eventually adopted. Taking this approach, in the early 1990’s, would some prescient BPR expert have designed a flexible, not necessarily optimally efficient, inventory management system that could be integrated with customers’ systems using the Internet?

In this section, such an approach is taken to predict how ontologies for the Semantic Web may evolve.

XML vs. Ontologies

XML and ontologies are two means of explicitly representing information applied so that a reader interprets shared data as intended by the data author. XML use for the WWW is analogous to business forms use, since informational structure represented in DTD's (terminology) is delineated from content represented as XML data (e.g. `<foo>7</foo>`).

The definition of 'ontology' used in this paper is that it "consists of a representational vocabulary with precise definitions of the meanings of the terms of this vocabulary plus a set of formal axioms that constrain interpretation and well-formed use of these terms" [6]. This is a more restrictive one than a "lowest common denominator" definition: "an ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning" [7]. For the Semantic Web, an ontology must be expressed in a formal language so that a given ontology expression can be interpreted and processed unambiguously by a machine. Models for communicating vocabulary and structure to humans such as *Yahoo!*'s taxonomy [8]—"light-weight ontologies"—and most conventional ER diagrams are too informally expressed for automatic machine processing of semantics. Ontology use for the Semantic Web then is analogous to use of business forms with standard operating procedures, since informational structure is represented as terminology; rules governing proper interpretation of the structure, as formal definitions and constraints (semantics or meanings); and content, as ontology ground terms (e.g. `foo(7)`).

Shared understanding about a community—information that its members possess—is always applied in solving problems in that community. The terminology used by community members can be codified as the community's DTD's. Ontologies, as "explicit representations of shared understanding" [9], can also be used to codify the terminology's semantics. For example, it must be assumed in using XML that the author and reader of `<foo>7</foo>` have the same understanding of what 'foo' means. This assumption need not be made in ontology use, since 'foo' can be explicitly defined. In comparing ways of codifying shared understanding using the Semantic Web, it must be acknowledged that XML is a much more mature technology than ontologies in terms of size of user community, availability of support tools, and viability of business models relying on the technology. *Therefore, ontologies can be adopted in situations where the capability to represent semantics is important enough to overcome XML's maturity advantages*. What are characteristic of these situations?

For form-based systems, innovations were adopted over existing technologies to reduce information, task, and coordination complexity, or uncertainty. Accepting that forms and XML/ontologies are analogous, and XML is the more mature, and ontologies, the more innovative, technologies for the Semantic Web, then ontology adoption will occur in situations where complexity or uncertainty is reduced more by ontology rather than XML use. Specifically, this occurs when semantics reduces complexity or uncertainty. So, the pros and cons of XML and ontology uses are first analyzed in terms of the three complexity reduction principles.

(1) *Bounded rationality*: XML use is less complex since semantics are not represented. Whereas many people can identify and classify terms, only some can systematically express meanings of these terms, never mind, represent them in a formal language. With XML use, however, there is increased uncertainty that crucial information for interpreting shared data is not represented. In situations where it is reasonable to assume that shared understanding can be implicitly applied (by assuming for example that everyone has been uniformly trained) or informally applied (by assuming for example that user manuals are referenced), the uncertainty of omission is mitigated.

(2) *Division of labor*: There is a clearer delineation of responsibilities in XML use. DTD and data sharing task designs are done by professionals, data entry and data sharing, by computers with some manual intervention. It may not be possible to automate, or even apply clerical skills, to data entry for ontology use because sometimes definitions and axioms are entered, and their formulations require skills beyond the merely clerical. Therefore, tasks for manipulating XML data are likely more efficient. However, for automated data sharing, an XML based system will be more susceptible to data that cannot be interpreted properly than an ontology based system, which is able to apply semantics for interpretation.

(3) *Near decomposability*: If interactions between near decomposable units are minimal, a corollary states that interactions within a unit are great. Such a unit can then organize to reduce complexity of interactions, guided by principles of bounded rationality and division of labor. As long as a unit can be considered nearly decomposable, (1) and (2) provide reasons for why XML use reduces complexity.

However, if near decomposability cannot be assumed, ontology use increases the likelihood that data can still be shared.

The following summarize the “XML vs. Ontologies” analysis: ***A unit is nearly decomposable for purposes of data sharing if it is reasonable to assume that shared understanding can be implicitly or informally applied to interpret data within that unit (a community). Within a near decomposable unit, it is important to reduce complexity in data sharing. If near decomposability cannot be assumed, reducing uncertainty of data sharing by explicitly and formally defining semantics in ontologies may be warranted. Unless reducing uncertainty is more important than reducing complexity for using the Semantic Web, XML will be a better or more proven data sharing platform than ontologies.***

This reflect Fox’s statement that as an organization structures to reduce complexity, it simultaneously faces increased uncertainty [3].

Using XML for Complexity Reduction

Figs. 1 presents models in which shared understanding is codified. They reflect structures borne to reduce coordination complexity. In the contracting model, the business network can be considered a near decomposable unit, since data is greatly shared between its companies and service, which are more strongly near decomposable. According to the “XML vs. Ontologies” analysis, XML use for data sharing within the network then is appropriate. An example of this model is *Covisint*, an on-line automotive industry exchange using *Commerce One*’s XML based xCBL™. In the functional orientation model, the enterprise is more near decomposable than its departments and functions, so

XML use within the enterprise is quite appropriate. For example, *WebMethods* provides XML based tools to enable companies to perform the data integration function.

Figure 1. Near Decomposable Units for Data Sharing: XML Appropriate

<Insert Picture>

Using Ontologies for Uncertainty Reduction

Photocopiers were used as slack resources that loosened the forms user's dependence on the designer, which led to users assuming some forms design responsibility. A parallel effect for data sharing is the assumption of some of the enterprise's data integration responsibility by departments or other entities within the enterprise. The following presents one such slack resources model.

Figure 2. Near Decomposable Units for Data Sharing – Ontology Appropriate

<Insert Picture>

In this model, the analog to the photocopier is the data modeling tool. Using the tool, knowledge workers—not specialized data modelers—who apply shared understanding for their jobs also codify it. Codified shared understanding is then used to translate data and prepare it for use by an external entity. "Bootleg" forms produced with photocopiers introduced uncertainty because tasks had not been designed to handle them. Similarly, the data modeling tool gives knowledge workers the ability to codify idiosyncratic shared understanding that will result in data requiring unforeseen or unexpected idiosyncratic interpretation by another entity. One way to acknowledge that uncertainty is inevitable is to not commit to how data from an entity will be interpreted, hence the '?' shown in the model.

In this model, it cannot be known a priori whether an entity and another with which its data needs to be shared are enclosed within a near decomposable unit. Complexity reduction afforded by the data modeling tool's use is offset by the uncertainty introduced that un-interpretable data is produced, if XML is used. In contrast, knowledge workers can explicitly represent semantics for interpretation and introduce less uncertainty if they use ontologies. Therefore, it is predicted that: ***Ontologies for the Semantic Web may be widely adopted, if there are ontology development tools that can be practically used by knowledge workers, not necessarily by ontologists*** *(specialized ontology modelers)***.**

The tool will be evaluated on factors such as ease of use and capability to express rich concepts without complex knowledge representation expertise. However, ontology adoption will not depend primarily on these factors. In Fig. 2, the rationale for considering an entity as a near decomposable unit is not to codify shared understanding; if it were, ontologists would codify. The rationale is a business need that can be satisfied by knowledge workers with useful skills. A popular knowledge management (KM) principle is that people will not contribute to a knowledge base if doing so takes too much time and effort away from their own jobs [10]. Many KM tools (e.g. *Intraspect*'s) are designed using this principle. Information to be shared is codified as a by-product of workers using the tool for tasks like e-mail processing important to their jobs.

Jasper and Uschold [7] categorize ontology applications as: neutral authoring, ontology as specification, common access to information, and ontology-based search. Only in ontology as specification—domain ontologies are created and used as a basis for specifying and developing software—is the ontology developed in the course of doing some other work, namely software development, and produced as a by-product. Therefore, it is predicted that: ***Ontologies are likely to be widely adopted, if an ontology developed by the knowledge worker is of use to the worker irrespective of whether it is used for data sharing. Therefore, ontologies may be widely adopted first for software specification.*** It can be argued that "light weight" ontologies for ontology-based search are already commonly used. However, these ontologies do not conform to the definition of ontologies used in this paper, since it is not likely that machines can interpret representations in such ontologies automatically.

An ontology for software specification is useful even if applied only once, say for a large software project [7]. For early authors to the WWW, intellectual curiosity was compelling enough reason to develop web sites about which most people would not know. Isolated ontology development for software specification, uncoordinated with other ontology-like efforts, i.e. a de-centralized approach, is a way of getting practical ontologies onto the Semantic Web. Few assumptions can be made about how such ontologies will be used by others, so they should be designed for flexibility and adaptability, and commit little to how they would be used. Therefore, it is predicted that: ***The first phase in the evolution of the Semantic Web may be the development of de-centralized, adaptive ontologies for software specification***

IV. Concluding Remarks

This paper attempts to predict the future of Semantic Web ontologies (web based analog to business forms *cum* standard operating procedures) by analyzing the history of paper based business forms. Forms innovations were adopted to reduce information, task, and coordination complexity. However, one such innovation, the photocopier, had a by-product effect of increasing uncertainty in forms processing. In evaluating possible adoption of competing analogs to forms for the Semantic Web—XML and ontologies—it has been argued that as long as the pressing need is to reduce complexity, XML use is preferable to ontology use. It has also been posited that the innovation of modeling tools allowing knowledge workers to codify idiosyncratic information and expect that information to be shared will increase uncertainty in data sharing. When this happens, the use of ontologies over XML to codify information will likely be desirable. These predictions confirm what some in the ontology community suspect may happen[1], and place emphasis on:

- Designing an ontology development tool demonstrated to be useful and useable to a knowledge worker, who is not a knowledge representation expert.
- Development of de-centralized, and adaptive ontologies, which have value in of themselves, but whose full potential will only be realized if they are used in combination with other ontologies in the future to enable data sharing. The immediate value may be use of ontologies for software specification.

It must be noted as a caveat that these predictions are not founded on a rigorous analytical or empirical model. Rather, they are argued using analogies and a conceptual model, and hence much further research is required to strengthen their validity. Nevertheless, they

[1] once the infrastructure technologies for representing ontologies in the Semantic Web are put into place, i.e. after languages like RDF, DAML, and OIL are further developed and standardized

are the reasonable product of a systematic analysis, and as such hopefully will provoke thought and motivate concrete research questions about the nascent Semantic Web. How does the ontology development tool work? How are de-centralized, adaptive ontologies constructed? How are such ontologies organized for data sharing in the future? The main contribution of this paper is that it provides a rationale as to why these may be the pressing questions to ask to understand how ontologies and the Semantic Web will unfold.

V. References

[1] Berners-Lee, Tim, Hendler, James, and Lassila, Ora (2001). "The Semantic Web", *Scientific American*, May.

[2] Simon, H.A. (1957). *Models of Man*. New York: Wiley.

[3] Fox, Mark S. (1981). “An Organizational View of Distributed Systems”, *IEEE Transactions on Systems, Man & Cybernetics*, Vol. 11, No. 1, pp. 70-80.

[4] Barnett, Robert. *Managing Business Forms*. Robert Barnett and Associates Pty Ltd.

[5] Simon, H. A. (1962). “The Architecture of Complexity”, *Proceedings of the American Philosophical Society*, Vol. 106, pp. 467-87.

[6] Campbell, A. E. and Shapiro, S. C. (1995). “Ontological Mediation: An Overview”, *Proceedings of the IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*, Menlo Park CA: AAAI Press.

[7] Jasper, Robert & Uschold, Mike (1999). “A Framework for Understanding and Classifying Ontology Applications”, in *IJCAI-99 Ontology Workshop*, Stockholm, Sweden, July.

[8] Labrou, Y., Finin, T. (1999). "Yahoo! as an Ontology - Using Yahoo! Categories to Describe Documents", In: *Proceedings of the 8th International Conference on Information and Knowledge Management*, November, Kansas City, MO, pp. 180-7.

[9] Gruber, Thomas R. (1993). "Towards Principles for the Design of Ontologies Used for Knowledge Sharing", In *International Workshop on Formal Ontology*, N. Guarino & R. Poli, (Eds.), Padova, Italy.

[10] Davenport, Thomas H. & Prusak, Laurence (1997). *Working Knowledge: How Organizations Manage What They Know*.