Web Metadata Standards: Observations and Prescriptions

David Bodoff and Patrick C.K. Hung, *Hong Kong University of Science and Technology*
Mordechai Ben-Menachem, *Ben-Gurion University of the Negev*

The World Wide Web has spawned numerous standards initiatives that aim to facilitate more powerful and interoperable functionality based on text exchange, but beyond mere Web page transfers. Software can take a Web page's data as input to further value-added processing, such as filtering items of interest, comparison shopping, finding potential business partners, and executing transactions. But software can do this only if the page's meaning is formalized and explicit. This is content metadata's role: to make the meaning of the Web page's data[1] formal and explicit to facilitate further processing by useful software tools.

We divide these content metadata standards into two categories:

- *Process standards* describe executable processes' behavior and facilitate application-to-application interaction.
- *Product standards* describe a physical or information product and facilitate the description's exchange.

Here, we review various Web content metadata standards[1] and offer observations on their development efforts. We're motivated both by the sheer number of standards and a concern that in our haste to advance these standards and their promised functionality, we might overlook key lessons learned in various disciplines, including software engineering, software reuse, and library science. We call particular attention to the apparent confluence of standards development and artificial intelligence, which raises additional possibilities and concerns.

Lessons learned in fields such as software engineering, library science, knowledge representation, and especially artificial intelligence offer new possibilities and questions for Web metadata standards development.

Metadata standards

Content metadata standards are built on top of infrastructure standards that standardize metadata representation and exchange. Here are two examples:

AI and Web Standards Converge

Many Web metadata standards are based on frame-like data structures. Artificial intelligence research in description logics (DL) has used frame-based metadata as the basis for knowledge representation and the classification/subsumption inferencing method, which is used to find an object's position in a hierarchy. Ontologies are based on such technology, and some standards efforts reflect AI's orientation toward inferencing and automatic content understanding.

As Deborah McGuinness, a leading DL systems developer put it, "Web usage … has drawn description logics out of ivory towers."[1] Web standards and AI have come together, but the convergence is not without tension. A standard is fixed by convention. Some standards don't fix every detail, but instead provide facilities—rudimentary languages—for their dynamic negotiation; examples include baud rate, transport protocol, and so on. But many Web standards such as ebXML (for process) and DAML (for products and knowledge) are powerful declarative languages (at the M2 layer), not models (at the M1 layer). In some quarters, these languages are considered a sufficient standard, while some Web standards efforts go still further by incorporating AI goals.

Our concern is that intermingling AI with standards efforts could undermine the more modest achievement of interoperability. Writing in 1992 about the Information Resource Dictionary System, an early metadata effort, John Sowa observed that because some IRDS committee members "take the term 'metainformation' seriously. … Practically every aspect of knowledge representation in AI becomes relevant. Most people on the committee recognize that point, but there are two reactions to it:

- 'Great! That gives us an opportunity to do all of AI within the IRDS committee.'
- 'Get serious! We've got products to deliver and problems to be solved this year.' "

(see www-ksl.stanford.edu/email-archives/srkb.messages/117.html).

Hopefully, the convergence of AI and standards will yield mutual benefits. Our concern is that attention to AI-like goals might divert energies from interoperability and other issues we've raised here, such as software testing and quality control, user search tools, proper access points, and user-friendly tools.

Reference

1. D. McGuinness, "Description Logics Emerge from Ivory Tower," *Proc. 9th Int'l Conf. Conceptual Structures* (ICCS 2001), Stanford Univ., 2001; www.ksl.stanford.edu/iccs2001/program.pdf.

- An *XML Document Type Definition* (XML DTD) lets authors specify a document's structure. A reader (human or software) can then refer to that DTD to parse the document and extract needed sections.
- The World Wide Web Consortium's *Resource Description Framework* (RDF) describes resources using a graph model in which links denote a property and the destination node represents that property's value.

Product and process content standards, which are our main focus, standardize the actual metadata descriptions of Web products and services. This content metadata is then represented and exchanged using the XML and RDF infrastructures. Many Web content standards adopt an object- or frame-based hierarchical orientation that fits well with RDF (which describes resources as graphs) and XML (which represents tree data structures as linear text). The result is a relationship between these content metadata standards and frame-based knowledge representation (see the "AI and Web Standards Converge" sidebar).

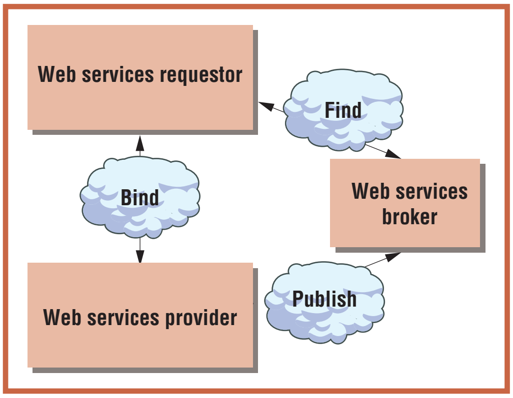


Figure 1. The publish-find-bind model. This model describes the relationship among the three primary participants in a Web service: the requestor, provider, and broker.

Process metadata standards

A *Web service* is a software process that supports application-to-application interaction over the Internet. Process metadata describes the interaction behavior, as well as how applications execute it, where it's located, the nature of its security provisions, how to send it data, and so on. Figure 1 shows a model of the relationship between a Web services requestor, provider, and broker.

In the publish-find-bind model, partici-

Table 1

Web metadata standards

Standard	Standards body	Year adopted	Metadata function	Supported tasks
Electronic Business Using Extensible Markup Language (ebXML); Business Process Specification Schema (BPSS)	Organization for the Advancement of Structured Standards (OASIS)	1999	Describes business collaboration	Publish and bind
ebXML Collaboration Protocol Profile (CPP)	OASIS	1999	Describes a Web service	Publish and find
Web Services Description Language (WSDL)	World Wide Web Consortium (W3C)	2002	Describes a Web service's interface	Publish
Universal Description, Discovery, and Integration (UDDI)	OASIS	2002	Describes and discovers businesses and their services	Find
Simple Object Access Protocol (SOAP)	OASIS	2000	Use for XML-based messaging protocol	Bind
Web Services Security (WS-Security)	OASIS	2004	Describes security enhancements to SOAP messaging	Bind
Platform for Privacy Preferences (P3P)	W3C	2002	Describes privacy preferences	Bind
Dublin Core (DC)	DC Metadata Initiative (DCMI); National Information Standards Organization (NISO)	1998 (DCMI) 2001 (NISO)	Describes resources	Find
Computer Interchange of Museum Information (CIMI)	CIMI Consortium	1998	Describes museum-related information	Find
Web Ontology Language (OWL)	W3C	2004	Describes knowledge	Understanding and inference

pants communicate using metadata standards. The provider uses metadata to describe its services. For example, the *Web Services Description Language* is an XML-based language that describes a Web service's interface (which operations it supports, which protocols it uses, and how requesting applications should pack the data). The Web services provider can then *publish* the WSDL document to the Web services broker via Universal Description, Discovery, and Integration (UDDI) registries, which serve as a kind of yellow pages for Web services. A requestor then queries the registry to *find* an appropriate Web service. In response, the requestor gets a WSDL document and tries to *bind* with that Web service using a messaging protocol—such as SOAP—specified by the WSDL service description.

To fully accomplish even one of the publish-find-bind operations, you might actually need many protocols. To fully describe (publish) an available Web service, for example, you might have to describe its function within business collaborations. Metadata standards such as the ebXML Business Process Specification Schema (BPSS), or the Business Process Execution Language for Web Services (BPEL4WS) let you choreograph a sequence of individual services into business collaboration. Using ebXML's Collaboration Protocol Profile (CPP), a Web service provider can then publish its process as one that fills a particular role within a business collaboration.[2] Table 1 presents a sample of Web metadata standards that include publish, find, and bind steps.

There are also XML-based security standards, such as the Security Assertions Markup Language (SAML) and WS-Security, and XML-based privacy standards, such as the W3C's Platform for Privacy Preferences (P3P).

This review is merely a starting point; there are other security standards, such as XML Signature (XML-DSig), XML Encryption (XEnc), the XML Key Management Specification (XKMS), and entire groups of process standards that aren't strictly Web related. One example of the latter is the US National Institute for Science and Technology's Process Specification Language.[3]

Product metadata standards

Initiatives in this category standardize descriptions of physical products, data, information resources, and documents. Dublin Core is an early Web metadata initiative. Its version 1.1 consists of 15 properties for describing an

Internet resource, including such basics as title, creator, and subject.

Many industries have developed metadata standards that are essentially controlled vocabularies—an agreed properties list that people can use to describe industry items. The content Standard for Digital Geospatial Metadata, for example, defines a common set of geospatial-data terms. Businesses that want to describe products and services can use RosettaNet's dictionaries to define the legal attributes. Museums have standards such as Spectrum for managing their holdings' life cycle. Some product standards let users define taxonomies, rather than just flat vocabularies.

Another product metadata category targets knowledge description. These standards, such as DARPA Agent Markup Language + ontology inference layer (DAML+OIL) and Web Ontology Language (OWL), are conceptual modeling tools that support inferences about taxonomies and reflect an AI orientation.

Prescriptions

We organize our prescriptive comments into three sections: software engineering, software reuse and library science, and AI. Although we can't lay claim to having all the answers, our hope is that these prescriptions will focus attention on key issues.

Lessons from software engineering

Relevant software engineering lessons include the need to consider testing costs and to investigate the trade-offs involved when a single standard aims to serve multiple purposes.

Issue #1: Ignoring the main action? As Table 1 shows, testing is apparently not among metadata's purposes. Software testing generally represents 30 to 40 percent of software project costs, yet we're unaware of any metadata standard that contains a single element aimed at facilitating testing. This is a major oversight, given not only the project costs statistic but also the fact that distributed, interorganizational systems pose unique testing obstacles. This concern extends more broadly to software quality assurance in general, including configuration management, defect tracking, traceability, and other quality issues. It's a natural tendency to develop standards with an eye toward new and exciting functionality, but it would be a mistake to ignore boring old problems such as testing costs.

Prescription #1: Metadata standards efforts must consider their impact on long-standing problems such as quality assurance and testing costs, rather than focus solely on new functionality.

Issue #2: Does one size fit all? Metadata can serve many purposes, as Table 1 indicates. In some cases, however, a single metadata standard proposes to fulfill many purposes simultaneously. ebXML's CPP, for example, announces the role a company wants to play in a business process, and the company's technical capabilities (such as digital signatures and so on). The CPP serves at least three different purposes:

- *Documentation*. The CPP acts as a requirements document to guide services implementation in custom software.
- *Configuration*. The CPP can guide software that negotiates an agreement with another company.
- *Access point*. The CPP is registered in a directory to announce a company's interest in doing business.

The CPP example raises questions: Can an executable declaration double as a requirements document and triple as its own access point? When can a single metadata standard effectively serve multiple purposes?

Some Web metadata standards imply that they can serve purposes beyond their original designation at no extra cost. In the CPP example we just gave, however, this free lunch isn't wholly convincing. Entire literatures exist on requirements analysis, declarative software, and indexing and access; it seems unlikely that a single metadata standard could effectively fill all these roles.

Prescription #2: Metadata can simultaneously serve different purposes, but its effectiveness at doing so shouldn't be assumed. When piggybacking multiple uses onto a single standard, the trade-offs require explicit investigation.

Lessons from software reuse and library science

Relevant lessons in these areas include the importance of access points and the question of how best to provide them, the need for practical search and navigation tools, and the drawbacks of offering multiple indexes.

It's a natural tendency to develop standards with an eye toward new and exciting functionality, but it would be a mistake to ignore boring old problems such as testing costs.

Software reuse and library science literature show that classification schemes and thesauri present real challenges to both developers and users.

Issue #3: Should metadata serve as its own access point? In library science, an access point is a searchable property through which users can locate (access) items. Our CPP example raises the question of whether metadata can serve well as its own access point or whether meta-metadata is required for that purpose. If the CPP serves as its own access point, searchers would submit queries such as, "Please show me CPPs that offer the role of insurer in a shipping process and that use digital signatures." On the other hand, meta-metadata—such as title or author—could be added. Such additional elements wouldn't be part of the *executable* CPP, but would describe the CPP in a way that might help searchers find it.

Is this meta-metadata necessary? Evidence from library science and information retrieval indicates that while content—data or metadata—can double as its own access point, meta-metadata that's specifically designed to facilitate access can play a role. Users don't like to search through formal structures. For example, many library users search by title rather than subject to avoid the subject catalog's controlled vocabularies and hierarchical structure. Similarly, Web users might be reluctant to search the standardized metadata looking for "an insurer's" (a controlled term) role in a "shipping process" (a standardized structure). Instead, they might prefer to search for a company name or a software vendor or a title of some kind. Although such fields aren't required to execute the CPP, users might prefer them for searching. For related reasons, IEEE Standard 1420.1 Basic Interoperability Data Model specifies that software reuse libraries include metadata. Such metadata is thoughtfully designed to serve as an access point, while ordinary metadata—such as that in a CPP—is not.

Web metadata standards developers might be tempted to let any metadata serve as its own access point simply because it can. However, experience in library science and software reuse suggests that providing proper access is a task in its own right.

> ***Prescription #3: Experience with library searches indicates that we might have to provide meta-metadata to make searching metadata elements easier for users.***

Issue #4: Cataloging for retrieval. Metadata for cataloging Web processes is related to metadata for cataloging software in reuse libraries. What does the software reuse literature say about successful classification methods? Many texts recommend faceted classification and thesauri. Yet, almost all texts also describe the disappointing state of reuse and often cite the difficulty of constructing and maintaining faceted classifications and thesauri. ebXML's registry information model has a placeholder for thesauri and provides a domain-specific classifications framework, but it supports hierarchical rather than faceted classification. When designing process metadata, we're apparently not utilizing the available technical lessons from software reuse literature.

Library science and software reuse literature also identify the lack of good search tools as a perennial problem. Library catalogs use subject classification and controlled vocabularies, but few tools are available to navigate the classification and thesaurus hierarchies. The lesson? No matter how ingenuous the classification, users need practical and powerful tool support, and such tools must be developed before the metadata will be widely used.

A final lesson indicates the importance of user education and training. One study of library patrons found that only 28 percent even knew that the library subject catalog uses a controlled vocabulary,[4] never mind their own proficiency with it. Other studies report difficulties with Boolean searches, which is particularly sobering when we compare its simplicity with the elaborate Web metadata registries that researchers currently propose. In short, software reuse and library science literature show that classification schemes and thesauri present real challenges to both developers and users, and have met with limited practical success.

> ***Prescription #4: Although recommended, cataloging methods such as classification schemes and thesauri don't guarantee metadata success. To help ensure that success, we must educate users and develop practical tools to support their search and navigation through classification and thesaurus hierarchies.***

Issue #5: Indexes and registries. Issue #3—whether metadata or meta-metadata is the better access point—entails a separate question about indexing. Should we index the access points? Should we index that index? What about the index to that index? In ebXML, UDDI is a standard method for registering an index to an index to an index to an index. Specifically, it registers documents containing pointers to a document registry containing pointers to descriptions of (and pointers to) Unified Modeling Language artifacts that describe a software component. It's difficult to imagine that users will easily traverse this many indexes.

Libraries provide an analogy. Libraries own numerous disjoint databases. First, suppose a given library's databases contain pointers to abstracts, rather than to actual documents. The library's databases index would thus correspond roughly to an ebXML registry. Now suppose this databases index isn't visible at the reference desk, and patrons must locate a record that tells them where to find the index. Further, suppose that the library expends resources standardizing the process by which librarians introduce such a record. That's UDDI in the ebXML context. It's difficult to imagine the return on investment of that effort, which standardizes the process of adding a record of an index to an index to an index.

In fact, UDDI remains a largely experimental standard that hasn't gained much traction. Part of the reason might be that the UDDI model of finding Web services introduces significant overhead: users must learn how to use a nontrivial UDDI querying language—and perhaps an additional query language at the target repository—before they ever get to a Web service, which requires them to learn additional technical details. The yellow-pages metaphor is appealing, but real yellow pages are easy to use, and the overhead is more than offset by users' enhanced ability to locate the items they need. This might not be the case with UDDI.

> ***Prescription #5: Standards should introduce additional indexes only if they offer users marginal search benefits.***

Lessons from AI and knowledge representation

Lessons from AI and knowledge representation include the need for practical abstractions and narrow domains that limit their use of inferences.

Issue #6: Impractical abstractions. The Object Management Group's Meta Object Facility (MOF) divides standards into layers: M0 is for metadata instances, M1 is a metadata model (such as class definitions), M2 is a metamodel (a language for defining metadata models), M3 is a meta-metamodel (a language for defining an M2 language), and so on.

To facilitate mere interoperability, it's effective to adopt an M1-layer standard. For example, this would mean adopting a particular relational schema (M1 layer) as a standard, rather than the relational language But, as ideas from knowledge representation and AI exert their influence, some metadata efforts are increasingly viewing language adoption as a sufficient standard in itself, with the hope that anyone speaking the same language—an ontology language, for example—will be "understood." This goal makes sense from an AI perspective, where the goal is to understand novel expressions.

Adopting a language as a standard in itself conflicts with the more modest goal of interoperability.

However, adopting a language as a standard in itself conflicts with the more modest goal of interoperability. Ronald Brachman classifies representation languages on the basis of the modeling primitives they use.[5] He distinguishes four levels—logical, epistemological, conceptual, and linguistic—that we refer to as *representation levels*. Nicola Guarino has further inserted an *ontological* level,[6] which is currently receiving much attention. Brachman writes that a language with a clear semantics uses lower-level primitives to define new, higher levels. For example, a language might use logical primitives to construct a shorthand epistemological tool that lets users create object classes or relations. The logical primitives in that example are neutral with respect to the epistemological level in that a language can use them to construct various epistemological tools, such as object classes or relations. Those new epistemological tools might then be adopted as primitives for a language at the next higher level to use in defining still higher-level authoring tools, and so on.

This analysis shows that we can expect interoperability problems if we adopt a language as a standard without further adopting a particular M1 model. The whole point of providing neutral language primitives at one level is to facilitate variety at the next higher level. Thus, adopting an M2 language as a standard not only allows conflict at higher representational levels, it actually facilitates it. This is true even of ontological languages that provide ontological primitives, because there are representational levels above the ontological. Figure 2 shows Michael Genesereth's example of two conceptually different models that both use the relational language.[7]

Figure 3 offers a helpful tool here, showing the four representational levels applied to all

Figure 2. Two relational language models. Although they use the same language, conceptual incompatibilities create interoperability problems between the (a) FatherMother table and the conceptually different combination of (b) parent and (c) gender tables.

Person	Father	Mother
John Jr.	John	Sally
Connie	Michael	Lorraine

(a)

Child	Parent
John Jr.	Sally
John Jr.	John
Connie	Michael
Connie	Lorraine

(b)

Name	Gender
Sally	F
John	M
Michael	M
Lorraine	F

(c)

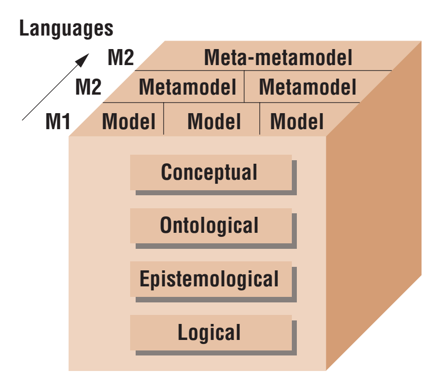


Figure 3. A model of the metadata standards effort. This model shows the four representational levels applied to all Meta Object Facility language layers.

the MOF language layers. Brachman's original analysis focused on the M2 language "slice," where representational levels correspond to language primitives. Applied to the M1 slice, the levels represent not different kinds of language primitives but different choices concerning their use. That is, the levels represent which logical statements (logical) and representations (epistemological) the model uses, what kinds of things (ontological) it represents, and so on.

We draw several related conclusions from this analysis. First, an M1-level metadata standard is ideal for promoting product and process interoperability. Second, we should view a standard M2-level language as a first step toward an M1 metadata standard. Finally, even at the M1 level, the higher the representational level, the more powerful the standard is at preventing conflicts. RosettaNet (www.rosettanet.org) demonstrates an M1 metadata standard's usefulness. If the M2 language provides for logical propositions through attribute-value pairs, then the M1 standard should specify the attribute names (as in RosettaNet's business properties) and the possible values (as in RosettaNet's global product identifiers). If the M2 language provides epistemological primitives for structuring attributes into frames and documents, the M1 standard should specify a particular frame structure choice (as in RosettaNet's business data entities). If the M2 language provides ontological primitives, the M1 metadata standard should use them to make ontological choices (this is often implicit).

Prescription #6: Ideally, a metadata standard should define concepts (a relatively high representational level) at the M1 layer, which offers relatively low language abstraction. While obviously possible, adopting an M2 language as a standard actually facilitates variation and conflict at representational levels above that language's primitives.

Issue #7: Ontologies. Any standardized taxonomy can help interoperability. For example, an industry group might propose a standard object taxonomy so that different authors use the same terminology. We call this a "convention ontology" because its effectiveness depends only on conventional acceptance. In contrast to this relatively simple purpose are the more ambitious purposes of using ontologies for inference and artificial sense making. For example, a potential business partner's system might see a product description on the Web, and infer from its detailed description that it's a kind of consumer electronics, even though it wasn't explicitly described in that way. Other potential partners, consumers, or government agencies (such as customs) might make different inferences about the product's categorization. This sort of inferencing capability requires that products be described in ways that conform to more demanding ontological rules that are the subject of ongoing research.

Christopher Welty and Nicola Guarino provide guidelines for constructing well-founded ontologies on the basis of philosophical principles rather than arbitrary convention.[8] When applied to such an ontology, inferences such as inheritance and classification/subsumption would yield correct results. In addition, independent individuals following those guidelines might contribute pieces of a wider ontology, with internally consistent results. Until such well-founded principles are finalized, however, the ontologies we create are based on mere convention. In such a case, we must

- Limit the convention ontology to a single narrow domain
- Use inheritance and classification/subsumption inferences cautiously

When a convention ontology's domain is too wide, definitional deficiencies emerge when developers attempt to use the same arbitrary definition in different contexts. Raphael Malyankar[9] built a maritime ontology using various sources, including standards, database schemas, symbology legends, and US Coast Pilot documents. He then considered reusing parts of previously built ontologies, including one that defined a bridge as a "trafficable passageway." In the maritime context, however, a bridge is a hazard, not something trafficable! Clearly, conflicts can emerge when developers apply an arbitrary definition across contexts.

In the case of inheritance and classification, the literature is replete with examples of mistaken inferences resulting from convention ontologies. One common problem is in the difference between constitution (being made of some substance) and subsumption (being a subcategory of something).[8] Composition works differently from other properties. A system should infer that if a table is composed of wood, then so is its leg; it shouldn't infer that if table is a good place to set dinner plates, then so is its leg.

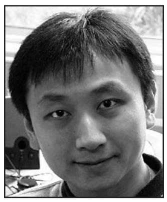
Prescription #7: Ontologies-from-first-principles are a knowledge representation research area, while the ontologies in everyday use as metadata standards are (merely) convention ontologies; as such, they should cover narrow domains with limited use of subsumption and other inferences.

Standards bodies and working groups such as those in Table 1 include serious and dedicated professionals who have likely considered the issues we've raised here. Perhaps in internal discussions, they made conscious choices to avoid testing metadata or to endorse using metadata as its own access point, and so on. The published recommendations, however, don't always divulge the basis for such choices. We've thus called attention to them as issues that deserve first-class public attention. Were standards developers to make explicit their considerations and final disposition on these choices, they might better benefit from one another's insights and experience, as well as from historical lessons.

About the Authors

David Bodoff is an assistant professor at Hong Kong University of Science and Technology, where he teaches courses on data communications and metadata. His research areas are information science, information retrieval, and the value of information in financial markets. He previously worked in industry developing expert systems. Bodoff received his PhD in information systems from New York University's Stern School of Business. Contact him at the ISMT Dept., Hong Kong Univ. of Science and Technology, Clear Water Bay, Hong Kong; dbodoff@ust.hk.

Mordechai Ben-Menachem is a researcher and adjunct lecturer at Ben-Gurion University in the areas of software engineering and project management. His primary research interest is management of very large software systems or installations. His consulting activities include fixing high-tech companies with technical or management difficulties. Contact him at Ben-Gurion Univ., P.O. Box 5613, Beer-Sheva 84156, Israel; quality@acm.org.



Patrick C.K. Hung is an assistant professor in the Faculty of Business and Information Technology at the University of Ontario Institute of Technology. His research interests include Web services security and privacy, business process integration, and electronic negotiation and contracting. He previously worked as a visiting assistant professor in Hong Kong University of Science and Technology's Department of Computer Science. He received his PhD in computer science from Hong Kong University of Science and Technology. Contact him at the Faculty of Business and Information Technology, Univ. of Ontario Inst. of Technology, 2000 Simcoe St. North, Oshawa, ON, L1H 7L7, Canada; patrick.hung@uoit.ca.

References

1. M. Dorr et al., *State of the Art in Content Standards*, OntoWeb Working Group on Process Standards, Nov. 2001; www.ontoweb.org/download/deliverables/D3.1.pdf.
2. S. Aissi et al., "E-Business Process Modeling: The Next Big Step," *Computer*, vol. 35, no. 5, 2002, pp. 55–62.
3. A. Knutilla et al., *Process Specification Language: An Analysis of Existing Representations*, US National Institute for Standards and Technology, 1998.
4. D. Steinberg and P. Metz, "User Response to and Knowledge about an Online Catalog," *College & Research Libraries*, Jan. 1984, pp. 66–70.
5. R.J. Brachman, "On the Epistemological Status of Semantic Networks," *Readings in Knowledge Representation*, R.J. Brachman and H. J. Levesque, eds., Morgan Kaufman, 1985, pp. 191–215.
6. N. Guarino, "Formal Ontology, Conceptual Analysis, and Knowledge Representation," *Int'l J. Human and Computer Studies*, vol. 43, nos. 5–6, 1995, pp. 625–640.
7. M. Genesereth, "Semantic Integration of Heterogeneous Databases," *Proc. Workshop Applications Logic Programming to Knowledge Management*, 2001; www.doc.ic.ac.uk/old-doc/semantics.pdf.
8. C. Welty and N. Guarino, "Supporting Ontological Analysis of Taxonomic Relationships," *Data and Knowledge Eng.*, vol. 39, no. 1, 2001, pp. 51–74.
9. R.M. Malyankar, "Maritime Information Markup and Use in Passage Planning," *Nat'l Conf. Digital Government Research*, 2001; www.digitalgovernment.org/library/library/pdf/asut_maritimeinfo.pdf.