

Using Knowledge Nodes for Knowledge Discovery and Collaboration. ¹

Per Christiansson

Aalborg University, Prof. IT in Civil Engineering.
Sohngaardsholmsvej 57, 9000 Aalborg
pc@civil.auc.dk, <http://www.civil.auc.dk/i6>

Abstract. Today most of the information we produce is stored digitally. We are slowly forced to leave behind us thinking about information as something stored in physical containers as books, drawings etc. We make it possible to dynamically create logical containers of information on the fly. The paper focuses on how we in the future can aggregate, classify and generalize digitally stored information in order to make it more accessible and how we can define underlying knowledge container models to support knowledge discovery and collaboration. Examples are picked from ongoing research and the outcomes are generally valid and in particular for the structural engineering field.

1 Introduction

Today most of the information we produce is stored digitally. We are slowly forced to leave behind us thinking about information as something stored in physical containers as books, drawings etc. We make it possible to create logical containers of information on the fly. This requires high level integration of those intranets, extranets, and Internet to which the physical containers (hard discs etc.) are connected. We know that the information is there somewhere in the cyberspace but how can we reach it and assess what we get back in terms of completeness and other quality parameters?

At the same time huge steps are taken on the building up of a global 'operating system' where agents and objects thrive - RDF (Resource Description Framework) to describe and exchange metadata over the networks, XML (Extensible Markup Language) to create application specific metadata formats, CORBA (Common Object Request Broker Architecture) for handling distributed objects and intelligent agents communication in client/server environments, and multicast protocols for optimal flow of information from one source to many receivers.

The paper focuses on how we in the future can aggregate, classify and generalize digitally stored information in order to make it more accessible and how we can define supportive underlying meta level knowledge container. Examples are picked from ongoing research and the outcomes are generally valid and in particular for the structural engineering field.

¹ From "Artificial Intelligence in Structural Engineering. Information Technology for Design, Collaboration, Maintenance, and Monitoring". Lecture Notes in Artificial Intelligence 1454. (Ed. Ian Smith). Springer-Verlag, Berlin. 1998. (pp. 48-59).

2. Areas of interest

As digital information will be easy accessible and flexibly packaged more focus will be on new tools for knowledge communication and competence collaboration as well as tools for knowledge experience capturing and storage for later use in projects and re-use in other projects. In parallel the knowledge discovery and data mining, KDD, tools will evolve.

A more or less conscious knowledge discovery process will take place in the project, global and even user digital domains. The increasing interest in the area is confirmed as you traverse the web; 'URL's for Data Mining' at <http://www.galaxy.gmu.edu/stats/syllabi/DMLIST.html>, 'Knowledge Discovery and Data Mining Web References' at <http://www.cs.uah.edu/~infotech/mineproj.html>, and 'Knowledge Discovery & Data Mining Web References' at <http://www.kdd.org/>.

We can thus distinguish some areas of particular interest for future research;

- to what detail will we classify digital information containers?
- what should the information container/wrapping granularity be to optimally support creation of digital knowledge containers?
- what knowledge representations will information containers support?
- who will mark information with subjective opinions (except from the authors)?
- how can information containers be associated with each other on different abstraction levels?
- how is bottom-up (meaning derived from content) and top-down (through classifiers and formalized structures) information search supported?
- how do intelligent agents navigate and find certain information patterns?
- how do we handle revised information?

The remainder of the paper will contribute to provide answers and general models to the above questions.

3. The Serfin and Mercurius Knowledge Nodes

A structural engineer is searching for information and possibly knowledgeable persons in the area of structural loadbearing capacity. He especially looks for high temperature steel properties in connection with repair of fire loaded paint protected beams. He contacts the Mercurius URL (Uniform Resource Locator) on the Internet. Mercurius, see figure 1, is a communication and information resource (demonstrator under development) through which knowledge produced at the Lund University is accessible. Information can be reached in three modes (a) through indexed free text search combined with search on documents similar to a found document, (b) by use of the public project and idea capture area where he can pose questions and look for potential project participants or (c) through establishment of a personal contact with a knowledgeable person at the university. In figure 1 it can be seen how the search domain may be restricted ('ange sökområde') to the local Mercurius knowledge container (concerning the knowledge communication process and information search itself), Lund University or the world.



Fig. 1. The Lund University Industry Knowledge Node for access of knowledge produced at the university and for capturing and display of ideas for future projects.

The engineer finds a reference to another URL, Serfin, via a set of keywords already used. The Serfin knowledge node, [2], is a communication and information resource for handling technical building maintenance knowledge. Figure 2 shows how he can choose between a coarse top-down search using controlled vocabularies (with optional graphic support) for five knowledge domains or plain free text search.

Both systems embody mechanisms for capturing and quality marking of stored knowledge. In the Mercurius system this process already exist in the university research and teaching procedures.

The Mercurius and Serfin systems contains digital information packaged as documents. These documents may in its turn contain text, images, graphics, video, sound, encapsulated calculation routines (in objects), etc. Documents are to some extent 'classified' with regard to covered knowledge domain and detailing level. Below we will further discuss how structures, content and functionality can be improved through high level modeling.

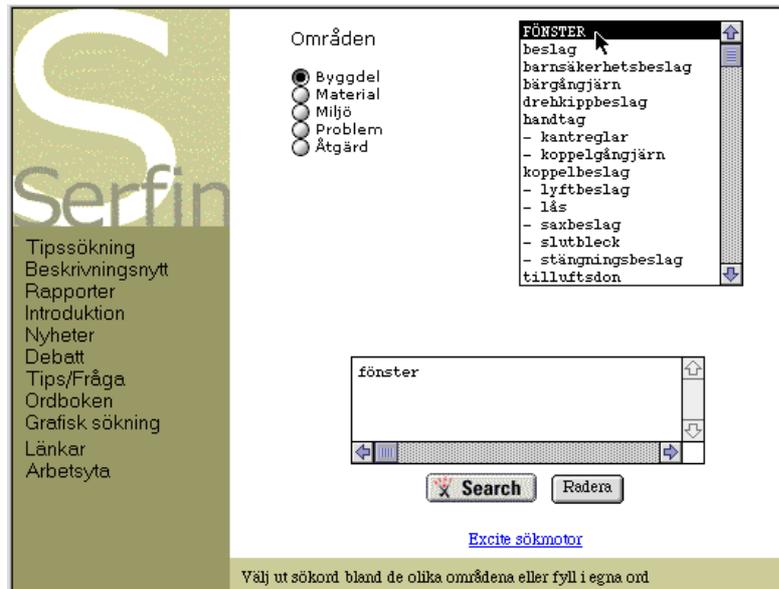


Fig. 2. Searching technical maintenance building information in the Serfin knowledge node. Choose 'Building Part' 'Window' ('Bygghet' 'Fönster') and eventually also Material, Environment, Problem type, Action - (Material, Miljö, Problem, Åtgärd). Add free text at your own wish. Relevance ordered feed-back is provided as well as search on similar documents. You can also send in a question or a tip of your own.

4. Logical Knowledge Containers and Knowledge Nodes

The personal competence and competencies co-operation will in the future as stated above be of central interest. Our personal information storage containers, today often stored in our personal portable computers, accommodate information with highly personal structure and semantics. When we exchange ideas and collaborate with other persons in *projects* we have to harmonize and to some extent formalize our common language.

Three overlapping levels of logical information repositories can be distinguished (1) the personal user dependent, (2) the project/cultural and (3) the global community dependent, see figure 3. On each level we will find long term rather well formalized containers in the form of databases and object stores, which are viewed and handled in project/cultural context through for example Structured Query Language, SQL, and

web browser interfaces. The inter project/cultural linkages can be facilitated with RDF and dynamically adapted on the user levels through use of for example XSL, Extensible Style Language, to specify web document styles.

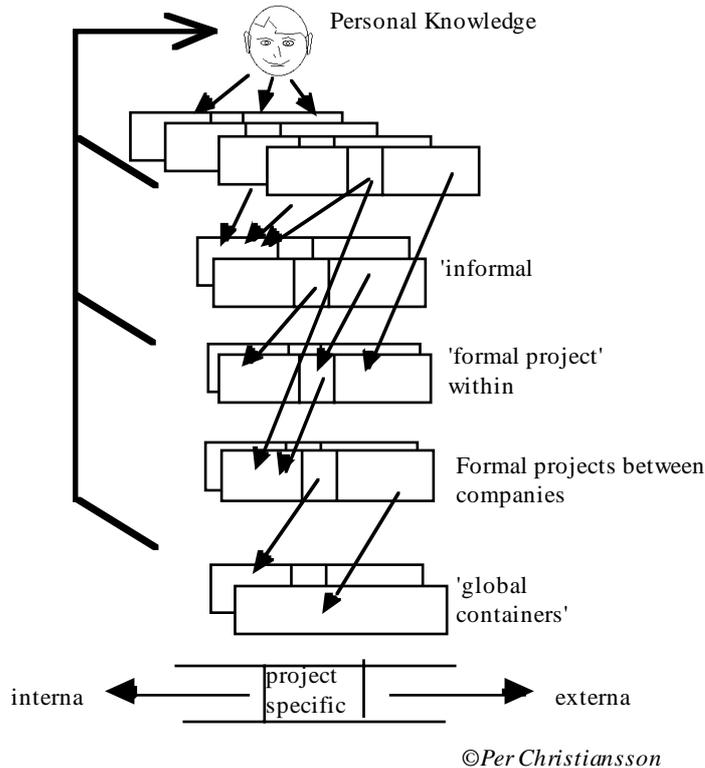
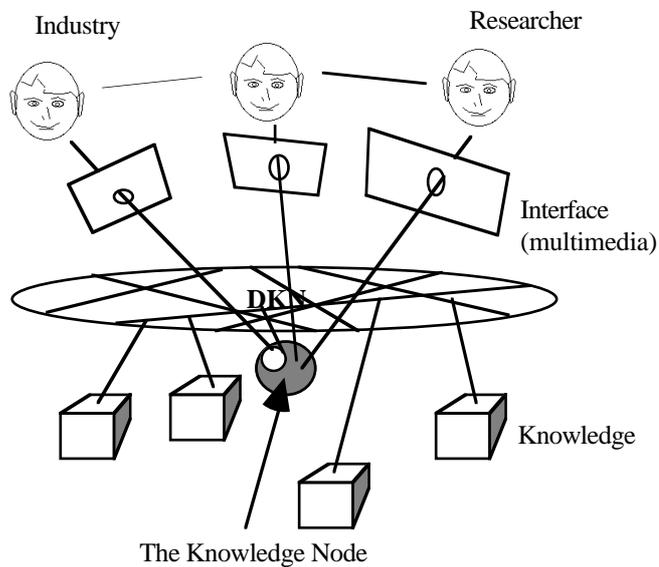


Fig. 3. Information gets more and more formal on its way to the long life global containers. Three levels containers can be distinguished; the personal, projects and the global level.

Persons and artifacts connect to the *Dynamic Knowledge Net*, DKN, [4]. The Internet and its services as World Wide Web today constitutes the DKN. DKN will evolve and perhaps (using metaphors) possess resemblance to the human brains dendrites and axons connecting what in artificial neural networks are called artificial neurons or Processing Elements, PE. [7].

A *Knowledge Node* is kind of high level processing unit and today equal to an URL, Uniform Resource Locator, on the Internet. A knowledge node, [3], has three main functions (a) dissemination of information on request or automatically channeled, (b) two way communication and feed-back capabilities through multimedia interfaces, and (c) access to a local knowledge bank and possibly meta knowledge about other knowledge nodes, see figure 4. The Mercurius and Serfin systems described above are example on Knowledge Nodes.



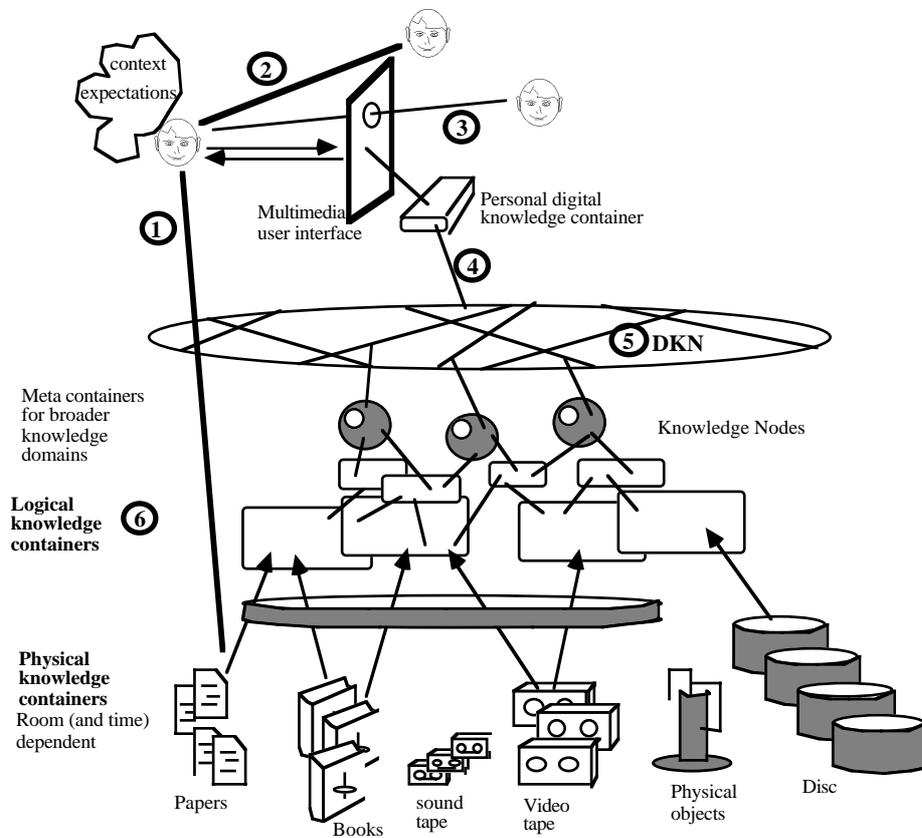
- Access and augment knowledge
- Communcation support

Per Christiansson, 1996

Fig. 4. The Knowledge Node can be regarded as a meta knowledge container and knowledge access control mechanism.

From [2] "The traditional physical information/knowledge containers as books, films, images, papers, etc. are at present in many cases also (or even only) stored in digital form in what we call logical ('virtual') knowledge containers. This latter containers have properties that from now on will completely change our view on how knowledge are structured and represented and interactively presented".

Figure 5 shows how the information access (line '1' in figure 5) to conventional physical knowledge containers as books and video tapes will change when most information is stored in digital format and packaged dynamically for different needs in non-physical (logical) containers. It is also shown how it is possible during collaboration to share information in a common workspace through multimedia interfaces ('3' in figure 5). . We talk about *logical containers* as contrast to physical when the physical wrapping is of importance (books, CDs, hard disks, video tapes, etc.).



Per Christiansson, Dec. 1996

Fig. 5. We will, which is already a fact, communicate and handle digitally stored knowledge in new ways. (1) user searches and finds knowledge stored on paper. (2) Persons meet in real life or (3) use simple multimedia interfaces as telephone. The multimedia interface expands to incorporate more of our senses (Computer Supported Collaborative Work, CSCW, and Virtual Reality, VR, etc.). (4) Part of your personal computer stored knowledge may be connected to the (5) Dynamic Knowledge Net, DKN, see [4]. Logical information containers (6) can be created and dissolved with little effort.

5. Knowledge navigation and search

The human brain is very good at discovering (often unconsciously) subtle hidden patterns in information. With improved search and presentation IT-tools we get help in this process. But we also get some help for deeper analyses to uncover hidden knowledge. We need this help to save time.

We may use tools like WEBSOM, [9], to automatically cluster information and provide us with an ordered map where similar documents lie near each other on the map. In this case the method is based on an unsupervised learning algorithm for analyzing and visualizing high-dimensional statistical data. We can train a neural net through supervised learning for example by feeding it with trigrams (consecutive letters from a text, three at a time) thus finding typical patterns in the text, [12], or train an intelligent agent to help us filter found web-documents based on a user meta model, [8]. We can also use more straightforward navigation tools which provide us with different views for graphic navigation in an URL (for example the Mapucciono Java applet (<http://www.ibm.com/java/education/mapuccino/java.map.html>) from IBM.

Figure 6 provides a basic model with three facets to access information in a selected digital knowledge container.

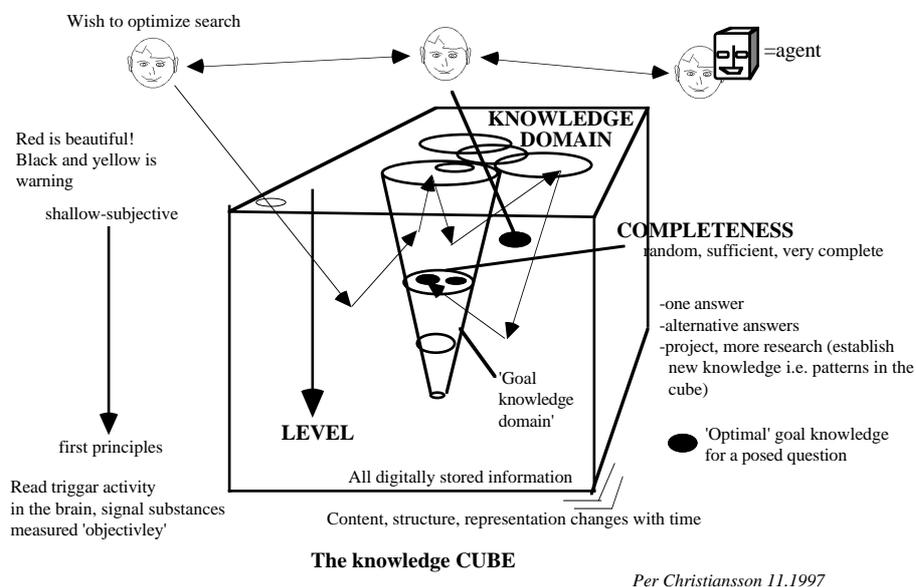


Fig. 6 Knowledge accessed from a node may be characterized according to level, completeness and domain. The one-way arrows denote the users search path towards a known or emerging goal knowledge domain. The result is stored in a logical knowledge container and is part of the 'goal knowledge domain'.

For example, a user wants to search the WWW for technical maintenance tips on removal of fire protection paint from wooden beams or frames. He will possibly be helped by an intelligent agent and start looking in metadata repositories for further links to information in the 'area' of technical maintenance, 'created' after 1990 in the Nordic countries. He may also do a discovery search world wide with no constraints on region or material (the right one-way upward arrow in figure 6 pointing to a new domain). After thus having narrowed in a potential goal domain he continues detailed indexed search in the 'description' parts of those web-objects. These analyses may

well lead to references and a jump to another unexpected knowledge domain. The search ends with a collection of supposedly sufficiently good advises.

The requisites to develop IT-tools to make the scenario come through are present and under development today namely the RDF, Resource Description Framework, and XML, eXtensible Markup Language. See [13], [10].

From [13]: " RDF metadata can be used in a variety of application areas; for example: in *resource discovery* to provide better search engine capabilities; in *cataloging* for describing the content and content relationships available at a particular Web site, page, or digital library; by *intelligent software agents* to facilitate knowledge sharing and exchange; in *content rating*; in describing *collections* of pages that represent a single logical "document"; for describing *intellectual property rights* of Web pages, and in many others. RDF with *digital signatures* will be key to building the "Web of Trust" for electronic commerce, collaboration, and other applications."

RDF using the XML as its main carrier syntax allows us to handle name spaces for different knowledge domains and hopefully support web client mediation between databases.

The RDF data model can be represented as a set of triples {Property Type, Node/Resource, Node or Property Value} or serialized to a tagged text using the XML, eXtensible Markup Language. (This XML-file can be parsed to a tree-like object structure which in its turn simplifies meta level object handling in the Dynamic Knowledge Net, DKN).

XML (a subset of SGML, Standard Generalized Markup Language) extends the HTML, Hypertext Markup Language, in that it focuses on content only and leave the user views (part of the 'user models') to be defined in a separate XSL, Extensible Style Language. XML uses the same formalism as HTML i.e. documents are expressed as nested tagged expressions (<author> <first> nn </first> <last> mmm </last></author>). Mark-up languages based on XML are developed now for different areas, for example; Conceptual Markup Language, CKML, for handling conceptual spaces [5], and to support Electronic Data Interchange, EDI, [1]. See also [10].

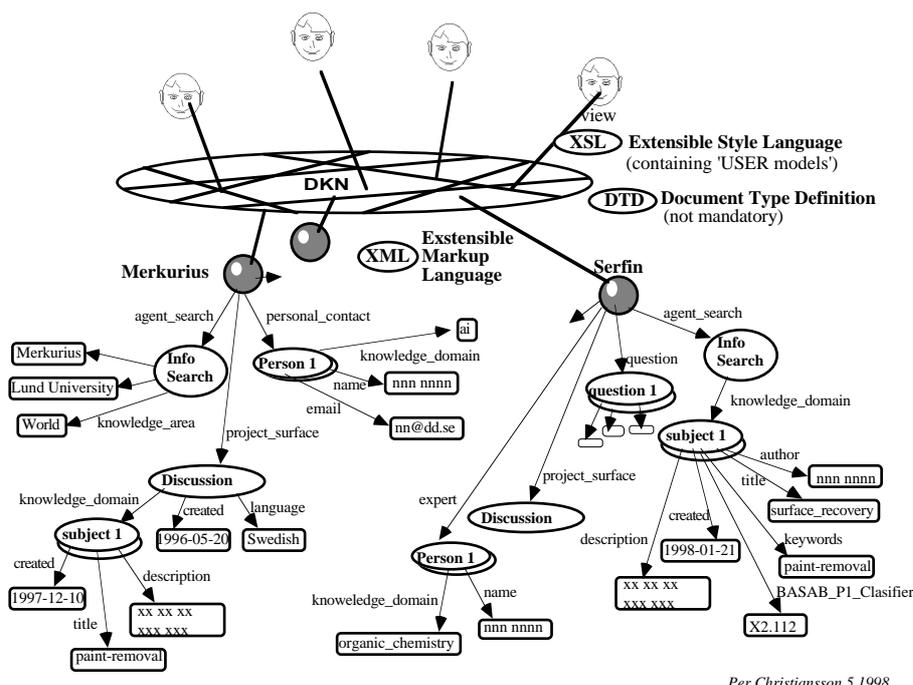
6. Serfin and Mercurius meta level information

The MERKURIUS, figure 1, and Serfin system, figure 2, today do not contain meta-tags. Dublin Core meta-tags, [6], can be semi-automatically created using Reggie, a Dublin Core metadata Java Applet based editor, [10]. SubElements proposals are given from pull-down menus. There is also a Dublin Core Generator, DCdot, from University of Bath, which can generate metadata on existing html pages. See <http://www.ukoln.ac.uk/metadata/dcdot>.

Table 1. Dublin Core Metadata generated by Reggie, [10]

```
<META NAME="DC.Creator.PersonalName" CONTENT="(LANG=sv) Hans Nilsson">
<META NAME="DC.Subject" CONTENT="(LANG=en) hot work, hot air, open flame, window, paint removal">
```

<META NAME="DC.Description" CONTENT="(LANG=en) Removal of paint from tree frame">
 <META NAME="DC.Publisher" CONTENT="(LANG=en) SERFIN Expert">



Per Christiansson 5.1998

Fig. 7. Part of the top level contents of the Mercurius and Serfin knowledge nodes expressed as directed graphs according to the Resource Description Framework, RDF. The application areas for the XML, eXtensible Markup Language, XSL, Extensible Style Language, and Document Type Definition, DTD, (logical structure of document) are also shown.

The fifteen Dublin Core metadata tags contain: Title, Author or Creator, Subject and Keywords, Description, Publisher (of the electronic version), Other Contributor, Date, Resource Type (technical report, etc.), Format (html, pdf,...), Resource Identifier (retrieval identifier), Source (from the electronic version it was derived), Language, Relation (with other resources), Coverage (geographical or temporal), Rights Management (link to ownership information).

Figure 7 shows how the Knowledge Nodes Mercurius and Serfin attached to the Dynamic Knowledge Net, DKN, can be described using directed graph notation according to the forthcoming Resource Description Framework, RDF. Such a description can be used in the conceptual modeling of the systems and later to facilitate high level couplings between the knowledge nodes. For example to discover pertinent competence persons and projects in other knowledge domains, for

comparative analysis of different knowledge domains, and to harmonize application vocabulary development.

7. Conclusions

We can now see a clear break-point in the development of the future meta leveling of the globally stored information and the development of a knowledge node framework. Much work will be spent on compiling non-overlapping and comparable vocabularies and name spaces for different application areas.

The container descriptions (now 'A longer, textual, description of the resource in Dublin Core terminology) are mostly written by their authors. But other commentary and feed-back descriptions will also be written and associated with the same content. These will be very important when container content quality shall be estimated.

There are clear links between RDF and Entity-Relationship descriptions which will be helpful when WEB documents and objects are going to be generated from long term highly formalized relational database containers.

The abstraction process (aggregation, characterization, and generalization) will be even more interesting than before in connection with studying collaboration between different competencies (architects, engineers, clients, environmental planners,..) in order to capture, formalize and link 'equivalent' concepts.

The agent concept will be used extensively to wrap different kinds of complex and compound knowledge representations. The above related languages will support the definition of both the inter agent and agent human communication formalisms.

We now experience the beginning of a shift to a global totally digital information handling. It is only five years since we started publish on the web and we are already in a phase of re-engineering it. May be it is time to reconsider some of the pioneering works done by for example Ted Nelson (HomePage at <http://www.sfc.keio.ac.jp/~ted/index.html>.) regarding version handling and hypertext growth.

Acknowledgments

I want to thank my research colleagues Fredrik Stjernfeldt and Gustav Dahlström at the KBS-Media Lab, Lund University, for their collaboration in the MERKURIUS (The Foundation for Knowledge and Competence Development KKS-2343:1/95) and SERFIN projects (The Swedish Building Research Council, BFR-950549-0).

References

1. Bryan, M.: Guidelines for Using XML for Electronic Data Interchange. Version 0.05, 25th January (1998). XML/EDI Group.
<http://www.geocities.com/WallStreet/Floor/5815/guide.htm>

2. Christiansson, P.: Experiences from developing a Building Maintenance Knowledge Node. In CIB Proceedings Information Technology Support for Construction Process Re-Engineering, IT-CPR-97. (1997) 89-101.
(<http://delphi.kstr.lth.se/reports/cibw78cairns1997.html>).
3. Christiansson, P.: Knowledge communication in the building industry. The Knowledge Node Concept. In Construction on the Information Highway. CIB Proceedings 198 (ed. Z. Turk) (1996) 121-132.
(<http://delphi.kstr.lth.se/reports/cibw78bled96.html>)
4. Christiansson, P.: Dynamic Knowledge Nets in a changing building process. Automation in Construction, Vol 2, nb 2, Elsevier Science Publishers B.V. Amsterdam, (1993) 307-322
5. Conceptual Knowledge Markup Language, CKML. (Robert Kent, Washington State University, Christian Neuss, Technische Hochschule Darmstadt)
<http://wave.eecs.wsu.edu/WAVE/Ontologies/CKML/RDF-to-CKML.html>
6. Daniel Jr., R., Ianella R., Miller E.: Expressing the Dublin Core in the Resource Description Framework: Suggestions based on an early examination of the problem. Los Alamos National Laboratory. (7 A4 pages). (1997)
http://www.acl.lanl.gov/~rdaniel/RDF/DC/ExpDC_2.html.
7. Freeman, J.A., Skapura, D., M.: Neural Networks. Algorithms, Applications, and Programming Techniques. Addison-Wesley Publishing Company. Reading Massachusetts. (1991) 17-18
8. Lagerstedt, R., Christiansson, P., Engborg U.: User Models in Search and Navigation Systems on the Internet". Proceedings of the Third Congress held in conjunction with A/E/C Systems'96. ASCE Technical Councils on Computer Practices. (1996) 21-27
(<http://delphi.kstr.lth.se/reports/aec96.html>)
9. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Self-Organizing Maps of Document Collections. Neural Networks Research Centre, Helsinki University of Technology. (5 A4 pages) (1997). <http://www.diemme.it/~luigi/websom.html>
10. Mace, S., Flohr, U., Dobson, R., Graham, T.: Weaving a Better Web. BYTE, March (1998) 58-68.
11. Metadata Tools and Services. Distributed Systems Technology Center. University of Queensland Australia. <http://metadata.net/dstc/>.
12. Modin, J.: KBS-Class: A neural network tool for automatic content recognition of building texts. Construction Management and Economics. Special issue on Information Technology in Construction. (1995) 411-416
13. Resource Description Framework (RDF) Model and Syntax W3C Working Draft 16 Feb 1998. <http://www.w3.org/TR/WD-rdf-syntax/>