

Electronic Publishing 2000, Kaliningrad/Svetlogorsk

**From Open Source to Open Information:
Collaborative Methods in Creating
XML-based Markup Languages**

Georg Rehm

Henning Lobin

Justus-Liebig-University Giessen
Research Unit for Applied and Computational Linguistics
Otto-Behaghel-Strasse 10 D
35394 Giessen, Germany

Phone: +49 641 99 29050, -29051

Fax: +49 641 99 29059

`Georg.Rehm@germanistik.uni-giessen.de`
`Henning.Lobin@germanistik.uni-giessen.de`

January 21, 2001

1 Introduction

Until the beginning of the last decade, the Internet was primarily used by scientific, educational, and military organisations for the exchange of information such as data files and electronic mail. The introduction of the easy-to-use hypertext system *World Wide Web* (WWW) has, however, begun a new era of the world-spanning computer network.

In this paper we examine a part of the Information Marketplace (Dertouzos, 1997) that will give users of the WWW a wide range of new possibilities for gathering information, a task that is predominantly carried out using index-based (e. g., `www.google.com`, `www.metacrawler.com`) or catalogue-based search engines like `www.yahoo.com`, for example. One of the major shortcomings of search engines is the lack of semantic certainty that results from both the absence of structure in the indexed documents as well as insufficient methods of information extraction and information retrieval regarding a generalized conceptual level (vs. the statistics-based word level that is still the most common method in search engine technology). For this reason, the user of a search engine is very often confronted with lots of documents that are beyond the scope of his or her search query.

The aforementioned lack of explicit structure in Web documents will be overcome in the next few years by an augmented use of XML (*Extensible Markup Language*, Bray et al., 1998) and a simultaneous turning away from HTML (*Hypertext Markup Language*, Raggett et al., 1997) that only allows an annotation of rather coarse textual elements. However, this new structural variety and liberty of XML bears certain dangers: As XML allows a free definition of concrete markup languages like HTML, a lot of proprietary XML-based annotation schemata could emerge that, in turn, make the process of automatic information extraction by search engines not easier but even more difficult, as a large part of the Internet's and especially the World Wide Web's success is based on the standardization of concrete markup languages.

In this paper, we outline a possible development that may counteract this XML babel. The main impetus for our prognosis is a paradigm in software development which has been successful for almost 20 years now. This paradigm, called *Open Source* (DiBona et al., 1999; Raymond, 1999), made possible, among other software packages, the free operating system *Linux*. The impetus behind Open Source will give new and decisive impulses for the use of quasi standardized XML-based markup languages and concrete schemata for related standards. These impulses will result in what we want to call *Open Information*.

2 Markup Languages and the World Wide Web

The markup language that is used to create Web documents is called HTML, Hypertext Markup Language (Raggett et al., 1997). HTML allows the enrichment of textual data with a set of precisely defined structuring and formatting instructions, *tags*. Using tags, one can explicitly mark rather coarsely structured textual elements like, e. g., headlines of various levels, tables, different kinds of lists (ordered, unordered etc.), embed inline images and hyperlinks. Tags cannot be combined in arbitrary ways. There is a formal, rule-based system, a grammar called the *Document Type Definition*, that specifies both the names of tags and their

possible combinations. This grammar, the formal definition of HTML, was created using the *Standard Generalized Markup Language* (SGML, Goldfarb, 1990). SGML is an abstract and very complex formal language that has been developed in order to have at hand a flexible formalism for defining arbitrary markup languages.

The explosion-like spreading of the Web in recent years and the emergence of new methods for interactivity made perfectly clear that HTML was only a first step for the Internet with regard to modelling information in structured ways. As HTML is only one single SGML application among infinitely many others, yet other applications seem to be better fitted for certain necessities within the World Wide Web domain, one can ask the question why not to utilize the complete possible range of SGML applications for the Web instead of one single, obviously limited language. If this were the case, specialised SGML systems could be used on the Web, as well as flanking standards as, e. g., HyTime (*Hypermedia/Time-Based Structuring Language*, ISO10744, 1997) for more sophisticated linking mechanisms and DSSSL (*Document Style Semantics and Specification Language*, ISO10179, 1996) for structural transformations and a more flexible and generic approach for rendering documents.

Even though this might seem obvious, it is hard to put into practice. The Web has become a mass medium and as such, it has established facts that are not compatible with the very extensive standard of SGML. SGML is not only very complicated but, in a formal way, so complex that online applications would probably not be able to process their information in reasonable time. A lot of SGML's properties reflect the state of the art of the early eighties, a time when no one thought of sharing both SGML applications and their data in distributed computer networks.

The emergence of these deficiencies was the origin of XML which has been declared an official standard by the *World Wide Web Consortium* (W3C) in early 1998 (Bray et al., 1998). XML is nothing but a simplified version of SGML. Every piece of information that has been marked up in XML is valid SGML information at the same time. The definition of XML is much more concise, shorter and more convincing in a logical way, as all the seldomly used and subtle details of SGML have been abandoned without restricting the power of the formalism. This reduction has been done so conclusively and coherently that XML is nowadays used in a wide range of different fields of application, even where delivering the data online is not of main interest.

One can consider XML as being an instrument for modelling structured information (Lobin, 2000). The idea of structured information is based on several underlying observations that have been originally made with regard to text documents: One can distinguish different hierarchical levels in texts. At first, there are sequences of letters and words, e. g. in a headline or in a quotation. Furthermore, there are abstract units that constitute categories like "headline" or "section". These abstract units are—in contrast to the textual units—not realized by means of linguistic signs but by typographic ones: The typeface of a headline is larger than the one of continuous text, the headline itself is separated and often numbered. The second observation is that the order of the aforementioned abstract and concrete units is not arbitrary but has to follow certain rules. On the one hand, these rules—the above-mentioned *Document Type Definition*—specify the hierarchical structure of abstract units of information to subordinate abstract or concrete units, on the other hand they particularize the linear order of units of the same hierarchical level. One can combine these rules to a

grammar of information units. The third observation: These grammars can be built such that information units along with their hierarchical and linear relations arrange themselves in the form of a tree: at the top, there is a root element that represents the text as a whole, the daughter nodes beneath the root element represent the parts that constitute the text on the upper level. This fragmentation is repeated until one arrives at the level of elementary textual units. In summary, one can say that structured information is the rule-driven arrangement of information units, just like we consider well-formed sentences as the rule-based composition of words into sentences.

Several other formalisms that have been developed simultaneously at the *World Wide Web Consortium* deal with the combination of XML documents (XPointer, XLink), the concurrent processing of more than one document type definition in one document (*Namespaces*) and the visualization of XML documents in browsers (*Cascading Style Sheets*, CSS, and *Extensible Style Language*, XSL).

One technique proposed by the World Wide Web Consortium for the creation of a basis that will allow a directed and efficient exploration of knowledge is the XML-based *Resource Description Framework* (RDF, see Lassila and Swick, 1999; Brickley and Guha, 1999). RDF provides mechanisms for the author or editor of a document to annotate metadata—data *about* data. Examples for metadata of a Web document are, e. g., the name of the author, the date of the latest revision, several keywords, a link to the associated organization and so on. RDF has been developed to guarantee a comprehensive and consistent explication of metadata in Web documents in order to simplify the exploration of large stocks of information as well as stating search queries more precisely. RDF provides, much like XML, only the definition of concrete schemata that can, in turn, be used to annotate documents. Several immediate problems come to mind: What kind of vocabularies are used for the definition of concrete RDF schemata? Which level of detail has to be considered when marking up metadata, and which kind of data should be allowed marking up this very data regarding the respective subject-matter or genre? Several answers to these questions with regard to the general classification of objects have been attempted in different fields (librarianship, architecture, the arts etc.) by proposing schemata (see Hudgins et al., 1999; Baca, 1998; Marchiori, 1998, for an overview). One such project is very popular in the Web, the *Dublin Core* initiative (see <http://purl.org/dc/> and Weibel et al., 1999), which is still work in progress. The Dublin Core aims at developing a mutual and extensible core of all suggested RDF schemata. The Dublin Core schema defines three groups of elements: *Content* (with units like, e. g. *Title*, *Subject*, *Description*, *Type* etc.), *Intellectual Property* (*Creator*, *Publisher*, *Contributor*, *Rights*) and *Instantiation* (*Date*, *Format* etc.). RDF is—particularly in conjunction with the Dublin Core schema—currently in widespread use throughout a lot of websites in order to annotate resources, i. e. Web documents and embedded objects.

The *Topic Maps* (ISO/IEC13250, 1999, cf. Rath, 1999, formerly known as *Topic Navigation Maps*) standard is another example of supporting formalisms that will, in the next few years, dramatically change our perception and use of the World Wide Web. The Topic Maps architecture, not unlike RDF, provides methods for explicitly annotating metadata in a standardized way. But while RDF is primarily concerned with “real” metadata (the author’s name, e. g.), Topic Maps deal with the contents of a document and relationships that it shares with other documents or even groups of documents, or, more general, information objects.

The purpose of making these relations explicit is, again, the simplification of content-based navigation and filtering. The basic idea is to merge certain information objects into groups that can be organized on an abstract level. Relations to concrete information units can be established by linking mechanisms, the entirety of content-based relations is a meaningful, independent type of information in itself. Relations between information units, *associations*, can, too, be merged into groups, so that one can deal with filtering on this relational level.

The main advantage in standardizing topical structures is that Topic Maps from totally different domains or fields of knowledge can be effectively merged together so that an all-encompassing semantic network can be formed. One can think of the possibility that the Web can be augmented by these semantic networks that can be used for content-based linking and navigation of contiguous ranges of knowledge.

3 The Open Source-Model in Software Development

The *Open Source* model in software development (cf. Vixie, 1999, for a comparison of this approach with traditional methods) comprises the free publishing, transfer, and permission to modify the source files of a program—the abstract instructions implemented by programmers in computer languages like, e. g., *C* or *Java*, that have to be converted into machine instructions by means of a compiler before executing them (cf. DiBona et al., 1999). The success of this approach (most of the electronic mail- and Web-servers are running Open Source software packages, see O'Reilly, 1999) is considered to be a phenomenon that is closely connected with the Internet and that has and will entail extensive implications on the whole line of communication- and information-technology.

The Open Source concept has its roots in thoughts that were first expressed in the middle of the eighties by software developer Richard M. Stallman. At that time, Stallman quit his job at MIT's AI laboratory to fully dedicate himself to the development of a free—"free as in freedom" (Stallman, 1999), not *free* as in *for free*—operating system that should be compatible with the industry's *de facto* standard, the UNIX system. Stallman did not agree to the general business trend of many software manufacturers to deny the publication of software packages' source codes, as they considered these to be corporate secrets that have to be protected. Only the source code enables the trained user and programmer to look for faulty segments of code, *bugs*, in order to fix them, or to integrate new features into existing programs by developing and adding code, or to take different parts of code from various programs as well as newly developed code to release software packages that have a completely new functionality. Without the source code, users do not have these possibilities and can—e. g. in case of a software malfunction—only hope that the manufacturer will fix this problem in the next release of the flawed software. Stallman was of the opinion that programmers do have an ethical right to free software and began his work on the operating system GNU (this recursively defined acronym means "GNU's Not UNIX"). On the basis of an existing commercial variant of the UNIX system, Stallman implemented his versions of an editor (*Emacs*), a compiler (*gcc*), and several tools (*gdb*, *make*) in order to replace the corresponding modules of the commercial system by free modules that he developed himself one by one. Furthermore, he incorporated other already existing free software packages into

the GNU system, e.g. the type-setting system *T_EX* or the windowing system *X Window* (see Buthenuth and Mock, 1992). The general public's interest in the philosophy of free software primarily increased, when the powerful editor Emacs was released which also was an impetus for different programmers to join Stallman and to work on additional components of the GNU system. Thereupon, Stallman established the *Free Software Foundation* (FSF, see <http://www.fsf.org>), a non-profit organization that should help realizing his general vision to create a UNIX-compatible operating system which can be freely shared in both binary- and source-form with other people.

When the first components of the GNU system became mature and stable pieces of software, Stallman was in the need of a legally protected license, to prevent commercial software manufacturers from simply adopting these modules, modifying them, and finally releasing them as parts of their own proprietary industrial properties. This was the main motivation (see Stallman, 1999, for others) for the development of the GNU *General Public License* (GPL) and the idea behind the distributional model of "Copyleft" ("All rights reversed"). The notion of *Copyleft* means enforcing the programmer's copyright with the simple purpose of explicitly marking his software as being *free*. This license allows anyone to use, copy, and modify, even distribute, any software package that is licensed under the GPL. However, no other restrictions than the GPL itself must be imposed on the software—modified or not: Every program that has been derived from GNU/GPL software by modifications, is subject to the GPL as well, so this model guarantees that originally free software will be free in alternative and extended versions, too, and will, therefore, always keep the status of being free in the legal sense of the GPL.

Within a couple of years, a collection of various programs emerged under the auspices of the FSF. It was common that dozens of programmers were participating in the development of a piece of software, often coordinated via the Internet, that was furthermore used to exchange new versions of a program's code. But the GNU system could not operate as a complete and independent UNIX system as the most important and most complex part still was not implemented: the kernel. The kernel of an operating system manages accesses to a computer's memory, its hard disks, schedules processes, provides drivers for all kinds of hardware etc. This obvious gap has been filled by *Linux*, developed by computer science student Linus Torvalds. Initially, Linux was only the name of a free UNIX kernel, but nowadays it is the common term for a UNIX-compatible operating system—comprised of the Linux kernel, the GNU system, and several other components—that is in widespread use in research and development departments and in educational institutions.

At the beginning of 1990, Torvalds was able to find a lot of participants for his rather complex software project through the very effective means of Internet-communication. His aim was to develop a functional UNIX kernel for educational purposes, and he took the UNIX variant MINIX, also created for the educational field, as the initial development basis (Raymond, 1999, reports that it is a crucial factor for the success of an Open Source project if one can build upon available software packages that are already implemented by third parties). Very frequently—in average about once a week—Torvalds released new versions of Linux on the Internet and soon, in days or even hours, he received electronic mails by hundreds of volunteers about errors in the most recent version of the kernel, often along with source code fragments that fix these very errors, so that Torvalds was able to incorporate

these pieces of program code into his own version of the kernel's sources. In a remarkably short period of time, a very robust, stable, and powerful UNIX kernel began to take shape, a kernel that was perfectly fitted to fill the most crucial gap the GNU system had to suffer from for several years.

At the beginning of 1998, a group around programmer Eric S. Raymond realized the potential of this approach in software development, that lead to such powerful and efficient pieces of software like *Linux*, the *de facto* standard for electronic mail servers, *sendmail* or the highly popular Web-server software *Apache*: a distributed group of enthusiastic programmers is—only connected through the Internet—able to develop extremely capable and effective software in such a fast time that traditional software manufacturers could never achieve. The motives behind the individual volunteers are manifold: One may participate in a project just to get to know the latest state of the art techniques in programming servers, e. g., another one may want to head a team of developers to acquire certain expertise, with the consequence of increasing his or her chances in the job market, others may take part to satisfy their own ambitions and yet others may just need specialized software that no one else has written, so they write these programs themselves (Ettrich, 2000). One of the reasons that make this approach so succesful is that a knowledgable user of a certain software package is able to report errors via electronic mail, and, furthermore, can even supply proposals for the correction of an error by sending in improved source code (Raymond, 1999). Only the free availability of the sources of each program makes this approach possible. Raymond, along with his group, called this principle “*Open Source Software*” (see <http://www.opensource.org>), to delimit their approach from the potentially ambiguous notion of “free software”. Furthermore, they consider Open Source to be a marketing strategy to suggest the—as has been proved—successful concept of free software to commercial software vendors (see <http://www.berlios.de> for a similar initiative of the Research Institute for Open Communications Systems, FOKUS, of the German National Research Center for Information Technology, GMD; BerliOS's primary goal is to promote the concept of Open Source into small and midsized enterprises and the public administration in Germany).

Recently, a trend in the Open Source community emerges to no longer concentrate on operating system-specific issues and their implementation but to develop software that can be used by anybody as, for example, the very comfortable graphical desktop system KDE (*K Desktop Environment*) and the embedded office suite *KOffice* that provides very powerful text processing- and spreadsheet-applications, among others. But rudimentary prototypes that originate from research projects are available under Open Source licenses as well. One such prototype that could have a direct impact on the public perspective regarding Linux—provided that enough voluntary programmers are interested in doing support and development—is *Sphinx*, a prototype system for recognizing natural language speech (originally developed at Carnegie Mellon University), that at some time in the future could be one of the standard components of the Linux system for dictating letters, texts, electronic mails or speech-driven desktop navigation.

The Open Source paradigm is no longer restricted to the area of software development. At the beginning of 1998, Ph.D. student David Wiley wanted to publish several course materials on the World Wide Web and he wanted to make sure that—if someone adopts his material—his name, being the original author, will still be part of the documents. Further-

more, he wanted to guarantee that the material is not subject to changes that do not lie within its scope being educational material. With the support of both Stallman and Raymond, Wiley developed the *Open Content License* (OPL) on the basis of the GNU General Public License (see <http://www.opencontent.org>). The OPL allows a non-profit application of content and enforces unmistakable markings of the passages that have been modified by third parties. In case someone uses this material, the license guarantees that derived contents underlie the OPL as well. At the time of writing, the Website of the Open Content Initiative lists about 150 different Websites that have put their material under the OPL license (search engines currently report more than 5.500 hits to the keyword “opencontent”). These Websites not only offer educational material for a wide range of academic fields, but even essays and pieces of music. Several books are currently in preparation that will be published under a variant of the Open Content License that has been specifically modified for traditional print media.

Yet another example of an initiative that works in a similar manner to *Open Content* is Project Zvon (see <http://www.zvon.org>). The project members want to make use of current methods in text technology (XML, XSL) in order to simplify the free publication of all kinds of documents with the current focus on technical, Internet-related tutorials. Project Zvon wants to provide a sound technical environment for electronic publishing so that anybody who has got expertise in a certain field but is rather computer-illiterate can publish his or her information “on a large scale” under “something like [the] GNU Public License” that should be applied “for all aspects of information sharing”.

The Berkman Center for Internet and Society at the Harvard law school examines in projects like *Open Law*, *Open Governance*, or *Open Education* “real and possible boundaries in cyberspace between open and closed systems of code, of commerce, of government, and of education, and the relationship of law to each” (cf. <http://cyber.law.harvard.edu>). The project Open Law, e. g., provides interested Internet users with a platform to share their opinions and experiences regarding current Internet-related lawsuits. The goal of these forum-based discussions is to find arguments and to elaborate pleadings with the following premise: “an open development process best harnesses the distributed resources of the Internet community. By using the Internet, we hope to enable the public interest to speak as loudly as the interests of corporations.”

4 Open Information

Sections 2 and 3 examine the technological level and the progress of the World Wide Web since its beginning and the origins and crucial points of success of Open Source. Despite the superficial lack of affinity between these two subject matters, there could be, to our mind, an interesting trend in the years to come towards an application of the strategies and factors that have made the Open Source approach so successful, onto the creation of Web contents and especially metastructures.

Section 2 makes clear that there is a steady increase in structure regarding the Web and its underlying markup languages: The initial version of HTML was developed by Berners-Lee (1999, p. 41) “to look like” an SGML application to improve the internal acceptance of

his hypertext system at CERN. In the following years, more and more complex versions of HTML have been defined (this time actually based on SGML) that emphasized the issues of presentation and design. Finally, this trend of *less structure, more design* came to an end when the World Wide Web Consortium published the XML recommendation due to the immanent collapse of the Web. Now that a well-defined structure of documents can be ensured with the use of XML and its flanking standards (XSL, XPointer, XLink etc.), a lot of research departments are currently busy defining and specifying markup languages and representation formalisms for metadata like, e. g. RDF, or Topic Maps.

RDF and Topic Maps provide methods that differ in complexity to explicate information about single Web documents as well as relations that hold between multiple groups of documents. But there are more abstract fields of application: The *Open Directory Project* (<http://dmoz.org>) is a Web catalogue (similar to Yahoo) maintained by about 20.000 voluntary Internet users who take care of one or more categories each. This maintenance work includes both the integration of new documents into existing categories of the Open Directory hierarchy and the creation of new categories. This hierarchy is organized in a set of 15 trees, which have in their highest level concepts like *Arts, Health, Recreation, or Science* and subsequently branch into more specialized nodes (like, e. g., *Arts: Movies: Genres: Silent Movies*, or *Science: Social Sciences: Language and Linguistics: Applied Linguistics*). Of special interest is, on the one hand, the fact that the creation—including the until now unknown concept of peer reviewing in this area—of the catalogue contents through *users* (and not, like Yahoo, through a group of full-time working editors) and, on the other hand, the fact that the complete hierarchy of the Open Directory Project is available for download as a huge RDF-annotated file which underlies a GPL-like license. At the time of writing, the hierarchy contains about 270.000 entries and multiple explicit relations that connect these entries.

If we consider these premises with regard to the thesis (Raymond, 1999, pp. 33) that one of the important factors of a successful Open Source project is an already existing relevant and modifiable basis implemented by third-parties, the Open Directory hierarchy would provide such a basis for further developments of a more specific hierarchy, e. g., or to use parts of this structure in other Open Source projects. In fact, the Open Directory itself mentions about 130 other “sites using [Open Directory Project] data”. The use of such a hierarchy—or, more generally, a semantic network—is not restricted to search engines. One could think of many possible fields of application, e. g. for means of disambiguation in natural language processing systems or even as part of more intelligent graphical user interfaces or online navigational aids (*agents*). As a consequence, the concept of team-based sharing and modification of freely available data is not necessarily limited to source files (*Open Source*) or the publishing resp. discussion of concrete contents (*Open Content, Open Law*), but can be extended to general and abstract meta structures that we want to call *Open Information* due to their almost unlimited range of use.

Raymond (1999) specifies, as already mentioned in section 3, additional factors for the success of the Open Source paradigm. A reasonable transfer of this concept into the notion of *Open Information* necessitates closer examination of these factors for their applicability. Open Source is successful because enthusiastic and experienced programmers enjoy implementing software of high quality, e. g. in order to meet certain requirements one may have, or to boost one’s professional career. This factor can only hold for the Open Information model

if the following condition is fulfilled: At first, there must be a certain need for freely available XML- or RDF-schemata or Topic Map-hierarchies. This need could be created, e. g., by an extensive support of these formalisms (that does currently not exist to its full extent) in the market-leading browsers, editors and other dedicated software (Berners-Lee, 1999, p. 172, speaks about a “common new genre on the Web” in a very similar context).

As soon as users will learn to value the advantages of these new technologies, especially much easier methods for searching and navigation, they will—as the past has shown (from Yahoo to the Open Directory Project, from proprietary search engine technology to Open Source products with equivalent features etc.)—be eager to create comparable but *free* information-infrastructures and offer these as *Open Information*. Another current trend in information technology is that more and more companies release their programs and source code under Open Source licenses (due to the success of Open Source) in order to ensure a global peer reviewing by thousands of programmers worldwide. We are of the opinion that it is most likely that this trend will not be limited to software or concrete Web contents, but will be extended to metastructures that will be created, published and maintained by volunteers and even companies. One classic example is the *DocBook* DTD (cf. <http://www.oasis-open.org/docbook/>) that has been co-developed by the publishing house O’Reilly & Associates in 1991. DocBook is still maintained and in widespread use in both traditional and electronic publishing, due to the existence of highly modular style sheets for manipulating DocBook-annotated data in standardized ways (DSSSL, XSL). As a consequence, it’s safe to say that Open Information could aid in the process of cross-media convergence, from traditional paper-based to all kinds (WWW, CD ROM, electronic books etc.) of electronic publishing.

The abovementioned development will comprise—similar to Open Source projects—user groups from varied backgrounds that will, by means of RDF and/or Topic Maps, create definitive semantic networks for their respective fields of interest, always reflecting the current state of knowledge. Via the Internet, these groups will both communicate and share their projects, data, and results. Thematically related projects will try to combine or even merge their hierarchies in order to enforce consistent and uniform schemata. In this manner, the following years could really see the emergence of what Berners-Lee (1999, pp. 177) calls the “Semantic Web” (“a web of data that can be processed directly or indirectly by machines”): All the different hypertext documents will be put into a global context by the users themselves (“This all works only if each person makes links as he or she browses, so writing, link creation, and browsing must be totally integrated”, Berners-Lee, 1999, p. 201). This global context will be, and that is the main advantage in comparison to the current situation of unstructured chaos, explicitly structured, so that processing—e. g. for purposes of automatic reasoning by means of a variety of topic-hierarchies in order to discover new relations that hold between non-neighbouring concepts—of the data will be almost guaranteed: “We will solve large analytical problems by turning computer power loose on the hard data of the Semantic Web.” (Berners-Lee, 1999, pp. 201).

Raymond (1999, p. 227), too, thinks that the Open Source approach will, in the near future, have a certain influence on fields beyond the scope of software development: “I expect the open-source movement to have essentially won its point about software within three to five years. Once that is accomplished, and the results will be manifest for a while, they will

become part of the background culture of non-programmers. At *that* point it will become more appropriate to try to leverage open-source insights in wider domains.” In his talk at the XML conference in 1999, Peter Murray-Rust called the definition of XML semantics one of the major problems. He warned that the development of varied XML schemata could emerge a “semantic and ontological warfare” which could only be prevented by independent non-profit organizations. The concept of *Open Information* and a sensible linking of several related projects could, in our view, be a possible solution for this threatening problem.

References

- BACA, MURTHA (editor) (1998): *Introduction to Metadata – Pathways to Digital Information*. Getty Information Institute.
- BERNERS-LEE, TIM (1999): *Weaving the Web – The Original Design and Ultimate Destiny of World Wide Web by Its Inventor*. San Francisco: Harper San Francisco.
- BRAY, TIM; PAOLI, JEAN AND SPERBERG-MCQUEEN, C. M. (1998): “Extensible Markup Language (XML) 1.0”. Technical Specification, World Wide Web Consortium. Available online: <http://www.w3.org/TR/1998/REC-xml-19980210>.
- BRICKLEY, DAN AND GUHA, R.V. (1999): “Resource Description Framework (RDF) Schema Specification”. Technical Specification, World Wide Web Consortium. Available online: <http://www.w3.org/TR/PR-rdf-schema/>.
- BUTHENUTH, ROGER AND MOCK, MARKUS U. (1992): “Abseits vom Kommerz – Die Philosophie des GNU-Projekts”. *c’t, Magazin für Computertechnik* (3): pp. 62–65.
- DETOUZOS, MICHAEL (1997): *What will be. How the new World of Information will change our Lives*. New York: HarperEdge.
- DIBONA, CHRIS; OCKMAN, SAM AND STONE, MARK (editors) (1999): *Open Sources: Voices from the Open Source Revolution*. Peking, Cambridge, Köln etc.: O’Reilly & Associates.
- ETTRICH, MATTHIAS (2000): “Wer kodiert? – Gedanken zur Freie-Software-Szene”. *iX, Magazin für professionelle Informationstechnik* (1): pp. 112–115.
- FEUERBACH, HEINRICH T. AND SCHMITZ, PETER (1999): “Freiheitskämpfer – Entwicklung freier Software gegen Patentierung”. *c’t, Magazin für Computertechnik* (16): pp. 79–81.
- GOLDFARB, CHARLES F. (1990): *The SGML Handbook*. Oxford: Oxford University Press.
- HUDGINS, JEAN; AGNEW, GRACE AND BROWN, ELIZABETH (1999): *Getting Mileage out of Metadata – Applications for the Library*, volume 5 of *LITA Guides*. Chicago: American Library Association.
- ISO10179 (1996): “Information Processing – Processing Languages – Document Style Semantics and Specification Language (DSSSL)”. International Standard, International Organization for Standardization, Genf. Available online: <http://www.ornl.gov/sgml/wg8/>.
- ISO10744 (1997): “Information Processing – Hypermedia/Time-Based Structuring Language (HyTime) – Second Edition”. International Standard, International Organization for Standardization, Genf. Available online: <http://www.ornl.gov/sgml/wg8/>.

- ISO/IEC13250 (1999): “Information Technology – Document Description and Processing Languages – Topic Maps”. International Standard, International Organization for Standardization, Genf. Available online: <http://www.ornl.gov/sgml/wg4/>.
- LASSILA, ORA AND SWICK, RALPH R. (1999): “Resource Description Framework (RDF) Model and Syntax Specification”. Technical Specification, World Wide Web Consortium. Available online: <http://www.w3.org/TR/REC-rdf-syntax/>.
- LOBIN, HENNING (2000): *Informationsmodellierung in XML und SGML*. Berlin, Heidelberg, New York etc.: Springer.
- MARCHIORI, MASSIMO (1998): “The Limits of Web Metadata, and beyond”. *Computer Networks and ISDN Systems* (30): pp. 1–9. (auch: Proceedings of the 7th International World Wide Web Conference, Brisbane, Australien).
- O’REILLY, TIM (1999): “Hardware, Software, and Infoware”. In: DiBona et al. (1999), pp. 189–196.
- RAGGETT, DAVE; HORS, ARNAUD LE AND JACBOS, IAN (1997): “HTML 4.0 Specification”. Technical Specification, World Wide Web Consortium. Available online: <http://www.w3.org/TR/REC-html40/>.
- RATH, HANS HOLGER (1999): “Mozart oder Kugel – Mit Topic Maps intelligente Informationsnetze aufbauen”. *iX, Magazin für professionelle Informationstechnik* (12): pp. 149–155.
- RAYMOND, ERIC S. (1999): *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Peking, Cambridge, Farnham etc.: O’Reilly & Associates.
- STALLMAN, RICHARD M. (1999): “The GNU Operating System and the Free Software Movement”. In: DiBona et al. (1999), pp. 53–70.
- VIXIE, PAUL (1999): “Software Engineering”. In: DiBona et al. (1999), pp. 91–100.
- WEIBEL, S.; KUNZE, J.; LAGOZE, C. AND WOLF, M. (1999): “Dublin Core Metadata for Resource Discovery”. Network Working Group, Request for Comments (RFC) 2413. Available online: <http://info.internet.isi.edu/ls/in-notes/rfc/files>.