CONTINGENT CONFORMANCE TO STANDARDS[1]

David T. Barnard

Department of Computing and Information Science
Queen's University
Kingston, Ontario Canada K7L 3N6
613-545-6050

Abstract

At Queen's University we have produced an implementation of the CCITT X.400 Message Handling System recommendations for production use on our main academic computer system, and a restricted subset implementation of the ISO Standard Generalized Markup Language as a platform for computer science and literary research. On the basis of these experiences it is argued that standards efforts should be carried out in closer collaboration with basic computer science research and should allow explicit subsetting of requirements, since both of these would lead to reduced cost and risk and thus more rapid and widespread acceptance of standards.

1. Introduction

Two separate efforts at Queen's University have made heavy use of international standards. The first is a major project undertaken by Computing and Communications Services to implement an X.400 Message Handling System. The second is a research effort jointly undertaken by faculty members in Computing and Information Science, and English Language and Literature. The goals of these efforts have been markedly different, and thus the attitudes to the standards used have also been very different. The following two sections describe the individual projects, and the final section of the paper compares their use of and attitudes toward standards.

2. CCITT X.400 Message Handling System

As is the case at many universities, at Queen's we make use of several computer networks for electronic mail and other services. Local interworking of these facilities was becoming a problem for users, so Computing and Communications Services decided to develop an implementation of the X.400 recommendations [CCITT84(5)]. This would serve as a common interface for all of the mail services in use, it would be under local control, and could take advantage of various transport systems available to it–using both ISO conforming systems (such as X.25) and vendor systems (such as IBM's RSCS).

A software product has been built that meets these goals. QK-MHS/VM is an X.400 system that runs under IBM's VM operating system. It implements the basic service elements and some of the optional features of the X.400 recommendations. Part of the development was supported by a cooperative project with IBM Canada, Ltd. The initial implementation used the RSCS transport system, but the product has been extended so that it now can operate in a fully ISO-conforming environment [Hooper87(10)].[1]

QK-MHS/VM is constructed with a separate User Agent and Message Transfer Agent, but both components run on the VM host. As an extension of the work, a prototype User Agent was implemented on a personal computer in such a way that it can communicate with the Message Transfer Agent running on the host [Kairi and Barnard87(13)].

QK-MHS/VM is in regular use at Queen's and copies of versions of the program have been sent to some other sites. It has successfully exchanged messages with other X.400 implementations, and is now acting as a permanent connection between CDNnet (which uses the X.400 and X.25 standards) and NetNorth/Bitnet/EARN.

3. ISO Standard Generalized Markup Language

The Mnemosyne Project is a long-term collaboration[1] aimed at developing improved tools for humanistic research using computers [Higgins and Savoy85(9)]. One of the projects undertaken has been an electronic book [Logan et al.87(15)].

As part of our collaborative activity, we have seen the need for establishing an archive of texts as a set of test data for some of the algorithms and software we intend to develop, and as a primary source that will be of interest to our humanistic colleagues. After searching for a standard to use as a basis for building such an archive, we settled on the Standard Generalized Markup Language (SGML) [ISO86(12)].[1] SGML provides a framework for defining document types and then encoding particular documents. We have defined several document types that are appropriate for literary research, together with over 200 tags for marking elements in documents [Fraser86(8); Barnard et al.88(4)].

While we do not at present have software that can directly process full SGML, we do have IBM's Script system and its more limited GML facilities [IBM85(11)]. These are heavily used at Queen's on the main academic computer, as well as with input prepared on other machines. We have used this system, together with a Unix system and the programs *awk* and *sed*, to work with parts of a large textbook comprising hundreds of files marked using GML [Barnard and Skillicorn88(3)].

To fully exploit the power of SGML, we need software that can parse and validate document type definitions and document instances. We are at present building a parser using traditional programming language implementation techniques [Karababa87(14)]. We do not intend to handle all of the features that are specified in the Standard–not even all of those that are required of "conforming" implementations. Our software will thus allow us to work within the Standard, but not to exploit all of its flexibility. It is certainly possible to build fully-conforming implementations, and we hope to install the XGML software on our system [Exoterica87(7)].

Other efforts in document encoding also use only some of the facilities that are available in the SGML standard. The Association of American Publishers has developed a set of tags for marking documents for publication (principally typesetting) [AAP86(1)]. The University of Chicago Press has issued guidelines for the preparation of electronic manuscripts by authors [Chicago87(6)]. Their guidelines include a list of tags that should be used in encoding documents; while these tags conform to SGML, and a simple document type definition can be inferred from the tag set, the document type definition is not given explicitly.

The Association for Computers and the Humanities (ACH) has recognized the need for some standardization of text-encoding among researchers, so that exchange and consistent use of documents can be facilitated. This has resulted in the formation of a Working Committee on Text Encoding Practices.[1] The initial effort of the Committee has been to propose "a five-phase project to develop and promote guidelines for standard text-encoding practices in preparing machine-readable texts for scholarly research." The proposal document states: "We propose to make a concerted effort to make our text-encoding guidelines, where possible, compatible with appropriate existing schemes, of which SGML and AAP appear at present to be the most promising." The document type definitions and the tag set developed at Queen's are one of the major inputs to the process that has been proposed to accomplish these goals.

A major concern at the initial meeting sponsored by the ACH was the need to maintain multiple views of a document in its encoded form. We have proposed several approaches to this problem, not all of which are fully within the SGML formal framework [Barnard and Karababa88 (2)].

4. Comparison of Uses

The X.400 and SGML efforts can be compared with respect to several factors.

(1) *Motivation and use:* The X.400 project was to produce software to be used as a primary system at Queen's, and for release to other institutions. The product had to interwork, in real time, with software developed by other organizations, complying with the X.400 recommendations. In short, it had to be a real production tool.

The use of SGML gave us portability of our document archive and of the application software we will write to use it, so that we can make these available to other researchers. We also want to take advantage of software, such as typesetting systems, produced to conform to the SGML standard. We are not, though, expecting to import document type definitions or document instances that exploit all the features of SGML. In short, SGML is a platform on which we will conduct other research.

(2) *Responsible unit:* The X.400 project was carried out by a service group, the Department of Computing and Communications Services, using professional management and programming staff; only a small non-critical piece of it was undertaken by a graduate student.

The SGML work has been carried out in the context of a research project involving faculty members and graduate students.

(3) *Conformance to the standards:* The X.400 system had to be a conforming one, to the extent that that is possible, since it must accept whatever is sent to it by other–possibly fully-conforming–implementations.

The SGML use could be restricted to the features of the standard that we considered necessary for our own work. It is important that we be able to exploit other software, and thus we must remain properly within the standard; but it is not so important that we accept other uses of the standard, and so we could use a subset of it.

(4) *Complexity:* Both the X.400 recommendations and the SGML standard are complex. Building fully-conforming software systems is an expensive task. In both cases, there are only a few such systems in the world at present. The Queen's X.400 system aspires to conformance, but our SGML does not and will not.

While the experience reported here is limited, it does encompass extensive exposure to two very different standards and application environments. The following remarks flow from this experience.

(1) Early use of standards is a high-cost, high-risk activity, but it has potentially high payoff. Building software to conform to new standards often involves solving unanticipated problems. As with many such problems in computing, the initial solutions are expensive to produce, and the anticipated costs may be unrealistically low compared to actual costs (in development and execution time). Of course, early developers may seek a competitive advantage from their investment of effort, but a risk remains. Early implementations are also essential for ensuring the usefulness and self-consistency of the standards.

(2) Since standards are often very complex, subsets of features should be defined that could be implemented together for various levels of conformance. The SGML standard does this to some extent, but even what it considers to be required is very complex.

(3) There should be a good computer science foundation for standards work. While there can be a tremendous intellectual distance between academic computer scientists and professional development organizations, this distance must be bridged. Many of the problems being addressed by standards activities can benefit from basic research that has been done–there may be algorithms available for dealing with (parts of) the problem, there may be data structures that can be successfully applied, there may be analogous applications in other domains, and so on–and it appears that many of us involved in the standardization process are unaware of these applicable bits of knowledge.

One of the things that happens with any technology is that expert users adjust their expectations based on what they know (sometimes implicitly) to be possible and cost-effective. Some standards work does not seem to be based on a sufficient understanding of the underlying technical tools to appreciate what is easy and cost-effective. A closer collaboration with basic computer science research could influence standards designers to specify features that might be as useful as some that have been produced, but considerably easier to implement. Of course, such collaboration does not guarantee practicality - as some past language design effects demonstrate. Perhaps a new standard should be considered incomplete until a prototype implementation exists (which should also foster prototyping tools). The explicit provision of substantial subsetting capabilities, where appropriate and possible, is also a desirable aspect of standards work. Both of these would reduce the cost and risk associated with early use of standards, and lead to their more rapid acceptance.

5. References

(1) [AAP86] *Reference Manual on Electronic Manuscript Preparation and Markup*; Electronic Manuscript Series, Association of American Publishers (May 1986).

(2) [Barnard and Karababa88] D.T. Barnard and M.K. Karababa; Maintaining Multiple Document Structures Using SGML; Technical Report 88-204, Department of Computing and Information Science, Queen's University (1988).

(3) [Barnard and Skillicorn88] D.T. Barnard and D.B. Skillicorn; Machine-Assisted Textbook Publishing; to appear Electronic Composition and Imaging.

(4) [Barnard et al.88] D.T. Barnard, C.A. Fraser and G.M. Logan; Generalized Markup for Literary Texts; Literary and Linguistic Computing (to appear 1988).

(5) [CCITT84] CCITT, Study Group VIII; Recommendation X.400, Message Handling Systems: System Model-Service Elements (October 1984).

(6) [Chicago87] *Chicago Guide to Preparing Electronic Manuscripts*; University of Chicago Press (1987).

(7) [Exoterica87] Software Exoterica Corporation, XGML Version 3.0 Product Specification: An SGML System Conforming to International Standard ISO 8879 Standard Generalized Markup

Language (July 1987).

(8) [Fraser86] C.A. Fraser; An Encoding Standard for Literary Documents; M.Sc. Thesis, Department of Computing and Information Science, Queen's University (1986).

(9) [Higgins and Savoy85] L. Higgins and E. Savoy; Thanks for the Memory; interview with D.T. Barnard, R.G. Crawford and G.M. Logan in *Perspectives: Profiles of Research at Queen's University* (December 1985).

(10) [Hooper87] A.S. Hooper; X.400 as a Common Denominator for Electronic Messaging; Symposium on Computer Conferencing and Allied Technologies, University of Guelph, Guelph, Canada (June 1987).

(11) [IBM85] Document Composition Facility: Generalized Markup Language: Starter Set Reference; SH20-9187-3, International Business Machines Corporation (1985).

(12) [ISO86] ISO 8879, Information processing – Text and office systems – Standard Generalized Markup Language (SGML) (1986).

(13) [Kairi and Barnard87] K. Kairi and D.T. Barnard; Design and Implementation of an X.400 Stand-Alone User Agent; Computer Standards and Interfaces (to appear 1987).

(14) [Karababa87] M.K. Karababa; Processing the Standard Generalized Markup Language; M.Sc. Thesis, Department of Computing and Information Science, Queen's University (1987).

(15) [Logan et al.87] G.M. Logan, D.T. Barnard, and R.G. Crawford; A Computer-Based Edition of More's *Utopia*; in Randall Jones (ed.), *Computers and the Humanities*; Selected Papers from 7th International Conference on Computers and the Humanities, June 1985, Paradigm Press (forthcoming 1988).

---

[1]This work was supported in part by a cooperative agreement between Queen's University and IBM Canada Ltd., and in part by the Natural Sciences and Engineering Research Council of Canada under Grants A3014 and A0385.

[2]During the development of this software the author was involved only in a management role, serving as Director of Computing and Communications Services from 1982 to 1987. The system was designed by Bob Cavanagh and Andy Hooper, and implemented by Andy Hooper and Don Coleman.

[3]The other principal researchers in the Project are George M. Logan, Head of the Department of English Language and Literature, and Robert G. Crawford, Associate Dean of the Faculty of Arts and Science, who is also a member of the Department of Computing and Information Science.

[4]The author is currently involved in the official Canadian standards activities in this area as a member of the Canadian Advisory Committee on Text and Office Systems CAC/TC97/SC18 (Working Group on Text Description and Processing Languages).

[5]The author is a member of this Working Committee.