

SURVEY OF XML DEVELOPMENT TOOLS AND PROCESSES

Jussi Hanhijärvi
Student: 22476 w
e-mail: jussi.hanhijarvi@yle.fi

CONTENTS:

1.0 Summary.....	3
2.0 Introduction.....	4
3.0 Selection criteria based on development process	10
4.0 Integrated development environments IDE.....	14
5.0 Style sheet development tools.....	22
6.0 Text editors.....	24
7.0 DTD editors and generators.....	25
8.0 API environments	26
9.0 Word processors	28
10.0 Conversation tools	29
11.0 Delivery and publishing.....	31
12.0 Concluding remarks	36

Appendix A: Some application initiatives

References

1.0 SUMMARY

This report was written for HUT seminar "Digitaalisen Median Tutkimusseminaari" held at autumn 2000 under leading of Professor Petri Vuorimaa. Major goal in this work was to achieve a working knowledge of "state of the art" of XML development and related support tools.

In chapter 2 (introduction) will major concept of XML be discussed briefly in order to lay foundations on future considerations. Focus will be development tools oriented. Also some major published surveys will be examined here too. Notably works from Kuikka and Nikunen [3, 4], Pepper [5], Cover [6] and Garshol [14] will be briefly explained.

In chapter 3 we discuss topics that should be important for selection particular tools for specific project or company documentation and publication profile. Major emphasis has been laid on lifecycle of development and production process. There exist many models to describe SW process (for example waterfall, spiral, V-model etc), some of these can be applied directly to XML production process too. Without losing generality a widely adopted V model will be interpreted here. Also model elements for XML production process will be interpreted.

In chapter 4 will some integrated development environment (IDE) be explored. With IDE we mean such systems that can be used to combine different audio, video, textual and graphical elements to larger entity as a part of electronic publication system. Typically these system can have hundreds of documents and these are delivered on different mediums for example printed or presented through internet as www pages or can form core of an e-commerce system.

In chapter 5 will style sheet tools be considered. Special emphasis is laid on interactive XSL development and adaptive as well as corrective maintenance of XML site. Style maintenance is here important topic when we want to change style of whole site centrally. As there will be given an separate speech on XSL will this study be very brief and light.

In chapter 6 we discuss briefly some text editors that have SGML/XML extension.

In chapter 7 some more common DTD editors and generator will be discussed.

In chapter 8 will some API tools be considered. These tools are those that are primary dedicated to SW development and have some XML extensions too.

Most of the major word processor vendors have plans to build or have already some XML extensions. Of the many tools will main focus been laid on Microsoft initiatives and open-office (org) XML projects. Chapter 9

In chapter 10 will some conversation tools be considered. These tools are aimed to a) conversion from SGML, b) from non-structured specific documents and c) from outputs of some word processing tools to XML format.

In chapter 11 will delivery and publishing tools be discussed. Focus is in paper and web publishing other media will be discussed slightly.

In this study are parsers not unravelled as there exist separate speech (in this seminar) on this topic.

Last chapter “concluding remarks” we try to express some thought of found XML trends and usefulness of XML document creation with unravelled tools. This chapter is more or less a personal view on today’s technology..

Unfortunately, as commercial and no-commercial tools exist today in hundreds of different types, format and products we did not have enough time to test these “hands-on”. Moreover only very small fraction of all tools that exist on the “market” could be examined. Selection criteria was merely ad hoc than systematic exploring on the “market”.

2.0 INTRODUCTION

In this survey we concentrate on tools that can be used to create XML documents and related topics. XML is said to describe a category of documents, which:

- are stored in computer,
- are in machine manipulated form,
- partially describes instructions for programs that manipulate these documents,
- Contains clearly defined structure.

XML was originally aimed to allow structured documents (typically encoded in SGML) to deliver over internet as integral part of world wide web pages. One reason for this was that HTML had no means to encode specific structured element types (such as parts number and other identifier, prices and other calculable entities, search links to databases etc.) [13]. More accurately: development of various XML standard had the following goals [12, 13, 22]:

1. *It shall be straightforward to use XML over the Internet* XML documents should be easily and straightforwardly viewed and browsed. This goal will be fulfilled in near future when XML browser are as common as HTML browsers
2. *XML shall support wide variety of applications.* Although XML primary goal is to deliver documents over the web its not meant to restrict XML there. Also xml documents should be applicable in other areas too.
3. *XML shall be compatible with SGML* As a consequence XML is upward compatible with SGML.
4. *It shall be easy to write programs that process XML documents* Designers of XML standard had in their mind idea that competent programmer could write program that can process XML in few weeks¹.
5. *The number of optional features in XML is to be kept to an absolute minimum, ideally zero.* Firstly XML standards should be as complete as possible. Secondly additional features inevitably raises compatibility problems between different applications and users.
6. *XML documents should be human-legible and reasonably clear.* That is contents of XML documents should be readable and understandable even without specific browser also with normal text editor.

¹ However in practice it turns out that an average computer specialist use at least few weeks merely on work out how to receive and decode ISO 10646 [13].

7. *The XML design should be prepared quickly.* With XML (most) documentation problems should be solved right now.
8. *The design of XML shall be formal and concise.* This is in many ways consequence of rule 4. XML standard are based on Extended Backus-Naur notation and thus are based on solid well proven ground.
9. *XML Document shall be easy to create.* In the course of time there will exist sophisticated XML editors but before this era there should still be possible to create XML documents with one's favourite text editor.
10. *Terseness in XML mark-up is of minimal importance.* Some SGML features are designed to minimise the typing effort. These features are not supported in XML (in order to keep parsers simple).

In every days speech when one talks about XML he usually refers to whole group of "standards" and recommendations. These can be grouped to essential core standards, associated recommendations, supporting standards and application guides.

2.0.1 Core standards

XML 1.0 specifies what "tags" and "attributes" essentially are, but around it there exist still growing number of essential modules that specifies additional tasks [22]. Today W3 consortium has published either (almost final) draft or final version concerning following recommendations:

- **Xlink** which define standard way to add hyperlinks to an XML document. That is it defines elements that can be inserted in XML document in order to create and describe links between (networked) resources [23].
- **Xpath** is commonly refereed as a whole of recommendations XML Base, XML pointer and XML linking [23].
- **XML Base** defines syntax for providing same functionality as HTML BASE tag do. XML base guideline is closely related to Xlink and Xpointer [23]
- **Xpointer & Xfragments** define syntaxes to point parts of XML file. That is it defines syntax and semantics of language elements, that can be used as identifier to any URI references. These references locate resources of different Internet media types [23].
- **XSL** is language to express style sheets² i.e. specifies presentation of a class of those XML documents that use formatting vocabulary. XSL is based on a) XSLT style sheet transforming language and on b) vocabulary for defining formatting semantics [25].
- **DOM**, document object model, specifies function calls for manipulating XML (and HTML) files from a programming or scripting language [29].
- **XML Namespace** defines how [URL:s](#) can be associated with single tags and attributes within an XML document.
- **XML Schema** helps developer to precisely define their own XML formats.

² In many cases CSS style sheet language from HTML world is applicable to XML too.

2.0.2 Associated recommendations

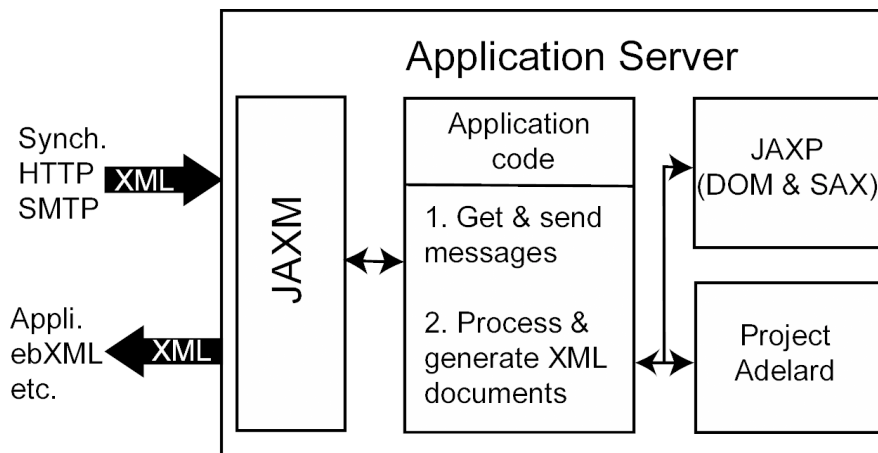
To XML associated recommendations are those that are not essential part, but useful in development of XML files. W3C have under consideration the following:

- **XML Query** guidelines is aimed to extract data from real and virtual documents on the web and thus provide interaction between web and database worlds. That is, XML files should accessed like database records.
- **RDF**, Resource Description Framework, describes integration of “variety of web-based metadata activities including, sitemaps, content rating, stream channel definitions, search engine data collection (web crawling), digital library collections and distributed authoring using XML”, as stated in [24]
- **SMIL** Synchronised multimedia is aimed to authoring of simple multimedia presentation, these presentation scan be composed of streaming audio, video, images, text and/or other media types [26].
- **SVG** is language to describe 2 dimensional vector graphics whose elementary elements consist of three types of graphics objects: vector shapes, images and text. Also SVG “drawings” can be interactive and dynamic (with aid of scripting) [27].
- **XML Signature**, gives guidelines for a) representing signature of web resources, b) portions of protocol messages and c) procedures to verify such signatures [28].

2.0.3 Supporting standards

There exist numerous official or “de facto” guidelines that are useful in development of XML document. Not to speak of various pictorial or aural presentation formats we mention:

- **ECMA Script** and its cousin **Java script** for manipulation of XML documents in browsers.
- **DOM** Document Object Model, “is platform and language-neutral interface that allow programs and scripts to dynamically access and update the content structure and style of documents” [29].
- **SOAP** (Simple Object Access Protocol) “is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses” [64]
- **ISO 106 46** character encoding
- **MathML** is description for expressing mathematical notations and capturing its structure and notations in XML document.
- **JAXP** (Java API for XML parsing), **JAXM** (Java API for XML messaging) and Adelard project are widely accepted software components. Position of these components in interactive XML session can be best described with the following picture



2.0.4 Application guides

Today there exist hundreds of application oriented guidelines and recommendations for XML documents, prepared by academic and non-academic institutions. Serious XML document designer should take these into account as they are jointly agreed and has been subject of expert critic. Application guides typically consist of document type definitions (dtd:s) and applicable vocabularies (tag definitions).

2.1 PREVIOUS SURVEYS

In this chapter have some previous work been revived. Also works from Kuikka & Nikunen [3, 4], Pepper [5,6], Cover [7] and Garshol [14] will be discussed.

It is important to notice although majorities of published tools (up to this day) are aimed to SGML development, there exist some commercial and public domain conversion tools to translate SGML to XML. Thus many SGML tools can be used to development of XML. As a rule, in use of these conversion tools warns the user of such properties that can't be converted.

2.1.1 Kuikka & Nikunen

One of the first published surveys on software for structured documents was done by Kuikka and Nikunen as early as 1994³ [3]. This survey was revised and expanded in 1998 [4]. The later reflects situation at the end of year 1997. In the first paper was 89 and in the later 207 such "packages" described, that claim to support structured documents and are SGML/XML compliant. Description, contact information, references, prices, operating system platform, 'type' and SGML support are listed for each software package. Also they had classified packages according alphabetical (name) order, in 18 'type' category and by price. Their 'type' categories can be seen on the following table. However, as the authors noticed most of the programs can not be classified according one single type property. Thus in the following table some items are counted more than once. From the 1994 list totally 24 programs was removed for the reasons of marketing, company production policies or from other reasons. These figures express the rapid growth of SGML/XML markets, that still continues its expansion.

³ Looked from today's viewpoint will the 1994 survey give some historical perspective. It contains highly readable tutorial on SGML too.

Property	items in 1994 survey	items in 1998 survey
Text editor	3	4
Structure editor	25	30
Desktop publishing	8	10
Formatter/Page layout program	3	13
Text search programs ⁴	7	19
Electronic delivery tool	6	16
Browser	0	25
Relational database	1	1
Text database	2	2
Structured text database	7	15
Document database	4	9
Database front end	0	4
Conversion program	33	61
Parser	10	28
API tool	15	35
DTD tool	11	23
DSSL tool	0	5
Other	0	1

From the focal point of this paper can some criticism laid on Kuikka & Nikunen selection criteria:

1. Their main emphases are on textual (and symbolic) documents. But documents contain often drawings, pictures and even photographs. Some authors include in the idea of publication audio and video artefacts too.
2. It's unclear what are the essential differences between "Desktop Publishing" and "Formatter/Page layout program".
3. They give no hints on which phase of development and production lifecycle of SGML process certain tool should be used. For example for the reader it is difficult to estimate if some tools can be used for maintenance at large or as specification aid.

2.1.2 Whirlwind guide and related work

Steve Pepper's widely approved *The Whirlwind Guide to SGML & XML Tools and Vendors* [5] is regularly and faithfully updated [from 1992 to 1997], online database⁵. The "Whirlwind guide" is originally based on his speech given at SGML Europe '94 in Montreux, *SGML TOOLS A brief introduction* [6]. This database (like the above refereed database prepared by Kuikka and Nikunen) is an extremely valuable on-line resource for SGML/XML user.

⁴ With text search program have the authors in mind such text preprocessing program that improves search process.

⁵ available at <http://www.infotek.no/sgmltool/guide.htm>

For the aim of his papers and database he states [6]:

"SGML is an enabling technology: it doesn't actually do anything in and of itself. In order to make it work for you, you need software tools: tools to help you design your application, tools to help you get your information into SGML format, and tools to help you do something useful with it once you've got it there. This presentation aims to give a brief overview of the kinds of SGML products currently available and some of the questions you should be asking yourself (and vendors) when choosing them."

The Guide is developed in several parts, with an introduction for each class of SGML software tool in the document overview. The Vendor and Tool directory is thus part of a larger document which explains the role played by different kinds of tools in an enterprise information management solution. The document sections in Whirlwind guide include:

- 1) Document abstract;
- 2) Introduction: Classifying SGML Tools (according to hardware and software platform, level of SGML support, and function or activity);
- 3) Hardware and Software Platforms;
- 4) Level of SGML Support: Feature support, Syntax support, Validation services offered;
- 5) Activities and Functionality (planning the application, capturing the data, managing the information, putting the information to work);
- 6) Directory of SGML Tools and Vendors. It supplies a listing of all (known) SGML tools, categorised according to functionality, along with names, addresses and telephone numbers of the vendors.

2.1.3 Survey of public domain software by Robin Cover

Robin Cover has also made a survey of public SGML/XML tools. Scope of his work, as he state in his "crisp" style, is: "Priority is given to "public" SGML/XML software in this document database since the scope of interest is mainly the Internet, where the ethic of public gift is highly esteemed. The wealth of SGML software made freely available for public use is evidence of that ethos". His survey is developed in 6 chapters:

- SGML Parsers, contains 8 p.c. tools
- SGML/HyTime Editing, Browsing, and Searching Tools, 23 p.c.
- SGML Data Conversion, Transformation, and Manipulation 14 p.c.
- SGML Formatting Tools 11 p.c.
- DSSSL Software Tools 12 p.c.
- XML/XSL/XLL Software Tools 33 p.c.

Some of the SQML tools that has direct XML extensions or can with relative small amount of effort be extended to XML has been counted in SGML groups.

Cover's database was last updated October 11, 2000. He gives short descriptions of the functionality and other properties of the tool. Gives hyper links to adherent [URL:s](#). Most of the

tools seem to be collected on the base that author or some user of the tool has announced its existence to oasis open organisation.

2.1.3 Garshol's list on "free" tools.

Lars Magnus Garshol maintains an online, regularly updated, database of public domain XML tools. The database includes classifications according products by 1) category, 2) name, 3) platform and 4) standard. Also he includes update listings and practical search engine in his site too. In standard class he have categorised tools according applicable XML standard like Xlink, Xpointer etc. Some application-oriented standards have been included in this class tool (such as MathML, EDIFACT). Platform class holds area that ranges from usual operating system base (like Unix, Windows, beOS etc) to running software environments (like Java, Perl, Python etc.)

Products by category listing are divided under following titles:

- *Parsers and Engines*, XML parsers and parsing toolkits, DOM implementations HyTime engines, DSSSL engines and various XML middlewares. 147 p.c.
- *Control Information development* which are tools to create, modify and document DTD:s, style sheets(XSL).30 p.c.
- *Document Storage and management*, support tools for document management (such as document oriented database engines) and search engines. 18 p.c.
- *Conversion*, includes tools to convert from other formats to XML, automated processing of XML documents as well publishing converters 23 p.c.
- *Electronic delivery*, here has been listed XML browsers and web publishing tools 19 p.c.
- *Editing and Composition*, tools for interactive creation and composition of XML documents. 19 p.c.

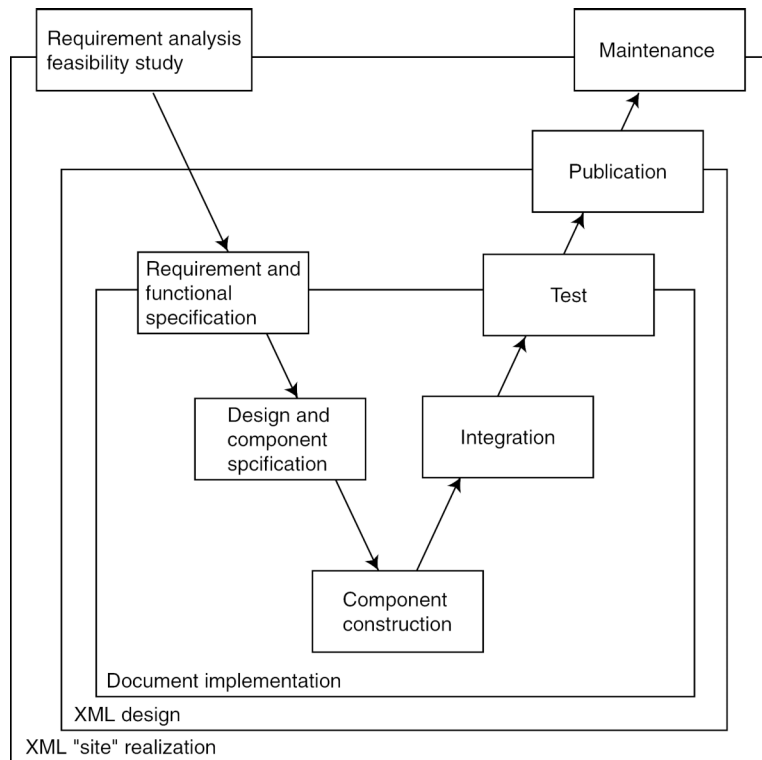
Garshol's lists are relatively dense. Product names (with hyperlinks), vendor, platform, link in Web and short description is given. His list contains totally 256 entirely public domain products, which gives a sign of the large interest in XML in public domain too.

Note: Maturity and coverage of certain tool is difficult to estimate from Garshols list as some tool are effort of large programming activity and some are purely academic exercise in some short term course.

3.0 SELECTION CRITERIA BASED ON DEVELOPMENT PROCES

There has been published very few articles on the development process of XML. However XML documentation should be planned, specified and designed under strict disciplines (as all engineering work should be). For example error in requirement analysis can have far reaching consequences on publication and investment policy of some company.

In software development there has been long tradition to model software processes i.e. that engineering work whose goal is to produce well functioning, reliable and applicable software to peculiar application. Such models as waterfall, spiral, v-grove etc. models are widely used for modelling this work. Although some of these models are more precise we restrict here to v-grove model in order to describe different factors that should be taken into account in tool selection process.



3.1 REQUIREMENT ANALYSIS AND FEASIBILITY STUDY

When one wants to realise a documentation system his first task should be analysis of the real requirements, feasibility of ideas and intended goals. In this phase relevant questions are for example:

- What is the real application? Carefully should application domain standards be analysed in order to define their applicability to the intended tasks.
- For what purpose system is aimed to? Carefully should be analysed the planned media, its restrictions, capability, expected load, benefits in publishing the same documents in several media. Typical examples of different media are IP or WAP networks, CD- or DVD ROM:s, DAB, DTV, paper format etc. Various conversions is important topic here too.
- To whom it's intended? Regardless of transport medium one should consider internet, extranet or intranet as delivery medium. For example some parts of the same document can be freely public whereas some part (of the same document) can be restricted to specific users only.
- What is the supporting technology? Such topics as software platform, net topology, database structures etc. should be unravelled.

3.2 REQUIREMENT AND FUNCTIONAL SPECIFICATION

In this phase one specifies exactly how the documentation system should be implemented. Input for this phase is output of previous requirement analysis. Also in this phase we usually

know roughly the documents type and contents. They can contain audio, video, text, speech images etc. Our task in this phase is to develop requirements for data capture and document forms, as well functions how documents are manipulated. Errors in this phase are fatal as output (implementation specification) dictates future implementations.

In this phase should at least the following topics be considered:

- Content analysis based on technical constraints
- Transmission media.
- Appropriate document structure
- All usable styles based on intended publishing media (paper, networked, cd-format etc)
- Service architecture that is believed to fulfil functional requirements (servlets, JSP, JAXM, JXP or other).
- Project organisation, estimated budget and workload.

3.3 DESIGN SPECIFICATION

Output of this phase should be detailed technical specification of implemented system. Here should for example the following topics be solved:

- Appropriate database structure (SQL tables, structured databases, etc.)
- Qualitative requirements for different document parts (text, audio video, images etc.)
- Tools to achieve required quality of document parts, for example different tools have different influence on workload and service architecture.
- Definition of test requirements and acceptance test.

3.4 COMPONENT CONSTRUCTION

In this phase will detailed implementation be performed according outputs of previous phases. Task activities includes data capturing, data transformation and data storage as well software construction too.

Typical work in data capturing are:

- Direct authoring
- Recording from various sources (like audio and video)
- Keyboarding from manuscript.
- Generating from another structured or unstructured source

In data transformation main task is to:

- translating from unstructured electronic source to structured XML format
- translating from another structured format to other structured format
- importing unstructured parts as a vital component of document.

In data storage edited, captured and transformed documents are stored usually in electronic form, for example in database (SQL, native XML, OO database) or simply as files. Here application oriented DTD's and vocabularies will be used in layout design. Style sheets will be implemented according XSL definitions. SMIL multimedia requirements are enforced etc.

In software design in this phase can be divided in different categories for example:

- 1) software that are used for data capturing and transformation as a pre storage activity.

- 2) Software that manipulate documents prior publication or delivery.
- 3) Software that implements user interactions within documents for example e-commerce business logic.

Majority of design tools is dedicated for the component construction.

3.5 INTEGRATION

In this phase will all the constructed components be integrated to an entirety. It's quite common that results of integration will lead further refinement of component constructions. For example parsing XML document before storing in database will reveal that service architecture is not appropriate or efficient.

3.6 TEST AND VALIDATION

In this model is assumed that in component construction and in integration phase sufficient 'debug – correction' cycle has been performed. Activities in this validation phase occur after integration. Typical activities are:

- Validation of XML documents as "fit for purpose" activity.
- Acceptance testing
- Worst case load testing.
- Conformance testing

3.8 PUBLICATION AND DELIVERY

Publication and delivery occurs only after designer, owner and customer are satisfied on the design. Here will all documents published in intended media using the planned architecture. Delivery of XML document can be accomplished directly with XML document itself or as combination of transformation process and a storage system that has no knowledge of document structure and style.

There exist many important factors that should be taken into account when choosing presentation platform:

- user interaction with document system as a functional activity
- screen typography and layouts, although there exist some common factors between different media (for example between paper and monitor) there exist still many separating factors too.
- information accessing, content can be accessed in linear fashion (like in paper format), hierarchically (navigating via structure), through hypertext links, via graphical elements etc.

3.9 MAINTENANCE

Maintenance activities can be grouped in three "class":

- Corrective maintenance, where observed faults and failures will be eliminated. For example in interactive XML based e-commerce system there can exist such requests that have not been thought before and can lead to system failure.

- Adaptive maintenance, where system will be brought to new state according changed requirements. Typical problem here is that, number of published document can increase to level where system cannot any more serve final user or selected document creation tools does not support anymore in intuitive way the increased amount of documents.
- Preventive maintenance, where with aid of new design system is brought to new faultless state when probable changes in operation environment occur. For example system load can grow to unacceptable level, thus either DTDs and style sheets should be simpler or processing power should be higher.

4.0 INTEGRATED DEVELOPMENT ENVIRONMENTS

4.1 IBM ALPHAWORKS AND RELATED TOOLS

IBM maintains large collection of XML related tools. These could be grouped in various chapter in this survey. But for convenience of reader we have collected them here. Most of the described tools are available (with associated licensing policy) at IBM alphaWorks web site at <http://www.alphaworks.ibm.com>. Some of the tools have public domain licenses and some are commercial. According IBM's announce there exist trend to integrate most of the tools to one design environment.

The IBM alphaWorks development environment try provide most of the essential tools to create Web services. IBM claims that their products are aimed to:

- “Discover - Browse the UDDI⁶ Business Registry to locate existing Web services for integration. The Web becomes an extension of the development environment.
- Create/Transform - Use powerful XML editing functions to quickly develop new Web services. Complete transformation (edit and mapping) tools are also provided so that developers can create Web services from existing XML, Java, or SQL applications.
- Build - Wrap existing bean components as SOAP⁷-accessible services and describe them in the Web services description language (WSDL). Generate SOAP proxies to Web services described in WSDL. Generate bean skeletons from WSDL. Minimal knowledge of SOAP or WSDL is required.
- Deploy - Deploy the Web service on the developer's machine or to a remote, production-level server for testing right away. After testing, publish the Web service immediately to the application server (WebSphere Application Server or Apache Tomcat).
- Test - Test applications as they run locally or remotely, and get instant feedback.

⁶ UDDI apparently IBMs own abbreviation, UDDI= Universal Discovery Description & Integration.

⁷ SOAP = Simple Object Access Protocol

- Publish - In addition to creating and deploying Web services, the development environment can also publish them to the UDDI Business Registry. This advertises your Web services so that other businesses can access them”.

IBM alphaWork has published the following 42 tools for XML in order to full fill the above mentioned features:

Bean Markup Language	Data Descriptors by Example
DataCraft	DirectDom Developer Kit
Dynamic XML for Java	Extensible Types
LotusXSL	P3P parser
PatML	RDF for XML
SOAP for Java	SVGView
TaskGuide Viewer	TeXML
Tspaces	VoiceXML
Voice Server SDK	Web Services Toolkit
VoiceXML for DirectTalk	XDRtoXSD
X-IT	XMI Toolkit
Xeena	XML BeanMaker
XML and Web Services DE	XML Diff and Merge Tool
XML Bridge for SAP	XML Enabler
XML EditorMaker	XML Generator
XML for C++	XML Lightweight Extractor
XML Interface for RPG	XML Parser for Java
XML Master	XML Security Suite
XML Productivity Kit for Java	XML TreeDiff
XML Translator Generator	Xplorer
XML Viewer	XSLbyDemo

Descriptions for these tools are directly borrowed from alphaWorks web site.

Bean Markup Language

“Bean Markup Language (BML) is an XML-based component configuration or wiring language customized for the JavaBean component model”.

Data Descriptors by Example (DdbE)

“DdbE accepts well formed XML documents as input and constructs a DTD”.

DataCraft

“Provides an XML view of databases and enables publishing XML forms to the Web.”

DirectDom Developer Kit

“The DirectDOM™ technology allows a Java developer to build rich graphical user interfaces by manipulating documents using the Document Object Model (W3C DOM) of a browser.”

Dynamic XML for Java

“The IBM Dynamic XML for Java (DXMLJ) Processor provides a facility to add dynamism to static XML documents through annotations. It allows one to introduce namespace-based annotations that can specify how a particular subtree in the XML document is to be interpreted or filled in. Also DXMLJ seamlessly blends XML and Java.”

Extensible Types

“Extensible Types (eTypes) is a Java component library that enables users to specify constraints and determine whether objects satisfy these particular constraints.”

LotusXSL

“LotusXSL implements an XSL processor in Java that can be used from the command line, in an applet or a servlet, or as a module in other program. By default, it uses the XML4J XML parser, but it can interface to any XML parser that conforms to the DOM level 2 or SAX level 1 specification.”

P3P parser

“P3P parser is a P3P⁸ protocol parser and constructor written in 100% pure Java. The package (com.ibm.p3p) contains classes and methods for parsing, generating, manipulating and evaluating P3P proposals and responses. It also contains a parser and evaluator for “A P3P Preference Exchange Language” (APPEL).”

PatML

“PatML is a pattern/match replacement tool for XML documents. It is a rule based specification language in which the user can specify how an XML document can be transformed for the sake of browsing (converted to HTML) or other purposes.”

RDF for XML

“A Java implementation of the RDF specification for creating technologies that search for data and describe, categorize, rate, and otherwise manipulate the data.” RDF for XML has retired from alphaWorks development environment.”

SOAP for Java

⁸ Platform for Privacy Preferences (P3P) is a protocol that enables the private exchange of personal information on the web. According W3C : “The goal of P3P is to enable Web sites to express their privacy practices and enable users to exercise preferences over those practices. P3P products will allow users to be informed of site practices (in both machine and human readable formats), to delegate decisions to their computer when appropriate, and allow users to tailor their relationship to specific sites”

“IBM-SOAP is a Java reference implementation of the SOAP (Simple Object Access Protocol) v1.1 specification. IBM has contributed to the open Apache SOAP project.”

SVGView

“SVGView uses Java 2D and XML Parser for Java to parse, process, and render SVG files locally or over the Internet. The viewer passes the document to the parser, which creates the data tree structure. The parser then traverses the tree in Java 2D, which calls the appropriate functions in the Java2 API. For example, if a square needs to be drawn, the relevant Java2D function draws the square at the appropriate location.”

TaskGuide Viewer

“TaskGuide Viewer is an XML-based tool for creating wizards. This wizard-creation tool makes computer tasks easier by breaking complicated tasks into sequential, simple steps that can be performed using a graphical, user-friendly interface.”

TeXML

“The path to print begins with your XML document. You write an XSL transform which accepts your document type and outputs a new XML document which conforms to the TeXML document type. The java program, TeXMLatte transforms any document conforming to the TeXML document type into TeX. You may now use your TeX processor to produce typeset output from XML.”

Tspaces

“Tspaces is a set of network communication buffers called tuple spaces and a set of APIs (and classes that implement the API) for accessing those buffers. Tspaces allows heterogeneous, Java-enabled devices to exchange data with little programming effort. The package includes server software that implements the buffers and client software for accessing the buffers. Tspaces provides group communication services, database services, URL-based file transfer services, and event notification services.”

Voice Server SDK

“The IBM Voice Server SDK Beta is a toolkit based on VoiceXML Version 1.0. This package allows you to create speech-enabled applications and test them on a desktop workstation before deploying them in a telephony environment.”

VoiceXML

“VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed-initiative conversations. The goal is to provide voice access and interactive voice response (e.g. by telephone, PDA, or desktop) to web-based content and applications.”

VoiceXML for DirectTalk

“VoiceXML for DirectTalk is a prototype implementation of a VoiceXML rendering application that runs on the IBM DirectTalk family of voice response products.”

Web Services Toolkit

“Web services are self-describing, self-contained, modular applications that can be mixed and matched with other Web services to create innovative products, processes, and value chains. Web services are Internet applications that fulfill a specific task or a set of tasks that work with many other web services in an interoperable manner to carry out their part of a complex work flow or a business transaction.”

X-IT

“X-It is a Java based application for batch processing of XML files. X-It takes a batch of XML files and processes/ transforms them based on the operation specified.

X-It supports the following operations:

1. Adding a PI/Comment to the XML files
2. Deleting specific nodes from XML file
3. Finding a given text and replacing with a new value
4. Validating the XML file against the specified DTD
5. Sorting the XML file “

XDRtoXSD

“XDRtoXSD is a program which takes as input an XML Schema written in the XDR schema language (used by Microsoft Internet Explorer and BizTalk) and translates them into an equivalent schema expressed in the W3C XML Schema language, the emerging standard for representing XML vocabularies, grammars and constraints.”

Xeena editor

“The editor takes as input a given DTD, and automatically builds a palette containing the elements defined in the DTD. Users can thus create/edit/expand any document derived from that DTD, by using a visual tree-directed paradigm. The visual paradigm requires a minimum learning curve as only valid constructs/elements are presented to the user in a context-sensitive palette.”

XMI Toolkit

“The Toolkit is a Java component that converts UML information between Rational Rose Models and XMI-standard XML files. The Toolkit can also generate DTDs directly from your models. A Reference Implementation of XMI, with source code, is included. The functionality includes the ability to generate Java from Rational Rose and UML models and to convert Java to Rational Rose and UML models. This refresh includes an updated framework API that allows you to read and write XMI 1.1 files. Additionally, a sample application using adapters to serialize and deserialize Java objects using XMI is provided.”

XML and Web Services DE

“This tool allows HTML, Java, SQL and XML developers to quickly extend existing e-business applications so that they can deliver business informational Web services. Database developers can also use SQL as a programming language to quickly build data-aware Web services. Web developers can create Web services with minimal knowledge of Java, XML or SOAP.”

XML BeanMaker

“The XML BeanMaker is a software tool written in Java to generate Java beans out of XML Schema files (DTD files) automatically. The goal of the BeanMaker is to alleviate the tedium of having to write get, set and notification methods on Java classes corresponding to XML structures. It generates this code automatically.”

XML Bridge for SAP

“The XML Bridge for SAP is designed to provide XML integration between SAP R/3 systems on the one side and arbitrary SAP R/3 or non-SAP systems on the other side. To do this two principal design decisions have been made while working out the design for XML Bridge for SAP:”

XML Diff and Merge Tool

“The Compare and Merge Tool for XML is a Java program that can be used for reconciling or understanding changes that a single user has made to his XML document OR for reconciling or understanding changes that several people have made to a single document.”

XML EditorMaker

“The XML EditorMaker is a software tool written in Java to generate Visual editors out of XML Schema files (DTD files) automatically. The goal of the EditorMaker is to alleviate the tedium of having to write editors for editing XML documents. It generates the editor automatically.”

XML Enabler

“The XML Enabler is a servlet that can successfully implement stylesheets such as the LotusXSL technology. Using the XML Enabler, developers with any kind of browser can now send requests to a servlet and as the servlet responds, it formats the data using different XSL stylesheets. The system administrator can then configure which stylesheets go with which browser types.”

XML for C++

“IBM's XML for C++ parser (XML4C) is based on Apache's Xerces-C XML parser, which is a validating XML parser written in a portable subset of C++. XML4C integrates the Xerces-C parser with IBM's International Components for Unicode (ICU) and extends the number of encodings supported to over 150. It consists of three shared libraries (2 code and 1 data) which provide classes for parsing, generating, manipulating, and validating XML

documents. XML4C is faithful to the XML 1.0 Recommendation and associated standards (DOM 1.0, SAX 1.0, DOM 2.0, SAX 2.0 etc). Source code, samples and API documentation are provided with the parser.”

XML Generator

“Creating test cases for XML applications by hand from DTDs can be tedious and may not cover all possible or required instances. IBM's XML Generator is a Java program designed to automate this process by generating "random" instances of valid XML from a single input DTD. The XML Generator engine can create an XML file or can be accessed via the Document Object Model (DOM) API.”

XML Interface for RPG

“XML Interface for RPG, C and COBOL provides access to DOM and SAX level 1 APIs in RPG, C and COBOL programs. It allows procedural applications to create new or parse existing XML documents, facilitating the use of XML as both a datastore and IO mechanism in these languages.”

XML Lightweight Extractor

“Given a DTD, the XML Access Server Lightweight Extractor (XLE) allows a user to annotate the DTD to associate its various components with underlying data sources, and when requested, extracts data from the data sources and assembles the data into XML documents conforming to that DTD. “

XML Master

“XML Master (XMas) consists of two parts:

- 1) The XMas application - an editor for designing and generating custom Java beans to work with XML documents that conform to a certain DTD (Document Type Definition).
- 2) The XMas Bean Suite - a collection of Java beans that can be used for modeling XML structures and getting access to their parts via XML-oriented GUI components.”

XML Parser for Java

“XML Parser for Java is a validating XML parser written in 100% pure Java. The package (com.ibm.xml.parser) contains classes and methods for parsing, generating, manipulating, and validating XML documents. XML Parser for Java is believed to be the most robust XML processor currently available and conforms most closely to the XML 1.0 Recommendation.”

XML Productivity Kit for Java (XPK4J)

“The early access release of XPK4J contains two components. The first component resides in the directory 'xpk4j' and provides an infrastructure for wiring together Java beans that

process XML data. The second component resides in the directory 'xbeans' and provides a tool that can create the source code for JavaBeans from XML data files. The generated beans can then be used to read, manipulate, and write XML data.”

XML Security Suite

“In this release of XML Security Suite, we⁹ provide reference implementations of DOMHASH, a proposed canonicalized digest value for XML document, and its two sample applications. DOMHASH can be a basis for XML digital signature that is being discussed in both IETF and W3C.”

XML Translator Generator

“XML Translator Generator (XTransGen) enables you to convert XML documents and data based on one DTD to another without writing XSL scripts or any program code. It can also be used to filter data from HTML documents into XML documents. This process works in two phases: (1)Setup: first a common example in the two DTDs is used to generate a translator once. (2)Use: this translator is then used to translate XML documents.”

XML TreeDiff

“XMLTreeDiff is a set of Java beans designed to perform fast differentiation and update of DOM structures. XMLTreeDiff works in many ways like diff and patch. However, rather than differentiating the file representations of the documents (that is, the XML files), XMLTreeDiff runs directly on the DOM's themselves. This way, the differences are directly expressed in terms of native tree operations like change node, delete node or insert node, rather than line mismatches. The advantages of this approach are several: it avoids the need to convert the DOM trees to file format prior to comparing them; with that, it eliminates the 'false negative' reports caused by dissimilar file representations of equivalent DOM structures; finally it avoids the need to infer the tree structural meaning of a line difference report.”

XML Viewer

“XML Viewer for Java is a Java application that displays any well-formed XML document. You can traverse the document's hierarchy using the tree views, find out the attributes of a particular node in the hierarchy and view the source of the XML and associated DTD files. You can also view the source of a selected node in the XML source view and the definition of the node in the DTD source view.”

Xplorer

“Xplorer allows the user's to do the following things:

⁹ also IBM

- 1) Search for XML files based on the XML file name, Document type and some advanced search options like element name and value, attribute name and value, PI, etc.
- 2) Check the validity of XML files.
- 3) View a valid XML file in the XML viewer. Only valid files can be viewed in the XML viewer. If you try to view an invalid XML file, an error dialog appears and errors are logged in the Log Window. “

XSLbyDemo

“XSLbyDemo is a technology for generating XSLT rules on the basis of editing operations conducted under the WYSIWYG mode of Page Designer, which is a full-fledged HTML authoring tool provided with IBM WebSphere Studio.”

4.2 XMETAL 2.0

Xmetal is editor from Softquad, company that previously published widely adopted HotMetalPro HTML editor [31]. It supports HTML, XML and SGML document formats, also tag definitions and DTD:s creation process. It has an database import wizard which helps to contact to any ODBC (SQL) database. Database wizard can be used to create database import and content conversion to XML.

Document editing capabilities contains among other things:

- CSS for display formatting and typical word processor like style editing features.
- Plain text editing mode, for tag and attribute insertion and edit.
- Drag and drop for text, images [URL:s](#) etc.
- Features common in many text processing tools: macros, templates, spell checking etc
- Support for Unicode, remote DTD's and internal subsets
- Preview with integrated browser also support external browsers too
- Contains a validator for error reporting
- Etc.

5.0 STYLE SHEET TOOLS

5.1 XSL LINT

XSL Lint is basically an Perl script that checks errors in XSLT style sheet. It's written by Norman Walsh and is said to detect the following anomalies [39]:

- An incorrect XSLT URI.
- Failure to provide a version.
- Duplicate match patterns.
- Modes that are used but never defined.
- Modes that are defined but never used
- Named templates that are used but never defined

- Named templates that are defined but never used
- Templates that use name= where match= was probably intended
- xsl:call-template elements that contain anything other than xsl:with-param
- Variable/parameter references that are not defined at the point of use.
- fo: elements that aren't part of the XSL 1.0 PR.
- fo: element properties that aren't part of the XSL 1.0 PR.

XSL lint is freely available at: <http://nwalsh.com/xsl/xslint/>

5.2 XSLIDE

xslide is an package for Emacs (20.3.1) major mode for editing XSLT style sheet. It has both syntax coloring, automatic completion and convenience functions to run XSLT engines [35]. Features of xslide revision 0.1.2 is claimed to include [40]:

- xsl-process function that runs XT and collects the output
- Font lock highlighting so that the important information stands out;
- xsl-complete function for inserting element and attribute names;
- xsl-insert-tag function for inserting matching start- and end-tags;
- Automatic completion of end-tags;
- Automatic indenting of elements;
- Comprehensive abbreviations table to further ease typing.

Xslide is freely available at: <http://www.mulberrytech.com/xsl/xslide/>

5.3 JADE

Jade (James' DSSSL Engine) is James Clarks famous DSSSL engine which has been as an ultimate reference for other tools in this genre. It's an general SGML tool for conversion of SGML to other SGML DTDs or to output formats like RTF, XML flow object tree and TeX. Jade can process XML documents as input too [35].

Jade is excellently documented and is freely available at <http://www.jclark.com/jade/>

5.4 OPENJADE

OpenJade is a variant of Jade [36]. OpenJade is a project undertaken by the DSSSL community to maintain and extend Jade. OpenJade is freely available at <http://openjade.sourceforge.net/>

5.5 DSC

DSC (An online DSSSL syntax checker and implementation framework, including the DSSSL transformation language) written by Henry Thompson and is based on James Clarks SP SGML parser. It implements the DSSSL transformation language, but not the style language, and also contains a nearly complete Scheme (R4RS) implementation. DSC also provides clear error messages, making it useful as a debugging tool [37].

DSC is available at <http://www.cogsci.ed.ac.uk/~ht/dsc-blurb.html>

6.0 TEXT EDITORS

6.1 AMAYA

Amaya is the W3C browsing and authoring test environment. It contains a WYSIWYG style of interface (similar to that of the popular commercial browsers). Features of Amaya includes the following [43]:

- Lets user both browse and author web pages
- Maintains a consistent internal document model adhering to the DTD definitions.
- Is able to work on several documents at a time. Multiple XHTML, MathML and SVG can be edited at a time.
- Helps to create hypertext links
- Includes a collaborative annotation application.
- Is extendable.

Amaya is available at <http://www.w3.org/Amaya>

6.2 IRIS XML EDITOR

IRIS XML editor edits documents according to DTDs generated by the IRIS DTD GENERATOR. According to the vendor this allows the editor to be heavily customized for guiding the user, with wizards and other helpful info.

The tool is available at <http://www.cabinfo.com/download.htm> by filling the form at CEI site.

6.3 VISUAL XML

Visual XML is written by Pierre Morel in Java with SWING foundation classes. It allows users to edit DTD and XML documents [41].

It is available at <http://www.pierlou.com/visxml>

6.4 XED

XED (An XML document instance editor) written by Henry Thompson It's a relatively simple XML editor written in C, Python and Tk. It tries to read the DTD in order to be able to suggest valid elements to be inserted at any point in the document during editing session. XED is claimed to support the following features [42]:

- Single-window WYSIWYG presentation
- Detailed error messages with line and character position information in case of well-formedness errors on input
- Add, remove and rename balanced start/end tag pairs and empty elements
- Add, remove and rename attribute name/value pairs
- Add or remove comments, CDATA sections and processing instructions

- Context-sensitive tag and attribute menus
- 'Smart' diagnosis of mixed versus element-only content
- Filling of text content, indenting of element-only content
- Structure-sensitive point-and-sweep selection paradigm
- Structure-preserving cut and paste
- Multiple undo
- Single-key movement into, out of and around elements
- Key bindings based on xxxPad under WIN32; based on Emacs under Unix
- Emacs modules

XED is available at <http://www.ltg.ed.ac.uk/~ht/xed.html>

6.4 MERLOT

Merlot is an open source Java based XML editor for data-oriented applications rather than document-oriented applications. It runs on any Java 2 based virtual machine (also JDK 1.2.2 or JDK 1.3) It has a extensible application interface that allows the editing of some elements to be customized for individual DTDs.

Merlot is available at <http://www.merlotxml.org>

7.0 DTD EDITORS AND GENERATORS

7.1 DTDDOC

dtddoc is written by Lars Magnus Garshol. It is written in Python and can be used to generate HTML and DocBook RefEntry documentation for XML DTDs. It reads the DTD and an XML documentation file (using SAX) and generates the output documentation of these.

It's freely available at <http://www.garshol.priv.no/download/software/dtddoc/>.

7.2 LIVEDTD

LiveDTD, written by Robert Stayton in Perl, parses XML DTDs and generates HTML files from the DTDs with cross-links to element and parameter entity [34]. Thus documents can be viewed with common HTML browser. Live DTD's capabilities are as best when DTD are complex (like in DocBook) as it has clear user interface.

It is freely available at <http://www.sagehill.net/livedtd/>

7.3 EZDTD

ezDTD is freely available DTD editor written by Duncan Chen . It stores DTDs in its own internal metadata format, however it can import external DTDs. It can export DTDs into HTML format and save them as either in SGML or XML format [50].

It is available at :<http://www.geocities.com/SiliconValley/Haven/2638/ezDTD.htm>

7.4 FIRSTSTEP EXML

FirstStep EXML tool integrates EXPRESS language to XML. In the home site of Product Data Integration Technologies (PDIT) is stated that [49]:

“As a data encoding language, XML is a very flexible and "Internet-friendly", but it lacks many of the semantic features that data management professionals expect and applications require to ensure data semantics and integrity [1]. Therefore, the PDI project adopted EXPRESS (ISO 10303-11) as the data specification language that governs the semantics of data exchanged with an XML encoding. This necessitated a mapping from EXPRESS to XML DTD syntax. FirstSTEP EXML is a small software tool developed by PDIT to convert an EXPRESS schema into an equivalent XML DTD”.

FirstSTEP EXML is available at <http://www.pdit.com/pdml/exmlintro.html>

8.0 API ENVIRONMENTS AND XML MIDDLEWARE

8.1 SAX

SAX (the Simple Api for Xml) is a simple event-based API for XML parsers. It is not an official standard, since the participants of the xml-dev mailing list developed it instead of standards body members. However, SAX is very much a de facto standard since it is supported by most XML parsers and is used by lots of applications.

SAX is free for both commercial and non commercial users and is available at <http://www.megginson.com/SAX>. It is regularly versioned and updated, the current version is dated at Friday 5 May 2000.

8.3 JDOM

JDOM is an API for representing the XML document tree structure similar to the DOM, but much simpler and designed specifically for Java using the collections API. JDOM structures can be built from XML files, DOM trees and SAX events and can be converted to the same [48].

It is available at <http://www.jdom.org>

8.4 PYTHON XML PACKAGE

The Python XML package is a collection of various XML tools written in Python language. This collection has been put together by the Python XML Special Interest Group (The Python XML-Sig), a group of volunteers led by Andrew M. Kuchling. This collection contains following parts:

- Saxlib, Python translation of SAX 1.0
- Pyexpat a James Clark's expat C parser module wrapped up as a Python module. That is by compiling Pexpat to Python interpreter one can have fast C parser interface to Python
- 4DOM which also support both DOM 1.0 and DOM 2.0 model
- amlarch.py an architectural forms engine
- module for serializing and deserializing Python objects to XML

Package is available at <http://www.python.org/sigs/xml-sig/status.html>

8.5 XML UTILITIES

Collection of various XML utilities written in Java by David Brownell. Collection includes 3 packages:

- SAX2 XML utilities, including →fred enhanced parser for strong XML (i.e. Sun parser) and DOM tree parser. Also it has adapters for HTML and XML parsers from Sun and Oracle.
- DOM2 implementation
- SAX/XML Conformance Test Harness, is a package for SAX and XML conformance tests. In tests can be used various databases like OASIS/NIST XML Conformance Test Suite.

Author has performed some conformance test and published result in his site. Also he have written a good tutorial on conformance tests [53].

Package is available at <http://home.pacbell.net/david-b/xml/>

8.6 XMLBLASTER

Essentially xmlBlaster is publish/subscribe middleware for client/server XML applications, also it implements so called MOM (Message Oriented Middleware). It's written in Java by collaborative xmlBlaster team. Communication to the server is based on Common Object Request Broker, CORBA (using JacORB) or Java RMI, also other protocols are supported too (like XML-RPC or email). Subscriber can use XPath expressions to filter the messages they wish to receive. xmlBlaster use Sun's XML parser.

Package is available (free under LGPL – licence) at <http://www.xmlBlaster.org/>.

8.7 S4/TEXT

S4/Text is a development library of API objects for the famous Microsoft Word processor. It enables to build tailored tools to create, access and manipulate structured documents based on Word platform [32]. S4/text is based on Word's Object Model and support Microsoft's DCOM object passing protocol.

S4/Text seems to be a valuable tool in environments where need to store centrally and generation of more or less fixed format documents is relatively large. For example many governmental institutions may find it useful.

9.0 WORD PROCESSOR AND OFFICE TOOLS

9.1 OPEN OFFICE ORGANISATION (STAROFFICE)

Open Office is an public and open project that provides typical office tools like word processor, spreadsheet, presentation graphics. They have on XML file format sub project too, led by Michael Brauer. As mission statement for XML subproject they state:

“Our mission is to create an open and ubiquitous XML-based file format for office documents and to provide an open reference implementation for this format.

Core Requirements (these items are absolutely required)

- 1) The file format must be capable of being used as an office program's native file format. The format must be "non-lossy" and must support (at least) the full capability of a StarOffice/OpenOffice document. The format is likely to be used for document interchange but that use alone is not enough.
- 2) Structured content should make use of XML's structuring capabilities and be represented in terms of XML elements and attributes.
- 3) The file format must be fully documented and have no "secret" features.
- 4) OpenOffice must be the reference implementation for this file format.

Core Goals (these items are highly desired)

- 1) The file format should be developed in such a way that it will be accepted by the community and can be placed under community control for future development and format evolution.
- 2) The file formats should be suitable for all office types: text processing, spreadsheet, presentation, drawing, charting, and math.
- 3) The file formats should reuse portions of each other as much as possible (so for example a spreadsheet table definition can work also as a text processing table definition).”

9.2 STILO WEBWRITER

Stilo Web Writer is easy to use and intuitive XML authoring tool. It's key features is claimed to include [38]:

- Generates DTD 'on the fly'
- Contains graphic DTD view
- Contex-sensitive editing

- Simple keystroke navigation element creation
- Assured to be conformant with XML
-

Documents can be constructed in different ways:

- Edit document with DTD
- Create document 'from scratch'
- Edit well formed document (one without DTD)
- Use an existing DTD as the basis of dokument
- Contains conversion from XML to HTML
- Generates mathematical expressions in MathML and OpenMath

Stilo WebWriter is available at <http://www.stilo.com/webwritertechnologyoverview.htm>

10.0 CONVERSATION TOOLS

10.1 DB2XML

DB2XML (written by Volker Turau in Java) is a tool for transforming data from relational databases into XML documents. It is written in 100% pure Java.

DB2XML is claimed to provide three main functions [47]:

- Transforming the results of database queries or complete databases into XML documents or into HTML documents using XSLT stylesheets.
- Providing attributes describing characteristics of the data (i.e. meta data).
- Easy integration of XSLT stylesheet processors
-

DB2XML can be used as a standalone tool, as a servlet API (to generate dynamic server side XML documents) or as a database to XML API. It requires a JDBC driver (or a ODBC driver using the JDBC-ODBC bridge). It has been tested with several different SQL databases and drivers including Oracle, Microsoft Access and SQL-Server, InstantDB and MySQL.

10.2 JEDI

Jedi stands for Java Extraction and Dissemination of Information. It's claimed to be [46]:

“JEDI is an extensible, fault tolerant parser component for the extraction of semi-structured data from textual sources, especially from the world wide web. It uses simple, grammar-based syntactical source descriptions and an associated lightweight data model to provide for sophisticated and flexible rewrite and restructuring facilities which grant seamless, integrated access to various heterogeneous sources”.

It is written by Peter Frankhauser and Gerald Huck at German National Research Center for Information Technology (GMD - Forschungszentrum Informationstechnik GmbH) and is available at <http://www.darmstadt.gmd.de/oasys/projects/jedi/index.html>

10.3 MAJIX

Is a tool to convert RTF documents to XML form. It's written in Java by TetraSix company.

It's downloadable at <http://www.tetrasix.com>

10.4 TIDY

Tidy is small package to detect errors in XML and HTML markups and to some extent to fix them. Also with Tidy one can clean up bad XML and HTML and in many instance can convert from poor HTML to XML.

Tidy is freely available at <http://www.w3.org/People/Raggett/tidy>

10.5 ACE

Ace is strongly typed programming language for processing SGML and XML tools. Ace is part of commercial Structured Information Manager product range (SIM). Essentially SIM includes a high performance SGML/XML server for multi gigabyte database and a web server. Although SIM is commercial product is Ace freely downloadable for non-commercial applications at <http://www.simdb.com/simdb%20content%2FAbout%20SIM%2Fdownloads>.

10.6 METAMORPHOSIS

MetaMorphosis is SGML/XML tree transformation engine that can be used to convert between SGML/XML applications, to publishing formats etc. MetMorphosis is joint development of Ovidus and Robert Bosh GmbH. Ovidus has applied MetaMorphosis on two other product:

- MetaHelp, a system for generating HTML Help and Windows Help documents from SGML or XML documents.
- MetaPage (info only in German), a style sheet based batch formatting for high volume formatting.

MetaMorphosis is claimed to be the standard SGML/XML toolkit in the German automotive industry for all kinds of SGML/XML conversions. It has the following features (according producer of the tool):

- DTD information in MetaMorphosis available; arbitrary DTDs can be loaded and used for processing
- Plug-in architecture; currently available plug-ins:
 - MMDtd, provides DTD information and local parsing/evaluation
 - MMObdc, access to ODBC databases
 - MMPinsler, draws tree diagrams in various formats
 - MMSystem, full access to file system features and functions and other system calls
- powerful SDK:

- DOM-compatible object API
- C++ and COM binding
- Windows, Linux and Solaris

For Linux MetaMorphosis is free and for MS windows (95/98/NT) there exist 30 day demo versions at http://www.ovidius.com/metamorphosis/index_e.html.

10.7 XTAL

XTAL is general Java Package written by Oliver Zeigerman. XTALs architecture consist of front end for reading data in and back end to produce outputs. Today there exist only one XM front end and two back end to produce XML or TeX¹⁰. More will be probably be developed.

XTAL is based on ANTLR ¹¹ (and Java) description language.

XTAL is available at <http://www.zeigerman.de/xtal.html>

10.8 X-TRACT

X-tract is said to be “grep-like tool for XML documents”. That is, it is a loosely to XQL query language based document search tool. Also X-tract is command line tool that returns all those subtrees from XML or HTML documents which match a query pattern.

X-tract was written by Malcolm Wallace. Sources and binaries are freely available at <http://www.cs.york.ac.uk/fp/Xtract>.

11 DELIVERY AND PUBLISHING

11.1 JXRTF

JXRTF provides means to generate RTF output from XML document with aid of XSLT. Its is written in Java but uses C++/C libraries through JINI interface. It's freely available at <http://oss.jway.lu/jxrtf.html>.

11.2 XRS

XRS is an XML-aware search engine, which can perform structure-based queries on XML documents. The search engine has been written by Dongwook Shin in C or C++, but is also accessible in Java through a JNI interface. The author has developed some new techniques for indexing and retrieving structured documents efficiently. Thus improving search times. In XRS home site is available a demo of client side search applet communicating with servlets that uses the server side XRS engine.

XRS is available at <http://dlb2.nlm.nih.gov/~dwshin/xrs.html>

¹⁰ It can sound confusing to translate XML to XML But with such translation can be used to translate one XML instance of one DTD to completely different one.

¹¹ ANTLR Another Tool for Language Recognition is a computer language tool that provides a framework to construct compilers, translators etc. from grammatical descriptions.

11.3 PASSIVE T_EX

Passive T_EX is a tool written by Sebastian Rahtz at Oxford University Computing Services. It's a T_EX implementation of XSL (Working Draft March 2000) and MathML. Tool reads XML documents (also containing XSL formatting objects and MathML elements) and uses LaTeX to produce formatted output.

As key features of Passive T_EX is stated:

- PassiveTeX is a library of TeX macros which can be used to process an XML document which results from an XSL transformation to formatting objects.
- PassiveTeX provides a rapid development environment for experimenting with XSL FO, using a reliable pre-existing formatter
- Running PassiveTeX with the pdfTeX variant of TeX generates high-quality PDF files in a single operation.
- PassiveTeX shows how TeX can remain the formatter of choice for XML, while hiding the details of its operation from the user.

Passive T_EX is available at <http://users.ox.ac.uk/~rahtz/passivetex/>

11.4 MOD XSLT

mod.xslt is simple Apache web server module to serve XML based content. It uses a configuration file that maps requested document names to XSLT and XML source files. Also on URL request module opens sourcefile and determine its DOCTYPE, Based on DOCTYPE associated XSL file will be opened. To the requesting browser returned document type is text/html. Translation occurs transparently to the user.

Module is available at <http://modxslt.userworld.com>.

11.5 PHP

PHP is a HTML-embeded server side scripting language that have XML extensions too. Its syntax is borrowed from C, Java and Perl with some unique PHP add-ons. The primary goal of PHP is to allow for web developers to write dynamically generated pages quickly.

PHP is available at <http://www.php.net>

11.6 PXSLSERVLET

PXSLServlet is written by Paul Tchistopolskii. In principle it is a Java servlet that can be used to publish relational database items as HTML with aid of XSL style sheets. It is available at <http://www.pault.com/Pxsl/>

11.7 Rich Men's Document Management System

Rich Men's Document Management System abbreviated `rm -d ms`¹²

Cornerstones in the design of `rm -d ms` is claimed to be as stated in their site^[44]:

- Server side application logic is implemented by XSLT stylesheets.
 - XSLT is an appropriate programming paradigm for frameworks.
 - XSLT like languages with less limitations are on the way.
 - The stylesheet may call hard coded 'legacy' components.
- An application operates on one comprehensive persistent XML document.
 - You will be able to use an XML view on an ordinary relational database as a document.
 - You can use XPath syntax to combine XML from different sources.
 - The management system hides the storage model.
- Write access is logged, old versions are archived.
 - The user may take back any operation.
 - The units of work are not limited by database transactions.
- The System uses the transaction and the recovery facilities of relational database systems.
- The prototype is implemented in Java.

11.8 IBM TECHEXPLORER

IBM techexplorer is claimed to be: "a plug-in for Navigator and Internet Explorer and an ActiveX control for applications like Microsoft PowerPoint and Word. techexplore enables the display of TeX, LaTeX and MathML documents and the publishing of interactive scientific material on the Web. Version 3.0 includes full support for MathML 1.01, augmented display of LaTeX, new ways to enliven techexplorer documents via C++, Java, JavaScript, the DOM, and a web-based equation editor". It has a Mathematica connectivity extension too.

30 day Trial version is available at <http://www-4.ibm.com/software/network/techexplorer/>
See separate description on IBM alphaWork activities too (chapter 4 in this document).

11.9 SOFTAWRE AG'S TAMINO

Tamino is an acronym for Transactional Architecture for Managing Internet Objects. Tamino is a server architecture that is based on XML for data storage, exchange and retrieval. Data can be almost any type of documents, including, audio, video, spreadsheets, HTML pages XML formatted information, images etc. Gartner Group¹³ has assessed Tamino [18]. As conclusion they state as Taminos strengths:

- "Uses Native XML for Database Management Tasks. Native XML means that Tamino stores XML data without modification and the database engine itself works with XML" They claims that with this feature it is the only native XML database architecture on the market.

¹² Apparently an Unix joke about Microsoft.

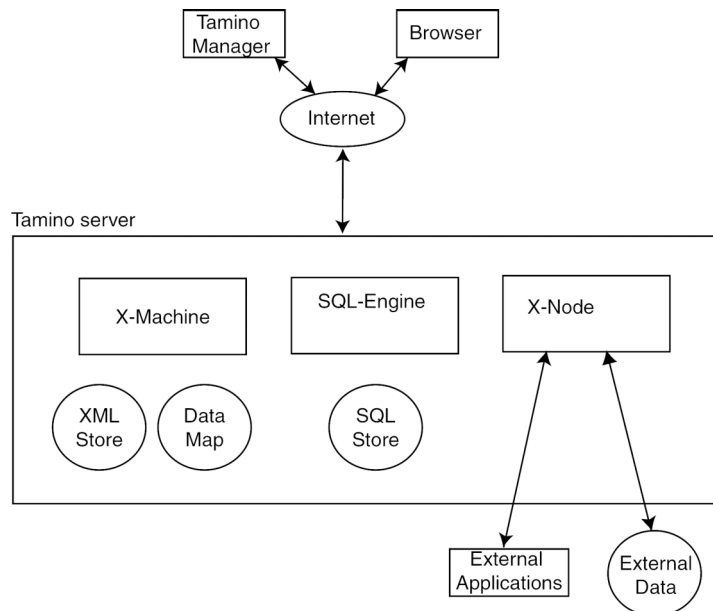
¹³ Gartner Group is an privately financed company that provides business technology research, consumer and market intelligence, consulting, conferences and decision-making tools. Their evaluation reports are widely referenced and are valuable source for different decision making processes.

- “Uses DOM for Customizing XML documents”
- “Vendor Has Longstanding Database Expertise”
- “SQL RDBMS Tools available”
- “Uses XQL for Queries and Indexes”

While conventional relational database (like SQL) is well suited for situations where data is relatively stable and can be ordered to tables and rows. This behaviour of data is not fully appropriate when data is “rich” like in multimedia (audio and video) files, software components (like Java beans), object oriented data structures or complex textual documents (like entire books). Most databases that is claimed to support XML rely heavily on relational database model and thus store data in rows and tables. Every time when XML document is retrieved from relational database will it be converted to XML, thus deteriorating performance. XML documents could be stored decomposing its elements along database tables making modifications to single document a difficult task. They can be stored as BLOBs too with penalty of poor search efficiency and large overhead in indexing when documents are added or changed.

Tamino uses XML as database paradigm. It uses XML schemas and DTDs as primary means of structuring, organizing, storing and describing data. Thus supporting full indexing and searching of data. Moreover user can modify structure of document without losing efficiency and there exist no need to write programs for XML – RDBMS mappings. Tamino goes even further as it can parse and structure any valid XML document and generate DTD automatically, thus updating to document related database while using relational databases structure of documents should be specified before it can be imported to database.

Tamino's general architecture is depicted in the following picture:

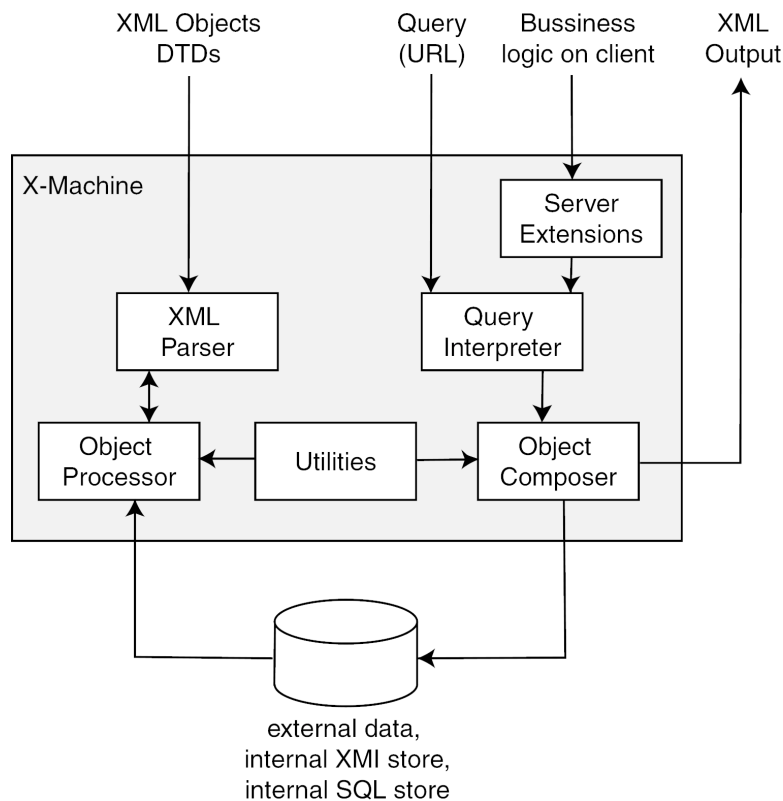


The X-machine and XML store provide processing and storing of XML documents. SQL engine and SQL store process SQL data. X-node provide interface to external applications and data

Data Map has central role in storing of schemas, DTDs, stylesheet, vocabularies as well generation and retrieval of data.

X-machine performs actual XML processing. Its architecture is depicted in the following picture:

The parser validates incoming and outgoing XML documents even if they do not contains DTDs. XQL compliant query interpreter provides information retrieval. Server extension can handle client logic to the server side (like CGI or servlets) or can contain embedded behaviour in XML object or document. The embedded behaviour can be written in C, C++ or Java code and it can contain references to COM/DCOM object brooker, thus facilitating connections to external applications and legacy systems (like existing databases most notably Adbas and Natural applications).



11.10 APACHE XML PROJECT

Around Apache web server there exist XML project too [64]. They state in their site (URL) for the goals of apache xml project:

- “to provide commercial-quality standards based XML solutions that are developed in an open and co-operative manner.
- to provide feedback to standards bodies (such as IETF and W3C) from an implementation perspective, and
- to be a focus for XML-related activities within Apache projects.”

Currently they have 6 sub projects. Namely

- **Xerces** (named after the Xerces Blue butterfly) provides fully-validating (i.e. error reporting) parsers. These implement the W3C XML and DOM (Level 1 and 2) standards, as well the de facto SAX (version 2) standard. Xerces parsers are available both in C++ and Java. Future implementations will include XML Schema support too.
- **Xalan** (named after a rare musical instrument) is the W3C XSLT and XPath implementation. They claim that: "The stylesheet processor is feature-rich and robust. The XPath Processor is useable as a stand-alone unit. Xalan uses the Bean Scripting Framework (BSF) to implement Java or script extensions, features multiple document output extensions." Future implementation of Xalan will include SQL/JDBC extensions.
- **Cocoon** is a framework for XML web publishing. It is claimed to bring: "a whole new world of abstraction and ease to consolidated web site creation and management based on the XML paradigm and related technologies."
- **FOP** is print formatter driven by XSL formatting objects. It is a Java 1.1 application that transform XSL object tree into PDF document. The formatting object tree, can be in the form of an formatted XML document or can be passed in memory as a DOM Document or (sometimes) SAX events.
- **Xang** is aimed to build data-driven, cross-platform Web applications that integrate disparate data sources. "The Xang architecture cleanly separates data, logic and presentation."
- **Apache Soap** project is an implementation of the draft W3C protocol SOAP . Originally it is based on, and supersedes, the IBM SOAP4J implementation.

12 CONCLUDING REMARKS

In this report we have unravelled XML development processes and some typical development tools. Although it seems at a first hand that amount of described tools are large are these only small fractions of these that exist on the market. Many important and essential tools are not included at all. Reason for this is that the author did not have enough time to explore them.

On the other hand reading this report one can deduce that activity in the XML area is very lively. It is hoped that XML could solve many problems that we have today in the web. These problems are for example:

- World wide web are more or less an large chaos, there exist over one billion unstructured home page.
- Search in the web (using different search engines) is often effortless and boring.
- There exist no clear and simple mechanism to store, produce and retrieve documents based on their structure.
- Multimedia data (like audio and video) cannot be indexed and searched based on their contents using pure HTML.

Large amounts of tools and development packages reflect definitely that there exists certain need to XML like approach. And more tools will be see their birth almost daily

Appendix B: References

Probably the most comprehensive on-line reference and excellent starting point in XML study is:

<http://www.oasis-open.org/cover>

Almost every aspect of SGML/XML is listed and annotated there. Also W3C's homepage is valuable reference point at:

<http://www.w3c.org>

In this study has the following documents been refereed (in order to keep this list as short as possible has in many cases only starting point of the URL (site) been mentioned):

- [1] **Bosak J., Bray T.:** *XML and the Second-Generation Web*, Scientific American, Feature Article, May 1999
- [2] **Freter T.:** *XML: Mastering Information on the web, Beyond Text and Graphics XML makes Web pages function like applications, XML: It's the Future of HTML, XML: Document and Information Management*, A four article series published by Sun (www.sun.com)
- [3] **Kuikka, E., Nikunen, E.:** *Rakenteisten tekstien käsittelyjärjestelmistä* Report A/1994/4. Kuopio, Finland: University of Kuopio, Department of Computer Science and Applied Mathematics, January 11, 1994. iii + 82 pp.
- [4] **Kuikka, E., Nikunen, E.:** *Survey of Software for Structured Text*. Report A-1998-1. Kuopio, Finland: University of Kuopio, Department of Computer Science and Applied Mathematics, 1998.
- [5] **Pepper, S.:** *The Whirlwind Guide to SGML & XML Tools and Vendors*. Regularly updated, online database, available at <http://www.infotek.no/sgmltool/guide.htm>
- [6] **Pepper, S.:** *SGML TOOLS A brief introduction*, SGML Europe '94, Montreux, also available at <http://www.infotek.no/sgmltool/94paper.htm>
- [7] **Cover R.:** *Public SGML/XML Software*, regularly updated, online database available at <http://oasis.oasis-open.org/cover/publicSW.html>
- [8] **Goosens M.:** *XML, a new start for the Web*, Lecture notes may 2000, available at
- [9] **McLaughlin B:** *Java and XML*, first edition June 2000, ix + 479 pp. O'Reilly & Associates
- [10] **Bosworth A.:** *Microsoft's Vision of XML*, SGML/XML 1998 Paris, also available at <http://www.oasis-open.org/cover/bosworthXML98.html>
- [11] **Bosak J.:** *Media Independent Publishing Four Myths about XML* IEEE Computer, Vol 31, No. 10 October 1998, pp. 120 - 122
- [12] **Walsh N.:** *A Technical Introduction to XML*, AborText Inc. 1997, 1998, World Wide Web Journal also available at <http://nwalsh.com/docs/articles/xml/>
- [13] **Anon.:** *The Extensible Markup Language (XML)* Ethos Technology Briefings Serie) available at <http://www.sgml.u-net.com/xml.htm>

- [14] **Garshol L.M.:** *Free XML tools and software*, available at <http://www.garshol.priv.no>
- [15] **Sommerville I.:** *Software Engineering* Third Edition, Addison Wesley 1989
- [16] **Rook P.(ed):** *Software Reliability Handbook*, Elsevier Applied Science 1990.
- [17] **Anon.:** *CeBit 2000; Software AG backs XML*, available at <http://www.softwareagusa.com>
- [18] **Anon.:** *Software AG, Inc. Tamino*, evaluation report 1580, Gartner Group
- [19] **Anon:** *XML Project*, OpenOffice organisation, available at <http://www.openoffice.org>
- [20] **Anon.:** *StarOffice XML FileFormat, Working Draft, Technical Reference Manual*, October 7, Sun Microsystems, Inc
- [21] **Anon.:** *Open Office FAQ*, available at <http://www.sun.com/developers/openoffice/fag.htm>
- [22] **Bos B.:** XML in 10 points (7.really), available at <http://www.w3.org/1999/XML-in-10-points>
- [23] **Conolly D.:** *W3C XML Pointer, XML Base and XML linking* April 2000, available at <http://www.w3c.org/XML/linking.html>
- [24] **Swick R., Miller R., Schloss B., Singer D., Brickley D.:** *Resource Description Framework*, October 2000, available at <http://www.w3c.org/RDF>
- [25] **Froumentin M.:** *Extensible Stylesheet Language (XSL)*, November 2000, available at <http://www.w3c.org/Style/XSL>
- [26] **Michel T.:** *Synchronised Multimedia*, November 2000, available at <http://www.w3c.org>
- [27] **Lilley C.:** *W3C Scalable Vector Graphics (SVG)*, November 2000, available at <http://www.w3c.org/Graphics/SVG/Overview.html>
- [28] **Eastlake, D.:** *XML-Signature WG*, November 2000, available at <http://www.w3c.org/> Signature
- [29] **LeHégaret P.:** *Document Object Model (DOM)*, November 2000, available at <http://www.w3c.org/DOM>
- [30] **Diaz A.:** *W3C's Math Home Page*, October 2000, Available at <http://www.w3.org/Math>
- [31] **Anon.:** *Xmetal 2.0*. available at <http://www.softquad.com>
- [32] **Anon:** *S4/TXT Architecture*, available at http://www.i4i.com/tech_s4text_arch.htm
- [33] **Anon.:** *IBM alpha Works*, available at <http://www.alphaworks.ibm.com>
- [34] **Stayton R.:** *LiveDTD*, available at <http://www.sagehill.net/livedtd>
- [35] **Clark J.:** *Jade – James' DSSSL Engine*, available at <http://www.jclark.com/jade>

- [36] Martin D.PH: OpenJade Source Control Repository Home Page, available at <http://openjade.sourceforge.net>
- [37] **Anon.:** *Announcing dsc-2.0: An online DSSSL syntax checker and implementation framework, including the DSSSL transformation language*, available at <http://www.cogsci.ac.uk/~ht/dsc-blurb.html>
- [38] **Anon.:** *Stilo WebWriter Technical Overview*, available <http://www.stilo.com/webwritechoverview.htm>
- [39] **Walsh N.:** *XSL Lint*, available at <http://nwalsh.com/xsl/xslint>
- [40] **Anon.:** *xslide – Emacs Major Mode for XSL Stylesheets*, at <http://www.mulberrytech.com/xsl/xslslide>
- [41] **Morel P.:** *Visual XML*, available at <http://www.pierlou.com/visxml/body.htm>
- [42] **Thompson H.S.:** *XED: An XML document instance editor*, available at <http://www.ltg.ed.ac.uk/~ht/xed.html>
- [43] **Vatton I.:** *Welcome to Amaya W3C's Editor/Browser*, available at <http://www.w3.org/Amaya>
- [44] **Anon.:** *The Rich Men's Document Management System*, at <http://rmdms.sourceforge.net/welcome.html>
- [45] **Tchistopolskii P.:** *PXSLServlet*, available at <http://www.pault.com/pxsl>
- [46] **Huck G., Frankhauser P.:** *Java Extraction and Dissemination of Information*, available at <http://www.darmstadt.gmd.de/oasys/projects/jedi/index.html>
- [47] **Turau V.:** *DB2XML 1.3 Transforming relational databases into XML documents*, available at <http://www.informatik.th-wiesbaden.de/~turau/DB2XML>
- [48] **Anon.:** *JDOM*, available at <http://www.jdom.org>
- [49] **PDIT Web Development Group:** *FirstSTEP EXML*, at <http://www.PDML.org/exmlintro.html>
- [50] **Anon.:** *ezDTD 1.5* available at <http://www.geocities.com/SiliconValley/Haven/2638/ezDTD.htm>
- [51] **Anon.:** *SIG for XML Processing in Python*, available at <http://www.python.org/sigs/xml-sig/>
- [52] **Brownell D.:** *XML and SAX2 Software*, at <http://home.pacbell.net/david-b/xml/>
- [53] Brownell D: *Conformance Testing for XML Processors*, at <http://www.xml.com/pub/a/1999/09/conformance/index.html>
- [54] **Anon.:** *What is xmlBlaster*, at <http://www.xmlBlaster.org/>
- [55] **Anon.:** *Majix*, available at <http://www.tetrasix.com>
- [56] **Ragett D:** *Clean up your Web pages with HTML TIDY*, <http://www.w3c.org/People/Raggett/tidy>
- [57] **Anon.:** *The Metamorphosis product family*, at http://www.ovidus.com/metamorphosis/index_e.html

- [58] **Anon.:** *Download Ace Scripting Language* at <http://www.simdb.com/simdb%20content%2Fabout%20SIM%2Fdownloads>
- [59] **Wallace M.:** *Xtract: a 'grep'-like tool for XML documents* at <http://www.cs.york.ac.uk/fp/Xtract>
- [60] **Anon.:** *Opensource Extensible XML Modelling Application* at <http://www.merlotxml.org/>
- [61] **Anon.:** *Apache Module for XSLT*, at <http://modxslt.userworld.com>
- [62] **Rahtz S.:** *PassiveTex*, at <http://users.ox.ac.uk/~rahtz/passivetex>
- [63] **Shin D.:** *XRS: XML Retrieval System*, at <http://dlb2.nlm.nih.gov/~dwshin/xrs.html>
- [64] **Anon.:** *Welcome to the Apache XML project* at <http://xml.apache.org>