Author(s): Draganova, Chrisina
Article title: XML in the computing Curriculum
Year of publication: 2006
Citation: Draganova, C. (2006), 'XML in the computing Curriculum', In: Proceedings of Computer Science Education Conference, Veliko Turnovo University, Bulgaria, pp 393 – 398.

XML in the Computing Curriculum

Chrisina Draganova

XML in the Computing Curriculum: *XML (Extensible Markup Language) provides a way of structuring and describing data in a simple text format understandable for human beings and in the same way easily interpretable by computer programs. It is widely used in industry and it forms a fundamental standard for a number of emerging technologies, including Web Services, RDF (Resource Description Framework), RSS (Really Simple Syndication), as well as derivatives languages such as WML (Wireless Markup Language) and XHTML (Extensible HTML). This paper describes the experience of teaching XML at our department and the approach of embedding XML topics in already existing courses. It also points out the underpinning theoretical concepts related to XML.*

Key words: *XML; Computing Curriculum.*

INTRODUCTION

XML is a markup language, a subset of the meta language Standard Generalized Markup Language (SGML) designed to describe data in a structured way. XML documents are text files which contain tags used to mark the structure of the data and the data itself. These documents are easily understood by people and manipulated by computer programs. XML provides a standard format for exchanging data between heterogeneous components and improves communications between different systems. XML is the main underlying technology in major developments related to Internet computing. Consequently the industry requires graduates to have knowledge and skills related to these technologies. However, finding the right balance between traditional "computer science" theory and current marketable technologies is a challenging task for many computer departments.

Tucker et al. [11] argues that by reducing the amount of material taught from theoretical computing, such as discrete mathematics, algorithm complexity, finite state automata, etc., and replacing them by techniques that reflect current technologies the students will not be prepared adequately to meet the challenges of future developments in the computing field. On the other hand the modern curriculum should be sensitive to technological changes as emphasized in [7]. This can not be done without either reducing the number of theoretical topics required as a core, or incorporating these topics throughout the curriculum.

Paxton [9] and Brookes [2] suggest approaches for making the computing theory relevant by relating it to popular technologies such as XML. Paxton [9] illustrates the concept of the abstract data type tree using XML and the concept of Backus-Naur Form (BNF) using DTD. In the approach suggested by Brookes [2] the computing theory is used to support the understanding of XML rather than the opposite. A number of papers such as [3], [4] and [5] propose practical ways and lecture materials for including leading edge vocational areas based on XML in the computing curriculum.

This paper presents the approach of how XML technologies are covered in the curriculum at the Department of Computing Communication Technology and Mathematics (CCTM) at London Metropolitan University and how the different topics map to respective theoretical concepts. By incorporating such material in our programmes we aim to equip students with the necessary skills and knowledge of today's technologies demanded by industry and at the same time to provide them with solid understanding of relevant fundamental theoretical concepts that will help them to cope with the future developments in the computing field. The next section gives an overview of the main XML technologies covered in our courses and it discusses issues related to teaching XML. The last section gives the conclusions.

TEACHING XML TECHNOLOGIES

Background

The CCTM department at London Metropolitan University offers a number of undergraduate programmes in computing, multimedia, mathematics and electronics. A full list of the offerings can be found on [6]. These programmes have modular structure and are organised on three levels: certificate, intermediate and honours equivalent to full time studies in the first, second and third years. Each level involves 8 modules of study including core and optional modules. The modules discussed in this paper are offered on the "BSc Computing" and "Foundation Degree (FD) of Computing" courses. The "BSc Computing" is a three year degree course that aims to equip graduates with a range of technical and personal skills necessary for the analysis, design, implementation and maintaining of software systems, to give exposure to the latest developments in computing applications and to provide graduates with intellectual and transferable skills to enable them to work and progress in different environments. The "Foundation Degree of Computing" is a two year vocation oriented degree which aims to develop work-specific skills, relevant to the computing industry and to enable graduates to make immediate contributions in the workplace on finishing the course.

One of the most important and rapidly developing areas covered in the curricula is the Net centric computing area. It includes a range of sub-specialties such as computer communication network concepts and protocols, multimedia systems, web standards and technologies, network security, wireless and mobile computing, and distributed systems [7]. We offer modules on each of the levels, certificate, intermediate and honours, covering these subjects to some extend. Our approach reinforces the understanding of fundamental theoretical concepts through the presentation of specific up to date technologies and tools, and provides a lot of practical activities. The certificate level modules discuss the main network protocols and architectures, the current web design technologies and standards, Internet applications and emerging technologies. The intermediate level modules cover more in depth communications and networks, client-server computing and web development. The honours level modules focus on web-application and distributed system architectures, network management and security. Each module has a one-hour lecture, one hour tutorial and two hours of lab sessions per week. The modules are typically delivered through a variety of lectures, tutorials and workshops. The lectures are used to present a framework for understanding the topics covered and to provide starting points for the students' own reading. Tutorials are used for further developing the theoretical content and practising the technical skills outlined during the lectures through exercises and research tasks. Workshops are used to introduce the students to specific development skills and to provide hands-on experience. Case studies related to real word examples are embedded in the practical activities enabling students to better understand the theoretical and technical content of the lectures. XML topics are incorporated in these modules and put in the context of the respective subjects.

Teaching XML topics in Net centric modules

This subsection describes the XML topics covered in the Net centric computing modules and discuses issues related to teaching this material. Full details of XML and its applications can be found for example in [1] and [12].

Introductory XML topics including XML syntax, DTD (Document Type Definition), XHTML (Extensible HTML), WML (Wireless Markup Language), XSL (Extensible Style Sheet Language) and XSLT (XSL Transformations) are taught in certificate level modules in the context of web design and development. **XML** is a universal language for data specification, independent of its internal computer representation used for separation of data and presentation, data sharing, transportation and storage, software configuration and defining a new markup language. **DTD** provides instructions about the structure of the data, defines the list of legal elements and their types. **XHTML** is a combination of XML and HTML, designed to provide a language for developing web pages that can be displayed correctly by all browsers. XHTML is a W3C

recommendation standard [12]. **WML** is a markup language based on XML and is designed to be used for rendering documents on a variety of mobile devices. **XSL** and **XSLT** describe how XML documents are displayed and transformed. The transformed documents are typically displayed in a Web browser, saved in a file or database or passed to another application for further processing.

XHTML and WML are presented as derivative languages of XML and exercises for validating XHTML and WML documents are used to demonstrate DTDs. XSL and XSLT are introduced through exercises related to transforming XML documents into different formats and emphasizing the importance of separating the data from the presentation. The underlying theoretical concept in this material is mirroring the logical structure of XML documents to tree structures, and describing the handling of XSL as a process of transforming between tree structures [1]. XSLT also provides an opportunity to demonstrate the use of declarative programming as opposed to imperative programming approaches [2].

In the intermediate level modules XML technologies are considered in the context of data driven web development and web services. J2EE web components [8] and PHP [10] server side technologies are used for the development of dynamic web sites in the Internet modules for BSc Computing and FD Computing, respectively. The knowledge of XML gained in the first year certificate modules is applied to explain web container configurations, JavaServer Pages Standard Tag Library, Java beans and the deployment of web applications in the BSc Computing module "Introduction to Web Development". Web Services and the main standards including SOAP Transport Protocol (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery and Integration) are introduced in the FD Computing module "Internet Technologies". **Web services** technologies are based on open XML standards, which aim to provide seamless communication between heterogeneous software applications. **SOAP** Transport Protocol provides the messaging mechanisms for interoperable exchange of messages over HTTP between clients and web services. SOAP Protocol includes messages with conventional XML, encoding rules for data types and mechanisms for remote procedure calls and responses. **WSDL** is standardized XML format for describing web services providing information about what exactly a web service does how to invoke it and where to find it. **UDDI** is a registry standard defining data structures and API operations for publishing and finding Web services. Practical examples demonstrating how to build a web server provider and client through specific PHP NuSOAP implementation are used in the lab sessions. Students are also required to develop clients for consuming real Web services such as Google Web API and Amazon Web Services [4]. Teaching these topics has given students exposure to new emerging technologies providing the opportunity to reinforce the learning of fundamental programming concepts, better understand Internet technologies, distributed systems and interoperability.

In the third year honours course XML related technologies are considered in the context of enterprise computing and distributed system architectures. The two models for processing and the representation of XML documents DOM (Document Object Model) and SAX (Simple API for XML) are discussed in detail during lectures and tutorials. XML **DOM** is the W3C Recommendation API that "defines the logical structure of documents and the way a document is accessed and manipulated". DOM-complained parsers need to implement the document structure as a tree and the DOM system as an object system where trees and nodes are represented as objects which contain both data and processing. The DOM topic reinforces the concepts of tree data structures, traversing trees as well as object oriented concepts including object representations, methods, inheritance, overriding, overloading and polymorphism. **SAX** is a model for processing XML documents based on a event-driven paradigm. With SAX-based parsing the document is processed element by element incrementally. As the parser finds an element it searches for an appropriate event handler. If there is one it passes the element for processing. Otherwise it moves to the next element. Students are already familiar with the event-driven paradigm through programming of GUIs using Java Foundation Classes/Swing APIs in previous modules. SAX parsing provides an opportunity to reinforce the learning of this paradigm.

Web services technologies are considered at honours level in more depth compared to the intermediate level presentation using Java based implementation. The students' understanding of XML applications is extended by discussions such as: (1) the place of XML in the two leading platforms Microsoft .Net and Java used for building enterprise applications, (2) XML in Mobile Computing, and (3) XML in distributed technologies.

Table 1 lists the XML topics included in the different modules/levels/courses and the respective underlying theoretical concepts illustrated through these topics. Similar linking for part of the topics is suggested in [2] and [8].

Table 1 Mapping between XML topics, underplaying theoretical concepts, modules, levels, courses

XML topic	Theoretical Concepts	Module/Level/Course
XML DTD	Data tree structures; Data types; Data modelling;	Internet and the World Wide Web/Certificate/ BSc Computing Web-site Design and Implementation Certificate/ FD Computing Internet Technologies Intermediate/ FD Computing
XSL XSLT XPath	Data transformation; Data tree structures; Manipulation of tree structures; Declarative programming; Data types; Expressions; Functions;	
XHTML WML	Data presentation; Separation data from presentation;	
Web Services: SOAP WSDL UDDI XML-RPC	Interoperability; Data modelling; Client-server paradigm; Remote Procedure Calls; Synchronous/Asynchronous communication; Messaging protocols; Data transportation;	Internet Technologies/ Intermediate/ FD Computing Programming the Internet/ Honorous/ BSc Computing Distributed Systems/ Honorous/ BSc Computing
XML DOM	Tree representation and traversing; Parsing; Object oriented concepts; Data transformation	Programming the Internet/ Honorous/ BSc Computing
XML SAX	Event-driven programming; Data transformation	
XML Schema	Data types and Data structures; Regular Expressions; Data modelling;	
XQuery	Functional programming; Expressions; Functions; Operations; Iteration; Sorting; Recursion; Data types	

The assessment strategies for the above modules include typically two hours unseen examinations covering the major topics discussed in the modules and individual or group assignments requiring the development of software artefacts. The student's experience is evaluated at the end of each module by an anonymous survey. The surveys indicate that, although students find these modules not easy, they enjoy the practical work and they are motivated by the opportunity to gain marketable knowledge and skills.

CONCLUSION

This paper has presented an overview of the approach used at the Department of Computing Communication Technology and Mathematics at London Metropolitan University for teaching XML technologies and it has illustrated how XML topics link to theoretical concepts in computing. Incorporating such popular technologies into the courses appears to motivate the students and at the same time it helps them grasp important fundamental theoretical knowledge in computing. The challenge is the constant need of updating the material and selecting relevant topics and tools. This requires careful planning of resources and staff training as well as institutional support. However, the experience of learning XML and related applications has proved to be rewarding and exciting for students and academic staff. The material presented here could be used as an example of embedding emerging technologies in the computing curriculum through existing modules.

– –

REFERENCES

[1] Bates, C. XML in Theory and Practice, John Wiley & Sons, England, 2003.

[2] Brookes, W. Computing Theory With Relevance, *ACM International Conference Proceeding Series* **57,** *in* Proc. of the sixth conference on Australian computing education **30**, 9-13, 2004.

[3] Conrad, M., French, T. A gentle transition from Java programming to Web Services using XML-RPC, in JICC7 Proceedings, 2005
URL: http://www.perisic.com/xmlrpc/jicc9/jicc9xmlrpcPaper.pdf

[4] Draganova, C. Web Services and PHP, in Proc. Conference on Information Technology in Education, 116-121, 2004

[5] French, T., Stephens D. Teaching Java, XML and e-Commerce: Some reflections of past and future approaches from the "front-line" at Luton University 1999-2002, in JICC7 Proceedings, 2003
URL: http://www.ics.ltsn.ac.uk/pub/jicc7/french3.doc

[6] London Metropolitan University web site
URL: http://www.londonmet.ac.uk/depts/cctm/cctm_undergrad/cctm_undergrad_home.cfm

[7] IEEE Computer Society and ACM, Computing Curricula 2001: Computer Science, Final Report (December 2001) **URL:** http://www.acm.org/sigcse/cc2001

[8] J2EE tutorial **URL:** http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html, 2006

[9] Paxton, J. XML in the CS curriculum: pointers and pitfalls, *The Journal of Computing Sciences in Colleges*, **17** (2), 106–111, 2001.

[10] PHP **URL:** http://www.php.net/, 2006

[11] Tucker, A. B., Kelemen, C. F. & Bruce, K. B. Our curriculum has become math-phobic!, in 'Proc. 32nd SIGCSE technical symposium on Computer Science Education, SIGCSE 2001', Charlotte, NC, USA, ACM Press, 243-247, 2001.

[12] World Wide Web Consortium **URL:** http://www.w3.org/XML/, 2006.

За контакти:

Dr Chrisina Draganova, Department of Computing, Communication Technology and Mathematics, London Metropolitan University, E-mail: c.draganova@londonmet.ac.uk