The Role of XML in TMN Evolution

David Lewis,
UH communications A/S,
Denmark.
dl@uhc.dk

Jens D. Mouritzsen,
Department of Product Development,
TeleDanmark,
Denmark.
jdm@tdk.dk

1.1. Abstract

This paper reviews the potential of XML a key technology in the evolution of TMN. The current state and applications of XML is reviewed with respect to the communication management domain. XML is suggested as a way to build on TMN's strengths while addressing its weaknesses in terms of being over-specialized and difficult to mediate with. An example of a current industrial application of XML in TMN system procurement is given.

1.2. Keywords

XML, TMN, Alarm Surveillance

2. 1. INTRODUCTION

This paper discusses how XML and related standards may impact on the evolution of TMN and how such an evolution might be incorporated into TMN platforms.

XML has a natural affinity with communications management. The hierarchical structure of XML documents reflects well that of the Manager-Agent models typical in management applications. Therefore, the use of general purpose XML tools to define management information models and to process information in management applications becomes an attractive possibility.

Another key feature of XML is its advanced support for hyperlinked information and thus its potential for use over the WWW. Presenting management information in XML allows user applications to leverage the low cost and ubiquity of WWW browsers and servers. In addition, XML-based management applications can exploit existing support for secure interactions and firewall traversal using HTTP.

This paper first gives an overview of XML, its potential benefits and relevant applications. It then discusses some of the evolutionary pressures currently being brought to bear on TMN before discussing in detail how XML may be applied to TMN to address these pressures. An example is provided of how XML is being applied by a major telecommunications provider as part of its TMN strategy.

3. 2. XML OVERVIEW

This section provides a brief overview of XML before discussing its application in areas relevant to communications management.

3.1. 2.1 The XML Standards Family

XML was standardized within the WWW Consortium (W3C) with the aim of bringing mark-up based publishing to the Web [1]. In addition to supporting human access to information, XML has also found strong applications in data processing and system integration due to it ability to be machine processable but in an implementation independent way.

A key feature of XML is the separation of a document's content from how it is presented. The latter is defined not in the XML document, but in a separate Stylesheet that can be associated with it [2]. The format of stylesheets is defined by W3C in the form of the eXtensible Stylesheet Language (XSL) [3]. XSL is extended to include support for transforming an XML document of one type into one of another type using the XSL Transformation (XSLT) [4].

XML also benefits from standardised API such as XML & HTML Document Object Model (DOM) [5] which is an API that supports to manipulation of HTML or XML documents.

3.2. 2.2 XML for Data Communications

XML is being used in several quarters for the definition and encoding of messages in data communication protocols. Though such messages could be transported in many different ways, including via TCP sockets or as parameters in simple method calls using distributed processing platforms such as CORBA, DCOM or RMI, the immediate advantages of XML come from when the message are transported using HTTP. This allows management communication to easily traverse firewalls, which is currently a problem for all management protocols and many distributed processing protocols. Other security mechanisms available in the WWW, such as secure interactions via SSL, could also be exploited using this approach.

Several lightweight protocols for exchanging messages or performing object-oriented RPC operations have been proposed including the Scarab Open Source communications framework, Microsoft's Simple Object Access Protocol (SOAP), XML-RCP and XML Transfer Protocol.

The XML Encoding Rules are an attempt to define an equivalent for the BER for ASN.1 using XML. This has been initially defined to support the use of ASN.1 for exchanging information between geographical information systems, but is done in a general purpose way that may also support the encoding for transmission of ASN.1 types currently used in OSI and Internet management models. Such an approach is being further examined by the ITU-T's Q22/7 which is defining mappings between ASN.1 and XML Schema datatypes.

From comparisons of XML and its transport using HTTP to CORBA [6][7] some disadvantages of XML are visible, particularly in the overhead introduced though text-only encoding of data and in support for RPC like behavior, i.e. return values and exception handling. Additionally, XML/HTTP solutions suffer from the lack of availability of common distributed processing support services provided by more mature platforms, e.g. CORBA services.

XML/HTTP does benefit from being readily deployable on the Internet, with the accompanying proven availability, scalability and firewall interoperability, which CORBA currently lacks. Industry agreements on XML interoperability seem to be reached relatively easily, partially due to XML's flexibility, simplicity and technology independence. This, coupled with the factors described above, indicate that XML/HTTP may be preferred over CORBA or other distributed object technologies for simple inter-domain management interactions over the Internet. Distributed object technologies may be preferred for demanding distributed applications operating within a domain. It is interesting to note that the OMG has also recently issued an RFP that aims to be able to represent XML documents as OMG IDL data types and therefore to support the transport of XML documents via CORBA interfaces. Solutions such as this point to the benefits of being able to separate the representation of data in XML from its transport, which may exploit a number of alternative protocols.

3.3. 2.3 XML for Model Specification

The ability of XML to express a huge range of document types means it has strong potential in the specification of management information and management interfaces.

The OMG has defined the XML Model Interchange facility (XMI) [8] for exchanging object-based models between CASE tools using XML. XMI is an instantiation of the Meta-Object Facility (MOF) [9], so that XML can be used to exchange any model conforming to the OMG's MOF, such as UML models or CORBA Component Models. This presents the possibility of mapping between XMI documents and other types of XML management data documents, providing a potentially useful integration route to UML-based CASE tools during analysis and design of specifications (as now advocated in the TMN Methodology for the Definition of Management Interfaces).

The OMG has also opted to use XML for the descriptor of assembly packages, to support the deployment and runtime configuration of CORBA component assemblies [10]. XML is used in a similar way for Enterprise Java Beans. Thus XML tools can play a useful role in the future configuration and deployment of component based management applications.

3.4. 2.4 XML in Web-based Management

In the management domain, the most visible user of XML is the Distributed Management Task Force (DTMF). The DMTF defines standards for the unified management of applications, computers and networks, primarily aimed at the enterprise management market. It has defined an extensible Common Information Model (CIM) [11] expressed in its own Meta-Object Format (MOF, not to be confused with the OMG's Meta Object Facility). The DMTF runs the Web

Based Enterprise Management (WBEM) initiative, which specifies an XML mapping for the CIM [12] so that managed object classes and instances can be encoded in XML. It also defines a set of XML definitions [13] for messages that exchange information on managed object classes and instances using HTTP.

Some work has been performed in mediating between TMN and earlier versions of WBEM [14]. This involves the mediation from a WBEM agent equivalent via a gateway to an OSI manager. This solution requires a mapping from the CIM instances expressed in the MOF to GDMO and GRM, including the support for class information, i.e. agent meta data. In addition a mapping is defined between CMIP and the Hyper-Media Management Protocol, which is now replaced in WBEM by the DMTF's CIM to XML/HTTP mapping.

Another XML related standardization initiate is the telecommunications Mark-up language (tML) which has recently begun in T1M1. This aims to provide a framework for the development and repository based publication of XML schema for network management. Though currently this initiative is at an early stage, one interesting contribution has been the description of Q3ML [15]. This is an XML management interface for OSS based on TMN. It supports the XML encoding of TMN managed objects and ASN.1 attribute values and their transport over a HTTP based protocol with messages modeled on CMIS primitives. This contributions also points out the ease of mediation from Q3ML to other format using XSLT.

3.5. 2.5 The Potential Benefits of XML

The use of XML family of standards presents many potential benefits to the communications management sector. As well as the benefits of being able to exploit the ubiquity, low cost and security of the WWW infrastructure (e.g. HTTP browsers and servers) for management, XML offers further benefits from the wide skills base and the range of tools and system integration mechanisms its broad applicability provides.

The expressive potential of XML makes it possible to represent the wide range of information models present in existing management standards and deployed solutions. This allows common tools to be used for browsing, editing and processing these models, offering cost savings in application development. Such tools could, for example, be based on the DOM which is designed to be used with any programming language, and which enables programmers to build XML documents, to navigate their structure and to add, modify, or delete elements and content.

The separation of data structure and presentation in XML offer the benefits of flexible configuration of user applications, which could operate in WWW browsers. Integration with XSL style-sheets would allow the dynamic location and loading applications for manipulating the data in a user application, e.g. using Java applets. Additionally, multiple style-sheets could be used for the same agent, allowing a user application to dynamically load different sub-components for manipulating the agent's data in different ways, allowing the cost and complexity of the management application to be flexibly matched to the user's actual needs.

The widespread uptake of XML as the lingua franca of the e-commerce industry, as seen in the standards activities of bodies such as Open Application Group and ebXML, also provides a major incentive for adopting XML-based solution for management. It promises the wide availability of appropriate expertise, the possibility of reusing standard XML business object definitions for management and the easier integration of management and other XML-based business applications. This is especially useful when developing applications at the service and business layers of TMN.

The separation of information represented in XML from the mechanisms to present, transmit and store that information also makes it a flexible solution for the specification and manipulation of corporate data, as is shown by widening XML support in database products. The need to store and retrieve management information in databases makes this another potential benefit of using XML.

These benefits of XML are discussed in more detail with respect to TMN platforms and application in section 4.

4. 3. TMN EVOLUTION

TMN today exists in a very different environment today compared to when it was first developed. A wide range of open distributed processing techniques can potentially support the requirements of telecommunications management. However, TMN still retains its strength in the wide range of conforming standards that already exist, with a well established common models providing reusable definitions of useful managed object attributes (e.g. X.721), management functions (e.g. logging), and network models (e.g. M.3100). In addition, scoping and filtering and fine grained notification control enables very scalable implementations, with agents supporting more than a million managed object instances being possible. To successfully evolve, therefore, TMN must continue to fully exercise these strengths to ensure interoperability, flexibility and scalability in future management systems.

The challenges that TMN faces, and which it must evolve in order to meet, stem largely from the rise of many popular general purpose technologies suitable for open distributed computing. The two main categories of such technologies are Web-based ones, which include many applications of HTML, CGI, Java Applets and XML, and object-oriented middleware based ones, such as OMG's CORBA, Sun MicroSystem's Java/RMI and Microsoft's DCOM. The wide application of these technologies have led to more widespread and lower-cost tool support and much broader skills base than can ever be achieved from the specialized application range of TMN. There is therefore an imperative for communications management solutions to embrace the advantages of these general-purpose technologies and avoid the specialized nature of current TMN solutions.

As CMIP is used at the network and network element level, where scalability is a prime concern and mature standards are sought, mediation solutions between GDMO/CMIP and other technologies have emerged, e.g. JIDM. They support the co-existence required to allow some management systems to migrate to different technologies while others remain implemented using CMIP. Such mediation support for the interoperation between TMN conformant technologies and emerging one will continue to be required in the future.

TMN needs to evolve if it is to supply the same architectural and interoperability support to the management of future IP-based and mobile networks that it has done to telecoms networks. The general convergence in the communications market will mean TMN must support the integration of a wider range of management technologies emerging from different sources and controlled in different bodies. Assuming the pace of adoption of new technology within the ITU cannot be greatly increased, TMN must exploit its current strengths while facilitating the rapid integration of TMN systems with other technologies. This places the following further requirements on the successful evolution of TMN:

- TMN models must be made more amenable to implementation in technologies not currently conformant to TMN protocol standards.
- The rapid development of mediation solutions between TMN conformant systems and systems implemented in emerging but non-conformant technologies must be facilitated.
- From a more concrete perspective, evolution routes for TMN that exploit XML should aim to provide interfaces and APIs that will be easy to use and familiar to developers of XML applications in other fields.

5. 4. THE APPLICATION OF XML TO TMN

This section discussed approaches to how XML can be applied within TMN's existing structure, both in assisting in the generation and publication of MIB specifications and as the basis for a new protocol binding for CMIS. An example application of XML from a telecommunications provider is presented to further explore the issues involved.

5.1. 4.1 Using XML for Representing MIB Specifications

XML can be used to specify an XML vocabulary for specifying management models. This can be done as a literal mapping from GDMO, using its keywords as the basis for tags. Alternatively it can be done in a more general manner that supports the definition of management object classes and attributes that is independent of the syntax of any existing modeling language, i.e. GDMO, SMI, IDL, MOF, but readily mappable to them.

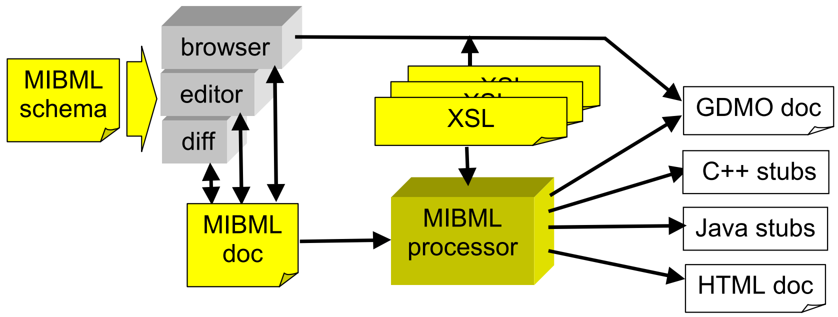

Fig 1.: Use of a MIB Specification Mark-up Language

Such an approach may therefore be used in providing a XML vocabulary for defining agent MIBs as XML documents (lets call this vocabulary MIBML). To support existing MIB specifications, a standard mapping for going from a GDMO document to a valid MIBML document must be defined. The MIBML will be aimed at aiding the generation, reading, maintenance and publication of both standardized and proprietary MIBs though the exploitation of general purpose XML tools and APIs. For instance, XSL style sheets could be used to govern the browsing and editing of MIBs in various formats, including containment and inheritance tree views and in-line presentation of packages, attributes, ASN.1 values etc. Appropriate use of XML name spaces will allow this approach to support the on-line importation and browsing of fragments of external specifications, a common requirement in GDMO specifications using packages and inheritance.

To ensure backward compatibility with GDMO processing tools an XSL style-sheet definition for mapping MIBML document to GDMO documents would also be required. A MIBML document generated from an original GDMO document should, when presented using the style-sheet rules, produce a GDMO specification equivalent to the original model, i.e. a resulting manager application generated from the original model will interoperate with an agent generated by the reproduced model or vice versa.

Generation of manager and agent stub code in a TMN platform can exploit the potential of XSL and XSLT. The output of the MIBML processor may be controlled by XSL definitions, mapping from

MIBML to whatever output format is required. This makes the process of developing templates for new forms of output much more familiar to anyone with experience in developing XSL style-sheets of transforms.

5.2. 4.2 Using XML for Transporting Management Information

To exploit XML in transporting management information, a literal approach could be taken where CMIS primitives are encoded as XML, possibly using the XER mapping. Alternatively, fragments of XML conforming to agent specific MIBs could be used for encoding the values of MOs and attributes.

Such XML messages may be transported directly via TCP/IP, however to benefit from WWW infrastructure especially for remote user or inter-domain interactions, HTTP may be preferred. The main problem faced by the use of HTTP is the fact that HTTP only supports data retrieval from a server and not asynchronous messages in the other direction. An analysis of various solutions to this problem in [16] concludes that to avoid problems with TCP connection timeouts and firewall negotiation, the manager should also be able to act as a HTTP server and the agent as a HTTP client.

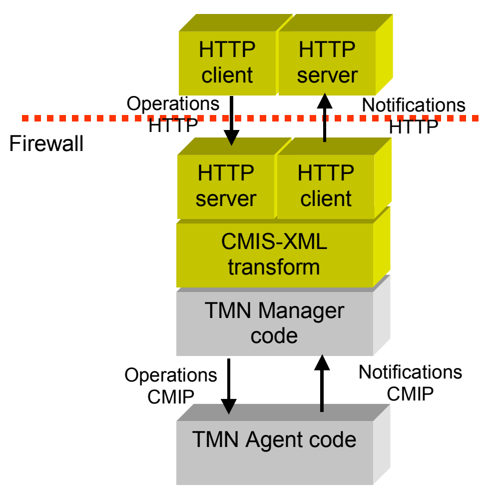

Fig 2: Use of XML for transport of TMN management information

A first stage in rolling out XML access to TMN management information would be the construction of HTTP/XML to CMIP gateways (as depicted in fig. 2) which would allow access to existing CMIP agents.

6. 5. EXAMPLE: TDK's XML ALARM SURVEILLANCE SERVICE

This section presents the “Simple Textual Alarm Interface” (STAI/QXML) specification used by Tele Danmark Communications (TDC). This specification specification defines the minimum requirements for a text-based alarm reporting function from equipment (in TMN called a Network Element) to a Q-adapter.
Two alarm formats are defined. One which is in accordance with the TMN format for alarm reporting and one, which contains the required information to provide a TMN-like alarm reporting function. For both formats a XML Document Type Declaration (DTD) and a W3C Schema are defined. The specification including alarm examples is available at (http://muun1.tdu.dk/xml/stai-xml.)
The reasoning for introducing a low-level and low-cost method for Alarm Surveillance is based on experience with vendors of additional service and supporting systems outside the mainstream telecommunication world. The requirement for a standardised Network Management interface, either TMN or IETF, are generally overlooked and neglected by these vendors. STAI/QXML has been proposed to the vendors, to integrate important service equipment and systems into the operator's central alarm surveillance function, in order to ensure and enhance an overall service quality to the customers.
The idea behind the specification is that alarms from equipment are transmitted to the Q-adapter as pure text on a TCP connection, i.e. as TCP data. The alarm information is formatted using a QXML template as an XML document according to the rules defined in the specification. By utilising XML the vendors can utilise low cost XML tools to generate alarms, and Q-adaptor developers can utilise standard XML techniques to process alarms.

The vendor has to specify a model describing the management aspects of the equipment. This model equals a GDMO description [17] in terms of TMN. The model shall consist of:

- A description of the modules comprising the equipment and their relationships. In GDMO terms this is a hierarchy of Managed Objects (MO), where the root of the tree represents the equipment (Network Element) itself. The class (type) of the MO must also be specified, and the name of each MO must be defined.
- Which type (communication, equipment etc.) of alarms each module (Managed Object) may emit. A total list of alarms is not required. For each alarm type it is necessary to define the corresponding event type.
- It is also required to specify whether or not the equipment generates alarm-cleared events. If alarm cleared events are generated, a specification and description of the alarm cleared event for each alarm are required

Furthermore the vendor has to specify, whether or not the equipment generates keep-alive messages, and the interval at which these are sent to the Q-adapter. The equipment has the responsibility to establish the TCP connection and to re-establish the connection, if the connection is broken.

5.1 QXML Alarm Template

This example shows an alarm based on the QXML alarm template, where the information provided by the vendor and the dynamic information provided by the equipment at runtime is shown in bold typeface.

```
<?xml version="1.0" encoding='ISO-8859-1' ?>
<!DOCTYPE roiv-apdu SYSTEM "stai.dtd">
<roiv-apdu>
   <invokeID>45</invokeID>
   <operation-value oper="m-EventReport" />
   <argument>
         <managedObjectClass>smscFacility</managedObjectClass>
         <managedObjectInstance>
```

```
            <distinguishedName>
                  <rdn attrname="managedElementId">smsc</rdn>
                  <rdn attrname=" softwareId">AD</rdn>
            </distinguishedName>
      </managedObjectInstance>
      <eventTime timetype="mix">20000531102959.9+0200</eventTime>
      <eventType evtype="communicationsAlarm" />
      <eventInfo>
            <probableCause cause="5" />
            <perceivedSeverity severity="minor" />
            <notificationIdentifier>345</notificationIdentifier>
            <additionalText xml:lang="en-uk">This is an alarm
                  </additionalText>
      </eventInfo>
   </argument>
</roiv-apdu>
```

The specification also define a QXML template for keep-alive messages to be send from the equipment to the Q-adaptor at pre-defined interval.

The STAI Document Type Declaration (STAI DTD) and the STAI Schema are defined based on OSI alarms as defined in X.711 [18], X.721 [19] and X.733 [20]. However the CMIP protocol and the BER (Basic Encoding Rules) of ASN.1 is ignored because of the nature of XML.

As shown in the alarm example, an alarm is an instance of a roiv-apdu element.

The *roiv-apdu* element corresponds to the ASN.1 encoding of an OSI alarm in a CMIP PDU. A *roiv-apdu* contains the following sequence of elements:

```
<!ELEMENT roiv-apdu (invokeID, operation-value, argument)>
```

The *roiv-apduType* is defined as a xsd:complexType (W3C Schema) with content set to *elementOnly*, and has no attributes. The contained elements *invokeID*, *operation-value,* and *argument* are defined as a xsd:sequence.

```
<xsd:complexType name="roiv-apduType" content="elementOnly">
    <xsd:sequence>
        <xsd:element name="invokeID" type="xsd:positiveInteger" />
        <xsd:element name="operation-value" type="operation-
               valueType" />
        <xsd:element name="argument" type="argumentType" />
    </xsd:sequence>
</xsd:complexType>
```

The *argument* element can be viewed as a container for the elements *managedObjectClass*, *managedObjectInstance*, *eventTime*, *eventType*, and optional *eventInfo*.

```
<!ELEMENT argument (managedObjectClass,
   managedObjectInstance, eventTime, eventType, eventInfo?)>
```

The W3C Schema is defined using complexType and sequence similiar to the *roiv-apduType.*
Every managed object has a *relative distinguished name* (RDN) that identifies the object within the scope of its containing (superior) object. The *distinguished name* (DN) of an object comprises the sequence of RDN's from the root of the tree down to the managed object itself.
The sequence of *rdn's* identifying the managed object is an ordered sequence, i.e. the position of a *rdn* in the sequence is significant. To help to maintain this ordering the element *rdnlist* is defined to allow recursive usage. A *rdnlist* element contains one *rdn* element followed by zero or one *rdnlist* element:

```
<!ELEMENT rdnlist (rdn, rdnlist?)>
<!ELEMENT rdn (#PCDATA)>
<!ATTLIST rdn
          attrname   CDATA   #REQUIRED>
```

The *rdnlistType* is defined as a xsd:complexType with content set to *elementOnly*, and has no attributes. The contained elements *rdn,* and *rdnlist* are defined as a xsd:sequence, where the element *rdn* must occur one and only one time, and the element *rdnlist* must occur zero or one time:

```
<xsd:complexType name="rdnlistType" content="elementOnly">
    <xsd:sequence>
        <xsd:element name="rdn" type="rdnType"
                     minOccurs="1" maxOccurs="1" />
        <xsd:element name="rdnlist" type="rdnlistType"
                     minOccurs="0" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>
```

The *distinguishedNameType* is defined as a xsd:complexType with content set to *elementOnly*, and has no attributes.

```
<xsd:complexType name="distinguishedNameType" content="elementOnly">
    <xsd:sequence>
        <xsd:element name="rdnlist" type="rdnlistType"
                     minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>
```

The *moduleNameString* element is a simpler way to identify the module (managed object) than using the *distinguishedName* element construction. The module name is one long string where the individual parts (simple names of the modules/managed objects defining the nodes in the naming tree) are separated by a separator character. The first part is the topmost part (nearest the root node) in the naming tree. The used separator character is specified in the attribute *sep* and the value of the attribute is the hexidecimal value (as a string) of the separator character. The following separator characters are allowed:

- #, the attribute *sep* has the value x23
- @, the attribute *sep* has the value x40
- \, the attribute *sep* has the value x5C

```
<!ELEMENT moduleNameString (#PCDATA)>
<!ATTLIST moduleNameString
          sep  (x23 | x40 | x5C) "x40">
```

The *moduleNameStringType* is defined as a xsd:complexType derived by extension from the type xsd:string. It has only one attribute (*sep*), which specifies the separator character. The attribute *sep* is defined as a xsd:simpleType based on xsd:NMTOKEN and with the allowed values defined as enumerations.

7. 6. CONCLUSIONS AND FURTHER WORK

This paper outlines how XML can play an important role in the evolution of TMN. XML will allow TMN to preserve its strengths while making it accessible to a wider range of software developers and making it more easily interoperable with the rapidly increasing range of technologies that can be applied to management. This paper shows how XML may be applied both as an aid in TMN specification and publication process and in the transport of information between managers and agents. In these roles, XML is used in place of GDMO and CMIP respectively, allowing it to fully exploit the existing strengths of TMN.

The primary advantages of such XML applications are that they will make TMN application programming more accessible to a broad range of XML developers and will allow easier integration with the growing range of general-purpose XML tools. In addition, these applications increase the capability of TMN to support mediation between different management technologies and information models.

The use of XML for accessing agents implementing service level TMN models is being prototyped in the EU-funded project FORM using an XML-HTTP to CMIP gateway (similar to that described in section 4.2). The feasibility of using an existing TMN platform (UHC's Q3ADE) for supporting management information models represented in XML is also being examined. The DMTF's XML mapping for the CIM is being used as the MIBML in this assessment. The gateway implementation has been developed to explore the ease of applying XML tools (a DOM implementation is used) and is not intended as a platform for assessing performance issues of an XML-CMIP mapping, which will require yet further study.

8. Acknowledgments

This work was conducted under the partial funding of the EU through the IST project FORM (contract IST-1999-103571). The views expressed in this document do not necessarily reflect those of this consortium. The authors would also like to thank Lever Wang of DeriveIT and Ed White of Worldcom for sharing their informative views on the issues presented in this paper.

9. References

[1] Extensible Markup Language (XML) 1.0, W3C Recommendation: REC-xml-19980210, World Wide Web Consortium, Feb 1998, http://www.w3.org/TR/REC-xml.

[2] Associating Style Sheets with XML documents, Version 1.0, W3C Recommendation 29 June 1999, http://www.w3.org/1999/06/REC-xml-stylesheet-19990629

[3] Extensible Stylesheet Language (XSL). W3C Working Draft, World Wide Web Consortium http://www.w3.org/TR/WD-xsl

[4] XSL Transformations (XSLT), Version 1.0, W3C Recommendation, World Wide Web Consortium, 16 November 1999, http://www.w3.org/TR/1999/REC-xslt-19991116

[5] Document Object Model (DOM), Version 1.0, W3C Candidate Recommendation:CR-DOM-Level-2-20000307, World Wide Web Consortium, 7 March 2000, http://www.w3.org/TR/2000/CR-DOM-Level-2-20000307

[6] XML and CORBA, synergistic or competitive?, S. Vermeulen, B. Bauwens, F. Westerhuis, R. Broos, Proceedings of the 7th International Conference on Intelligence in Services and Networks, IS&N'00, Athens, Greece, March 2000

[7] CORBA and XML: Conflict or Cooperation? A Watson, Java Developers Journal, volume 4, issue 9, September 1999

[8] XML Metadata Interchange, ad/98-10-05, OMG, Oct 1998

[9] Meta Object Facility, revised submission, ad/97-08-14, OMG, Aug 1997

[10] CORBA Components: Joint Revised Submission, OMG TC Document orbos/99-02-05, Mar 1999

[11] Common Information Model (CIM) Specification v2.2, Distributed Management Task Force, 14th June 1999.

[12] Specification for the Representation of CIM in XML, Version 2.0, Distributed Management Task Force, June 2nd, 1999

[13] Specification for CIM Operations over HTTP, Version 1.0, Distributed Management Task Force, August 11th, 1999

[14] Integration of WBEM-based Management Agents in the OSI Framework, O. Festor, P. Festor, N.B. Youssef, L. Andrey, Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, Boston, USA, pp 49-64, IEEE, May 1999.

[15] Q3ML - XML Based Management Interfae for OSS, L.Wang, DeriveIt, T11.5/2000-204, June 2000.
[16] Push vs. Pull in Web-Based Network Management, J.P. Martin-Flatin, Proceedings of the Sixth IFIP/IEEE International Symposium on Integrated Network Management, Boston, USA, pp 4-18, IEEE, May 1999.
[17] ITU Recommendation X.722: Data communication networks — Information Technology — Open Systems Interconnection — Structure of Management Information: Guidelines for the definition of managed objects.
[18] ITU Recommendation X.711: Common management information protocol specification for CCITT applications.
[19] ITU Recommendation X.721: Data communication networks — Information Technology — Open Systems Interconnection — Structure of Management Information: Definition of Management Information.
[20] ITU Recommendation X.733: Data communication networks — Information Technology — Open Systems Interconnection — Systems Management: Alarm Reporting Function.

10.Biographies

David Lewis graduated in Electronic Engineering at the University of Southampton in 1987 and in 1990 received a MSc. in Computer Science from University College London where since he has worked as a research fellow. He has worked primarily on the EU funded projects in which he has been responsible for leading teams developing integrated, multi-domain service management systems. He has a Ph.D., on a service management development framework for the open services market.

Jens Mouritzsen gained his BSc in Electronic engineering at Herlev Teknikum in 1990. Same year he joined the Development division in KTAS (now Tele Danmark). He has worked partly on the EU funded projects on service and network management (TMN), and partly on internal development projects including implementation of alarm surveillance systems.