XML for plug-and-play applications in service delivery

P. Brandt MSc
J. Bruijning PhD
TNO Informatie- en Communicatietechnologie,
Brassersplein 2, 2612 CT Delft, the Netherlands
E: paul.brandt@tno.nl

P. Brandt, J. Bruijning. XML for plug-and-play applications in service delivery. Gerontechnology 2007; 6(3):147-154. Personal independence and active societal participation up to a high age call for a good open-system architecture and plug-and-play applications preventing 'vendor lock in'. The supportive software which uses the ICT hardware infrastructure is the subject of this contribution. TCP/IP forms the standard for exchanging data, however, for turning data into information a number of (vendor-based) protocols are currently available. XML, the eXtensible Markup Language, as formulated in 1998, may become the de facto standard due to its popularity and its management by a number of standards development organizations. Like any ordinary language, XML has grammar and domain vocabularies. Established domain vocabularies used for delivering care services include HL7 (clinical data and practices) and CCR (care data in real-time). In addition, semantic web applications are expressed in XML and the architecture of a plug-and-play environment has been formed in SOA (Service Oriented Architectures). Mass-individualized teleservices so urgently needed in our aging society, may be realized through extensive use of XML.

Keywords: open standards, protocols, XML, service oriented architectures

Increasing personal independence, prolonging active participation in society including an ubiquitous and friendly access to public information, integrated leisure and care processes for the ageing population, are impacts expected by the European Commission from the 2007 call 'ICT for Independent Living and Inclusion', which is part of the 7th Framework Programme of the European Community for research, technological development and demonstration activities. One of the targets focuses on open systems architectures, standards and platforms that enable systems and services for independent living, smart work places and mobility. These should support seamless integration and plug-and-play operation of sensors, devices, sub-systems and integrated care services into cost-effective, reliable, secure and trusted systems[1]. This approach calls for a separation of functionality and the coordination of open protocols to create these desired outcomes. In this model the software makes no assumption as to how the application is used; this process is left to the coordinating function formed within or among the dedicated applications. In this way, applications from different producers can be integrated within the same ICT infrastructure.

Traditionally, vendors have developed and marketed their own solutions, resulting in a large variety of non-interoperable proprietary applications. This led to 'vendor lock-in' where customers were practically bound to their supplier for continued support and for additional functionality (a supply-driven market). Since mass implementation is required for our services to meet the needs of the ageing society, 'vendor lock-in' should be prevented by using open standards for communication over the ICT hardware.

From the ICT viewpoint, one needs both applications and infrastructure to deliver services that can be shaped to individual needs and used in a variety of circumstances (a more demand-driven market). The applications delivering user functionality reside within devices whose specific embedded software is present at the user's site or at that of the service provider and it assists, supports, informs and monitors the user at home or 'on the road'. In the plug-and-play situation these devices become operational as soon as they are connected to the ICT infrastructure.

The ICT hardware structure itself consists of the physical communication lines (copper, coax, glass fibre) or wireless communication options (WIFI, RFID, Bluetooth, Zigbee), and dedicated servers that may be intermittently or permanently in contact with the equipment in the field and with each other[2].

The supportive software that uses the ICT hardware infrastructure to enable application functionality is the subject of this contribution. We present the state-of-the-art and expected near-future developments in the internet protocol and XML as open software protocols for ICT communication used in service delivery. Collaborative design techniques involving end-users, for instance, older adults, are also important for a successful outcome, but are not treated in this contribution.

TCP/IP

Data communications is the means through which applications can exchange the information needed to serve the user. The fully open Internet Protocol or TCP/IP is not a commercial development. It was invented by Vinton Cerf and Bob Kahn under contract from the US Department of Defence to interconnect dissimilar systems[3]. This protocol is a common ground for different data communication technologies to interoperate, and all modern applications use it for interoperation. TCP/IP will exchange data, not information. For example, TCP/IP has sent the number '42', a number which in itself is meaningless. But if one knows that it denotes a body temperature, it has become information. Transforming data to information is the aim of secondary protocols, built on top of TCP/IP. Initially these secondary protocols were bound to specific applications. For instance, e-mail applications need to distinguish address fields, dates, as well as more technically oriented information regarding routing and data consistency. In order to support these application requirements, protocols such as SMTP, IMAP and POP3 have been established. At the same time, SMTP, IMAP and POP3 cannot support any application other than e-mail. Hence, each network application developed its own protocol suite to run on top of TCP/IP. The more elaborate these applications grew over time, the more complex the supporting protocols grew to fulfil them. The result has been a high number of protocols with partly or fully overlapping functionality, some competing to become the 'de facto standard' in their application domain thus creating vendor lock-in.

Birth and growth of XML

In order to turn this untamed growth and remove the barrier to mass-introduction, a secondary open protocol was needed that followed two overall laws: First, the protocol should be independent of any application and, secondly, the protocol should be open, that is independent from any single vendor, but receiving commitment from all main vendors. The goal of the proposed protocol is to turn data into information. Since supporting any and every application with a single protocol is a utopian ideal, a language was invented to foster the creation of dedicated protocols. On the 10th of February, 1998, XML was born[4], the eXtensible Markup Language, applying open standards. XML is a uniform, so-called meta language comprising other languages and protocols that bridge the gap between data and informa-

tion, and is managed by a conglomerate of Standards Development Organizations such as W3C, Oasis, IETF[5,6] to foster interoperability among systems of different origin. XML specifies linguistic rules that enable one to express oneself, presenting the specifics that are needed to communicate information.

Just as with ordinary languages XML has two aspects: the grammar to provide the technical communication layer and the domain vocabulary to carry the information. The XML family of standards provides functionality that can be put into practice on the technical communication level as well as the domain vocabulary level since it operates on XML data itself. For example, the XML security standards establish guiding principles for implementing security that can be put into practice for secure communications as well as ensuring secure electronic health care records.

XML grammar

The technical communication layer establishes a common ground for communicating peers to enter into conversation. This common ground includes agreements on how to distinguish words, how to represent the various data types (numbers, booleans and alike), how additional functionality such as security should be plugged in, as well as how to resolve gracefully unexpected or erroneous communication patterns. These critical agreements inform us that, for example, the characters '4' and '2' in that specific order represent the number 42. The power of this approach should not be underestimated; as stated by Knox et al., "XML has grown from a little-known standard in 1998 into the basis for a Web computing infrastructure"[7]. The rapidly growing number of devices supporting XML has made it a de facto standard for delivering services.

Since XML allows the combination of numerous collaborative devices in a simple way, it is well suited to improve ICT-based services for public information, home automation, integrated leisure and care processes.

Domain vocabulary

The domain vocabulary provides a common notion of understanding between communicating peers. It includes agreements on what entities the words represent and how words are combined into valid sentences and what conversation patterns are allowed. These important agreements tell us that the number 42 represents temperature, that this temperature has been expressed in units of Centigrade and applies to a human addressed as Mr. Arthur Dent[8]. The process of arriving at a well-balanced, common domain vocabulary is a painful and cumbersome but nevertheless important process[9]. Without such common vocabulary exchanging information is virtually impossible. Some domains have well-established open domain-vocabularies; those relevant for delivering care services to older adults include HL7, *open*EHR and CCR.

HL7 is the electronic protocol for the health care community which successfully achieved a common ground for information exchange between health professionals implemented as protocols in clinical practices and well-established formats to express clinical data[10]. It originated in 1987, as 'ANSI standard for healthcare specific data exchange between computer applications'. The first HL7 standard had already been released in 1990; however, it was not released as an XML expression until 2000. Although an American initiative, it is now also widely used in Europe.

In a nutshell, "*open*EHR" consists of an international community delivering a set of XML specifications for a highly stable reference model for a distributed health computing environment. This provides a language and model for expressing clinical archetypes, templates and a terminology for use with electronic health record (EHR) systems[11]. The *open*EHR work is well

embedded in European (CEN TC/251) and international (HL7 and ISO) standardisation. The *open*EHR Foundation publishes all its specifications and builds reference implementations of them, as open source software.

A final example is the Continuity of Care Record (CCR) which is a core data set of the most relevant and timely facts about a patient's healthcare. It is routinely prepared by a practitioner at the conclusion of a healthcare encounter in order to enable the next practitioner to readily access the patient's information. It includes a summary of the patient's health status (for instance, problems, medications, allergies) and basic information about insurance, advance directives, care documentation, and care plan recommendations. It also includes identifying information and the purpose of the CCR[12].

An expanding collection of references to standards activities and formal specifications used in clinical research and healthcare industries is available for the reader interested in learning more about these protocols[13].

Semantic Web

The Semantic Web is an evolution of the World Wide Web that provides an infrastructure of globally interconnected information sources that are available to systems as opposed to humans. The conceptual basis is that the data that is generally hidden within Web pages, understandable only to human readers, and meaningful only in the highly specific context of that Web page, can be made globally available to other computers on the network as information. Systems accessing that information can combine it with other information relevant to a specific context, such as to relate that Mr. Arthur Dent belongs to the human species, to determine that a body temperature of 42˚C falls outside the safe operational limits for human beings and to deduce that Mr. Arthur Dent needs to consult a medical doctor immediately. This requires the ability to define an explicit and detailed model of the concepts that apply to the application domain; ontologies are one such suitable means for representing this knowledge, providing a shared domain theory that can be used for communicating information in a common context. Additionally, ontologies can be employed for deductive reasoning and model manipulations. The Semantic Web initiative[14] is dedicated to providing the necessary tools and standards to achieve this level of information exchange, correlation and reasoning: XML Schemata, Resource Description Framework (RDF), Ontology Interchange Language (OIL), DARPA Agent Markup Language (DAML) and Web Ontology Language (WOL) are examples of standards (that are expressed in XML) to facilitate the Semantic Web[15]. The Semantic Web provides an adaptive and self-learning environment for persuasive intelligence without the intervention of humans.

Bridging requirement & design

Architecture is the bridge between *requirements*, that identify what needs to be realized, and *design,* that describes the means to the realization. The architecture identifies main functional blocks, the way they interoperate, and the supporting technology. Architecture relies on common standards and building blocks to define the 'needed service to be built'[16].

Much information must be exchanged before a service can be delivered. Applications need to inform each other what they can and cannot do, and how their services can be invoked. Some components of these applications will assume a coordinative role, supervising the activities of other components. Where faults occur they must be caught and remedies applied by these coordinating services. Not only must service be delivered, but administrative actions such as reporting and billing must also be executed. Most importantly,

it must be possible to alter existing services and add new functionality in a simple and non-disruptive way. Collectively we term this the architecture of a plug-and-play environment.

With the help of well designed architectures, open protocols can combine functionality from different sources. These functions can be gleaned from systems operated by different vendors situated at widely spaced geographical locations, by partners that may own diverse pieces of information. From just a few building blocks, service providers can assemble offerings tailored to their individual customer. Cost savings to the customers can be realized if vendors consider forgoing software development in-house and choose to purchase mass produced software from third parties.

A consequence of the architecture view is that the users must select an architect for their system. The architect should not have any commercial interest in the building effort. Rather, the architect supervises the building work and ensure the guidelines are followed and proper quality control is observed. The importance of having an independent architect can not be overemphasized; however, in practice it is often overlooked.

Service orientation

An insightful statement of the famous Dutch computer scientist E.W. Dijkstra is a maxim for XML development: "Minimal coupling, maximal cohesion"[17]. Preventing interdependencies of functions and providing loose coupling by design is a key requirement for achieving system interoperability whilst preserving guarantees for simple maintenance, as well as flexibility and endurance, over the full system life cycle. This issue is the main driver behind the concept that is currently known as service orientation.

The model behind the service orientation concept is: Service providers deliver well-specified, stand-alone services to consumers to fulfill individual needs. Services are described in terms of what they deliver to the consumer. How service providers produce their services is fully transparent to the service consumer. Delivery of services is conducted according to mutual agreements on functional as well as non-functional (i.e., Quality of Service, security levels, performance guarantees) prerequisites that have been described by a formally specified interface.

Service Oriented Architectures (SOA) therefore distinguish themselves on the following characteristics[18,19]:

(i) All functions have been defined as services; each service embodies a logical functional compound at a coarse level of granularity, representing system functions, or business functions or even complete business transactions that contain several low-level operations.

(ii) All services are independent of each other and can operate as stand-alone functions. Service recipients are unaware of service internals and are not interested in how their functionality is accomplished. The fact that the service provides the agreed functionality within the agreed upon operational constraints is the only important factor.

(iii) A service is accessible only through its interfaces. From this perspective it is only relevant to know that services are somehow interconnected whilst it is completely irrelevant to know (a) The location of the service; (b) The specifics of the communication protocol to address the service (although this may be important for implementation).

(iv) An SOA explicitly provides the means and techniques to minimize the consequences of failures, such as asynchronous communication, mechanisms for transaction roll-back, reliable messaging, redundant deployment and more. Although application of these techniques is not restricted to SOAs, an SOA provides these techniques as default implementations.

As a result, within an SOA a business process can be designed using loosely coupled, selected, re-usable and distributed services in a process-specific order. This is known as service orchestration and is typically suited for collaboration, even across organizational boundaries.

System interoperability

We have established a domain vocabulary and a Semantic Web to provide a shared understanding of our data. We have established an open communication layer as vehicle to carry that data between peers. We now must establish the ability to operate on that data, or more precisely, to establish the ability for systems to interoperate. Web services represent the contemporary SOA implementation. They are fully supported by the main vendors (Microsoft, Sun Microsystems, IBM, Oracle, BEA to name a few) and are implemented by well accepted, open standards expressed in XML, from which SOAP (an acronym that has survived its initial, now incorrect, abbreviation) for service interoperation, WSDL (Web Service Description Language) for service definition and UDDI (Universal Description, Discovery and Integration) for service publication and discovery form the basic triangle. Standards for security, transactions, business process orchestration and reliable messaging, to name a few, continue to evolve[20]. Standardization bodies recognize the risk of unmanageable complexity and have employed corrective actions, such as providing standards (WS-I: Web Service Interoperability) for the sole purpose of controlling complexity in web services.

Mr Arthur Dent revisited

Now, let us visit our hypothetical Mr. Arthur Dent again in whose home XML has been implemented in different services.

Living in France, the eldest son of Mr. Arthur Dent decided to monitor his older father in his home in Pennsylvania so as to detect his falling down, and perhaps notify emergency workers. To that end the father's home has been equipped with several XML based wireless sensors and actuators, amongst others an accelerometer in the floor of his home to detect a strong collision of his body with the floor; and an infrared movement sensor to detect the presence or absence of movement; a speakerphone that can also be used as separate speaker or microphone. All sensors will relay XML data to a coordinating function on a home server via the home network whilst all actuators can accept Web Service calls from it.

On event of a shock wave detected by the accelerometer, it will pass the event to the home server that will gauge the amplitude of the signal. To prevent false positives, the home server will interrogate the infrared sensor for any presence of movement during the last 3 seconds and absence of movement for the upcoming 3 seconds. In case of a positive indication, it initiates a countdown mechanism and activates a speaker using Web Services which audibly requests (by voiceXML) Mr. Dent to press a button on a panel on the wall of his room if he is alright. If he fails to do it soon enough, the system will notify his eldest son in France by setting up a telephone call between his landline phone and Mr. Dent's speakerphone (using Web Services to a call-me-now service of his telecom-provider), or, in absence of him answering the phone, sending an SMS-message (using Web Services and XML) to his mobile phone. The eldest son can also authorize the home system to call local emergency workers to tend to his father's injuries at his home.

This scenario can be extended to include detection of spasms or, along different dimensions, to support emergency workers providing care to the victim, or orchestrate administrative processes with health insurance companies. And the nice thing is that all devices mentioned originate from different vendors, were entirely interoper-

able, and were all chosen for their good price/value rate.

Conclusion

Increased personal independence, prolonged active participation in society, friendly access to public information, and integrated leisure and care processes for the ageing population, call for a number of new ICT-based services. Successful service development to older adults when using XML for Plug-and-Play applications is enhanced when:

(i) A domain vocabulary for leisure and home care can be established between all stakeholders; this is a dynamic construct that will evolve as the project matures;

(ii) Investments to guarantee future interoperability are ensured by building loosely coupled systems;

(iii) Ontologies are created after a significant part of the required vocabulary has been created and vetted; and applied only when the need for reasoning or manipulation of taxonomies arises;

(iv) An independent architect supervises the complete development life cycle and ensures the proper guidelines are followed;

(v) Chosen vendors must: (a) Avoid proprietary technologies, (b) Choose only technologies based on open standards to be implemented; (c) Use TCP/IP for data communication, where possible in combination with XML; (d) Avoid using 'improved' versions of an open standard and thus avoid vendor lock-in; and (e) Not be allowed to design the architecture of a service delivery system.

Following these guidelines will result in the optimal use of vendor-independent plug-and-play solutions for teleservices so urgently needed by our ageing society.

References

1. European Commission. ICT – Information and communication technologies. Work programme 2007-08. Challenge 7: ICT for Independent Living and Inclusion. Released December 22, 2006; http://cordis.europa.eu/fp7/ict/programme/challenge7_en.html; retrieved March 15, 2007
2. Kearns WD, Fozard JL. High-speed networking and embedded gerontechnologies. Gerontechnology 2007;6(3):135-146.
3. Cerf VG, Kahn RE. A protocol for packet network interconnection. IEEE Transaction on Communications 1974;22(5):637-648
4. Bray T, Paoli J, Sperberg-McQueen CM. Extensible Markup Language (XML) 1.0; W3C Recommendation 10-February-1998; http://www.w3.org/TR/1998/REC-xml-19980210; retrieved March 15, 2007
5. Internet Society. Internet Standards Organizations; http://www.isoc.org/standards/orgs.shtml; retrieved March 15, 2007
6. Uniforum Association. Internet Standards Organizations; http://www.uniforum.org/publications/ufm/mar96/sidebar.html; retrieved March 15, 2007
7. Knox RE, Wagner R, Linden A, Abrams C, Andrews W, Friedman T, Feinberg D, Harris-Ferrante K, Hostmann B, Logan D, Valdes R, Bell T, Fabbi M, Knox M, Schulte RW, Schlegel K. Hype Cycle for XML Technologies. Stanford: Gartner; 2005
8. Adams D. The hitchhiker's guide to the Galaxy. New York: Ballantine; 1995
9. Smith B, Ceusters W. HL7 RIM: An Incoherent Standard. Studies in Health Technology and Informatics 2006;124(August):133-138
10. Health Level 7. HL7 ANSI-Approved Standards; www.hl7.org; retrieved March 15, 2007
11. Ingram D, Schloeffel P, Heard S, Kalra D, Lloyd D, Beale T. Introducing openEHR. The openEHR Foundation; version 1.1 02 Jan 2006; http://svn.openehr.org/specification/TRUNK/publishing/openEHR/introducing_openEHR.pdf; retrieved June 19, 2007
12. ASTM Subcommittee E31.28. WK4363 Standard Specification for the Continuity of Care Record (CCR); http://www.astm.org; retrieved March 15, 2007
13. Cover R, editor. Cover Pages – XML in Clinical Research and Healthcare Industries. Organization for the Advancement of Structured Information Standards (OASIS); http://xml.coverpages.org/healthcare; retrieved March 15, 2007
14. Berners-Lee T. Business Model for the

Semantic Web. October 2001; http://www.w3.org/DesignIssues/Business; retrieved March 15, 2007
15. Palmer Sean B. The Semantic Web: An Introduction. September 2001; http://infomesh.net/2001/swintro/; retrieved March 15, 2007
16. Aerts ATM, Goossenaerts JBM, Hammer DK, Wortmann JC. Architectures in context: on the evolution of business, application software, and ICT platform architectures. Information & Management 2004;41(6):781-794
17. Baars C. SOA/ESB The Business-IT, Cibit | Serc, presentation held at the SOA seminar, Utrecht, the Netherlands, October 2005
18. Jaakkola L. Applying Service-Oriented Architecture to Geographically Distributed Industrial Information Systems, Master thesis, Helsinki University of Technology; 2005
19. Holtkamp M. Public Services Integration & Interoperability, Master thesis, University of Twente; 2006; http://asna.ewi.utwente.nl/education/Student%20assignments/completed%20bachelor%20and%20master%20assignments/holtkamp.pdf; retrieved March 15, 2007
20. innoQ Deutschland GmbH. Web Services Standards as of Q1 2007; http://www.innoq.com/resources/ws-standards-poster/; retrieved March 15, 2007