Incorporating interoperability into a distributed eLearning system

Timothy Read, Felisa Verdejo and Beatriz Barros
Departamento de Lenguajes y Sistemas Informáticos, UNED, Madrid, Spain[1]
{tread, felisa, bbarros}@lsi.uned.es

Abstract: In this article the incorporation of interoperability into a distributed eLearning system, namely the ActiveDocument platform, is considered. Three degrees of interoperability are defined for such a platform, namely syntactic, semantic, and functional. Subsequently, for the first two degrees of interoperability, a proposal is made for the way in which it can be obtained, backed up with details about the design and implementation of the ActiveDocument platform and its tools, which show how it has been (or is being) incorporated. The third degree of interoperability is still being studied and as such is not presented here. Finally, some conclusions are drawn for future work.

Defining the degrees of interoperability in distributed eLearning systems

Pedagogically speaking, it is generally accepted that learning is more effective when it is undertaken in a social context, i.e., when the students have to work together to consolidate aspects of the learning process. Traditionally, distance learning has placed the students at a disadvantage with respect to their classroom-based peers due to the frequent situation of isolation that it entails. eLearning systems, as briefly described here, encapsulate aspects of distributive systems in the sense that they involve (to a lesser or greater degree) interactive collaborative activities which require underlying cooperative systems that transparently provide the infrastructure needed to connect the students to the content servers and to each other, and integrate the different distributed tools and resources.

Heiler [3] defined interoperability among components of distributed systems as the ability to exchange services and data. Such interoperability can be argued to contain three distinct aspects: the structure of the data being used, the content of that data, and the protocols and techniques used to actually carry out the exchange. These three aspects can be broken down into three degrees of interoperability in a distributed system: syntactic, semantic, and functional. The first, **syntactic interoperability**, can be seen to be quite easy to achieve, and refers to interchanges where only the surface structure of the transferred data is important for coherent functionality of the distributed components, independently of any meaning. For such programs to actually work all that is needed is some form of simple common underlying representational format (e.g., data stored as strings of text) so that the programs know what to expect, but little else.

The second, **semantic interoperability**, as defined by Heiler, refers to exchanges of services and data where the exchanges 'make sense', i.e., where the requester and provider have a common understanding of the 'meaning' of requested services. She went on to note that semantic agreements are often lacking when new unanticipated applications of old data are made. In the problem domain in which this paper is situated, it is important that, as far as possible, application data be both syntactically and semantically interoperable in the sense that they can be shared between distinct types of computing devices that could be used by students taking part in an eLearning activity. The third and final type of interoperability, **functional interoperability**, refers to the ways in which the programs in the system can actually execute together in a coherent way, what protocols need to be established, and what functional configuration is used: should they always run in a standard client-server configuration or whether more flexibility is possible from a P2P[2] functional model.

The AD Platform [4] is a computational architecture for eLearning that enables collaborative learning activities (grounded in Activity Theory) that involve a variety of resources to be specified and undertaken. The platform is

[1]This work has been funded by EA_2C_2 CICYT TIC2001-007.

composed of three parts: firstly, a set of authoring tools for the creation and configuration of the different ADs required for the specification of the learning activities. Secondly, a repository of distributed learning objects that consists of a variety of tools and resources for the described activities. Thirdly, an architecture that generates the user environment necessary to carry out the described activities, together with the appropriate resources and tools. For a given learning scenario there are various ADs, which describe the activities to be carried out and related resources and tools, the structure of the groups of students that are taking part, and the outcome of the work for each student. It is the outcome AD which is the really 'active' part, in the sense that it grows with references to the activities and the results produced by each student as s/he progresses. Due to the nature of the problem domain being used here, where students will want to connect from different types of computing devices both to educational content in the AD platform and also to one another, all three degrees of interoperability are important.

Given these domain objectives, the distributive nature of the AD platform, and the fact that the client-server-based functionality of the AD platform has already been dealt with elsewhere [4], which touches on the subject of functional interoperability, in the subsequent sections, only syntactic and semantic interoperability will be discussed in this paper.

Syntactic interoperability in the ActiveDocument platform

XML is a meta-language for the creation of other languages that can be used to represent data, and is not as such a representational format in itself. Therefore, in the area of educational systems, as in any specialist domain, given the types of representations that are necessary for educational, essentially document-based content, work is being undertaken to establish a standard language that can be used for this purpose. So far no single language has appeared but several candidates are being worked on such as, for example, DC, IMS, IEEE Learning Objects Metadata [LOM], Educational Markup Language [EML] and Tutorial Markup Language [TML].

Syntactic interoperability can be obtained for data based in XML in two different ways. Firstly, the data can be returned to a requester where the existence of a common DTD permits it to be validated and subsequently used. Secondly, depending on the type of petition received, the provider can transparently transform the data into a different structure (without the petitioner realising that a different underlying XML-based format is being used) using an XSLT (XSL [eXtensible Stylesheet Language] Transformations) processor, and a suitable set of XSL rules. XSL is a sub-set of DSSSL (Document Style Semantics & Specification Language), which is an SGML-based standard for the presentation of marked-up documents in the Web.

The possibilities available for XSLT-based transformation are particularly useful for document-based information, which is very often the case in educational applications. The AD platform makes use of this transformational capability. A simple example of its application is where three students access the same fragment of course material in different ways. The first is identified by the input servlet as a user on a desktop PC with a standard Web browser. The requested educational content is retrieved from the database and transformed by the XSLT processor into XHTML and returned to the user. The second user is identified as connecting from a WAP browser (using a mobile phone or PDA). This time the same data is transformed into WML and returned via the WAP gateway. The third and final user sends an e-mail to request a hard copy of the material since s/he does not have access to a browser, whereupon the same underlying material is converted into PDF and returned to the user by e-mail. As can be appreciated, this is an example of how to achieve syntactic interoperability between the different components of the platform. No claim is made for any underlying semantic homogeneity since the information that is actually presented to a user in the scenario is in no way checked or semantically interpreted at any stage of the process.

Practical implications of syntactic interoperability in the ActiveDocument platform for eLearning and distance education

The architecture of the AD platform and the type of syntactic interoperability discussed here have implications for some real world limitations of eLearning when used in a traditional distance learning context. In some educational institutions where distance learning is undertaken, only eLearning is offered, i.e., only on-line versions of the courses are available to the students. Hence, if a student does not have a suitable computer and access to the Internet, then they cannot undertake the course. In the UNED eLearning is being introduced in a gradual way, and as such on-line access is not obligatory yet. Therefore, a scenario can be imagined where students with varying degrees of access to computing technology and the Internet undertake the same course. A combined eLearning and distance learning environment would oblige the teacher to prepare both on-line and off-line versions of the same course material. While it is true that the on-line version could be simply printed out from a Web browser and sent to students, as anyone who has done this will know, the structure of printed hypertext is not so clear when it is 'flattened' onto paper format. The AD architecture attempts to solve this problem by contemplating access to the educational content at four discrete levels (paper-based materials, PC with CDROM, PC with Internet access, and non PC-based computing technology such as PDAs and TV set-top boxes). In each case, the syntactic interoperability of the underlying data and associated transformational technology of the platform permit a functionally reduced version of a particular level to be produced.

Designing semantic interoperability into the ActiveDocument platform

Reiterating the words of Heiler [3], it can be seen that achieving semantic interoperability in this type of problem domain is a difficult task, and one for which the objectives need to be clearly specified. In the case of the general document-based educational content, such interoperability is not necessary, since the presentation of the material to a student in a suitable browser (or in hard copy) is sufficient. However, since this domain is intended for far richer interactive and collaborative activities (than mere content browsing), involving a variety of distributed tools used to generate and handle XML data structures that are subsequently stored in the AD platform, it is particularly important that these results are reusable by other tools in the same domain.

In order to design semantic interoperability into a distributed system, it is necessary to anticipate the types of uses that the data undertaking a particular task will have and the tools to be used. In the case of the AD platform where different tools have been integrated, it is necessary to consider the semantic interoperability on a tool-by-tool, 'result type' by 'result type' basis. This process can be illustrated by considering a particular case in detail. The tool in question was originally designed as a visual collaborative annotation tool that could be used by a group of students to collectively study any image. An example of its use would be where students use it to study the spectrum of a chemical substance. Typically, in the first part of a virtual experiment the students work separately to obtain data about a chemical substance and then work together in small groups, using the results that they have gained so far, to analyse the composition of the spectrum in order to identify it. The students can highlight a part of the spectrum (a peak of the graph) by drawing a rectangle around it and then make assertions regarding it, which can take one of three forms: a comment, a question, or a hypothesis. As the contributions appear on the screen, the students can also make assertions about what their colleagues have added until they reach an agreement about the nature of the chemical substance under study.

This tool consists of a central server (completely separate from the AD platform) that manages the debates in progress and a set of Java applet clients with a graphical front-end. The internal representation used by the server, as would be expected, is a set of Java objects. When the integration of this tool with the AD platform was contemplated, it was necessary to consider how the current state of an ongoing discussions could be shared with it, in such a way that the student could access the data from other tools in a 'meaningful way', i.e., making appropriate use of the data in a transparent fashion. This consideration gave rise to the following classification:

1. For the original version of the graphical tool. As such, a conversion would be needed to form the Java objects that the server program requires from the underlying data model. The data represented in the objects can be identified in terms of the type of comment and its structural relation to others, as well as the identified zones.
2. For incorporation as an image that captures the overall collaborative process in a word processor, which the students would use at the end of the experiment both to document what has happened and to prepare a report.
3. For use with newer types of portable computing devices with limited graphical capabilities. In order to take part in a future discussion it may be necessary for the students to be mobile, i.e., be able to get up from their desks to gather data to validate comments and hypotheses made by their peers in real-time. In which case it may not be practical to have to return to use the PC-based version of the tool. This type of activity could be undertaken using some form of hand held computer such as a PDA or even an Internet enabled mobile phone.

In order to provide a version of the data for each of the three applications noted above, two things are necessary: firstly, an underlying data representation that is sufficiently rich to capture the semantic features of the problem domain; and secondly, software that is able to extract and convert data from the underlying model into the required format. Such software can be seen to belong to the class of semantic mediators (as defined by Degoulet et al. [1]; based upon Wiederhold's [5] notion of mediator). The underlying data model needs to be able to represent the structure of the collaborate process together with the set of image features that can be identified in a non-graphical representation (in the form of text). In the case of the example shown here, that of chemical analysis, the visual elements that need to be referenced are the peaks of the graph.

Since a reduced version of the Java development kit is available for handheld computing devices such as the Palm PDA (J2ME), it is possible to design a program with a simplified interface that would use the same underlying data model, but with a textual presentation instead of the graphical one (due to the interface limitations of such devices[2].), and an explicit textual representation of the visual features of the image. The requirement of being able to convert the graphical representation into an image for use in word processors can be achieved by using SVG (Scalable Vector Graphics), an XML language for representing graphical data, which in turn can be converted into any other image format, such as GIF or JPEG. Hence, the state of the discussion is represented as an underlying XML-based representation and the semantic mediator translates this data into (or out of) a particular output format depending on the type of client that is making the request. Each client makes a semantic interpretation of the same data in different but related ways. In the case of the client with the graphical interface, the data represents a hierarchical structure of related rectangles and text elements and their associated geometry characteristics. In the case of the PDA that requires a text only representation of the same domain, the data represents a list of identifiable elements and a related set of nested text elements. Finally, in the case of the client that requires a static image, the data represents an equivalent geometrical SVG version of the visual representation of the structure of the discussion presented in the graphical interface (where no notion of relation or hierarchy exists), which is subsequently converted into a standard image format.

Conclusions and future work

In this article eLearning has been presented as a problem domain for distributed systems. The interoperability of the components in such a system is important not only for its current functionality (permitting a flexible interconnection of programs and sharing of data and services) but also to permit new types of Internet-enabled computer devices to be connected. Three degrees of interoperability have been identified in the AD platform, namely syntactic, semantic and functional. A definition of each type has been provided together with a presentation of how the first two types have been incorporated into this platform. In each case, the AD platform or some related tool was presented to illustrate engineering questions regarding design and implementation.

Maintaining syntactic interoperability as new tools (with new requirements) are incorporated into the platform appears to be quite straightforward (undertaken in terms of the dynamic structural rearrangement of data using XSLT-based technologies). Semantic interoperability is maintained as long as new tools do not place requirements on system data that have not been considered previously! To attempt to limit this problem, future work is needed to undertake a semantic categorisation of the system data in an extensive way, such that new tools and requirements placed on the application of the AD platform can be subsumed by mapping them onto the types of data that are available, and how they can be combined, transformed, or interpreted. Finally, the incorporation of functional interoperability has only just begun, and as such is not detailed in this paper.

References

1. Degoulet, P., Sauquet, D., Jaulent, M.C., Zapletal, E. & Lavril, M. "Semantic interoperability in health information systems". In *Proceedings of the IMIA WG 6 Conference on Natural Language and Medical Concept Representation.* Jacksonville, Florida (1997)
2. Graham, R.L. *Peer-to-Peer: Toward a Definition*. http://www.ida.liu.se/conferences/p2p/p2p2001/p2pwhatis.html (2001)
3. Heiler, S. "Semantic Interoperability". ACM Computing Surveys, vol. 27, no. 2 (1995)
4. Verdejo, M.F., Barros, B., Read, T. & Rodriguez-Artacho, M. "A system for the specification and development of an environment for distributed CSCL scenarios". In S.A. Cerri, G. Gouardères, F. Paraguaçu (eds.) *Intelligent Tutoring Systems*. Berlin: Springer Verlag (2002)
5. Wiederhold, G. "Mediators in the Architecture of Future Information Systems". *IEEE Computer*, March (1992)

[2] It is possible to use a fully graphical representation as in the applet, but the result is very difficult to see.