telcert Assuring interoperability in technology-enhanced learning

Learning Technology & TELCERT

Technology & Market Needs

TELCERT was borne out of longstanding market needs: from learning technologists, standards communities and researchers and providers of services to support a elusive goal – interoperability of content, services and learning systems that could be assured from the start

The use of technology to support learning has advanced significantly in recent years, both in terms of the increasing sophistication of online learning services and the availability of open specifications which have brought about convergence in the techniques used by vendors to achieve systems integration and in particular content portability. There are many products which offer interpretations of the specifications, but there are currently no independent, third-party mechanisms for assessing conformance to a common interpretation of the specifications, or a widely acknowledged reference implementation against which people can compare their efforts. Implementations often implement differing subsets of the specification, add extensions, use alternate bindings (e.g. Resource Description Framework, versus XML schema) or are implemented upon differing underlying technologies (e.g. JAVA versus .Net, SOAP, …).

Barriers to Interoperability

There are costly drawbacks for vendors and customers in the current situation:

- Vendors have to design and implement their own test suites, usually concentrating on the specific ways in which they interpret and use the open specifications.
- Customers setting up learning management systems are faced with the need that minor incompatibilities between systems from different vendors add up, have to be analysed and must be overcome by developing case-specific mediation software.

The Advanced Distributed Learning Initiative has piloted testing with their SCORM test system, but this addresses a single profile of the specifications, aimed at meeting the training needs of the US military. As is evident by the growing number of implementations around the world, online learning is not a 'one size fits all' undertaking. Both vendors and users seek reassurance that the desired interoperability between implementations can be achieved, such as that provided through the scrutiny of a standard conformance test suite. Conformance testing needs to be allied with the capability to adequately service localisation issues required to meet the full requirements of user communities. This is of special importance in Europe with its large diversity of culture and learning settings.

These issues are compounded by the fact that new technologies and methodologies are gaining popularity amongst implementers, and these in and of themselves raise challenges for any conformance testing which might be put in place in this domain.

TELCERT Objectives

Thus the Scientific and Technical objectives of the TELCERT project are:

1. Communicate the novel techniques and technologies (e.g. JAVA, UML, XML and new code/API stub generation tools) being adopted for implementation of technology enhanced learning products and services
2. Raise awareness of how to define community requirements in a consistent manner using Application Profiling techniques
3. Demonstrate how state-of-the art research into testing can be exploited to define flexible conformance testing strategies, able to accommodate application and data localisation
4. Assist the content industry and tool vendors in supporting Application Profiles in their commercial offerings which will successfully pass the conformance tests
5. Execute a coordinated plan of dissemination activities to aid European and International adoption of the project results
6. Achieve post-project adoption of the conformance testing developed and the Application Profiles this represents

XML, UML and Standards Development

It is established practice by Learning Technology Specification and standardization organizations to develop a specification/standard by:

- Defining an appropriate 'Information Model' that describes the core data objects and their properties;
- Defining the binding of the information model onto an appropriate data representation. This typically takes the form of XML (Extensible Mark-up Language) and, to a lesser extent, RDF (Resource Description Format);
- Describing the recommended practice and implementation when using the binding to support a particular application.

The state-of-the-art followed by organizations such as IMS is to describe the Information Model using the Unified Modelling Language (UML); UML is the internationally recognized object-oriented software system design methodology. The usage of UML allows a specification to be developed that includes a behavioural description as well as just the data model as normally expressed in an XML-based specification. UML support tools such as Rational Rose, TogetherJ and Poseidon allow a UML description to be saved in various forms including JAVA. This means that an appropriately constructed UML-based specification can be readily exported as an Application Programming Interface (API). Most UML tools use the XML Metadata Interchange (XMI) as their interchange format, thereby enabling different tools to edit a specification. There is a growing base of knowledge on how best to represent the details of a UML description in XML and XML-based formats and XML is at the core of most Learning Technology Specifications.

The adoption of behavioural descriptions is particularly significant because it enables the creation of robust Conformance Specifications. This in turn allows the development of a clear test strategy and the enforcement of compliance requirements. The conversion of a UML description to XML requires the mapping of the class operations onto the corresponding response/request messages. These messages are themselves XML documents that are exchanged using an appropriate messaging infrastructure e.g.

SOAP+HTTP/HTTPS. The state-of-the-art is now focused on using WSDL (Web Services Description Language) as the binding of a specification's information model to a web services application.

In most learning systems, several specifications/standards have to be combined to support a particular application. Even in the case of behavioural-oriented specifications/standards this requires the tailoring of the specifications to the particular application domain; the result is called an 'Application Profile' (AP). An AP enables the intrinsic flexibility of a specification/standard to be removed to fit a particular domain; Learning Technology Specifications/standards are designed to support any practice, which make it impossible to construct a suitably robust Conformance Specification for compliance. Therefore, an AP is crucial for the creation of a useful and realizable Conformance Specification.

Technology Challenges for TELCERT

- How should a Learning Technology Specification best be represented in UML to facilitate the derivation of the corresponding Conformance Specification? This assumes specifications that are based upon behavioural descriptions and/or data models;
- What is the best mechanism for the tool-based translation of a UML description into its equivalent XML, WSDL and JAVA bindings? This enables the support of multiple APIs using a single Information Model;
- How should a Learning Technology Specification be constructed to facilitate the preparation of an Application Profile and the associated Conformance Specification? This requires the development of a core specification from which the Application Profiles are derived;
- What is the best approach to minimize the number of Application Profiles created, for example, due to geographic localisations? It may be possible to identify and agree a core Application Profile or at least maximize reuse of elements across APs;
- What is the best way to construct content for an Application Profile that can be used to test the degree of conformance of a system to a particular Conformance Specification and how easy is it to re-author that content to support a different Application Profile.

Technology Options & Directions

Prior experimentation in implementing Content Packaging Tools has demonstrated that any such tool, in the context of the TELCERT development, would have to be capable of being configured on-the-fly to (1) a given user-community specification AP, (2) user interface language and (3) backwards support of previous release versions of the specifications included in a given profile. Significant flexibility and ease of modification by both content creators and test administrators is therefore required.

A candidate approach offering the capability to support this diversity is to provide a Content Re-Engineering Tool with a common core engine that supports a DOM, with XML import and export, local persistence and remote communications, with a defined API for interfacing with a plug-in editor. The plug-in editors are driven by the relevant XML Schema, together with a set of additional XML 'helper' files that support the editor UI and map the UI elements to the appropriate DOM elements. However, there is a real need for robust testing against diverse profiles (supported specifications, language, vocabulary substitution, element sub-sets) in order to validate this approach.

Providing a reference implementation of such a tool, mirroring the test system, will significantly accelerate the process of getting tool developers up to speed with developing products that will successfully complete conformance testing.

Existing test methodologies, such as are described in ISO 9646 (OSI Conformance Testing Methodology and Framework) and IEEE 1003.3 (POSIX test methods) address testing of protocols and APIs. ISO 9646 also includes a formal, processable test notation (TTCN3) which has been extended to address object testing. These standards underlie the tests that industry has developed for UNIX, Bluetooth, GSM, and other ETSI standards, WAP and many others. But very little has been done in testing XML content and processors beyond the core W3C specifications.

The test suites developed for WAP Forum and the recent Open GIS test engine both developed by The Open Group include elements that may be applicable to the learning technology test problem as they are dealing with XML based specifications as well as providing the test execution infrastructure necessary to support conformance testing.

Test tools must of course support all of the operational requirements for conformance testing, and must also deal with the requirement for testing to Application Profiles that may be unique to a given community. Rather than expect those communities to retain the skills needed to implement tests specific to their needs, the test system comprised of the test tools developed in this project must be able automatically to adapt and extend their test coverage by processing the Application Profile descriptions.

The already mentioned UML is becoming the de facto industrial standard for software system modelling and documentation, while the recent massive initiative of the OMG to Model Driven Architecture pursues a complete separation between the abstract specification of system functionality and that of its implementation on a specific technology platform. This separation between the abstract system specification and an implementation for a given platform is central to the approach defined in TELCERT for supporting multiple Application Profiles against a given set of Learning Technology Specifications.

Innovation in Software Testing

This project will also provide a significant twist to research in software testing and validation. Nowadays huge attention is attracted by the use of explicit models in software development, as opposed to implicit or undefined models using more traditional software design approaches. After having been almost ignored for decades by practitioners, rigorous specification and design methods finally enter the industrial laboratories as a strategic means for developing higher quality and more reliable systems. Considering the adopted modelling notations, key features for exploration are:

- formality, so that vital system properties can be automatically verified, using, e.g., model-checking techniques
- provisioning for adequate abstraction and composition mechanisms, so that complex, open and evolving systems can be effectively described
- standardisation, to counteract the proliferation of dialects and to favour interoperable interchangeable models.