CORBA-based Runtime Environments for Standardized Distributed Learning Architectures

Luis Anido Rifón
University of Vigo, Spain
lanido@ait.uvigo.es

Martín Llamas Nistal
University of Vigo, Spain
martin@ait.uvigo.es

Manuel J. Fernández Iglesias
University of Vigo, Spain
manolo@ait.uvigo.es

Manuel Caeiro Rodríguez
University of Vigo, Spain
mcaeiro@ait.uvigo.es

Juan M. Santos Gago
University of Vigo, Spain
jsgago@ait.uvigo.es

Judith Rodríguez Estévez
University of Vigo, Spain
jestevez@ait.uvigo.es

Abstract

The learning technology standardization process is taking the lead role in the research efforts into computer-based education. Institutions like the IEEE or the US Department of Defense have set up committees to deliver recommendations and specifications on this area to provide interoperability between different educational systems. The first part of this paper shows an up-to-date survey on this field. In the second part we present our contribution to this area: a distributed architecture to develop interoperable educational frameworks over a CORBA domain interface. Our system aims at the standardization the development process of distributed educational environments from reusable software components. We focus our attention on the runtime environment, which is responsible for contents delivering, student tracking and course routing.

Keywords: Learning Technology Standardization, Architectures for Distributed Learning, CORBA, Software Interfaces,

Introduction

The increasing use of the Internet and its technological capabilities allowed a high number of Internet-based distance learning platforms to show up. As they are usually developed ad-hoc to meet the requirements of a particular institution, heterogeneous systems appear with no interoperability mechanism among them. There exist important efforts in the learning technologies standardization process leaded by several institutions and projects. Their main aim is to contribute to the definition of standards on learning data and metadata and recommendations for the development of software architectures devoted to computer-based education. The first part of this paper presents an up-to-date survey on the learning technologies standardization process.

In addition, most distance learning systems share some common functionality usually implemented from the scratch by each of them. The existence of reusable software elements that implemented that functionality in a generic way would drastically reduce the time needed to develop a new computer-based educational system. The open definition of the interfaces provided by each component contributes to the interoperability among different systems and the standardization process of learning technologies. Our work is mainly focused on this field. The second part of this paper is devoted to the presentation of a CORBA (OMG, 2001c) domain interface to develop scalable learning platforms making use of agreed recommendations on learning resources.

A Survey on the learning technologies standardization

Much work has been done and is being done in the learning technologies standardization area. Among the main contributors to this effort let us mention the IEEE's Learning Technologies Standardization Committee (LTSC, 2001), the IMS Global Learning Consortium (IMS, 2001), the Aviation Industry CBT Committee (AICC, 2001), the US Department of Defense's Advanced Distributed Learning initiative (ADL, 2001), the Alliance of Remote Instructional Authoring and Distribution Networks for Europe project (ARIADNE, 2001), Getting Educational Systems Talking Across Leading Edge Technologies (GESTALT, 2001), the PROmoting Multimedia access to Education and Training in EUropean Society (PROMETEUS, 2001), the European Committee for Standardization Information Society Standardization System, Learning Technologies Workshop (CEN/ISSS/LT, 2001) and the Gateway to Educational Materials project (GEM, 2001).

The IEEE's LTSC is the institution that is actually gathering recommendations and proposals from other learning standardization institutions and projects. Specifications that have been approved by the IEEE go through a more rigorous process to become ANSI or ISO standards. In fact, a new ISO/IEC JTC1 Standards Committee for Learning Technologies, SC36 (ISO/IEC JTC1 SC36, 2001), was approved in November 1999. Below we present the main outcomes of these standardization efforts obtained so far. The most outstanding results are from metadata for learning resources, definitions of learner records and profiles, formats for course structures and packages, formats for questions and tests and definitions of learning architectures and run time environments. This standardization is being developed at the time of this writing and, therefore, this survey may encourage the reader to contribute to this process from his/her own experience.

Metadata for Learning Resources

The learning metadata definition area has been one of the main focuses for the learning standardization community during the last few years. Metadata is just data about data, in this case, data about educational data and resources. The purposes of these definitions are, among others: to allow humans to search, evaluate, acquire and use learning objects, to enable sharing and exchanging of learning objects across any technology-supported learning system, let computer agents to automatically and dynamically compose personalized lessons for an individual learner, to enable educational institutions to express educational content and performance standards in a standardized format that is independent of the content itself. In short, it aims at the standardization of learning resources description.

Important outcomes have been already delivered. One of the main contributors to this effort is the IEEE LTSC's Learning Objects Metadata (LOM) working group. The LOM specification (Hodgins, 2001), version 6 from February 2001, describes learning content cataloguing information. It specifies the syntax and semantics of learning object metadata, defined as the attributes required to fully and adequately describe a learning object. Relevant properties of learning objects include type of object, author, owner, terms of distribution, format, teaching or interaction style, grade, level, mastery level and prerequisites. The structured approach to metadata definition implies that the actual data elements of a learning resource are grouped into meaningful categories. The base LOM scheme consists of nine such categories: General, Lifecycle, Meta-metadata, Technical, Educational, Rights, Relation, Annotation and Classification.

LOM metadata is becoming a standard de-facto among the learner community. However, this specification is the result of the effort of many contributors, among them, the European ARIADNE project and the IMS project stand out. ARIADNE uses LOM version 3.8, to which it contributed significantly, to index and exploit its network of interconnected knowledge pools (KPS). The IMS Learning Resources Metadata Specifications (Anderson, 2000) is directly based on the IEEE's LOM with some changes based on implementation testing and detailed document reviews by the IMS Technical Board, which will probably be incorporated into the IEEE specifications. The IMS metadata specification identifies a minimum set of IEEE metadata elements called the IMS core (19 out of 86 LOM elements). The remaining IEEE metadata elements form the IMS Standard Extension Library, SEL, (67 out of 86 LOM elements). The IMS has also completed a survey to identify taxonomies and vocabularies, which can be used as values for the defined metadata elements.

The DoD's ADL Sharable Courseware Object Reference Model (SCORM) (Dodds, 2001), January 2001, applies the IEEE/IMS definitions to the three components of the SCORM model: raw media, content and courses. It provides the link between general specifications and specific content model. Other system that has extended the LOM definition is GESTALT that delivered its own metadata specification: GEMSTONES (Foster, 2000). The main extensions of LOM include external rights management and the improvement of the quality of service description. GEMSTONES are used by the GESTALT brokerage service to locate learning resources. The Gateway to Educational Materials (GEM) system also provides a brokerage service based on extensions of Dublin Core (DC, 2001) metadata.

Learner records and profiles

Description of learner's profiles and records has also been studied in order to deliver recommendations on standards that allow the exchange of student data. The IEEE LTSC's Public and Private Information (PAPI) specification (Farance, 2000), describes portable and implementation-independent learner records. Learner records are organized into four major categories: personal, preference, performance and portfolio information. They describe information about the learner, about his/her technical, learning and physical preferences, about learner's history and about his/her current works. A particular file format to store student performance data was also defined by the AICC as part of its guidelines for interoperability (Hyde, 2000).

Based on PAPI, the IMS Enterprise Data Model Specification (Collier, 2000) is aimed at administrative services that need to share data about learners, courses, performance, etc., across platforms, Operating Systems, user interfaces and so on. This data model is supported through the use of three data objects: person, group and group membership. The person object contains elements describing an individual of interest to the learning environment. The group object contains elements like a course instance, training programs, academic programs, clubs, courses, etc. The group membership object contains elements describing the membership of a person or group in a group. Group members may be instructors, learners, content developers, managers, mentors or administrators.

Course Structure Formats

The US Department of Defense ADL initiative, as part of its SCORM model, has identified an XML-based representation of a course structure format (CSF) that can be used to define all course elements, structure and external references necessary to move a course from one system to another. It is not course packaging, as the course structure format is just one (albeit very important) of the elements needed to move a course from a given system to another one. CSF describes a course using three groups of information. The first group, called globalProperties, is the data about the overall course. The second, called block, defines the structure of the course, and the third group, objectives, defines a separate structure for learning objectives with references to course elements within the assignment structure. Course sequencing is defined using prerequisites and completion requirements for blocks, assignable units and objectives.

ADL's CSF is derived from the AICC content model for course structures, properties, and objectives. The AICC in its guidelines for interoperability (Hyde, 2000), has described a file for the basic data on the structure of a course. It includes all of the assignable units and blocks in the course. Its order in the file implies (but does not force) an order for presentation to the student. A personalized order is allowed using a fully specified table of prerequisites.

Course Packaging

The IMS project is leading the standardization process in this particular field. The IMS Content & Packaging Specification (Anderson, 2000) makes it easier to create reusable learning resources. The key element is the package: an abstract description of a unit of reusable content. A package must provide all files and data needed to transfer the learning resources it embodies from one system to another. It is also possible to aggregate a package into a higher level one. The two components of a package are the manifest file and the physical resources. The manifest contains a metadata description of the package as a whole, one or more ways of organization of the content, and can include or reference sub-manifests which describe the packages that have been included or referenced. The physical resources are a collection of resources physically included within a package.

Questions and Tests Interoperability

In February 2000, the IMS project has delivered the first specification on questions and test interoperability (Smythe, 2000). It addresses the need to share test items and other assessment tools across different systems, and describes the data structures needed to provide interoperability between questions and test systems, particularly those that are Internet-based. The key elements are assessments (basic test units), sections (containers for sections and items with a common testing objective) and items (the fundamental self-contained question/response block). The specification defines a taxonomy that describes a set of response types and different forms for each of them.

Learning Architectures and Run Time Environments

The IEEE LTSC's Architecture and Reference Model working group delivered the Learning Technology Systems Architecture (Farance, 1999), draft 5, in December 1999. The LTSA specification covers a wide range of learning systems. It is pedagogically neutral, content neutral, culturally neutral and platform neutral. Five refinement layers of architecture are specified, from the highest to the lowest levels as: learner and environment interactions, human-centered and pervasive features, system components (with four processes: deliverer, learning entity, evaluation, and coach; and two stores: learning resources and learner records), stakeholders perspectives and priorities and API's coding and protocols. They are applicable to a broad range of learning scenarios.

Regarding the specification of concrete run time environments, the work by the AICC (AICC, 2001) and the DoD's ADL (ADL, 2001) stand out. The AICC guidelines for interoperability of Computer-Managed Instruction (CMI) systems (Hyde, 2000) and the ADL's SCORM (Dodds, 2000) based on the AICC specifications, deal with a common problem: in the past, authoring systems made the customer a captive of his own CMI system. In order to avoid this dependence between CMIs and CBTs (contents), a standard approach is defined to allow a single CMI system to initiate lessons from different CBT vendors. To accomplish this function, CMI and CBT must communicate by means of standard types of data: data from CMI to CBT to start the lesson, data from a CBT system to CMI needed to record student performance and assign the next learning unit, and data needed for evaluation of a lesson such as item response data, simulation performance data, etc.

Additionally, the GESTALT project (GESTALT, 2001) identifies a run time architecture made up of components from previous ACTS projects (ACTS, 2001): GAIA, Renaissance and Prospect. The system architecture comprises a learning environment, an administration system, an asset management system, a service for user profiles and a resource discovery service. Business objects comprising the interfaces among the various software components within the GESTALT architecture are identified. Middleware solutions were used for this purpose. DCOM (DCOM, 2001) for interfaces among systems to be run in the same institution and CORBA (OMG, 2001c) for those interfaces among systems from different institutions.

A Corba Architecture for Computer-based Education

So far, the most outstanding results in the learning technologies standardization area are from learning metadata, definitions of learner records and models, and course structure formats. Standardization of the software that supports Computer-Managed Instruction or Learning Management Systems is more difficult to cope with. The next step should be to provide software that could be easily reused. Open interfaces would also allow interoperability at run time even among systems from different institutions, to share not only learning resources, but also learning services.

Our aim is to contribute in the field of learning technologies standardization and reuse in the run time environment area. We have defined a prospective CORBA domain interface to develop distributed learning environments using reusable elements instead of starting from scratch. These components have open interfaces that allow other objects, even from different systems, to access their services. In addition, the learning resources managed and delivered could be easily reused and localized as international format and metadata standards are used. Thus, interoperability both at business-logic and data level are achieved.

CORBA Middleware

CORBA (OMG, 2001c) is an object-based distributed architecture that allows distributed and heterogeneous applications to interoperate on a network. CORBA is a standard defined by more than 800 institutions that form the Object Management Group, OMG. Over a software bus, CORBA objects interact with each other via standard contracts written in the Interface Definition Language, IDL (OMG, 2001c). The OMG has also defined the interfaces for 15 basic services for distributed computing (OMG, 2001b). Nowadays, the hot topic for the OMG is the definition of high level services, clearly oriented to a particular business domain. Examples of already existing domain interfaces (OMG, 2001a) are in the areas of: Telecommunications, Manufacturing, Finance and Healthcare. They identify the objects needed, and their interfaces, in those domains to cover the needs of a wide range of requirements. Different vendors may change the implementation but the functionality is the same thanks to the agreed interface. Thus, CORBA domain interfaces contribute to the interoperability among systems. In addition, the OMG is an open consortium with no ties with particular software platforms or programming languages. Therefore, we chose CORBA as our middleware framework among other available options (e.g. DCOM (DCOM, 2001) or Enterprise Java Beans (EJB, 2001)).

Brokers for Learning Resources

An electronic broker is an entity that supplies to customers (students) specialized information about products or services (courseware) available from vendors (learning providers). We have identified the brokerage architecture shown in Figure 1. It is aimed at educational resources location, but it does not impose any particular restriction on this and therefore, it could be used for general brokerage purposes. We use well-known metadata formats like Dublin Core, LOM, IMS core and IMS SDL and GEMSTONES, which have been introduced in the first part of this paper. Their syntax has been defined as CORBA IDL interfaces. The core element of the architecture is the CORBA Broker that defines every needed method to locate learning resources described by the mentioned metadata formats. Search coordination is defined using the Search Engine Manager that sends queries to Search Engines specialized in each metadata format (these also make transparent the storage mechanism) and manages possible mappings among them. Thus, a user query using the IMS core could match resources whose description is stored using LOM format. The Learner Profiles Manager offers through its interface information about the learner's preferences (described using IEEE PAPI) that could be used by the Broker to customize the results sent back to the user. For example, they would only return those resources that can be efficiently managed by the student's equipment. These objects may run in the same computer or may communicate through a network using the CORBA IIOP protocol (OMG, 2001c).

Brokers can be federated to provide a global-wide scope to the searches. The Federation Manager defines all the methods needed to manage the search over a set of federated brokers. A simple example to illustrate this concept is shown in Figure 2. A client sends a location query to the Broker A through the CORBA bus (1). The Broker may decide to adapt query results to learner's preferences. It would request them (2 and 3) to the Learner Profiles Manager (if available). The search parameters would be sent to the Federation Manager (4) together with some additional parameters like the scope of the federated search and/or the desired maximum number of results. In this case the search is forwarded to Broker B (5.B) and Broker C (5.C). Search results (6.B and 6.C) are used by the Federation Manager to compose the global reply to be sent to Broker A (7). Federated searches and local search results filtered using the learner's preferences are used to compose the reply to the client (8).

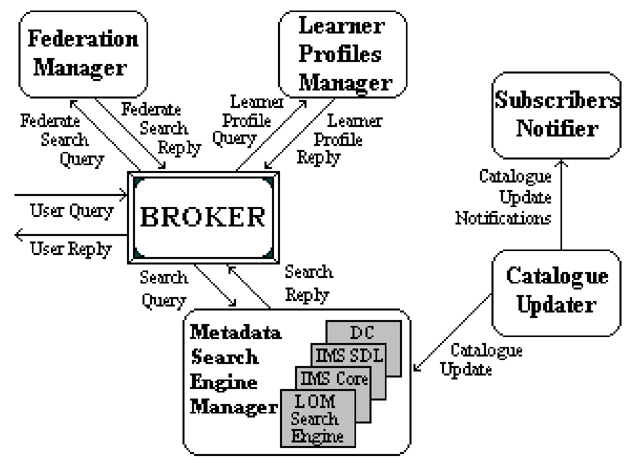

Figure 1: Brokerage Architecture for Learning Resources

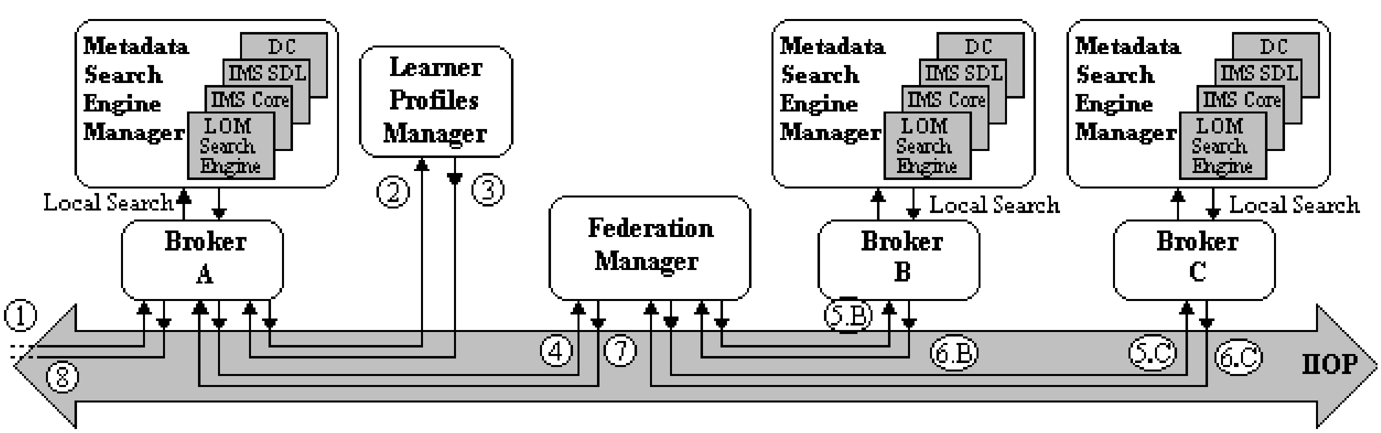

Figure 2: Using the CORBA IIOP protocol to provide a world wide scope for learning resources location

Distributed Learning Environments

The CORBA-based architecture for interoperable learning environments that we propose is outlined in Figure 3. Object interfaces are used as the basis for a computer-based education domain interface. Clients access learning services, through standard CORBA communication mechanisms, using the interfaces provided by Learning Server Objects. Two different interfaces have been defined: an AICC/ADL run time compliant interface and our Standard Run-time Interface (SRI). The latter defines standard mechanisms to interact with a learning environment (e.g. login, course index, next unit, previous unit, next assignment, etc.) and advanced features like predictive navigation (the object evaluates which learning resources are more likely to be requested next and sends them to the client side if their prerequisites were already fulfilled).

Navigation Manager objects could impose particular navigation sequences based on designer criteria and/or students' performance. So far, we defined interfaces for the AICC/ADL Course Structure Format (CSF) and a basic hierarchical structure with no prerequisites. Learning Server Objects use these interfaces to evaluate the next learning unit to be sent to a student, to check the prerequisites to access a unit or to resume student status in a particular course from stored data about previous learning sessions.

Students' performance tracking is done by Student Trackers who receive information from Learning Server Objects through Student Tracking channels based on the CORBA event service. Trackers are responsible for storing tracking information and providing later access to it. AICC-compliant and PAPI-compliant interfaces have been already defined. Performance Information at a higher level becomes grading information. Trackers report Administration objects whenever they detect students have fulfilled all needed requisites to pass a course. A CORBA event channel, Student Grading, is used to send this information from Trackers to Administration Objects, which update students' records. So far, administrative data are stored, accessed and managed using a PAPI-compliant interface. Learning Server Objects may use PAPI preference information to fit their behavior to each student.

The Learning Resources Repository Manager is responsible for making transparent for the rest of the system the particular storage mechanism. Both learning contents and metadata is stored by this object and accessed through it.

The IMS Packaging Adapter defines through its interface operations to create and aggregate learning packages using the IMS packaging specification. It also provides operations to extract packaged data.

The services discussed above are accessed following

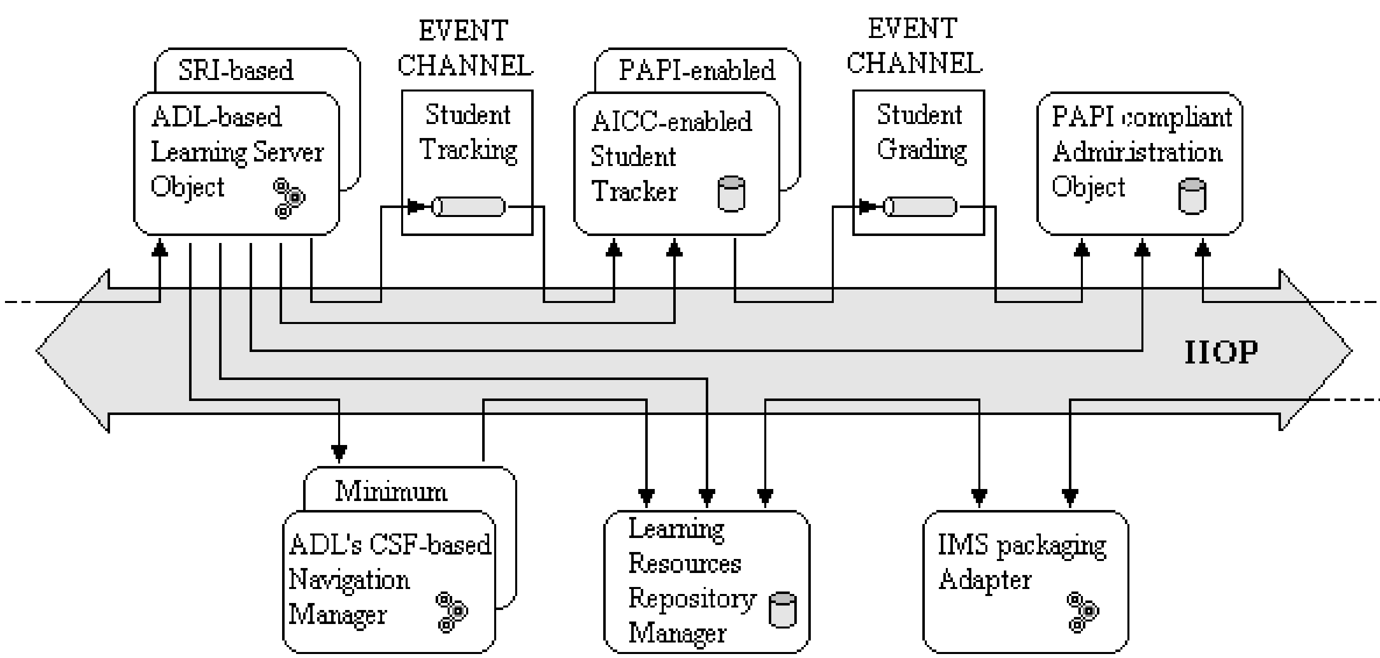

Figure 3: A CORBA-based architecture for interoperable computer-based learning environments

CORBA standard mechanisms. Defined objects do not impose restrictions regarding their physical location. They may run all of them in the same computer or distributed over a local network or over the Internet. Communication mechanisms are based on IIOP protocol that relies on TCP/IP standard protocol.

A CORBA Components Model definition

Defined objects could be implemented following the forthcoming CORBA Components Model (CCM) (OMG, 2001c) which is part of the CORBA 3.0 specification. The CCM extends the traditional CORBA object model by defining features and services that allow application programmers to implement, manage, configure and deploy software components that integrate commonly used CORBA services. The CCM standard not only enables greater software reuse but it also provides greater flexibility for dynamic configuration of CORBA applications. A component is a basic CORBA meta-type that can be referenced by multiple object references. Components can interact with external entities through the following port mechanisms:

- *Supported interfaces.* They inherit from other interfaces or components and are used by "component-unaware" clients.
- *Facets.* Also known as provided interfaces. They are unrelated interfaces that need to be related through inheritance and allow clients to navigate among them.
- *Receptacles.* They are used to specify the object connections among components and objects.
- *Event sources/sinks.* Components can express their interests to each other by supplying and consuming events through event sources and sinks.
- *Attributes.* Component attributes provide a standard mechanism for setting component states and are intended to be used by the CCM framework to configure the component.

As an example, let us show in Figure 4, a CCM proposal for the distributed learning architecture presented in Figure 3.

In this case, the same component offers different facets to provide the same functionality which was implemented by different objects in the traditional CORBA model. Communication among components is directly carried out through receptacles and event sources/sinks. These features, together with the deployment capabilities offered by the CCM seems to make it a suitable framework to develop this kind of distributed systems. Unfortunately, at the time of this writing no CORBA implementation supports the CCM.

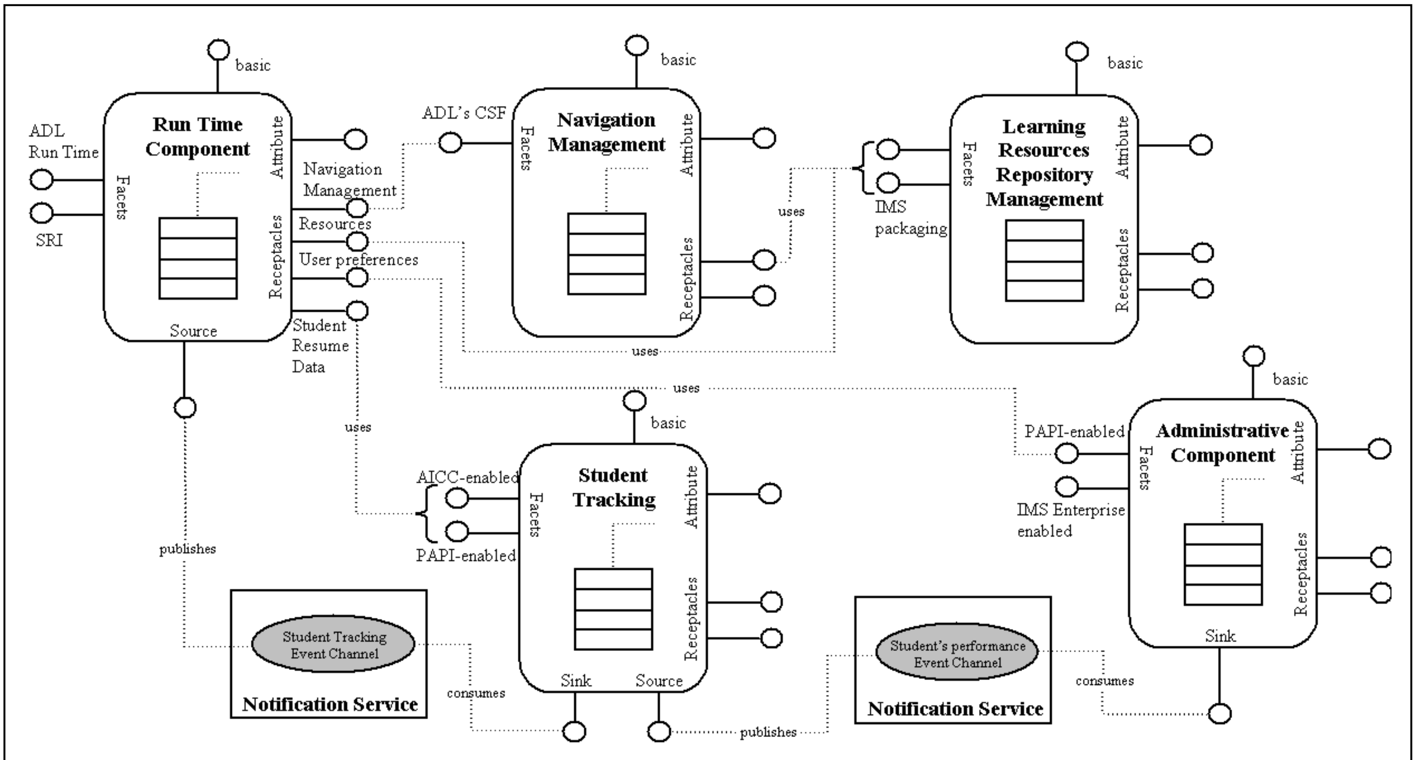

Figure 4: A CCM-based model to develop scalable and interoperable distributed learning environments

Learning Environment Services

First part of this paper introduces a generic description on our *Learning Environment* subsystem. The final UML (Jacobson, 1999) class diagram is depicted in Figure 5. Responsibilities are mainly divided into a *TrackingManager* object that follows student performance during a learning session, and a *NavigationManager* object that performs course routing according to a particular course structure format and student previous actions. These two objects work together to perform basic course routing and trading. *Learning Server Factory* creates a *Learning Server* object to manage each student that access a different course.

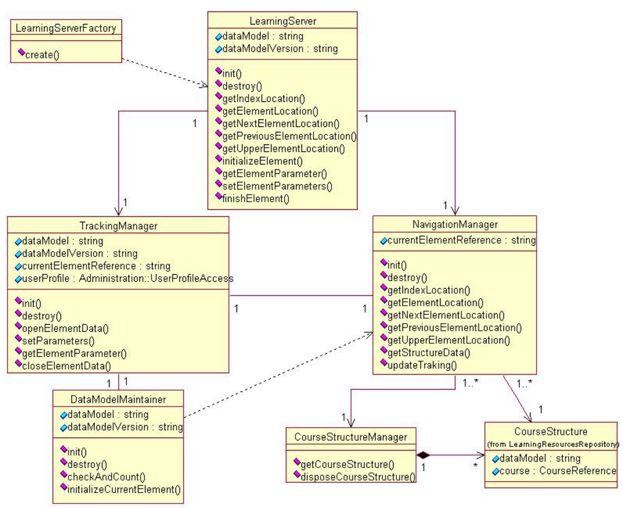

Figure 5: Learning Environment's UML Class Diagram

At the present moment four software engineers are specifying our architecture definition using IDL. Below we show part of the IDL specification for the *Learning Environment* component. Complete specifications are available from the authors on request. The eventual interface definition will come after the Technology Adoption Process defined by the OMG.

```
module LearningEnvironment {
  // Type Definitions
  typedef string URL;
  typedef
       LearningResourcesRepository::CourseReference
CourseReference;
  struct Trace {
    string name;
    string value;
  };
```

```
typedef sequence<Trace> TraceSeq;

  // Exceptions
  exception NoElementOpened {};
  exception ElementNotClosed {};
  exception NotFound {};
  exception DataModelError {
    long code;
    string description;
  };
  exception InvalidReference {};
  exception NoMoreElements {};

// Interfaces
  interface CourseStructureManager {
    struct StructureRecord {
      LearningResourcesRepository::CourseStructure struc-
ture;
      long useCount;
    };

    typedef sequence<StructureRecord> StructureRecordSeq;
       LearningResourcesRepository::CourseStructure get-
CourseStructure(
         in CourseReference reference);
       void disposeCourseStructure(in CourseReference refer-
ence);
    };
    ...
    ...

  interface TrackingManager {
    readonly attribute string dataModel;
    readonly attribute string dataModelVersion;
    readonly attribute string currentElementReference;
    readonly attribute Administration::UserProfileAccess
userProfile;
    void init(in Administration::UserProfileAccess user,
      in CourseReference course,
      in NavigationManager navigationReference);
    void destroy();
    void openElementData(in string elementReference)
      raises( ElementNotClosed );
    void setParameters(in TraceSeq paramSet, in boolean
check)
      raises( NoElementOpened, DataModelError );
    string getElementParameter(in string elementReference,
in string param)
      raises( NoElementOpened, DataModelError, NotFound
);
    void closeElementData()
      raises( NoElementOpened );
    };
```

```
interface NavigationManager {
    readonly attribute string currentElementReference;
    void init(in CourseReference course, in TrackingMan-
ager trackingReference);
    void destroy();
    URL getIndexLocation();
  URL getElementLocation(in string elementReference)
     raises( InvalidReference );
    URL getNextElementLocation()
     raises( NoMoreElements );
    URL getPreviousElementLocation()
     raises( NoMoreElements );
    URL getUpperElementLocation()
     raises( NoMoreElements );
    string getStructureData(in string elementReference, in
string path)
     raises( InvalidReference );
    void updateTraking();
  };
  ...
  ...
};
```

Defined services include common functionality for Learning Runtime environments. Developers of particular Web-based learning systems benefit from the offered services and their reuse. Thus, time to market is reduced. As an example of applicability, we developed a Web-based courseware system that conforms to the US Department of Defence ADL runtime model. This model is bound to be accepted by the learning technology standards community as the common way for launching and getting lesson information in a Web-based distance learning environment. For this, we just needed to develop a thin layer between the Web browser and the objects that composes system architecture. Interactions between them are presented in the UML interaction diagram included in Figure 6.

Conclusion

The CORBA architecture for computer-based learning presented in this paper allows the development and the deployment of distributed learning systems using reusable software elements. Scalability is fully provided because objects can be installed individually and later connected to

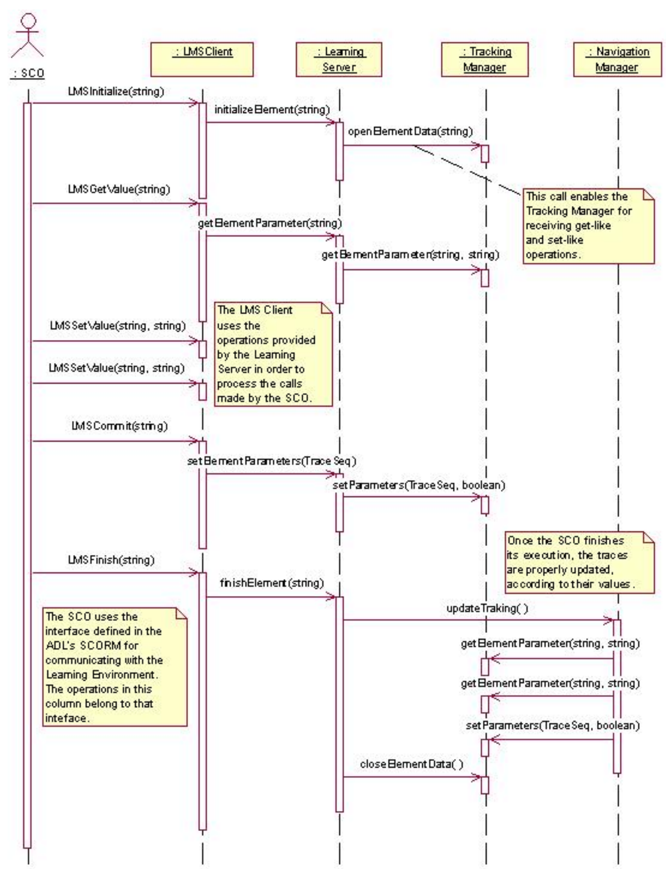

Figure 6: UML interaction diagram between an ADL-compliant Web-based Runtime environment a the Learning Environment

other components to provide a more refined functionality (e.g. simple or performance-based navigation, single or federated searches, etc.). Developers do not need to start their developments from scratch because defined objects are reusable. Therefore, the development and deployment process is improved and can be carry out quicker.

This paper presents object interfaces as open definitions of object behavior that, together with the use of internationally agreed formats for learning data and metadata (an up-to-date survey on this area appears in the first part of this paper) make our model a full interoperable framework for distributed computer-based learning systems. This leads to the possibility of using these interfaces as a new CORBA domain interface.

Acknowledgements

We want to thank the *Xunta de Galicia* for its support to this project under grant PGIDT00TIC3220PR (project *"Arquitecturas Distribuidas para Teleservicios"*)

References

ACTS (2001). Advanced Communicaction Technology and Services (ACTS) projects. Information on-line. Accessed March 9, 2001 at http://www.infowin.org/ACTS/

ADL (2001). Advanced Distributed Learning. Department of Defense (DoD). Information on-line. Accessed March 9, 2001 at http://www.adlnet.org

AICC (2001). Aviation Industry CBT Committee. Information on-line. Accessed March 9, 2001 at http://www.aicc.org

Anderson, T., & McKell, M. (2000). *IMS Content & Packaging Specification*, IMS Global Consortium Inc. Accessed March 9, 2001 at http://www.imsproject.org/content

Anderson, T., & Wason, T. (2000). *IMS Learning Resources Metadata Information Model*. IMS Global Consortium Inc. Accessed March 9, 2001 at http://www.imsproject.org/metadata

ARIADNE (2001). The Alliance of Remote Instructional Authoring and Distribution Networks for Europe. Information on-line. Accessed March 9, 2001 at http://ariadne.unil.ch

CEN/ISSS/LT (2001). European Committee for Standardization (CEN), Information Society Standardization Systems (ISSS), Learning Technologies Workshop (LT). Information on-line. Accessed March 9, 2001 at http://www.cenorm.be/isss/Workshop/lt

Collier, G., & Veres, W. (2000). *IMS Enterprise Information Model*, IMS Global Consortium Inc. Accessed March 9, 2001 at http://www.imsproject.org/enterprise

DC (2001). Dublin Core metadata. Information on-line. Accessed March 9, 2001 at http://purl.org/dc

DCOM (20001). Microsoft's Distributed COM (DCOM). Information on-line. Accessed March 9, 2001 at http://www.microsoft.com/com/tech/dcom.asp

Dodds, P. (2000). *SCORM, Sharable Content Object Reference Model*, ADL. Accessed March 9, 2001 at http://www.adlnet.org/Scorm/scorm_index.cfm

EJB (2001). Enterprise Java Beans. Information on-line. Accessed March 9, 2001 at http://java.sun.com/products/ejb/

Farance, F., & Tonkel, J. (1999). *Learning Technology Systems Architecture (LTSA) Specification. Draft 5*, Farance Inc., Edutool Division. Accessed March 9, 2001 at http://ltsc.ieee.org/doc/wg1/ltsa/ltsa_05.pdf

Farance, F. (2000). Draft Standard for Learning Technology. Public and Private Information (PAPI) for Learners, IEEE Inc. Accessed March 9, 2001 http://edutool.com/papi

Foster, P., Kraner, M., & Graziano, A. (2000). *Courseware Metadata Design. Gestalt Extensions to Metadata Standards for ON-line Education Systems (GEMSTONES)*, British Telecommunications. Accessed March 9, 2001 at http://www.fdgroup.co.uk/gestalt/metadata.html

GEM (2001). Gateway to Educational Materials project. Information on-line. Accessed March 9, 2001 at http://www.geminfo.org

GESTALT (2001). Getting Educational Systems Talk Across the Leading-edge Technologies. Information on-line. Accessed March 9, 2001 at http://www.fdgroup.co.uk/gestalt

Hodgins, W. (2001). *Draft Standard for Learning Object Metadata (LOM)*. Working Draft 6.0. IEEE, Inc. Accessed March 9, 2001 at http://ltsc.ieee.org/wg12/LOM_WD6.pdf

Hyde, J. (2000). *CMI Guidelines for Interoperability. Revision 3.4*, AICC CMI Subcommittee. Accessed March 9, 2001 at http://www.aicc.org/docs/tech/cmi001v3-4.pdf

ISO/IEC JTC1 SC36 (2001). International Standardization Organization (ISO), Institute Electrotechnical Commision (IEC), Learning Technologies Subcommittee. Information on-line. Accessed March 9, 2001 at http://www.jtc1sc36.org

IMS (2001). IMS Global Learning Consortium. Information on-line. Accessed March 9, 2001 at http://www.imsproject.org

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Modeling Language User Guide*. Addison-Wesley Longman.

LTSC (2001). Learning Technologies Standardization Committee. Information on-line. Accessed March 9, 2001 at http://ltsc.ieee.org

OMG (2001a). CORBA domain interfaces. Information on-line. Accessed March 9, 2001 at http://www.omg.org/library/cdomain.html

OMG (2001b). CORBA services. Information on-line. Accessed March 9, 2001 at http://www.omg.org/library/csindex.html

OMG (2001c). *The Common Object Request Broker (CORBA): Architecture and Specification*. OMG Group. Accessed March 9, 2001 at ftp://ftp.omg.org/pub/docs/formal/01-02-33.pdf

PROMETEUS (2001). PROMoting Multimedia access to Education and Training in European Society. Information on-line. Accessed March 9, 2001 at http://prometeus.org

Smythe, C., & Shepherd, E. (2000). *IMS Question & Test Specification*, IMS Global Consortium Inc. Accessed March 9, 2001 at http://www.imsproject.org/question

Biographies

Luis Anido received the Telecommunication Engineering (1997) degree from the University of Vigo with Honors by the Spanish Department of Science and Education and by the Galician Regional Government. He joined the Telecommunication Engineering faculty of the University of Vigo where he is working towards his PhD. His main interests are in the field of New Information Technologies applied to distance learning and in the area of Object-oriented distributed systems. He has published more than 50 internationally refereed papers in these fields and the 1999 young researchers award by the Galician Royal Academy of Sciences.

Martín Llamas received the Telecommunication Engineering (1986) and Doctor in Telecommunications (1994) degrees from the Technical University of Madrid, Spain. In addition to teaching, he is involved in research in the areas of Formal Description Techniques and Computer Based Training. He is the head of the GIST research group and New Technologies Division at the University of Vigo.

Manuel J. Fernández received the Doctor Ingeniero de Telecomunicación degree from the Universidad de Vigo in 1997. He is a Profesor Titular at the Departamento de Tecnologías de las Comunicaciones, Universidad de Vigo. He has participated in several research projects in the areas of Telecommunications and Software Engineering and has written several papers in international refereed journals. His interests include protocol engineering, information retrieval and Internet programming. Currently he is the Secretary of the School of Telecommunications at the Universidad de Vigo.

Manuel Caeiro received the Telecommunication Engineering (1999) degree from the University of Vigo. Currently, he teaches Computer Architecture at the Telecommunication School where he is working towards his PhD.

Juan Santos is a Telecommunication Engineer (1998). He has been involved in several projects related to distance learning and e-commerce. Now, he is Profesor Asociado at the University of Vigo.

Judith Rodriguez received her Telecommunication Engineering (1999) degree from the University of Vigo. In addition to teaching, her main interests are in the Educational Metadata and Information Retrieval.