Existing Standards Analysis for Alfanet

Project Deliverable Report

Deliverable D31 – Existing Standards Analysis

Workpackage	WP3. Standards			
Task	T31. Existing Standards Analysis			
Date of delivery	**Contractual**	30 09 2002	**Actual**	28 01 2003
Code name	ALFANET_D31.doc		**Version** 1.0	draft ☐ final ☑
Type of deliverable	Report			
Security (distribution level)	Public			
Contributors	OUNL, UNED			
Authors (Partner)	Rene van Es, John van der Baaren, Peter van Rosmalen, Jocelyn Manderveld Rob Koper (OUNL); Jesús G. Boticario, Carmen Barrera, Antonio Rodríguez, Olga Santos (UNED)			
Contact Person	Peter van Rosmalen Open University of the Netherlands - OTEC Valkenburgerweg 177 6401 DL. Heerlen		Tel: + 31-45 576.2889 Fax: + 31-45 576.2800 Email: peter.vanrosmalen@ou.nl	
WP/Task responsible	OUNL			
EC Project Officer	Elena Coello			
Abstract (for dissemination)	This deliverable consists of two parts. Part I gives an overview of existing bodies for standardization and an introduction to existing standards (or specifications) in the areas relevant for ALFANET including learning standards, knowledge management standards, human capital profiles, multi agent architecture standards and other general technical standards. Part II contains a selection of the standards proposed to be used within ALFANET. Each proposed standard is accompanied by an in depth discussion reviewing the motivation, fit with the project goals and the consequences of inclusion.			
Keywords List	Standards, Standard Bodies, Learning Technology Standards, Knowledge Management Standards, Multi-agent Standards, Human Capital Profiles			

ALFANET Project Coordination at: Software AG España, S.A. Ronda de la Luna, 4
Tel: +34 91 8079411 - fax: +34 91 8079447 - email: carana@softwareag.es Tres Cantos, E-28760 Madrid

Executive Summary

Introduction

The present document describes the results of activity T31: Existing Standards Analysis. The purpose of T31 is to study and describe current standards and to investigate their applicability for ALFANET and additionally gain first insight in which way ALFANET may contribute to current standards of the e-learning field in the adaptation and personalisation on contents and learning methods.

This deliverable consists of two parts.

Part I gives an overview of existing bodies for standardization and an introduction to existing standards (or specifications) in the areas relevant for ALFANET including learning standards, knowledge management standards, human capital profiles, multi agent architecture standards and other general technical standards. The result of part I are used to guide the discussion and selection for part II.

Part II contains a selection of the standards proposed to be used within ALFANET. Each proposed standard is accompanied by an in depth discussion reviewing the motivation, fit with the project goals and the consequences of inclusion.

Description of conclusions/results[1]

The standards studied are all related to learning technology directly or that have a strong relation with learning technology.

The main finding for learning technology standards is to focus upon the use of IMS-LD. The objective of ALFANET to offer a highly adaptive, personalized learning experience including a variety of pedagogical methods requires the capability to model both structure and process, including the specification of roles and activities. IMS-LD (including Edubox) offers this capability and equally important in depth knowledge of IMS-LD is available and directly accessible, so a quick start can be made. Moreover, IMS-LD allows the integration of specifications. Finally, IMS-LD is gaining international interest.

To assure a successful inclusion a number of points has to be taken care of:

- The relation of IMS-LD with other standards to be included have to be investigated, in particular in the area of knowledge management and multi agent architectures, and the technical consequences have to be documented.
- All partners of ALFANET have to acquire the appropriate level of knowledge of IMS-LD. This ranges from the basic knowledge of the language, to the design process of ‘units of learning’, to the technical consequences to be able to fit IMS-LD in the proposed ALFANET application.

Student related data is advised to be stored in a common dossier such that it can be exchanged between the ALFANET subsystems. For this purpose IMS-LIP specification is recommended. IEEE LOM can be used to support part of the adaptive functionality of ALFANET. The meta-data specifications create a uniform way for describing learning resources so that they can be more easily found (discovered) and subsequent used.

The main finding for standards on multi-agent architecture that were studied revealed that the most relevant standard is the FIPA. The FIPA standard is relevant for the communication with the educational subsystem. It could be useful to design a part of the dossier as a data structure common to both agent and educational subsystem. Both subsystems read and write to this common dossier. The exact modules where this standard could be used are not known at this stage.

[1] Specifications and standards are regularly updated and new specifications emerge. It is advised always to look for the latest version. This study has been prepared in the period June – December 2002. Each section refers to the version discussed.

Table of Contents

1. Introduction

1.1 Situation

The present document describes the results of activity T31: Existing Standard Analysis, which purpose it was to study and describe technological and educational standards relevant for reaching the objectives of the ALFANET project. The main goal of the ALFANET project is to build an e-learning system that will take advantage of the new internet related techniques, human interaction, and machine learning, to allow:

a) organisation personnel to have interactive, adaptive and personalised e-learning experiences bringing them the opportunity to learn and experiment on matters that are relevant for their work.

b) organisations to control and efficiently manage intellectual capital, promoting the evolution of employees in specialised & multidisciplinary areas for their work.

In addition the project aims to contribute to the educational standardisation efforts for adaptive education and to define a business model for e-learning.

1.2 Purpose

The purpose of the document is to provide an overview of available standards on learning technologies and agent technologies that are relevant for the development of the ALFANET system. At the start of the project the following main requirements, constraints and basic assumptions were set out for the ALFANET system:

- The system will present effective and adaptive instruction to learners in a web-based learning environment.
- The system will allow learning content providers and educational centres to provide learning contents in such a way that these contents can be adapted to the personal needs.
- The system will allow collaboration between users (learners, teachers). It will facilitate the development of virtual communities which can manage workgroups on the web.
- The system will be secure, meaning that all personal information of users of the system will remain confidential, safeguards will be installed to avoid unauthorised access to secure the integrity of the system and measures will be taken to secure the copyrights of the learning materials.

Technically the solution is anticipated as:

- Some system modules will be based on a multi-agent approach.
- Adaptation of instruction to learners will be inferred from user models that are acquired from available learner data and the learner's interaction with the system. The models are build by applying a set of machine learning techniques.
- The system will be based on advanced pedagogical models (active learning, collaborative learning, …).
- The system will build upon and contribute to existing standards for describing and publishing learning materials, including standards for adaptive instruction.

In the light of these requirements the main objectives for task T31 were to:

1. Identify existing standards that could be readily used or adapted to fulfil the basic functional requirements of the ALFANET system
2. Identify technologies and approaches that could form the basis for the design of the ALFANET system in order to fulfil the main functional requirements for the system.
3. Identify standards that the ALFANET system that are recommendable to adhere to or should be able to interface or integrate with.

This document is intended to be read by designers and developers of the ALFANET system.

1.3 Overview

In order to achieve the objectives for Task T31 as stated in the previous section all project participants identified possible relevant topics for study. This document shows the result of that study.

This report is build up of two parts, part one provides an overview of existing standards, part two examines the role and relation of the identified standards in part one for the application within ALFANET.

Part I

The standards that could be usable to ALFANET are categorized according to these categories:

- Learning technologies specifications
- Knowledge management standards
- Human capital profiles
- Multi-agent architecture standards
- Other technical standards

Learning technologies specifications

In this chapter first the bodies that are involved in creating specifications for learning technology are identified and it is specified what role they play. Then the most relevant and promising specifications are elaborated. This is done according to the specification body that is involved in the development of that particular specification. Per specification a brief description is given, its components are described, the relation with other specifications are explained, and the applicability of the standard for ALFANET is explained.

Knowledge management standards

This chapter starts with an analysis of the domain of knowledge management. Next, it focuses on analysis of standards related with the knowledge representation (ontologies and related standards) and different approaches about skills, competencies and curricula standards. Finally the applicability for ALFANET is indicated.

Human capital profiles

This chapter deals with the HR-XML's efforts that are focused on standards for staffing and recruiting, compensation and benefits, training and workforce management. The chapter provides an overview of these standards, indicates the relations that these standards hold with other similar standards, and it indicates the use of these standards for ALFANET.

Multi-agent architecture standards

The chapter starts with an explanation of the characteristics of agents. Then a multi-agent environment is decomposed into its main parts which are then explained in more detail. Several methodologies that could be used to construct a multi-agent environment are elaborated. The chapter concludes with recommendations for ALFANET.

Other technical standards

This chapter contains those standards that could not directly be categorized in the other chapters. The first standard is P3P which is a platform to exchange privacy information. The second is Learner profiles which has more or less the same function as the human capitol profiles described in chapter 4. For this standard related standards are mentioned. The chapter concludes with indications for use within ALFANET.

Part II

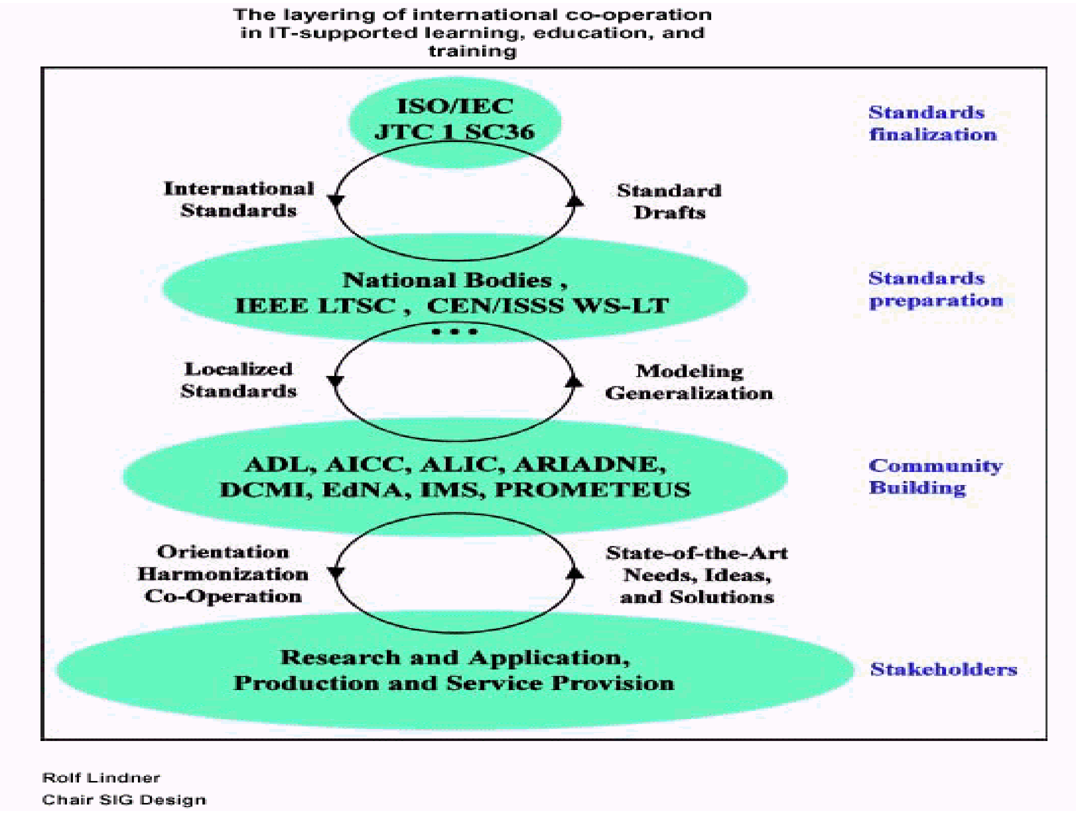
Part II contains a selection of the standards proposed to be used within ALFANET. Each proposed standard is accompanied by an in depth discussion reviewing the motivation, fit with the project goals and the consequences of inclusion. There are two sections in this part of the document. The first section deals with the selection of learning technology recommendations. The arguments for the selection are presented and several scenarios are shown to indicate where/how the selected recommendations are of use within ALFANET.

The second section deals with the selection of recommended standards for agent technology. The arguments for the selection are presented.

Part 1: Overview of existing Bodies, Specifications and Standards

1. Bodies (for learning technologies)

1.1 General overview



1.2 Bodies

In the area of learning technology the following organisations are mentioned in the report 'Making Sense of Learning Specifications & Standards' by the Masie Center 8 march 2002:

ADL Initiative: (Advanced Distributed Learning)

An initiative by the U.S. Department of Defense and its partners in industry, academia, and the private and federal sectors to achieve interoperability across computer and Internet-based learning courseware through the development of a common technical framework, which contains content in the form of re-usable learning objects. This group is responsible for authoring the SCORM document. (http://www.adlnet.org)

From the ADL Web site: The purpose of the ADL initiative is to ensure access to high-quality education and training materials that can be tailored to individual learner needs and made available whenever and wherever they are required. This initiative is designed to accelerate large-scale development of dynamic and cost-effective learning software and to stimulate an efficient market for these products to meet the education and training needs of the military and the nation's workforce of the future. It will do this through the development of a common technical framework for computer and net-based learning that will foster the creation of re-usable learning content as "instructional objects."

AICC (Aviation Industry Computer-Based Training Committee):

An international association of technology-based training professionals that develops training guidelines for the aviation industry. AICC is developing standards for interoperability of computer-based and computer-managed training products across multiple industries. (http://www.aicc.org)

From the AICC Web site: The AICC's mission is to provide and promote information, guidelines and standards that result in the cost-effective implementation of CBT and WBT.

ALIC (Advanced Learning Infrastructure Consortium) (Japan):

From the ALIC Web site: Our objective is to establish an active society by reasonably and effectively providing a learning environment, which enables anyone to learn anytime, anywhere, according to the goals, pace, interests and understanding of individuals and groups. Also, we attempt to foster experts who will be the origin of global competitiveness. (http://www.alic.gr.jp/eng/index.htm)

ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe):

From the ARIADNE Web site: ARIADNE is a research and technology development project pertaining to the "Telematics for Education and Training" R&D program sponsored by the European Union. The project focuses on the development of tools and methodologies for producing, managing, and re-using computer-based pedagogical elements and telematics-supported training curricula. Validation of the project's concepts is currently taking place in various academic and corporate sites

across Europe. (http://ariadne.unil.ch)

CEN/ISSS (European Committee for Standardization/Information Society Standardization System):

From the CEN/ISSS Web site: The mission of CEN/ISSS is to provide market players with a comprehensive and integrated range of standardization-oriented services and products, in order to contribute to the success of the Information Society in Europe. (http://www.cenorm.be/isss)

EdNA (Education Network Australia):

From the EdNA Web site: EdNA Online is a service that aims to support and promote the benefits of the Internet for learning, education, and training in Australia. It is organised around Australian curriculum, its tools are free to Australian educators, and it is funded by the bodies responsible for education provision in Australia - all Australian governments. (http://www.edna.edu.au/EdNA)

DCMI (Dublin Core Meta-data Initiative):

From the DCMI Web site: The Dublin Core Meta-data Initiative is an open forum engaged in the development of interoperable meta-data standards that support a broad range of purposes and business models. DCMI is dedicated to promoting the widespread adoption of these standards and developing specialized meta-data vocabularies for describing resources that enable more intelligent information discovery systems. DCMI's activities include consensus-driven working groups, global workshops, conferences, standards liaison, and educational efforts to promote

widespread acceptance of meta-data standards and practices.

GEM (Gateway to Educational Materials):

From the GEM Web site: The Gateway to Educational MaterialsSM is a Consortium effort to provide educators with quick and easy access to thousands of educational resources found on various federal, state, university, non-profit, and commercial Internet sites. GEM is sponsored by the U.S. Department of Education and is a special project of the ERIC Clearinghouse on Information & Technology. Teachers, parents, administrators can search or browse The Gateway and find thousands of

high quality educational materials, including lesson plans, activities, and projects from over 414 GEM Consortium member sites. (http://thegateway.org)

IEEE (Institute of Electrical and Electronics Engineers):

The IEEE's Learning Technology Standards Committee is working to develop technical standards, recommended practices, and guides for computer implementations of education and training systems. From the IEEE Web site: The mission of IEEE LTSC working groups is to develop technical Standards, Recommended Practices, and Guides for software components, tools, technologies, and design methods that facilitate the development, deployment, maintenance, and interoperation of computer implementations of education and training components and systems. (http://ltsc.ieee.org)

IMS Global Learning Consortium (Instructional Management System):

IMS is a global consortium with members from educational, commercial, and government organizations dedicated to defining and distributing open architecture interoperability specifications for e-Learning products. From the IMS Web site: IMS Global Learning Consortium, Inc. (IMS) is developing and promoting open specifications for facilitating online distributed learning activities such as locating and using educational content, tracking learner progress, reporting learner performance, and exchanging student records between administrative systems. IMS has two key goals:

1. Defining the technical specifications for interoperability of applications and services in distributed learning, and
2. Supporting the incorporation of the IMS specifications into products and services worldwide. IMS endeavours to promote the widespread adoption of specifications that will allow distributed learning environments and content from multiple authors to work together (in technical parlance, "interoperate"). (http://www.imsproject.org)

ISO (International Organization for Standardization):

From the ISO Web site: The ISO is a worldwide federation of national standards bodies from some 140 countries, one from each country. ISO is a non-governmental organization established in 1947. The mission of ISO is to promote the development of standardization and related activities in the world with a view to facilitating the international exchange of goods and services, and to developing cooperation in the spheres of intellectual, scientific, technological and economic activity. ISO's work results in international agreements which are published as International Standards. Currently ISO/SC 36 is involved in learning technology standardization.

(http://www.iso.org)

2. Learning technology specifications

2.1 Instructional Management Systems (IMS)

IMS Global Learning Consortium, Inc. (IMS) is developing and promoting open specifications for facilitating online distributed learning activities such as locating and using educational content, tracking learner progress, reporting learner performance, and exchanging student records between administrative systems

The IMS project defines the following separate specifications.

- *Learning Resource Meta-data*). This is a specification of meta-data used to identify "learning resources".
- *Content packaging*). A specification of how to assemble and distribute content in "packages".
- *Resource identifiers*). This defines persistent, location independent resource identifiers.

- *Question & Test Interoperability (QTI)*. This defines the structure of questions and tests, and the grouping of these.
- *Enterprise*). This defines the way information on the learning 'enterprise' (instructional processes) is shared.
- *Learner information packaging*. This specifies how to record and share information on the learner.
- *Reusable Competency Definitions*). An information model for describing, referencing and exchanging definitions of competencies, primarily in the context of online and distributed learning.
- *Simple Sequencin*g. This defines how to associate sequencing information with *content packs*) and its default behaviour.
- *Learning Design*. This describes the elements and structure of an unit of learning

Each specification has (or will have) at least three main parts:

- Information model — an abstract description of the area modelled
- Binding — binding to a particular language. For all specifications XML is the language of choice
- Best practice — explanation of how to apply the model.

2.2 IMS - Learning Resources meta-data

2.2.1 General

The meta-data specifications create a uniform way for describing learning resources so that they can be more easily found (discovered), using meta-data aware search tools that reflect the unique needs of users in learning situations

A learning resource is any information- or tool object that can be reused in different environments.

In order to reuse the resources, several descriptions of the resource are required. The meta-data proposal offers descriptive 'layers' that are represented as consecutive XML elements in the XML binding document.

2.2.2 Components

A meta-data instance is a single specification, that is: a single XML document. This is a 'conforming LOM meta-data instance'. The components of a single meta-data specification are:

- General — Context independent features of the resource. Offer handles for search and retrieval.
- Lifecycle — Features of the lifecycle of the resource. Manage the (change history and) version of the resource
- Meta-metadata — Features of the description rather than the resource. Manage the meta-data entry itself.
- Technical — Technical aspects of the resource. The resource is assumed to be available in an electronic form; a hardback book is hard to describe using this scheme.
- Educational — Educational aspects of the resource. This includes the level of interactivity, for what user (type and level) the resource is intended, and such.
- Rights — Legal aspects of using the resource, i.e. costs and copyright.
- Relation — Possible typed relations with other resources.
- Annotation — Comments on the educational use of the resource.
- Classification — Some classification of the resource, based on a taxonomic path, keywords within the taxonomy.

These components are represented as subelements of the <lom> root element which is required for all meta-data instances. If the set of constructs is not sufficient, extensions can be made. The proposal suggests the use of alternative namespaces to identify these extensions; it does not provide a strategy.

2.2.3 Relation with other standards

General

Meta-data is not bound to a particular language. It can for example be represented in HTML as follows:

```
<META name="description" content="The IMS meta-data system.">
<META name="keywords" CONTENT="IMS, Metadata, Meta Data, meta-data,
fields, online, on-line, on line, knowledge, distributed, instruction,
education, learning">
```

Relation with Dublin Core

IMS meta-data covers the Dublin Core and a mapping is provided within the specification.

Relation with IEEE

IMS and ARIADNE collaborated to develop the meta-data specification that is being used by the ADL and has been provided to IEEE P1484 for possible accrediting by the Learning Objects Metadata working group. Just recently, July 2002, the IEEE group did release the official IEEE 1484.12.1 – 2002 Final LOM draft standard document. Note. IEEE restricts itself to the information model, it does not specify a binding.

Relation with SCORM

Within the SCORM, the SCORM Meta-data Application Profiles are specializations of the IMS Learning Resource Meta-data Specification Version 1.1 The SCORM imposes additional constraints on the application of the specification.

The SCORM Version 1.2 contains three meta-data application profiles: 1/ Resource (SCORM Version 1.1 Raw Media Meta-data) 2/ SCO (SCORM Version 1.1 Content Meta-data) 3/ Content Aggregation (SCORM Version 1.1 Course Meta-data)

Relation with IMS – Learning Design

IMS Meta-data can be used to provide meta-data to specific elements designed in IMS Learning design. Therefore these two specifications can coexist and each fulfil a specific need.

2.2.4 Applicability for ALFANET

The objective of the meta-data specifications, i.e. to create a uniform way for describing learning resources so that they can be more easily found (discovered) and subsequent used appropriately is clear. However, there are some concerns, adaptivity may require very specific information. Moreover, ALFANET has to be aware that it is not merely the metadata that assures finding and applying but equally important are e.g. a consistent educational design approach and a clear perspective on object size.

2.2.5 References

IMS Learning Resource Meta-data Information Model. Final specification, Version 1.1. IMS global learning consortium, Inc., 2000.

Available at: http://www.imsproject.org/specifications.html

IMS Learning Resource Meta-data Best Practice and Implementation Guide. Final specification, Version 1.1. IMS global learning consortium, Inc., 2000.

Available at: http://www.imsproject.org/specifications.html

IMS Learning Resource Meta-data Best Practice and Implementation Guide. Public draft specification, Version 1.2. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IEEE Learning Object Metadata, IEEE 1484.12.1 – 2002, Final LOM draft standard document, July 2002.

Available at: http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf

2.3 IMS - Content Packaging

2.3.1 General

The IMS Content Packaging Specification describes how to collect reusable content objects such that they are useful in a variety of learning systems. It describes data structures that are used to provide interoperability of Internet-based content with content creation tools, Learning Management Systems (LMS), and run-time environments. The objective of the IMS Content Packaging Specification is to define a standardized set of structures that can be used to exchange content. The scope of the IMS Content Packaging Specification is on defining interoperability between systems that wish to import, export, aggregate, and disaggregate packages of learning content.

2.3.2 Components

An IMS Content Package contains two major components:

1. a (required) special XML document describing the content organization and resources of the package. The special file is called the Manifest file (imsmanifest.xml) because package content and organization is described in the context of manifests.

2. the physical files referenced in the Manifest.

This is shown in the following figure, taken from *[ims-cp-best-1.1.2]*.

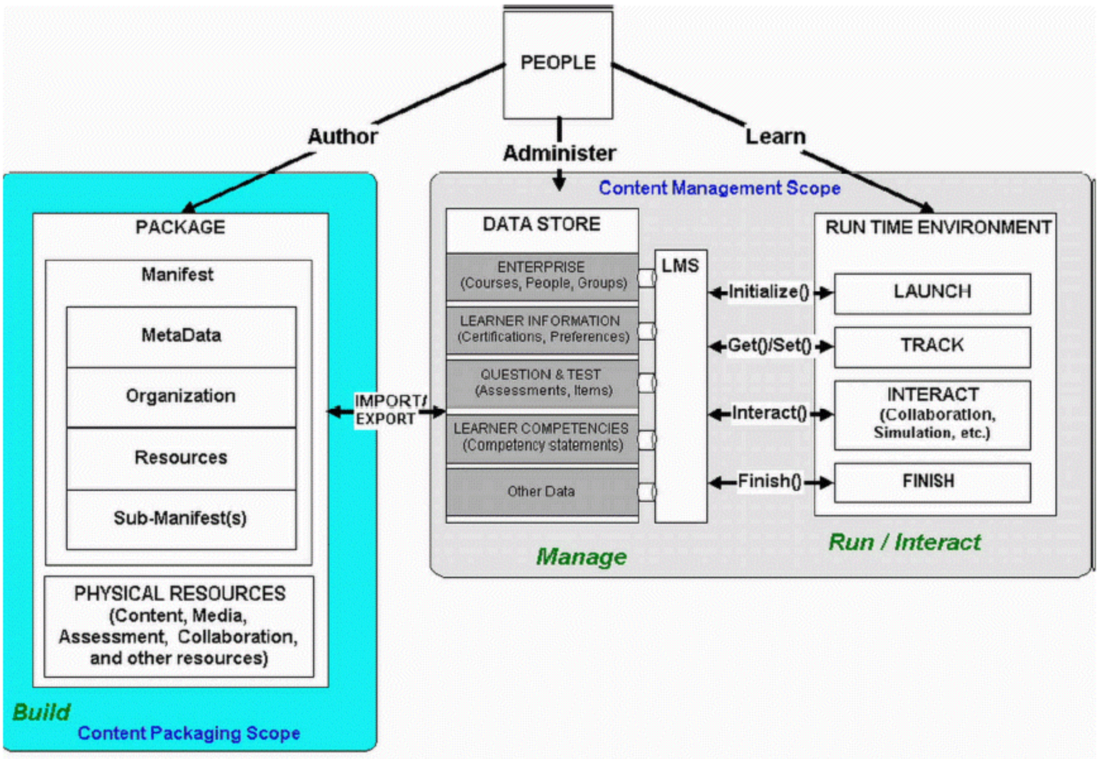


Figure 2-1. IMS Content Package Components

IMS Content Package shown on the left, and related IMS specifications on the right.

The CP defines the following components:

Package
 Manifest
  Metadata?
   Schema
   Schemaversion
   #any
  Organizations
   Organization*
    Title?
    Item*
     Title?
     Item*
     metadata?
     #any
    Metadata?
    #any
   #any
  Resources
   Resource*
    Metadata?
    File*
     Metadata?
     #any
    Dependency
     #any
  Manifest*
  #any
 Physical resources...

- A *package* represents a unit of usable (and reusable) content. It is typically stored in a folder and passed on as a zip file. This contains the physical resources and the manifest file

- A *manifest* is a description in XML of the resources comprising meaningful instruction. Submanifests may occur, in which case these are interpreted in the context of the outer manifest. Submanifests are intended to be independent of their containing manifests, just like boxes within a box are still capable of holding stuff on their own.

- Within the manifest a *metadata* section is available and may, for example be used to merge the resource descriptions into a catalog. A bias exists toward the IMS meta-data but is not part of the specification.
- A Manifest may also contain zero or more static ways of organizing the instructional resources for presentation using *organizations*. The organization consists of items that 1) reference a resource, 2) reference a manifest, and/or 3) hold one or more subitems. More than one organization may be supplied which are deemed to be equivalent in learning outcomes; a default organization is required.

  The *Simple Sequencing* (see one of the next sections) initiative introduces and alternative approach here; a single organization is expressed using a SS specification.
- The *resources* component can describe external resources, as well as the physical files that the package consists of. These files may be media files, text files, assessment objects or other pieces of data in electronic form.

2.3.3 Relation with other standards

IMS CP has been extended for *SCORM conformance* using ADL specific information elements. ADL specific elements necessary for packaging SCORM content are defined within an ADL namespace.

2.3.4 Applicability for ALFANET

The IMS LD specification builds on Content Packaging.

2.3.5 References

IMS Content Packaging Best Practice Guide. Final specification, Version 1.1.2 IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Content Packaging XML binding. Final specification, Version 1.1.2 IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Content Packaging Information Model. Final specification, Version 1.1.2 IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

2.4 IMS – Resource Identifiers

2.4.1 General

There is a need for persistent, location-independent, resource identifiers across multiple (IMS) specifications. A persistent, location-independent, resource identifier is defined as an instance of a data type or data format associated with an item which provides a persistent, immutable label with global scope and indefinite lifetime.

2.4.2 Components

The RI handbook defines 1) Globally Unique Identifiers (GUIDs) and 2) Universally Unique Identifiers (UUIDs). The IETF Uniform Resource Name (URN) is proposed as a candidate for the IMS UID scheme.

The URN is a calculated string and is not registered. The URN form is: *URN:[nid]:[nss]* where [nid] is the Namespace Identifier and [nss] is the Namespace Specific String. Example: *URN:ISBN:0-395-36341-1*

The following application strategy is followed.

— Existing URN schemes should be used for all objects which have a formal URN scheme.

— Large organizations should obtain their own NID. This includes the IMS itself. IMS develops a NSS scheme for associated organisations.

— An organization receiving an object with a URN will determine if it trusts the uniqueness of the URN; it must preserve the URN and upon export or transmission, it must label the object with the original URN.

— Two objects with the same URN are identical. Lexical equivalence of URNs is based on their encoding scheme.

— An object can only have 1 URN.

An example of an IMS GUID would be (URN within the IMS NID using a sourced NSS without a scheme):
URN:IMS-PLIRID-VO:DUNS:05-218-4116::6ba7b8149dad11d180b400c04fd430c8

2.4.3 Relation with other standards

The IEEE Learning Object Meta-data (LOM) Specification has placeholders to store pan-organizational identifiers.

IMS Content Packaging has identifier fields which are recommended to be unique across all organizations. The IMS Competency Working Group intends to use unique identifiers as the fundamental identification mechanism for reusable competency definitions. There are numerous existing and proposed repositories for competency, skills, and outcomes definitions. If this specification is to have any practical effect, it will be necessary to reference these definitions in a unique and machine-retrievable way. Note that none of the IMS specifications identify an adequate scheme to define the data type or representation for such a unique identifier.

The handbook references existing identification schemes such as ISBN, DOI, DUNS, MAC, URN.

2.4.4 Applicability for ALFANET

This specification could be used to provide the units of learning with a location independent resource identifier. However there is no foreseen use of a distributed content repository at this stage, thus there would be very limited usage.

2.4.5 References

IMS Persistent, Location-Independent, Resource Identifier Implementation Handbook. Version 1.0 Final Handbook. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

2.5 IMS – Question and test

2.5.1 General

The IMS Question & Test Specification addresses the need to share test items and other assessment tools across different systems. It builds upon an envisioned workflow which includes authors, assessors, candidates, tutors and so on. The question and test system itself holds an assessment engine which reads a repository of questions and tests, and information on eligibility and performances. It then evaluates the responses in terms of producing scores and feedback.

The QTI specification is defined in XML to promote the widest possible adoption.

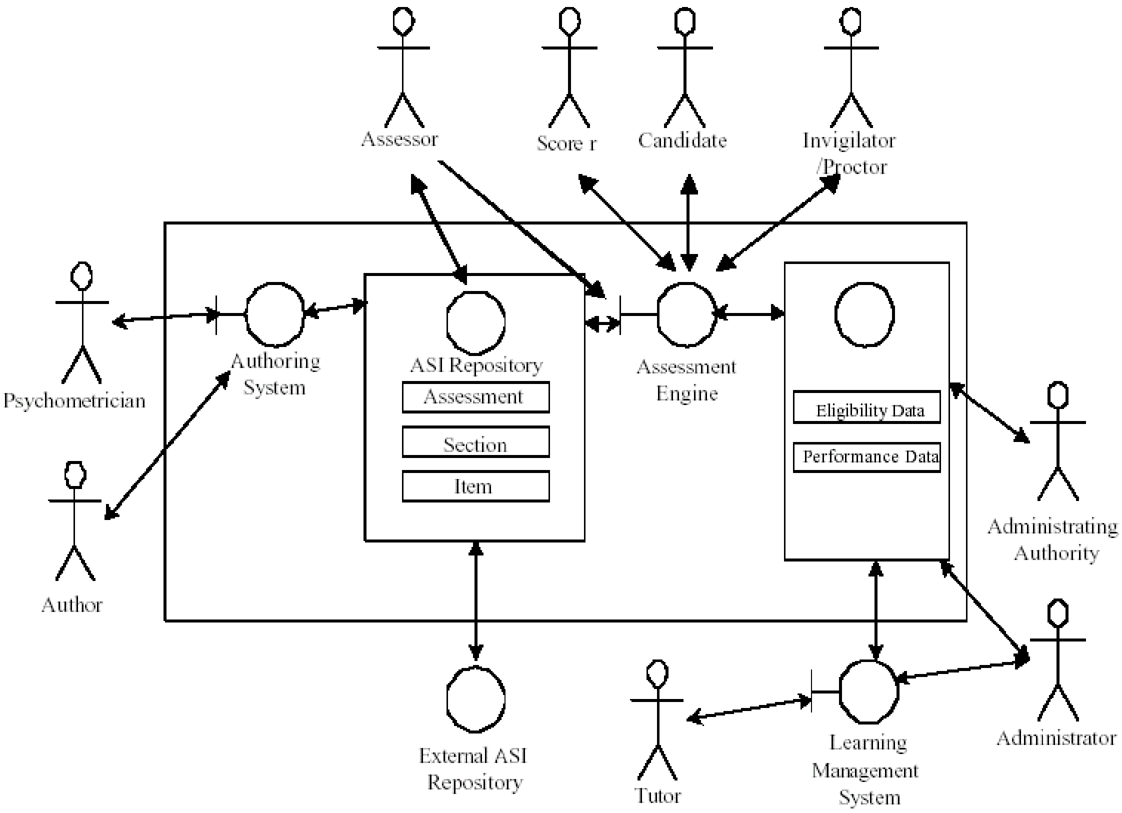


Figure 2-2. IMS representation of the assessment system. (Taken from *[IMS-QTI-INFO-1.1]*)

2.5.2 Components

There are three kinds of reusable data objects in the QTI model, representing *assessments, sections, items (ASI)*. The ASI objects all have meta-data in accordance with IMS Meta-data specifications.

- *ASSESSMENT Data Object* — A complete *assessment*, for example: a test. This may be a complex collection of several parts, known as Sections and Items. An Assessment object contains all of the information to make the use of individual Items meaningful. This means that, apart from the sections themselves, it includes the relationships between the sections, the group evaluation processing and the corresponding feedback.

  The user interacting with an assessment is known as the *Participant*.

  — The assessment meta-data is defined in terms of 1) general QTI meta-data fields, that have no specific internal structure, 2) single specifications for typing and processing the assessment (not for retrieval).

— Objectives and rubric are textually defined (materials and text flows).

— Assessment control holds control flags for presenting the assessment, using switches for showing feedback, hints and solutions. It also defines for whom this material is available ('view', such as All, Administrator, Assessor, Candidate)

— Assessment processing is implemented using 1) variable declaration structure and 2) a score expression language that reads and writes these variables. The language is completely score variable based, for example: “If variable A has value V, set variable B, and show feedback F”.

— Assessment feedback is material shown on completion of the assessment.

— The assessment specification is completed by an ordered set of sections.

- *SECTION data object* — A *Section* object contains all of the information to meaningfully group together Items. Apart from the Items it includes the relationships between the Items and the selection criteria of the Items.

  — Section meta-data is 1) general QTI meta-data, followed by 2) processing meta-data for this section (number of items, selection sequence etc).

  — Objectives and rubric, see elsewhere.

  — Section control see elsewhere

  — Sections are embedded.

  — Section processing is the processing of accumulated responses and scores of the embedded sections. Again the score expression language is used.

  — Feedback, see elsewhere.

  — The section specification is completed by an ordered set of items.

- *ITEM Data Object* — An *Item* object contains information on how to present a question and its subsequent processing to the user. The structure of the Item includes one or more actual questions and responses as well as its presentation format, the range of possible responses, the ways in which the responses are to be processed, and the possible solutions and hints to the Item.

  The item is of a particular *response-type*, i.e. the item is typed in accordance with the kind of response required, such as a *logical ID* (for MC questions) or a *string* (for fill-in-the-blanks).

  — Item meta-data is composed like the assessment and section meta-data. It includes directions like how many hints are permitted, level of difficulty, rendering type etc.

  — Objectives, see elsewhere.

  — Item control, see elsewhere

  — (Item) rubric holds material for specific views, see assessment.

  — Presentation, which is a container for response types and rendering forms. A simple presentation has just one response type.

  — Response processing is defined in terms of outcomes (declare and set variables) and conditions as described for assessment and section.

  — Item feedback is textual by nature, and may include the solution (for self-tests for example) and hints.

Reponse types

An author must determine for a question the *response type* and map this onto a *rendering type*. The response type can be described using three orthogonal feature sets: response structure, multiplicity and timing.

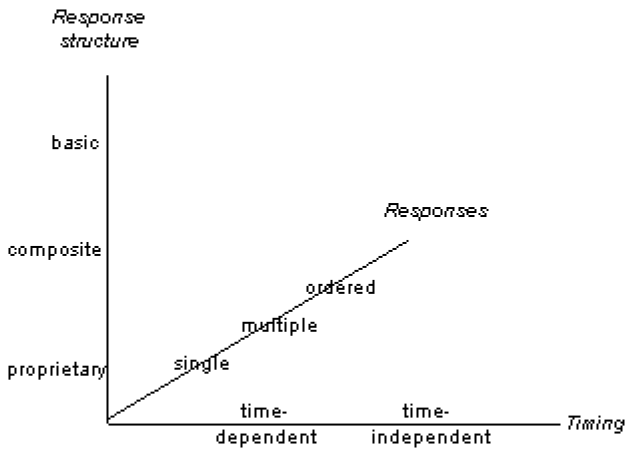


Figure 2-3. QTI response types

The following levels are inherent to the QTI response type (see *figure*):

- Basic response — One single type of response is expected. For example: a string.
- Composite response — Several grouped responses. For example: three answers that are thematically related.
- Time dependent — The time used to generate the response is part of the response handling. For example, when 'slow' offer different question.
- Time independent — Time is of no importance to the reponse handler.
- Single response — A single response is returned for the item, whether or not the item is multiple response.
- Multiple response — Several responses are returned for each item.
- Ordered response — Multiple, where order is significant.

Response types denote the kinds of results of processing a question. This is any of Logical ID (LID), XY coordinate (XY), String (STR), Numerical (NUM) or Logical Group (GRP). Per response type several rendering types can be chosen.

The response type values are also associated with a *duration* and an *identifier*. The duration element is the period between the item being triggered and the response(s) being supplied. The identifier ensures that the scoring attributes can be correlated to the generating response.

Rendering types

Each of the response types may be based on an item rendered in a particular way. For example, a MC question may be rendered as a list of, say, 4 items, or as a hotspot representation. Render types are part of the IMS specification. These rendering types all produce a response type.

The following rendering types are defined (with associated response types).

- True/false LID
- Multiple choice LID
- Multiple response LID

- Image hotspot XY
- Fill-in-blanks STR NUM
- Select text STR
- Slider LID NUM
- Drag object LID GRP
- Drag target LID GRP
- Order objects LID XY
- Match items LID GRP
- Connect the points LID XY
- Short answer STR (not in response types overview)
- Essay STR (not in response types overview)

Object model

The above constructs (assessments, sections, items, responses), are part of the QTI object model. This also includes objects for dealing with the outcomes. These objects are all reflected in the XML binding (through elements and attributes). The object model includes

- Assessment / <assessment> — as described.
- Section / <section> — as described.
- Item / <item> — as described
- Activity selection — selection of the next activity determined by the progress and results obtained upto the moment of activity selection
- Accumulation process — the reconciliation of all the evaluation outputs to produce an overall Assessment/Section evaluation
- Scoring weights — the scoring weights that are to be assigned to the results output from the response processing
- Response processing — the processing and evaluation of the user responses
- Presentation — the rendering of the content and the possible responses
- Examinee record — the set of collated results that are output from the complete process. This is a 'life-long' record in that it contains the historical progress of the individual
- Outcomes — the set of outcomes that are to be evaluated by the response processing object. These determine the scoring metrics to be applied to the response evaluations
- Response — the responses that are supplied by the user of the Items i.e. the input user selections
- Flow — the underlying presentation structure that defines the block relationship between the different *material* components
- Material — the content that is to be displayed

2.5.3 Relation with other standards

The IMS QTI will be presented to Aviation Industry CBT Committee, which works with ADL specifications.

ISO has a JTC1/SC36 — learning technology. This forum does however not yet define standards in the QTI field.

IMS is actually the only formal QTI specification body.

2.5.4 Applicability for ALFANET

A remark similar as made for the IMS meta-data specifications should be made. It is not clear how one of the key issues of ALFANET, adaptivity, can be supported. Among others the next use-cases will be considered for inclusion in **later** releases of the specification [taken from chapter 2.2, *IMS QTI: Selection & Ordering 1.2*]:

- Conditional selection on Item Outcome/Response - this comes from survey questions which the instructions read, if your answer is "No", skip to Item 'XXXX'. For example, if the answer to "Do you have a computer at home?" is 'No', there is no need to ask "How many hours a week do you use if for homework?" A related example is if the participant was always asked two multiple-choice questions. The first asked for the answer to some problem and the second asked what strategy the student used to arrive at the answer. Obviously, the text of the second question is dependent on the first;
- Simulated Cases - a more complex generalization of the previous case e.g. a "simulated medical exam" by having a medical student make a series of choices about patient treatment. The next question would follow on in sequence from the previous decisions;
- A form of Computer Adaptive Testing (CAT) that refers to those kinds of adaptivity that are based on current estimates of examinee ability e.g. the result of outcome processing and not just response processing. Typically, Items are chosen to maximize "value of information" or "weight of evidence". For example, in IRT-CAT Items whose difficulty matches the current best estimate of examine proficiency can be selected;

2.5.5 References

IMS Question & Test Interoperability ASI: Best Practice & Implementation Guide. Final specification, Version 1.2. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Question & Test Interoperability ASI: XML binding Specification. Final specification, Version 1.2. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Question & Test Interoperability ASI: Information Model Specification. Final specification, Version 1.2. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Question & Test Interoperability ASI: Selection & Ordering. Final specification, Version 1.2. IMS global learning consortium, Inc., 2002.

Available at: http://www.imsproject.org/specifications.html

2.6 IMS – Enterprise

2.6.1 General

The IMS Enterprise Specification is aimed at administrative applications and services that need to share data about learners, courses, performance, etc., across platforms, operating systems, user interfaces, and so on.

The basic architectural model for the Enterprise V1.1 Specification is shown in the following figure *[IMS-EN-INF-1.1]*.:

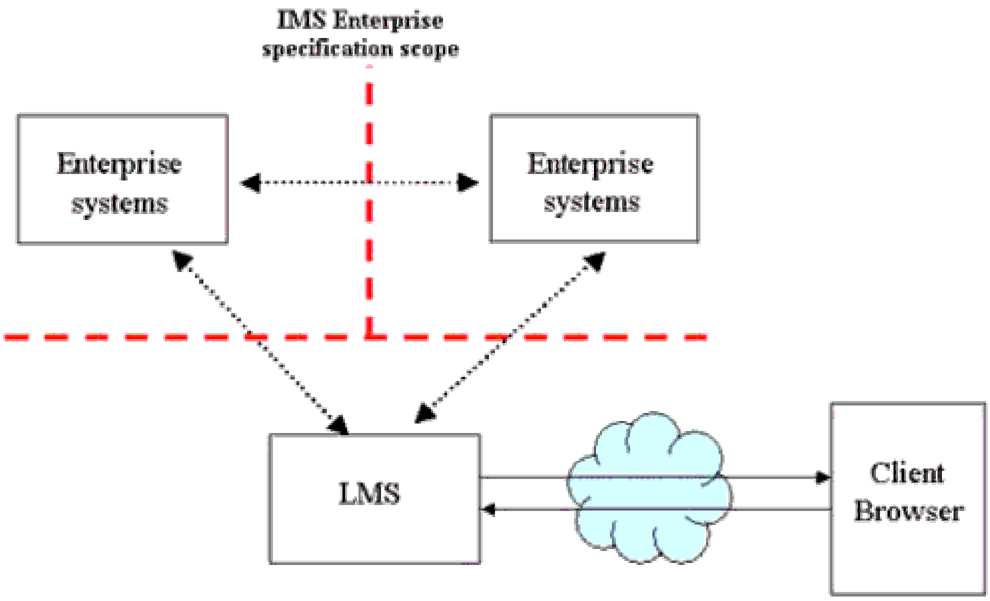


Figure 2-4. The basic Enterprise system architectural model.

In this architecture the scope of the IMS Enterprise Specification is shown as the dotted line. The scope of the interoperability is the data model of the objects being exchanged and not the associated behavioural model or the required communications infrastructure.

It defines a standardized set of structures that can be used to exchange data between Learning Management systems (LMS) and systems mentioned below. These structures provide the basis for standardized data bindings that allow software developers and implementers to create Instructional Management processes that interoperate across systems developed independently by various software developers.

The targeted systems are:

- Human Resource Systems — track skills and competencies and define eligibility for training programs.
- Student Administration Systems — support the functions of course catalog management, class scheduling, academic program registration, class enrolment, attendance tracking, grade book functions, grading, and many other education functions.
- Training Administration Systems — support course administration, course enrolment, and course completion functions for work force training.
- Library Management Systems — track library patrons, manage collections of physical and electronic learning objects, and manage and track access to these materials.

Note that this is confined to the same enterprise or organisation; it does not cross enterprises.

2.6.2 Components

The process components that the enterprise model focuses on are:

- Profile management — personal information.
- Group management — group related information, including class creation and scheduling.
- Enrolment management — information on registering and assignment of instructors.
- Result processing — dealing with the outcomes of the learning processes.

The specification builds upon groups and persons, and persons in groups (members).

- Group — This object contains elements describing a group of interest to the Learning Management environment. The most common is a Course Instance, but they may also include Training Programs, Academic Programs, Course sub-groups, clubs, etc. A group can also have any number of relationships with other groups. Properties include type of group, description, organisation, time frame in which this group is active, enrolment info, common contact info, relationships with other groups.
- Person — This data object contains elements describing an individual of interest to the Learning Management environment. This includes user's ID, name, demographical info, contact info, photo.
- Group Member — This data object contains elements describing the membership of a person or group within a group. Group members may be instructors, learners, content developers, members, managers, mentors, or administrators. This includes member ID's, role of that member within the group, subroles, status, timeframe for this member, final result of membership, and member contact info.

2.6.3 Relation with other standards

This specification shares data objects and model with IMS meta-data and IMS profile specification.

The specification lists relations with other standards as part of *[IMS-EN-BEST-1.1]*. In this case 'mappings' to these related standards are envisioned.

In EML no persons are defined, only roles. 'Groups' are defined by allowing learners or staff members to be assigned to a role which have multiple persons assigned to it.

2.6.4 Applicability for ALFANET

The scope of the IMS Enterprise Specification is focused on defining interoperability between systems residing within the same enterprise or organization. Other specifications e.g. from HR-XML (http://www.hr-xml.org) (c.f. chapter 4 of Part I) do also focus on inter-company exchange of human resources (HR) data.

2.6.5 References

IMS Enterprise Best Practice and Implementation Guide. Version 1.1 Public Draft IMS global learning consortium, Inc., 2002.

Available at: http://www.imsproject.org/specifications.html

IMS Enterprise XML Binding specification. Version 1.1. Public Draft IMS global learning consortium, Inc., 2002.

Available at: http://www.imsproject.org/specifications.html

IMS Enterprise Information Model. Version 1.1 Public Draft IMS global learning consortium, Inc., 2002.

Available at: http://www.imsproject.org/specifications.html

2.7 IMS – Learner information packaging

2.7.1 General

The Learner Information Package (LIP) specification provides a means to package learner information to the point that the resultant data is ready for exchange between disparate systems. “Learner Information” is the broad range of information that may be used by different systems to support the learner's activities. The systems using the specification may all provide part of the information needed for fully operational Learning systems. Flexibility of framework is therefore essential: the most elements in the LIP are optional, and the specification can be extended.

The LIP does not provide a protocol to exchange the packages. Also, LIPs are packaged as *Content Packaging specifications* and exchanged in that fashion.

2.7.2 Components

The components (called segments) of the LIP are as follows:

- Identification — Basic information that helps identify an individual. Elements like name and address (Vcard) are contained in this area.
- Goal — Learner's personal goals and aspirations. It allows status tracking related to any item in this area. A nested structure provides facilities for capturing sub-goals.
- QCL — Qualifications, certifications and licenses. It reflects accomplishments already completed along with a structure to indicate the source of the QCL and level attained.
- Accessibility — Learner preferences, language information, disability/accessibility information and technical/physical preferences.
- Activity — Education/training work and service of the learner. It is designed for flexibility for capturing disparate activities. This area goes beyond the simple recording of the activity and result by providing a space to include activity digital representations related to the activity, e.g. a code sample or a digital representation of a work of art.
- Competency — Elements for capturing skills the learner has acquired. Skills contained in this segment are associated with formal or informal training or work history. These skills may be related to other information reflected in the Activity and/or QCL segments.
- Interest — Information on hobbies and other recreational activities. These items may be related to QCL data and may also contain digital representations.
- Transcript — A placeholder for emerging standards from other organizations. This area introduces the concept of an exrefrecord that might be used to store another data format. One example might be the inclusion of an ANSI X.12 U.S. University Academic Transcript in its native EDI format. Similarly, one might store a PDF of the same document.
- Affiliation — Information on the descriptions of the organizations associated with the learner. This may include work groups, clubs or professional associations.
- Security Key — Learner information such as passwords or security keys.
- Relationship — Description of the relationships of data contained in the other segments. All relationships in LIP have been moved from the other segments to focus the collection of such information in this segment. This segment is also important for targeting previously provided information for update or deletion.

2.7.3 Relation with other standards

Student Educational Record (Transcript), ANSI ASC X.12-TS130, ANSI, April 1998.

Profile Format: Design Specification, Daniel Lipkin, Saba Inc, May 2000.

IEEE PAPI Specification - Learning Technology: Public and Private Information, Version 6.0, IEEE LTSC P1484, June 2000.

IMS *Content Packaging*, IMS *Meta-data*, IMS *QTI*

2.7.4 Applicability for ALFANET

Within a distributed system design the various system components need to exchange information. Within the various components different ways may be used format user data. To the outer world however it is highly recommended that one 'language is spoken'. For student information IMS-LIP looks like a usable specification.

2.7.5 References

IMS Learner Information Packaging Best Practice & Implementation Guide. Final specification, Version 1.0. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Learner Information Packaging XML Binding. Final specification, Version 1.0. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Learner Information Packaging Information Model Specification. Final specification, Version 1.0. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

Primer for the IMS Learner Information Package. Version 1.0. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

2.8 IMS – Reusable Competency Definitions

2.8.1 General

This specification defines an information model for describing, referencing and exchanging definitions of competencies, primarily in the context of online and distributed learning. "Competency" is used in a very general sense that includes skills, knowledge, tasks, and learning outcomes. This specification gives a way to formally represent the key characteristics of a competency independent of its use in any particular context. It enables interoperability among learning systems that deal with competency information by providing a means for them to refer to common definitions with common meanings.

The core information in a reusable competency definition is an unstructured textual definition of the competency with a globally unique ID. This information may be refined using a user-defined model of the structure of a competency. The specification is intended for interchange by machines.

2.8.2 Components

The specification is build on the following components:
`GUID`

```
Title
Description
Optional Definition (consists of:)
   Model
   Zero or more Statements (consists of:)
      StatementID
      StatementName
      StatementText
Metadata
```

Using this structure a human readable as well as automated record can be made of the competency. The author of a competency definition is free to use the Definition element in the way that best describes the competency. The *model* part identifies the model that the statements are based on.

2.8.3 Relation with other standards

Competencies are defined and structured in many ways in different communities of practice (ACRL, CASAS, CPA, Mager, NOICC, O*Net, PASS, SCANS, TATS). This specification allows communities of practice to exchange information according to the model they use. Extensibility can be achieved by defining the structure of the competency definition or by including LOM elements in the Meta-data portion.

This specification aligns with other IMS specs where the following constructs are concerned:

— IMS GUID guidelines and practices.

— IMS LR meta-data for extension.

2.8.4 Applicability for ALFANET

This recommendation focuses mainly on the content exchange between various systems. It is not foreseen that this feature is used in ALFANET therefore this specification has very limited value now. The moment the scale of ALFANET is broadened and competitive content providers enter the scene the use of this specification might be reconsidered.

2.8.5 References

IMS Reusable Competency Definitions Information Model. Version 0.1. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

2.9 IMS – Simple Sequencing

2.9.1 General

The IMS Simple Sequencing Specification provides a way to describe an intended behaviour and resulting learning experience from a collection of learning content when the Content Resources are delivered to a learner in a managed environment. Simple Sequencing defines the relative order in which elements of content are to be presented and the conditions under which a piece of content is selected or skipped in the presentation. It incorporates rules that describe the branching or flow through the content according to the outcomes and interactions of a learner with the different pieces of content. It also describes how learner actions and events cause the sequencing process to select and deliver Content Resources to the learner.

Content packs are in themselves static, and do not specify how content should be offered in a chain in order to express didactical structure or logic. When an IMS Simple Sequencing specification is recorded as part of a *Content Packaging* specification, it records the *organization* of items within the pack, and the organization of items within items. Sequencing and navigation information will enable systems to present elements of aggregate content in a predictable manner, while reacting consistently to learners' interactions with learning resources.

SS goals are:

- Describe the "intended behaviour of content" — the way content should be consumed by the learner. To this end SS presents a *content sequencing definition model*.
- Describe the expected behaviour of a learning technology system. To this end SS presents a *behavior model*.
- Describe the kind of interactions a learner has made with the system, which may influence the sequencing. This results in a *status tracking model*.
- Describe the format to encoding the sequencing descriptions.

The specification is abstract and independent of learning activity type, i.e. format or intent. Content need *not* use a communications adapter, such as an API implementation (cf. *ADL SCORM "Shareable Content Objects"* . The model does not prescribe an implementation.

2.9.2 Components

The SS model is based on the following assumptions, see also the next *figure*.

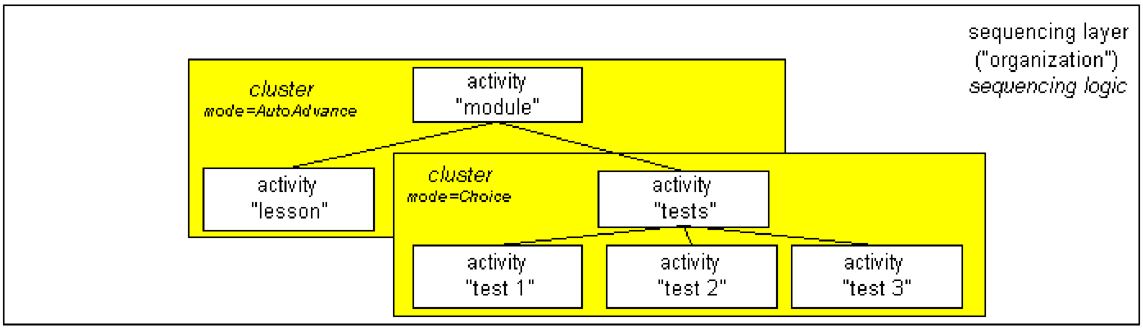


Figure 2-5. Representation of a simple sequencing specification.

- The SS defines sequences of *learning activities*. A learning activity may be loosely described as an instructional event or events embedded in a content resource, or as an aggregation of activities that eventually resolve to discrete content resources with their contained instructional events.
- Learning activities are modelled (conceptually) as a tree (*activity tree*). The tree consist of nodes (= activity descriptions) that may themselves be sectioned into sub-activities. Example: A lesson is followed by three tests.

  The content sequencing definition model defines the sequence of these nodes: under what conditions may the "next" node be accessed. For example: first test can be accessed only when lesson is "completed".

  Activities are always interpreted in the context of the parent activity (parents and children form a *cluster*). For example, when the parent activity is disabled, the child activities are inaccessible.
- Each node has a state for each learner, which is the basis for sequencing. Due to a number of events states may change and therefore sequencing may be evaluated differently. Example: Learner skips a test.

  The sequencing state model defines the data model for recording and updating such states.
- The sequencing model is build out of four processes: 1/ navigation, 2/ sequencing, 3/ delivery, 4/ rollup, described below. These processes take part in a "sequencing loop", which activates the processes (in that order).

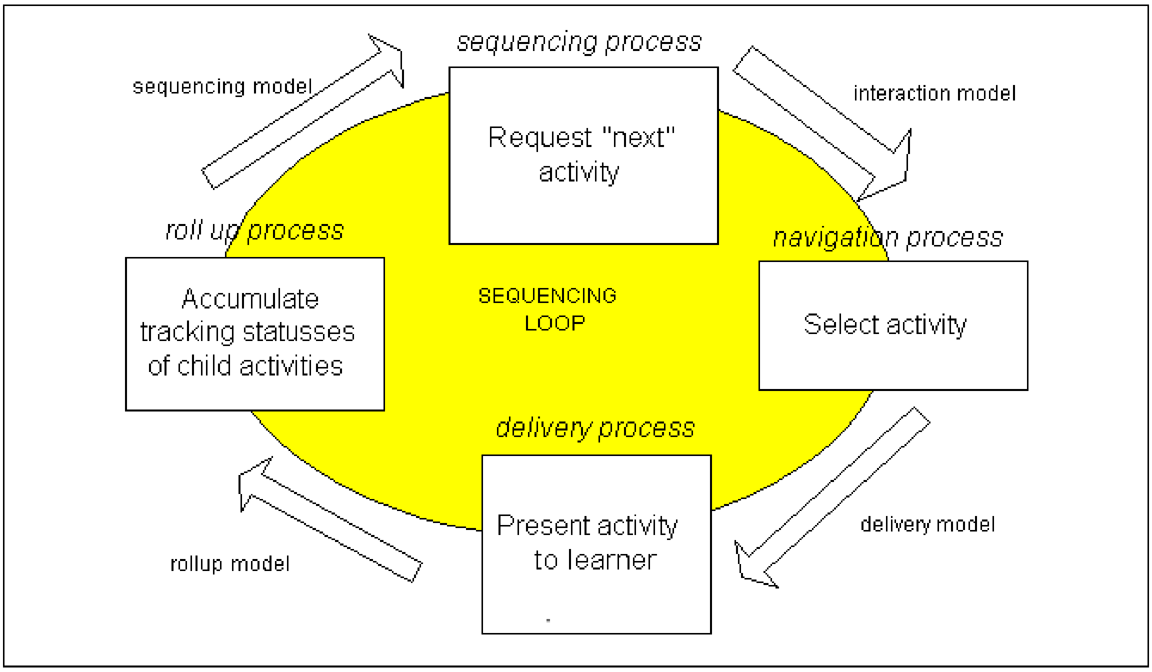


Figure 2-6. Simple Sequencing — the *sequencing loop*.

The sequencing loop

- 1/ *Navigation* — move through activity tree by user choice, external triggers such as timeout, etc. The specification only focusses on logical navigation events, i.e. those that trigger sequencing requests.
- In a web browser, this can be visualized on screen as buttons *[go to previous activity]* (history listing) or *[stop]*
- Sequencing *control modes* allow a particular way to access the nodes in a cluster to be defined: modes are *Flow* (system chooses next activity based on "continue" or "previous" sequencing request), *Choice* (learner chooses an activity), and *AutoAdvance* (next activity in sequence is selected).
- 2/ *Sequencer* — determine the candidate node (i.e. node queued to be delivered). This is done as a result of a sequencing request. A *sequencing request* is the expression of a desire to traverse the conceptual activity tree in a particular direction relative to the current activity; to a particular activity; or to exit a cluster or the entire aggregation. A sequencing request occurs as the result of an event such as a navigation event triggered by a learner, or a system generated event.

  This could be visualized on screen as a button: *[get next activity]*

  The sequencing process is guided by *sequencing rules*, that take the form of [condition, action] pairs (with possible subrules). The condition tests for status information, such as mastery, progress and 'limit conditions' status. The action is to disable, skip, hide, deny forward progress or hide all activities. Sequencing rules may take precedence over sequencing requests.

  Limit conditions are defined on activities and impose a constraint on access to the activity. For example: the maximum number of attempts is reached.

  Note that activities can be available concurrently. The 'main' sequence is always available; parallel sequences can be defined at will. Activities are associated with one sequence only.

  Finally, based on *tracking status* information the selected activities can be processed in a particular way (ignored, highlighted, recorded in a dossier etc.). Such status information includes mastery status (pass, fail, score) and progress status (activity completion status, duration, time-spans, counts etc.).
- 3/ *Delivery* — determine if node's content can and may be delivered to the learner ("validation").

In a browser this may result in a selected activity to be shown on screen (content, task description etc.).

Delivery may follow a *delivery mode* associated with the activity. Such modes include “browse”, “review”, “normal”. This ensures that the same sequence and delivery can be processed in different “user roles”.

- 4/ *Rollup* — determine the “results” of a sequencing process. The rollup information is the accumulation of tracking information on sub activities.

  This could be visualized on screen as an updates score frame.

  The learning designer may explicitly define the way rollup information should be accumulated. *Rollup states* determine if the activity contributes to the rolled up date values of the parent. *Rollup rules* determine the way the rollup information should be accumulated. Examples of rollup rules are “passed if”, “failed if”, “completed if”, “incomplete if”. The conditions are expressed in terms of a predefined number of sub activities that meet a condition (“passed if 60% of sub activities passed”).

  Three statuses can be “rolled up”: Completion status (e.g. parent is complete when 2 children are complete), scores (e.g. parent scores grand total of child scores), mastery status, and duration (e.g. total duration is duration of selected activities).

The specification also introduces *control modes*. These are superimposed rules on how to interpret events and deliver content to the learner.

- Event-driven mode — sequences are traversed based on learner and navigation events.
- Completion-driven mode — sequences are traversed based on learning activity completion and exit.
- Selection-driven mode — sequences are traversed based on learner's choice.

2.9.3 Sequence definition model

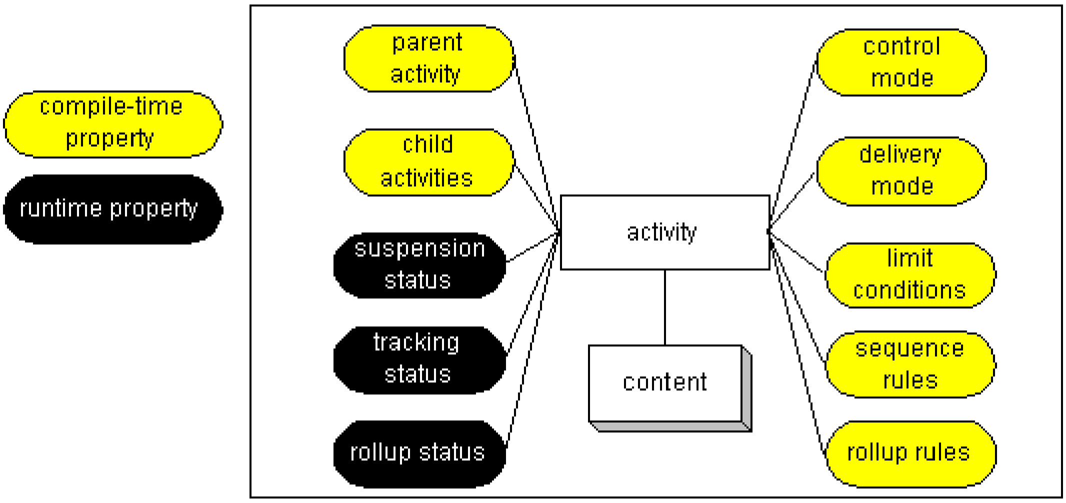


Figure 2-7. *Activity* overall structure.

The SS specification introduces the following “features” of sequences and sub-structures. The format is as follows:

- Optional multiplicity indicator (? = optional, * = zero or more, + is one or more)
- The name of the feature
- Possible values of the feature, defaults are placed between [...].

- Explanation of the feature.

The features are as follows.

2.9.4 Relation with other standards

Through content packaging relations exist with *IMS content packaging* and *ADL SCORM*. Through sequencing relations exists with EML (OUNL) and *IMS QTI.*

In *IMS LD* Simple Sequencing can be used to elaborate the sequencing of the item structures which occur at various places into IMS LD. In the activity structures of IMS LD specific placeholders for Simple Sequencing are included.

2.9.5 Applicability for ALFANET

At this stage for ALFANET there is no added value in applying Simple Sequencing. Its starting assumption is a single, individual user model excluding collaborative learning scenario's thus making it less usable within the context of ALFANET.

2.9.6 References

IMS Simple Sequencing Scope document 1.0. IMS global learning consortium, Inc., 2001.

Available at: http://www.imsproject.org/specifications.html

IMS Simple Sequencing Specification. Public draft 0.7.5, May 2002.

Available at: http://www.imsproject.org/specifications.html

2.10 IMS – Learning Design

2.10.1 General

The IMS Learning Design specification (IMS-LD) supports the use of a wide range of pedagogies in online learning. Rather than attempting to capture the specifics of many pedagogies, it does this by providing a generic and flexible language. This language is designed to enable many different pedagogies to be expressed. The approach has the advantage over alternatives in that only one set of learning design and runtime tools then need to be implemented in order to support the desired wide rage of pedagogies. The language was originally developed at the Open University of the Netherlands (OUNL), after extensive examination and comparison of a wide range of pedagogical approaches and their associated learning activities, and several iterations of the developing language to obtain a good balance between generality and pedagogic expressiveness.

IMS Learning Design is intended to be a "high level" specification describing the complete learning experience, including learner-learner communication and collaboration in group settings.

2.10.2 Unit of Learning = IMS Content Package + IMS Learning Design

The primary use of IMS Learning Design is to model *units of learning* by including an IMS Learning Design in a content package, preferably – but not necessarily - an IMS Content Package. It this specification it is assumed that IMS Learning Design is being used with IMS Content Packages to model units of learning. How this is done is explained in this section.

IMS Content Packages describe their contents in an XML document called the 'package manifest'. The Manifest may include structured 'views' into the resources contained in that package; each 'view' is described as a hierarchy of items called an 'organization'. Each item refers to a Resource that, in turn, can refer to a physical file within the package. It can however also refer to an external resource. Figure 2-8 depicts the entire IMS Content Packaging conceptual model.

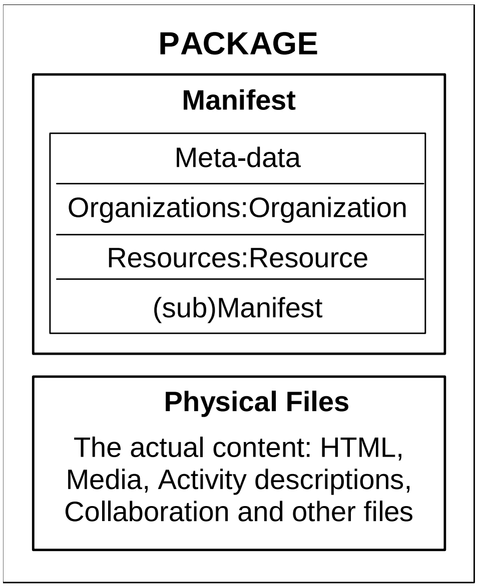


Figure 2-8. Structure of an IMS Content Package.

The Manifest is the information structure defined in the Content Packaging specification. It is contained within a package as an XML file with a fixed, pre-defined name (imsmanifest.xml). This enables it to be found amongst the many other content files that may be contained in a package.

The integration of a Learning Design into the Content Packaging Structure is set out in the next figure (Figure 2-9).

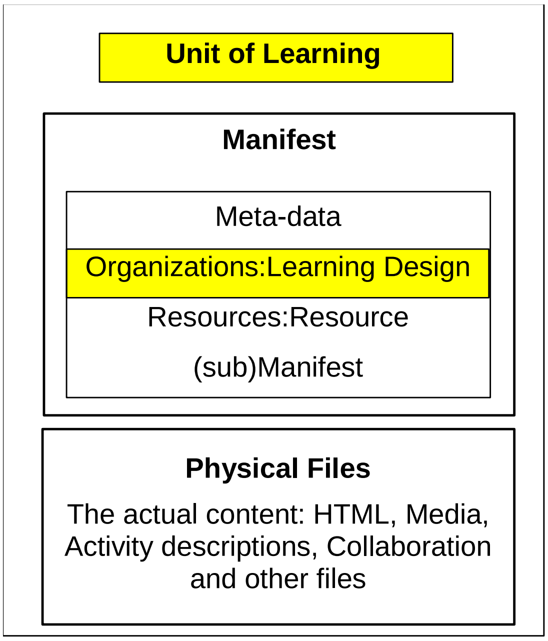


Figure 2-9. The structure of a Unit of Learning, composed by including an IMS Learning Design within the Organizations part of IMS Content Packaging

To create a unit of learning, IMS Learning Design is integrated with an IMS Content Package by including the learning design element as another kind of organization within the 'organizations' element, using the standard namespace for Learning Design. When the standard namespace is "[standard-namespace-for-learning-design]", then learning design elements are included as follows (ignoring irrelevant elements & attributes):

```
<manifest>
  <metadata/>
  <organizations>
     <learning-design xmlns="[standard-namespace-for-learning-design]">
         [add learning design elements here]
     </learning-design>
  </organizations>
  <resources/>
</manifest>
```

The italics have to be filled in with the appropriate namespace and elements respectively.

In a package that includes a learning design element, the optional organization element within organizations is ignored. This mechanism is in conformance with the extensibility mechanisms IMS Content Packages provide. If an organizations element contains a learning design element, any 'organization' element in the same organizations element is ignored and only the learning design element is read by the runtime system. Where other content organization elements are desired, they can be included in sub manifests, as sub packages may be aggregated in the same way as in normal content packages.

2.10.3 Three Levels of Implementation

Learning Design specifies three levels of implementation and compliance. This description is therefore partitioned to reflect this. However, each level will be mapped to separate XML Schemas.

Learning Design Level A includes everything described so far. It thus contains all the core vocabulary needed to support pedagogical diversity. Levels B and C add three additional concepts and their associated capabilities in order to support more sophisticated behaviours.

Learning Design Level B adds Properties and Conditions to level A, which will enable

personalization and more elaborate sequencing and interactions based on learner portfolio's. It can be used to direct the learning activities as well as record outcomes. The separation of Properties and Conditions into a separate Schema also enable it to be used independently of the rest of the Learning Design specification, typically as an enhancement to IMS Simple Sequencing.

Learning Design Level C adds Notification to level B, which, although a fairly small addition to the specification, adds significantly to the capability, but potentially also to the implementation task where something similar is not already in place.

The approach taken in this specification is therefore not to define a single large schema with a core of mandatory elements and numerous optional elements, but rather to define a complete core that is yet as simple as possible, and then to define two levels of extension that capture more sophisticated features and behaviours.

2.10.4 A framework for learning design

At level A, Learning Design specifies a time ordered series of *activities* to be performed by learners and teachers (*role*), within the context of an *environment* consisting of *learning objects* or *services* (see Figure 2-10 for an UML diagram that shows how the various terms used here hang together). Analysis of existing design approaches (see e.g. Koper, 2000, 2001, 2002) revealed that this was the common model behind all the different behaviorist, cognitive and (social) constructivist approaches to learning and instruction.

Most formal learning design strategies start reasoning from *learning objectives*, but one may also start from the *learning activities*, the *support activities* (usually provided by the teacher), or the *environment*. Often, a lot of design variables are already fixed and thus are constants in the design process. For instance, in most situations the *roles* are predetermined (student, teacher, mentor, assessor, …), and so is the global time schedule (e.g. semesters). Focusing on the knowledge transfer tradition, it is implicit that the *learning activities* always are variants on the theme: 'learn the knowledge provided'. In this case, one may concentrate on the question what knowledge and what test resources one should provide. In classroom teaching teacher activities are constrained by the possibilities the classroom affords. Etc.

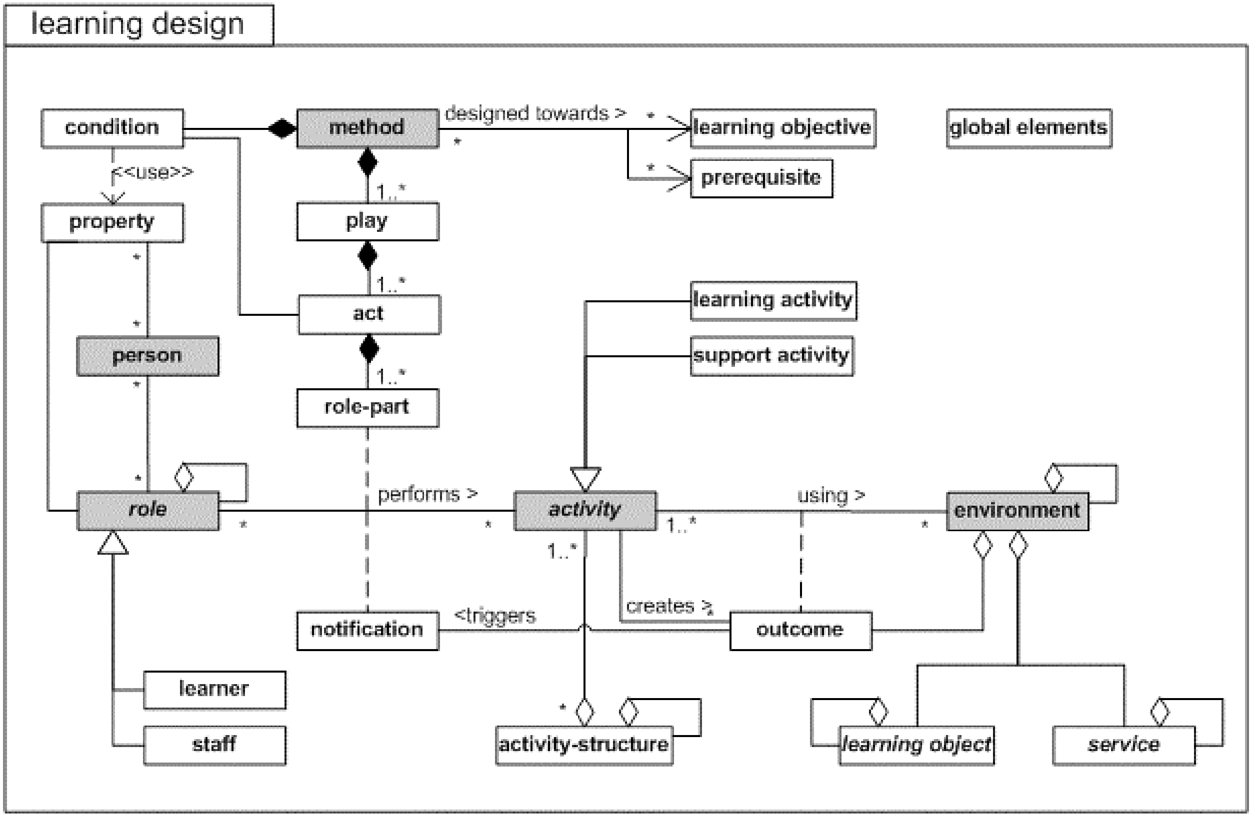


Figure 2-10. Conceptual model of overall learning design structure at level C; UML class diagram; major classes are greyed to enhance readability

For more advanced learning purposes, *properties* and *conditions*, and *notifications* are required. Levels B and C of the Learning Design specification provide these. *Properties*, specified at level B, are needed to store information about a *person* or a group of persons (*role*). So for a student its progress may be stored, perhaps in a dossier; for a teacher information on papers graded may be stored. *Conditions*, also part of level B, constrain the actual evolution of the didactic scenario. They are set in response to specific circumstances, preferences, or the characteristics of specific learners (e.g. prior knowledge). An example of a condition would be 'when the learner has learning style X, present the activities in random order'. The idea is of course that randomness allows the student to freely explore the materials. *Notifications*, specified in addition to the *properties* and *conditions* of level B at level C, are mechanisms to trigger new *activities*, based on an event during the learning process. For instance: the teacher is triggered to answer a question when a question of a student occurs; or the teacher should grade a report, once it has been submitted. Etc.

From the point of view of the Learning Design specification, the learning-design element is the top level element. However, a Learning design is typically (though not necessarily) embedded in an IMS Content Package, where it is placed with the Organizations element:

```
manifest
     metadata
     organizations
          learning-design
     resources
     manifests (submanifests of included packages)
```

It can therefore be seen as a more sophisticated alternative to the original Organization and item, which provides a hierarchy of tree-structure for the underlying content. Note that as content packages can be nested using embedded sub-manifests, when Learning Design is embedded in a content package, existing content packages can be reused and referenced from within the learning-design element. Learning Design can thus be seen as a higher level 'wrapper' for learning content and services that supports the coordination of multiple users and adds a number of other features. The information model illustrates in very general terms how to prepare a Content Package that contains a Learning Design.

2.10.5 Level A Information Model

2.10.5.1 Conceptual model

The conceptual UML model for level A is in Figure 2-11.

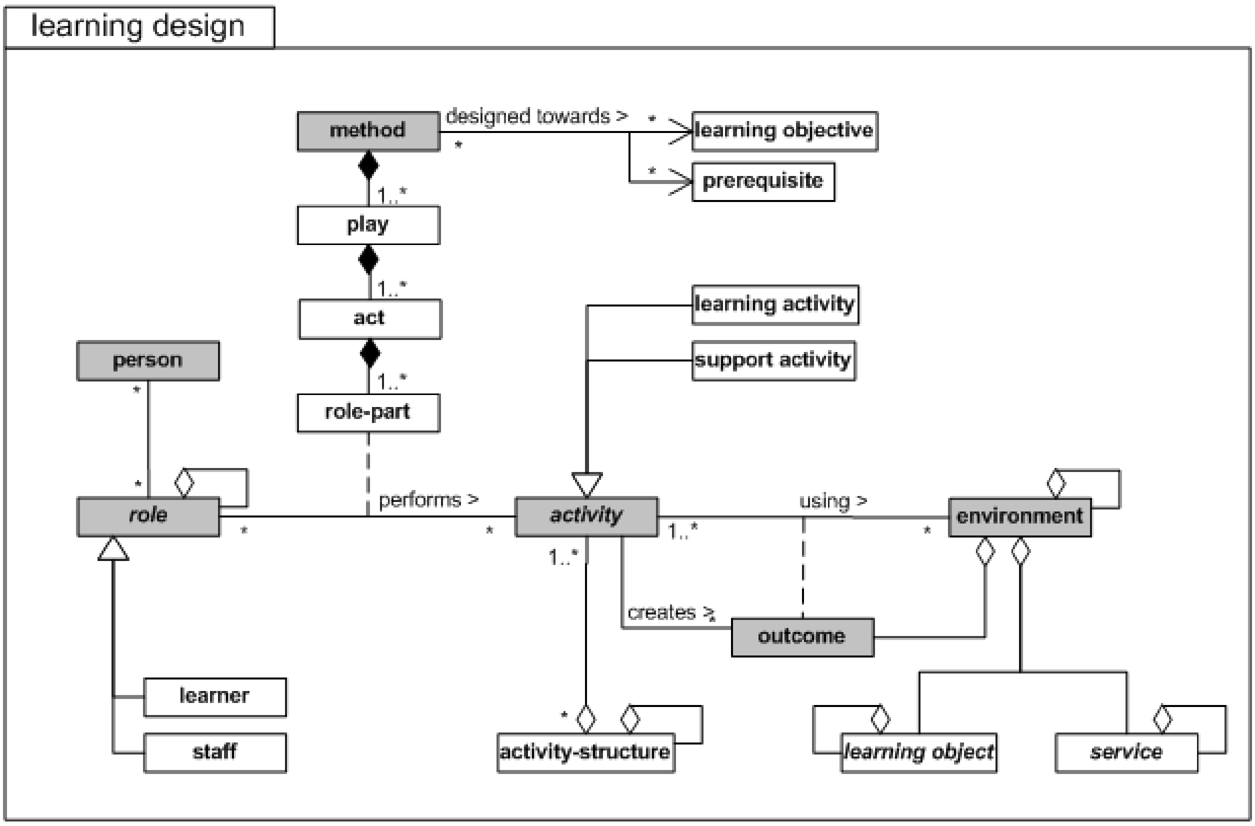


Figure 2-11. Conceptual model of level A.

2.10.5.2 Information schema 'learning-design'

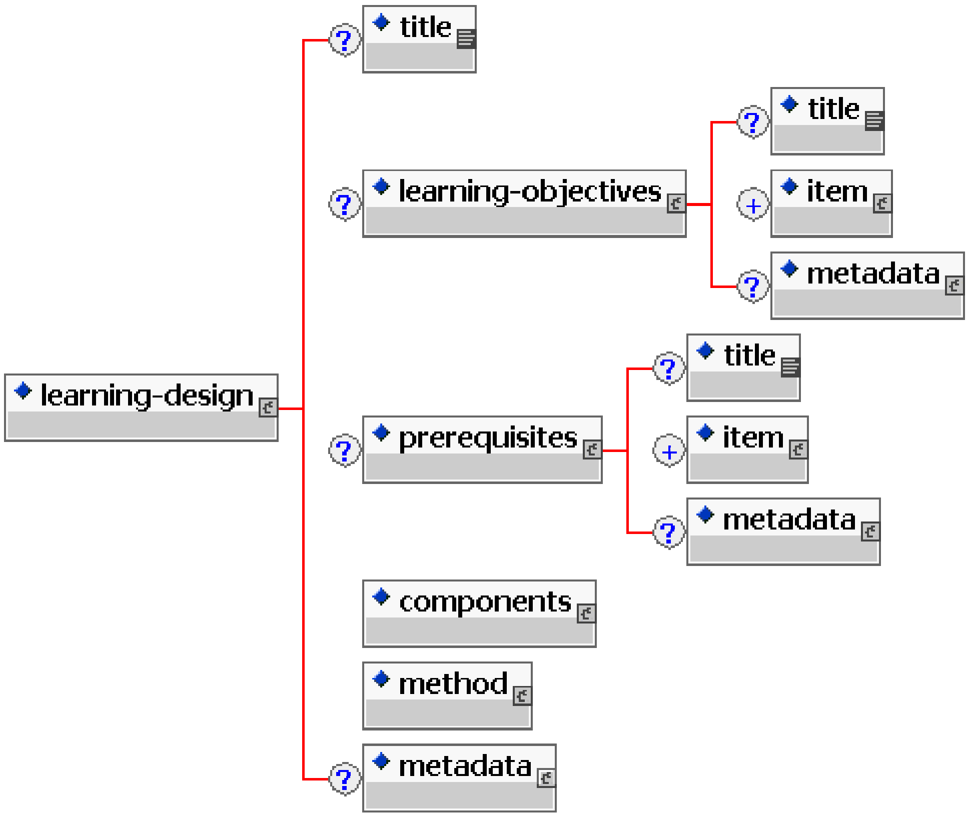


Figure 2-12. XML schema of Level A.

2.10.6 Level B Information Model

Level B provides additional elements, which significantly extend the ability of a learning designer to control the learning flow within a Unit of Learning. The main elements added are:

1. Properties
2. Conditions

The addition of properties and conditions affect different models:

1. The model of *components* is extended with the element *properties*, this is the place where the properties are declared.
2. The model of *complete-activity*, *complete-act*, *complete-play* and *complete-unit-of-learning* are extended to include the element *when-property-value-is-set*.
3. The model of *on-completion* is extended to include the element *change-property-value*.
4. The model of *service* is extended to include the element *monitor*.
5. The model of *email-data* is extended with two attributes (*email-property-ref* and *username-property-ref*) referring to global properties with data.
6. The model of *time-limit* is extended with one attribute (*property-ref*) referring to a property with data.
7. The element *method* is extended to include the element *conditions*.
8. The model of *complete-act* is extended to include the element *when-condition-true*.

9. A separate group of *global-elements* are included to read and set properties from all sorts of XML-based content schemas (e.g. XHTML).
10. Use is made of the W3C global attribute *class* to enable show and hide conditions on content elements in all sorts of XML-based content schemas (e.g. XHTML).

2.10.6.1 Conceptual model

The conceptual UML model for level B is in Figure 2-13. The grey marked classes are added to the model of level A.

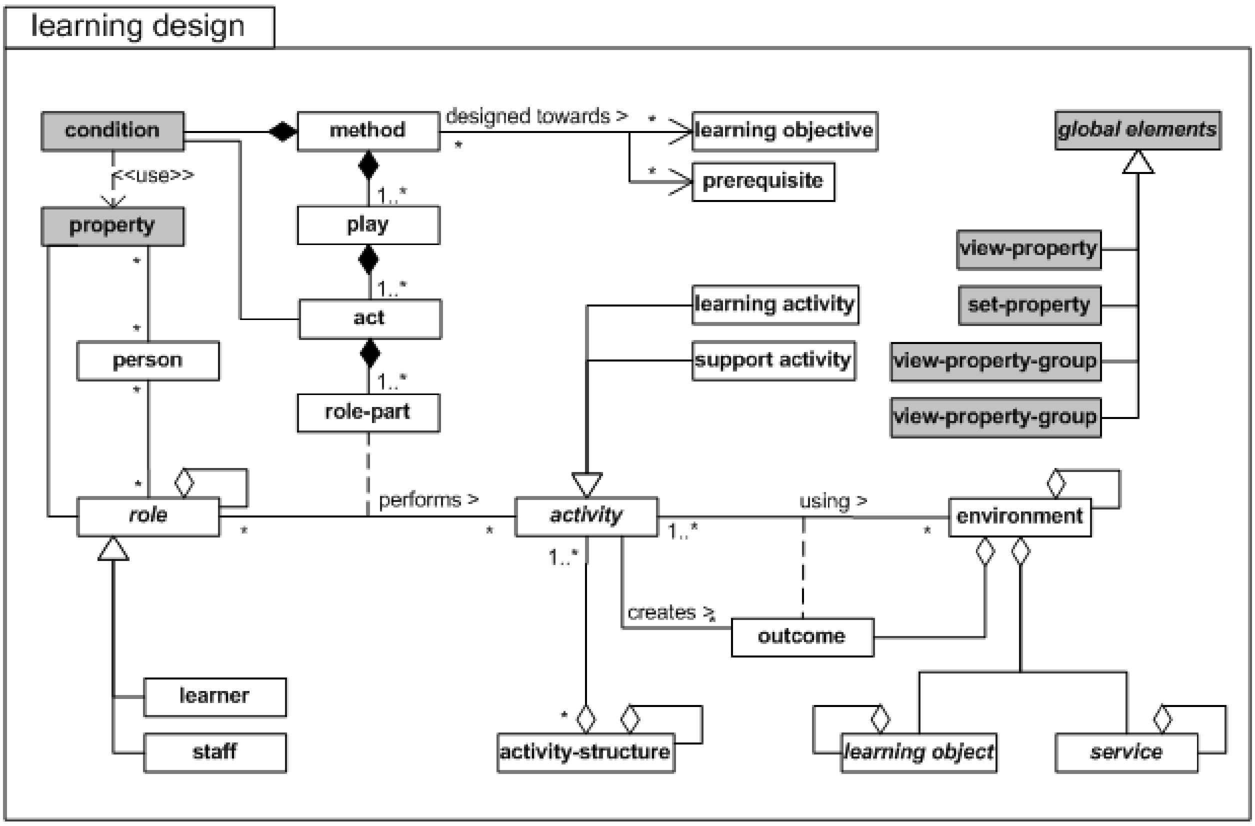


Figure 2-13. Conceptual model of level B

The runtime system, or 'user-agent' is expected to keep record of property-values and property-definitions for users and roles in a so-called 'dossier'.

Properties are defined and or declared (for already defined global properties) under learning-design/components/properties and operated upon with property-operation elements (view-property, set-property, conditions, change-property-value, etc.).

There are several types of properties.

1. *Local properties* (element name: *loc-property*) are stored with a scope local to the run of a unit of learning. They are defined and used in the unit-of-learning. The value of this property is the same for every user in the run of the unit-of-learning, but can differ in different runs.
2. *Global properties* (element name: *glob-property*) are accessible outside the context of a unit of learning (e.g. by more than one unit of learning). They can be defined in one unit of learning en used in another one. In IMSLD global properties can be defined. Runtimes are expected to control whether a defined global property URI already exists or not. Global properties - once defined - may never change definition. So when the property already exists the definition is ignored.

3. *Personal properties* (element name: *locpers-property* and *globpers-property*) are owned by a person (local or global). These properties are used for personalization. E.g. a portfolio that works across units of learning can be modeled with globpers (global personal) properties. The personal properties can be stored in a personal, portable 'dossier'.
4. *Role properties* (element name: *locrole-property*) are owned by a role and are always local. Every user in a specific role can access this property and has the same value in the same run of the unit of learning.

User-agents are expected to operate on properties in a secure way and with a maximum performance (to be detailed by the implementer).

The Scope of Global Properties

Global properties have to be maintained in a persistent storage. The organization or institution that controls the persistent storage effectively determines the scope of global properties by allowing or denying access to the storage.

Typically a runtime system will have access to the persistent storage. However, there may be a number of different runtime systems accessing the same storage. The scope of the global properties is therefore extended to all these runtime systems.

A distinction can be made between global personal properties and the generic global properties.

The generic global properties are typically under the control of the organization or institution that provides the learning, so the learning provider determines their scope.

If at some point in the future, there is worldwide access to learner's progress files, and these are used to maintain the data generated during learning activities, then the scope of global personal properties (globpers-property) is potentially actually global, assuming the runtime systems that a learner is concurrently using all have access to the same persistent data. An example might be a person who as an employee is undertaken training courses at work while in their own time is registered as a part-time distance learner in a university across the globe.

However the issues of architectures, security, ownership and control all need to be worked out and agreed on before this can happen and this is part of a larger problem that faces the uptake and use of the IMS LIP specification for lifelong learning.

So, for the near, and perhaps medium term future, personal learner information is likely to be maintained separately by each organization or institution that provide the learning (despite the problems this creates for lifelong learners). So for the time being, the learning provider will be likely to also determine the scope of global personal properties.

The other large issue is that of gaining widespread agreement as to the names, type and vocabulary of global learner properties that will allow them to be used across systems.

2.10.6.2 Information schema 'properties'

The element *properties* is added to the content model of the element *components*.

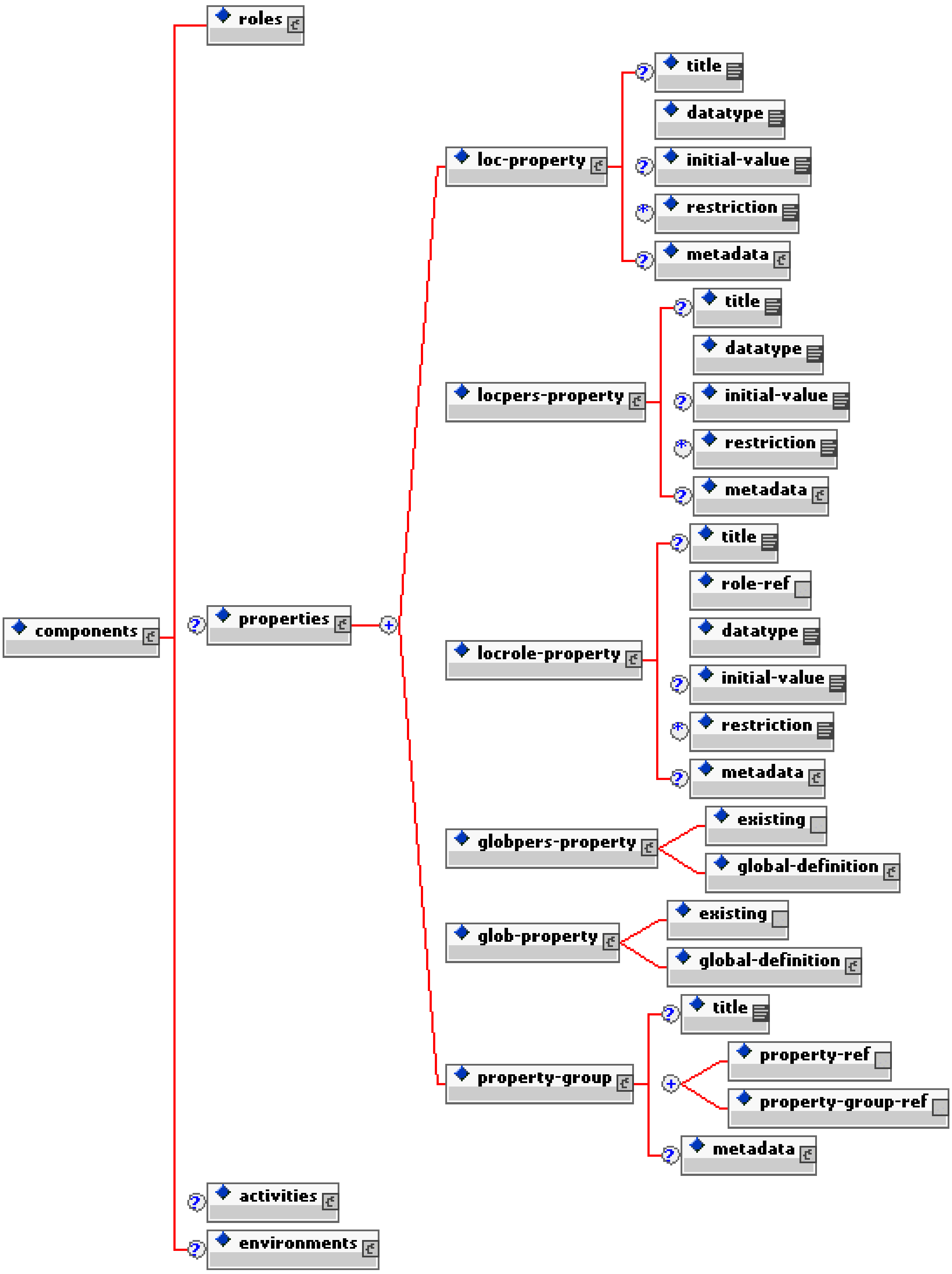


Figure 2-14. XML schema of Level B.

2.10.7 Level C Information Model

Level C adds the capability for a learning designer to specify the sending of messages and setting of new activities based on certain events. The runtime system, or 'user-agent' is expected to support a notification mechanism. Notifications are event driven mechanisms, which can be directed towards elements in the system or to human users.

Notifications affect the following content models of level B elements:

1. The *on-completion* model is extended with a notification element.
2. The *then* model is extended with a notification element.
3. Global elements *set-property* and *set-property-group* are both extended with a notification element.

2.10.7.1 Conceptual model

Figure 2-15 provides the conceptual UML model for level C. The grey marked class is added to the model of level B.

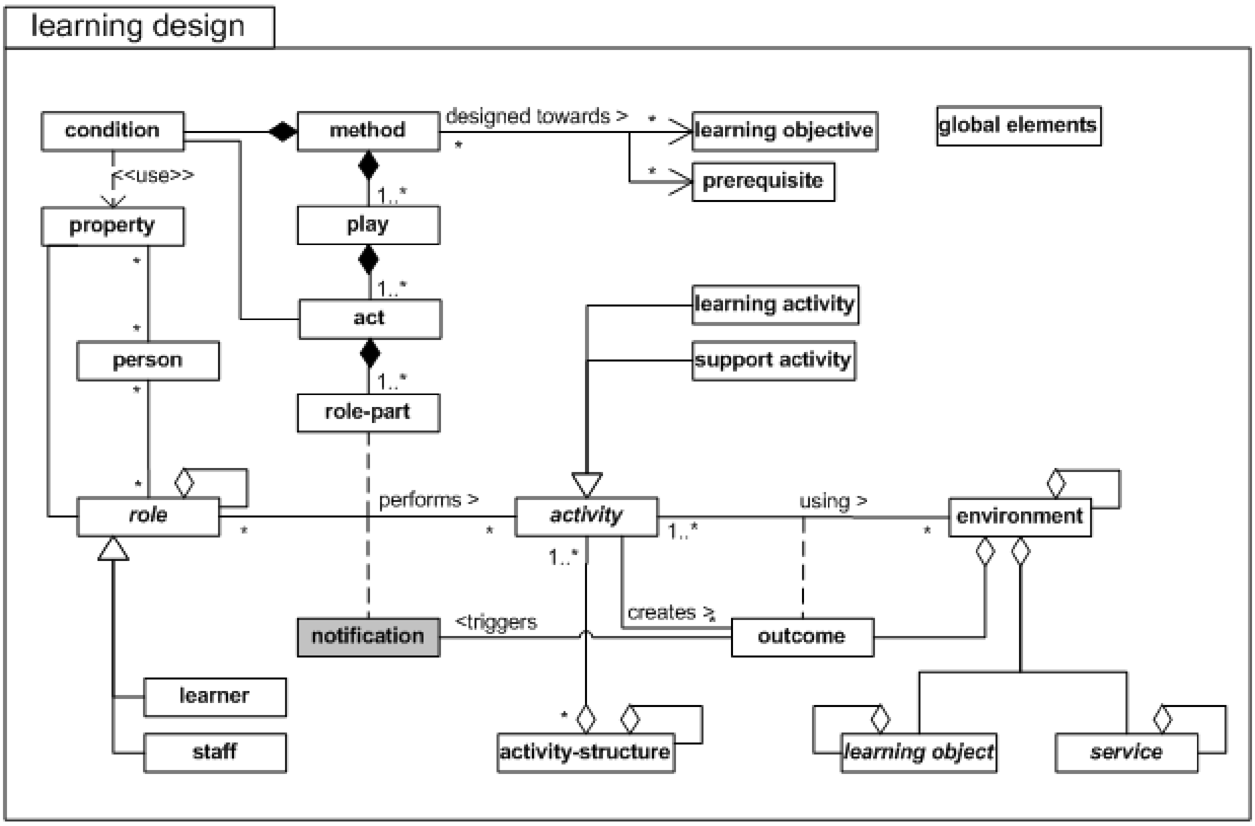


Figure 2-15. Conceptual model of level C.

2.10.7.2 Information schema 'notification'

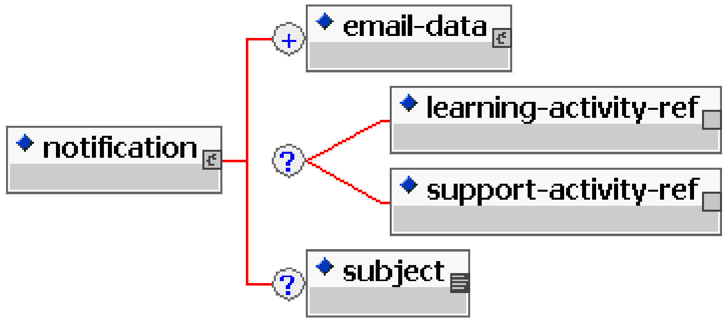


Figure 2-16. XML schema of Level C.

2.10.8 Relation with other standards

Learning Design is a specification for "describing learning and instructional design". The specification is based, at least in part, on the work on EML (Educational Modelling Language) by the Open University of the Netherlands (OUNL).

One of the main requirements for IMS-LD is that it must fit in with available standards and specifications (*compatibility).* Obviously*,* standards develop and equally this applies to IMS-LD. In other words the integration with other standards (or specifications) is defined by the version selected. The current public draft builds upon IMS Content Packages and hence IMS metadata, moreover it is possible to include SCORM content. For the final version it is proposed to optionally integrate IMS Simple Sequencing and to explore the integration of QTI.

2.10.9 Applicability for ALFANET

The objective of ALFANET to offer a highly adaptive, personalized learning experience including a variety of pedagogical methods requires the capability to model both structure and process, including the specification of roles and activities. IMS-LD (including Edubox) offers this capability and equally important in depth knowledge of this EML is available and directly accessible, so a quick start can easily be made. Moreover, IMS-LD can be considered as an integrative upper-layer to many existing specifications. Finally, the interest in and use of Educational Modelling Languages in general, and in particular in the OU EML, is growing:

The IMS Learning Design (LD) specification offers functionalities that no comparable system can offer, such as *re-usability*, *multiple roles* in collaboration and *personalised learning paths*. Creating effective learning material is costly, and using LD as the working language means that the material can automatically be adapted to different delivery platforms or incorporated into different courses, thus rendering it truly *re-usable*. LD has no inherent preconceptions of the underlying pedagogy. It can be used to describe conventional 'linear' courses, but equally well suited to describe much richer learning environments, such as collaborative learning exercises involving role playing with multiple roles. *Multiple roles* can be defined both for students and staff.

IMS-LD has the flexibility to support individual paths through the learning material. This functionality of *personalization* means that learners are not constrained, for example, to follow the single learning path defined by the course designer at the time of creation. For example, students can be presented with tests, and the outcomes can then be used to offer them further supporting material on a specific topic or to skip entire modules if they already have sufficient understanding.

All together this makes IMS-LD a logical candidate for inclusion in ALFANET.

2.10.10 References

Version 1 - Public Draft Specification - PDF

http://www.imsglobal.org/specificationdownload.cfm

2.11 ADL Sharable Content Reference Model (SCORM)

2.11.1 General

SCORM is a product of the *Advanced Distributed Learning (ADL)* initiative. The purpose of the ADL is to ensure access to high-quality *education and training materials* (resources) that can be tailored to individual learner needs (personalization) and made available whenever and wherever they are required (medium neutral). ADL provides a common technical framework for computer and net-based learning that will foster the creation of reusable learning content as "instructional objects", i.e. objects used for instructional purposes, in an instructional environment

The ADL vision can be summarized as follows.

> ADL development envisions the creation of learning “knowledge” libraries, or repositories, where learning objects may be accumulated and catalogued for broad distribution and use. These objects must be readily accessible across the World Wide Web, or whatever form our global information network takes in the future.
>
> It is expected that the development of such repositories will provide the basis for a new instructional object economy that rewards content creators for developing high quality learning objects and encourages the development of whole new classes of products and services that provide accessible, sharable and adaptive learning experiences to learners.
>
> The development of reusable, sharable learning objects is key to ADL’s long term vision.
>
> *[scorm-1.1]*

The *Sharable Content Object Reference Model (SCORM)* defines a 1/ Content Aggregation Model for learning, and a 2/ Runtime Environment for leaning objects.

- 1/ *Content Aggregation Model* — A model for collecting content objects such that is can be referenced, passed and reused in different learning environments. This is the basis for forming large repositories of learning resources.
- 2/ *Runtime Environment* — A specification of the interface to the actual software objects that access the learning materials. Defines API and data model for these objects.

This focus should support the construction and interoperation of learning management systems.

A *Learning Management System (LMS)* is a set of functionalities, possibly implemented in a (suite of) software tools(s), that deliver, track, report on and administer learning content, student progress, and student interactions. Such a system references content objects or aggregations of these as *Content Structure Format (CSF)* objects. The LMS is intended to track the learner. This includes gathering student profile information, delivering content to learner, monitoring interactions, and determining what the learner should do next.

In October 2001 version 1.2 of SCORM was released. This release of the SCORM adds specific SCORM Content Packaging application profiles derived from the IMS Content Packaging specification. These profiles map the Content Structure Format (CSF) from the SCORM Version 1.1 into the general IMS specifications.

This version of the SCORM also updates the meta-data section to refer to the latest work developed by the IMS Global Learning Consortium Inc and IEEE LTSC. The updates include changes to the information model and XML binding. This version of the SCORM also changed the names of the meta-data application profiles to better align with changes to the Content Aggregation Model for the SCORM Version 1.2 and in general with the IMS Content Packaging nomenclature.

2.11.2 Components

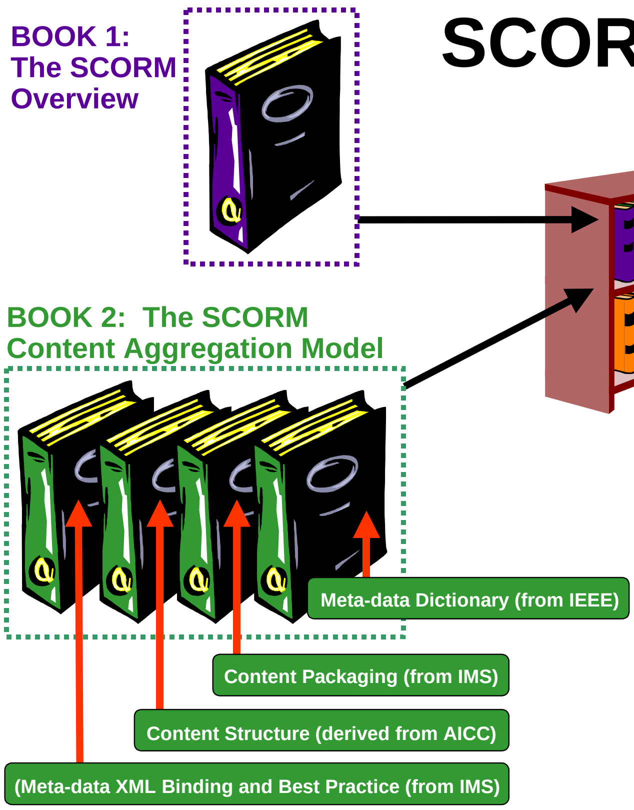


Figure: The SCORM as a collection of specifications.

SCORM1.2 is described as a collection of three types of books:

- Book 1 (Introduction) contains an overview of the ADL initiative, the rationale for the SCORM and a summary of the technical specifications and guidelines contained in the remaining sections.
- Book 2 (The SCORM Content Aggregation Model) contains guidance for identifying and aggregating resources into structured learning content. This book describes a nomenclature for learning content, describes the SCORM Content Packaging and references the IMS Learning Resource Meta-data Information Model, itself based on the IEEE Learning Technology Standards Committee (LTSC) Learning Objects Metadata (LOM) Specification that was developed as a result of a joint effort between the IMS Global Learning Consortium, Inc. and the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE). Together, these specifications form the SCORM Content Aggregation Model.
- Book 3 (The SCORM Run-Time Environment) includes guidance for launching, communicating with and tracking content in a Web-based environment. This book is derived from the run-time environment functionality defined in AICC's CMI001 Guidelines for Interoperability. ADL collaborated with AICC members and participants to develop a common *Launch* and *API* specification and to adopt the AICC Data Model for Web-based data elements.

2.11.3 Relation with other standards

ADL does not develop standards but SCORM references specifications and guidelines developed by other organizations and aims at adapting and integrating them with one another to form a more complete and easier to implement model.

The 1.2 version of the SCORM Content Aggregation Model (or CAM) includes several specifications from standardization bodies (see Figure above). These specifications are combined in the SCORM Meta-data Information Model.

The SCORM run-time environment is based on AICC specifications. A sample run-time environment is implemented and a set of tools is made available one can used to test conformance to SCORM 1.2.

2.11.4 Applicability for ALFANET

SCORM is a very well known initiative that:

- puts substantial effort into the integration of different specifications;
- collects and refines conformance tests;
- also includes specifications for a runtime environment.

However even with version 1.2, the scope of SCORM is too limited to be of much use for ALFANET. Version 1.3 is expected to include Learner Information (IMS-LIP) and Simple Sequencing (IMS-SS), but this version is not available yet. Learning Design (IMS-LD) is not mentioned yet.

2.11.5 References

Advanced Distributed Learning Sharable Content Object Reference Model Version 1.2 (October, 2001). Available at http://www.adlnet.org

Sharable Content Object Reference Model (SCORM) Packaging Application Profiles. Version 1.0 DRAFT (April 20, 2001)

2.12 AICC Computer Managed Instruction (CMI)

2.12.1 General

The Aviation Industry CBT (Computer-Based Training) Committee (AICC) is an international association of technology-based training professionals. The AICC develops guidelines for aviation industry in the development, delivery, and evaluation of CBT and related training technologies. The AICC CMI subcommittee introduces the concept of *Computer-managed Instruction (CMI)*. CMI systems manage both courseware and students in a training environment. It is primarily a scheduler of CBT materials. Through *[CMI001]* the project offers guidelines on 1/ Communication between a CMI system and a lesson, 2/ Moving a course between different CMI systems, 3/ Storing lesson evaluation data. Guidelines for the format and content of files are also described.

The relation between CMI and CBT can be depicted as follows.

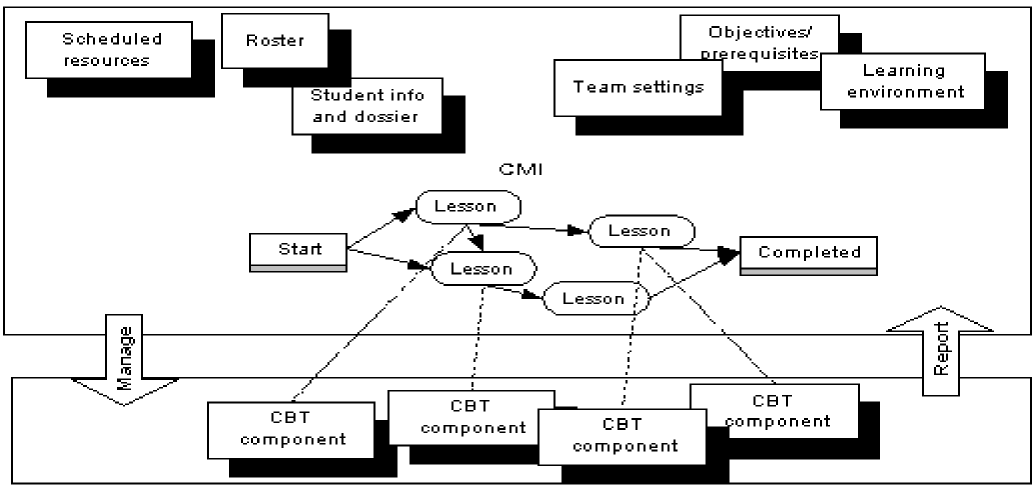


Figure 2-17. Relation between CMI and CBT.

While the CBT is focussed on providing information to a learner in an interactive way, the CMI deals with sequencing of lessons, prerequisites and objectives, rostering and student assignment, and feedback analysis. It can also provide a basis for testing.

2.12.2 Relation with other standards

AICC is coordinating activities with ADL and IEEE LTSC to develop learning technology standards. However AICC only provides recommendations to develop various learning content types for computer managed instruction systems and recommendations for the courseware delivery stations and peripherals.

2.12.3 Applicability for ALFANET

AICC itself does not provide any learning technology specification, these are provided by ADL and IEEE LTSC. Therefore AICC recommendations are not directly applicable for the interoperability of the ALFANET system but they could be taken into account during the design phase of ALFANET.

2.12.4 References

AICC web. http://www.aicc.org

3. Knowledge management standards

3.1 General overview

Thomas Davenport et all [Davenport and Prusak, 1998] says that 'Knowledge management is concerned with the exploitation and development of the knowledge assets of an organization with a view to furthering the organization's objectives. The knowledge to be managed includes explicit, documented knowledge and tacit, subjective knowledge. Management of this knowledge entails all the processes associated with the identification, sharing and creation of knowledge. This requires systems for the creation and maintenance of knowledge repositories, and to cultivate and facilitate the sharing of knowledge and organization learning. Organizations that succeed in knowledge management are likely to view knowledge as an asset and to develop organizational norms and values, which support the creation, and sharing of knowledge.'

From the multiple definitions existing from the term, we can extract the following main characteristics:

- KM relates to both theory and practice
- Definitions are not predicated on information technology
- KM is multi-disciplinary
- People and learning issues are central to KM
- Technology is a useful enabler rather than a central tenet at the heart of KM.

There are different initiatives developing standards for KM [EKMF, 2001]:

- The Standards Australia International (SAI) aims to describe the overal concept of KM in a **KM framework**.
- The Global Knowledge Economics Council (GKEC), in US, has published a proposal for candidate terms and definitions for a **KM vocabulary** based on definitions from the Organization of Economic Cooperation and Development (OECD)and aim to receive accreditation through ANSI the ISO standard development.
- In UK, the British Standards Institution (BSI) has initiated a committee for the development of KM standards.
- In Germany, a special committee of the DIN has listed KM as a subject to be investigated for its relevance for RTD driven standardisation

- The European Knowledge Management Forum (EKMF) is building up a KM community in Europe, which aims to support commonality in KM terminology, application and implementation: to share the latest developments in the KM domain and to define open standards and common approaches to KM for making it known and applicable to a broad European business public.
  The work performed suggests that the most widely supported needs are for a KM Framework, a multilingual KM Terminology and the development of an Implementation Methodology. The EKMF paper on *Knowledge Management Standardisation* includes a Current Standards Activities Map and suggests a potential development roadmap with a 10 year timeframe.

Next, we analyse the standards related with the knowledge representation (ontologies and related standards) and different approaches about skills, competencies and curricula standards.

3.2 Ontologies and related standards

Glossaries, Specialized dictionaries, Standard terminology lists, Reference data, Authority files, Classification schemas, Taxonomies, Thesauri, Ontologies are some techniques used in the knowledge representation.

A controlled vocabulary is a standard system of terminology used for coding, classifying, or otherwise uniquely identifying data and information.

A **taxonomy** is a classification system that divides a subject area hierarchically into progressively smaller subdivisions. Taxonomies have been used for years to classify many forms of knowledge: products and services in 'yellow pages'; web sites in directories like Yahoo; books in library subject areas. Blooms' Taxonomy of Educational Objectives [Bloom, 1957] is an example of the educational field, categorizing level of abstraction of questions that commonly occur in educational settings.

A **thesaurus** is basically a network of interrelated terms within a particular domain, and although it will often contain other information (such as definitions, examples of usage, etc.), the key feature of a thesaurus is the relationships, or associations, between terms. Given a particular term, a thesaurus will indicate which other terms mean the same, which terms denote a broader category of the same kind of thing, which denote a narrower category, and which are related in some other way. Commonly used association types like “broader term”, “narrower term”, “used for” and “related term” are defined in standards for thesauri such as Z39.19, ISO 5964 and ISO 2788.

One widely used knowledge representation formalism In the field of AI (Artificial Intelligence) is that of *conceptual graphs*, whose building blocks are *concepts* and *conceptual relations*.

3.2.1 Ontology

From existing bibliograph ([Gruber, 1993], [Guarino, 1998], [Pérez-lópez, 1999]) about ontologies we conclude some sentences.

An ontology is an explicit specification of a conceptualisation.

The use of an ontology serves for:

- Use and re-use existing information sources
- Locate, gather, monitor and retrieve relevant information
- Fuse content from disparate sources

There are different kinds of ontology according to their level of generality:

- *Top-level ontologies* describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain: it seems therefore reasonable, at least in theory, to have unified top-level ontologies for large communities of users.
- *Domain ontologies* and *task ontologies* describe, respectively, the vocabulary related to a generic domain (like medicine, or automobiles) or a generic task or activity (like diagnosing or selling), by specializing the terms introduced in the top-level ontology.
- *Application ontologies* describe concepts depending both on a particular domain and task, which are often specializations of *both* the related ontologies. These concepts often correspond to *roles* played by domain entities while performing a certain activity, like *replaceable unit* or *spare component*.

Education Ontologies are analysed in Deliverable D12. [ALFANET-D12, 2002] of this project, in the appendix 11.

Some top-level ontologies are Cycorp Ontology [CYCORP, 2002] and Standard Upper Ontology (SUO) that we describe in next section.

There are several initiatives of standardization to provide a unified model for representing knowledge and linking it with the information resources, and to define ontologies.

Topic maps is introduced by the Davenport Group. One of their main applications is that of solving the findability problem of information, that is: how to find the information you are looking for in large body of information. Topic maps can also be used for knowledge management, for web portal development, content management, and enterprise application integration (EAI). Topic maps are also being described as an enabling technology for the semantic web. TopicMaps.Org produced the XTM (XML Topic Maps) syntax for topic maps, which is used by nearly all topic map software today

RDF (Resource Description Framework) was developed by the W3C (World Wide Web Consortium) as part of its semantic web effort. It is mainly intended for use in the semantic web, but it is also being described variously as a content management technology, a knowledge management technology, a portal technology, and also as one of the pillars of e-commerce.

Resource Description Framework [Bray, 2001] is a framework for describing and interchanging metadata. It is built on the following rules.

1. A **Resource** is anything that can have a URI; this includes all the Web's pages, as well as individual elements of an XML document. An example of a resource is a draft of a document and its URL is http://www.textuality.com/RDF/Why.html

2. A **Property** is a Resource that has a name and can be used as a property, for example `Author` or `Title`. In many cases, all we really care about is the name; but a Property needs to be a resource so that it can have its own properties.

3. A **Statement** consists of the combination of a Resource, a Property, and a value. These parts are known as the 'subject', 'predicate' and 'object' of a Statement. An example Statement is "The Author of http://www.textuality.com/RDF/Why.html is Tim Bray." The value can just be a string, for example "Tim Bray" in the previous example, or it can be another resource, for example "The Home-Page of http://www.textuality.com/RDF/Why.html is http://www.textuality.com."

4. There is a straightforward method for expressing these abstract Properties in XML

OIL (Ontology Inference Layer) is an initiative funded by the European Union programme for Information Society Technologies. OIL is a semantic web technology based on RDF, and is intended to solve the findability problem, support e-commerce, and enable knowledge management.

An OIL ontology [Ontoknowledge, 2002] consists of the actual *ontology definition*, defining a particular ontological vocabulary, preceded by an *ontology container*, which is concerned with describing features of such an ontology and is based on Dublin Core Metadata Element Set (Version 1.1) standard.

An OIL ontology definition contains descriptions of classes, slots, and individuals.

- **Classes** are collections of objects. They are unary predicates such as person.
- Classes may be related to other classes by stating that one is a **subclass** of another.
- Classes will typically also contain information about how their members relate to other objects. **Slots** are binary relations. They may also be related to each other via the notion of sub relation. Slots may also be constrained through the use of slot-constraints. One can limit the type of a slot filler by stating that it must be of a particular **type**. One can also define **cardinality restrictions**.
- Term descriptions may also be combined by using Boolean connectives to form complex Boolean expressions.

DAML (DARPA Agent Markup Language) was created as part of a research program started in August 2000 by DARPA, a US governmental research organization. DAML seems to be entirely focused on supporting the semantic web, and is based on OIL. DAML provides a rich set of constructs with which to create ontologies and to markup information so that it is machine readable and understandable.

The **OWL Web Ontology Language** is being designed by the W3C Web Ontology Working Group in order to provide a language that can be used for applications that need to understand the content of information instead of just understanding the human-readable presentation of content. The OWL language can be used to allow the explicit representation of term vocabularies and the relationships between entities in these vocabularies. In this way, the language goes beyond XML, RDF and RDF-S in allowing greater machine readable content on the web. The OWL language is a revision of the DAML+OIL web ontology language incorporating learnings from the design and application use of DAML+OIL.

The *31 July 2002* the Web Ontology Working Group has released three first Working Drafts. The Feature Synopsis, Abstract Syntax and Language Reference describe the OWL Web Ontology Language 1.0 and its subset OWL Lite. Automated tools can use common sets of terms called ontologies to power services such as more accurate Web search, intelligent software agents, and knowledge management. OWL is used to publish and share ontologies on the Web. [OWL, 2002]

Currently there are some efforts to achieve a consensus between the two communities (Topic Maps and W3C Semantic Web related efforts RDF, OIL, DAML, OWL): experts are working to validate the approach consisting of using RDF to represent Topic Maps [Garshol, 2002].

RDF is very similar to a basic directed graph

OIL provide more sophisticated classification, using constructs from frame-based AI.

DAML+OIL: a language for expressing far more sophisticated classifications and properties of resources than RDFS, which also adds facilities for data typing based on the type definitions provided in the W3C XML Schema Definition Language (XSDL) and gives designers more expressiveness in classifying resources.

The general perception is that Topic Maps is higher-level that RDF, OIL and DAML. We provide a deeper analysis of Topic Maps below.

3.2.2 Standard Upper Ontology (SUO)

The IEEE Standard Upper Ontology (SUO) Study Group aims to develop a standard which will specify the semantics of a general-purpose upper level ontology. It will enable computers to utilize it for applications such as data interoperability, information search and retrieval, automated inferencing, and natural language processing. An ontology consists of a set of concepts, axioms, and relationships that describe a domain of interest. An upper ontology is limited to concepts that are meta, generic, abstract and philosophical, and there-fore are general enough to address (at a high level) a broad range of domain areas. [IEEE SUO 2001].

This standard will specify an upper ontology that will enable computers to utilize it for applications such as data interoperability, information search and retrieval, automated inferencing, and natural language processing. An ontology is similar to a dictionary or glossary, but with greater detail and structure that enables computers to process its content. (it is repeated in the previous paragraph)Concepts specific to given domains will not be included; however, this standard will provide a structure and a set of general concepts upon which domain ontologies (e.g. medical, financial, engineering, etc.) could be constructed.

A. AUTOMATED REASONING: The standard will be suitable for automated logical inference to support knowledge-based reasoning applications.

B. INTER-OPERABILITY: The standard will provide a basis for achieving Inter-Operability among various software and database applications.

1) Application developers can define new data elements in terms of a common ontology, and thereby gain some degree of interoperability with other conformant systems.

2) Applications based on domain-specific ontologies that are compliant with this standard will be able to interoperate (to some degree) by virtue of the shared common terms and definitions.

3) The SUO will play the role of a neutral interchange format whereby owners of existing applications will be able to map existing data elements just once to a common ontology. This provides a degree of interoperability with other applications whose representations conform to SUO. This entails the SUO being able to be mapped to more restricted forms such as XML, database schema, or object oriented schema.

C: APPLICATION AREAS Between the application areas the group mentions "Educational applications in which students learn concepts and relationships directly from, or expressed in terms of, a common ontology. This will also enable a standard record of learning to be kept."

The Standard Upper Ontology (SUO) will provide definitions for general-purpose terms, and it will act as a foundation for more specific domain ontologies. It is estimated that it will eventually contain between 1000 and 2500 terms and roughly ten definitional statements for each term. The SUO will have a variety of purposes, some of which can be glossed as follows [Niles, 2001].

- Design of new knowledge bases and databases. Developers can craft new knowledge and define new data elements in terms of a common ontology, and thereby gain some degree of interoperability with other compliant systems.
- Reuse/integration of legacy databases. Data elements from existing systems can be mapped just once to a common ontology.
- Integration of domain-specific ontologies. Such ontologies (if they are compliant with the SUO) will be able to interoperate (to some degree) by virtue of shared terms and definitions.

3.2.3 Topic Maps

Topic maps [Ontopia, 2002] are a ISO standard for describing knowledge structures and associating them with information resources. As such they constitute an enabling technology for knowledge management. Dubbed “the GPS of the information universe”, topic maps are also destined to provide powerful new ways of navigating large and interconnected corpora, making possible to represent immensely complex structures.

The basic concepts of the model — Topics, Associations, and Occurrences (TAO) —

Topics are the most fundamental concept, are the "things" itself. A topic is an object within a topic map that represents a subject. Topics can be categorized according to their kind. In a topic map, any given topic is an instance of zero or more **topic types**. Topics have three kinds of characteristics: names, occurrences, and roles in associations.

A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called **occurrences** of the topic.

A topic **association** asserts a relationship between two or more topics. Associations between topics can be grouped according to their type.

Topic maps started life as a way of representing the knowledge structures inherent in traditional back of book indexes, in order to solve the information management problems involved in creating, maintaining and processing indexes for complex documentation. As the model evolved, their scope was broadened to encompass other kinds of navigational aid, such as glossaries, thesauri and cross references.

However, instead of simply replicating the features of a printed index, the topic map model *generalizes* them, extending them in many directions at once and thereby enabling navigation in hitherto undreamt of ways. With topic maps a user can wander at leisure through a multidimensional topic space of knowledge before deciding which information resources are relevant, instead of wading through volumes or megabytes of data in order to find what he or she is looking for. Similarly, queries based on topic maps can be much more accurate than simple full text searching. From being a useful but often underused adjunct to the main body of information, indexes (when based on topic maps) look set to become the *sine qua non* of information delivery and consumption.

The generality and expressive power of the topic map model bring with it other advantages that go far beyond those traditionally associated with indexes. The close similarity to semantic nets gives an idea of how topic maps, even without any occurrences connecting them to an information pool, can become valuable resources in their own right.

The ability to encode arbitrarily complex knowledge structures and link them to information assets indicates a major role for topic maps in the realm of knowledge management: Topic maps can be used to represent the interrelation of roles, products, procedures, etc. that constitute corporate memory, and link them to the corresponding documentation.

They enable multiple alternative models of knowledge domains to coexist, and to work together, in a way that has not been available before. Topic Maps are capable of supporting and revealing immensely complex interrelationships within and among the concepts related to various fields of endeavour, and to provide master indexes to arbitrarily large and comprehensive bodies of information. http://www.infoloom.com/tmfaq.htm

The XTM (XML Topic Maps) specification provides a model and grammar for representing the structure of information resources used to define topics, and the associations (relationships) between topics. Names, resources, and relationships are said to be *characteristics* of abstract subjects, which are called *topics*. Topics have their characteristics within *scopes*: i.e. the limited contexts within which the names and resources are regarded as their name, resource, and relationship characteristics. One or more interrelated documents employing this grammar is called a “topic map.” http://www.topicmaps.org/xtm/index.html

The XML-based interchange syntax developed as part of XML Topic Maps has been adopted as part of the ISO standard, and new work items have been approved by ISO for a Topic Map Query Language (TMQL) and a Topic Map Constraint Language (TMCL). ISO 13250 itself is being augmented by a Reference Model and a Standard Application Model, and a number of OASIS technical committees are working on published subjects. Finally, a lot of effort is being put into understanding the relationship between topic maps and RDF, the metadata framework developed by the W3C.

Topic Map technology [Moore, 2000] have potential applications in three areas [Ahmed, 2000]: individual and shared workspaces and knowledge management:

Individual workspaces

Using a Topic Map to create an individual workspace gives the user a means of better managing access to frequently used documents and to organise data in multiple ways. Topic Maps can be used to create logical paths from an abstract concept to a specific document in a way which more closely matches the way the user thinks. Tools are needed to make the construction, maintenance and navigation of such Topic Maps as easy as possible and to integrate as tightly as possible with the day-to-day tools and processes. Topic Maps allow a user to relate single data instances to multiple subject areas - such as a standard text referenced from multiple projects. Topic Maps also give the application the freedom to link to resources in other tools (such as email, PIM systems and remote documents) - enabling the user to pull information from many disparate sources into a single coherent set for their use.

Tools are already available that aid in this form of personal organisation. Topic Maps may be used as an interchange format between such products and/or platforms - for example moving my mind map from my PC to my Palm and back or creating a 'mobile' workspace on an Internet-accessible site that can travel with me.

Shared workspaces

Shared workspaces enable users to share knowledge by communicating to each other the associations and relationships between data instances. Multiple Topic Maps may be combined with relatively little effort, to quickly generate a composite view of the same data set. Topic Maps created by individual users can thus be shared across an organisation, enabling many other users to gain the insights and benefit from the knowledge encoded in the Topic Map. As with any Topic Map application, data instances may be in a repository or located elsewhere within or outside the organisation - as long as it can be addressed in some way.

When user share their workspaces, Topic Map merging rules and applying additional scoping using added themes can be used to ensure that the perspective of different people are combined only to the degree desired by the end-user.

Knowledge management

Topic Maps can be used to encode ontologies prepared by one or more subject matter experts. Such a map may be used simply to transfer an ontology from one tool to another, or as a 'publishing medium' for an ontology. A topic map engine combined with other analysis tools (such as linguistic analysis tools) could be used to automatically annotate documents according to a given ontology and record the resulting annotation as a Topic Map. Again, Topic Map merging rules could be used to generate composite or comparative views of the same data set using different ontologies or analysis methods.

3.3 Skills, Competencies, Curricula

3.3.1 IBSTPI Competencies

The International Board of Standards for Training, Performance and Instruction [IBSTPI, 2002] has issued competencies and performance statements for Instructors, Training Managers and Instructional Designers.

Instructor Competencies

There are 14 core competencies and each one has between four and nine performance statements. They define the generic instructor role, independent of settings and organizations.

The current list of Instructor Competencies is given below

1. Analyse course materials and learner information.
2. Assure preparation of the instructional site.
3. Establish and maintain instructor credibility.
4. Manage the learning environment.
5. Demonstrate effective communication skills.
6. Demonstrate effective presentation skills.
7. Demonstrate effective questioning skills and techniques.
8. Respond appropriately to learners' needs for clarification or feedback.
9. Provide positive reinforcement and motivational incentives.
10. Use instructional methods appropriately.
11. Use media effectively.
12. Evaluate learner performance.
13. Evaluate delivery of instruction.
14. Report evaluation information.

New reviewed competencies are expected to be published late 2002. They will include competencies for on-line teachers and trainers as well as for classroom-based instructors.

Related work: Competencies for Online Teaching [Spector, 2001].

3.3.2 Information and Technology Literacy Standards

The Information and Technology Literacy Standards [DPI, 1998] identify and define the knowledge and skills essential for all Wisconsin students to access, evaluate, and use information and technology. The purpose of these standards is to identify information and technology competencies for all students throughout the pre-kindergarten to grade twelve (PK-12) curriculum. They are grouped into four categories or content standards specifying what a student should know and be able to do. The first two content standards focus on technology use and information processing skills. The latter two build upon the initial categories by adding competencies that deal with attitudes, appreciation, independent learning, teamwork skills, and personal and social responsibility.

A. MEDIA AND TECHNOLOGY - select and use media and technology to access, organize, create, and communicate information for solving problems and constructing new knowledge, products, and systems.

B. INFORMATION AND INQUIRY - access, evaluate, and apply information efficiently and effectively from a variety of sources in print, non-print, and electronic formats to meet personal and academic needs.

C. INDEPENDENT LEARNING - apply technological and information skills to issues of personal and academic interest by actively and independently seeking information, demonstrating critical and discriminating reading, listening, and viewing habits, and striving for personal excellence in learning and career pursuits.

D. LEARNING COMMUNITY - demonstrate the ability to work collaboratively in teams or groups, use information and technology in a responsible manner, respect intellectual property rights, and recognize the importance of intellectual freedom and access to information in a democratic society.

Each content standard is divided into performance standards that tell how students will show that they are meeting a standard. Each performance standard includes a number of indicators that detail how students will demonstrate proficiency in a particular performance area.

3.3.3 NCPQ Standards and Indicators: Curricula

The US National Consortium for Product Quality [NCPQ, 1995] developed a list of Standards to guide the Curriculum Review process.

Content Standard School-to-Work curricula must focus on the integration of academic foundations into career development, life skills, and occupational competencies.

Instructional Standard School-to-Work curricula, through active and applied learning experiences in school, community, and work-based settings, must enable students to acquire problem-solving, communication, and reasoning strategies.

- To what extent do the instructional strategies include active and meaningful learning experiences that correspond to stated student outcomes?
- To what extent do the instructional strategies include teaching techniques that enhance the SCANS thinking skills: creative thinking, decision making, problem solving, seeing things in the mind's eye, knowing how to learn, and reasoning?
- To what extent can the suggested instructional strategies be adapted to different learning styles?
- To what extent do the instructional strategies (ie, activities and projects) reflect the diversity of today's workforce?
- To what extent do the instructional strategies incorporate team or small group projects?
- To what extent do the instructional strategies encourage students to interact with each other, instructors, and the community?
- To what extent do the instructional strategies develop students' critical thinking and problem solving skills?
- To what extent do the instructional strategies develop students' skills of writing, speaking, listening, and following directions?
- To what extent do the instructional strategies provide the students with real-world experiences (both in and out of the classroom) which reinforce academic and technology applications?

Student Assessment Standard Assessments within School-to-Work curricula must be student-focused in measuring attitudes, knowledge, and skills, as well as their application to problem solving within the classroom and workplace environment.

- To what extent are student teams, as well as the individual student, assessed?
- To what extent does the assessment tool(s) measure the attitude, knowledge, and/or skill presented in the material?
- To what extent does the assessment process include feedback and alternative testing opportunities?
- To what extent are performance and portfolio assessments used to measure student knowledge and skills?
- To what extent can the assessments detect change over time?
- To what extent are appropriate assessment methods provided that directly reflect student outcomes?

Equity and Diversity Standard School-to-Work curricula must reflect content which portrays and celebrates the active participation of all individuals in the nation's workforce, communities, and educational institutions.

3.4 Applicability for ALFANET

Though the application of ontologies for ALFANET is not clear at this moment they might become useful at a later stage.

Ontologies related standards

- Integration with external KM tools (generate and export new knowledge; import existing knowledge)
- Generation of new knowledge and classification of existing items based on the concepts defined. Express additional knowledge structures on top of original resources
- Means to structure and access the knowledge in a repository. Explicit ontologies are the most promising technical vehicle for transforming document repositories into proper knowledge repositories.

Skills, Competencies, Curricula

- Serve as a check list on the design and validation of our solution.

3.5 References

[Ahmed, 2000] Ahmed, Kal. Topic Maps for repositories, XML Europe, 2000, http://www.gca.org/papers/xmleurope2000/papers/s29-04.html

[ALFANET-D12, 2002] D12 - State of the Art, ALFANET

[Bloom, 1957] Bloom, B.S. Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. New York ; Toronto: Longmans, Green.
http://www.coun.uvic.ca/learn/program/hndouts/bloom.html

[Bray, 2001] Bray, Tim. What is RDF? http://xml.com/pub/a/2001/01/24/rdf.html

[CYCORP, 2002] Cycorp Ontology: http://www.cyc.com

[Davenport and Prusak ,1998], "Working knowledge", Harvard Business School Press

[DPI, 1998] Wisconsin Model Academic Standards for Information & Technology Literacy, http://www.dpi.state.wi.us/dltcl/imt/itlstfst.html

[EKMF, 2001] IST Project No 2000-26393
D1.1 KM Framework (EXTRACT): KM Technologies and Tools
D2.1 “Standardised KM application models” – Part B Common Approaches in KM. KM Technologies.
EKMF Position Statement on Knowledge Management Standardizaton

[Garshol, 2002] http://xmlhack.com/read.php?item=1579

[Gómez-Pérez, 1999] "Ontological Engineering: a state of the art"

[Gruber, 93] A translation Approach to portable ontology specification, Knowledge Adquisition, 5(2):199-220, 1993. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[Guarino, 1998] Formal Ontology and Information Systems, Proceedings of FOIS’98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15. http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/FOIS98.pdf

[IBSTPI, 2002] http://www.ibstpi.org

[IEEE SUO 2001] The IEEE Standard Upper Ontology web site: http://suo.ieee.org

[Moore, 2000] Topic Map technology - the state of the art, Graham Moore, http://www.gca.org/papers/xmleurope2000/papers/s22-04.html

[NCPQ, 1995] NCPQ Standards and Indicators, http://ncrve.berkeley.edu/ST2.1/NCPQStandards.html

[Niles, 2001] Towards a Standard Upper Ontology, Ian Niles, Adam Pease, Teknowledge Corporation, http://projects.teknowledge.com/HPKB/Publications/FOIS.pdf

[Ontoknowledge, 2002] http://www.ontoknowledge.org/oil/

[Ontopia, 2002] http://www.ontopia.net/topicmaps/what.html; http://www.ontopia.net/topicmaps/materials/tao.html

[OWL, 2002] http://www.w3.org/TR/2002/WD-owl-features-20020729/

[Spector, 2001] Competencies for Online Teaching, J. Michael Spector and Ileana de la Teja, http://www.ericit.org/digests/EDO-IR-2001-09.shtml

4. Human Capital Profiles

4.1 General overview

HR-XML (http://www.hr-xml.org) is a global, independent, non-profit consortium dedicated to enabling e-commerce and inter-company exchange of human resources (HR) data. The work of the Consortium centres on the development and promotion of standardized XML vocabularies for HR. HR-XML's efforts are focused on standards for staffing and recruiting, compensation and benefits, training and workforce management. The Consortium has a membership of companies represented in 22 different countries [6].

The HR-XML Consortium is driven by the needs and priorities of its members. Any member can propose that the Consortium undertake a standards activity. Proposals must be submitted to the HR-XML Business Steering Committee (BSC) for review. Before proposals are submitted to the BSC, they must include the names of at least three sponsor organizations and satisfy certain other pre-requisites.

They have completed the following specifications [16]:

4.2 Components

BackgroundCheck-1_0, Version 1.0 (2002-05-01) (Approved) The BackgroundCheck schema supports background check requests to third-party suppliers of background checking services. The specification also supports the return of search results.

EntityIdentifiers, Version 1.0 (2002-05-01) (Approved) This specification outlines a methodology for identifier management that can be implemented across HR-XML Consortium schemas. The specification includes a set of design norms and recommendations as well as a XML Schema data type to use for entity identifiers. This specification addresses identifiers that are used as "keys". Keys enable data in one transaction to be associate with data in other separate transactions.

Resume, Version 2.0 (2002-05-01) (Approved) The Resume 2.0 specification provides a definition for an XML Resume. The Resume 2.0 specification includes modules for employment, education, and military history.

Staffing Industry Data Exchange Standards (SIDES), Version 1.0 (2002-05-01) (Approved) Staffing Industry Data Exchange Standards -- commonly known by the acronym "SIDES" -- is a comprehensive suite of data exchange standards designed to offer new efficiencies and cost savings for staffing customers, staffing suppliers, and other stakeholders in the staffing supply chain. Major modules include: StaffingOrder; HumanResource; Assignment; StaffingSupplier; StaffingCustomer; StaffingAction; Extended TimeCard; and Invoice (an extended version of OAGIS 8.0 Invoice).

Time Expense Reporting, Version 2.0 (2002-05-01) (Approved) TimeCard 2.0 is a simple definition of the elements required to report time worked and expenses incurred. Some of the changes in TimeCard 2.0 are: Batch support has been removed from schema (this will be handle at the envelope level); AdditionalData modified to allow extensions; Changed dates to use CPO's DateTimeDataTypes, etc.

UserArea, Version 1.0 (2002-05-01) (Approved) This specification sets out a standard mechanisms for extending HR-XML Schemas.

ContactMethod, Version 1.0 (2002-01-31) (Approved) Contact Method provides XML schema designers the patterns they need to capture postal addresses, phone numbers, e-mail, and on-line and wireless messaging.

JobAndPositionHeader, Version 1.0 (2002-01-31) (Approved) The Job and Position Header specifications are high-level entities that may be used within a variety of HRM models and business processes. Both entities may be categorized into fragments such as Duties and Responsibilities, Work Policy, Requirements, and Work Schedule.

PayrollBenefitContributions, Version 1.0 (2002-01-31) (Approved) The PayrollBenefitContributions specification allows participant contributions information to be sent to a third party administrator. This specification is designed to support U.S. payrolls.

TimeCardConfiguration, Version 1.0 (2002-01-31) (Approved) The TimeCardConfiguration specification is designed for use with HR-XML's Time Expense Reporting specification. TimeCardConfiguration allows the trading partners to describe the allowed values for the various elements of a time card.

WorkSiteAndEnvironment, Version 1.0 (2002-01-31) (Approved) The WorkSite and WorkSite Environment specifications contain information pertaining to the site or location of a job or position; the environment of a job or position; dress code; and safety equipment.

Competencies, Version 1.0 (2001-Oct-16) (Approved) The competencies schema allows the capture of information about evidence used to substantiate a competency and ratings and weights that can be used to rank, compare, and otherwise evaluate the sufficiency or desirability of a competency.

DateTime Data Types, Version 1.1 (2001-Oct-16) (Approved) This specification sets out an approach for HR-XML Schema designers to require or prohibit the Time Zone designation for date, time and dateTime values in a consistent manner.

Enrollment, Version 1.0 (2001-10-16) (Approved) This specification supports the transfer of benefits enrollment data among U.S.-based employers, third-party administrators, benefit suppliers/vendors, and other parties involved in the administration or provision of employee benefits.

PersonName, Version 1.2 (2001-Oct-16) (Approved) Prescribes the form of the Person Name object used in HR-XML specifications. This update provides a version in XML Schema as well as in DTD.

PostalAddress, Version 1.2 (2001-Oct-16) (Approved) Prescribes the form of the PostalAddress object used in HR-XML specifications. This update provides a version in XML Schema as well as in DTD.

Time Expense Reporting, Version 1.0 (2001-10-16) (Approved) This specification for an "XML timecard" supports the reporting of time worked as well as certain expenses that might be reported by contract or temporary staff.

Effective Dating, Version 1.0 (2001-07-17) (Approved) This specification sets out guidance for using effective dates throughout the Consortium's work.

PersonName, Version 1.1 (2001-07-17) (Approved) PersonName Version 1.1 is an update of an earlier specification. Minor changes have been made and certain Version 1.0 components have been deprecated. Version 1.1 is backwardly compatible with Version 1.0.

PostalAddress, Version 1.1 (2001-07-17) (Approved) PostAddress Version 1.1 is an update of an earlier specification. Minor changes have been made and certain Version 1.0 components have been deprecated. Version 1.1 is backwardly compatible with Version 1.0.

Staffing Exchange Protocol, Version 1.1 (2001-07-17) (Approved) Staffing Exchange Protocol Version 1.1 includes explicit support for procurement of temporary and contract staff. A wide variety of changes also have been made from the previous version to improve the flexibility and completeness of the SEP DTDs and to make them easier to deploy globally. Version 1.1 is designed to be backwardly compatible. Version 1.0 documents are valid against Version 1.1 DTDs. Certain Version 1.0 features are now deprecated.

Provisional Envelope Specification, Version 1.0 (2000-11-08) (Provisional) A simple envelope that can be used to implement HR-XML Consortium specifications.

Staffing Industry Data Exchange Standards

SIDES is based on the eXtensible Markup Language (XML). SIDES modules are defined using XML "schemas." The SIDES schemas define a common language for data exchanges and provide a rich and customisable set of options to fit a wide variety of needs.

SIDES provides value in many ways. It can provide process efficiencies and improved service by:

- Shortening order fulfilment cycle as a result of improved front-office integration with multiple suppliers

- Reducing errors and processing time by eliminating the need to re-key data from paper sources, “copy and paste” between systems, or otherwise manually integrate data between different systems
- Streamlining communication through the use of the SIDES ***StaffingAction*** module (This allows process-oriented communications to be accomplished on-line, at the customer’s convenience. Reliance on “phone-tag” for scheduling and other process-oriented communications is eliminated.)
- Improving accuracy and timeliness of invoicing as a result of SIDES ability to directly relate invoice data to customer-confirmed assignment information and customer-approved timecard data

SIDES was developed in a modular fashion so that it could be implemented by the wide variety of stakeholders within the staffing supply chain. The complete set of SIDES modules can be used to support end-to-end staffing processes or SIDES modules can be use to fit component-by-component implementations.

Widespread implementation of SIDES allows organizations to interface with multiple trading partners throughout the staffing supply chain for the cost of a single interface. By implementing SIDES, organizations can avoid having to engineer many separate exchange mechanisms.

SIDES provides the trading partners with a common data dictionary, so each is able to map their data to a predetermined common specification. This greatly reduces the costs for investigating and proposing dictionary and mapping strategies for each.

SIDES provides for each of the common business process in the human capital supply chain, including requisitions, orders, rates, assignments, time and expense capture, and invoicing. Standard data definitions for information flows throughout the staffing lifecycle substantially streamline staffing processes.

SIDES is centrally maintained by the HR-XML Consortium. SIDES 1.0 was approved by HR-XML Consortium members 2002 April 29. While SIDES took many thousands of working hours to create, it is freely available. There are no licensing charges [17].

XML Resume Specification

The HR-XML Consortium, the global non-profit organization dedicated to creating data interchange standards for human resources, has approved a new XML resume specification. HR-XML’s Resume 2.0 specification will enable a range of innovative new applications and services that will benefit job seekers as well as employers.

“Today, employers are drowning in sea of unstructured resumes, which are difficult to match against business requirements,” according to Chuck Allen, Director, HR-XML Consortium, Inc. “HR-XML’s versatile, but structured XML Resume format, has great potential to change the search for qualified employees from a ‘shot-gun approach’ to a targeted exercise. Likewise, structured resumes may give job seekers an opportunity to more effectively communicate their unique abilities to potential employers,” according to Allen.

The Resume 2.0 specification represents a substantial improvement over the resume definition included in HR-XML’s Staffing Exchange Protocol 1.1. SEP 1.1 was defined using Document Type Definitions (DTDs). Resume 2.0 is defined using the World Wide Web Consortium’s more powerful and flexible XML Schema Definition Language (XSD). This makes the Resume 2.0 specification more modular and extensible than the prior DTD-based version.

Resume 2.0 has already been incorporated within HR-XML’s Staffing Industry Data Exchange Standards. The Employment History, Education History, and Military History modules within Resume 2.0 also are shared with HR-XML’s new Background Checking specification. In addition, the Resume specification will be part of a future XSD-based version of HR-XML’s Staffing Exchange Protocol.

“HR-XML’s Resume 2.0 has tremendous applicability to the hiring management cycle – particularly in recruiting, intake, screening, and candidate selection processes,” said Kathi Dolan, Research Analyst, Manpower Inc. “Having a standardized way to communicate an individual's history and qualifications both internally and externally provides concrete benefits for everyone involved," according to Dolan.

“Resume 2.0 is an integral part of HR-XML’s forthcoming Version 2.0 recruiting and staffing specifications,” said Nicholas Scobbo, HRIS Analyst, Mitre Corporation, and chair of HR-XML’s Recruiting and Staffing Workgroup. “HR-XML’s Resume specification should go a long way in helping companies deal with the multiple sources and formats of resumes and in increasing the efficiency of recruiting systems and processes," according to Scobbo.

Membership within the HR-XML Consortium is open to HR professionals, vendors, consultants, and other users or providers of HR systems and services.

4.3 Applicability for ALFANET

For the moment it is unclear whether or not standards in the area of human capital profiles could be useful in the context of the ALFANET project.

4.4 References

HR-XML: http://www.hr-xml.org

5. Multi-agent Standards

5.1 General overview

As seen from Distributing Artificial Intelligent, a multi-agent system is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity. More recently, the term multi-agent system has been given a more general meaning, and it is now used for all types of systems composed of multiple autonomous components showing the following characteristics [Flores-Méndez, 1999]:

- each agent has incomplete capabilities to solve a problem
- there is no global system control
- data is decentralized
- computation is asynchronous

Therefore, a multi-agent system should have the following skills [Camacho, 2002]:

- Social Organization
- Coordination
- Cooperation
- Negotiation
- Communication

Agent Interaction [Flores-Méndez, 1999]

Interaction is one of the most important features of an agent [Nwana, 1996]. In other words, agents recurrently interact to share information and to perform tasks to achieve their goals. Researchers investigating agent communication languages mention three key elements to achieve multi-agent interaction [Finin, Labrou and Mayfield, 1997][Huhns and Singh, 1998][Peng, Finin, Labrou, Chu, Long, Tolone and Boughannam, 1998]:

- A common agent communication language and protocol
- A common format for the content of communication
- A shared ontology

5.2 Multi-agent architecture standards

5.2.1 Languages of communication between agents [Flores-Méndez, 1999]

There are two main approaches to designing an agent communication language [Genesereth, 1998]. The first approach is **procedural**, where communication is based on executable content. This could be accomplished using programming languages such as Java or Tcl. The second approach is **declarative**, where communication is based on declarative statements, such as definitions, assumptions, and the like.

Because of the limitations on procedural approaches (e.g., executable content is difficult to control, coordinate, and merge), declarative languages have been preferred for the design of agent communication languages. Most declarative language implementations are based on illocutionary acts, such as requesting or commanding; such actions are commonly called **performatives**. One of the more popular declarative agent languages is KQML.

KQML, which is an acronym for **Knowledge Query and Manipulation Language** [Genesereth and Fikes, 1992], was conceived both as a message format and a message handling protocol to support run-time knowledge sharing among agents [Finin, Labrou and Mayfield, 1997]. This language can be thought of as consisting of three layers: a communication layer (which describes low level communication parameters, such as sender, recipient, and communication identifiers); a message layer (which contains a performative and indicates the protocol of interpretation); and a content layer (which contains information pertaining to the performative submitted).

5.2.2 Ontologies [Flores-Méndez, 1999]

Ontologies are defined as specification schemes for describing concepts and their relationships in a domain of discourse [Finin, Labrou and Mayfield, 1997]. It is important that agents not only have ontologies to conceptualise a domain, but also that they have ontologies with similar constructions. Such ontologies, when they exist, are called **common ontologies**.

Ontolingua [Gruber, 1993] is often mentioned in the literature as a system that provides a vocabulary for the definition of reusable, portable and shareable ontologies. Ontolingua definitions are described using syntax and semantics similar to those of the **Knowledge Interchange Format** [Ginsber, 1991], also known as KIF, which is a format to standardize knowledge representation schemes based on first-order logic.

Knowledge Interchange Format (KIF) is a computer-oriented language for the interchange of knowledge among disparate programs developed by the ARPA-sponsored Knowledge Sharing Effort. It has declarative semantics (i.e. the meaning of expressions in the representation can be understood without appeal to an interpreter for manipulating those expressions); it is logically comprehensive (i.e. it provides for the expression of arbitrary sentences in the 1rst-order predicate calculus); it provides for the representation of knowledge about the representation of knowledge; it provides for the representation of non-monotonic reasoning rules; and it provides for the definition of objects, functions, and relations. [from UMBC Agent Web] Semantically, there are four categories of constants in KIF ([Genesereth, 1991], [Genesereth & Fikes, 1992]): object constants, function constants, relation constants, and logical constants. Object constants are used to denote individual objects. Function constants denote functions on those objects. Relation constants denote relations. Logical constants express conditions about the world and are either true or false. KIF is unusual among logical languages in that there is no syntactic distinction among these four types of constants; any constant can be used where any other constant can be used. This feature allows the reification of formulas as terms used in other formulas, making it possible to make statements over statements. This introduces second-order features in KIF, which provides an important extension of first-order logic.

Message-Oriented Middleware (MOM) denotes a type of software systems for managing transactional message queues as the basis of asynchronous message passing. Well-known products include IBM MQSeries and Sun JMQ. A standard MOM application programming interface for Java, called Java Messaging Service (JMS) has been proposed by Sun [Wagnerr, 2000].

5.2.3 MAS Architectures Standardization [Flores-Méndez, 1999]

Exist several independent industrial and research groups started to pursue the standardization of multi-agent technology. Prominent efforts, such as those of the Object Manager Group (OMG), the Foundation for Physical Agents (FIPA), the Knowledge-able Agent-oriented System (KAoS) group, and the General Magic group are briefly described below.

The **OMG** group proposes a reference model as a guideline for the development of agent technologies [Virdhagriswaran, Osisek and O'Connor, 1995]. This model outlines the characteristics of an agent environment composed of agents (i.e., components) and agencies (i.e., places) as entities that collaborate using general patterns and policies of interaction. Under this model, agents are characterized by their capabilities (e.g., inferencing, planning, and so on), type of interactions (e.g., synchronous, asynchronous), and mobility (e.g., static, movable with or without state). Agencies, on the other hand, support concurrent agent execution, security and agent mobility, among others.

The Foundation for Intelligent Physical Agents (**FIPA**) is a multi-disciplinary group pursuing the standardization of agent technology. FIPA's approach to MAS development is based on a "minimal framework for the management of agents in an open environment." This framework is described using a reference model (which specifies the normative environment within which agents exist and operate), and an agent platform (which specifies an infrastructure for the deployment and interaction of agents).

KAoS is described as "an open distributed architecture for software agents." The KAoS architecture describes agent implementations (starting from the notion of a simple generic agent, to role-oriented agents such as mediators and matchmakers), and elaborates on the interactive dynamics of agent-to-agent messaging communication by using conversation policies.

General Magic is a commercial endeavour researching mobile agent technology for electronic commerce. Conceptually, this technology models a MAS as an electronic marketplace that lets providers and consumers of goods and services find one another and transact business. This marketplace is modelled as a network of computers supporting a collection of places that offer services to mobile agents. Mobile agents, which are entities that reside in one particular place at a time, have the following capabilities [White, 1997]:

- they can **travel**, to move from one place to another
- they can **meet** other agents, which allows them to call one another agent's procedures
- they can create **connections**, to allow an agent to communicate with another agent in a different place
- they have **authority**, which indicates the real-world individual or organization that the agent represents
- they have **permits** to indicate the capabilities of agents

5.2.4 Methodologies

In the last years have been arising methodologies for the construction of the systems multi-agents, next four methodologies comment:

5.2.4.1 MAS-CommonKADS [Iglesias, 1998]

The methodology consists of the development of seven models: Agent Model, that describes the characteristics of each agent; Task Model, that describes the tasks that the agents carry out; Expertise Model, that describes the knowledge needed by the agents to achieve their goals; Organisation Model, that describes the structural relationships between agents (software agents and/or human agents); Coordination Model, that describes the dynamic relationships between software agents; Communication Model, that describes the dynamic relationships between human agents and their respective personal assistant software agents; and Design Model, that refines the previous models and determines the most suitable agent architecture for each agent, and the requirements of the agent net-work.

The application of the methodology consists of the development of the different models. Each model consists of constituents (the entities to be modelled) and relationships between the constituents. A textual template is defined for each constituent in order to describe it. The states of the constituents describe their development: empty, identified, described or validated.

The software process model of the methodology combines the risk-driven approach with the component-based approach. The general process is risk driven, that is, in every cycle the states of the models to be reached are defined for reducing the perceived risks. When a state consists of identifying components, the developed components (agents, services, knowledge bases, etc.) are candidates for reusing.

5.2.4.2 Gaia [Wooldridge, Jennings, and Kinny, 2000]

The Gaia methodology is both general, in that it is applicable to a wide range of multi-agent systems, and comprehensive, in that it deals with both the macro-level (societal) and the micro level (agent) aspects of systems. Gaia is founded on the view of a multi-agent system as a computational organisation consisting of various interacting roles. The key concepts in Gaia are roles, which have associated with them responsibilities, permissions, activities, and protocols. Roles can interact with one another in certain institutionalised ways, which are defined in the protocols of the respective roles.

5.2.4.3 Belief-Desire-Intention (BDI) [Kinny, Georgeff and Rao, 1996]

An early attempt to define a multiagent systems methodology was developed by Kinney, Georgeff, and Rao. They proposed a set of specialized Object-Oriented models for developing a system of Belief-Desire-Intention (BDI) agents. In this methodology, there are two sets of models: external and internal.

From the external viewpoint, the system is decomposed into agents, their responsibilities, the services they perform, the information they require, and their external interactions. These characteristics are captured in two models: the Agent Model and the Interactions Model. The Agent Model describes the hierarchical relationship between different abstract and concrete agent classes, and identifies the agent instances that may exist within the system, their multiplicity, and when they come into existence. The Interaction Model describes the responsibilities of an agent class, the services it provides, associated interactions, and control relationships between agent classes.

From the internal viewpoint, the elements required by particular agent architectures are modeled for each agent using three models that describe its informational and motivational state and its potential behavior: the Belief Model, the Goal Model, and the Plan Model. The Belief Model describes the information about the environment and internal state that an agent of that class may hold, and the action is may perform. The Goal Model describes the goals that an agent may possibly adopt, and the events to which it can respond. Finally, the Plan Model describes the plans that an agent may possibly employ to achieve its goals or respond to events it perceives. It consists of a plan set which describes the properties and control structure of individual plans.

5.2.4.4 MaSE [Deloach, Wood and Sparkman, 2001]

MaSE is a methodology for the analysis of multiagent systems and provides solid foundation for the design and development of multiagent systems. MaSE not only takes advantage of goal-driven development, but also uses the power of multiagent systems by defining roles, protocols and tasks in the analysis phase. Another less obvious advantage that MaSE has is that its steps are defined at a fine level of granularity making the transition between models simpler and more straightforward than many of the techniques discussed above. MaSE also provides more guidance on how models relate to each other.

5.3 Applicability for ALFANET

FIPA standards are the currently accepted standards for heterogeneous and interacting agents and agent-based systems.

The project aims to work with a multiagent approach to afford some modules functionality. Multiagents, by nature, have requirements of openness, communication capacity with other external agents, an extension to implement new agents. Therefore it is recommended to comply to those FIPA standards relevant for the communication between agents.

5.4 References

Camacho Fernández, D. Recuperación e Integración de Información disponible en Web para la Resolución de Problemas, Universidad Carlos III de Madrid, 2002.

Deloach, S.A., Wood, M.F. and Sparkman, C.H. Multiagent Systems Engineering**.** International Journal of Software Engineering and Knowledge Engineering Vol. 11, No. 3 (2001) 231-258

Finin, T., Labrou, Y. and Mayfield, J. KQML as an Agent Communication Language. In: *Software Agents*, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pages 291-316.

Flores-Mendez R.A. Standardization of Multi-Agent System Frameworks, University of Calgary 1999.

Genesereth, M. An Agent-based Framework for Interoperability. In: *Software Agents*, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pages 317-345.

Genesereth, M. and Fikes, R. *Knowledge Interchange Format, Version 3.0 Reference Manual*, Technical Report, Computer Science Department, Stanford University, USA., 1992.

Ginsberg, M. The Knowledge Interchange Format: The KIF of Death. In: *AAAI Magazine*, Fall 1991, Volume 12, Issue 3, pages 57-63.

Gruber, T.R. A Translation Approach to Portable Ontology Specifications. In: *Proceedings of the Knowledge Acquisition for Knowledge-Based Systems (KAW'93)*, Gaines, B.R. and Musen, M. (Eds.), Banff, Canada, 1993, pages 199-220.

Huhns, M.N. and Singh, M.P. Agents and Multi-agent Systems: Themes, Approaches, and Challenges. In: *Readings in Agents*, Huhns, M.N. and Singh, M.P. (Eds.), San Francisco, Calif., Morgan Kaufmann Publishers, 1998, pages 1-23.

Iglesias, C.A., Definición de una Metodología para el desarrollo de Sistemas Multiagente, Tesis Doctoral Enero 1998

Kinny, D. Georgeff, M. and Rao, A. "A Methodology and Modelling Technique for Systems of BDI Agents," in Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW '96. Lecture Notes in Artificial Intelligence, vol. 1038. Springer-Verlag, Berlin Heidelberg, 1996.

Nwana, H. S. and Ndumu, D. T. A Perspective on Software Agents Research, Nwana, H. S. and Ndumu, D. T. IN: The Knowledge Engineering Review, 14(2), pages 125-142, Cambridge University Press 2000.

Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W.J. and Boughannam, A. A Multi-Agent System for Enterprise Integration. In: *Proceedings of the Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology*, H.S. Nwana and D.T. Ndumu (Eds.), London, UK, March, 1998, pages 155-169.

Virdhagriswaran, S., Osisek, D. and O'Connor, P. Standardizing Agent Technology. In: *ACM StandardView*, 1995, Volume 3, Number 3, pages 96-101.

Wagner, G. The Aois Glossary On Agent-Orientation in Information Systems, 2000

White, J.E. Mobile Agents. In: *Software Agents*, J.M. Bradshaw (Ed.), Menlo Park, Calif., AAAI Press, 1997, pages 437-472.

Wooldridge, M., Jennings, N. and Kinny, D. "The Gaia Methodology for Agent-Oriented Analysis and Design," Journal of Autonomous Agents and Multi-Agent Systems. vol. 3(3), 2000.

6. Other technical standards

6.1 P3P Platform for Privacy Preferences

It is important to establish both formal and informal trust relationships with users when building a personalised, authentication-mediated system. A formal 'privacy statement' for the service is a necessity. Users should know exactly what data will be holding about them, and the purposes to which it will be put.

The Platform for Privacy Preferences Project [P3P, 2002] 1.0, developed by W3C, provides a standard, simple, automated way for users to gain more control over the use of personal information on Web sites they visit. It is an XML-based language for expressing Web site privacy policies.

At its most basic level, P3P is a standardized set of multiple-choice questions, covering all the major aspects of a Web site's privacy policies. Taken together, the answers present a machine readable version of the site's privacy policy, a clear snapshot of how a site handles personal information about its users. P3P-enabled Web sites make this information available in a standard, machine-readable format.

P3P enabled browsers can "read" this snapshot automatically and compare it to the consumer's own set of privacy preferences. P3P enhances user control by putting privacy policies where users can find them, in a form users can understand, and, most importantly, enables users to act on what they see.

6.2 Learner Profiles

The **Universal Learning Format** is a framework for enabling the cross-industry exchange of learning content such as education catalogues, course content, competency libraries, certification tracks, and learner profiles. It includes a set of XML-based formats for creating robust, reusable XML-based documents. Using this framework, learning providers can seamlessly exchange a variety of learning content as well as make their learning content universally available for search and discovery.

Profile Format is an XML-based representation for describing learner profile information. Learner profiles comprise a variety of data about learners, including personal and job information, learning history, goals and plans, and held competencies and certifications.

vCard is a simple standard which specifies a common set of fields for personal profile data; in this sense it plays a similar role to that played by the Dublin Core element set in document description.

Other markup languages (not related with learner profiles) are:

- o Learning Material Markup Language (LMML)
- o Tutorial Markup Language (TML)

6.3 Relation with other standards

The Universal Learning Format is based on the work done by standards bodies such as Instructional Management System (IMS), Advanced Distributed Learning (ADL), and the Institute of Electrical and Electronic Engineers (IEEE). It embraces and extends the existing online learning standards advanced by these organizations and is designed to take advantage of new standards as they emerge.

Profile Format captures this information in an XML-based format using RDF to define metadata for describing learners. Profile Format incorporates several existing metadata standards, including the Dublin Core and vCard, which ensures compatibility with existing person/profile descriptions.

6.4 Applicability for ALFANET

P3P can be used to express the privacy policies of the ALFANET website.

Learner Profiles can be taken into account in the design the User Model Learner Profile, as a check-list of information to consider, more than a format to represent the data.

6.5 References

IEEE Personal and Private Information (PAPI) draft standard, see http://ltsc.ieee.org/wg2/index.html.

IMS Learner Information Packaging (LIP) specification, see http://www.imsproject.org/profiles/libinfo01.html.

Saba Profile Format, see http://www.saba.com/standards/ulf/Overview/profile.htm.

vCard standard, see http://www.imc.org/rfc2426.

[P3P, 2002], http://www.w3.org/P3P/

Part II The role and relation of ALFANET towards standards

Overview of the ALFANET selection

In this part the ALFANET selection of the standards and specifications identified in part 1 will be shown and further explained in the context of the project. In chapter 1 a selection will be presented of the selected learning technology specifications illustrated by hypothetical scenarios. Chapter 2 presents the standards related to agent technology. Chapter 3 summarizes the main outcome of this study and points for further attention.

1. Selected learning technology recommendations for ALFANET

1.1 Overview of the ALFANET selection

Learning technology specifications

Two important practical criteria for including specifications should be that:

- there already exists usable software and/or tools based on the specification;
- a major part of the specification is relevant for the problems addressed in the project and it is not needed to extend or change the specification in a major way before it can be used.

When neither is the case using a specification will give more problems then pleasure. For the learning technology specifications it seems that the following should be included:

1. IMS-LIP – This specification is used as a solid start for building the student model. The student model or portfolio could be used within ALFANET to exchange information between the various components that ALFANET is made of. Within a distributed system design the various system components need to exchange information. Within the various components different ways may be used format user data. To the outer world however it is highly recommended that one 'language is spoken'. For student information IMS-LIP is a usable specification.
2. IMS-LD – addresses relevant problems for the project and a good player exists. The IMS Learning Design (LD) specification offers functionalities that no comparable system can offer, such as *re-usability*, *multiple roles* in collaboration and *personalised learning paths*. IMS-LD has the flexibility to support individual paths through the learning material. This functionality of *personalization* means that learners are not constrained
3. IEEE-LOM – the meta-data are expected to be key when implementing the adaptive functionality of ALFANET. The objective of the meta-data specifications, i.e. to create a uniform way for describing learning resources so that they can be more easily found (discovered) and subsequent used appropriately is clear. IEEE-LOM could be used to describe and find units of learning, Learning Objects, Activities, etc. (which are formatted in IMS-LD).

Other IMS specifications (Question&Test, Enterprise, Simple Sequencing, …) all cover relevant ground but, at this moment, do not cover key functionality required by ALFANET.

1.2 ALFANET use of learning technology standards

To come to a better understanding of the recommended standards some scenarios are worked out. These scenarios are merely to illustrate how information is exchanged between the various components of the ALFANET system. This means that the suggested adaptation here does not reflect the final adaptation worked out in the ALFANET project. The scenarios presented show the use of learning technology specifications in relation to ALFANET in order to:

I. Logon to a course.
II. Dynamically adapt course content.
III. Dynamically adapt presented links within a unit of learning.
IV. Stimulate collaboration between students.
V. Inform developers on design pitfalls.

1.2.1 Scenario I: Student logon

Description:

A student is subscribed to a course and is logging on to it via the application called ALFANET.

Scenario steps:

1. The student wants to enter into the ALFANET application and is providing his username and password.
2. The application accepts the user data and requests the courses that this student is subscribed to, from the students portfolio (Repository).
3. The courses that this student is subscribed to are returned.
4. The courses that this student is subscribed to are made visible trough ALFANET.
5. The student can select a course from the list presented.
6. ALFANET starts the course in Edubox
7. Edubox connects to the unit of learning and investigates the students' portfolio to find out students' progress for this course.

Step	Data used	Usable standard
1	Personal data	LIP
	Username/password	LIP
2	Registered courses	LIP
	Course last visited	LIP
5	Course username/password	LIP
7	Student progress	LIP
	Reference to unit of learning	LOM

1.2.2 Scenario II: Content adaptation

Description:

A student is subscribed to a course. When accessing the course only the outline of the course is visible to the student. From the outline the student can only enter the first module. This module has to be completed by every student prior to the other modules of the course. Based on the choices made by the student in the first module the profile of the student has been created and has an initial set of the students' preferences and has identified the learning gaps. Based on this profile the application sets out a learning path that best matches the student's preferences and learning gaps. This means that some of the activities are left out others are presented at a more easy of difficult level. During the prolongation of the course the application keeps on monitoring the students' progress and choices to keep the students' profile as accurate as possible, this results in dynamic learning path though the course.

Scenario steps:

1. It is assumed that the student is already logged on to a specific course.
2. From there on the student is performing an activity within a unit of learning.
3. While the student is working on the activities the student data entered (i.e. entering answers to open questions, making choices from multiple answers or waiving an activity) and the data registered by Edubox are passed on to the students portfolio.
4. The application continuously monitoring the student portfolio and as it finds new data it analyses it. Based on the collected data the application concludes that this branch of the course isn't appropriate for this student.
5. ALFANET search in the content repository the activity that is more appropriate.
6. An activity is found in the content repository that matches the topic of the unit of learning but with the desired characteristics.
7. The analysis result is send to the student with an advice to modify the course difficulty level (a positive reaction from the student results in the modification of the course outline).
8. The student has to indicate whether or not he/she wants the course to be modified.
9. ALFANET modifies the course outline and sends the result to the corresponding modules.
10. ALFANET updates the student portfolio with the modified course outline, personal profile and intervenient data.
11. The student previously ended an activity and is about to select the next activity. The activity list the student sees is the one updated by the application.
12. When an activity is selected by the student, ALFANET requests this activity from the content repository.
13. The activity is received from the content repository and presented to the student.
14. The student can continue with the activities in the unit of learning.

Step	Data used	Usable standard
3	Student activity data	LD
4	Learning objectives	LD
	Competency	LIP
5	Search Metadata	LOM
6	Object Metadata	LOM
9	Reference to activity	LOM
10	Student preference	LD

1.2.3 Scenario III: Link adaptation

Description:

While working on an assignment a student has to choose a subject of interest. Based on this subject the application searches the content repository to retrieve URLs that refer to web pages that are of use for the student. The found URLs are dynamically put on the course page the student is working on.

Scenario steps:

1. It is assumed that the student is already logged on to a specific course.

2. From there on the student is performing an activity within a unit of learning. The activity involves the selection of a topic for further investigation. The topic can be selected from a list that is presented to the student.
3. The topic chosen by the student is put in the students' portfolio
4. ALFANET continuously monitoring the students' portfolio and detects that a topic is selected for the assignment.
5. ALFANETstarts searching the content repository according to the selected topic.
6. ALFANET receives the search results and analyses them.
7. The relevant links found for the selected topic are presented within the current unit of learning from where the topic was selected.
8. The result is presented to the student in the format of URLs.
9. Depending on the destination of the URL this can happen:
    a. The destination of the URL falls within the outline of the course.
    b. The destination of the URL falls outside the course. This could be for example a web page or an e-book or another course.
10. The URLs followed by the student are registered in the students' portfolio

Step	Data used	Usable standard
3	Students' choice	LD
5	Search parameters	LOM
6	Search results (URLs or object IDs)	LOM
10	Students'choice	LD

1.2.4 Scenario IV: Collaboration adaptation

Description:

While a student is working though a unit of learning ALFANET detects that another (group) student is working through the same unit of learning. One of the students is experiencing trouble with the unit of learning. ALFANET detects this by comparing study patterns of this student with that of previously successful students. ALFANET creates a discussion group for this unit of learning. The students are notified that there is a newsgroup for the subject at hand.

Scenario steps:

1. It is assumed that the students x and y are already logged on to a specific course. They do not necessarily need to be online at the same time and they do not need to be at the same point in the course.
2. From there on the students are performing activities within a unit of learning.
3. The progress the students make is registered in the portfolios of the students.
4. By analysing various parameters and usage data it is detected that student y is having trouble with this unit of learning. Also, for student x is going through the same unit of learning with less difficulty.
5. To support the interaction between the students and to prevent the overload of a human tutor the application searches an existing newsgroup for this unit of learning.
6. Depending on the search result the application creates a new newsgroup for the unit of learning or uses an existing newsgroup.
7. The students are made aware (notified) that there is a newsgroup for this unit of learning.
8. Depending on the action undertaken by the students the students' portfolio is updated.

Step	Data used	Usable standard
3	Student results	LD
	Student performance data	LD
	Questionnaire results	LD
4	Activity results	LD
	Student progress data	LD
5	Search query with unit of learning, cohort information, preferences.	LIP
6	URL to newsgroup	LIP
7	Notification message to student	LD
8	Students' response	LIP

1.2.5 Scenario V : Course design adaptation

Description:

ALFANET continuously monitor the activities performed by students taking courses. Although ALFANET monitor the behaviour of the individual students also some interferences can be made of all students that have studied a particular unit of learning. By analysing the behaviour across students it could become possible to find pitfalls in activities or to locate errors. The detected errors are then coupled back to the authors of the units of learning so that they can modify the design of the single unit or a sequence of units of learning.

Scenario steps:

1. A developer creates or modifies a unit of learning and puts it in the content repository so that it becomes generally available.
2. It is assumed that students are logged on to the course containing a particularly unit of learning. This unit of learning is shown to the student.
3. Students have to perform activities within this unit of learning.
4. Data of the performed activities is stored in the students' portfolio. This data is accompanied by additionally collected data, for example data on student time performance, the amount of iterations, logs containing visited pages etc.
5. ALFANET is continuously monitoring all student portfolios to track down where a significant amount of students show different behaviour than expected by the course design.
6. If the application found unexpected student behaviour of a significant amount of students at a certain point in a unit of learning it creates a report that is send to the registered developer of that particular unit of learning. It is expected that the developer analysis the unit of learning according to the report in order to improve the unit of learning.

Step	Data used	Usable standard
1	Metadata for re-use	LOM
4	Monitoring data	LIP
	Application log data	LIP
5	Aggregated data for unit of learning	LD
6	Evaluation results for unit of learning	LD

2. Selected agent technology standards for ALFANET

2.1 Introduction

Although agents have a lot of advantages and are very suitable for solving certain kinds of problems when there is no global system control, data is decentralized, computation is asynchronous and the capabilities to solve the problem are distributed among different modules, they have also big drawbacks, regarding mainly to system performance. Therefore, it has to be very well analysed in advance which part of the system should be implemented using agent technology and which not. In any case, agent technology will be used in ALFANET, and it would be very sensible to benefit from existing standards in this field.

Agents technology can provide dynamism to the system functionality since agents are autonomous elements that communicate with one another while solving each particular problem. Thus, a strong communication language is needed to allow the agents to communicate each other.

The Foundation for Intelligent Physical Agents (**FIPA**) has already made a great effort standardizing the agent technology field and, for this reason, it seams quite reasonable to base the development to be done in ALFANET regarding agents technology in FIPA standards.

Moreover, the use of FIPA standards allow us to make use of existing tools that provide a framework to work with agents technology without having to make all the development from scratch. It allows us also to pay no attention to the evolution of these standards, since it is the responsibility of the tool used to update itself to the new standard specification or to integrate the new standards that appear in the future.

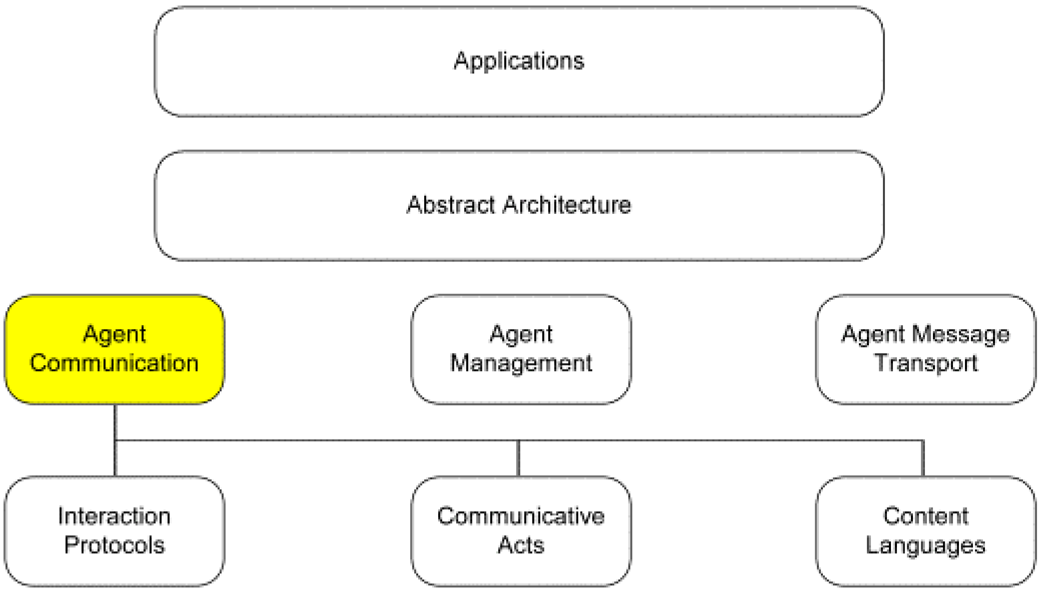
Agents technology is merely another technology that can facilitate the development of certain parts of the ALFANET system which '*accidentally*' need an internal communication that is pre-specified by standards.

2.2 FIPA Multi-Agent System Standard

The Foundation for Intelligent Physical Agents (FIPA) was formed in 1996 as a non-profit organisation with the remit of producing software standards for heterogeneous and interacting agents and agent-based systems across multiple vendors' platforms. This is expressed more formally in FIPA's official mission statement:

> *The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.*

FIPA specifications are divided into five categories: Applications, Abstract Architecture, Agent Management, Agent Message Transport and Agent Communication. The Agent Communication is also divided into Interaction Protocols, Communicative Acts and Content Languages. The following figure summarizes the relationships among them:



Applications:

FIPA Application specifications are example application areas in which FIPA agents can be deployed. They represent ontology and service descriptions specifications for a particular domain. One of these specifications is the FIPA Agent Software Integration Specification, which is a description of agents and an ontology for supporting agent integration with software systems. This specification is still in a experimental state, and it can be seen in http://www.fipa.org/specs/fipa00079/XC00079B.pdf.

Abstract Architecture:

The purpose of the FIPA Abstract Architecture is to foster interoperability and reusability, which leads to the identification of architectural abstractions linked by their relationships. It makes a distinction between those elements which can easily be defined in an abstract manner, such as agent message transport, FIPA ACL, directory services and content languages, and between those elements that cannot, such as agent management and agent mobility. The FIPA Abstract Architecture specifications deal with the abstract entities that are required to build agent services and an agent environment and is defined in http://www.fipa.org/specs/fipa00001/SC00001L.pdf.

Agent Message Transport:

The FIPA Agent Message Transport Specifications deal with the delivery and representation of messages across different network transport protocols, and it is defined in http://www.fipa.org/specs/fipa00067/SC00067F.pdf.

Agent Management:

The FIPA Agent Management Specification provides the framework within which FIPA agents exist and operate. It establishes the logical reference model for the creation, registration, location, communication, migration and retirement of agents. It deals with the control and management of agents within and across agent platforms and it is defined in http://www.fipa.org/specs/fipa00023/SC00023J.pdf.

Agent Communication:

The FIPA Agent Communication Specifications deal with Agent Communication Language (ACL) messages, message exchange interaction protocols, speech act theory-based communicative acts and content language representations. Developers of multi-agent systems require specialised communication techniques in order to structure the interactions in their agent systems. Ad hoc techniques are usually not sufficiently well designed or documented to be consistently extensible and implementable by others, or generally applicable to a wide set of agent problems. The FIPA specifications for agent communication address these issues. The core of these specifications was largely completed in FIPA 97, but this specification set has required continual maintenance and development since then. The description of the structure of FIPA ACL is defined in the FIPA ACL Message Structure Specification in http://www.fipa.org/specs/fipa00061/SC00061G.pdf.

FIPA Interaction Protocols (IPs) specifications deal with pre-agreed message exchange protocols for ACL messages and requirements for new interaction protocols.

FIPA Communicative Act (CAs) specifications deal with different utterances for ACL messages. The library of FIPA communicative acts and requirements for new communicative acts is defined in http://www.fipa.org/specs/fipa00037/SC00037J.pdf.

FIPA Content Language (CL) Specifications deal with different representations of the content of ACL messages. The general description of the requirements for a FIPA content language is still experimental and is defined in http://www.fipa.org/specs/fipa00007/XC00007B.pdf. There exist four representations already defined, FIPA-SL, FIPA-CCL, FIPA-KIF and FIPA-RDF, but except for the first one (FIPA-SL) which is already a standard, the other three are still experimental:

- FIPA SL Content Language Specification is a general purpose representation formalism that may be suitable for use in a number of different agent domains and is defined in http://www.fipa.org/specs/fipa00008/SC00008I.pdf.

- FIPA CCL Content Language Specification is intended to enable agent communication for applications that involve exchanges about multiple interrelated choices and is based on the representation of choice problems as Constraint Satisfaction Problems (CSPs). It is defined in http://www.fipa.org/specs/fipa00009/XC00009B.pdf.
- FIPA KIF Content Language Specification goal is to serve as a language for use in the interchange of knowledge among disparate computer systems (created by different programmers, at different times, in different languages...) and is defined in http://www.fipa.org/specs/fipa00010/XC00010C.html.
- FIPA RDF Content Language Specification deals with how objects, propositions and functions can be expressed in RDF and is based on an entity-relationship model; proposes XML as encoding syntax. It is defined in http://www.fipa.org/specs/fipa00011/XC00011B.pdf.

2.3 FIPA Standard Compliant Platforms

FIPA platforms, that is, platforms that are compliant with FIPA standards have already been implemented by various companies. A sample of the most relevant ones is the following:

Platform Name	Organisation
Comtec Agent Platform	Comtec (Japan)
Java Agent Development Framework (JADE)	CSELT (Italy)
April Agent Platform (AAP)	Fujitsu Laboratories of America (USA)
FIPA-OS	Nortel Networks (UK)
ZEUS Agent Building Toolkit	BT Laboratories (UK)

From this list, both ZEUS and JADE are the ones who have achieved a higher level of functionality, and are briefly described next:

ZEUS

The ZEUS toolkit provides a library of software components and tools that facilitate the rapid design, development and deployment of agent systems. The three main functional components of the ZEUS toolkit these are:

- The agent component library is a collection of software components that implement the functionality necessary for multi-agent systems.
- The ZEUS toolkit provides an integrated suite of editors that guide developers through the stages of our comprehensive agent development methodology. During this process developers describe the agents within their application, how they interact, and the tasks they perform.
- The Visualisation Tools collect information on agent activity, interpret it and display various aspects in real-time. This is our solution to the inherently difficult problem of analysing and debugging a multi-agent system where all the data, control and active processes are distributed.

JADE

The goal of JADE is to simplify the development of multi-agent systems while ensuring standard compliance through a comprehensive set of system services and agents in compliance with the FIPA specifications: naming service and yellow-page service, message transport and parsing service, and a library of FIPA interaction protocols ready to be used.

The JADE Agent Platform complies with FIPA specifications and includes all those mandatory components that manage the platform, that is the ACC, the AMS, and the DF. All agent communication is performed through message passing, where FIPA ACL is the language to represent messages.

The agent platform can be distributed on several hosts. Only one Java application, and therefore only one Java Virtual Machine (JVM), is executed on each host.

The communication architecture offers flexible and efficient messaging, where JADE creates and manages a queue of incoming ACL messages, private to each agent; agents can access their queue via a combination of several modes: blocking, polling, timeout and pattern matching based.

The full FIPA communication model has been implemented and its components have been clearly distincted and fully integrated: interaction protocols, envelope, ACL, content languages, encoding schemes, ontologies and, finally, transport protocols.

The transport mechanism, in particular, is like a chameleon because it adapts to each situation, by transparently choosing the best available protocol. Java RMI, event-notification, HTTP, and IIOP are currently used, but more protocols can be easily added via the MTP and IMTP JADE interfaces. Most of the interaction protocols defined by FIPA are already available and can be instantiated after defining the application-dependent behaviour of each state of the protocol.

SL and agent management ontology have been implemented already, as well as the support for user-defined content languages and ontologies that can be implemented, registered with agents, and automatically used by the framework.

Apart from these FIPA platforms, the **Java Community Process** (JAS) addressing Java Interfaces for agent services has recently started [JAS-2000]. The Java Agent Services project is an initiative to define an industry standard specification and API for the development of network agent and service architectures. There is of course no doubt that the most pervasive technology in use today for creating FIPA agent systems is Java. However, to date there exists no standard Java API for creating them, an omission that must be rectified if agents are to penetrate the business applications world. The JAS initiative intends to answer this requirement by developing an API, in the 'javax.agent' namespace, that instantiates the architectural features of the FIPA Abstract Architecture. This not only acts as a validation of the Abstract Architecture, but also forms the basis for creating commercial applications based on FIPA specifications.

2.4 Conclusion

As it has been said, the project could work well with a multiagent architecture in some parts of the system to afford the project objectives. Agents technology will be used to make easier the development of certain parts of the project. And ALFANET can benefit of the standardization effort already done by external organizations in the agent technology field.

FIPA standards are the currently accepted standards for heterogeneous and interacting agents and agent-based systems. There exist also platforms that allow the development of multi-agent systems based on the FIPA standards. One of these platforms is JADE, which has already been briefly described. In addition to the specifications of this platform which show its great potential, UNED can speak advisory about it since a multiagent architecture has been implemented in JADE there, obtaining agents FIPA compliant. Nevertheless, we should also pay attention to the evolution of the JAS project. Currently, this project is in its initial stage, but it will possible provide in a near future a standard Java API for the development of FIPA compliant agents.

As summary, the ALFANET agents will be FIPA compliant taking special attention to the FIPA standards relevant for the communication between agents (ACL - Agent Communication Language, and CL - Content Language)

3. General conclusions

From the learning point of view we recommended IMS-LD, IMS-LIP and IEEE LOM. The objective of ALFANET to offer a highly adaptive, personalized learning experience including a variety of pedagogical methods requires the capability to model both structure and process, including the specification of roles and activities. IMS-LD offers this capability and equally important in depth knowledge of LD is available and directly accessible. IMS-LIP can be used as a solid start for building the student model. IEEE LOM can be used to support part of the adaptive functionality of ALFANET. The meta-data specifications create a uniform way for describing learning resources so that they can be more easily found (discovered) and subsequent used. Other specifications (e.g. IMS Question&Test, IMS Enterprise, IMS Simple Sequencing, …) all cover relevant ground but, at this moment, do not cover key functionality required by ALFANET.

On knowledge management standards it would certainly be advisable to use existing ontology for the domains used in the project when they exist. Domain knowledge will certainly be needed to drive machine learning techniques applied by the agents.

On multi-agent architecture standards, the use of a FIPA platform will facilitate the compliance of current FIPA standards, paying special attention to the FIPA standards relevant for the communication between agents (ACL - Agent Communication Language, and CL - Content Language). In order to let agents communicate with the rest of modules of the system (educational subsystem) it could be useful to design a part of the dossier as a data structure common the both agent and educational subsystem. Both subsystems read and write to this common dossier.

.