INFORMATION SOCIETY TECHNOLOGIES (IST) PROGRAMME

From Information to Knowledge

Diogene

D8.1 e-Learning Standard Monitoring First Report – version 1.0

Prepared for the European Commission
under Contract No. IST-2001-33358 as a deliverable from

WP 8: Standard Monitoring and System Upgrading

Date of issue: 20/12/2003
Version: 1.0
Distribution: public

Revision History

Date	Version	Circulation	Reviewed paragraphs	Authors	Responsible Partner	Approved by
20/12/03	1.0	Public	Initial Version	DIIMA	DIIMA	Enver Sangineto

Table of Contents

D8.1 ***e-Learning Standard Monitoring First Report – version 1.0***

1. INTRODUCTION

For standards already applied in Diogene it will contain the description of novelties introduced by new versions and a cost/benefit estimation of their upgrade
For standards not yet applied in Diogene it will include a critical review and a cost/benefit estimation of their adoption

This survey has the following structure:

- Chapter 2 provides some introductory material useful in the rest of the document.
- Chapter 3 discusses some e-learning standards of interest.
- Chapter 4 talks about the standards currently under Examination for the Diogene project.
- Chapter 5 critically comments the adoption of the standards discussed in the previous chapter in the Diogene project.

1.1. REFERENCES

- [1] CRMPA DIOGENE Glossary, 2002.
- [2] IEEE P1484.12 Learning Object Metadata Working Group. LOM Approved Working Draft 4. (http://ltsc.ieee.org/doc/wg12/LOM_WD4.PDF).
- [3] IMS LIP Information Model Specification, 2001, (http://www.imsproject.com/profiles/lipinfo01.html).
- [4] IMS Learning Resource Meta-Data Best Practice and Implementation Guide, 2001, version 1.2.1, Final Specification (http://www.imsproject.org/specifications.html).
- [5] IMS Question and Test Interoperability: an overview, 2001. Public draft specification version 1.2 (http://www.imsproject.org/specifications.html).
- [6] L. Aiello and A. Micarelli, 1990. SEDAF: An intelligent educational system for mathematics. Applied Artificial Intelligence- An International Journal 4 (15), pp. 15-36.
- [7] J. R. Anderson, C. F. Boyle and G. Yost, 1985. The geometric Tutor. Proceedings of the IJCAI,. Los Angeles, CA.
- [8] J. R. Anderson and B. J. Raiser, 1985. The LISP Tutor. Byte 10 (4).
- [9] T. Berners-Lee. Semantic Web Road map. World Wide Web Consortium Archive. http://www.w3.org/DesignIssues/Semantic.html.
- [10] T. Berners-Lee. Interpretation and Semantics on the Semantic Web. World Wide Web Consortium Archive (http://www.w3.org/DesignIssues/Interpretation.html).
- [11] D. G. Bodrow and T. Winograd, 1977. An overview of KRL, a knowledge representation language. Cognitive Science 1, pp. 3-46.
- [12] B. Bos, 1999. XML in 10 points. World Wide Web Consortium Archive. (http://www.w3.org/XML/1999/XML-in-10-points.html).
- [13] R. J. Brachman and H. J. Levesque, 1984. The tractability of subsumption in frame-based description languages. In Proceedings of AAAI-84, 4th Conference of the American Association of Artificial Intelligence, pp. 34-37.
- [14] R. J. Brachman and H. J. Levesque editors, 1985. Readings in knowledge representation. Morgan Kaufmann, San Matteo CA.
- [15] M. Cialdea, A. Micarelli, D. Nardi, J. C. Spohrer and L. Aiello, 1990. Meta-level reasoning for diagnosis in intelligent tutoring systems. Technical Report, Dipartimento di Informatica e Sistemistica, Universita' di Roma "La Sapienza".
- [16] S. Cranefield, 2001. UML and the Semantic Web.

- [17] S. Cranefield and M. Purvis, 1999. UML as an Ontology Modelling language. Proceedings of the Workshop on Intelligent Information Integration, 16th IJCAI.
- [18] F. Donini, M. Lenzerini, D. Nardi and A. Schaerf, 1992. From subsumption to instance checking. Technical Report, Dipartimento di Informatica e Sistemistica, Universita' di Roma "La Sapienza".
- [19] F. Donini, M. Lenzerini, D. Nardi and A. Schaerf, 1996. Reasoning in Description Logics. In G. Brewka editor, Principles of Knowledge Representation and Reasoning, Studies in Logic, Language and Information, pp. 193-238. CLSI Publications.
- [20] D. Dubois and H. Prade, 1980. Fuzzy sets and systems. Theory and Applications. Academic Press.
- [21] D. Fensel, F. van Harmelen, I. Horrocks: OIL: A Standard Proposal for the Semantic Web. Deliverable 0 in the European IST project OnToKnowledge. (http://www.ontoknowledge.org/oil/downl/otk.del02.pdf).
- [22] Genesereth. Logical Foundation of Artificial intelligence.
- [23] M. R. Genesereth and R. E. Fikes, 1992. Knowledge Interchange Format, Version 3.0 Reference Manual. Computer Science Department, Stanford University, Technical Report Logic-92-1, June 1992.
- (http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps).
- [24] J. R. Hartley and D. H. Sleeman, 1973. Towards intelligent teaching systems. International Journal of Man-Machine Studies, 5, pp. 215-236.
- [25] J. Heflin, J. Hendler, S. Luke. SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078 (UMIACS TR-99-71). 1999. (http://www.cs.umd.edu/projects/plus/SHOE/pubs/techrpt99.pdf).
- [26] R. Iannella, 1998. An idiot's guide to the Resource Description Framework. The new Review of Information Networking 4.
- [27] A. N. Kolomogorov, 1956. Foundations of the theory of probability. Chelsea, NY.
- [28] H. J. Levesque and R. J. Brachman, 1987. Expressiveness and tractability in knowledge representation and reasoning. Computational Intelligence 3, pp. 78-93.
- [29] J. W. Lloyd, 1986. Foundations of Logic Programming. Springer Verlag, Berlin.
- [30] G. F. Luger and W. A. Stubblefield, 1998. Artificial Intelligence. Structures and strategies for complex problem solvers. Addison-Wesley Longman.
- [31] A. Micarelli, F. Mungo, F. S. Nucci and L. Aiello, 1991. SAMPLE: An Intelligent Educational System for Eletrical Circuits. Journal of Artificial Intelligence in Education, 2 (3), pp. 83-99.
- [32] E. Miller, 1998. An introduction to the Resource Description Framework (http://www.dlib.org/dilib/may98/miller/05miller.html).
- [33] M. Minsky, 1975. A framework for representing knowledge. I P. J. Winston editor, The Psychology of Computer Vision, McGraw-Hill NY pp. 211-277.
- [34] J. F. Nicaud and M. Vivet, 1988.Les tuteurs intelligents: Realisations et tendance de recherches. In Technique et Science Informatiques, 7 (1), pp. 21-45.
- [35] W. Paper, 2001. Capitalizing on the Learning Object Economy. Learning Objects Network.
- [36] R. Reiter, 1980. A logic for default reasoning. Artificial Intelligence 13, pp. 81-132.
- [37] E. Rich and K. Knight. Artificial Intelligence, second edition. McGraw-Hill, 1991.
- [38] S. Russel and P. Norvig. Artificial Intelligence, a modern approach. Prentice Hall, 1995.
- [39] F. Sebastiani, 1996. Logica e Rappresentazione della Conoscenza. Servizio Editoriale Universitario di Pisa.
- [40] D. H. Sleeman, 1987. PIXIE: A shell for developing intelligent tutoring systems. In Artificial Intelligence and Education: Learning environments and tutoring systems, R. H. Lawler and M. Yazdani Eds. Norwood NJ: Ablex Publishing, pp. 239-265.
- [41] H. Stuckenschmidt and H. Wache, 2000. Context Modelling and Transformation for Semantic Interoperability. In: Knowledge Representation meets Databases - Proceedings of the Workshop at ECAI 2000. (http://www.tzi.de/~heiner/public/KRDB-00.pdf).
- [42] L. A. Zadeh, 1975. Fuzzy logic and approximate reasoning. Synthese 30.
- [43] W3C. Resource Description Framework (RDF) Schema Specification 1.0, 2000. W3C Candidate Recommendation 27 March 2000. (http://www.w3.org/TR/2000/CR-rdf-schema-20000327).
- [44] Advanced Distributed Learning. SCORM Overview. (http://www.adlnet.org/index.cfm?fuseaction=scormabt).
- [45] Advanced Distributed Learning (ADL). (http://www.adlnet.org/)

- [46] Outline Processor Markup Language (OPML). (http://www.opml.org/)
- [47] J. Allen. Making a Semantic Web. (http://www.xml.com/pub/a/2000/07/17/syndication/rss.html)
- [48] R. E. Kent. Conceptual Knowledge Markup Language: The Central Core. The Ontology Consortium. (http://sern.ucalgary.ca/ksi/kaw/kaw99/papers/Kent1/CKML.pdf)
- [49] R. Wille, 1982. Restructuring lattice theory: an approach based on hierarchies of concepts, in, I. Rival (ed.), Ordered Sets, Reidel.
- [50] B. Ganter and R. Wille, 1989. Conceptual scaling, in, F. Roberts (ed.), Applications of Combinatorics and Graph Theory in the biological and Social Sciences, Springer-Verlag.
- [51] B. Barwise and J. Seligman, 1997. Information Flow: The Logic of Distributed Systems, Cambridge University Press.
- [52] M. Biezunski, M. Bryan, S. R. Newcomb (eds), 1999. Topic Maps: Information Technology - Document Description and Markup Languages, ISO/IEC 13250:2000. (http://www.y12.doe.gov/sgml/sc34/document/0129.pdf)
- [53] Edutella. (http://edutella.jxta.org/)
- [54] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, T. Risch, 2002. Edutella: A P2P Networking Infrastructure Based on RDF. WWW2002, Hawaii. (http://edutella.jxta.org/reports/edutella-whitepaper/)
- [55] Stanford University. Protégé 2000. (http://protege.stanford.edu)
- [56] Stanford University. Planning a Protégé 2000 Project. Protégé 2000 User Guide. (http://protege.stanford.edu/doc/users_guide/)
- [57] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative Ontology Development for the Semantic Web. (http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/ 2002_iswc_ontoedit.pdf)
- [58] S. Bechhofer, I. Horrocks, C. Goble and R. Stevens. OilEd: a Reason-able Ontology Editor for Editor for the Semantic Web. (http://www.cs.man.ac.uk/~horrocks/Publications/ download/2001/oiled-dl.pdf)
- [59] J. C. Apirez, O. Corcho, M. Fernandez-Lopez and A. Gomez-Perez, 2001. WebODE: a Scalable Workbench for Ontological Engineering. K-CAP'01, Canada. (http://mkbeem.elibel.tm.fr/paper/KCAP2001-35.pdf)
- [60] Stanford University Knowledge Systems Laboratory. Ontolingua. (http://www.ksl.stanford.edu/software/ontolingua/)
- [61] Information Sciences Institute. SENSUS Ontosaurus. (http://mozart.isi.edu:8003/sensus2/)
- [62] U. Visser, H. Stuckenschmidt, H. Wache, T. Vogele, 2000. Enabling Technologies for Interoperability. NEC Research Index. (http://citeseer.nj.nec.com/459286.html)
- [63] Stanford University. Scalable Knowledge Composition (SKC). (http://www-db.stanford.edu/SKC/)
- [64] W3C. Requirements for a Web Ontology Language. W3C Working Draft 08 July 2002. (http://www.w3.org/TR/webont-req/)
- [65] MIT. Open Knowledge Initiative. (http://web.mit.edu/oki/)
- [66] m-Learning Project. (http://www.m-learning.org/)

1.2. INVOLVED ORGANIZATIONS

The e-learning standard definition landscape is quite a complex one. Currently, half a dozen organizations are working to develop industry standards in this field. They include the following:

- IMS, a consortium of members that include major software developers and vendors, training and education representatives, and government agencies.
- ADL, an initiative of the U.S. Department of Defence which aims to ensure the interoperability of future e-learning technologies purchased by the U.S. government.
- LTSC, a branch of the Institute of Electrical and Electronic Engineers (IEEE) with a long-standing reputation as an accrediting body for technology standards.
- AICC (the Aviation Industry CBT Committee), an association of technology developers that spans well beyond the aviation industry.

All these groups are increasingly working together in order to harmonize their efforts towards a comprehensive set of standards for distance learning.

2. E-LEARNING STANDARDS EMPLOYED IN DIOGENE

This section describes the standards that are currently adopted in the Diogene project.

IMS QTI

The IMS Question & Test Interoperability Specification provides proposed standard XML language for describing questions and tests. The specification has been produced to allow the interoperability of content within assessment systems. This will be useful for publishers, certification authorities, teachers, trainers, publishers and creators of assessments, and the software vendors whose tools they use. Authoring tools, and publishers, may publish XML and this data can be imported into other authoring tools and delivery systems.

More specifically, the IMS Question & Test Interoperability (QTI) specification describes a basic structure for the representation of question (item) and test (assessment) data and their corresponding results reports. Therefore, the specification enables the exchange of this item, assessment and results data between Learning Management Systems, as well as content authors and, content libraries and collections. The IMS QTI specification is defined in XML to promote the widest possible adoption. XML is a powerful, flexible, industry standard markup language used to encode data models for Internet-enabled and distributed applications. The IMS QTI specification is extensible and customisable to permit immediate adoption, even in specialized or proprietary systems. Leading suppliers and consumers of learning products, services and content contributed time and expertise to produce this final specification. The IMS QTI specification, like all IMS specifications, does not limit product designs by specifying user interfaces, pedagogical paradigms, or establishing technology or policies that constrain innovation, interoperability, or reuse.

Related material can be found at: http://www.imsglobal.org/question/index.cfm

IMS CONTENT PACKAGING

The IMS Content Packaging Specification provides the functionality to describe and package learning materials, such as an individual course or a collection of courses, into interoperable, distributable packages. Content Packaging addresses the description, structure, and location of online learning materials and the definition of some particular content types.
The Content Packaging Specification is aimed primarily at content producers, learning management system vendors, computing platform vendors, and learning service providers. Learning materials described and packaged using the IMS Content Packaging XML format should be interoperable with any tool that supports the Specification. Content creators can develop and distribute material knowing that it can be delivered on any compliant system, thereby protecting their investment in rich content development.

This specification is included in the SCORM standard and will be discussed in detail in the related section.

Related documentation is available at: http://www.imsglobal.org/content/packaging/index.cfm

IMS LEARNER INFORMATION PACKAGE (LIP)

Used in the project for student management.

Another major standardization effort, the Learner Information Package (LIP), comes from the IMS, a consortium of institutions including government agencies, software developers and vendors, and training and education representatives.

Version 1.0 of the IMS Learner Information Package Specification was released to the public in March 2001. The IMS LIP has partly been derived from the IEEE PAPI Learner (versions 5.0 and 6.0).

The LIP specification provides a way of packaging learner information for exchange between disparate systems. It focuses on learner information, that is, the wide range of information that can be used by different systems to support the learner's activities. The semantics of the packages being exchanged may vary depending on the context; this is determined by the services participating in the exchange. Furthermore, learner information can be packaged from a variety of environments, not only human resources, student information and learning management systems.
An important aspect of the implementation of the XML-based specification to note is that nearly all LIP elements are optional. Depending on needs, data can be packaged to match the basic LIP segment structure or to match the structure of information on either side of the exchange. Either approach is acceptable.

LIP can be used for individual learner information packaging (for example, a student submitting his or her resume to an e-learning website) or for organizational exchange (both intra-organization, like data about employees, or extra-organization, like the certification of a student's achievements to a third-party institution).

The IMS LIP data structure.

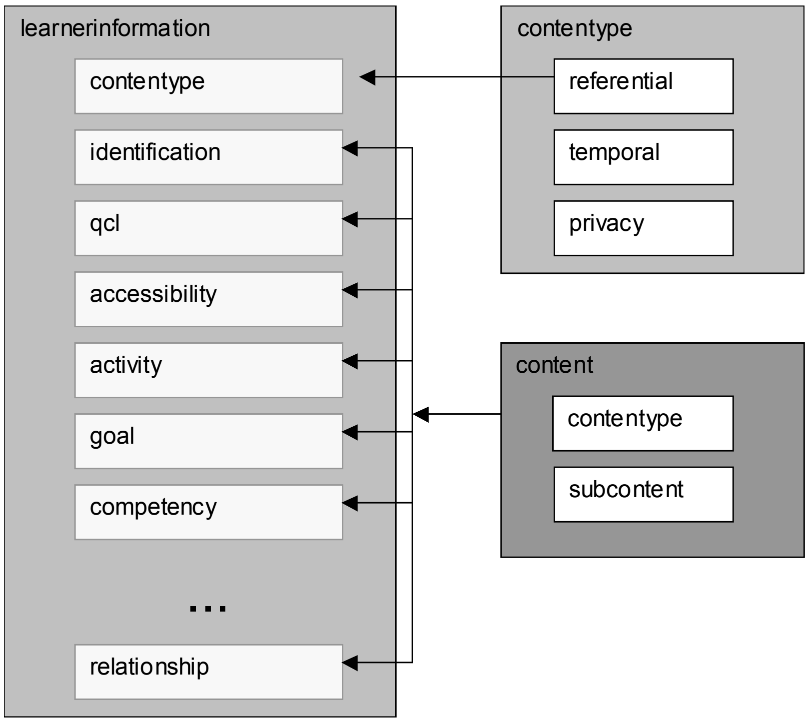


Figure 2.1.

CORE DATA STRUCTURES

LIP is structured around eleven core data structures, as follows:

1. Accessibility – Data regarding the accessibility of learner's information as defined through:
   - Language: the definition of a learner’s language proficiencies.
   - Preference: the definition of a learner’s cognitive, physical and technological preferences.
2. Activity – The activity the learner is engaging in, comprising:
   - Learning activity reference: an external reference mechanism to the learning materials.
   - Definition: the definition of the materials studied.
   - Product: the materials developed by the learners themselves.

- Testimonial: statements attesting to the capabilities of the learner.
- Evaluation: the results of the evaluations undertaken.

3. Affiliation – The learner's professional affiliations and associated roles.
    - Competency – The competencies of the learner.
    - Goal – The learner's goals and sub-goals.
4. Identification – The learner identification data. They comprise:
    - Formatted Name: the learner's name, formatted.
    - Name: the learner's name.
    - Address: the learner's addresses.
    - Contact info: electronic-based contact information about the learner.
    - Demographics: demographics information about the learner.
    - Agent: the representatives permitted to act on behalf of the learner.
5. Interest – Hobbies and recreational interests of the learner.
6. Qcl – A description of the qualifications, certifications and various licenses of a learner.
7. Relationship – the set of relationships that are to be defined between the learner and their identification, accessibility, qualifications, competencies, goals, activities, interests, transcripts, security keys and affiliations.
8. Security key – the security-related information for the given learner.
9. Transcript – the transcripts that summarize the performance of the learner.

A full, detailed list of all LIP data elements would be of little interest. What is important is that the standard has been designed to be extensible, in order to accommodate any possible learner data. Of course, the extensions obtained would be proprietary additions.

Relationship with the IEEE LTSC PAPI Specification

As mentioned earlier, the IMS LIP work incorporated the IEEE PAPI specification. Figure 4 describes such the relationship.

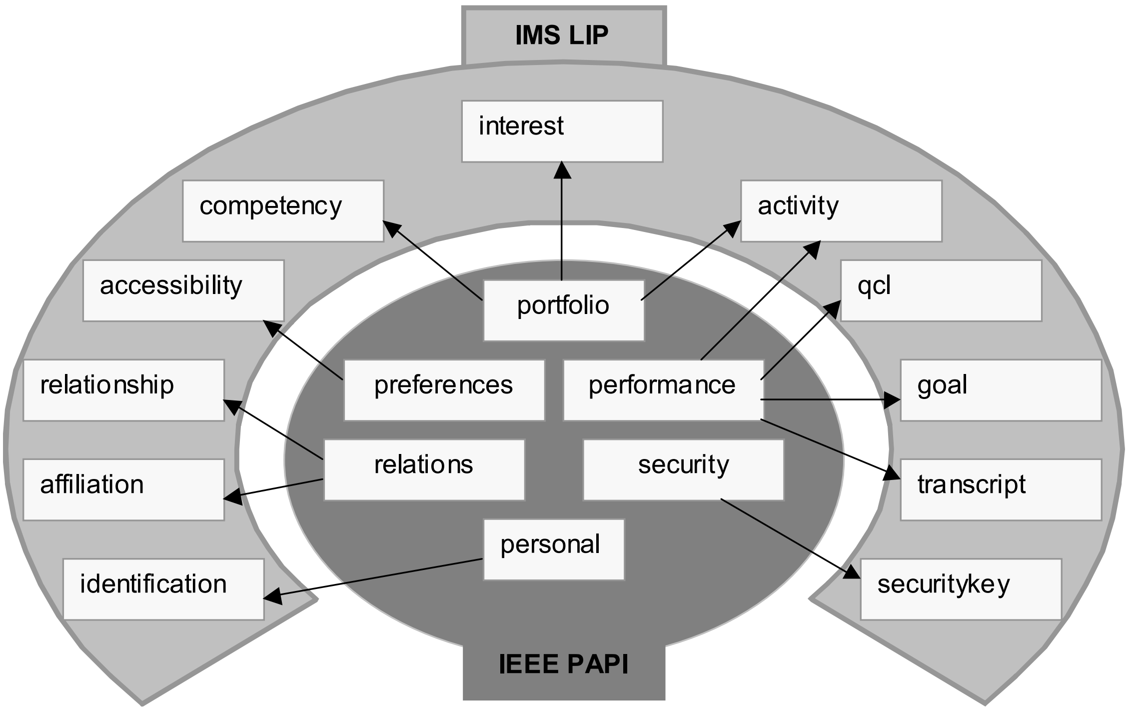


Figure 2.2.

An arrow in Figure 4 indicates the mapping between one data structure and another. Hence, data belonging to the IEEE PAPI personal group can be put in the identification IMS LIP data group when using the latter specifications.

Compatibility with the IMS Content Packaging Specification

The synergic use of IMS LIP and IMS CP is important in real-world e-learning systems, and is discussed here. The IMS Content Packaging specification can be used for the packaging of a LIP XML instance for a single learner and for the aggregation of several instances of a single learner or for multiple learners.

IMS/ IEEE Learning Object Metadata Standard (LOM) P1484.12 (Part of SCORM 1.3)

Documents on this standard are available at http://ltsc.ieee.org/wg12/

LOM stands for Learning Object Metadata and is a standard defined by the Learning Tecnology Standardization Committee (LTSC) of IEEE. As in the IMS Metadata Specification, the LOM standard has a hierarchical structure. Each of its elements can be mandatory, optional or conditional (in this case, the attribute values depend on the presence or absence of a value for another optionally defined attribute). Some attributes can have, as a value, a list of terms rather than a single term. This list can be ordered or unordered. Finally, the shared vocabularies of terms of the metadata values can be defined as: restricted (only the values of the vocabulary are admissible values for that attribute) or open (the vocabulary groups a suggested list of terms frequently used by applications but also other terms are admissible values for that attribute).
The main elements of LOM schema actually are 47 (against the only 15 of Dublin Core) and are grouped in 9 categories as shown in the following table .

Category	Description	Elements
General	Groups all context independent features together with the semantics descriptors of the resource.	Identifier, Title, CatalogEntry, Language, Description, Keywords, Coverage, Structure, Aggregation Level.
Life Cycle	Groups the features related to the life cycle of the resource.	Version, Status, Contribute.
Meta Metadata	Groups the features of the description itself rather than the resource which is described.	Identifier, CatalogEntry, Contribute, Metadata Scheme, Language.
Technical	Groups the technical features of the resource.	Format, Size, Location, Requirements, Installation Remarks, Other Platform Requirements, Duration.
Educational	Groups the didactic and pedagogical features of the resource.	Interactivity Type, Learning Resource Type, Interactivity Level, Semantic Density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Description, Language
Rights Management	Groups the resource features depending on the type of use we plan for it.	Cost, Copyright and Other Restrictions, Description.
Relation	Groups the resource features which relate it to other resources.	Kind, Resource, Identifier, Description, CatalogEntry.
Annotation	Allows comments to be made about the resource's educational content.	Person, Date, Description.

Classification	Describes a particular classification system in which the resource is defined (such as taxonomies, conceptual graphs, and so on).	Purpose, TaxonPath, Description, Keywords.

ADL SCORM

The SCORM (Sharable Content Object Reference Model) [44] is a set of specifications for developing, packaging and delivering education and training materials whenever and wherever they are needed. SCORM is a product of the U.S. Government's initiative in Advanced Distributed Learning (ADL) [45], which aims to provide access to high-quality materials that are easily tailored to individual learner needs.
The SCORM applies current technology developments – from groups such as the IMS Global Learning Consortium, the Aviation Industry Computer-Based Training Committee, the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) and the Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC) – to a specific content model to produce recommendations for consistent implementations by the vendor community.

Although choice and competition in the marketplace are generally a good thing, the rapid growth of LMS vendors has created a significant problem for content authors. Without a common specification for packaging online courses, LMS vendors organize their content databases in any fashion they choose. As a result, every vendor is using a different format for packaging their courses. Although they are all delivered by http, or some other standard protocol, if an author tries to move a course from one LMS to another, they find that this task is very time-consuming. In many cases, moving from one LMS vendor to another requires complete reconstruction of the course materials.

ADL is evolving a set of specifications for packaging online courses that will not only make it easier to transport a course from one LMS to another, it will also achieve other desirable goals as well. A course packaged following the SCORM specifications can be transported from one LMS to another with minimum modifications. SCORM-compliant courses leverage course development investments by ensuring that compliant courses are:

- Accessible: the ability to locate and access instructional components from one remote location and deliver them to many other locations.
- Interoperable: the ability to take instructional components developed in one location with one set of tools or platform and use them in another location with a different set of tools or platforms.
- Durable: the ability to withstand technology changes without redesign, reconfiguration or recoding.
- Reusable: the flexibility to incorporate instructional components in multiple applications and contexts.

Some of the main organizations involved in SCORM.

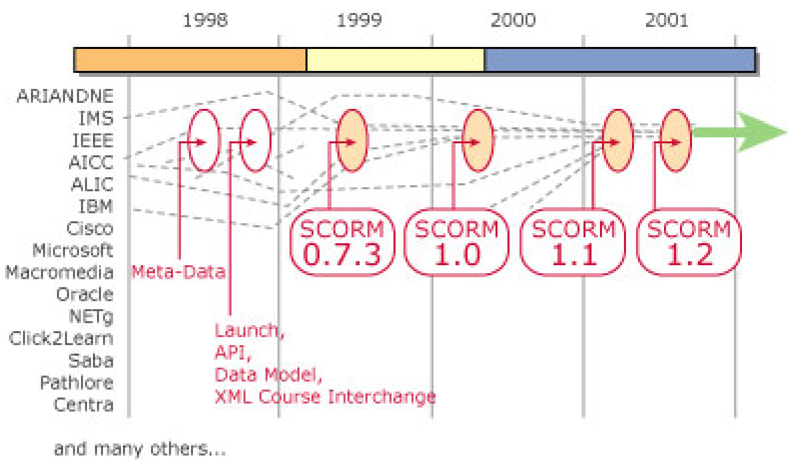


Figure 2.4.

As shown in the following figure, all of the specifications and guidelines contained or referenced can be viewed as separate "books" gathered together into a growing library.

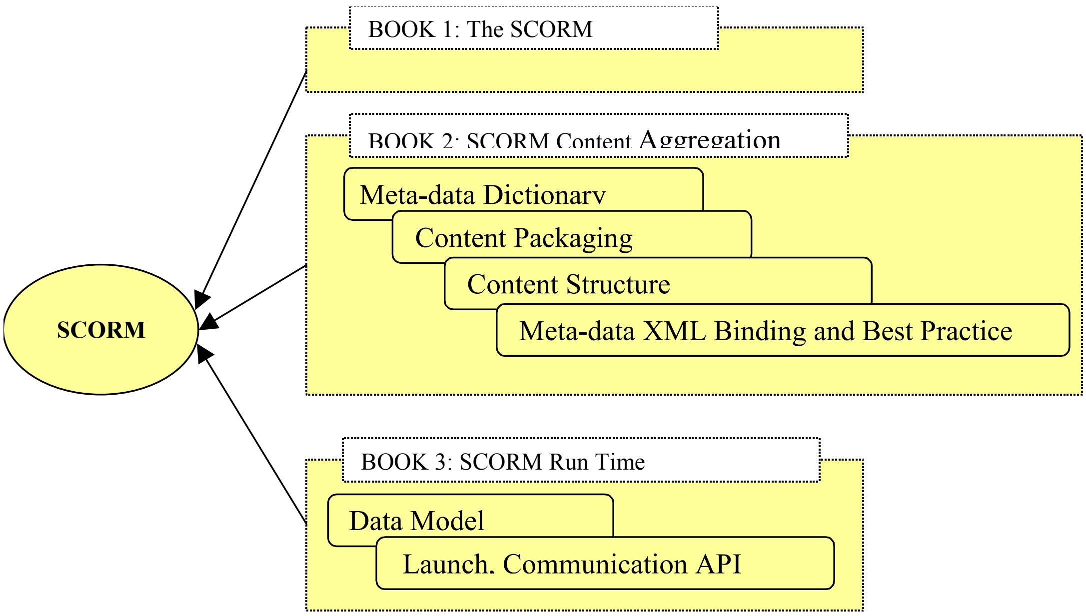


Figure 2.5.

IEEE STANDARD FOR XML SCHEMA BINDING FOR LEARNING OBJECT METADATA (PART OF SCORM 1.3)

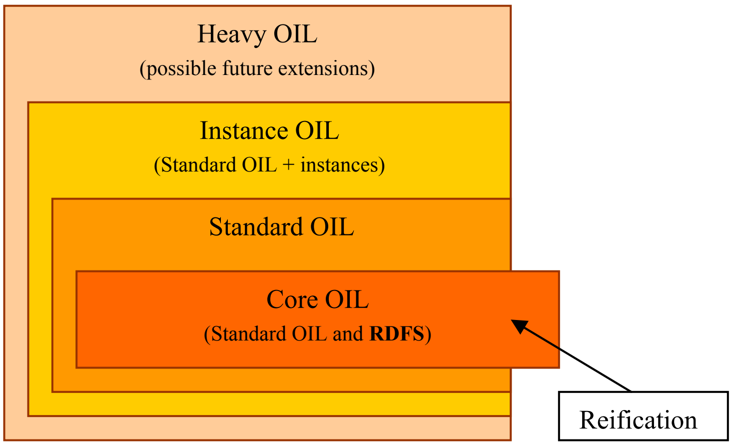
This specification has been mentioned in the previous section about SCORM and it is currently adopted in the Diogene project for representing LOM instances.

The following two standards represent ontologies in the Diogene project.

OIL

The Ontology Inference Layer (OIL) is another ontology representation language compatible with Web standards. It derives from Frame representation languages (Section 4.1) and Description Logics (Section 3.4); indeed, its objective is to build on top of RDF Schema by adding modelling constructs for Description Logics.
OIL is based on a multi-layer architecture. Each layer is an extension of the level below, obtained by adding expressiveness to it. The aim of this architecture is to allow intelligent agents not able to perform complex inferences to access only low layers of a description, in order to be able to partially understand it. The following figure shows the relation between OIL and RDF.

Figure 3.1



As shown in the figure, Core OIL largely overlaps with RDFS (exception for reification features). This means that simple RDF-based agents can also process OIL ontologies, although they will not be able to completely understand them.
Standard Oil includes a set of modelling primitives which ensure an expressive power enough to perform inferences. This layer allows individual variables to be expressed in term definitions.
Instance OIL includes a complete instance integration and database functionalities.
Heavy OIL, finally, is a layer providing possible future extension.
An OIL ontology is a structure whose elements can themselves be sub-structures. Moreover, these elements can be mandatory, optional (indicated by the symbol "?") or multiple (indicated by the symbol "+" or "*", depending on whether the empty value is admissible).

OWL

The OWL Web Ontology Language [64] is being designed by the W3C Web Ontology Working Group as a revision of DAML+OIL. It takes DAML+OIL as its basis and its expressiveness will be close to that of DAML+OIL.
The following introduction accounts for the latest extensions (as of 18 August 2003).

OWL can be thought of being composed of three sublanguages. Such languages are designed for use by specific communities of implementers and users. We report them here in order of increasing expressiveness:

- *OWL Lite* supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies. Owl Lite also has a lower formal complexity than OWL DL.
- *OWL DL* supports those users who want the maximum expressiveness while retaining computational

completeness (all conclusions are guaranteed to be computed) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

- *OWL Full* is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Each of these languages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded. Hence, for example every legal OWL Lite ontology is a legal OWL DL ontology, while every legal OWL DL ontology is a legal OWL Full ontology and every valid OWL Lite conclusion is a valid OWL DL conclusion. Finally, every valid OWL DL conclusion is a valid OWL Full conclusion.

Ontology developers adopting OWL should consider which sublanguage best suits their needs. The choice between OWL Lite and OWL DL depends on the extent to which users require the more-expressive constructs provided by OWL DL and OWL Full. The choice between OWL DL and OWL Full mainly depends on the extent to which users require the meta-modeling facilities of RDF Schema (e.g. defining classes of classes, or attaching properties to classes). When using OWL Full as compared to OWL DL, reasoning support is less predictable since complete OWL Full implementations do not currently exist.

OWL Full can be viewed as an extension of RDF, while OWL Lite and OWL DL can be viewed as extensions of a restricted view of RDF. Every OWL (Lite, DL, Full) document is an RDF document, and every RDF document is an OWL Full document, but only some RDF documents will be a legal OWL Lite or OWL DL document. Because of this, some care has to be taken when a user wants to migrate an RDF document to OWL. When the expressiveness of OWL DL or OWL Lite is deemed appropriate, some precautions have to be taken to ensure that the original RDF document complies with the additional constraints emposed by OWL DL and OWL Lite. Among others, every URI that is used as a class name must be explicitly asserted to be of type owl:Class (and similarly for properties), every individual must be asserted to belong to at least one class (even if only owl:Thing), the URI's used for classes, properties and individuals must be mutually disjoint.

In particular DAML+OIL is extended in two ways:

- A lower step-in threshold (which is perceived to be too high for DAML+OIL) is provided.
- A human-readable presentation syntax is provided, besides an RDF/XML-based syntax.

The W3C working draft specifies eight design goals for the Web ontology language:

- Shared ontologies. Ontologies should be publicly available and different data sources should be able to commit to the same ontology for shared meaning. Also, ontologies should be able to extend other ontologies in order to provide additional definitions.
- Ontology evolution. An ontology may change during its lifetime. A data source should specify the version of an ontology to which it commits. An important issue is whether or not documents that commit to one version of an ontology are compatible with those that commit to another. Both compatible and incompatible revisions should be allowed, but it should be possible to distinguish between the two. Note that since formal descriptions only provide approximations for the meanings of most terms, it is possible for a revision to change the intended meaning of a term without changing its formal description. Thus determining semantic backwards-compatibility requires more than a simple comparison of term descriptions. As such, the ontology author needs to be able to indicate such changes explicitly.
- Ontology interoperability. Different ontologies may model the same concepts in different ways. The language should provide primitives for relating different representations, thus allowing data to be converted to different ontologies and enabling a "web of ontologies."
- Inconsistency detection. Different ontologies or data sources may be contradictory. It should be possible to detect these inconsistencies.

- Balance of expressivity and scalability. The language should be able to express a wide variety of knowledge, but should also provide for efficient means to reason with it. Since these two requirements are typically at odds, the goal of the web ontology language is to find a balance that supports the ability to express the most important kinds of knowledge.
- Ease of use. The language should provide a low learning barrier and have clear concepts and meaning. The concepts should be independent from syntax.
- Compatibility with other standards. The language should be compatible with other other commonly used Web and industry standards. In particular, this includes XML and related standards (such as XML Schema and RDF), and possibly other modeling standards such as UML.
- Internationalisation. The language should support the development of multilingual ontologies, and potentially provide different views of ontologies that are appropriate for different cultures.

These design goals motivate a number of requirements for a Web Ontology Language. The Working Group currently feels that the requirements described below are essential to the language:

- Ontologies must be objects that have their own unique identifiers, such as a URI reference.
- Two terms in different ontologies must have distinct absolute identifiers (although they may have identical relative identifiers). It must be possible to uniquely identify a term in an ontology using a URI reference.
- Ontologies must be able to explicitly extend other ontologies in order to reuse terms while adding new classes and properties.
- Resources must be able to explicitly commit to specific ontologies, indicating precisely which set of definitions and assumptions are made.
- It must be possible to provide meta-data for each ontology, such as author, publish-date, etc. The language should provide a standard set of common metadata properties. These properties may or may not be borrowed from the Dublin Core element set.
- The language must provide features for comparing and relating different versions of the same ontology. This should include features for relating revisions to prior versions, explicit statements of backwards-compatibility, and the ability to deprecate terms.
- The language must be able to express complex definitions of classes. This includes, but is not limited to, sub classing and Boolean combinations of class expressions (i.e., intersection, union, and complement).
- The language must be able to express the definitions of properties. This includes, but is not limited to, sub properties, domain and range constraints, transitivity, and inverse properties.
- The language must provide a set of standard data types. These data types may be based on XML Schema data types.
- The language must include features for stating that two classes or properties are equivalent.
- The language must include features for stating that pairs of identifiers represent the same individual.
- In general, the language will not make a unique names assumption. That is, distinct identifiers are not assumed to refer to different objects (see the previous requirement). However, there are many applications where the unique names assumption would be useful. Users should have the option of specifying that all of the names in a particular namespace or document refer to distinct objects.
- The language must provide a way to allow statements to be "tagged" with additional information such as source, timestamp, confidence level, etc.
- The language must support the ability to treat classes as instances. This is because the same concept can often be seen as a class or an individual, depending on the perspective of the user.
- The language must support the definition and use of complex/ structured data types. These may be used to specify dates, coordinate pairs, addresses, etc.
- The language must the support the specification of cardinality restrictions on properties. These restrictions set minimum and maximum numbers of object that any single object can be related to via the specified property.
- The language should have an XML serialisation syntax.
- The language should support the specification of multiple alternative user-displayable labels for the objects within an ontology. This can be used, for example, to view the ontology in different natural languages.

- The language should support the use of multilingual character sets.
- In some character encodings, e.g. Unicode based encodings, there are some cases where two different character sequences look the same and are expected, by most users, to compare equal. Given that the W3C I18N WG has decided that early uniform normalization (to Unicode Normal Form C) as the usual approach to solving this problem, any other solution needs to be justified.

An OWL ontology is a sequence of axioms and facts, plus inclusion references to other ontologies which are considered to be included in the ontology. OWL ontologies are web documents, and can be referenced by means of a URI.
Axioms are used to associate class and property IDs with either partial or complete specifications of their characteristics, and to give other logical information about classes and properties.
Class axioms include the common, widely understood, frame idiom. The abstract syntax used here is meant to look somewhat like the syntax used in some frame systems. Each frame-like class axiom contains a collection of more-general classes; a collection of local property restrictions, in the form of restriction constructs; and a collection of descriptions. The restriction construct gives the local range of a property, how many values are permitted, and a collection of required values. Descriptions are used to specify boolean combinations of restrictions and other descriptions as well as construct sets of individuals. Classes can also be specified by enumeration or be made the same or disjoint.
Properties can be the equivalent to or subproperties of others; can be made functional, inverse functional, or transitive; and can be given global domains and ranges. However, most information about properties is more naturally expressed in restrictions, which allow local cardinality and range information to be specified.
Facts state information about particular individuals in the form of a class that the individual belongs to plus properties and values. Individuals can either be given an individualID or be anonymous (blank nodes in RDF terms).

To recap, OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is a revision of the DAML+OIL web ontology language incorporating lessons learned from the design and application of DAML+OIL.

In the following section we will introduce an older knowledge representation format, still interesting for several aspects.

SHOE

Simple HTML Ontology Extensions (SHOE) is a language for knowledge representation based on the ontology approach (see Section 4.2.2) and XML syntax. Like XML, SHOE is also an extension of HTML.
In the present work on knowledge representation methods, we have often underlined the necessity of choice, in defining a representation formalism, between large expressive power or good computational costs. SHOE is not as expressive as other ontology systems such as Ontolingua, based on the logic language KIF (see Section 4.2.2). On the contrary, it is quite simple and has a tractable computational complexity.
As in the case of RDF, it is also possible in SHOE to include, in a same description, different ontology sources specifying their URL, as shown in the example below.

OTHER INITIATIVES NOT INCLUDED IN THE DIOGENE PROJECT

This section covers other standardization initiatives that use student modeling, however marginally.

IEEE P1484.2 (PAPI LEARNER)

Public and Private Information (PAPI) for Learners (PAPI Learner) is a standard effort aimed at providing the syntax and semantics of a student model, including knowledge, learning styles, skills, abilities, records and personal information, all at multiple levels of granularity. This standard specifies the syntax and semantics of a "Learner Model", which characterizes a learner (either a student or knowledge worker) and his or her knowledge/abilities. This will include elements such as knowledge (from coarse to fine-grained), skills, abilities, learning styles, records, and personal information. The standard will allow these elements to be represented in multiple levels of granularity, from a coarse overview, down to the smallest conceivable sub-element. It will allow different views of the Learner Model (learner, teacher, parent, school, employer, etc.) while addressing the sensitive issues of privacy and security.
The working group for the Learner Model (P1848.2) has the following purposes:

- To enable learners (students or knowledge workers) to build lifelong personal learner models.
- To enable personalized instruction and effective instruction.
- To provide educational researchers with a standardized source of data.
- To provide a foundation for the development of additional educational standards, from a student-centred learning focus.
- To provide architectural guidance to developers of education systems.

A simple view of the IEEE 1484.2 overall organization is provided in Figure 2.

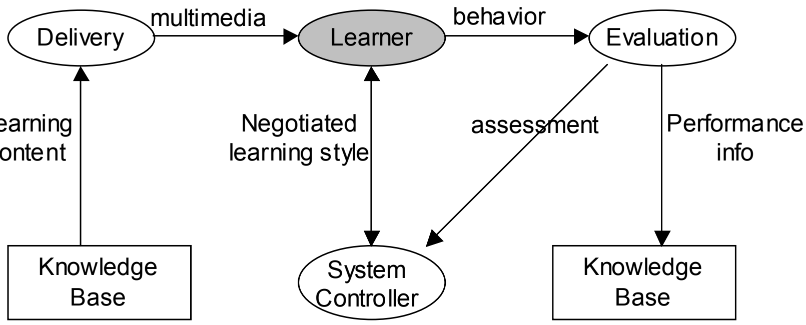


Figure 2.3.

The main architectural feature of the PAPI Learner standard is its logical division. It separates the security and the administration of several types of learner information (also called Profile Information or Learner Profiles):

- Personal information like name, address and social security number. It is not directly related to the measurement and recording of learner performance and is primarily concerned with administration. Usually this type of information is private and secure.
- Relations information, e.g., cohorts, classmates. This concerns the learner's relationship to other users of learning technology systems, such as teachers, practitioners, and other learners.
- Security information. This is concerned with the learner's security credentials, such as passwords, challenges/responses, private and public cryptographic keys, and biometrics.
- Preference information: useful and unusable I/O devices, learning styles and physical limitations. It describes preferences that may improve human-computer interactions.
- Performance information, like grades, interim reports, log books. This pertains to the learner's history, current work or future objectives and is created and used by learning technology components to supply enhanced learning experiences.
- Portfolio information: accomplishments, works and so on. This information is a representative collection of a learner's works or references to them that is intended to illustrate and justify the student's abilities and attainments.

The PAPI Learner Standard is not limited to these six types of information and may be integrated with other systems, protocols, formats, and technologies.

Generally speaking, standards that attempt to be omni-comprehensive risk becoming large and unmanageable. The designers of PAPI Learner aimed at concrete diffusion by avoiding catch-all specifications and focusing on essential, learning-related data while providing an extension mechanism for customization. Another aim of this standard is to ensure different levels of granularity within information definitions. Hence, there are no coarse-grained information definitions, but rather a continuum, with many gradations and variations. For example, learner information can vary in "distance" from the trainee; local information, typically, is characterized by online availability, higher performance access, and fewer security restrictions. There are also various possible degrees regarding privacy.

PAPI Learner information has been designed in order to follow a well-defined set of requisites. In particular PAPI Learner information must support:

- Cultural conventions (like measure units, currencies and other linguistic conventions).
- Institutional conventions. This mainly concerns learning institutions, each supporting its own conventions (like grading systems, or course denominations). An engineering goal of PAPI Learner is to "let the market solve the problem" of choosing/reducing the number of coding schemes to the "right" level.
- Simple application paradigms, in order to promote adoption. That is, it should require minimal effort to incorporate PAPI Learner codings, APIs, and protocols into existing real-world applications.
- Other engineering requirements include:
  - Controlling access to information to the extent necessary.
  - Maximizing performance of accessing data.
  - Supporting varying data and information structures.
  - Supporting varying coding and extension mechanisms.
  - Supporting varying information partitioning schemes.
  - Letting the market choose the best coding scheme(s).
  - Supporting varying types of online, "sometimes", and offline connectivity.
  - Supporting varying geographic (nomadic) access to information.

Learners' data can be exchanged in three different ways: (i) by means of the external specification, i.e., only PAPI Learner coding bindings are used while some other data communication method is mutually agreed upon by data exchange participants; (ii) by using the control transfer mechanism to facilitate data interchange, e.g., PAPI Learner API bindings and (iii) by employing data and control transfer mechanisms. e.g., PAPI Learner protocol bindings.

Also security features are part of the conceptual model definition, as follows:

- Session-View-Based. Security features are provided on a per-session, per-view basis. Each security session is initiated by an accessor (a user or agent that requests access). The accessor provides security credentials that authenticate the accessor, authorize the accessor, or both. A view represents a portion of PAPI Learner information and is similar to the notion of a database "view". Each view established represents a session, i.e., the "session" represents the duration of access and the "view" represents the scope of access.
- Security Parameter Negotiation. Data interchange participants negotiate security parameters prior to, during, and after each session. The security parameters are defined in the PAPI Learner bindings.
- Security Extension. Additional security features may be used that were not foreseen. The method of incorporating security extensions is defined in the PAPI Learner bindings.
- Access Control. Accessors may attempt read or write access to data elements, to create new data elements (separately or within aggregates), to destroy data elements (separately or within aggregates), and/or to change attributes of data elements. Other access methods, if any, are implementation-defined.
- Identification. The methods for identifying learners are implementation-defined. A related standard, IEEE 1484.13 ("Simple Human Identifiers"), defines the data type associated with a learner identifier.
- Authentication. The methods of authenticating users are outside the scope of the PAPI Learner Standard.
- De-identification. PAPI Learner prescribes that all information, except learner personal information,

should be de-identified (that is, students should not be identified). The methods of de-identifying learners and their information are outside the scope of this Standard.

- Authorization. The methods of authorizing operations are implementation-defined.
- Delegation. The methods of delegating administration, authority, or credentials are implementation-defined.
- Non-Repudiation. The methods of non-repudiation are implementation-defined.
- Repudiation. The methods of repudiating data, users, or credentials are implementation-defined.
- Privacy. This Standard supports security frameworks and approaches that permit the implementation of a wide variety of privacy frameworks.
- Confidentiality. This Standard supports access controls and the partitioning of information types that permit the implementation of a wide variety of confidentiality frameworks.
- Encryption. PAPI Learner supports several security frameworks and techniques that permit the integration of various encryption models and technologies.
- Data Integrity. This Standard supports information assurance frameworks and approaches that permit the implementation of a wide variety of data integrity frameworks.
- Validation of Certificates. PAPI Learner does not require validation of student performance information, but supports the parameterisation of automated validation.
- Digital Signature. This Standard adopts third-party signing frameworks harmonized with ISO/IEC 15945.
- IEEE 1484.2.3, PAPI Learner Information Security Notes, contains information about applying security techniques and technologies to PAPI Learner implementations.

SABA PROFILE FORMAT

Profile Format (http://www.saba.com/standards/ulf/Specification/specPROF.htm) 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. 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.

By employing Profile Format to describe the learners in a system, learning providers can extend their learning management architecture to support all of the following:

- Searches of critical learner metadata such as name, title, role, learning results, and held competencies and certifications
- Tracking the learning history of individual learners
- Assignment of competencies (with proficiency levels) and certifications to learners
- Assignment of learning goals to learners and tracking of progress towards fulfilment of those goals
- Creation of distributed profiles, where portions of a learner’s profile are provided by different sources
- Compatibility with standard web search engines

Profile Format is based on open standards and is designed to reflect the following principles:

- Compatibility with emerging industry standards for learning profiles, including ongoing work in IMS and IEEE.
- Extensibility to easily accommodate future growth and change.

Profile Format employs an XML standard known as RDF (Resource Description Framework), the standard for defining metadata to describe web-based resources. The use of RDF makes it possible to define a set of unique RDF properties and merge these properties with properties defined in existing standards, such as vCard and Dublin Core. RDF also provides a unified mechanism for manipulating and querying this merged metadata. Furthermore, the use of RDF allows Profile Format to support distributed profiles, where portions of a learner's profile are provided by different sources.

A Profile Format document is an RDF document that contains one or more Description elements, where each

element describes a learner in the system. Each Description element contains a unique identifier and a set of property/value pairs that fully describe the learner. These properties can draw from any of the Profile Format RDF schemas.
Each Description element has an attribute that unambiguously identifies the learner being described. This attribute can be either of the following:
· id
· about
The Description element can also include the xml:lang attribute for specifying the language in which the metadata description is authored. The xml:lang attribute contains the ISO 639 /RFC 1766 language code with an optional geographic identifier, such as en for English, or fr for French.

Profile Format subdivides learner information into the following categories:
· Personal information. This includes information such as name, address, title, role(s), and organisation membership. All personal information is represented using RDF mappings of the vCard specification.
· Learning information. Profile Format defines a set of RDF properties that capture information on a learner's current learning (current enrolments) and learning history (transcript).
The learning property specifies a URL to a learning resource described in a Catalog Format document. The specified resource is a held offering in the learner's transcript.
In its simplest form, the learning property contains only the URL of the held learning offering, for example:
<profile:learning rdf:resource="http://www.saba.com/learning/catalog.rdf#JAVA101"/>
The learning property can also be a structured property, with substructure properties that qualify the status of the learning offering and the conditions under which it was attained. For qualified instances of the learning property, the URL of the learning resource is captured using the rdf:value property.
· Goal information. Profile Format defines a set of RDF properties that capture information about a learner's goals. Goals can encompass both business and professional objectives for a learner and include the following additional information:
- planned actions for achieving the goal
- learning interventions
- accomplishments

The goal property defines a learner's goal.

The goal property can also be a structured property, with substructure properties that provide details about the goal and its status. For qualified instances of the goal property, the URL of a qualified goal is captured using the rdf:value property.
· Observation information. Profile Format defines a set of RDF properties that capture information reflecting a learner's progress towards specific goals. These observations track specific, measurable milestones on the path towards achieving a goal.
· Competency information. Includes information on held competencies. This category contains a pointer to a competency defined in a Competency Format document, with optional properties describing how the competency was attained.
· Certification information. This category contains a pointer to a certification track defined in a Certification Format document, with optional properties describing how the certification was attained.
· Preference information. Profile Format defines a set of RDF properties that capture information reflecting the learning preferences of a learner. Includes information on learner preferences, such as home language and country.
· Profile Information. Includes information about the profile itself, such as the date it was generated and the language it is in. All profile information is represented using the RDF mappings of Dublin Core.

THE EUROPEAN INITIATIVE FOR E-LEARNING

The CEN/ISSS Workshop on Learning Technologies (WSLT) has commissioned a survey of current Educational Modelling Languages (EMLs). The Study is lead by The Open University of the Netherlands, with the participation of The British Open University and UNED (Universidad Nacional de Educacion a Distancia, Madrid, Spain).
EMLs are being developed and used around the world. They tend to be based on XML and are used to create

highly-structured course material. An EML-based course might offer features such as: re-useable course material, personalized interaction for individual students, media independence, etc.
WSLT is working on a reparatory agreement that may ultimately lead to an official European standard. Through standardization, it would be possible, for example, for course material to be re-used between standard-compliant platforms from different vendors. Unfortunately, this initiative is still in its early stages and it will possibly take several years for such a standard to be issued.

SCHOOLS INTEROPERABILITY FRAMEWORK (SIF)

The Schools Interoperability Framework (SIF) is an industry initiative to develop a technical blueprint for K-12 software that will enable diverse applications to interact and share data now and in the future. It addresses the variety of data exchange needs in school districts. As the first component, SIF 1.0 focuses primarily on student and transportation data. Future versions of the SIF specification will expand upon the type of data that can be exchanged.
SIF has two deliverables: the SIF Message Specification (for defining the messages that each application can exchange with others) and the Implementation Specification (that defines the software implementation guidelines for SIF). This latter specification does not make any assumption of what hardware and software products need to be used to develop SIF-compliant applications. Instead, it only defines the requirements of architecture, communication, software components, and interfaces between them.
The goal of SIF is to ensure that all SIF-compliant applications can achieve interoperability, regardless of how they are implemented. SIF is truly an open industrial initiative. It is focused on supporting interoperability between (North-American) schools-based educational administration systems whereas initiatives like LIP are focused on learner educational information.

ANSI TS 130 STUDENT EDUCATIONAL RECORD

The ANSI TS130 contains the format and establishes the data contents of a Student Educational Record (Transcript) Transaction Set for use within the context of Electronic Data Interchange (EDI) environment. The student transcript is used by schools and school districts, and by post-secondary educational institutions to transmit current and historical records of educational accomplishment and other significant information about students enrolled at the sending schools and institutions. The transcript may be sent to other educational institutions, to other agencies, or to prospective or current employers. The student transcript contains personal history and identifying information about the student, their current academic status, dates of attendance, courses completed with grades earned, degrees and diplomas awarded, health information (Pre-Kindergarten through Grade 12 only), and testing information.

INTERNET VCARD SPECIFICATION

The vCard specification allows the open exchange of Personal Data Interchange (PDI) information typically found on traditional paper business cards. The specification defines a format for an electronic business card, or vCard. Such a specification is suitable as an interchange format between applications or systems; its format is defined independently of the particular method used to transport it (it supports a file system, point-to-point public switched telephone networks, wired-network transport, or some form of unwired transport). The vCard can be used to forward personal data in an electronic mail message, or for automating the filling out of web-based forms in HTML pages, etc.

INTERNET2 EDUPERSON

The eduPerson specification, created by EDUCASE and Internet2, aims at services that provide seamless access to network-accessible information, regardless of where or how the original information is stored. It achieve this by providing a set of standard higher-education attributes for an enterprise directory, which facilitate inter-institutional access to applications and resources across the higher education community. It is a technologically–driven specification, in that EDUCAUSE/Internet2 eduPerson task force has the mission of defining a Lightweight Directory Access Protocol (LDAP) object class that includes widely-used person attributes in higher education.

HR-XML CONSORTIUM SPECIFICATIONS

The HR-XML Consortium is an independent, non-profit association dedicated to the development and promotion of a standard suite of XML specifications for permitting e-commerce and the automation of human resources-related data exchanges. The mission of the HR-XML Consortium is to develop and publish open data exchange

standards based on XML.
Some of the Consortium's initial targets for standardization activities include recruiting, staffing, compensation and benefits. The Consortium's Recruiting and Staffing workgroup is working on a first version of Staffing Exchange Protocol (SEP), an XML-based messaging framework that supports dynamic, real-time staffing transactions over the Web. Such a protocol will allow for operations like job opportunities postings to job boards, related updating, recall and searches, etc.

Universal Language Framework (ULF)

ULF is a proprietary standard from Saba inc. that complies with IMS standards as regards student modelling. ULF utilizes many of the industry standards for exchanging learning data in a Web environment (including ADL, IMS, LRN, IEEE LTSC, Dublin Core, and vCard) bringing together the key elements of these standards into a comprehensive and fully integrated solution.

Edutella

Peer-to-peer (P2P) networks are a relatively new method of sharing information across distributed, heterogeneous networks, without the requirement of central point of control. The Edutella project [53] was begun in response to the fact that information in P2P networks is no longer organised in navigable hypertext-like structures, but is stored on numerous peers waiting to be queried. We need to know what we want to retrieve and which peer is able to provide the resources in order to submit a query [54]. In some cases, especially concerning the exchange of educational resources, queries are complex, building on standards like IEEE-LOM/IMS metadata with multiple metadata entries.
To complicate matters further, concentrating on domain specific formats has meant that current P2P implementations are fragmenting into niche markets rather than developing unifying mechanisms for future P2P applications. There is, in fact, a danger that unifying interfaces and protocols introduced by the WWW will become lost in the forthcoming P2P arena.
The Edutella project addresses these shortcomings of current P2P applications by building on the W3C metadata standard RDF. The project is a multi-staged effort to scope, specify, architect and implement an RDF-based metadata infrastructure for P2P-networks. To enable interoperability and reusability of educational resources, the infrastructure must be flexible enough to accommodate complex and varying metadata sets, and avoid creating another special purpose application suitable only for a specific application area which is outdated as soon as metadata requirements and definitions change.
The aim is to provide the metadata services required to enable interoperability between heterogeneous JXTA (Juxtapose) applications. The first application will focus on a P2P network for the exchange of educational resources (using schemas like IEEE LOM, IMS, and ADL SCORM to describe course materials).
JXTA is a set of XML-based protocols covering typical P2P functionalities. It provides a Java binding offering a layered approach for creating P2P applications. JXTA enables remote service access and provides additional P2P protocols and services, such as peer discovery, peer groups, peer pipes, and peer monitors.
In a typical P2P-based scenario in which learning resources are being exchanged, universities have two roles:

- Content provider. They do not lose control over their learning resources but still provide them for use within the network.
- Content consumer. Teachers and students benefit from having access not only to a local repository, but also to a whole network, using queries over the metadata distributed within the network to retrieve required resources.

Edutella connects highly heterogeneous peers, each making its metadata information available as a set of RDF statements. By implementing a set of Edutella services, the distributed nature of the individual RDF peers connected to the network can be made completely transparent. Edutella services include a query service, replication, annotation and mapping:

- Query service. The query service is intended to be a standardised query exchange mechanism for RDF metadata stored in distributed RDF repositories and serves as both a query interface for individual RDF repositories located at peers, and as a query interface for distributed queries spanning multiple RDF repositories.
- Replication service. This service complements local storage by replicating data in additional peers to achieve data persistence and availability while maintaining data integrity and consistency. Replication of metadata is the initial focus.
- Mapping, mediation and clustering service. While groups of peers will usually agree on using a

common schema (e.g. SCORM or IMS/LOM for educational resources), extensions or variations may be needed in some locations. The purpose of the Edutella Mapping service is to manage mappings between different schemata and use them to translate queries over one schema to queries over another schema. Mapping services will also provide interoperation between RDF- and XML-based repositories. Mediation services actively mediate access between different services, while clustering services use semantic information to set up semantic routing and semantic clusters.

· Annotation service. In order to be applicable in a wide range of application scenarios, the annotation tool must be independent from any particular domain, and support a wide range of semantic definitions.

3. STANDARDS UNDER EXAMINATION FOR THE DIOGENE PROJECT

The following standards are currently being considered for inclusion in the Diogene Project.

IMS REUSABLE DEFINITION OF COMPETENCY OR EDUCATIONAL OBJECTIVE (RDCEO)

This standard provides a means to create common understandings of competencies that appear as part of a learning or career plan, as learning pre-requisites, or as learning outcomes.

It is available at: http://www.imsglobal.org/competencies/index.cfm

The Reusable Definition of Competency or Educational Objective (RDCEO) specification provides a means to create common understandings of competencies that appear as part of a learning or career plan, as learning pre-requisites, or as learning outcomes. The information model in this specification can be used to exchange these definitions between learning systems, human resource systems, learning content, competency or skills repositories, and other relevant systems. RDCEO provides unique references to descriptions of competencies or objectives for inclusion in other information models.

This specification defines an information model for describing, referencing, and exchanging definitions of competencies, primarily in the context of online and distributed learning. In this specification, the word 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 Definition of Competency or Educational Objective (RDCEO) is an unstructured textual definition of the competency that can be referenced through a globally unique identifier. This information may be refined using a user-defined model of the structure of a competency.

The RDCEO specification provides a means to create common understandings of competencies that appear as part of a learning or career plan, as learning pre-requisites, or as learning outcomes. The information model in this specification can be used to exchange these definitions between learning systems, human resource systems, learning content, competency or skills repositories, and other relevant systems. RDCEO provides unique references to descriptions of competencies or objectives for inclusion in other information models.

The RDCEO that conform to this specification are intended for interchange by machines, but the information they contain is currently intended for human interpretation. This specification does not address the aggregation of smaller competencies into larger competencies and does not address how competencies are to be assessed, certified, recorded, or used as part of a process such as instructional design or knowledge management. It also does not specify how records of competencies associated with an individual are structured, stored, or exchanged.

IMS LEARNER INFORMATION PACKAGE ACCESSIBILITY FOR LIP

Defines two new sub-schemas for the IMS Learning Information Package that define a means to specify accessibility preferences and learner accommodations.

The Accessibility for LIP defines two new sub-schemas for the IMS Learning Information Package that define a means to specify accessibility preferences and learner accommodations. These preferences go beyond support for disabled people to include kinds of accessibility needs such as mobile computing, noisy environments, etc. The <accessForAll> element defines accessibility preferences that were left for future work in the IMS Learner Information Package (LIP) specification version 1.0. The "accessibility" data structure includes the following elements: <language>, <preference>, <eligibility>, and <disability> in the LIP. This specification adds the <accessForAll> element under <accessibility> because it is intended to address the needs of learners beyond those with disabilities. The <disability> element is deprecated henceforth.

It is available at: http://www.imsglobal.org/accessibility/index.cfm

IMS DIGITAL REPOSITORIES

Provides recommendations for the interoperation of metadata indexed collections of resources accessible via a network without prior knowledge of their structure

It is available at: http://www.imsproject.org/digitalrepositories/index.cfm

The diagram in Figure 2.1 below depicts the DRI functional architecture. The interactions are shown by the red (solid) lines. The diagram maps out three entity types that define the space where e-learning, digital repositories, and Information Services interact, and which provide a context for exploration of the problem space.

The three entities are:

- Roles (e.g., Learner, Creator, Infoseeker, Agent)
- Functional Components for Resource Utilizers, Repositories, Access Management, and Procurement Services
- Services, such as Registries and Directories

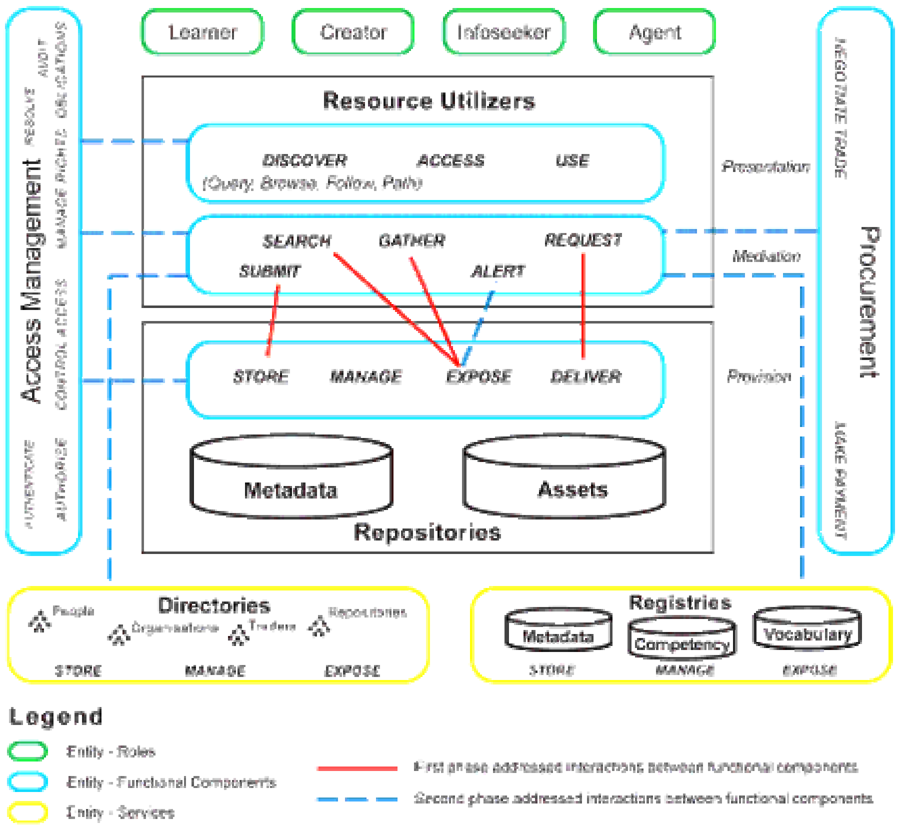


Figure 2.1 The DRI functional architecture

The solid red lines, between a number of functions between Resource Utilizers and Repositories, indicate the interactions between core functional components that support interoperability, including:

- SEARCH, GATHER, (ALERT)/EXPOSE
- REQUEST/DELIVER
- SUBMIT/STORE
- DELIVER /STORE between two repositories

IMS ENTERPRISE SPECIFICATION

Supports interoperability between Learning Management Systems (LMS) and enterprise systems like Human Resource Systems, Student Administration Systems, Training Administration Systems and Library Management Systems

The IMS Enterprise Information Model describes data structures that are used to provide interoperability of Internet-based Instructional Management systems with other Enterprise systems used to support the operations of an organization.

The objective of the IMS Enterprise Information Model is to define a standardized set of structures that can be used to exchange data between different systems. 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 major classes of Enterprise applications supported by this model are Training Administration, Student Administration, Library Management, and Human Resource systems.

Note: The scope of the IMS Enterprise specification is focused on defining interoperability between systems residing within the same enterprise or organization. The documents comprising the IMS Enterprise specification are not targeted at solving the issues of data integrity, communication, overall security, and others that are inherent when investigating cross-enterprise data exchange.

It is available at: http://www.imsglobal.org/enterprise/index.cfm

IMS VOCABULARY DEFINITION EXCHANGE

Defines a grammar for the exchange of simple machine-readable lists of values, or terms, together with information that may aid a human being in understanding the meaning or applicability of the various terms

It is available at: http://www.imsglobal.org/vdex/index.cfm

STANDARD FOR RDF BINDING FOR LEARNING OBJECT METADATA DATA MODEL 1484.12.4

Provides the W3C RDF (Resource Description Framework) binding to enable the exchange of Metadata instances between conforming systems that implement the IEEE 1484.12.1 data model (LOM Standard).

It is available at: http://ltsc.ieee.org/wg12/par1484-12-4.html

STANDARD FOR LEARNING TECHNOLOGY COMPETENCY DEFINITIONS P1484.20

Provides a means to create common understandings of competencies that appear as part of a learning or career plan, as learning pre-requisites, or as learning outcomes. It is a set of recommended practices for Digital Rights Expression Languages Suitable for e-Learning Technologies.

It is available at: http://ltsc.ieee.org/wg20/par1484-20.html

This standard is still in an early stage of development. This standard shall specify the mandatory and optional data elements that constitute a Competency Definition as used in a Learning Management System, or referenced in a Competency Profile. This standard is intended to satisfy the following objectives:

- Provide a standardized data model for reusable Competency Definition records that can be exchanged or reused in one or more compatible systems
- Reconcile various existing and emerging data models into a widely acceptable model
- Provide a standardized way to identify the type and precision of a Competency Definition
- Provide a unique identifier as the means to unambiguously reference are usable Competency Definition regardless of the setting in which this Competency Definition is stored, found, retrieved, or used. For example, metadata that describe learning content may contain a reference to one or more Competency Definition records that describe the learning objectives for the content.
- Provide a standardized data model for additional information about a Competency Definition, such as a title, description, and source, compatible with other emerging learning asset metadata standards
- Provide a controlled vocabulary to express how competency definitions are semantically related.

IMS Simple Sequencing (Part of SCORM 1.3)

Defines a method for representing the intended behavior of a learning experience such that any learning technology system can sequence discrete learning activities in a consistent way. Because of its complexity we report here just few details. The interested reader can see the related documentation available at: http://www.imsproject.org/simplesequencing/.

The following figure describes the sequencing loop devised in the underlying conceptual model.

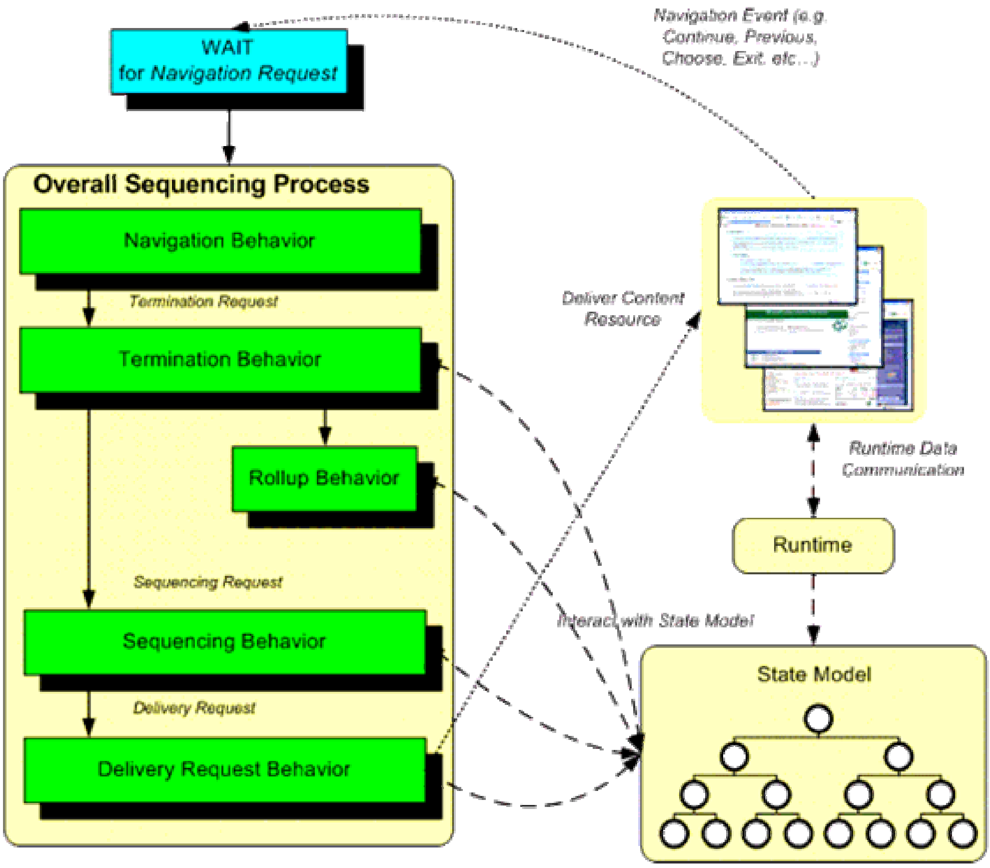


Figure 4.1 - The sequencing loop and its main steps throughout the sequencing process.

Figure 4.1 above shows the various steps in the sequencing process. The control process is shown on the left. In normal operation, the overall sequencing process flows from navigation behavior to termination behavior to sequencing behavior to delivery behavior, followed by a wait while the learner interacts with the content resource.

The right side of Figure 4.1 shows the learner's view of the learning experience. A content resource is delivered to the learner. The learner interacts with the content resource and results may be returned to the tracking model. The learner triggers an event that maps to a navigation request. The navigation request triggers the various steps in the sequencing process.

Throughout all sequencing processes, a collection of state and tracking model data is maintained. Content resources may directly set values in the tracking model through a runtime communications interface to tracking model; this interface is not part of the specification and is not required. All of the other sequencing processes access and update elements of the tracking models.

Changes to the state of an activity occur because of learner interaction with a content resource delivered for the activity or one of its descendents. If an external event affects the tracking, such as an instructor changing a grade, the model assumes processes are invoked as required to update the activity state model.

Changes to the activity state model and tracking model that occur outside of the scope of delivering the learning experience via the sequencing process are outside of the scope of the specification. Systems that implement such side effects and additional capabilities must deal with these conditions accordingly.

The specification describes a collection of sequencing data that comprises the following information:

- **controlMode**
  describes the control that can be exerted over the particular activity (for example whether is a forward-only activity)

- **sequencingRules**
  specifies detailed rules for the following possible rules:

  - precondition

  - exit condition

  - postcondition

- **limitConditions**
  defines the limits for the fruition of the given activity, like for example the attempt limit (for example this activity can be attempted for no more than 3 times)

- **auxiliaryResources**
  specifies auxiliary resources, if any.

- **rollupRules**
  defines the set of rollup rules that are to be applied to an activity. Each activity may have an unlimited number of rollup rules and the order in which they are defined is not significant.

- **objectives**
  defines the set of objectives that are to be associated with an activity. Each activity must have at least one primary objective and may have an unlimited number of objectives.

- **randomizationControls**
  specifies the descriptions of how children of an activity should be ordered during the sequence process. Simple Sequencing processes may reference the randomization control data for any activity in the activity tree and when this is absent the default values should be used.

- **deliveryControls**
  represents the description of the actions and controls used when an activity is delivered. Simple Sequencing processes may reference the delivery control data for any activity in the activity tree. The delivery controls are optional and if these are absent then the default values must be used.

- **#wildcard**
  This is the Simple Sequencing Binding extension facility, for expanding the sequencing definition as needed.

IMS CONTENT PACKAGING SPECIFICATION VERSION 1.1.3 (PART OF SCORM 1.3)

This specification (updated on July 1, 2003) corrects an editorial error that inadvertently left the **`<variation>`** element set in Table 4.1 of the IMS CP Information Model; this element is now removed from the specification.

The related documentation is available at: http://www.imsglobal.org/content/packaging/index.cfm

IEEE DATA MODEL FOR CONTENT OBJECT COMMUNICATION 1484.11.1 (PART OF SCORM 1.3)

This Standard describes a data model to support the interchange of agreed upon data elements and their values between a learning-related content object and a runtime service (RTS) used to support learning management. This Standard does not specify the means of communication between a content object and an RTS nor how any component of a learning environment shall behave in response to receiving data in the form specified. This Standard is based on a related data model defined in the "Computer Managed Instruction (CMI) Guidelines For Interoperability," version 3.4, defined by the Aviation Industry CBT Committee (AICC). To balance the need to support existing implementations with the need to make technical corrections and support emerging practice, this Standard selectively includes those data elements from the CMI specification that are commonly implemented; renames some data elements taken from the CMI specification to clarify their intended meaning; modifies the data types of data elements taken from the CMI specification to reflect ISO standard data types and internationalization requirements; removes some organizational structures used in the CMI specification to group data elements that are specific to the AICC community of practice and not generally applicable; and introduces some data elements not present in the CMI specification to correct known technical defects in data elements taken from that specification.

There is widespread acknowledgement that the data model for content object communication defined in the AICC "Computer Managed Instruction (CMI) Guidelines for Interoperability", version 3.4, has broad applicability to systems used for learning management. The purpose of this standard is to build consensus around, resolve ambiguities, and correct defects in this data model for the data exchanged between learning-related content and a runtime service used to support learning management.

It is available at: http://ltsc.ieee.org/wg11/par1484-11-1.html

IEEE ECMA SCRIPT APPLICATION PROGRAMMING INTERFACE FOR CONTENT TO RUNTIME SERVICES COMMUNICATION (1484.11.2) (PART OF SCORM 1.3)

The ECMAScript API for content-to-runtime-services communication defined in the AICC "CMI Guidelines for Interoperability" version 3.4, has broad applicability to systems used for learning management. The purpose of this Standard is to build consensus around, resolve ambiguities, and correct defects in this ECMAScript API for exchanging data between learning-related content and a runtime service used to support learning management.

The documentation is available at: http://ltsc.ieee.org/wg11/par1484-11-2.html

DIGITAL RIGHTS EXPRESSION LANGUAGE (DREL) P1484.4

This specification focuses on the study of existing specifications that may be considered to be rights expression languages, more specifically the DREL that are most general and that are getting the most attention in industries related to learning, education and training.
Some examples of such DREL's exist in:

- ODRL – Open Digital Rights Language
- MPEG REL – Extensible Rights Markup Language
- OeB – Open E-book
- IEEE Std 1420.1b-1999 – IEEE Standard for Information Technology – Software Reuse, IP Rights Framework

In addition, the DREL subcommittee is surveying a number of the initiatives that are developing or proposing standards for digital rights management. This can be used as a good source of technical documentation and references to the LTSC-DREL WG. The initiatives chosen here are either globally significant or significant within the learning technology community:

- MPEG-21 (Motion Picture Experts Group – 21)
- OASIS
- CEN/ISSS Workshop on Learning Technology
- IMS Global Learning Consortium
- The Open Knowledge Initiative
- Indecs2rdd Rights Description Dictionary
- ONIX (Online Information Exchange) Metadata Specification
- OMA (Open Mobile Alliance) Rights Expression Language

This activity is still in an early stage of development and cannot be considered stable enough to be included into the Diogene project.

The documentation is available at: http://ltsc.ieee.org/wg4/par1484-4.html

4. CONCLUSIONS

In this document we explored the main currently available e-learning standards, covering also knowledge representation format of interest to e-learning.

Our aim was to provide a document focused on monitoring the most important e-learning standards, to describe the tools available for each of them and providing a critical discussion about the pros and cons of each standard.

In the following section each standard considered for the inclusion in the Diogene project will be briefly reviewed and a trade-off assessment of its adoption in the project will be discussed.

CRITICAL DISCUSSION

IMS CONTENT PACKAGING SPECIFICATION VERSION 1.1.3 (PART OF SCORM 1.3)

The inclusion of the new modifications to this standard in the Diogene Project should be adopted as a major priority given also the little impact on the already-compliant facilities and data structures featured in the Diogene project.

IEEE DATA MODEL FOR CONTENT OBJECT COMMUNICATION 1484.11.1 (PART OF SCORM 1.3)

This activity is still in an early stage of development and cannot be considered stable enough to be included into the Diogene project. The most recent version as of December 2003 (ballot draft 2) is available at: http://ltsc.ieee.org/wg11/files/IEEE_1484.11.1-ballot-D2.zip

IMS DIGITAL REPOSITORIES

This standard could be considered for inclusion in the Diogene project, providing that:

- it is suited for digital content sharing and interoperability and
- it relies upon other standards (SCORM, etc.) so that its adoption should come at a reltively low cost for the project.

Anyway, given some considerations (it is not a major focus for the Diogene project and it can be safely adopted

in a later stage of development of the project) it is advised to postpone its adoption.
Hence, in a later stage of development of the Diogene project would become more clear the feasibility and related trade-offs of the adoption of this specification.

Digital Rights Expression Language (DREL) P1484.4

This activity is still in an early stage of development and cannot be considered stable enough to be included into the Diogene project.

IEEE ECMA Script Application Programming Interface for Content to Runtime Services Communication (1484.11.2) (Part of SCORM 1.3)

This specification is meaningful to be implemented in the Diogene project only if the related standards (those defined in the AICC "CMI Guidelines for Interoperability" version 3.4) are going to be included as well. At present it can be postponed for more urgent specification inclusions.

IMS Enterprise Specification

The inclusion of this standard in the Diogene Project is can be relatively expensive given its purpose and the absence in Diogene of data structures and services that models this information.

IMS Learner Information Package Accessibility for LIP

The inclusion of this standard in the Diogene Project is to be considered as an auxiliary feature that can be safely postponed for giving priority to more important specification inclusions.

OWL Lite

The inclusion of this standard in the Diogene Project is to be considered as a primary need for the project itself and should be considered as one of the most important priorities in the process of standards updating.
To recap, DAML+OIL evolved and its evolution is OWL. DAML+OIL specs were in fact used by W3C as the starting point to define OWL.

In particular, the adoption of OWL Lite can be considered strategic for a number of reasons. First of all, it is a relatively simple and easy to implement specification that is much more up-to-date than the knowledge representation standard already adopted in Diogene. Furthermore, it can be the basis for the adoption of more articulated and powerful versions of OWL like OWL-S (see the related section in the previous chapter).

Furthermore, some considerations are needed: first of all, is important to point out that SHOE is no longer supported and that work at the University of Maryland on Web Ontologies is continuing using OWL and DAML+OIL. Secondly, in August 2003 the OWL specifications became a W3C Candidate Recommendation. And, lastly, a Protégé plug-in supporting OWL already exists and is commonly used in the community.

Concluding, the adoption of the OWL Lite specification within the Diogene project should be considered as one of the main updating specification issues.

DAML-S and OWL-S

These two specifications cover in a greater detail/expressiveness what is covered by the OWL Lite specification discussed before. While their inclusion in the Diogene project would be useful, they can be safely postponed until the adoption of OWL Lite (to lower the adoption risks and handle more easily the needed development effort). In this way, after the successful adoption of OWL Lite within the project, a successive assessment study will evaluate the feasibility (trade-offs) of adopting one or both these two specification for knowledge representation.

Standard for Learning Technology Competency Definitions P1484.20

This standard is still in an early stage of development and major revisions are expected. Hence its adoption in the Diogene project is discouraged.

Standard for RDF Binding for Learning Object Metadata data model 1484.12.4

This inclusion should not come at a low cost to the Diogene project because of the need of creating ad-hoc XML parsers facilities for RDF that are rather different from the current implementation in Diogene. Hence, to avoid a costly development effort it is suggested to avoid the adoption of this specification.

IMS Reusable Definition of Competency or Educational Objective (RDCEO)

Given that the IEEE/LTSC has requested and received permission to use the RDCEO specification as the basis for an IEEE competency definition standard this specification becomes an important standard to be considered for inclusion in the Diogene project, even if not ready at this moment (revisions are expected for the inclusion of this specification in the IEEE/LTSC competency definition standard).

IMS Simple Sequencing (Part of SCORM 1.3)

The inclusion of this standard in the Diogene Project is controversial.
In fact, Diogene has already a module (called LIA) functionally-equivalent to the many of features provided in the simple sequencing specification.
Anyway, being the module designed and developed before the simple sequencing specification went public, the two subsystems are uncorrelated in both the knowledge representation and LOM manipulation. This can be a serious, conceptual problem because Diogene model its sequencing behavior after the LIA module. Changing this to be Simple Sequencing-compliant could denature the whole Diogene project.

Regarding its adoption costs, with a rough approximation should take at least *eight months / man* even if this is a somehow rough estimate in that many factors can concur to enlarge this effort estimation.

To conclude, the adoption of the IMS Simple Sequencing standard is a controversial choice that should be taken considering the many aspects involved. A conservative, prudent choice would be to postpone the adoption of this standard as long as no clearer information on the estimation effort and overall risk for the Diogene project has been taken.

IMS Vocabulary Definition Exchange

At present this standard is in an early public draft and as this, it is unfeasible for inclusion in Diogene. It is highly advisable to wait until a more stable version of the standard will be made available.