



IMS Application Profile Guidelines Overview

Part 1 – Management Overview Version 1.0

Date Issued: 10 October 2005

Latest version: <http://www.imsglobal.org/ap/>

IPR and Distribution Notices

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

IMS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on IMS's procedures with respect to rights in IMS specifications can be found at the IMS Intellectual Property Rights web page: http://www.imsglobal.org/ipr/imsipr_policyFinal.pdf.

Copyright © 2005 IMS Global Learning Consortium. All Rights Reserved.

If you wish to copy or distribute this document, you must complete a valid Registered User license registration with IMS and receive an email from IMS granting the license to distribute the specification. To register, follow the instructions on the IMS website: <http://www.imsglobal.org/specificationdownload.cfm>.

This document may be copied and furnished to others by Registered Users who have registered on the IMS website provided that the above copyright notice and this paragraph are included on all such copies. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to IMS, except as needed for the purpose of developing IMS specifications, under the auspices of a chartered IMS project group.

Use of this specification to develop products or services is governed by the license with IMS found on the IMS website: <http://www.imsglobal.org/license.html>.

The limited permissions granted above are perpetual and will not be revoked by IMS or its successors or assigns.

THIS SPECIFICATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE CONSORTIUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS SPECIFICATION.

Copyright © 2005 by IMS Global Learning Consortium, Inc.
All Rights Reserved.

The IMS Logo is a registered trademark of IMS Global Learning Consortium, Inc.

Document Name: IMS Application Profile Guidelines Overview

Date: 10 October 2005

Executive Summary

This document is the first of two parts which together, constitute the IMS Application Profile Guidelines:

- Part 1 – Management Overview
- Part 2 – Technical Manual

This Management Overview describes what an application profile is in the context of the IMS specifications and the benefits to be gained from undertaking such an exercise – namely more closely meeting the needs of the target user community whilst harnessing the specifications to aid integration and enhance interoperability between tools, products and services which vendors would supply to that community. Guidance is offered on the key factors for deciding whether or not to embark upon a profiling exercise and a process outlined for how to proceed with such an activity. Conformance issues around an application profile are briefly discussed, as are technology and implementation issues beyond the scope covered by the specifications.

The document is offered as a guideline, based upon the experience of a number of user communities in adopting and implementing the specifications, in the hope that their experience will be useful to others facing the same issues which they have had to work through with their users and suppliers. Nothing in this document is mandatory – ultimately the choices are made by implementers and the users of their offerings. However, the document does capture a viable process for helping vendors more closely meet the needs of a community, without necessarily breaking broader interoperability and maximizing the use of implementations against one or more base specifications.

The term Application Profile is in common usage in the meta-data community and generally refers to the adaptation, constraint, and/or augmentation of a meta-data scheme to suit the needs of a particular community. The intention being to define a profile of a specific schema for its application to a particular community. For the purposes of this discussion, the source schema is the original schema being profiled, whilst the derived schema is the output of the profiling activity. This process may include one or more of the following actions to generate a derived schema:

- Selection of a core sub-set of elements and fields from the source schema (expressed perhaps as a mandated sub-set which must be supported as a minimum);
- The addition of elements and/or fields (normally termed extensions) in a prescribed manner, to the source schema, thus generating the derived schema;
- Substitution of a vocabulary with a new, or extended vocabulary to reflect terms in common usage within the target community, e.g., a bounded set of competencies or a curriculum model;
- A comprehensive description of the semantics and common usage of the schema and constituent terms as they are to be applied across the community.

Application Profiling can be summarized as:

- a) Localization – the specialization of one or more conceptual data schemas (source schemas) to the precise needs of a community, generating a derived schema;
- b) Representation – mapping the localized conceptual schema(s) to a generic binding for interchange;
- c) Transaction – define how the abstract interface and service model, i.e., the APIs, and implied/stated transactions are to be realized utilizing a concrete platform technology.

The effort involved in developing an Application Profile can be significant and the task itself is likely to necessitate repeated consultation with the target community if it is to accurately meet their needs. The target community could be a single organization, e.g., a global corporation, a commercial training trade association, or in an educational setting, it could equally be an academic institution, a funding body, a regulatory agency, or a government ministry.

Table of Contents

EXECUTIVE SUMMARY	2
1. INTRODUCTION	4
1.1 CONTEXT AND SCOPE	4
1.2 DEFINITIONS	4
1.3 REFERENCES	7
2. PRINCIPLES OF APPLICATION PROFILING	8
2.1 WHAT IS APPLICATION PROFILING?	8
2.2 APPLICATION PROFILING IN THE CONTEXT OF LEARNING TECHNOLOGY SPECS	8
2.2.1 <i>The Role of IMS Specifications</i>	8
2.2.2 <i>Lessons Learned from Adoption</i>	9
2.2.3 <i>Benefits of a Consistent Approach to Application Profiling</i>	10
2.2.4 <i>Profiling an IMS Specification</i>	11
3. THINGS TO CONSIDER BEFORE DEVELOPING A PROFILE	13
3.1 REASONS FOR NOT CREATING AN APPLICATION PROFILE	13
3.2 WHEN TO CREATE AN APPLICATION PROFILE	13
4. OUTLINE OF A PROCESS FOR CREATING AN APPLICATION PROFILE	15
4.1 FEASIBILITY AND RISK ANALYSIS	15
4.2 CAPTURING THE REQUIREMENTS	15
4.3 AGREEING THE DECISION RULES	16
4.3.1 <i>Mandatory and Optional Decision Rules</i>	16
4.3.2 <i>Further Conflict Resolver</i>	17
4.4 PROJECT GROUP GUIDELINES	17
5. CONFORMANCE ISSUES FOR APPLICATION PROFILES	19
5.1 APPLICATION PROFILE CONFORMANCE WITH THE BASE SPECIFICATION(S)	19
5.2 CONFORMANCE OF IMPLEMENTATIONS TO THE APPLICATION PROFILE	19
ABOUT THIS DOCUMENT	21
LIST OF CONTRIBUTORS	21
REVISION HISTORY	22
INDEX	23

1. Introduction

1.1 Context and Scope

This document is the first of two parts which together, constitute the IMS Application Profile Guidelines:

- Part 1 – Management Overview
- Part 2 – Technical Manual

This Management Overview describes what an application profile is in the context of the IMS specifications and the benefits to be gained from undertaking such an exercise – namely more closely meeting the needs of the target user community whilst harnessing the specifications to aid integration and enhance interoperability between tools, products, and services which vendors would supply to that community. Guidance is offered on the key factors for deciding whether or not to embark upon a profiling exercise and a process outlined for how to proceed with such an activity. Conformance issues around an application profile are briefly discussed, as are technology and implementation issues beyond the scope covered by the specifications.

The document is offered as a guideline, based upon the experience of a number of user communities in adopting and implementing the specifications, in the hope that their experience will be useful to others facing the same issues which they have had to work through with their users and suppliers. Nothing in this document is mandatory – ultimately the choices are made by implementers and the users of their offerings. However, the document does capture a viable process for helping vendors more closely meet the needs of a community, without necessarily breaking broader interoperability and maximizing the use of implementations against one or more base specifications.

Section 2.2.2 ‘Lessons Learned from Adoption’ in particular, highlights the fact that adoption of the specifications often entails a selection of the specifications to adopt, some changes to the information model for these specifications and some adaptation (e.g., language, vocabularies) to serve a particular community. The available documentation for these profiles is highly variable and rarely captures the process by which they were derived. The Application Profile Guidelines whitepaper makes explicit such a process in this Management Overview and offers further guidance in the Technical Manual on how an application profile should be developed and documented.

Having opted to create an application profile in the manner prescribed, achieving interoperability across implementations of that profile is still dependent upon a number of independent, ongoing factors, not least:

- *Consistent interpretation by implementers of the application profile;*
- *Consistent use of vocabularies by the information sources;*
- *Consistent use of the information by users of the information.*

1.2 Definitions

Acceptance Test Criteria	Criteria (e.g., user requirements) guiding the final testing of a system (generally in its operational environment) to enable the customer to determine whether it can be accepted.
ADL	Advance Distributed Learning Programme
AICC	Aviation Industry CBT Committee
ALIC	Advanced Learning Infrastructure Consortium
API*	Application Program Interface. An application program interface is an implementation of a <i>Service Access Point</i> (SAP) or collection of SAPs. A set of standard software interrupts, calls, functions, and data formats that can be used by an application program to access network services, devices, or operating systems.
Application Profile	A description of the use of a single technical specification to meet the needs of a particular community.
BSI IST/43	UK Learning Technology Standardization group.

CEN/ISSS LT Workshop	Centre for European Normalization, Information Society Standardization Service Learning Technology Workshop.
CWA	CEN Workshop Agreement
Certification*	Certification is the process undertaken to determine whether or not an implementation of an IMS specification conforms to that specification as stated by the associated conformance statement.
Conformance*	This is the statement of the properties that an implementation of a specification must possess in order to be defined as providing the functionality defined within the specification. The implementation may provide other functionality beyond the scope of the defined conformance.
Conformance Testing	Testing to evaluate the adherence or non-adherence of an implementation to a specification.
Content Packaging*	A unit of usable (and reusable) content as defined within the <i>IMS Content Package Specification</i> . An IMS Content Package consists of a logical description of the package (the <i>Manifest</i>) and the physical resources.
Content Re-Engineering Tool	Tool to modify content resources or their logical descriptions.
DOM	The Document Object Model is a platform and language-neutral interface that allow programs and scripts to dynamically access and update the content, structure and style of documents.
Domain Profile*	Customizing parts of one or more standards and/or specifications to meet the needs of a particular market or community i.e. a domain. A set of one or more base standards and/or specifications, and where applicable the identification of chosen classes, subsets, options, vocabularies and parameters of those standards/specifications necessary for accomplishing a particular function. In this context, the SCORM is a Domain Profile. In general a Domain Profile will not consist solely of IMS specifications.
EIFEL	European Institute for e-Learning
ELIG	European e-Learning Industry Group
European SchoolNet	Membership-based consortium of the ministries of education of many of the European member-state and Eastern European countries.
HTTP	Hyper-Text Transfer Protocol. An Internet protocol i.e. a part of the Internet Protocol Suite, which defines message format and transmission for media objects in a TCP/IP network. HTTP is typically used to transmit HTML documents between a web server and a web client e.g. a browser.
ICP	International Conformance Program
IEEE LTSC	Learning Technology Standardization Committee of the IEEE (see <i>IMS Abstract Framework Glossary</i> for a more complete definition).
IMS	IMS Global Learning Consortium
ISO/IEC JTC1/SC36*	Learning Technology Committee to Joint Technical Committee 1 (JTC 1) - The International Organization for Standardization and the International Electro technical Commission has formed a Joint Technical Committee (JTC1) that is focused on the area of Information Technology standardization. ISO/IEC JTC1/SC36 (Sub Committee 36) is intended to address standardization in the area of information technologies that support learning, education and training.

Learning Technology Specification	<p>A number of these (by way of example) are available for download at no charge from the IMS Global Learning Consortium website at http://www.imsglobal.org Each Learning Technology Specification is generally comprised of three documents:</p> <ul style="list-style-type: none"> • Information Model – covering some semantics, conceptual schema and data elements and the requirements expressed as UML use cases; • Binding Document – offering an explicit XML binding for the Information Mode; • Best Practice Guide – offering examples of implementations, how to create valid extensions and general guidance on implementing tools/applications which exploit the Learning Technology Specification.
LIP	Learner Information Package
LOM	Learning Object Metadata
MIT	Massachusetts Institute of Technology
Model-Based Testing	An approach to software testing that bases common testing tasks such as test case generation and test result evaluation on a model of the application under test.
OKI*	Open Knowledge Initiative. OKI is defining a service-based architecture, consisting of service and Application Programming Interface (API) specifications, designed to support educational software, e-learning applications, and learning management systems. OKI also provides support services to its developer and architectural specification communities, though on-line forums, documentation, training, and community events. OKI is led by the Massachusetts Institute of Technology.
OMG	Object Management Group
ROI	Return On Investment
SCORM*	Sharable Content Object Reference Model (see <i>IMS Abstract Framework Glossary</i> for a more complete definition).
SIF*	Schools Interoperability Framework (see <i>IMS Abstract Framework Glossary</i> for a more complete definition).
SOAP*	Simple Object Access Protocol. SOAP provides the definition of an XML document which can be used for exchanging structured and typed information between peers in a decentralized, distributed environment.
Stub	A dummy or skeletal implementation of a piece of code temporarily used to develop or test another piece of code that depends on it.
Test Suite	Software tools for testing the degree to which software or hardware conform to the requirements of a standard. Used in software development to assure quality on completion and post completion to demonstrate conformance and achieve certification for customer purposes.
Test System	The combination of test software, test documentation, and test procedures that check an implementation for conformance to a standard.
UI*	User Interface. The visual presentation and its underlying software through which a user interacts with an application.
UML	Unified Modeling Language. A language proposed by the OMG for specifying, visualizing, constructing and documenting the artifacts of a software system as well as for business modeling; it is the de-facto standard diagramming notation for object-oriented modeling.
VP	Vice-President
WSDL*	Web Services Description Language (see <i>IMS Abstract Framework Glossary</i> for a more complete definition).

XMI	XML Metadata Interchange. Codification to enable easy interchange of metadata between modeling tools and repositories in distributed heterogeneous environments, for sharing object models and other metadata over the Internet.
XML*	Extensible Mark-up Language. XML is a flexible way to create common information formats and share both the format and the data on the World Wide Web, intranets, and elsewhere.

* The entries denoted by '*' are taken from the IMS Abstract Framework Glossary [IAF, 03].

1.3 References

- [GWS, 05a] *General Web Services Base Profiles Public Draft v1.0*, C.Schroeder, S.Raju and C.Smythe, [IMS/GLC](#), January 2005.
- [GWS, 05b] *IMS General Web Services UML to XML Binding Auto-generation Public Draft v1.0*, C.Schroeder, S.Raju and C.Smythe, [IMS/GLC](#), January 2005.
- [IAF, 03] *IMS Abstract Framework: Glossary v1.0*, C.Smythe, [IMS/GLC](#), July 2003.

2. Principles of Application Profiling

2.1 What is Application Profiling?

Within the IMS context, Application Profiling is the tailoring of specification (by amending the binding of the specification) to suit the needs for its application to a particular community.¹ For the purposes of this discussion, the source schema is the original schema being profiled, whilst the derived schema is the output of the profiling activity. This process may include one or more of the following actions to generate a derived schema:

- Selection of a core sub-set of elements and fields from the source schema (expressed perhaps as a mandated sub-set which must be supported as a minimum);
- The addition of elements and/or fields (normally termed extensions) in a prescribed manner, to the source schema, thus generating the derived schema;
- Substitution of a vocabulary with a new, or extended vocabulary to reflect terms in common usage within the target community, e.g., a bounded set of competencies or a curriculum model;
- A comprehensive description of the semantics and common usage of the schema and constituent terms as they are to be applied across the community.

Thus, Application Profiling can be summarized as:

- a) Localization - the specialization of one or more conceptual data schemas (source schemas) to the precise needs of a community, generating a derived schema;
- b) Representation - mapping the localized conceptual schema(s) to a generic binding for interchange;
- c) Transaction - define how the abstract interface and service model, i.e., the APIs, and implied/stated transactions are to be realized utilizing a concrete platform technology.

Some key reasons for developing an Application Profile include:

- a) To meet technical and other requirements and preferences specific to a project, community, domain, or region;
- b) To address ambiguity and generality in a specification or standard;
- c) To foster semantic interoperability, e.g., through the use of commonly understood vocabularies;
- d) To facilitate testing for conformance and successful interoperability.

2.2 Application Profiling in the Context of Learning Technology Specs

2.2.1 The Role of IMS Specifications

Each of the present set of IMS learning technology specifications has been driven by requirements (normally expressed as use cases) from a cross-section of potential users of the target specification. A user of a specification in this sense encompasses:

- Vendors of e-Learning platforms, tools, and services wishing to address a need amongst their existing and prospective customers;
- Third-party suppliers of ancillary or related services, e.g. hardware, network services etc.;
- Institutions representing adopting communities;

1. The term Application Profile is in common usage in the meta-data community. Since such meta-data is ordinarily stored in a database and only accessed by a dedicated application, the form of its internal representation is not normally mandated. However, a specific and commonly understood syntax is obviously required for data interchange and commonly adhered-to protocols imposed if such interchange is to be conducted as run-time transactions. Further information on how the meta-data world uses the phrase Application Profile can be obtained from Dublin Core Metadata Initiative. IMS has extended the definition and usage of the phrase Application Profile to be specification and application independent.

- Researchers wishing to harness specifications in applied research of next generation tools and services.

The intention has been and remains, to ensure that specifications going through the IMS process are well grounded in established practice and are sufficiently general to meet the needs of a number of distinct users rather than a special case.

It is now common practice for the requirements for a given specification to consist of a set of Use Cases, expressed in UML. These Use Cases anchor the specification against the precise requirements of the developers of the specification and the user communities with which they have consulted. The Use Cases themselves form a core part of the specification. In fact, the Use Cases expressed in a specification will often be a synthesis of a broader Use Case Portfolio and appropriately scope the specification to meet the common ground across a large cross-section of user needs.

Following the IMS General Web Services approach ([GWS, 05a], [GWS, 05b]) runtime, in the form of a behavior model, is now being made explicit through the inclusion of a Service Model in the given specification. The use of a Service Model allows the behavior to be expressed while still permitting selection from a variety of bindings at the implementation phase. There are, and will continue to be for the foreseeable future, a (small) number of alternative technology bindings, reflecting popular development and runtime platforms in the market. By necessity, a specification has to be neutral with respect to these alternatives or else it effectively cuts off adopters reliant upon platforms not covered explicitly in the specification.

2.2.2 Lessons Learned from Adoption

Experience of increasing adoption of the specifications across both vertical domains, i.e., K-12, vocational training, higher education, corporate training, basic skills for life, and geographical regions, has made evident a recurring process of adaptation of the specifications to meet the specific needs of each community. There are now a number of examples of communities for whom this process has been undertaken; see Table 2.1.

Agencies undertaking this process clearly have a well identified community in mind and have researched the precise needs of that community, in order to both select the sub-set of the specifications required and propose the necessary changes and extensions to meet their needs.

This emerging model for the adoption process is encouraging as it would seem to confirm that the IMS specifications have indeed been kept sufficiently general for them to have broad-based appeal and offer utility across communities. An Application Profile on the other hand, clearly enhances the utility of a specification to a community and, if adhered to, promises greater interoperability between members of that community.

Table 2.1 Application Profiles of the IMS specifications.

Profile Name	Owner	Sector/Region	IMS Specifications Included
ALIC	Advanced Learning Infrastructure Consortium	Japanese training	Meta-data Content Packaging QTI
CanCore	Industry Canada	Canadian Higher Education	Meta-data
CELEBRATE	CELEBRATE Consortium	European Schools	Meta-data
European Diploma Supplement	CEDEFOP	European HE	Learner Information Package
Guidelines for the production of learner information standards and specifications	CEN/ISSS	European Education	Learner Information Package

Profile Name	Owner	Sector/Region	IMS Specifications Included
Normetic	Crepuq	Canadian Higher Education	Meta-data
SingCore	eLearning Competency Centre	Singapore education and training	Meta-data QTI Content Packaging
SCORM	ADL	Primarily Corporate and Governmental Training, but also used in other sectors.	Meta-data Content Packaging Simple Sequencing
UK eGIF	Office of the eEnvoy, UK	British education across schools, FE, HE and Lifelong Learning (e.g. Ufi)	Meta-data Content Packaging Learner Information Package
UK LeAP	BSI	British FE/HE	Learner Information Package

2.2.3 Benefits of a Consistent Approach to Application Profiling

Experience to-date has identified real benefits to be gained from closer collaboration across communities in developing these profiles, particularly in agreeing basic rules to be followed, and adopting a consistent format for documenting each Application Profile.

- Agreeing a consistent set of rules for constructing a profile will bound the changes that can be made thus ensuring greater interoperability across conformant Application Profiles;
- Providing consistent documentation of Application Profiles will enable vendors to more easily build products and services that span multiple communities with simple configuration settings for localization;
- The growing number of publicly documented Application Profiles will allow subsequent adopting communities to select and reuse elements of existing profiles, rather than develop from first principles;
- Ultimately, providing strongly typed, machine readable definitions of these Application Profiles will enable runtime context negotiation between domains to facilitate data exchange and interoperability across communities.

2.2.4 Profiling an IMS Specification

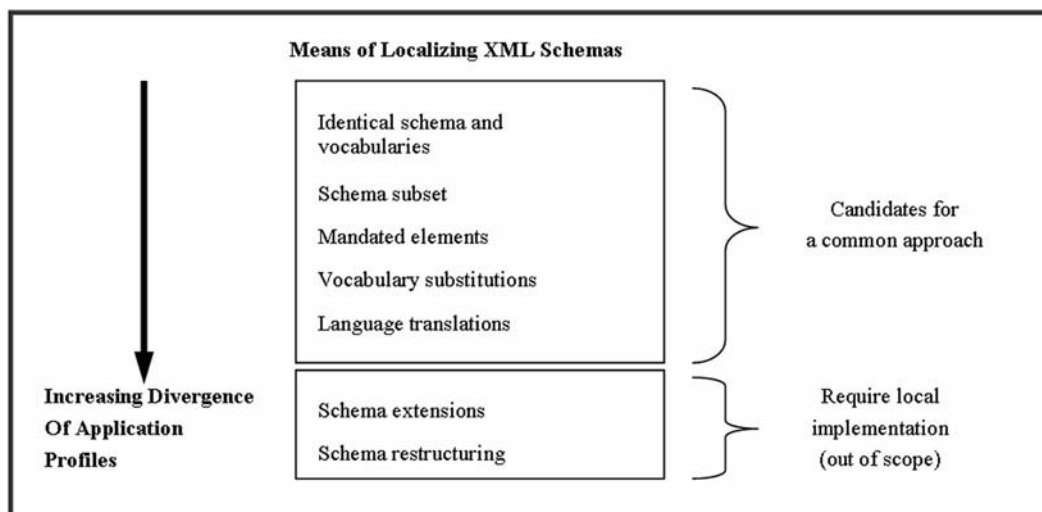


Figure 2.1 Application Profile Schema Localization.

Data coverage of the specification is normally presented as a Data Model (often expressed as an XML schema, but increasingly as a set of object classes within a UML description). Figure 2.1 indicates the scope for user communities to generate localized Application Profiles which can derive the benefits of a common approach. Figure 2.2 depicts the transition from Use Cases through Specification to Application Profile.

Application Profiles, like IMS specifications, may contain information models (abstract information structures not bound to specific technologies), vocabulary definitions, and technology bindings (to XML Schema for example). An Application Profile can also contain more detailed information, such as policies, procedures, and quality assurance practices. This is because Application Profiles can be created for a wide range of purposes beyond just technical interoperability, such as ensuring that there is consistent usage of terminology, or that the appropriate amount of detail is provided to describe resources, or to ensure that particular methods are used to arrive at the final meta-data.

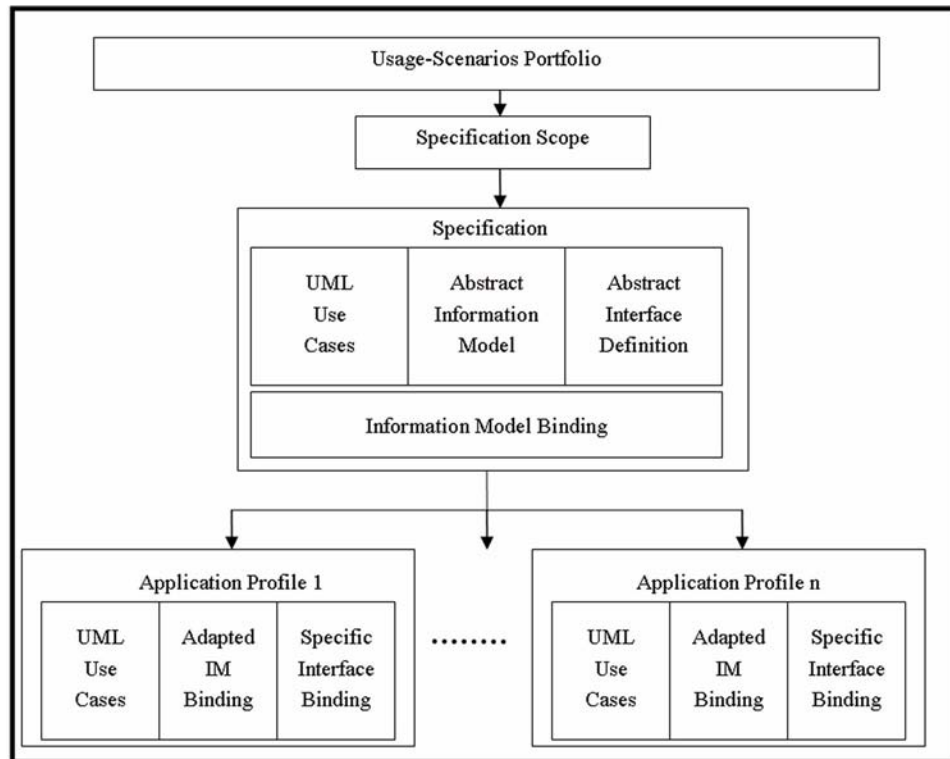


Figure 2.2 Deriving Application Profiles from a Specification.

3. Things to Consider Before Developing a Profile

The effort involved in developing an Application Profile can be significant and the task itself is likely to necessitate repeated consultation with the target community if it is to accurately meet their needs. The target community could be a single organization, e.g., a global corporation, a commercial training trade association, or in an educational setting, it could equally be an academic institution, a funding body, a regulatory agency, or a government ministry.

Before starting on the development of an application profile, it is advisable to undertake a cost/benefit analysis to answer the question “Do the anticipated benefits of the application profile to the community outweigh its likely development, implementation, and maintenance costs?” Some factors to consider in reaching a decision might include:

- Is the target community clearly identified, and if it is, what is its size?
- Which specifications are the community adopting and are their specific, additional needs well understood, or can they at least be gathered?
- What is the likely timeline and effort involved?
- Is the community a viable market for solution providers to target?
- Might alternative strategies meet the need (e.g., changing processes or adopting another existing profile)?

3.1 Reasons for Not Creating an Application Profile

Applying Occam’s Razor to Application Profiles:

“DO NOT CREATE APPLICATION PROFILES BEYOND NECESSITY”

Use an existing standard, specification, or application profile whenever possible.

Application Profiles can be:

- Expensive to create;
- Expensive to maintain;
- Expensive to implement;
- Expensive to test and assure interoperability;
- They can reduce interoperability with those outside the community.

The results of a full lifecycle cost-benefit analysis of an application profile might show for example that it would be more cost effective to change the current way things are done and the information needed than to create an application profile. We like our differences, but they need to be useful and significant differences, not arbitrary ones.

In general, it is better to avoid creating non-interoperable application profiles (see later) for ‘learning content’, very broadly defined (including learning activities and designs, metadata, assessments, etc., as well as traditional content), where:

- a) Content within the community may need to be used by others outside it;
- b) Content created outside the community may need to be assimilated.

The benefits of content related standards increase with the scale of their use and reuse (offsetting costs, allowing higher quality, etc.). Application profiles, when inappropriately used, limit the scalability and hence the benefits of open specifications and standards.

3.2 When to Create an Application Profile

Non-interoperable application profiles are more appropriate for closed communities with specific purposes, processes, and community specific information to exchange. So, the following are some conditions under which it is appropriate to create an application profile:

- There is a clear interoperability need that cannot be met by any existing specification;
- The alternative would be to create a completely new specification;
- An existing specification meets a significant part of the need and can act as the base specification from which the proposed Application Profile can be derived;
- There is a need to maintain complete, or some degree of, interoperability with the base specification;
- There are existing systems that already implement the base specification and therefore provide some or all of what is needed and can be easily adapted.

4. Outline of a Process for Creating an Application Profile

It is recommended that the following issues are considered in preparing to develop an application profile:

- Identify the community of use and its unmet interoperability needs;
- Closely examine existing specifications, standards, and/or application profiles that might already meet these needs;
- Determine if there is sufficient vendor/user expertise to define the application profile and likely market demand to justify the effort;
- Agree the rules by which decisions will be reached in the case of conflicting proposals on the implementation of the application profile;
- Set out a workplan and recruit the team of vendor/user experts to develop the application profile.

4.1 Feasibility and Risk Analysis

Before entering into the detailed implementation, it is worthwhile considering the following:

- a) From the community identify:
 - The domain experts and user representatives who can articulate the requirements;
 - The system suppliers to the community who will have to implement the profile.
- b) Determine the size of the community market or confirm that the available funds are able to support the cost of the proposed profile;
- c) Determine that there are enough suppliers willing to implement the proposed application profile.

It should then be evident whether there are the skilled individuals available who would be able to define the application profile and what the chances are of successful adoption of the profile by the target community.

4.2 Capturing the Requirements

Having decided to proceed with specifying an application profile, the task is one of identifying variability points, where the community's needs and/or context differ from those assumed by the base specification. The following preliminary steps can help identify scope and area of focus for the application profile:

- Identify the community purpose that needs to be facilitated;
- Identify the larger context in which this purpose plays;
- Identify the actors, both humans and systems;
- Identify the data entities and their relationships;
- Identify the process(es) to be supported;
- Identify the supporting information systems (current and planned) that need to be connected;
- From the actors, entities, processes, and participating information systems, create a 'domain model' or 'context model';
- Identify the information to be exchanged;
- Identify the events/triggers for these exchanges;
- Identify what needs to happen in response to these events/triggers;
- Identify the existing specification, standard and/or application profile that most closely meets these needs and matches the domain model. This will be the base from which the application profile will be derived;
- Identify the changes that need to be made to the selected specification to meet the community-unique needs.

4.3 Agreeing the Decision Rules

Before starting on the main task it is generally helpful to establish a set of ‘decision rules’ and have all participants agree to them. Agreements on the application profile should, as far as possible, be decided by consensus, but where there is no consensus, agreed decision rules help to resolve any deadlocks rapidly before they cause breakdown. Where this is not possible a ‘decision rule’ needs to be agreed at the outset to break any impasse. Three very simple examples of a decision rule would be:

- 1) **Mandatory rule:** Agree to adopt all mandatory fields as a minimum;
- 2) **Restrictive rule:** Don’t add fields beyond necessity – or without general agreement.
“When in doubt, leave it out”;
- 3) **Inclusive rule:** The aim is to enable people to meet their needs, so if a significant minority needs something, then include it after taking into account all cost and interoperability factors.

But these examples may be too simple, so an alternative decision rule might be:

- When it comes to deciding what information or action is required, then the community representatives’ vote decides. If these are evenly split, then the suppliers’ vote decides;
- When it comes to deciding how the information or action is to be represented, the suppliers’ vote decides, and if split, a community vote decides.

More refined decision rules can be formulated to meet the composition of the team, but the main point is that good decision rules help a team make progress.

Hopefully you will not need to call on your decision rules but they should be there and made clear from the outset.

4.3.1 Mandatory and Optional Decision Rules

Further decision rules are needed on which elements or message calls should be mandatory and which should be optional. This applies both to the task of going through the base specification and deciding which elements to include and which to exclude, and to the task of handling any new features required for the application profile. To do this, there first needs to be agreement on what the terms ‘mandatory’ and ‘optional’ are to be applied to in the application profile. There are two typical things to which these terms could apply:

- a) A data or transaction instance of the specification;
- b) A system that implements the specification.

Typically, when it is decided to make a field optional, it is because participants can provide usage scenarios where it would not make sense or be undesirable to have to include it. However ‘optional’ has been taken by some to mean that producers of systems do not have to implement it. This can result in the anomaly of implementers arbitrarily deciding which set of optional fields to leave out and yet they still claim conformance to the specification. Against this, it has to be pointed that:

- a) Taking such an approach, any two vendors’ systems are unlikely to interoperate unless they happen to leave out exactly the same subset of optional fields;
- b) There will always be valid instances of the specification which their systems cannot handle.

This interpretation of ‘optional’ applying to systems rather than to data instances leads to a highly unsatisfactory situation for users. Users are only interested in interoperability. For users, the only value of a conformance claim is if it leads to interoperability.

At the very least, when setting out an application profile it is necessary to distinguish between ‘Optional for data and transaction instances’ and ‘Optional for conforming systems.’ It is important to gain agreement, otherwise much time can be wasted through discussions being at cross purposes.

Where a subset of the base specification is all that is needed to meet the needs of a community, then the application profile can indicate which parts of the base specification do not need to be implemented by systems that are to support the application profile. These are then the elements that are ‘optional for systems’. All remaining mandatory parts and

‘optional for instances’ parts are then taken to be ‘mandatory for systems’. That is, any system that conforms to the profile must be able to handle instances that use any or all of the parts that are defined in the profile as optional. However, community and user representatives should be aware that every ‘mandatory for systems’ item has a direct implementation cost and that cost has to be passed back to users if the implementer is to stay in the game. Their task of gathering and entering the data for these elements will also carry a cost to them as the user community. Again this goes back to, and is part of, the cost benefit analysis. Guidance for the rules is:

- Further Optional and Mandatory decision rules therefore also need to be agreed. Typical ones are:
 - Include as mandatory if everyone needs it;
 - Include as optional for instances if everyone finds it useful;
 - Include as optional for instances if a significant majority need it.
- Other rules that need to be agreed are:
 - Should an element be in or out if a significant minority need it (probably include as optional);
 - Should an element be in or out if a significant majority find it useful (could go either way, but see Further Conflict Resolver next).
- The end rule is:
 - Otherwise exclude it (i.e., it is a part of the base specification that is not part of the application profile, but remain ‘optional for systems’ that implement the application profile, e.g., if a system has already implemented the whole of the base specification, it could well be desired that it should be able to import full base specification instances).

4.3.2 Further Conflict Resolver

Just as it is possible to have application profiles of a specification, so it is possible to have one or more sub-application profiles. If there is a significant minority with clear, unmet needs, which others actively don’t want, then a sub-application profile can be defined for the minority as part of the same process.

4.4 Project Group Guidelines

The next step is for the Project Group to create the application profile. It is useful for the duration of the work to maintain an ‘Issues List’. This can have headings for the issue number, an outline of the issue, pros, cons, the resolution and the reasons for it. This helps to keep the work focused and maintains a history of the decisions made. It also helps new members get up to speed with the work without needing a session to revisit all the arguments and agreements already made. The actual work can include the following tasks:

- a) Community representatives identify/prepare/bring a set of Usage Scenarios that exemplify the need;
- b) Abstract these into one or more generic Use Cases;
- c) Compare the Domain Model and the Base Specification to identify any new elements that may need to be added (note again both significant extra implementation costs and loss of interoperability stem from adding new elements to the existing base);
- d) Compare each Use Case with the base and, for each of the elements and actions, determine which are needed and which are not;
- e) Community Reps Identify/Prepare/Bring a set of Usage Scenarios that Exemplify the Need;
- f) Abstract these into one or more Generic Use Cases;
- g) Compare the Domain Model and the Base Specification to identify any new elements that may need to be added (Note: both significant extra implementation costs and loss of interoperability stem from adding new elements to the existing base);
- h) Use existing fields where possible, but take care not to ‘stretch’ them too far – if their meaning changes, then ‘semantic’ interoperability will be lost – which is hard to test for, and may also involve different processing on the part of the systems involved which can also lead to interoperability failure;

- i) Compare each Use Case with the Base Specification and, for each of the elements and actions, determine which are needed and which are not;
- j) Each use case may produce a different application profile, or it may be found that more than one use case is served by the same application profile, or it may be that one application profile meets all the use cases. However it is better to go through each use case separately and determine the required data that need to be exchanged and actions that are needed to support the exchanges;
- k) To get to the first draft of an application profile, it is better to go through the fields of the base specification fairly rapidly to get the large picture and then circulate for feedback;

The above activities determine the content of the application profile/s. The next steps are:

- l) Produce the first draft of the application profile, in accordance with the guidelines set out in Part 2 of this document and circulate for comment;
- m) If at all possible it is very helpful to have a prototype implementation of the profile carried out to test it in parallel with the later stages of its development. This tends to surface ambiguities, trivial errors, and perhaps features that are difficult to implement;
- n) Call meetings as necessary to resolve disagreements arising from the feedback and to refine the application profile. This can take time;
- o) It may be desirable, depending on the size and complexity of the profile, to leave a period of time between a 'Candidate' release of the profile and the 'Final' release, to allow several implementations to be completed and tested together. It is highly desirable to have at least two 'reference' implementations from different sources available that work together as these provide reliable systems for other implementers to test their systems against. It is even better if these can be open source with an unrestricted, non-viral license that allows other implementers to use them as templates for their own implementations;
- p) Release the final agreed version of the application profile.

5. Conformance Issues for Application Profiles

Recall that the ultimate justification for developing an application profile is to enable suppliers to more closely meet the needs of the target community. Having implementers adopt the profile as the basis for their tools, products, and services is the starting point for achieving interoperability within the community. But many of these implementers may have already adopted the relevant base specifications in any case. These existing implementations can be harnessed by ensuring that wherever possible, the application profile conforms to the base specification(s) it draws from. Thus there are two issues to be considered with respect to conformance:

- Conformance of the application profile to the base specification(s);
- Conformance of vendor tools, products, and services to the application profile.

5.1 Application Profile Conformance with the Base Specification(s)

It is preferable for Application Profiles to be interoperable with the base specification from which they are derived.

- Conformant and Non-Conformant Application Profiles:
 - An application profile is conformant with its base specification if all instances that conform to the profile also conform to the base specification;
 - An application profile is non-conformant with its base specification if there can be instances that conform to the profile but that do not conform to the base specification.
- An Interoperable Application Profile (of an Information Model):
 - May be a proper sub-set of the base specification (leaves out optional elements or commands);
 - Does not exclude any Mandatory fields;
 - May make Optional fields Mandatory;
 - Maintains the same vocabularies or only specifies vocabularies that are subsets of the vocabularies defined in the base specification.
- A Non-Interoperable Application Profile can result from:
 - Making mandatory elements optional;
 - Changing the data-typing of elements taken from the base;
 - Adding new elements to the base specification, unless they use extension points provided in the base specification and the base specification warns applications to allow for this;
 - Adding new terms to the vocabularies defined in the base specification;
 - Adding new vocabularies that replace vocabularies defined in the base specification;
 - Changing explicitly defined vocabularies of the base specification.

In general, implementations of application profiles that are conformant with the base specifications, i.e., are a subset, should be able to send information to systems that conform to the base specification, but systems that (fully) conform to the base specification may be able to generate information with a system that only conforms to the application profile cannot handle.

If the application profile only provides extensions, i.e., is a superset of the base specification, then conforming systems should be able to receive information from systems that only conform to the base specification but systems only conforming to the base specification cannot be expected to handle information from systems implementing a superset application profile.

5.2 Conformance of Implementations to the Application Profile

Ideally, during the period in which the application profile has been constructed, one or more implementers have committed the profile to working code. Interoperability and functionality can then be tested across these implementations and one or more adopted as reference implementations against which further implementations can be

tested. Events such as code bashes and plugfests are extremely useful in providing concrete instances of interoperability problems which can then be addressed in the profile and, if appropriate, also in the base specification. As the profile matures and becomes more stable, there may be a desire for a dedicated test system which can be used to test all implementations.

At this point it is perhaps worthwhile to recall that any conformance test, whether using a reference implementation or a dedicated test system cannot offer a 100% guarantee of interoperability. For anything other than trivial cases it is impossible to test exhaustively or replicate test conditions perfectly, so no matter how comprehensive the test in its scope, it cannot be conclusive. It is rather a means of reducing risk of non-interoperability by applying a common test to all comers. But ultimately, it only indicates pass or fail against the stated test and thus constitutes an indirect indicator of the likelihood of two tested items inter-working directly.

Most of the original IMS specifications are defined as an information model with an XML binding. Future revisions of the specifications are, where appropriate, attempting to provide a behavioral model by the addition of an abstract interface definition. An application profile by comparison has an adapted information model for each of the specifications it utilizes, along with a specific interface binding for each. Thus in defining a profile, additional decisions need to be made regarding technologies being adopted for implementation (i.e., the technology binding) and the service model beneath the APIs determining how data exchanges are transacted (e.g., sequencing, control, fault recovery). The conformance constraints for an application profile may therefore cover runtime behavior in a very precise manner.

The tests to be performed by such a test system should be documented and made available as a set of acceptance test criteria. Implementers can then examine this to see exactly what tests their offerings will be subjected to and what the pass and error state conditions are for each test.

About This Document

Title	IMS Application Profile Guidelines Part 1 – Management Overview
Editors	Kevin Riley (IMS)
Team Co-Leads	Scott Wilson (JISC)
Version	1.0
Version Date	01 August 2005
Status	Final
Summary	This document offers guidance on specifying an Application Profile for a Community of Practice.
Revision Information	10 October 2005
Purpose	This document is not a formal IMS specification. Its purpose is to act as an aid to adopters of IMS specs in drafting up an Application Profile, detailing how they are using the specifications for their implementation. As such, the Application Profile has value as an aid to implementation across a community.
Document Location	http://www.msglobal.org/ap/apv1p0/imsap_oviewv1p0.html

To register any comments or questions about this specification please visit:
<http://www.msglobal.org/developers/ims/imsforum/categories.cfm?catid=17>

List of Contributors

The following individuals contributed to the development of this document:

Name	Organization
John Bell	Ufi
Ingo Dahn	University of Koblenz-Landau
Norm Friesen	Industry Canada
Peter Hope	Canadian National Defence
Jeff Merriman	MIT
Bill Olivier	JISC
Kevin Riley	IMS
Anthony Roberts	Industry Canada
Colin Smythe	IMS
Scott Wilson	JISC

Revision History

Version No.	Release Date	Comments
Final 1.0	10 October 2005	The first formal release of this document.

Index

A

Acceptance test criteria 4
 ADL 4, 10
 AICC 4
 ALIC 4, 9
 API 4, 6
 Application Profile 2, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

B

BSI 4, 10

C

CEN workshop agreement 5
 CEN/ISSS 5, 9
 Certification 5, 6
 Conformance 2, 4, 5, 6, 8, 16, 19, 20
 Content Packaging 5, 9, 10

D

DOM 5

E

European e-Learning Industry Group (ELIG) 5
 European Institute for e-Learning (EifEL) 5
 European SchoolNet 5

H

HTTP 5

I

Implementation 2, 4, 5, 6, 9, 13, 15, 17, 18, 19, 20, 21
 International Conformance Program (ICP) 5
 Interoperability 2, 4, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20
 ISO/IEC SC36 5

L

Learner Information Package 6, 9, 10
 Learning Object Metadata 6
 Learning Technology Standardization Committee of the IEEE 5
 Localization 10

M

Mandatory 2, 4, 16, 17, 19
 Massachusetts Institute of Technology 6, 21
 Meta-data 2, 8, 9, 10

O

Object Management Group 6
 Open Knowledge Initiative 6

P

Profiling 2, 4, 8

Q

Question and Test Interoperability 9, 10

R

Return on investment 6

S

Schools Interoperability Framework (SIF) 6
 Sharable Content Object Reference Model (SCORM) 5, 6, 10
 SOAP 6
 Stub 6

T

Test system 20

U

UML 6, 7, 9, 11
 User Interface 6

V

Vocabularies 4, 5, 8, 19

W

WSDL 6

X

XMI 7
 XML 6, 7, 11, 20

IMS Global Learning Consortium, Inc. ("IMS/GLC") is publishing the information contained in this IMS Application Profile Guidelines Overview ("Specification") for purposes of scientific, experimental, and scholarly collaboration only.

IMS/GLC makes no warranty or representation regarding the accuracy or completeness of the Specification.

This material is provided on an "As Is" and "As Available" basis.

The Specification is at all times subject to change and revision without notice.

It is your sole responsibility to evaluate the usefulness, accuracy, and completeness of the Specification as it relates to you.

IMS/GLC would appreciate receiving your comments and suggestions.

Please contact IMS/GLC through our website at <http://www.imsglobal.org>

Please refer to Document Name: IMS Application Profile Guidelines Overview

Date: 10 October 2005