IMS Outcomes Management Service Information Model

Version 1.0

Public Draft

Version 2.0

Date Issued: 30 June 2011

Latest version: http://www.imsglobal.org/lis/

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 © 2011 IMS Global Learning Consortium. All Rights Reserved.

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.

Permission is granted to all parties to use excerpts from this document as needed in producing requests for proposals.

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.

© 2011 IMS Global Learning Consortium, Inc.
All Rights Reserved.

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

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Outcomes Management Service Overview

The Outcomes Management Service (OMS) specification is the definition of how systems manage the exchange of information that describes Outcomes. The Outcomes Management Service specification is constructed following the recommendations documented in the IMS GLC Abstract Framework (IAF) [IAF, 03a], [IAF, 03b], [IAF, 03c]. This means that this specification is based upon the concepts of:

- Interoperability – Outcomes Management Service focuses on the exchange of Outcomes information between systems. There are no definitions in the specification on how the data is managed within the systems;
- Service-oriented – Outcomes Management Service defines the exchange of information in terms of the services being supplied by the collaboration of the systems;
- Component-based – for example, the Outcomes Management Service is combined with the Group Management Service, Person Management Service, Course Management Service, Membership Management Service and Bulk Data Exchange Management Service to provide the Learning Information Services [LIS, 11a], [LIS, 11b];
- Layering – the Outcomes Management Service is a part of the Application Services layer but it interacts with the services available in the Common Services layer e.g., authentication;
- Behaviors and Data Models – the Outcomes Management Service are defined in terms of their Behaviors and data models. The Behaviors cause changes in the state of the data model and the state of the data model will only be altered as a result of a clearly defined Behavior;
- Multiple Bindings – the Outcomes Management Service information model is to be defined using the Unified Modelling Language (UML). This enables reliable mapping of the information model into a range of different bindings. The binding of immediate importance is the Web Services Description Language (WSDL);
- Adoption – whenever appropriate, the Outcomes Management Services specification makes use of other IMS GLC and non-IMS GLC standards and specifications.

The logical structure of the Outcomes is composed of LineItems, Results and ResultValues. A LineItem is a collection of Results for a particular CourseOffering, CourseSection, etc. and each Result and/or LineItem may have an associated set of values, ResultValue, which are used to constrain the Result. The standard create, read, update, replace, delete and simple discovery operations are available for managing LineItems, Results and ResultValues. Other operations are defined to support specific business requirements. Full operation of all of the Behaviors in this service will require access to the data models of the corresponding IMS GLC Course Management Service (CMS) [CMS, 11] and the IMS GLC Membership Management Service (MMS) [MMS, 11] specifications.

1.2 Scope and Context

This document is the IMS GLC Outcomes Management Services Information Model v1.0 and as such it is used as the basis for the development of the following documents:

a) IMS GLC Outcomes Management Service WSDL Binding v1.0 [OMS, 11] – the description of the WSDL binding of the Information Model.

The core uses-cases for the Outcomes Management Service are described as a subset of the Learning Information Services Specification [LIS, 11b].

This information model defines the Outcomes Management Service Abstract Application Programming Interface (a-API). The Learning Information Services specification, of which the Outcomes Management Service is a component, is a series of Behavioral models that define how the data models are to be manipulated. These Behavioral models are described using the Unified Modelling Language (UML) [SDN07, 07].

1.3 Structure of this Document

The structure of this document is:

2. Outcomes Management Service Description — The description of the overall structure and operation of the Outcomes Management Service. This includes the description of the architectural model and the domain object model;

3. Behavioral Model — The definition of the operations of Outcomes Management Service application service. This focuses on the description of the Behaviors supported by the service;

4. Interface Data Model — The definition of the data models exchanged between the Outcomes Management Service End Systems. These are the parameters exchanged across the interoperability interface;

5. End System Data Model — The definition of the data models for the Outcomes Management Service End Systems. This addresses the persistence of the data with respect to interoperability;

6. Extending and Profiling the Service — Identification of the ways in which the Outcomes Management Service can be extended both in terms of the addition of new constituent services and proprietary extensions to a service;

Appendix A Summary List of Service Status Codes — A summary list of the status codes by for the operations of the Outcomes Management Service;

Appendix B Vocabularies — A summary of the set of vocabularies that are used within the specification;

Appendix C File-based Data Exchange — The out-of-band file exchange used in response to receiving a URL for an external data file that contains the request data.

1.4 Version Compatibility

The Outcomes Management Service was not developed in the IMS GLC Enterprise Services v1.0 specification [ES, 04]. Instead, a simplified form of functionality was supported using the IMS GLC Membership Management Services (MMS) v1.0 [MMS, 04]. In general, there is NO backwards compatibility between the usage of the OMSv1.0 and the ways in which MMSv1.0 has been implemented to support outcomes management. Vendors may define compatibility bridges for their own implementations but these are outside the scope of this specification.

1.5 Nomenclature

a-API	Abstract Application Programming Interface
API	Application Programming Interface
CMS	Course Management Service
IAF	IMS GLC Abstract Framework
LIS	Learning Information Services
LTU	Learning Tools Interoperability
MMS	Membership Management Service
OMS	Outcomes Management Service
PIM	Platform Independent Model
PSM	Platform Specific Model
RFC	Request For Comment
SDN	Specification Development Note

UML	Unified Modelling Language
WSDL	Web Services Description Language

1.6 References

[APG, 05a] *IMS GLC Application Profile Guidelines Overview: Part 1 – Management Overview v1.0*, IMS Global Learning Consortium, K.Riley, October 2005. http://www.imsglobal.org/ap/index.html.

[APG, 05b] *IMS GLC Application Profile Guidelines White Paper: Part 2 Technical Manual*, S.Wilson and K.Riley, Version 1.0, IMS Global Learning Consortium, October 2005. http://www.imsglobal.org/ap/index.html.

[BDEMS, 11] *IMS GLC Bulk Data Exchange Management Service v1.0 Information Model v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[CMS, 11] *IMS GLC Course Management Service v1.0 Information Model v2.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[ES, 04] *IMS GLC Enterprise Services Overview v1.0 Final Release*, C.Vento and C.Smythe, IMS Global Learning Consortium, June 2006.

[GMS, 11] *IMS GLC Group Management Service v1.0 Information Model v2.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[GWS, 05] *IMS GLC General Web Services WSDL Binding Guidelines v1.0 Final Specification*, C.Schroeder, J.Simon and C.Smythe, IMS Global Learning Consortium, December 2005.

[IAF, 03a] *IMS Abstract Framework: Applications, Services & Components v1.0*, Ed. C.Smythe, IMS Global Learning Consortium, July 2003.

[IAF, 03b] *IMS Abstract Framework: Glossary v1.0*, Ed. C.Smythe, IMS Global Learning Consortium, July 2003.

[IAF, 03c] *IMS Abstract Framework: White Paper v1.0*, Ed. C.Smythe, IMS Global Learning Consortium, July 2003.

[LIS, 11a] *IMS GLC Learning Information Services v2.0 Overview v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[LIS, 11b] *IMS GLC Learning Information Services v2.0 Specification v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[LIS, 11c] *IMS GLC Learning Information Services v2.0 Best Practices & Implementation Guide v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[MMS, 04] *IMS GLC Membership Management Services Information Model v1.0 Final Release*, C.Vento and C.Smythe, IMS Global Learning Consortium, June 2004.

[MMS, 11] *IMS GLC Membership Management Service v2.0 Information Model v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[OMS, 11] *IMS GLC Outcomes Management Service v1.0 WSDL Binding v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, May 2011.

[SDN07, 06] *IMS GLC Specification Note: UML Profile for Platform Independent Model Descriptions of Specifications for Data Models v1.0*, C.Smythe, IMS Global Learning Consortium, October 2006.

[SDN11, 06] *IMS GLC Specification Note 11: Vocabulary Definition, Registration & Maintenance Procedures*, C.Smythe, IMS Global Learning Consortium, October 2006.

[VDEX, 04] *IMS/GLC Vocabulary Definition Exchange Best Practice and Implementation Guide, Version 1.0 Final Specification*, A. Cooper, IMS Global Learning Consortium, 2005. Online version: http://www.imsglobal.org/vdex/vdexv1p0/imsvdex_bestv1p0.html

2 Outcomes Management Service Description

2.1 An Abstract Representation

It is important to remember that this document contains the description of the underlying information model in terms of the abstract API. The manner in which this abstract representation is visualized is not intended to dictate the implementation form of an Outcomes Management Service. The breakdown of the service into its interface classes is a convenient way to document the set of Behaviors. The internal organization of an implementation of the full abstract API is beyond the scope of this specification. The only constraint is that the external Behavior of the abstract API complies with this specification. This means that a .NET, J2EE, etc. physical implementation of this abstract API does not have to represent the functionality using the same breakdown of operations/methods. This physical implementation is not subject to the conformance specification.

It is important to note that the UML representation of the interfaces is used to help develop and document the Outcomes Management Service Information Model. It is not a requirement for an implementation to implement this interface as defined i.e., to use the same parameters, etc. Conformance against this specification will be confirmed by inspecting the appropriate binding of the information model and ensuring that the relevant information is present and that different sequences of activity result in the predicted and mandated Behavior. It is essential that the Behaviors described by each of the operations are fully supported and it is also essential that the Behaviors described by different sequences be also maintained.

2.2 Outcomes Management Service Architecture & Specification Model

The basic architectural model for the Outcomes Management Service specification is shown in Figure 2.1. In this architecture the scope of the IMS GLC Outcomes Management Service specification is shown as the dotted line. The scope of the interoperability is the data and Behavioral models of the objects being exchanged.

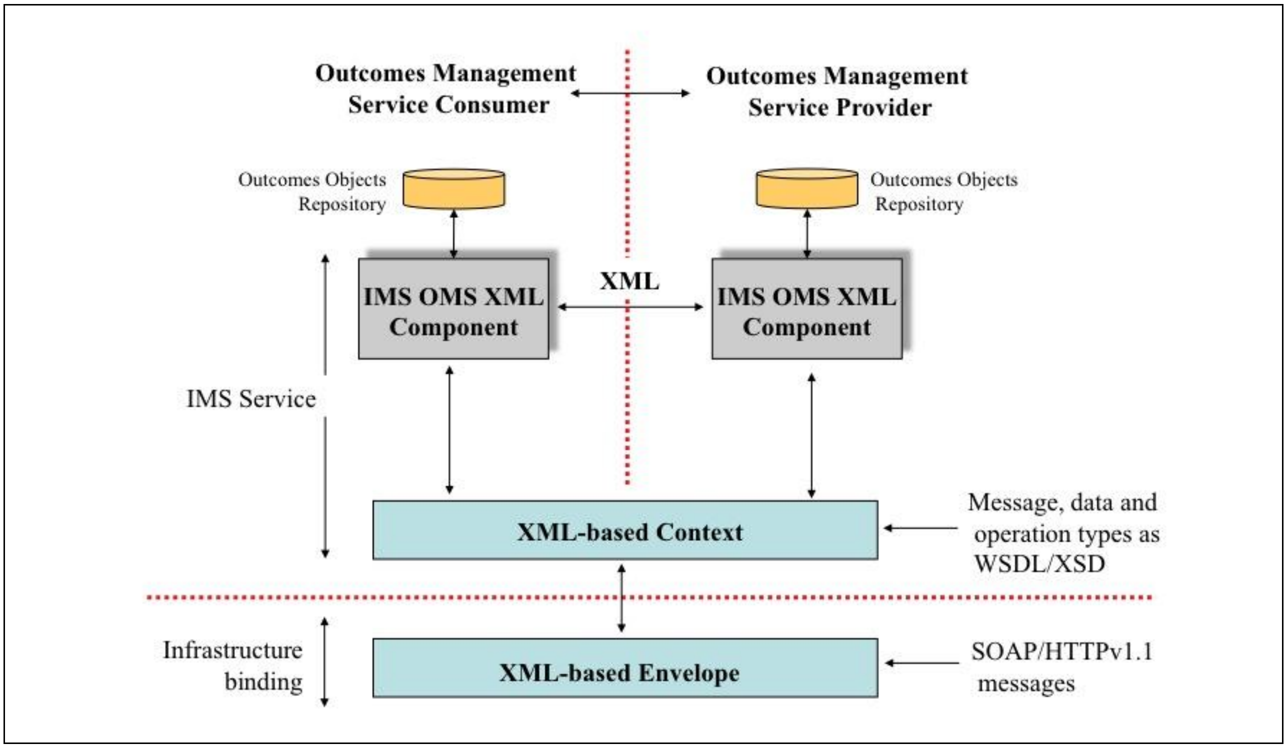

Outcomes Management Service Consumer
Outcomes Management Service Provider
Outcomes Objects Repository
Outcomes Objects Repository
IMS OMS XML Component
XML
IMS OMS XML Component
IMS Service
XML-based Context
Message, data and operation types as WSDL/XSD
Infrastructure binding
XML-based Envelope
SOAP/HTTPv1.1 messages

Figure 2.1 Outcomes management service architecture model.

It is important to remember that the structure of the exchanged information has NO bearing on how the same information is contained within the 'source' and 'target' Learning Information Services systems (the Outcomes object repositories in the two end-systems). It is simply a representation of the data used to facilitate exchange between the end-systems. The only constraint on the end-system repositories is that they provide data persistence consistent with the required behavior.

2.3 Outcomes Composite Objects

It is important to note that this is an **interoperability** specification and as such it makes no statements about how information is stored within the exchanging end systems. The objects in the end-systems **must** be persistent otherwise sequences of operation on the same object will not be possible. Reference to these objects in the interface is through a ‘sourcedId’ however this identifier does not have to be the key stored within the end-systems. If different keys are used in the end-systems then it is the responsibility of the end-systems to maintain the mapping between that key and the ‘sourcedId’ i.e., the interface must never be exposed to the keys of the end-systems.

There is no such thing as an Outcomes object. Instead, Outcomes are reflected in three types of object each of which has its own ‘sourcedId’ which are linked to create the Outcomes data model as shown in Figure 2.2.

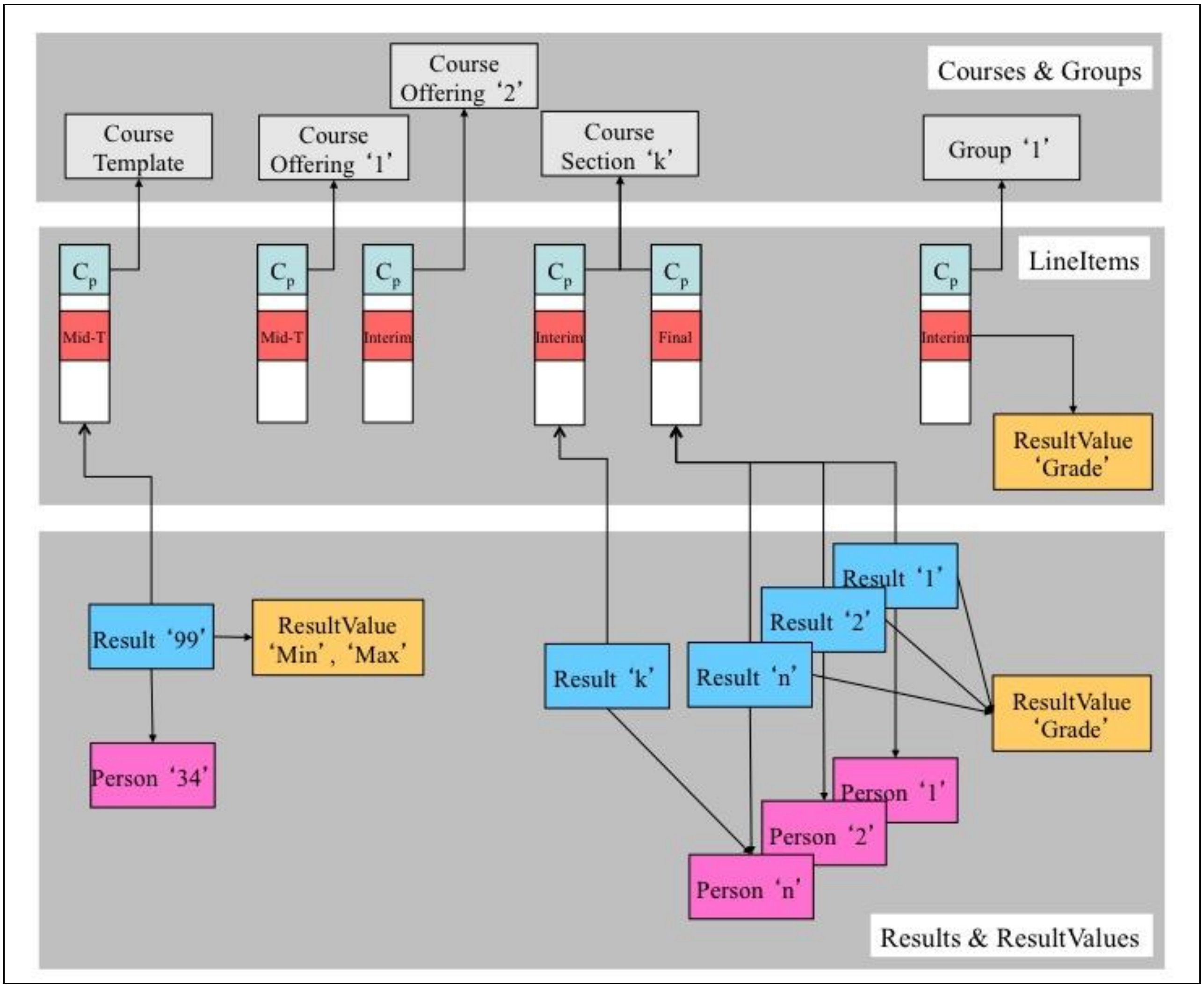


Figure 2.2 Schematic representation of the outcomes data model.

The three classes of object are:

- LineItem – the set of results for a particular CourseTemplate, CourseOffering, CourseSection and Group (each result is a Result object);
- Result – the container for a result for a particular Person (each result can be related to a ResultValue object);

- ResultValue – the permitted range of values for a Result and/or a LineItem.

As shown in Figure 2.2 the LineItem is the key object that is used to establish the relationship between course components and the results for each person. A LineItem has three fields:

- The identifier (that is the sourcedId) of the associated course components (denoted as C_p in Figure 2.2). More than one LineItem can point to the same course component;
- The type of LineItem (this is an open vocabulary as in Figure 2.2 this includes 'Interim', 'Final', etc.);
- The ResultValue that defines the nature of all of the results associated with the LineItem (each Result must be associated with one, and only one, LineItem).

Each Result is associated with a LineItem and a Person i.e., the Person who has received the result for that particular course component. A Person may have several results, for the same or different course components, and so more than one result may be associated with a Person. The unit of marking for the Result is defined by the ResultValue associated with the Result. In general, many Results will be associated with each type of ResultValue. A ResultValue can also be assigned to a LineItem thereby defining all Results to be of the same ResultValue.

A system that wishes to support all of the behaviors of the OMS must also contain the Course data model as defined in the IMS GLC Course Management Services Information Model v1.0 specification [CMS, 11] and may have to support the Group data model in the IMS GLC Group Management Services Information Model v2.0 specification [GMS, 11].

2.4 Synchronous & Asynchronous Services

Within the context of the Outcomes Management Service the definition of synchronous and asynchronous services is:

- Synchronous – the source service is blocked until the final response from the target service is received. A schematic representation of the information flow for a synchronous service is shown in Figure 2.3;
- Asynchronous – the source service is not blocked and so more than one request can be outstanding at any moment in time. A schematic representation of the information flow for an asynchronous service is shown in Figure 2.4.

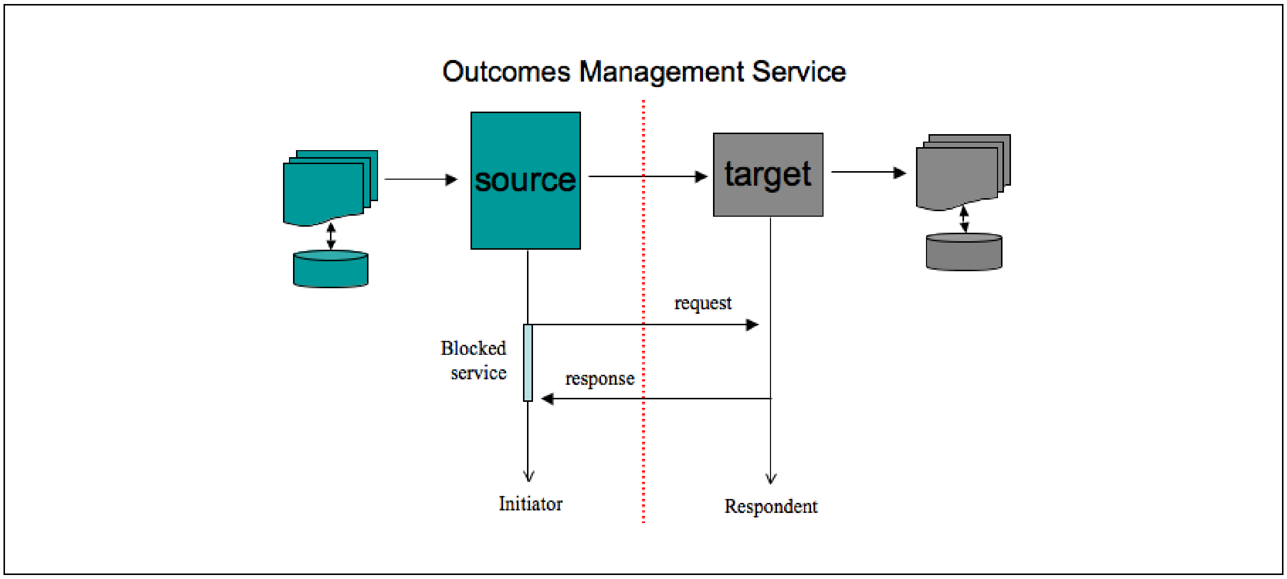


Figure 2.3 Synchronous service action flow.

It is stressed that the abstract-API does not differentiate between synchronous and asynchronous services[1]. The support for these two approaches is differentiated at the binding level only.

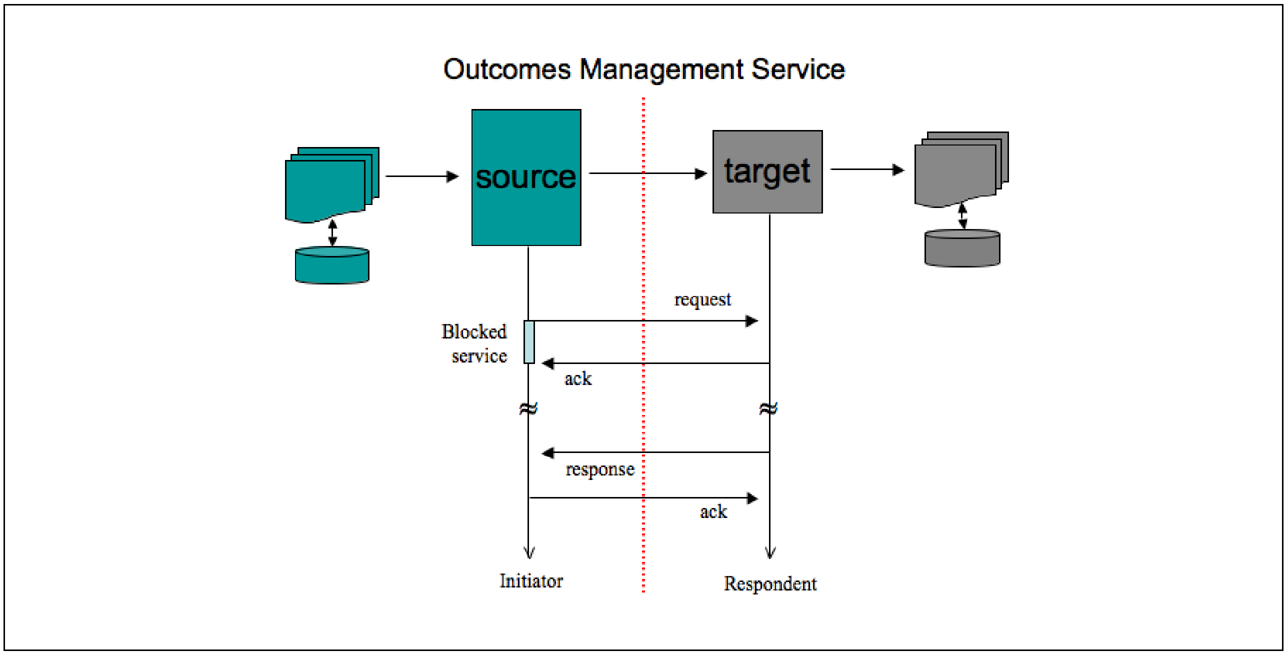


Figure 2.4 Asynchronous service action flow.

The key difference is that for an asynchronous service more than one request can be issued at any one time (it should be noted that an asynchronous service can be supported using synchronous messaging). In both cases the service assumes a perfect messaging system i.e., request, response and acknowledgement messages have a guaranteed delivery grade of service.

2.5 Handling the Service Status Codes

Each operation in a service is mapped to an appropriate message exchange pattern. Any response/acknowledgement message will contain status information. This status information provides contextual information about the completed success or otherwise of the operation. There are two types of status information that are available to the end-systems:

- Business transaction – these are the status reports that reflect the business logic of the transactions being exchanged by the end-systems. This status information will be contained within the message header under a specially defined data structure. The status information contained herein is also used to contain any error codes i.e., error reporting is handled as a subset of status information reporting;
- Messaging fault– these are fault codes that are reported by the messaging infrastructure and which are carried in the messages.

It is important to note that messaging errors may indicate that the original request never reached the service provider end-system. In this case the service consumer implementation that handles the status information is responsible for mapping the message infrastructure failure codes to the equivalent business transaction status code. The message infrastructure failure codes have no meaning with respect to an IMS GLC specification. The IMS GLC specifications do not describe how the status information is to be handled within an end-system i.e., this will depend on how the abstract API is physically realized within an implementation. Therefore, it is important that an implementation can:

[1] In many implementations of the abstract-API the synchronous and asynchronous services would require different operation calls. This is just one example where an implementation does not match the definition of the abstract-API.

- Combine the transaction status information and any message fault error codes in a single integrated status reporting mechanism. Any other system failure information that is made available by an implementation should also use the same mechanism;
- Examine the status information reported after the completion of the appropriate phase of an operation and especially once the operation has been completed. This may require an explicit status information call or it may be reported as part of the API call;
- Differentiate the status information reports for each transaction within an operation. Remember that some specifications provide operations that can contain more than one transaction request and that a different status report may be given for each of those transactions.

Exception handling is the system’s response to known or unknown error conditions. Exception handling is outside the scope of an IMS GLC specification. However, an error condition should not cause the end-systems to fail in an uncontrolled manner. The requirement for every operation to return status information is to enable an implementation to terminate in a controlled fashion.

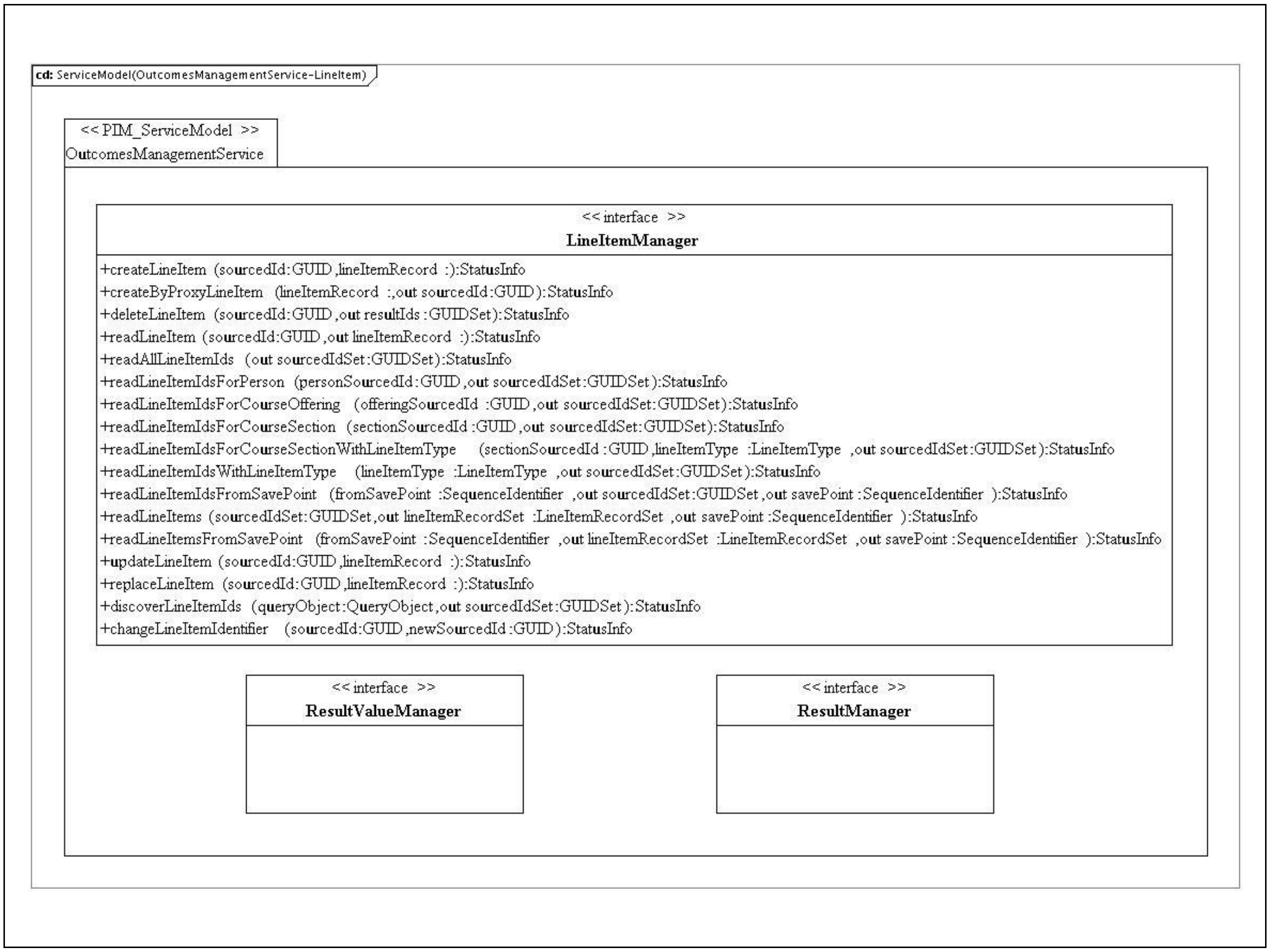
3 Behavioral Model

3.1 Service Definition

The OutcomesManagementService is used to model the service responsible for manipulating information about outcomes. The OutcomesManagementService interfaces are shown in Figures 3.1, 3.2 and 3.3. The OutcomesManagementService is split into three interfaces: LineItemManager that supports the manipulation of LineItem objects; ResultManager that supports the manipulation of Result objects; ResultValueManager that supports the manipulation of ResultValue objects.

3.2 LineItemManager Interface Description

The LineItemManager interface class describes the operations that are permitted on LineItem objects (as shown in Figure 3.1). The set of operations are summarized in Table 3.1.



```
+createLineItem (sourcedId:GUID ,lineItemRecord :):StatusInfo
+createByProxyLineItem (lineItemRecord :,out sourcedId:GUID ):StatusInfo
+deleteLineItem (sourcedId:GUID ,out resultIds :GUIDSet):StatusInfo
+readLineItem (sourcedId:GUID ,out lineItemRecord :):StatusInfo
+readAllLineItemIds (out sourcedIdSet:GUIDSet):StatusInfo
+readLineItemIdsForPerson (personSourcedId:GUID ,out sourcedIdSet:GUIDSet ):StatusInfo
+readLineItemIdsForCourseOffering (offeringSourcedId :GUID ,out sourcedIdSet:GUIDSet):StatusInfo
+readLineItemIdsForCourseSection (sectionSourcedId :GUID ,out sourcedIdSet:GUIDSet):StatusInfo
+readLineItemIdsForCourseSectionWithLineItemType (sectionSourcedId :GUID ,lineItemType :LineItemType ,out sourcedIdSet:GUIDSet ):StatusInfo
+readLineItemIdsWithLineItemType (lineItemType :LineItemType ,out sourcedIdSet:GUIDSet ):StatusInfo
+readLineItemIdsFromSavePoint (fromSavePoint :SequenceIdentifier ,out sourcedIdSet:GUIDSet ,out savePoint :SequenceIdentifier ):StatusInfo
+readLineItems (sourcedIdSet:GUIDSet ,out lineItemRecordSet :LineItemRecordSet ,out savePoint :SequenceIdentifier ):StatusInfo
+readLineItemsFromSavePoint (fromSavePoint :SequenceIdentifier ,out lineItemRecordSet :LineItemRecordSet ,out savePoint :SequenceIdentifier ):StatusInfo
+updateLineItem (sourcedId:GUID ,lineItemRecord :):StatusInfo
+replaceLineItem (sourcedId:GUID ,lineItemRecord :):StatusInfo
+discoverLineItemIds (queryObject:QueryObject ,out sourcedIdSet:GUIDSet ):StatusInfo
+changeLineItemIdentifier (sourcedId:GUID ,newSourcedId :GUID ):StatusInfo
```

Figure 3.1 OutcomesManagementService LineItemManager interface definitions.

Table 3.1 Summary of operations for LineItemManager.

Operation	Description
createLineItem	To request the creation of a populated LineItem object on the target system where the source is responsible for the allocation of the unique SourcedId.
createByProxyLineItem	To request the creation of a populated LineItem object on the target system where the target is responsible for the allocation of the unique SourcedId.
deleteLineItem	To request the deletion of a LineItem object. The LineItem object is deleted with all of its associated relationships.
readLineItem	To read the full contents of the identified LineItem object. The target must return all of the data it has for the identified LineItem object.
readAllLineItemIds	To obtain all the SourcedIds assigned to LineItem objects.
readLineItemIdsWithLineItemType	To obtain the set of SourcedIds for the LineItem objects which have a particular state.
readLineItemIdsForPerson	To obtain the set of SourcedIds for the LineItem objects which are associated with a particular Person object.
readLineItemIdsForCourseOffering	To obtain the set of SourcedIds for the LineItem objects which are associated with a particular CourseOffering object.
readLineItemIdsForCourseSection	To obtain the set of SourcedIds for the LineItem objects which are associated with a particular CourseSection object.
readLineItemIdsForCourseSectionWithLineItemRType	To obtain the set of SourcedIds for the LineItem objects with the required state, which are associated with a particular CourseSection object.
readLineItemIdsFromSavePoint	To obtain the set of SourcedIds for LineItem objects which have been altered since the requested reference point. The reference point is set as ‘zero’ at creation and incremented after every write operation.
readLineItems	To obtain the LineItem objects for a defined set of SourcedIds. This results in a single transaction that may require the exchange of a large volume of data in the response message.
readLineItemsFromSavePoint	To obtain the set of LineItem objects which have been altered since the requested reference point. The reference point is set as ‘zero’ at creation and incremented after every write operation. This results in a single transaction that may require the exchange of a large volume of data in the response message.
updateLineItem	To write new content into the identified LineItem object. The target must write the new data into the LineItem object. This is an additive operation.
replaceLineItem	To replace the content of the identified LineItem object.

	The target must write the new data into the LineItem object. This is a destructive write-over of all of the original information. In the case of the object not existing, this operation acts as an implied ‘createLineItem’.
discoverLineItemIds	To obtain the set of SourcedIds for LineItem objects whose properties agree with those defined in the query/filter.
changeLineItemIdentifier	To change the SourcedId for a LineItem object.

Note: In most cases the above operations act on a single instance of a Group object i.e., ‘createLineItem’, ‘createByProxyLineItem’, ‘deleteLineItem’, ‘readLineItem’, ‘replaceLineItem’ and ‘updateLineItem’.

3.2.1 CreateLineItem() Operation

Name:	createLineItem
Return Function Parameter:	*StatusInfo* – the status of the creation request. The permitted status codes are defined in Tables 3.2 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the sourcedId allocated by the source system. This is the identifier that must also be assigned within the target system. *lineItemRecord:LineItemRecord* – the LineItem data that is to be stored in the new object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'createLineItem' request the target is instructed to create the populated LineItem object and to allocate that structure the SourcedId passed by the source. If the supplied SourcedId has already been allocated to another object then the request is rejected and the appropriate failure code is returned. The reference point identifier is set to 'zero' for the object in both the source and target.
Notes:	This request contains the initial content for the LineItem object. More content can be added/replaced using the 'updateLineItem' and/or 'replaceLineItem' requests.

Table 3.2 Status codes for the 'createLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The creation request has been fully and successfully implemented by the target system and the object has been created with the supplied SourcedId.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocinusefail'	The target could not allocate the unique SourcedId to the object as the identifier is already in use.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=overflowfail'	The target could not create the LineItem record due to lack of target allocation memory.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	An invalid LineItemType value has been supplied.
'CodeMajor=Failure'	An unknown context is supplied.

‘Severity=Status’ ‘CodeMinor=contextunknown’	
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=gradingnotpermitted’	The service consumer is not authorized for grade upload for this CourseSection object.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary extensions that are used in the object’s data model.

3.2.2 CreateByProxyLineItem() Operation

Name:	createByProxyLineItem
Return Function Parameter:	*StatusInfo* – the status of the creation request. The permitted status codes are defined in Tables 3.3 and A.2.
Supplied (in) Parameters:	*lineItemRecord:LineItemRecord* – the LineItem data that is to be stored in the new object.
Returned (out) Parameters:	*sourcedId:GUID* – the identifier allocated by the target to the newly created LineItem object.
Behavior:	When the source issues the 'createByProxyLineItem' request the target is instructed to create the populated CourseLineItem object and to allocate that record a unique SourcedId. The reference point identifier is set to 'zero' for the LineItem object in both the source and target.
Notes:	This request contains the initial content for the LineItem object. More content can be added/replaced using the 'updateLineItem' and/or 'replaceLineItem' requests.

Table 3.3 Status codes for the 'createByProxyLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The creation request has been fully and successfully implemented by the target system and the object has been created with the SourcedId generated by the target.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocfail'	The target could not allocate a unique SourcedId to the object because there are no more spare identifiers available.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=overflowfail'	The target could not create the object due to lack of target allocation memory.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	An invalid LineItemType value has been supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=contextunknown'	An unknown context is supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=gradingnotpermitted'	The service consumer is not authorized for grade upload for this CourseSection object.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success'	The target has stored a subset of the sent data record i.e., some of the

‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	optional data has not been stored (all mandatory data has been supplied and stored).
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary extensions that are used in the object’s data model.

3.2.3 DeleteLineItem() Operation

Name:	deleteLineItem
Return Function Parameter:	*StatusInfo* – the status of the deletion request. The permitted status codes are defined in Tables 3.4 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier to be used by the target to identify the LineItem object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'deleteLineItem' request the target is instructed to delete the identified LineItem object and to remove any associations. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned. The LineItem is deleted and all of the associated Result objects.
Notes:	Deletion of the LineItem object does not necessarily result in the destruction of the data within the server. The true state of the data in the target is unknown.

Table 3.4 Status codes for the 'deleteLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The deletion request has been fully and successfully implemented by the target system and the objects have been deleted. The corresponding child objects have been deleted.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be deleted.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor= deletefailure'	The target system has not been able to delete the identified object because there are Result objects associated with the LineItem.

3.2.4 ReadLineItem() Operation

Name:	readLineItem
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.5 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the LineItem object to be read.
Returned (out) Parameters:	*lineItemRecord:LineItemRecord* – the LineItem data that is read from the object.
Behavior:	When the source issues the 'readLineItem' request the target is charged with retrieving the identified object from its database and returning this data to the source. The target is responsible for ensuring that the object contains valid data. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The returned LineItem object can only be trusted if the corresponding status code is 'success'.

Table 3.5 Status codes for the 'readLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the source system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has only returned a subset of the data expected by the source e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=unknownextension'	The source cannot process the proprietary extensions that are used in the object's data model.

3.2.5 ReadAllLineItemIds() Operation

Name:	readAllLineItemIds
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.6 and A.2.
Supplied (in) Parameters:	None.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of sourcedIds for all of the LineItem objects in the target's Outcomes database.
Behavior:	When the source issues the 'readAllLineItemIds' request the target is charged with returning the SourcedIds of all of the LineItems in its line item database.
Notes:	If no SourcedIds have been allocated then the returned data set is empty and the success status code returned.

Table 3.6 Status codes for the 'readAllLineItemIds' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.2.6 ReadLineItemIdsForPerson() Operation

Name:	readLineItemIdsForPerson
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.7 and A.2.
Supplied (in) Parameters:	*personSourcedId:GUID* – the identifier of the Person object whose LineItem sourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the LineItem objects for the Person in the target's Outcomes database.
Behavior:	The target system must return all of the LineItem SourcedIds for the identified Person object. If the Person object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned. The association between a Person and the LineItems is defined through a Membership of a Course object e.g., a CourseOffering. The LineItems are then searched for those that have the association with the corresponding Course object.
Notes:	The returned object can only be trusted if the corresponding status code is 'success'. This Behavior requires that the system support the Membership and Course data models [MMS, 11], [CMS, 11].

Table 3.7 Status codes for the 'readLineItemIdsForPerson' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The Person object identifier is unknown in the target system and so the relationships could not be identified.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the object databases and so cannot locate the identifiers.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.2.7 ReadLineItemIdsForCourseOffering() Operation

Name:	readLineItemIdsForCourseOffering
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.8 and A.2.
Supplied (in) Parameters:	*offeringSourcedId:GUID* – the identifier of the CourseOffering object whose LineItem sourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of sourcedIds for all of the LineItem objects for the CourseOffering in the target's Course database.
Behavior:	The target system must return all of the LineItem SourcedIds for the identified CourseOffering object. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.8 Status codes for the 'readLineItemIdsForCourseOffering' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the LineItem identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The CourseOffering object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the LineItem database and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.2.8 ReadLineItemIdsWithLineItemType() Operation

Name:	readLineItemIdsWithStatus
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.9 and A.2.
Supplied (in) Parameters:	*lineItemType:LineItemType* – the LineItem objects with a particular column type whose SourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the LineItem objects of the appropriate status value in the target's Outcomes database.
Behavior:	The target system must return all of the LineItem SourcedIds that have the supplied status value. If an invalid status value is supplied then the request is rejected and the appropriate failure code is returned.
Notes:	The returned object can only be trusted if the corresponding status code is 'success'.

Table 3.9 Status codes for the 'readLineItemIdsWithLineItemType' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the LineItem identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	An invalid LineItemType value has been supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the LineItem database and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.2.9 ReadLineItemIdsForCourseSection() Operation

Name:	readLineItemIdsForCourseSection
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.10 and A.2.
Supplied (in) Parameters:	*sectionSourcedId:GUID* – the identifier of the CourseSection object whose LineItem sourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the LineItem objects for the CourseSection in the target's Course database.
Behavior:	The target system must return all of the LineItem SourcedIds for the identified CourseSection object. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.10 Status codes for the 'readLineItemIdsForCourseSection' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the LineItem identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The CourseSection object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the LineItem object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the source system.

3.2.10 ReadLineItemIdsForCourseSectionWithLineItemType() Operation

Name:	readLineItemIdsForCourseSectionWithLineItemType
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.11 and A.2.
Supplied (in) Parameters:	*sectionSourcedId:GUID* – the identifier of the CourseSection object whose LineItem sourcedIds are required. *lineItemType:LineItemType* – the LineItem objects with a particular column type whose SourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the LineItem objects of the required lineItemType for the CourseSection in the target's Course database.
Behavior:	The target system must return all of the LineItem SourcedIds for the identified CourseSection object. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.11 Status codes for the 'readLineItemIdsForCourseSectionWithLineItemType' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the LineItem identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The CourseSection object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	An invalid LineItemType value has been supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the LineItem object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the source system.

3.2.11 ReadLineItemIdsFromSavePoint() Operation

Name:	readLineItemIdsFromSavePoint
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the read request. The permitted status codes are given in Tables 3.12 and A.2.
Supplied (in) Parameters:	*fromSavePoint:SequenceIdentifier* – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the LineItem objects stored on the target. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readLineItemIdsFromSavePoint' the target returns the set of SourcedIds that have been altered from the defined reference point, supplied by the source, to the latest reference point identified in the target. If the reference counter in the source is greater than that in the target then an error code is returned and the target value for the reference point is returned.
Notes:	If no SourcedIds have been allocated then the returned data set is empty and the success status code returned.

Table 3.12 Status codes for the 'readLineItemIdsFromSavePoint' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified LineItem identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no LineItem identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointerror'	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointsyncerror'	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value has been updated to that supplied by the source system.

3.2.12 ReadLineItems() Operation

Name:	readLineItems
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.13 and A.2.
Supplied (in) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the LineItem objects to be read.
Returned (out) Parameters:	*lineItemRecordSet:LineItemRecordSet* – the set of lineItem records. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readLineItems' request the target is charged with retrieving the identified set of objects from its database. The associated read savePoint reference is updated and returned. If an object identified by the supplied set of SourcedIds cannot be located then the request is rejected and a partial success code is returned for the operation. The target is responsible for ensuring that the records contain valid data. The target should attempt to successfully complete as much of the request as possible.
Notes:	A lineItem object is only present in the data structure if it has been located in the target system. The enclosed data may result in a long response message.

Table 3.13 Status codes for the 'readLineItems' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatareturn'	The target has only returned a subset of the data object e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored LineItem object and so cannot return the data.

3.2.13 ReadLineItemsFromSavePoint() Operation

Name:	readLineItemsFromSavePoint
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the read request. The permitted status codes are given in Tables 3.14 and B.2.
Supplied (in) Parameters:	*fromSavePoint:SequenceIdentifier* – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
Returned (out) Parameters:	*lineItemRecordSet:LineItemRecordSet* – the set of group records. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readLineItemsFromSavePoint' request the target is charged with reading the objects that have been altered from the defined reference point. If the reference counter in the source is greater than that in the target then an error code is returned and the target value for the reference point is returned.
Notes:	If no objects have been allocated then the return message will be empty. The enclosed data may result in a long response message.

Table 3.14 Status codes for the 'readLineItemsFromSavePoint' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified LineItem object has been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointerror'	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointsyncerror'	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value has been updated to that supplied by the source system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatareturn'	The target has only returned a subset of the data object e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored LineItem object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the target system.
'CodeMajor=Failure'	Some mandatory part of the data has been detected as missing by the target

Status Code	Explanation of the Cause of the Code
‘Severity=Status’ ‘CodeMinor=incompletedata’	system.

3.2.14 ReplaceLineItem() Operation

Name:	replaceLineItem
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the replace request. The permitted status codes are given in Tables 3.15 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the LineItem object to be replaced. *lineItemRecord:LineItemRecord* – the LineItem data that is to be stored in the object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'replaceLineItem' request the target is charged with writing the supplied information into the identified record. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is a destructive write-over operation of the entire LineItem object. This is equivalent to a 'createLineItem' but for an object that already exists. If the object identified by the supplied SourcedId cannot be located then the request is interpreted as a 'createLineItem' invocation. The reference counter for the object is incremented by one in the target system.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.15 Status codes for the 'replaceLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The replace request has been fully and successfully implemented by the target system and the identified object has been changed on the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=createsuccess'	The LineItem object identifier is unknown in the target system and so a new object has been successfully created instead.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	An invalid LineItemType value has been supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=contextunknown'	An unknown context is supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=gradingnotpermitted'	The service consumer is not authorized for grade upload for this CourseSection object.
'CodeMajor=Failure' 'Severity=Status'	Some mandatory part of the data has been detected as missing by the target system.

Status Code	Explanation of the Cause of the Code
‘CodeMinor=incompletedata’	
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary extensions that are used in the object’s data model.

3.2.15 UpdateLineItem() Operation

Name:	updateLineItem
Return Function Parameter:	*StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.16 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the LineItem object to be updated. *lineItemRecord:LineItemRecord* – the LineItem data that is to be stored in the object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'updateLineItem' request the target is charged with writing the supplied information into the identified record. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the object is left in its original state. This is an additive write operation of all the data fields supplied in the update request and fields not supplied remain unchanged. If a field is constrained with a multiplicity of one then the 'updateLineItem' request acts as a replaceLineItem' request for that field. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned. The reference counter for the object is incremented by one in the target system.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.16 Status codes for the 'updateLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The update request has been fully and successfully implemented by the target system and the identified object has been changed on the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be updated.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	An invalid LineItemType value has been supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=contextunknown'	An unknown context is supplied.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=gradingnotpermitted'	The service consumer is not authorized for grade upload for this CourseSection object.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the target system.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=unknownextension’	The source cannot process the proprietary extensions that are used in the object’s data model.

3.2.16 DiscoverLineItemIds() Operation

Name:	discoverLineItemIds
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.17 and A.2.
Supplied (in) Parameters:	*queryObject:QueryObject* – this is the query/filter instruction that is to be applied by the target to discover the corresponding LineItem objects.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the LineItem objects whose content conform to the query conditions.
Behavior:	When the source issues the 'discoverLineItemIds' the target applies the query instructions to the set of LineItem objects and returns the set of SourcedIds that uphold the query. If no LineItem objects have the required properties the returned data set is empty and the success status code returned. If the target does not understand or cannot apply the requested query then an error status is returned.
Notes:	The internal structure of this QueryObject is undefined (it is should be treated as a String). Later versions of this specification will look at the established best practices for clarification on the use of this operation.

Table 3.17 Status codes for the 'discoverLineItemIds' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The discover request has been fully and successfully implemented by the target system and the identifiers of the objects that uphold the search criteria are returned.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The discover request has been fully and successfully implemented by the target system and no LineItem object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownquery'	The query/filter instruction is unknown in the target system and so the query/filter could not be executed.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.2.17 ChangeLineItemIdentifier() Operation

Name:	changeLineItemIdentifier
Return Function Parameter:	*StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.18 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the LineItem object to be changed. *newSourcedId:GUID* – the new identifier to be allocated to the 'lineItem' object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'changeLineItemIdentifier' request the target is charged with replacing the original SourcedId with the new supplied SourcedId. All Membership entries must be similarly changed. All further references to the object must use the new 'sourcedId' otherwise an 'unknown' object failure status code is returned. If the object identified by the supplied 'SourcedId' cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The reference pointer value remains unchanged.

Table 3.18 Status codes for the 'changeLineItemIdentifier' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The change identifier request has been fully and successfully implemented by the target system and the LineItem object SourcedId has been changed on the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocinusefail'	The target could not allocate the new unique SourcedId to the LineItem object as the identifier is already in use.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The current LineItem SourcedId identifier is unknown in the target system and so the object identifier could not be changed.

3.3 ResultManager Interface Description

The ResultManager interface class describes the operations that are permitted on Result objects (as shown Figure 3.2). The set of operations are summarized in Table 3.19.

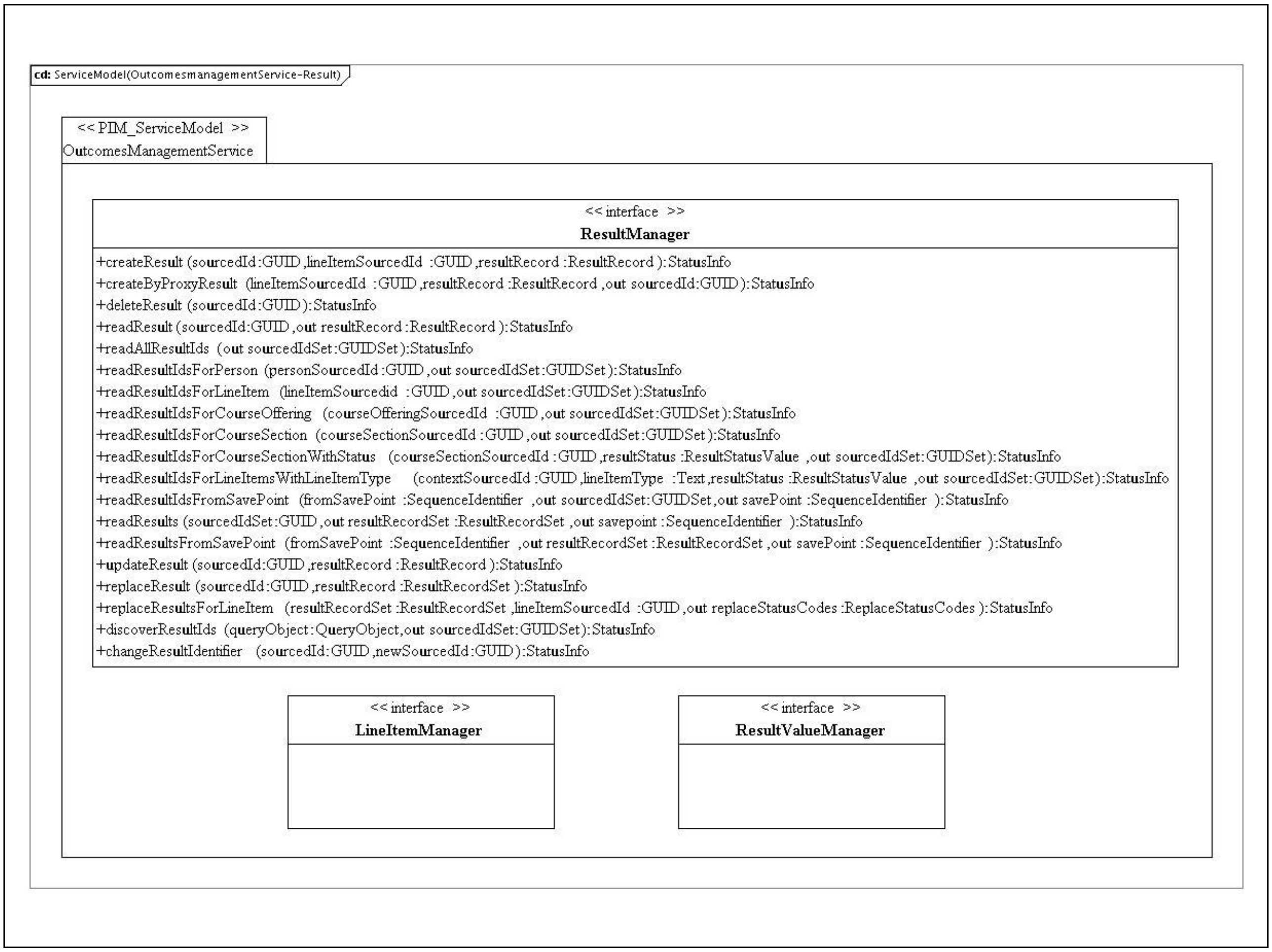

cd: ServiceModel(OutcomesmanagementService-Result)
<< PIM_ServiceModel >>
OutcomesManagementService
<< interface >>
ResultManager
+createResult (sourcedId:GUID ,lineItemSourcedId :GUID ,resultRecord :ResultRecord):StatusInfo
+createByProxyResult (lineItemSourcedId :GUID ,resultRecord :ResultRecord ,out sourcedId:GUID):StatusInfo
+deleteResult (sourcedId:GUID):StatusInfo
+readResult (sourcedId:GUID ,out resultRecord :ResultRecord):StatusInfo
+readAllResultIds (out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsForPerson (personSourcedId :GUID ,out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsForLineItem (lineItemSourcedid :GUID ,out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsForCourseOffering (courseOfferingSourcedId :GUID ,out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsForCourseSection (courseSectionSourcedId :GUID ,out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsForCourseSectionWithStatus (courseSectionSourcedId :GUID ,resultStatus :ResultStatusValue ,out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsForLineItemsWithLineItemType (contextSourcedId :GUID ,lineItemType :Text ,resultStatus :ResultStatusValue ,out sourcedIdSet:GUIDSet):StatusInfo
+readResultIdsFromSavePoint (fromSavePoint :SequenceIdentifier ,out sourcedIdSet:GUIDSet ,out savePoint :SequenceIdentifier):StatusInfo
+readResults (sourcedIdSet:GUID ,out resultRecordSet :ResultRecordSet ,out savepoint :SequenceIdentifier):StatusInfo
+readResultsFromSavePoint (fromSavePoint :SequenceIdentifier ,out resultRecordSet :ResultRecordSet ,out savePoint :SequenceIdentifier):StatusInfo
+updateResult (sourcedId:GUID ,resultRecord :ResultRecord):StatusInfo
+replaceResult (sourcedId:GUID ,resultRecord :ResultRecordSet):StatusInfo
+replaceResultsForLineItem (resultRecordSet :ResultRecordSet ,lineItemSourcedId :GUID ,out replaceStatusCodes :ReplaceStatusCodes):StatusInfo
+discoverResultIds (queryObject:QueryObject,out sourcedIdSet:GUIDSet):StatusInfo
+changeResultIdentifier (sourcedId:GUID ,newSourcedId:GUID):StatusInfo
<< interface >>
LineItemManager
<< interface >>
ResultValueManager

Figure 3.2 OutcomesManagementService ResultManager interface definitions.

Table 3.19 Summary of operations for ResultManager.

Operation	Description
createResult	To request the creation of a populated Result object on the target system where the source allocates the unique SourcedId. The Result object is tied to a LineItem object.
createByProxyResult	To request the creation of a populated Result object on the target system where the target allocates the unique SourcedId. The Result object is tied to a LineItem object.
deleteResult	To request the deletion of a Result object. The Result object is deleted but the associated ResultValue and LineItems remain.
readResult	To read the full contents of the identified Result object. The target must return all of the data it has for the identified Result object.
readAllResultIds	To obtain the set of SourcedIds assigned to Result objects.
readResultIdsForPerson	To obtain the set of SourcedIds for the Result objects which are associated with a particular Person object.
readResultIdsForLineItem	To obtain the set of SourcedIds for the Result objects which are associated with a particular LineItem object.
readResultIdsForCourseOffering	To obtain the set of SourcedIds for the Result objects which are associated with a particular CourseOffering object.
readResultIdsForCourseSection	To obtain the set of SourcedIds for the Result objects which are associated with a particular CourseSection object.
readResultIdsForCourseSectionWithStatus	To obtain the set of SourcedIds for the Result objects that are associated with a particular CourseSection object and have results of a particular status.
readResultIdsForLineItemWithLineItemType	To obtain the SourcedIds for the Result objects that are associated with a LineItem that has a particular LineItemType state.
readResultIdsFromSavePoint	To obtain the set of identifiers for Result objects which have been altered since the requested reference point. The reference point is set as ‘zero’ at creation and incremented after every write operation.
readResults	To obtain the Result objects for a defined set of identifiers. This results in a single transaction that may require the exchange of a large volume of data.
readResultsFromSavePoint	To obtain the set of Result objects which have been altered since the requested reference point. The reference point is set as ‘zero’ at creation and incremented after every write operation. This results in a single transaction that may require the exchange of a large volume of data in the response message.
updateResult	To add content to the identified Result object. The target must write the new data into the Result object. This is an additive operation.
replaceResult	To replace the content of the identified Result object. The target must write the new data into the Result object. This is a destructive write-over of all of the original information. In the case of the object not existing, this operation acts as an implied ‘createResult’.
replaceResultsForLineItem	To replace the Result records for all of the PersonSourcedIds for a specific LIneItem object. In the case of the object not existing, this

	operation acts as an implied ‘createResult’.
discoverResultIds	To obtain the set of SourcedIds for Result objects whose properties agree with those defined in the query/filter.
changeResultIdentifier	To change the SourcedId of a Result object.

Note: In most cases the above operations act on a single instance of a Group object i.e., ‘createResult’, ‘createByProxyResult’, ‘deleteResult’, ‘readResult’, ‘replaceResult’ and ‘updateResult’.

3.3.1 CreateResult() Operation

Name:	createResult
Return Function Parameter:	*StatusInfo* – the status of the creation request. The permitted status codes are defined in Tables 3.20 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the sourcedId allocated by the source system. This is the identifier that must also be assigned within the target system. *lineItemSourcedId:GUID* – the sourcedId of the LineItem object to which the Result object is assigned. *resultRecord:ResultRecord* – the Result data that is to be stored in the new object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'createResult' request the target is instructed to create the populated Result object and to allocate that structure the SourcedId passed by the source. If the supplied SourcedId has already been allocated to another object then the request is rejected and the appropriate failure code is returned. The reference point identifier is set to 'zero' for the Result object in both the source and target. The Result object is assigned to the identified LineItem object. If an invalid sourcedId is given for the LineItem object then an error code is returned and the Result object is not created.
Notes:	This request contains the initial content for the Result object. More content can be added/replaced using the 'updateResult' and/or 'replaceResult' requests.

Table 3.20 Transitions associated with the 'createResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The creation request has been fully and successfully implemented by the target system and the object has been created with the supplied SourcedId.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocinusefail'	The target could not allocate the unique SourcedId to the object as the identifier is already in use.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The LineItem object identifier is unknown in the target system and so the Result object could not be created.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=overflowfail'	The target could not create the Result record due to lack of target allocation memory.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status'	Some mandatory part of the data has been detected as missing by the target system.

‘CodeMinor=incompletedata’	
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatastorage’	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary extensions that are used in the object’s data model.

3.3.2 CreateByProxyResult() Operation

Name:	createByProxyResult
Return Function Parameter:	*StatusInfo* – the status of the creation request. The permitted status codes are defined in Tables 3.21 and A.2.
Supplied (in) Parameters:	*lineItemSourcedId:GUID* – the sourcedId of the LineItem object to which the Result object is assigned. *resultrecord:ResultRecord* – the Result data that is to be stored in the new object.
Returned (out) Parameters:	*sourcedId:GUID* – the identifier allocated by the target to the newly created Result object.
Behavior:	When the source issues the 'createByProxyResult' request the target is instructed to create the populated Result object and to allocate that record a unique SourcedId. The reference point identifier is set to 'zero' for the Result object in both the source and target. The Result object is assigned to the identified LineItem object. If an invalid sourcedId is given for the LineItem object then an error code is returned and the Result object is not created.
Notes:	This request contains the initial content for the Result object. More content can be added/replaced using the 'updateResult' and/or 'replaceResult' requests.

Table 3.21 Status codes for the 'createByProxyResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The creation request has been fully and successfully implemented by the target system and the object has been created with the SourcedId supplied by the target.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocfail'	The target could not allocate a unique SourcedId to the object because there are no more spare identifiers available.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The LineItem object identifier is unknown in the target system and so the Result object could not be created.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=overflowfail'	The target could not create the object due to lack of target allocation memory.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied

‘CodeMinor=partialdatastorage’	and stored).
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary extensions that are used in the object’s data model.

3.3.3 DeleteResult() Operation

Name:	deleteResult
Return Function Parameter:	*StatusInfo* – the status of the deletion request. The permitted status codes are defined in Tables 3.22 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier to be used by the target to identify the Result object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'deleteResult' request the target is instructed to delete the identified Result object and to remove any associations. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	Deletion of the Result object does not necessarily result in the destruction of the data within the server. The true state of the data in the target is unknown.

Table 3.22 Status codes for the 'deleteResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The deletion request has been fully and successfully implemented by the target system and the object has been deleted.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be deleted.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor= deletefailure'	The target system has not been able to delete the identified object.

3.3.4 ReadResult() Operation

Name:	readResult
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.23 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the Result object to be read.
Returned (out) Parameters:	*resultRecord:ResultRecord*– the Result data that is read from the object.
Behavior:	When the source issues the 'readResult' request the target is charged with retrieving the identified record from its database and returning this data to the source. The target is responsible for ensuring that the object contains valid data. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The returned Result object can only be trusted if the corresponding status code is 'success'.

Table 3.23 Status codes for the 'readResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the source system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has only returned a subset of the data expected by the source e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownextension'	The source cannot process the proprietary extensions that are used in the object's data model.

3.3.5 ReadAllResultIds() Operation

Name:	readAllResultIds
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.24 and A.2.
Supplied (in) Parameters:	None.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects in the target's Result database.
Behavior:	When the source issues the 'readAllResultIds' request the target is charged with returning the SourcedIds of all of the Result identifiers in its Result database.
Notes:	The returned set of SourcedIds may or may not have completed results. If no SourcedIds have been allocated then the returned data set is empty and the success status code returned. Note that, in general, the set of Result objects will not be assigned to the same LineItem.

Table 3.24 Status codes for the 'readAllResultIds' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.3.6 ReadResultIdsForPerson() Operation

Name:	readResultIdsForPerson
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.25 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the Person object whose Result sourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects for the Person in the target's Result database.
Behavior:	The target system must return all of the Result SourcedIds for the identified Person object. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.25 Status codes for the 'readResultIdsForPerson' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The Person object identifier is unknown in the target system and so the associated Result could not be identified.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the Outcomes database and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.3.7 ReadResultIdsForLineItem() Operation

Name:	readResultIdsForLineItem
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.26 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the LineItem object whose Result sourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects for the LineItem in the target's Outcomes database.
Behavior:	The target system must return all of the Result SourcedIds for the identified LineItem object. The Result identifiers are read directly from the LineItem data model. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The returned object can only be trusted if the corresponding status code is 'success'.

Table 3.26 Status codes for the 'readResultIdsForLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The LineItem object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the LineItem object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.3.8 ReadResultIdsForCourseOffering() Operation

Name:	readResultIdsForCourseOffering
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.27 and A.2.
Supplied (in) Parameters:	*offeringSourcedId:GUID* – the identifier of the CourseOffering object whose Result SourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects for the CourseOffering in the target's Course database.
Behavior:	The target system must return all of the Result SourcedIds for the identified CourseOffering object. This information is located by finding the LineItems that are associated with the identified CourseOffering. The corresponding Result identifiers listed in the LineItems are then returned. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.27 Status codes for the 'readResultIdsForCourseOffering' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The CourseOffering object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the object databases and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.3.9 ReadResultIdsForCourseSection() Operation

Name:	readResultIdsForCourseSection
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.28 and A.2.
Supplied (in) Parameters:	*sectionSourcedId:GUID* – the identifier of the CourseSection object whose Result SourcedIds are required.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects for the CourseSection in the target's Course database.
Behavior:	The target system must return all of the Result SourcedIds for the identified CourseSection object. This information is located by finding the LineItems that are associated with the identified CourseSection. The corresponding Result identifiers listed in the LineItems are then returned. If the CourseSection object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.28 Status codes for the 'readResultIdsForCourseSection' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The CourseSection object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the object databases and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.3.10 ReadResultIdsForCourseSectionWthStatus() Operation

Name:	readResultIdsForCourseSectionWithStatus
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.29 and A.2.
Supplied (in) Parameters:	*sectionSourcedId:GUID* – the identifier of the CourseSection object whose Result SourcedIds are required. *resultStatus:ResultStatus* – the status of the Result of the SourcedIds required. This will be from an agreed vocabulary
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects for the CourseSection with the required status in the target's CourseSection database.
Behavior:	The target system must return all of the Result SourcedIds, of the appropriate status, for the identified CourseSection object. This information is located by finding the LineItems that are associated with the identified CourseSection. The corresponding Result identifiers listed in the LineItems with the corresponding status are then returned. If the CourseSection object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.29 Status codes for the 'readResultIdsForCourseSectionWithStatus' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The CourseSection object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the object database and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	The defined status for the Result objects is unknown in the target system.

3.3.11 ReadResultIdsForLineItemWithLineItemType() Operation

Name:	readResultIdsForLineItemWithLineItemType
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.30 and A.2.
Supplied (in) Parameters:	*contextSourcedId:GUID* – the identifier of the LineItem object whose Result SourcedIds are required. *lineItemType:LineItemType* – the type of the LineItems required for inspection. This will be from an agreed vocabulary. *resultStatus:ResultStatus* – the status of the Result of the SourcedIds required. This will be from an agreed vocabulary.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the Result objects for the LineItem with the required LineItemType in the target's Outcomes database.
Behavior:	The target system must return all of the Result SourcedIds for the identified LineItem object with the relevant LineItemType. This information is located by finding the LineItems that are associated with the identified. The corresponding Result identifiers listed in the LineItems with the corresponding LineItemType are then returned. If the LineItem object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.30 Status codes for the 'readResultIdsForLineItemWithLineItemType' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The LineItem object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the object database and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidlineitemtype'	The defined LineItemType for the LineItem object is unknown in the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.

‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	The defined ResultStatus for the LineItem object is unknown in the target system.

3.3.12 ReadResultIdsFromSavePoint() Operation

Name:	readResultIdsFromSavePoint
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the read request. The permitted status codes are given in Tables 3.31 and A.2.
Supplied (in) Parameters:	*fromSavePoint:SequenceIdentifier* – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the Result objects stored on the target. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the ‘readResultIdsFromSavePoint’ the target returns the set of SourcedIds that have been altered from the defined reference point, supplied by the source, to the latest reference point identified in the target. If the reference counter in the source is greater than that in the target then an empty set is returned for the SourcedIds and the target value for the reference point is returned.
Notes:	If no SourcedIds have been allocated then the returned data set is empty and the success status code returned.

Table 3.31 Status codes for the ‘readResultIdsFromSavePoint’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the identified Result object identifiers have been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor= nosourcedids’	The read request has been fully and successfully implemented by the target system and no Result object identifiers were found.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Part or all of the returned data was detected as invalid by the source system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointerror’	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointsyncerror’	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value has been updated to that supplied by the source system.

3.3.13 ReadResults() Operation

Name:	readResults
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.32 and A.2.
Supplied (in) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the Result objects to be read.
Returned (out) Parameters:	*resultRecordSet:ResultRecordSet* – the set of result records. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readResults' request the target is charged with retrieving the identified set of objects from its database. The associated read savePoint reference is updated and returned. If the object identified by the supplied SourcedId cannot be located then the request is rejected and a partial success code is returned for the operation. The target is responsible for ensuring that the records contain valid data. The target should attempt to successfully complete as much of the request as possible.
Notes:	A result object is only present in the data structure if it has been located in the target system. The enclosed data may result in a long response message.

Table 3.32 Status codes for the 'readResults' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatareturn'	The target has only returned a subset of the data object e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored Result object and so cannot return the data.

3.3.14 ReadResultsFromSavePoint() Operation

Name:	readLineItemsFromSavePoint
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the read request. The permitted status codes are given in Tables 3.33 and A.2.
Supplied (in) Parameters:	*fromSavePoint:SequenceIdentifier* – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
Returned (out) Parameters:	*resultRecordSet:ResultRecordSet* – the set of result records. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readResultsFromSavePoint' request the target is charged with reading the objects that have been altered from the defined reference point. If the reference counter in the source is greater than that in the target then an error code is returned and the target value for the reference point is returned.
Notes:	If no objects have been allocated then the return message will be empty. The enclosed data may result in a long response message.

Table 3.33 Status codes for the 'readResultsFromSavePoint' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified Result object has been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=partialreadfail'	Some of the Result object identifiers are unknown in the target system and so those objects could not be read.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatareturn'	The target has only returned a subset of the data object e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointerror'	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored Result object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the target system.
'CodeMajor=Failure'	Some mandatory part of the data has been detected as missing by the target

Status Code	Explanation of the Cause of the Code
‘Severity=Status’ ‘CodeMinor=incompletedata’	system.

3.3.15 ReplaceResult() Operation

Name:	replaceResult
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the replace request. The permitted status codes are given in Tables 3.34 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the Result object to be replaced. *resultRecord:ResultRecord* – the Result data that is to be stored in the object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'replaceResult' request the target is charged with writing the supplied information into the identified record. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is a destructive write-over operation of the entire Result object. This is equivalent to a 'createResult' but for an object that already exists. If the object identified by the supplied SourcedId cannot be located then the request is interpreted as a 'createResult' invocation. The reference counter for the object is incremented by one in the target system.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.34 Status codes for the 'replaceResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The replace request has been fully and successfully implemented by the target system and the identified object has been changed on the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=createsuccess'	The Result object identifier is unknown in the target system and so a new object has been successfully created instead.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownmdvocabulary'	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.

‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary data model extensions used in the object.

3.3.16 ReplaceResultsForLineItem() Operation

Name:	replaceResultsForLineItem
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the replace request. The permitted status codes are given in Tables 3.35 and A.2.
Supplied (in) Parameters:	*resultRecordSet:ResltRecordSet* – the set of result records that are to replace the current Result objects; *lineItemSourcedId:GUID* – the identifier of the LineItem object for which the results are to be replaced. *replaceStatusCodes:ReplaceStatusCodes* – the list of status codes returned for the each of the replacement requests in the transaction.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'replaceResultsForLineItem' request the target is charged with writing the supplied information into the identified result records for the identified LineItem object. Each Result object replace is treated as an independent request and the corresponding status code is returned in the response message body (see Table B1.3 for the permitted values). This is a destructive write-over operation of the entire Result object. This is equivalent to a 'createResult' but for an object that already exists. If an identified Result object cannot be located then the request is interpreted as a 'createResult' invocation. The reference counter for each object is incremented by one in the target system. An overall status code is returned for the request as a whole.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.35 Status codes for the 'replaceResultsForLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The replace request has been fully and successfully implemented by the target system and the identified objects have been changed on the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=unknownobject'	The identified LineItem object is unknown i.e., the supplied sourcedId is not assigned to a Lineitem object.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Some of the request has failed due to one or more data validation failures. See the specific request codes (given in the *replaceStatusCode* parameter) for further information for the cause(s) of the failure.

3.3.17 UpdateResult() Operation

Name:	updateResult
Return Function Parameter:	*StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.36 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the Result object to be updated. *resultRecord:ResultRecord* – the Result data that is to be stored in the object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'updateResult' request the target is charged with writing the supplied information into the identified record. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is an additive write operation of all the data fields supplied in the update request and fields not supplied remain unchanged. If a field is constrained with a multiplicity of one then the 'updateResult' request acts as a replaceResult' request for that field. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned. The reference counter for the object is incremented by one in the target system.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.36 Status codes for the 'updateResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The update request has been fully and successfully implemented by the target system and the identified object has been changed on the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be updated.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status'	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.

‘CodeMinor=unknownmdvocabulary’	
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownextension’	The target cannot process the proprietary data model extensions used in the object.

3.3.18 DiscoverResultIds() Operation

Name:	discoverResultIds
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the discover request. The permitted status codes are given in Tables 3.37 and A.2.
Supplied (in) Parameters:	*queryObject:QueryObject* – this is the query/filter instruction that is to be applied by the target to discover the corresponding Result objects.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the Result objects whose content conform to the query conditions.
Behavior:	When the source issues the 'discoverResultIds' the target applies the query instructions to the set of Result objects and returns the set of SourcedIds that uphold the query/filter. If no Result objects have the required properties the returned data set is empty and the success status code returned. If the target does not understand or cannot apply the requested query/filter then an error status is returned.
Notes:	The internal structure of this QueryObject is undefined (it is should be treated as a String). Later versions of this specification will look at the established best practices for clarification on the use of this operation.

Table 3.37 Status codes for the 'discoverResultIds' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The discover request has been fully and successfully implemented by the target system and the object identifiers for the Results that uphold the search criteria.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The discover request has been fully and successfully implemented by the target system and no LineItem object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownquery'	The query/filter instruction is unknown in the target system and so the query/filter could not be executed.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.3.19 ChangeResultIdentifier() Operation

Name:	changeResultIdentifier
Return Function Parameter:	*StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.38 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the Result object to be changed. *newSourcedId:GUID* – the new identifier to be allocated to the Result object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'changeResultIdentifier' request the target is charged with replacing the original SourcedId with the new supplied SourcedId. All further references to the object must use the new SourcedId otherwise an 'unknown' object failure status code is returned. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The reference pointer value remains unchanged.

Table 3.38 Status codes for the 'changeResultIdentifier' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The change identifier request has been fully and successfully implemented by the target system and the Result object SourcedId has been changed on the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocinusefail'	The target could not allocate the new unique 'identifier' to the Result object as the identifier is already in use.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The current Result SourcedId identifier is unknown in the target system and so the object identifier could not be changed.

3.4 ResultValueManager Interface Description

The ResultValueManager interface class describes the operations that are permitted on ResultValue objects (as shown Figure 3.3). The set of operations are summarized in Table 3.39.

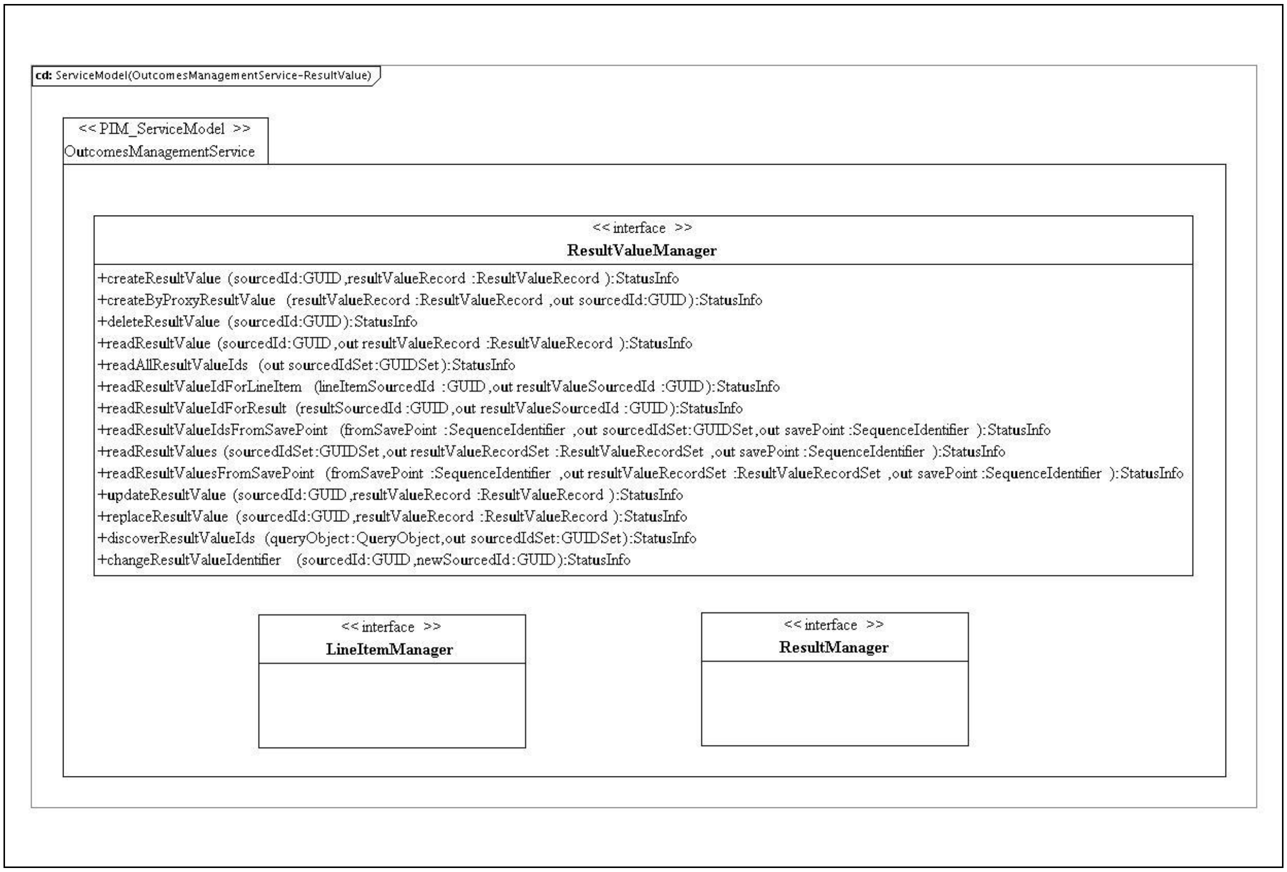

cd: ServiceModel(OutcomesManagementService-ResultValue)
<< PIM_ServiceModel >>
OutcomesManagementService
<< interface >>
ResultValueManager
+createResultValue (sourcedId:GUID ,resultValueRecord :ResultValueRecord):StatusInfo
+createByProxyResultValue (resultValueRecord :ResultValueRecord ,out sourcedId:GUID):StatusInfo
+deleteResultValue (sourcedId:GUID):StatusInfo
+readResultValue (sourcedId:GUID ,out resultValueRecord :ResultValueRecord):StatusInfo
+readAllResultValueIds (out sourcedIdSet:GUIDSet):StatusInfo
+readResultValueIdForLineItem (lineItemSourcedId :GUID ,out resultValueSourcedId :GUID):StatusInfo
+readResultValueIdForResult (resultSourcedId :GUID ,out resultValueSourcedId :GUID):StatusInfo
+readResultValueIdsFromSavePoint (fromSavePoint :SequenceIdentifier ,out sourcedIdSet:GUIDSet ,out savePoint :SequenceIdentifier):StatusInfo
+readResultValues (sourcedIdSet:GUIDSet ,out resultValueRecordSet :ResultValueRecordSet ,out savePoint :SequenceIdentifier):StatusInfo
+readResultValuesFromSavePoint (fromSavePoint :SequenceIdentifier ,out resultValueRecordSet :ResultValueRecordSet ,out savePoint :SequenceIdentifier):StatusInfo
+updateResultValue (sourcedId:GUID ,resultValueRecord :ResultValueRecord):StatusInfo
+replaceResultValue (sourcedId:GUID ,resultValueRecord :ResultValueRecord):StatusInfo
+discoverResultValueIds (queryObject:QueryObject,out sourcedIdSet:GUIDSet):StatusInfo
+changeResultValueIdentifier (sourcedId:GUID ,newSourcedId:GUID):StatusInfo
<< interface >>
LineItemManager
<< interface >>
ResultManager

Figure 3.3 OutcomesManagementService ResultValueManager interface definitions.

Table 3.39 Summary of operations for ResultValueManager.

Operation	Description
createResultValue	To request the creation of a populated ResultValue object on the target system where the source is responsible for the allocation of the unique SourcedId.
createByProxyResultValue	To request the creation of a populated ResultValue object on the target system where the target is responsible for the allocation of the unique SourcedId.
deleteResultValue	To request the deletion of a ResultValue object. The ResultValue object is deleted along with all of its associated relationships.
readResultValue	To read the full contents of the identified ResultValue object. The target must return all of the data it has for the identified ResultValue object.
readAllResultValueIds	To obtain the set of SourcedIds which have been assigned to ResultValue objects.
readResultValueIdForLineItem	To obtain the SourcedId for the ResultValue object which has been associated with a particular LineItem object.
readResultValueIdForResult	To obtain the SourcedId for the ResultValue object which is associated with a particular Result object.
readResultValueIdsFromSavePoint	To obtain the set of identifiers for ResultValue objects which have been altered since the requested reference point. The reference point is set as ‘zero’ at creation and incremented after every write operation.
readResultValues	To obtain the ResultValue objects for a defined set of identifiers. This results in a single transaction that may require the exchange of a large volume of data in the response message.
readResultValuesFromSavePoint	To obtain the set of ResultValue objects which have been altered since the requested reference point. The reference point is set as ‘zero’ at creation and incremented after every write operation. This results in a single transaction that may require the exchange of a large volume of data in the response message.
updateResultValue	To write new content into the identified ResultValue object. The target must write the new data into the ResultValue object. This is an additive operation.
replaceResultValue	To replace the content of the identified ResultValue object. The target must write the new data into the ResultValue object. This is a destructive write-over of all of the original information. In the case of the object not existing, this operation acts as an implied ‘createResultValue’.
discoverResultValueIds	To obtain the set of SourcedIds for ResultValue objects whose properties agree with those defined in the query/filter.
changeResultValueIdentifier	To change the SourcedId for a ResultValue object.

Note: In most cases the above operations act on a single instance of a Group object i.e., ‘createResultValue’, ‘createByProxyResultValue’, ‘deleteResultValue’, ‘readResultValue’, ‘replaceResultValue’ and ‘updateResultValue’.

3.4.1 CreateResultValue() Operation

Name:	createResultValue
Return Function Parameter:	*StatusInfo* – the status of the creation request. The permitted status codes are defined in Table 3.40 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the sourcedId allocated by the source system. This is the identifier that must also be assigned within the target system. *resultValueRecord:ResultValueRecord* – the ResultValue data that is to be stored in the new object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'createResultValue' request the target is instructed to create the populated ResultValue object and to allocate that structure the SourcedId passed by the source. If the supplied SourcedId has already been allocated to another object then the request is rejected and the appropriate failure code is returned. The reference point identifier is set to 'zero' for the ResultValue object in both the source and target.
Notes:	This request contains the initial content for the ResultValue object. More content can be added/replaced using the 'updateResultValue' and/or 'replaceResultValue' requests.

Table 3.40 Status codes for the 'createResultValue' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The creation request has been fully and successfully implemented by the target system and the object has been created with the supplied SourcedId.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocinusefail'	The target could not allocate the unique SourcedId to the object as the identifier is already in use.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=overflowfail'	The target could not create the ResultValue object due to lack of target allocation memory.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownmdvocabulary'	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure'	The target system could not identify the defined vocabulary term. This

‘Severity=Status’ ‘CodeMinor=unknownvocabulary’	may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownextension’	The source cannot process the proprietary data model extensions used in the object.

3.4.2 CreateByProxyResultValue() Operation

Name:	createByProxyResultValue
Return Function Parameter:	*StatusInfo* – the status of the creation request. The permitted status codes are defined in Tables 3.41 and A.2.
Supplied (in) Parameters:	*resultValueRecord:ResultValueRecord* – the ResultValue data that is to be stored in the new object.
Returned (out) Parameters:	*sourcedId:GUID* – the identifier allocated by the target to the newly created ResultValue object.
Behavior:	When the source issues the 'createByProxyResultValue' request the target is instructed to create the populated ResultValue object and to allocate that record a unique SourcedId. The reference point identifier is set to zero for the ResultValue object in both the source and target.
Notes:	This request contains the initial content for the ResultValue object. More content can be added/replaced using the 'updateResultValue' and/or 'replaceResultValue' requests.

Table 3.41 'createByProxyResultValue' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The creation request has been fully and successfully implemented by the target system and the object has been created with the SourcedId supplied by the target.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocfail'	The target could not allocate a unique SourcedId to the object because there are no more spare identifiers available.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=overflowfail'	The target could not create the object due to lack of target allocation memory.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the supplied data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownmdvocabulary'	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure'	The source cannot process the proprietary data model extensions used in

‘Severity=Status’ ‘CodeMinor=unknownextension’	the object.

3.4.3 DeleteResultValue() Operation

Name:	deleteResultValue
Return Function Parameter:	*StatusInfo* – the status of the deletion request. The permitted status codes are defined in Tables 3.42 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier to be used by the target to identify the ResultValue object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'deleteResultValue' request the target is instructed to delete the identified ResultValue object and to remove any associations. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	Deletion of the ResultValue record does not necessarily result in the destruction of the data within the server. The true state of the data in the target is unknown.

Table 3.42 Status codes for the 'deleteResultValue' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The deletion request has been fully and successfully implemented by the target system and the object has been deleted.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be deleted.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor= deletefailure'	The target system has not been able to delete the identified object.

3.4.4 ReadResultValue() Operation

Name:	readResultValue
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.43 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the ResultValue object to be read.
Returned (out) Parameters:	*resultValueRecord:ResultValueRecord* – the ResultValue data that is read from the object.
Behavior:	When the source issues the 'readResultValue' request the target is charged with retrieving the identified record from its database and returning this data to the source. The target is responsible for ensuring that the object contains valid data. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The returned ResultValue object can only be trusted if the corresponding status code is 'success'.

Table 3.43 Status codes for the 'readResultValue' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified object has been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the source system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has only returned a subset of the data expected by the source e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownextension'	The source cannot process the proprietary data model extensions used in the object.

3.4.5 ReadAllResultValueIds() Operation

Name:	readAllResultValueIds
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.44 and A.2.
Supplied (in) Parameters:	None.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of SourcedIds for all of the ResultValue objects in the target's ResultValue database.
Behavior:	When the source issues the 'readAllResultValueIds' request the target is charged with returning the SourcedIds of all of the ResultValue in the Outcomes database.
Notes:	If no SourcedIds have been allocated then the returned data set is empty and the success status code returned.

Table 3.44 Status codes for the 'readAllResultValueIds' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no ResultValue object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.4.6 ReadResultValueIdForLineItem() Operation

Name:	readResultValueIdForLineItem
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.45 and A.2.
Supplied (in) Parameters:	*lineItemSourcedId:GUID* – the identifier of the LineItem object whose ResultValue SourcedIds are required.
Returned (out) Parameters:	*resultValueSourcedId:GUID* – the SourcedId for all of the ResultValue object for the LineItem in the target's Outcomes database.
Behavior:	The target system must return the ResultValue SourcedId for the identified LineItem object. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.45 Status codes for the 'readResultValueIdForLineItem' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no ResultValue object identifier was found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The LineItem object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the LineItem object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.4.7 ReadResultValueIdForResult() Operation

Name:	readResultValueIdForResult
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.46 and A.2.
Supplied (in) Parameters:	*resultSourcedId:GUID* – the identifier of the Result object whose ResultValue sourcedIds are required.
Returned (out) Parameters:	*resultValueSourcedId:GUID* – the SourcedId for the ResultValue object for the Result in the target's Outcomes database.
Behavior:	The target system must return the ResultValue SourcedId for the identified Result object. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	None.

Table 3.46 Status codes for the 'readResultValueIdForResult' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no ResultValue object identifier was found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The Result object identifier is unknown in the target system and so the object could not be read.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the Result object and so cannot return the data.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.4.8 ReadResultValueIdsFromSavePoint() Operation

Name:	readResultValueIdsFromSavePoint
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the read request. The permitted status codes are given in Tables 3.47 and A.2.
Supplied (in) Parameters:	*fromSavePoint:SequenceIdentifier* – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the ResultValue objects stored on the target. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readResultValueIdsFromSavePoint' the target returns the set of SourcedIds that have been altered from the defined reference point, supplied by the source, to the latest reference point identified in the target. If the reference counter in the source is greater than that in the target then an error code and the target value for the reference point are returned.
Notes:	If no SourcedIds have been allocated then the returned data set is empty and the success status code returned.

Table 3.47 Status codes for the 'readResultValueIdsFromSavePoint' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the identified ResultValue object identifiers have been read from the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no ResultValue object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The requested data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointerror'	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=savepointsyncerror'	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value has been updated to that supplied by the source system.

3.4.9 ReadResultValues() Operation

Name:	readResultValues
Return Function Parameter:	*StatusInfo* – the status of the read request. The permitted status codes are defined in Tables 3.48 and A.2.
Supplied (in) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the ResultValue objects to be read.
Returned (out) Parameters:	*resultValueRecordSet:ResultValueRecordSet* – the set of resultValue records. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the 'readResultValues' request the target is charged with retrieving the identified set of objects from its database and placing this information in a persistent data file. The associated read savePoint reference is updated and returned. If the object identified by the supplied SourcedId cannot be located then the request is rejected and a partial success code is returned for the operation. The target is responsible for ensuring that the records contain valid data. The target should attempt to successfully complete as much of the request as possible.
Notes:	A resultValue object is only present in the data structure if it has been located in the target system. The enclosed data may result in a long response message.

Table 3.48 Status codes for the 'readResultValues' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The read request has been fully and successfully implemented by the target system and the object identifiers have been read from the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The source system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatareturn'	The target has only returned a subset of the data object e.g., only the mandatory parts.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=targetreadfailure'	The target system has detected an error in the stored LineItem object and so cannot return the data.

3.4.10 ReadResultValuesFromSavePoint() Operation

Name:	readLineItemsFromSavePoint
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.49 and A.2.
Supplied (in) Parameters:	*fromSavePoint:SequenceIdentifier* – the reference point from which all of the changed identifier actions are to be read. This is the value in the source system.
Returned (out) Parameters:	*resultValueRecordSet:ResultValueRecordSet* – the set of resultValue records. *savePoint:SequenceIdentifier* – the value of the reference point counter in the target system.
Behavior:	When the source issues the ‘readResultValuesFromSavePoint’ request the target is charged with reading the objects that have been altered from the defined reference point. If the reference counter in the source is greater than that in the target then an error code and the target value for the reference point are returned.
Notes:	If no objects have been allocated then the return message will be empty. The enclosed data may result in a long response message.

Table 3.49 Status codes for the ‘readResultsFromSavePoint’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The read request has been fully and successfully implemented by the target system and the identified ResultValue object has been read from the target system.
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=partialreadfail’	Some of the ResultValue object identifiers are unknown in the target system and so those objects could not be read.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointerror’	An error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=savepointsyncerror’	The value of the save point reference from the source was later than that of the target system. No identifiers have been returned. The target system savepoint value has been updated to that supplied by the source system.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=toomuchdata’	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
‘CodeMajor=Success’ ‘Severity=Warning’ ‘CodeMinor=partialdatareturn’	The target has only returned a subset of the data object e.g., only the mandatory parts.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=targetreadfailure’	The target system has detected an error in the stored Result object and so cannot return the data.
‘CodeMajor=Failure’	Part or all of the returned data was detected as invalid by the target system.

Status Code	Explanation of the Cause of the Code
‘Severity=Status’ ‘CodeMinor=invaliddata’	
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the data has been detected as missing by the target system.

3.4.11 ReplaceResultValue() Operation

Name:	replaceResultValue
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the replace request. The permitted status codes are given in Tables 3.50 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the ResultValue object to be replaced. *resultValueRecord:ResultValueRecord* – the ResultValue data that is to be stored in the object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'replaceResultValue' request the target is charged with writing the supplied information into the identified record. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is a destructive write-over operation of the entire 'result' object. This is equivalent to a 'createResultValue' but for an object that already exists. If the object identified by the supplied SourcedId cannot be located then the request is interpreted as a 'createResultValue' invocation. The reference counter for the object is incremented by one in the target system.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.50 Status codes for the 'replaceResultValue' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The replace request has been fully and successfully implemented by the target system and the identified object has been changed on the target system.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=createsuccess'	The ResultValue object identifier is unknown in the target system and so a new object has been successfully created instead.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
'CodeMajor=Failure' 'Severity=Status'	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.

‘CodeMinor=unknownmdvocabulary’	
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownextension’	The source cannot process the proprietary data model extensions used in the object.

3.4.12 UpdateResultValue() Operation

Name:	updateResultValue
Return Function Parameter:	*StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.51 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the ResultValue object to be updated. *resultValueRecord:ResultValueRecord* – the ResultValue data that is to be stored in the object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'updateResultValue' request the target is charged with writing the supplied information into the identified object. If any part of the write fails e.g., due to partial invalid data then the whole request is rejected and the record is left in its original state. This is an additive write operation of all the data fields supplied in the update request and fields not supplied remain unchanged. If a field is constrained with a multiplicity of one then the 'updateResultValue' request acts as a replaceResultValue' request for that field. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned. The reference counter for the object is incremented by one in the target system.
Notes:	The source is responsible for determining the reason of the failure.

Table 3.51 Status codes for the 'updateResultValue' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The update request has been fully and successfully implemented by the target system and the identified object has been changed on the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The object identifier is unknown in the target system and so the object could not be updated.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the data has been detected as missing by the target system.
'CodeMajor=Success' 'Severity=Warning' 'CodeMinor=partialdatastorage'	The target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored).
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownvocabulary'	The target system could not identify the defined vocabulary term. This may be due to an incorrect term or a missing vocabulary file.

‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownmdvocabulary’	The target system could not identify the defined metadata vocabulary term. This may be due to an incorrect term or a missing vocabulary file.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unknownextension’	The source cannot process the proprietary data model extensions used in the object.

3.4.13 DiscoverResultValueIds() Operation

Name:	discoverResultValueIds
Return Function Parameter:	*statusInfo:StatusInfo* – the status of the discover request. The permitted status codes are given in Tables 3.52 and A.2.
Supplied (in) Parameters:	*queryObject:QueryObject* – this is the query/filter instruction that is to be applied by the target to discover the corresponding ResultValue objects.
Returned (out) Parameters:	*sourcedIdSet:GUIDSet* – the set of identifiers of the ResultValue objects whose content conform to the query conditions.
Behavior:	When the source issues the 'discoverResultValueIds' the target applies the query/filter instructions to the set of ResultValue objects and returns the set of SourcedIds that uphold the query/filter. If no ResultValue objects have the required properties the returned data set is empty and the success status code returned. If the target does not understand or cannot apply the requested query/filter then an error status is returned.
Notes:	The internal structure of this QueryObject is undefined (it is should be treated as a String). Later versions of this specification will look at the established best practices for clarification on the use of this operation.

Table 3.52 Status codes for the 'discoverResultValueIds' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The discover request has been fully and successfully implemented by the target system and the identifiers of the objects that uphold the search criteria have been returned.
'CodeMajor=Success' 'Severity=Status' 'CodeMinor= nosourcedids'	The read request has been fully and successfully implemented by the target system and no ResultValue object identifiers were found.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownquery'	The query/filter instruction is unknown in the target system and so the query could not be executed.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=toomuchdata'	The data cannot be returned because it would exceed some physical system constraint e.g., permitted size of SOAP message.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Part or all of the returned data was detected as invalid by the source system.

3.4.14 ChangeResultValueIdentifier() Operation

Name:	changeResultValueIdentifier
Return Function Parameter:	*StatusInfo* – the status of the update request. The permitted status codes are given in Tables 3.53 and A.2.
Supplied (in) Parameters:	*sourcedId:GUID* – the identifier of the ResultValue object to be changed. *newSourcedId:GUID* – the new identifier to be allocated to the ResultValue object.
Returned (out) Parameters:	None.
Behavior:	When the source issues the 'changeResultValueIdentifier' request the target is charged with replacing the original SourcedId with the new supplied SourcedId. If the object identified by the supplied SourcedId cannot be located then the request is rejected and the appropriate failure code is returned.
Notes:	The reference pointer value remains unchanged.

Table 3.53 Status codes for the 'changeResultValueIdentifier' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The change identifier request has been fully and successfully implemented by the target system and the ResultValue object SourcedId has been changed on the target system.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=idallocinusefail'	The target could not allocate the new unique 'identifier' to the ResultValue object as the identifier is already in use.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unknownobject'	The current ResultValue SourcedId identifier is unknown in the target system and so the object identifier could not be changed.

3.5 State Machines

The permitted state activity on a LineItem, Result and ResultValue object are shown in Figures 3.4, 3.5 and 3.6 respectively. Each state diagram has three states (the arcs are annotated with the operations that are associated with the change of state):

- 'No Object' state – no object exists with a particular sourcedId;
- 'Object with Provider assigned sourcedId' – an object exists with the sourcedId allocated by the Provider system;
- 'Object with Consumer assigned sourcedId' – an object exists with the sourcedId allocated by the Consumer system.

The state machine for a LineItem object is shown in Figure 3.4.

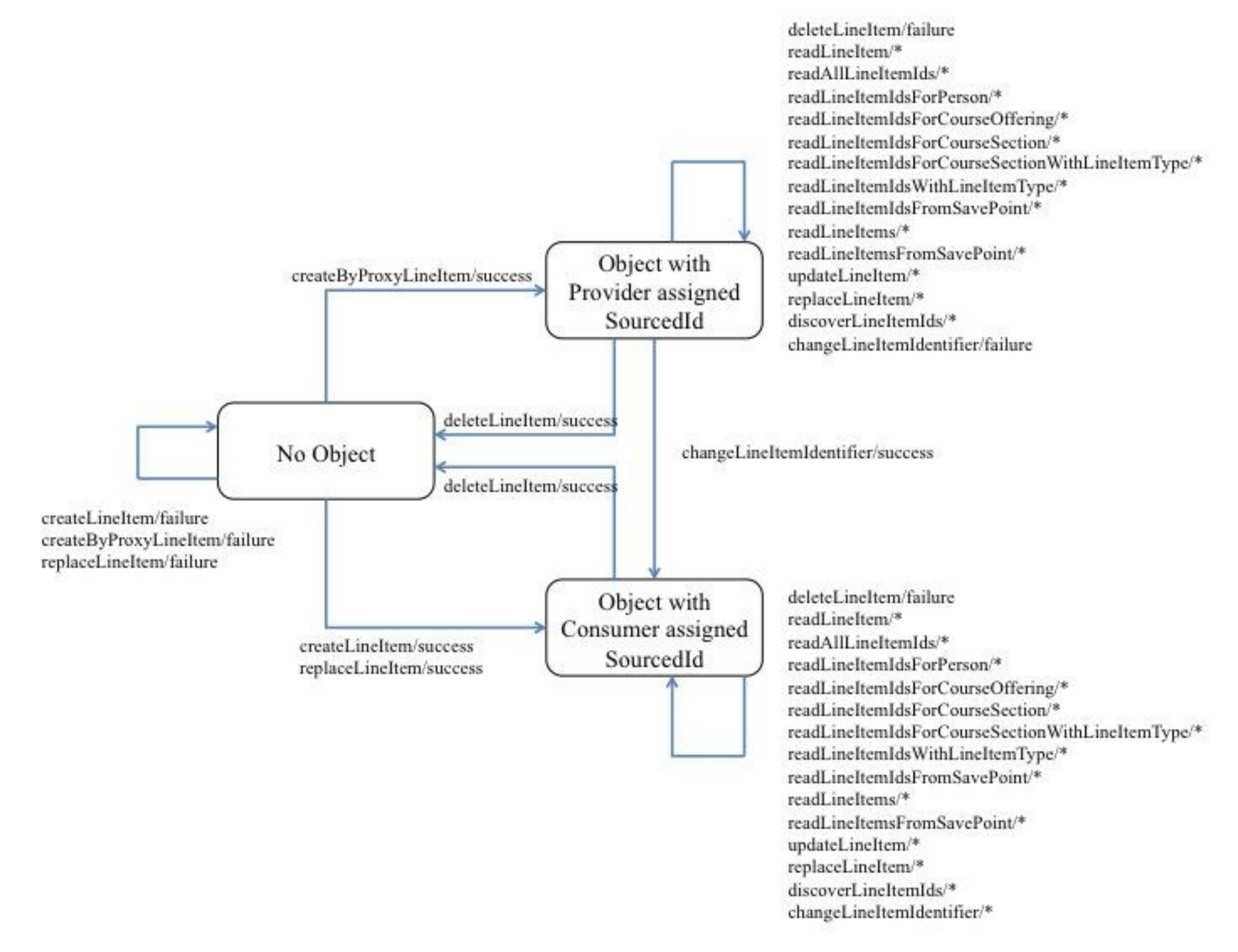

deleteLineItem/failure
readLineItem/*
readAllLineItemIds/*
readLineItemIdsForPerson/*
readLineItemIdsForCourseOffering/*
readLineItemIdsForCourseSection/*
readLineItemIdsForCourseSectionWithLineItemType/*
readLineItemIdsWithLineItemType/*
readLineItemIdsFromSavePoint/*
readLineItems/*
readLineItemsFromSavePoint/*
updateLineItem/*
replaceLineItem/*
discoverLineItemIds/*
changeLineItemIdentifier/failure
createByProxyLineItem/success
Object with Provider assigned SourcedId
deleteLineItem/success
No Object
deleteLineItem/success
changeLineItemIdentifier/success
createLineItem/failure
createByProxyLineItem/failure
replaceLineItem/failure
Object with Consumer assigned SourcedId
createLineItem/success
replaceLineItem/success
deleteLineItem/failure
readLineItem/*
readAllLineItemIds/*
readLineItemIdsForPerson/*
readLineItemIdsForCourseOffering/*
readLineItemIdsForCourseSection/*
readLineItemIdsForCourseSectionWithLineItemType/*
readLineItemIdsWithLineItemType/*
readLineItemIdsFromSavePoint/*
readLineItems/*
readLineItemsFromSavePoint/*
updateLineItem/*
replaceLineItem/*
discoverLineItemIds/*
changeLineItemIdentifier/*

Figure 3.4 State machine for a 'lineItem' object.

The start state is 'No Object' i.e., the LineItem object has not yet been created. Only the 'createLineItem()', 'replaceeLineItem()' and 'createByProxyLineItem()' operations are possible. Once the LineItem object has been created then it persists until a successful 'deleteLineItem()' operation is completed. The 'createLineItem()' and replaceLineItem()' operations take the system into the 'Object with Consumer assigned sourcedId' state whereas the 'createByProxyLineItem()' takes the system into the 'Object with Provider assigned sourcedId' state.

The system can be moved from the ‘Object with Provider assigned sourcedId’ state into the ‘Object with Consumer assigned sourcedId’ state by the successful completion of the ‘changeLineItemIdentifier()’ operation.

Once the system is in the ‘Object with Consumer assigned sourcedId’ or the ‘Object with Provider assigned sourcedId’ states then the ‘readLineItem()’, ‘readAllLineItemIds()’, ‘readLineItemIdsForPerson()’, ‘readLineItemIdsForCourseOffering()’, ‘readLineItemIdsForCourseSection()’, ‘readLineItemIdsForCourseSectionWithLineItemType()’, ‘readLineItemIdsWithLineItemType()’, ‘readLineItemIdsFromSavePoint()’, ‘readLineItems()’, ‘readLineItemsFromSavePoint()’, ‘updateLineItem()’, ‘replaceLineItem()’ and ‘discoverLineItemIds()’ operations are now possible.

This is the state machine for each LineItem object in the Service Consumer and the Service Provider. The binding of the Information Model must guarantee that these two state machines remain synchronized for each LineItem object.

The state machine for a Result object is shown in Figure 3.5.

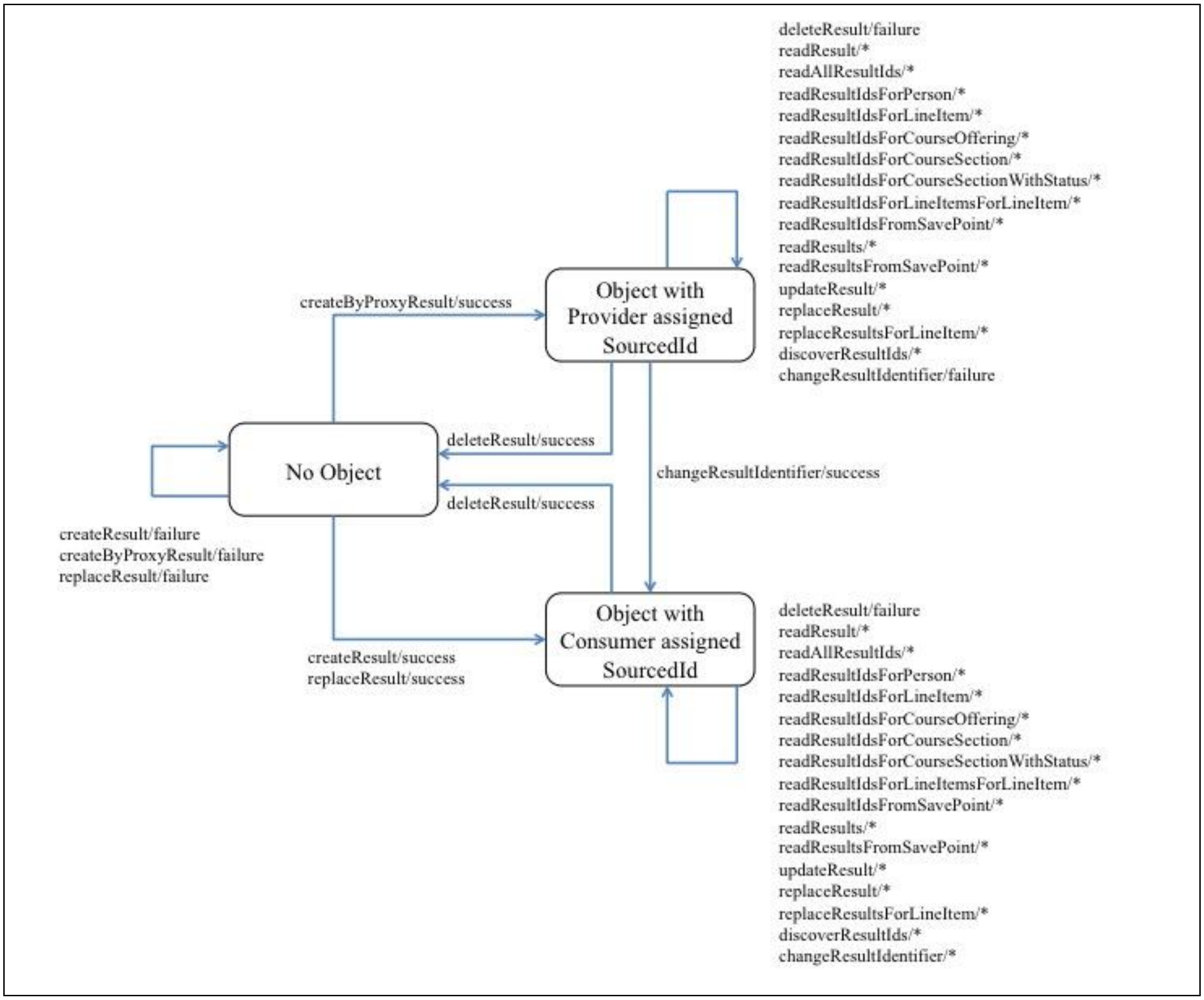

deleteResult/failure
readResult/*
readAllResultIds/*
readResultIdsForPerson/*
readResultIdsForLineItem/*
readResultIdsForCourseOffering/*
readResultIdsForCourseSection/*
readResultIdsForCourseSectionWithStatus/*
readResultIdsForLineItemsForLineItem/*
readResultIdsFromSavePoint/*
readResults/*
readResultsFromSavePoint/*
updateResult/*
replaceResult/*
replaceResultsForLineItem/*
discoverResultIds/*
changeResultIdentifier/failure
createByProxyResult/success
Object with Provider assigned SourcedId
deleteResult/success
No Object
changeResultIdentifier/success
deleteResult/success
createResult/failure
createByProxyResult/failure
replaceResult/failure
Object with Consumer assigned SourcedId
createResult/success
replaceResult/success
deleteResult/failure
readResult/*
readAllResultIds/*
readResultIdsForPerson/*
readResultIdsForLineItem/*
readResultIdsForCourseOffering/*
readResultIdsForCourseSection/*
readResultIdsForCourseSectionWithStatus/*
readResultIdsForLineItemsForLineItem/*
readResultIdsFromSavePoint/*
readResults/*
readResultsFromSavePoint/*
updateResult/*
replaceResult/*
replaceResultsForLineItem/*
discoverResultIds/*
changeResultIdentifier/*

Figure 3.5 State machine for a ‘result’ object.

The start state is ‘No Object’ i.e., the Result object has not yet been created. Only the ‘createResult()’, ‘replaceResult()’ and ‘createByProxyResult()’ operations are possible. Once the Result object has been created then it persists until a successful ‘deleteResult()’ operation is completed. The ‘createResult()’ and ‘replaceResult()’ operations take the system into the ‘Object with Consumer assigned sourcedId’ state whereas the ‘createByProxyResult()’ takes the system into the ‘Object with Provider assigned sourcedId’ state.

The system can be moved from the ‘Object with Provider assigned sourcedId’ state into the ‘Object with Consumer assigned sourcedId’ state by the successful completion of the ‘changeResultIdentifier()’ operation.

Once the system is in the ‘Object with Consumer assigned sourcedId’ or the ‘Object with Provider assigned sourcedId’ states then the ‘readResult()’, ‘readAllResultIds()’, ‘readResultIdsForPerson()’, ‘readResultIdsForLineItem()’, ‘readResultIdsForCourseOffering()’, ‘readResultIdsForCourseSection()’, ‘readResultIdsForCourseSectonWithStatus()’, ‘readResultIdsForLineItemWithLineItemType()’, ‘readResultIdsFromSavePoint()’, ‘readResults()’, ‘readResultsFromSavePoint()’, ‘updateResult()’, ‘replaceResult()’, ‘replaceResultsForLineItem()’ and ‘discoverResultIds()’ operations are now possible.

This is the state machine for each Result object in the Service Consumer and the Service Provider. The binding of the Information Model must guarantee that these two state machines remain synchronized for each result object.

The state machine for a ResultValue object is shown in Figure 3.6.

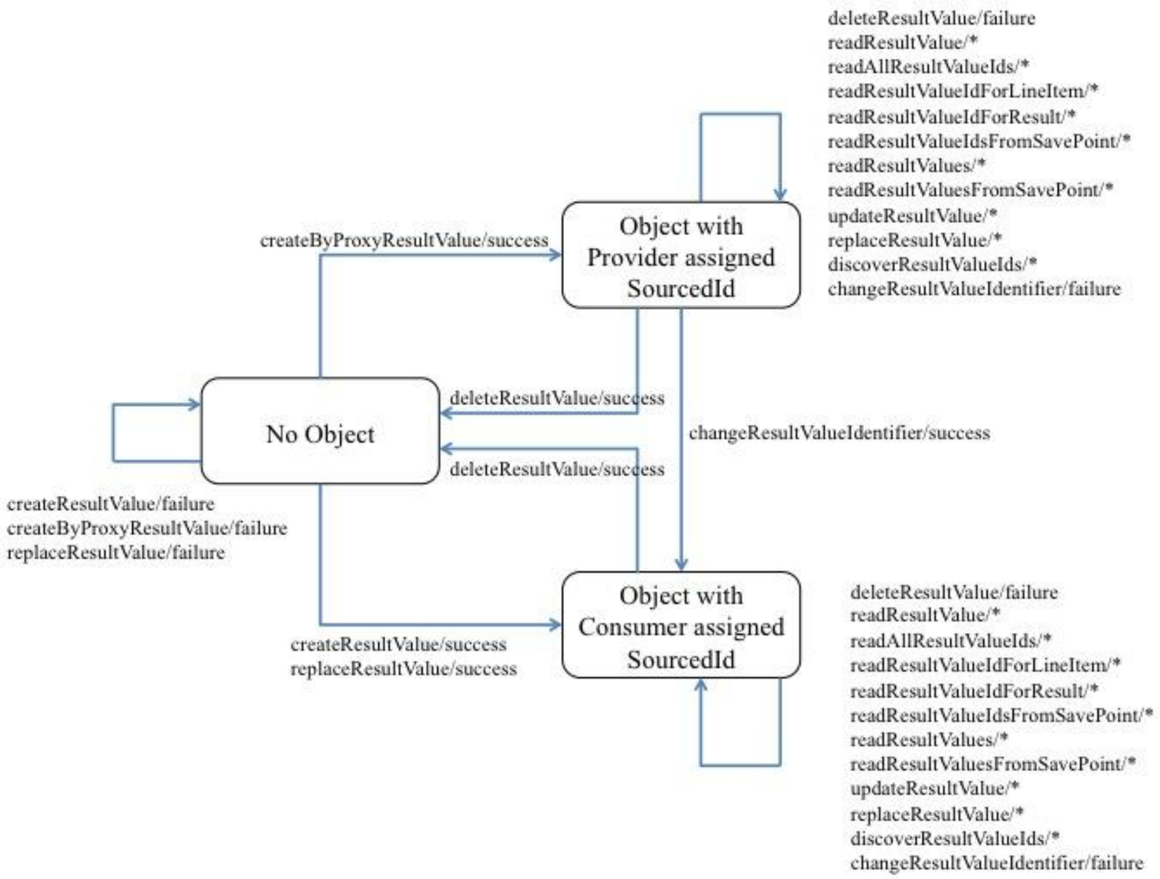

deleteResultValue/failure
readResultValue/*
readAllResultValueIds/*
readResultValueIdForLineItem/*
readResultValueIdForResult/*
readResultValueIdsFromSavePoint/*
readResultValues/*
readResultValuesFromSavePoint/*
updateResultValue/*
replaceResultValue/*
discoverResultValueIds/*
changeResultValueIdentifier/failure
createByProxyResultValue/success
Object with Provider assigned SourcedId
No Object
deleteResultValue/success
changeResultValueIdentifier/success
deleteResultValue/success
createResultValue/failure
createByProxyResultValue/failure
replaceResultValue/failure
Object with Consumer assigned SourcedId
createResultValue/success
replaceResultValue/success
deleteResultValue/failure
readResultValue/*
readAllResultValueIds/*
readResultValueIdForLineItem/*
readResultValueIdForResult/*
readResultValueIdsFromSavePoint/*
readResultValues/*
readResultValuesFromSavePoint/*
updateResultValue/*
replaceResultValue/*
discoverResultValueIds/*
changeResultValueIdentifier/failure

Figure 3.6 State machine for a ‘resultValue’ object.

The start state is ‘No Object’ i.e., the ResultValue object has not yet been created. Only the ‘createResultValue()’, ‘replaceResultValue()’ and ‘createByProxyResultValue()’ operations are possible. Once the ResultValue object has been created then it persists until a successful ‘deleteResultValue()’ operation is completed. The ‘createResultValue()’ and replaceResultValue()’ operations take the system into the ‘Object with Consumer assigned sourcedId’ state whereas the ‘createByProxyResultValue()’ takes the system into the ‘Object with Provider assigned sourcedId’ state.

The system can be moved from the ‘Object with Provider assigned sourcedId’ state into the ‘Object with Consumer assigned sourcedId’ state by the successful completion of the ‘changeResultValueIdentifier()’ operation.

Once the system is in the ‘Object with Consumer assigned sourcedId’ or the ‘Object with Provider assigned sourcedId’ states then the ‘readResultValue()’, ‘readAllResultValueIds()’, ‘readResultValueIdForLineItem()’, ‘readResultIdForResult()’, ‘readResultValueIdsFromSavePoint()’, ‘readResultValues()’, ‘readResultValuesFromSavePoint()’, ‘updateResultValue()’, ‘replaceResultValue()’ and ‘discoverResultValueIds()’ operations are now possible.

This is the state machine for each ResultValue object in the Service Consumer and the Service Provider. The binding of the Information Model must guarantee that these two state machines remain synchronized for each ResultValue object.

4 Interface Data Model

The set of operations described within the behavior model (Section 3) are based upon class descriptions specific to the parameters of the operations. It is these classes that are now described.

4.1 GUID Class Description

This is the data type for the globally unique sourcedIds. These GUIDs must be unique across the set of communicating end-systems within the Learning Information Services system. The internal format of the GUID is outside the scope of this specification but they must all be valid strings. Any implementation of the GUID class must be able to support GUIDs of at least 1024 octets in length i.e., the shortest permitted maximum length.

4.2 GUIDSet Class Description

This is the data-type for a set of GUIDs (zero or more). Any implementation of the GUIDSet must be able to contain at least 250,000 GUIDs i.e., the smallest permitted maximum number.

4.3 LineItemRecord Class Description

This is the data-type for Line Items. The data model for a Line Item is described in Section 5. A key difference for an object passed in the interface, as opposed to the requirement for an end-system, is that the content is dependent on the type of operation. A LineItemRecord object must consist of the SourcedId of the LineItem object and the LineItem object itself.

4.4 LineItemRecordSet Class Description

This is the data-type for a set of LineItemRecords (zero or more). Any implementation of the LineItemRecordSet must be able to contain at least 100,000 LineItemRecords i.e., the smallest permitted maximum number.

4.5 LineItemType Class Description

This is the data-type for the enumerated list of states used to define the status of a LineItem. The list of permitted values is: { MidTerm, Interim, Final } (this must be implemented as an extensible vocabulary).

4.6 QueryObject Class Description

This is the data-type for the query instruction. This is a String with the smallest permitted maximum length of 4096 octets. The internal structure of this string is undefined. Later versions of this specification will look at the established best practices for clarification on the use of this string.

4.7 ResultRecord Class Description

This is the data-type for Results. The data model for a Result is described in Section 5. A key difference for an object passed in the interface, as opposed to the requirement for an end-system, is that the content is dependent on the type of operation. A ResultRecord object must consist of the SourcedId of the Result object and the Result object itself.

4.8 ResultRecordSet Class Description

This is the data-type for a set of ResultRecords (zero or more). Any implementation of the ResultRecordSet must be able to contain at least 250,000 ResultRecords i.e., the smallest permitted maximum number.

4.9 ResultStatus Class Description

This is the data-type for the enumerated list of states used to define the status of a Result. The list of permitted values is: { Unmoderated, Tobemoderated, Pending, Completed } (this must be implemented as an extensible vocabulary).

4.10 ResultValueRecord Class Description

This is the data-type for Result Values. The data model for a Result Value is described in Section 5. A key difference for an object passed in the interface, as opposed to the requirement for an end-system, is that the content is dependent on the type of operation. A ResultValueRecord object must consist of the SourcedId of the ResultValue object and the ResultValue object itself.

4.11 ResultValueRecordSet Class Description

This is the data-type for a set of ResultValueRecords (zero or more). Any implementation of the ResultValueRecordSet must be able to contain at least 1000 ResultValueRecords i.e., the smallest permitted maximum number.

4.12 SequenceIdentifier Class Description

This is the data-type for the sequence identifier used to denote identify the synchronization reference point between the two communicating systems. The sequence is denoted by the date-time string YYYY-MM-DDTHH:MM:SS.NNN where 'YYYY' denotes the year, the first 'MM' string the month (01-12), 'DD' the day (01-31), 'HH' the hour (00-23), the second 'MM' string the minute (00-59), 'SS' the second (00-59) and 'NN' the millisecond value (000-999).

At initialization the value is set to '2000-01-01T00:00:00.000'. The value is changed to the current time for every operation that results in a change of the value of the data stored in the associated object.

All values are to be rounded down at the level of greatest resolution.

4.13 StatusInfo Class Description

This is the data-type for the status information returned by the target to the source. The structure of this class is described in the IMS GLC General Web Services specification v1.0 (Appendix A) [GWS, 05].

4.14 ReplaceStatusCodes Class Description

The PIM for the ReplaceStatusCodes class is shown in Figure 4.1.

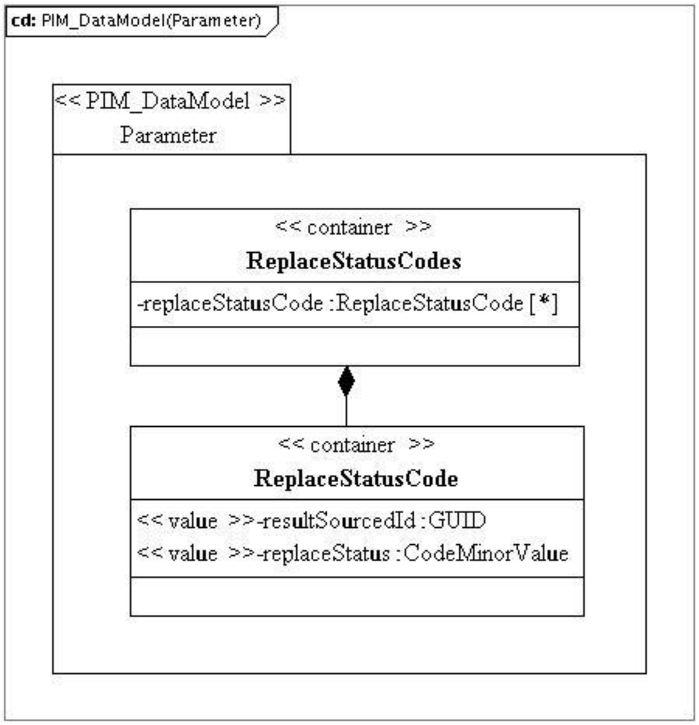

cd: PIM_DataModel(Parameter)
<< PIM_DataModel >>
Parameter
<< container >>
ReplaceStatusCodes
-replaceStatusCode : ReplaceStatusCode [*]
<< container >>
ReplaceStatusCode
<< value >>-resultSourcedId : GUID
<< value >>-replaceStatus : CodeMinorValue

Figure 4.1 ReplaceStatusCodes class diagram.

Table 4.1 Description of the ReplaceStatusCodes class.

Descriptor	Definition
Class name	ReplaceStatusCodes
Class type	container
Multiplicity	1
Parents	Service Parameter
Children	[replaceStatusCode], unordered
Description	The list of status codes returned for the ‘replaceResultsForLineItem’ operation.

4.14.1 ReplaceStatusCode Attribute Description

Table 4.2 Description of the 'replaceStatusCode' attribute for the ReplaceStatusCodes class.

Descriptor	Definition
Attribute name	replaceStatusCode
Data type	replaceStatusCode
Value space	container
Multiplicity	0..*
Description	This is the container for the returned status code for each attempted result replacement.

4.15 ReplaceStatusCode Class Description

The PIM for the ReplaceStatusCode class is shown in Figure 4.1.

Table 4.3 Description of the ReplaceStatusCode class.

Descriptor	Definition
Class name	ReplaceStatusCode
Class type	container
Multiplicity	0..*
Parents	ReplaceStatusCodes
Children	[resultSourcedId, replaceStatus]
Description	The status code for a specific result replacement attempt.

4.15.1 ResultSourcedId Attribute Description

Table 4.4 Description of the 'resultSourcedId' attribute for the ReplaceStatusCode class.

Descriptor	Definition
Attribute name	resultSourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	1
Description	The 'sourcedId' of the Result object whose content is being replaced.

4.15.2 ReplaceStatus Attribute Description

Table 4.5 Description of the ‘replaceStatus’ attribute for the ReplaceStatusCode class.

Descriptor	Definition
Attribute name	replaceStatus
Data type	Enumerated vocabulary.
Value space	This defines the status code for the attempted replace action. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model. The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.
Multiplicity	1
Description	The reported status of the result replacement request for the Result object identified by the associated ‘sourcedId’.

5 End System Data Model

The end system data model defines the persistence model that must be maintained by an end system to ensure the correct system behavior.

An informative overview of the entire Persistence Data Model is provided as a Platform Independent Model (PIM) expressed in UML constructs. All UML diagrams expressed as "Platform Independent Model" are non-normative. Normative tables defining the classes in this Information Model follow the informative UML diagrams. A full definition of the UML Profile and the terms used in the normative tabular descriptions in this document to describe the PIM can be found in [SDN07, 06].

In the tables in this section the character sequence "n/a" is used to mark a field "not applicable." Any field so marked is not relevant to the class being defined. Features so marked shall be ignored when binding a class defined by this Information Model.

5.1 Key terms and concepts

Classes in this information model are classified into four abstract class types. These abstractions are bound to specific data structures for machine processing in the IMS GLC Outcomes Management Services WSDL Binding [OMS, 11]. The abstract class types are:

- **container:** A container class may be a parent of one or more child classes;
- **value:** A value class shall not be a parent. That is, it shall not be a composite of characteristic, container, value, or unspecified class types. A value class shall always be a child of a container class and shall have semantic value within the scope of its parent class's semantic value;
- **unspecified: An unspecified class may be a parent.** An unspecified class serves as an extension point for this Information Model.

Table 5.1 lists the class descriptors used to describe the abstract classes and definitions of the descriptors.

Table 5.1 Class descriptors

Descriptor	Definition
Class name	The name given to the class being described.
Class type	The abstract class type of this class.
Data type	For value and characteristic classes, the allowed structure for valid values for the class. Valid data types are: **DateTime:** The DateTime represents a combined date and time in the format of ISO 8601 i.e., 'YYYY-MM-DDTHH:MM:SS±TZ'. The time is denoted in Coordinated Universal Time. **Decimal:** A real number. **GUID**: An identifier that is globally unique within the Learning Information Services System. This will be based upon the Normalized String data-type that has a constrained value-space. **Integer:** An integer. **NormalizedString:** A sequence of printable characters that does not contain carriage returns or tabs. **String:** A sequence of printable characters. **Text:** A language annotated string (this is in fact a separate class but it is treated as data-

Descriptor	Definition
	type for convenience). The string is accompanied by an attribute that denotes the language for the string. **AnyURI**: Any syntactically valid instance of a URI as defined in RFC3986. Note: Many of the foundational Specifications, Standards, and Recommendations referred to by this Information Model use RFC2396 and RFC2732 as the definitions of URI. These are made obsolete by RFC3986, but many of the foundational documents have not been updated to reference RFC3986. **URL:** A normalized string that is used to contain a Universal Resource Location. **unspecified:** The data type is not known or is not important.
Value space	The range of valid values for this class. If the value space is unspecified, it is not known or is not important.
Multiplicity	A property of a class indicating the number of times it may be used or appear in a given parent context. The values of this property are expressed as a range or shorthand for a range using this notation: • **‘0..1’** [optional; restricted] • **‘0..unbounded’** [optional; unrestricted] • **‘1..1’** [mandatory; restricted] • **‘1..unbounded’** [mandatory; unrestricted] Multiplicities may also appear in short-hand notation in the UML models. The short-hand equivalents shall be (exclusive of bracketed comments): • **‘*’** [optional; unrestricted] • **‘1’** [mandatory; restricted] • **‘1..*’** [mandatory; unrestricted] Where multiplicity is greater than one, the importance of the ordering of siblings is also indicated by appending either **“,”ordered** or **“,” unordered.** **ordered** specifies a sequence of siblings as listed, **unordered** specifies a collection or bag of siblings for which the order is not important.
Parents	Lists classes that may be parents of this class.
Children	Lists the possible child classes of this class in the form **“[” child *“,” child “]”.** One or more child classes may be expressed within square brackets. Each child class in the list shall be separated by a comma. Where more than one child is listed, the importance of the ordering of siblings is also indicated by appending either **“,”ordered** or **“,” unordered.** **ordered** specifies a sequence of siblings as listed, **unordered** specifies a collection or bag of siblings for which the order is not important
Description	Contains descriptions relating to the class and its values space.

In general, this specification does not define the ways in which an end system must be realized. However, the required interoperability behavior requires that an end system have certain characteristics. The static properties of these characteristics are defined in this Section, including:

- When an attribute has a multiplicity of ‘1..1’ then an end system must be capable of supporting one instance;
- When an attribute has a multiplicity of ‘1..*’ then an end system must be capable of supporting at least one instance. The specification will also define the smallest permitted maximum number of instances that must also be supported by the end system;
- When an attribute has a multiplicity of ‘0..1’ then an end system should support a single instance;

- When an attribute has a multiplicity of ‘0..*’ then the specification will define the smallest permitted maximum number of instances that must also be supported by the end system.

When the object is passed as part of a service call then attributes that have a ‘1..1’ or ‘1..*’ multiplicity may not be exchanged. This is because the specification of an end system defines capability; an operational system may or may not exchange the associated information.

5.2 The Structure of an Outcome

The conceptual data model for the structure of an outcome is shown in Figure 5.1. There are three facets of the Outcome structure:

- The components of the core data model of the course denoted in Figure 5.1 by the LineItem, Result and ResultValue classes;
- The reference structures that are used to identify specific instances of each part of the Outcomes are denoted in Figure 5.1 by the classes LineItemRecord, ResultRecord and ResultValueRecord.

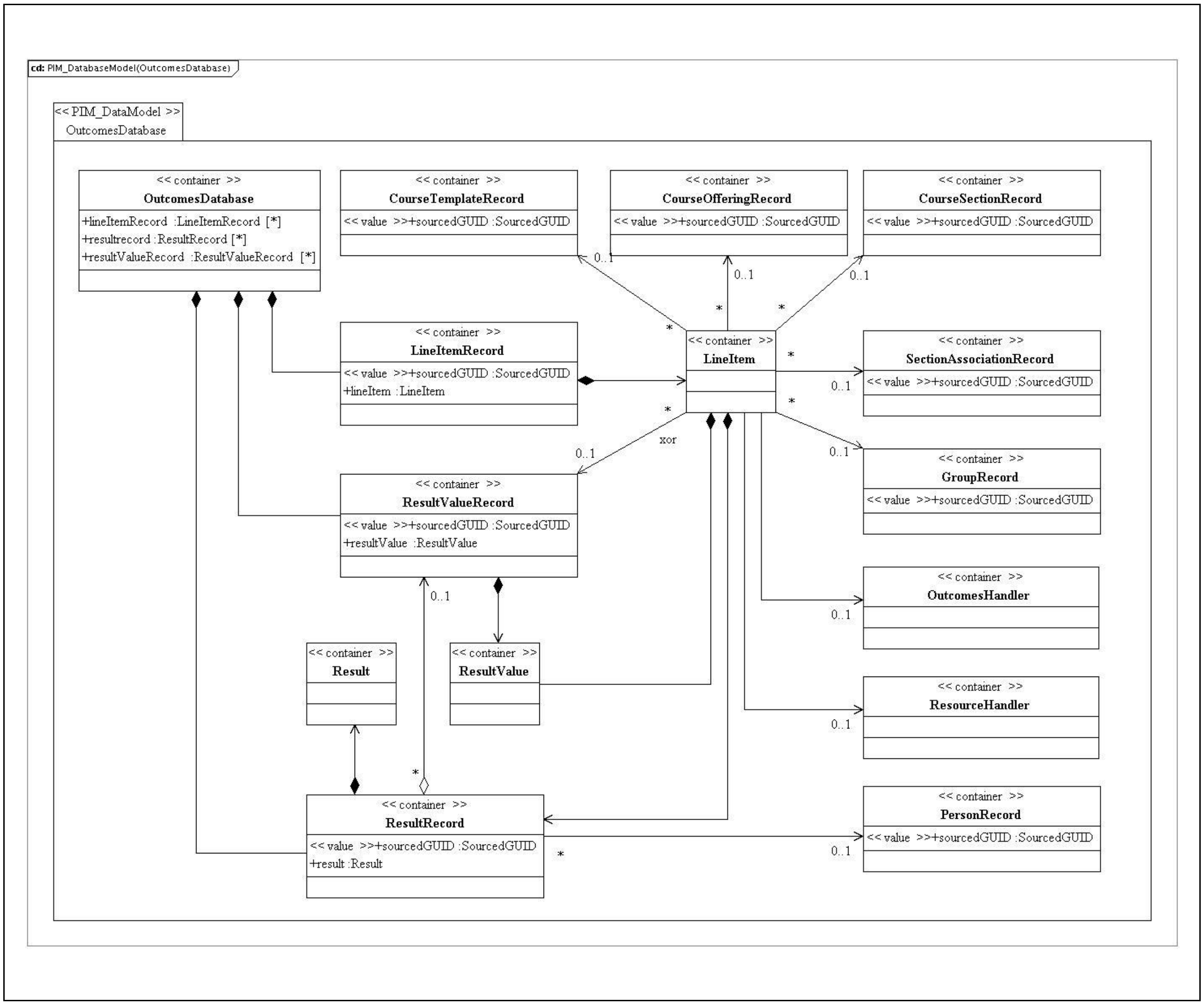


Figure 5.1 OutcomesDatabase class diagram.

5.3 OutcomesDatabase Class Description

Table 5.2 Description of the OutcomesDatabase class.

Descriptor	Definition
Class name	OutcomesDatabase
Class type	container
Multiplicity	1
Parents	N/A (Root)
Children	[lineItemRecord, resultRecord, resultValueRecord], unordered
Description	This is the database within the end-system that contains all of the Outcomes objects data. The database consists of the set of LineItem, Result and ResultValue objects, the set of GUIDs and the relationship mapping between the two. The manner in which this information is physically stored is outside of the scope of this specification.

5.3.1 LineItemRecord Attribute Description

Table 5.3 Description of the 'lineItemRecord' attribute for the OutcomesDatabase class.

Descriptor	Definition
Attribute name	lineItemRecord
Data type	LineItemRecord
Value space	container
Multiplicity	0..unbounded, unordered
Description	The set of uniquely identified LineItem instances. An instance of the OutcomesDatabase must support at least 1000 lineItemRecord instances.

5.3.2 ResultRecord Attribute Description

Table 5.4 Description of the 'ResultRecord' attribute for the OutcomesDatabase class.

Descriptor	Definition
Attribute name	resultRecord
Data type	ResultRecord
Value space	container
Multiplicity	0..unbounded, unordered
Description	The set of uniquely identified Result instances. An instance of the OutcomesDatabase must support at least 100,000 resultRecord instances.

5.3.3 ResultValueRecord Attribute Description

Table 5.5 Description of the 'ResultValueRecord' attribute for the OutcomesDatabase class.

Descriptor	Definition
Attribute name	ResultValueRecord
Data type	ResultValueRecord
Value space	container
Multiplicity	0..unbounded, unordered
Description	The set of uniquely identified ResultValue instances. An instance of the OutcomesDatabase must support at least 100 resultValueRecord instances.

5.4 LineItemRecord Class Description

Table 5.6 Description of the LineItemRecord class.

Descriptor	Definition
Class name	LineItemRecord
Class type	container
Multiplicity	0..unbounded, unordered
Parents	OutcomesDatabase
Children	[sourcedGUID, lineItem], ordered
Description	The LineItemRecord represents the association the unique identifier (sourcedGUID) for the LineItem object with the LineItem object itself. The identifier object is not a part of the LineItem object but both are managed within the Outcomes Database. There is an isomorphic association between each pair of identifier and LineItem objects.

5.4.1 SourcedGUID Attribute Description

Table 5.7 Description of the ‘sourcedGUID’ attribute for the LineItemRecord class.

Descriptor	Definition
Attribute name	sourcedGUID
Data type	SourcedGUID
Value space	container
Multiplicity	1
Description	This is the globally unique identifier that has been assigned to the associated LineItem object. Each line item object must have only one SourcedGUID but this may be changed, any number of times, during the object’s lifetime.

5.4.2 LineItem Attribute Description

Table 5.8 Description of the 'lineItem' attribute for the OutcomesDatabase class.

Descriptor	Definition
Attribute name	lineItem
Data type	LineItem
Value space	container
Multiplicity	1
Description	This is the LineItem component of the Outcome. A LineItem is a set of results that may be related to a CourseTemplate (Template), CourseOffering (Offering), CourseSection (Section), SectionAssociation (Association) or, in special cases, a Group.

5.5 ResultRecord Class Description

Table 5.9 Description of the ResultRecord class.

Descriptor	Definition
Class name	ResultRecord
Class type	container
Multiplicity	0..unbounded, unordered
Parents	OutcomesDatabase
Children	[sourcedGUID, result], ordered
Description	The ResultRecord represents the association the unique identifier (sourcedGUID) for the Result object with the Result object itself. The identifier object is not a part of the Result object but both are managed within the Outcomes Database. There is an isomorphic association between each pair of identifier and Result objects.

5.5.1 SourcedGUID Attribute Description

Table 5.10 Description of the 'sourcedGUID' attribute for the ResultRecord class.

Descriptor	Definition
Attribute name	sourcedGUID
Data type	SourcedGUID
Value space	container
Multiplicity	1
Description	This is the globally unique identifier that has been assigned to the associated Result object. Each result object must have only one SourcedGUID but this may be changed, any number of times, during the object's lifetime.

5.5.2 Result Attribute Description

Table 5.11 Description of the 'result' attribute for the ResultRecord class.

Descriptor	Definition
Attribute name	result
Data type	Result
Value space	container
Multiplicity	1
Description	This is the Result component of the Outcome. A Result is a recorded grade or mark.

5.6 ResultValueRecord Class Description

Table 5.12 Description of the ResultValueRecord class.

Descriptor	Definition
Class name	ResultValueRecord
Class type	container
Multiplicity	0..unbounded, unordered
Parents	OutcomesDatabase
Children	[sourcedGUID, resultValue], ordered
Description	The ResultValueRecord represents the association the unique identifier (sourcedGUID) for the ResultValue object with the result object itself. The identifier object is not a part of the ResultValue object but both are managed within the Outcomes Database. There is an isomorphic association between each pair of identifier and ResultValue objects.

5.6.1 SourcedGUID Attribute Description

Table 5.13 Description of the 'sourcedGUID' attribute for the ResultValueRecord class.

Descriptor	Definition
Attribute name	sourcedGUID
Data type	SourcedGUID
Value space	container
Multiplicity	1
Description	This is the globally unique identifier that has been assigned to the associated ResultValue object. Each ResultValue object must have only one SourcedId but this may be changed, any number of times, during the object's lifetime.

5.6.2 ResultValue Attribute Description

Table 5.14 Description of the 'ResultValue' attribute for the ResultValueRecord class.

Descriptor	Definition
Attribute name	ResultValue
Data type	ResultValue
Value space	container
Multiplicity	1
Description	This is the ResultValue component of the Outcome. A ResultValue is a set of grades or marks that can be assigned as the value for a Result object.

5.7 CourseTemplateRecord Class Description

Table 5.15 Description of the CourseTemplateRecord class.

Descriptor	Definition
Class name	CourseTemplateRecord
Class type	container
Multiplicity	0..1
Parents	CourseDatabase
Description	One of the components of a Course (see the IMS GLC Course Management Service Information Model v1.0 specification). A LineItem may be related to a CourseTemplate.

5.8 CourseOfferingRecord Class Description

Table 5.16 Description of the CourseOfferingRecord class.

Descriptor	Definition
Class name	CourseOfferingRecord
Class type	container
Multiplicity	0..1
Parents	CourseDatabase
Description	One of the components of a Course (see the IMS GLC Course Management Service Information Model v1.0 specification). A LineItem may be related to a CourseOffering.

5.9 CourseSectionRecord Class Description

Table 5.17 Description of the CourseSectionRecord class.

Descriptor	Definition
Class name	CourseSectionRecord
Class type	container
Multiplicity	0..1
Parents	CourseDatabase
Description	One of the components of a Course (see the IMS GLC Course Management Service Information Model v1.0 specification). A LineItem may be related to a CourseSection.

5.10 SectionAssociationRecord Class Description

Table 5.18 Description of the SectionAssociationRecord class.

Descriptor	Definition
Class name	SectionAssociationRecord
Class type	container
Multiplicity	0..1
Parents	CourseDatabase
Description	One of the components of a Course (see the IMS GLC Course Management Service Information Model v1.0 specification). A LineItem may be related to a SectionAssociation.

5.11 GroupRecord Class Description

Table 5.19 Description of the GroupRecord class.

Descriptor	Definition
Class name	GroupRecord
Class type	container
Multiplicity	0..1
Parents	GroupDatabase
Description	A Group object (see the IMS GLC Group Management Service Information Model v2.0 specification). A LineItem may be related to a Group when the Group has been used to extend the course structure.

5.12 PersonRecord Class Description

Table 5.20 Description of the PersonRecord class.

Descriptor	Definition
Class name	PersonRecord
Class type	container
Multiplicity	0..1
Parents	PersonDatabase
Description	A Person object (see the IMS GLC Person Management Service Information Model v2.0 specification). Each Result is associated with a Person object.

5.13 OutcomesHandler Class Description

Table 5.21 Description of the OutcomesHandler class.

Descriptor	Definition
Class name	OutcomesHandler
Class type	container
Multiplicity	0..1
Parents	IMS GLC Learning Tools Interoperability v1.0 specification
Description	An OutcomesHandler object (see the IMS GLC Learning Tools Interoperability v1.0 specification). A LineItem may be associated with an LTI OutcomesHandler object.

5.14 ResourceHandler Class Description

Table 5.22 Description of the ResourceHandler class.

Descriptor	Definition
Class name	ResourceHandler
Class type	container
Multiplicity	0..1
Parents	IMS GLC Learning Tools Interoperability v1.0 specification
Description	An OutcomesHandler object (see the IMS GLC Learning Tools Interoperability v1.0 specification). A LineItem may be associated with an LTI ResourceHandler object.

5.15 LineItem Class Description

The PIM for the LineItem is shown in Figure 5.2.

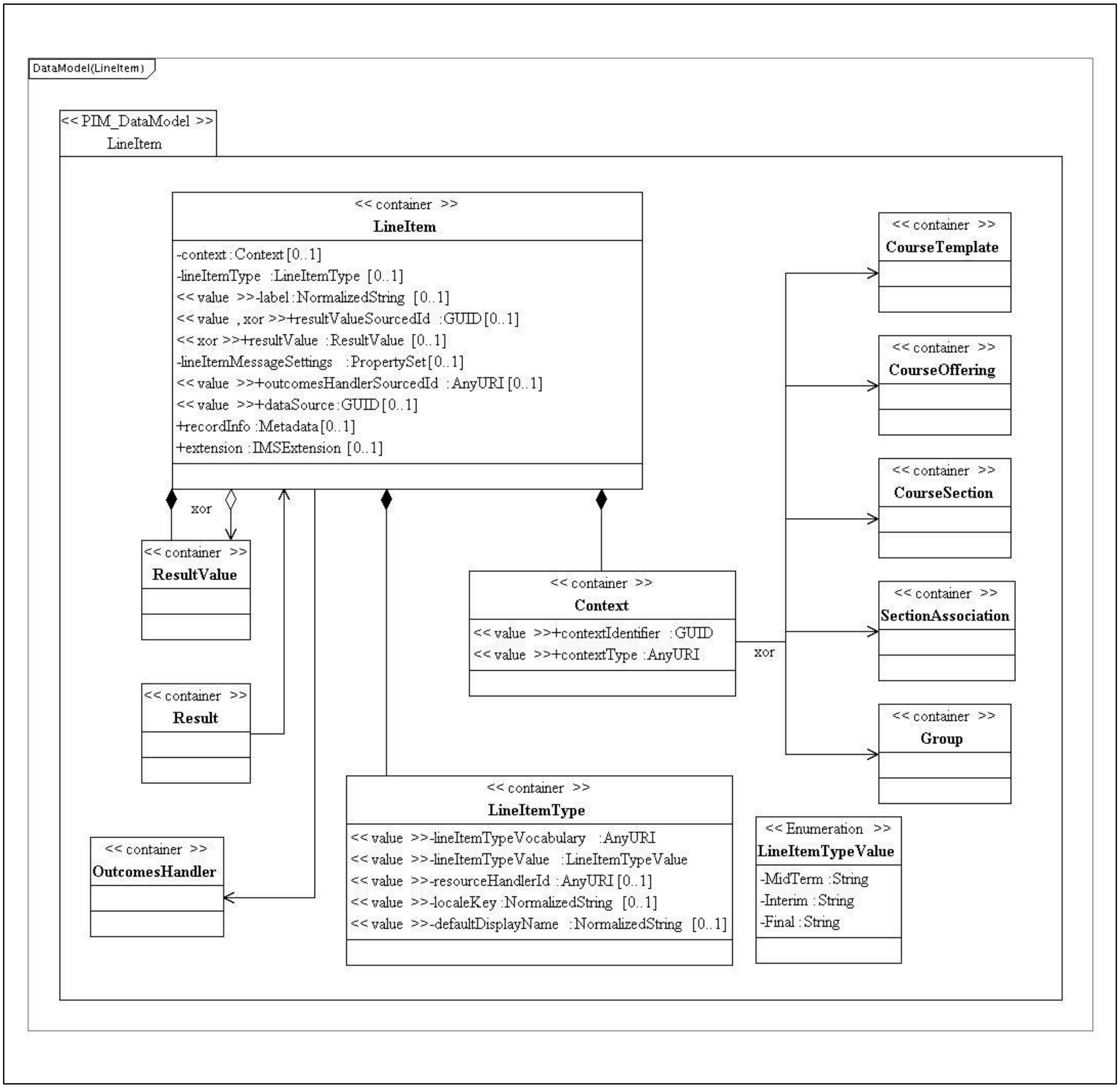

DataModel(LineItem)
<< PIM_DataModel >>
LineItem
<< container >>
LineItem
-context : Context [0..1]
-lineItemType : LineItemType [0..1]
<< value >>-label : NormalizedString [0..1]
<< value , xor >>+resultValueSourcedId : GUID [0..1]
<< xor >>+resultValue : ResultValue [0..1]
-lineItemMessageSettings : PropertySet [0..1]
<< value >>+outcomesHandlerSourcedId : AnyURI [0..1]
<< value >>+dataSource : GUID [0..1]
+recordInfo : Metadata [0..1]
+extension : IMSExtension [0..1]
xor
<< container >>
ResultValue
<< container >>
Result
<< container >>
OutcomesHandler
<< container >>
Context
<< value >>+contextIdentifier : GUID
<< value >>+contextType : AnyURI
xor
<< container >>
CourseTemplate
<< container >>
CourseOffering
<< container >>
CourseSection
<< container >>
SectionAssociation
<< container >>
Group
<< container >>
LineItemType
<< value >>-lineItemTypeVocabulary : AnyURI
<< value >>-lineItemTypeValue : LineItemTypeValue
<< value >>-resourceHandlerId : AnyURI [0..1]
<< value >>-localeKey : NormalizedString [0..1]
<< value >>-defaultDisplayName : NormalizedString [0..1]
<< Enumeration >>
LineItemTypeValue
-MidTerm : String
-Interim : String
-Final : String

Figure 5.2 LineItem class diagram.

Table 5.23 Description of the LineItem class.

Descriptor	Definition
Class name	LineItem
Class type	container
Multiplicity	1
Parents	LineItemRecord
Children	[context, lineItemType, label, resultValueSourcedId, resultValue, lineItemMessageSettings, outcomesHandlerSourcedId, dataSource, recordInfo, extension], ordered
Description	A LineItem is a list of Result objects that are, in general, associated with some course activity.

5.15.1 ContextType Attribute Description

Table 5.24 Description of the 'contextType' attribute for the LineItem class.

Descriptor	Definition
Attribute name	context
Data type	Context
Value space	container
Multiplicity	0..1
Description	This is the container for information that establishes the context for the lineItem.

5.15.2 LineItemType Attribute Description

Table 5.25 Description of the 'lineItemType' attribute for the LineItem class.

Descriptor	Definition
Attribute name	lineItemType
Data type	LineItemType
Value space	container
Multiplicity	0..1
Description	The container for the definition of the type of lineItem.

5.15.3 Label Attribute Description

Table 5.26 Description of the 'label' attribute for the LineItem class.

Descriptor	Definition
Attribute name	label
Data type	A Normalized String: [1-31 characters].
Multiplicity	0..1
Description	This corresponds to the text that typically appears in the column heading for the lineItem.

5.15.4 ResultValueSourcedId Attribute Description

Table 5.27 Description of the 'resultValueSourcedId' attribute for the LineItem class.

Descriptor	Definition
Attribute name	resultValueSourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	This is the sourcedId of the ResultValue that are used to define the context for the values contained in the Result objects for the LineItem. A lineItem will have either a sourcedId pointer to the resultValue or the embedded resultValue object itself.

5.15.5 ResultValue Attribute Description

Table 5.28 Description of the 'resultValue' attribute for the LineItem class.

Descriptor	Definition
Attribute name	resultValue
Data type	ResultValue
Value space	container
Multiplicity	0..1
Description	This is the embedded ResultValue structure for the lineItem. A lineItem will have either a sourcedId pointer to the resultValue or the embedded resultValue object itself.

5.15.6 LineItemMessageSettings Attribute Description

Table 5.29 Description of the 'lineItemMessageSettings' attribute for the LineItem class.

Descriptor	Definition
Attribute name	lineItemMessageSettings
Data type	PropertySet
Value space	container
Multiplicity	0..1
Description	The container for a collection of name/value pairs that must be embedded in each message sent to the Outcomes Handler.

5.15.7 OutcomesHandlerSourcedId Attribute Description

Table 5.30 Description of the 'outcomesHandlerSourcedId' attribute for the LineItem class.

Descriptor	Definition
Attribute name	outcomesHandlerSourcedId
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	0..1
Description	Points to a processing handler for the lineItem.

5.15.8 DataSource Attribute Description

Table 5.31 Description of the 'dataSource' attribute for the LineItem class.

Descriptor	Definition
Attribute name	dataSource
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	An identifier of the original source system of the LineItem object.

5.15.9 RecordInfo Attribute Description

Table 5.32 Description of the ‘recordInfo’ attribute for the LineItem class.

Descriptor	Definition
Attribute name	recordInfo
Data type	Metadata
Value space	container
Multiplicity	0..1
Description	The container for metadata about the LineItem object. No particular form of metadata is mandated.

5.15.10 Extension Attribute Description

Table 5.33 Description of the ‘extension’ attribute for the LineItem class.

Descriptor	Definition
Attribute name	extension
Data type	IMSExtension
Value space	container
Multiplicity	0..1
Description	The extension mechanism for the LineItem data model.

5.16 Context Class Description

The PIM for the LinetemType class is shown in Figure 5.2.

Table 5.34 Description of the Context class.

Descriptor	Definition
Class name	Context
Class type	container
Multiplicity	0..1
Parents	LineItem
Children	[contextIdentifier, contextType], ordered
Description	The container for all of the information that establishes the context for the lineItem.

5.16.1 ContextIdentifier Attribute Description

Table 5.35 Description of the 'contextIdentifier' attribute for the Context class.

Descriptor	Definition
Attribute name	contextIdentifier
Data type	GUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is the sourcedId of the course component to which the LineItem is associated. This is a sourcedId for either a CourseTemplate or CourseOffering or CourseSection or SectionAssociation or, in special cases, a Group.

5.16.2 ContextType Attribute Description

Table 5.36 Description of the 'contextType' attribute for the Context class.

Descriptor	Definition
Attribute name	contextType
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	This is a URN that describes the context for the associated LineItem. The use of this attribute is defined by the associated profile. For example, see the IMS GLC LTI v1.0 specification for the use of this attribute in the context of supporting learning tool interactions.

5.17 LineItemType Class Description

The PIM for the LinetemType class is shown in Figure 5.2.

Table 5.37 Description of the LineItemType class.

Descriptor	Definition
Class name	LineItemType
Class type	container
Multiplicity	0..1
Parents	ContextType
Children	[lineItemTypeVocabulary, lineItemTypeValue, localeKey, resourceHandlerId, defaultDisplayName], ordered
Description	The container for definition of the type of lineItem.

5.17.1 LineItemTypeVocabulary Attribute Description

Table 5.38 Description of the 'lineItemTypeVocabulary' attribute for the LineItemType class.

Descriptor	Definition
Attribute name	lineItemTypeVocabulary
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted lineItemType values. If this is a reference to a VDEX file it is the 'vocabIdentifier' of the vocabulary.

5.17.2 LineItemTypeValue Attribute Description

Table 5.39 Description of the 'lineItemTypeValue' attribute for the LineItemType class.

Descriptor	Definition
Attribute name	lineItemTypeValue
Data type	LineItemTypeValue
Value space	Vocabulary-based. The core vocabulary is given in Appendix B.
Multiplicity	1
Description	This defines the type/status of the LineItem. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model. The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.

5.17.3 ResourceHandlerSourcedId Attribute Description

Table 5.40 Description of the 'resourceHandlerSourcedId' attribute for the LineItemType class.

Descriptor	Definition
Attribute name	resourceHandlerSourcedId
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	0..1
Description	A LineItemType may optionally be associated with a resource handler that is declared in the Tool Profile. The resourceHandlerURN is the fully qualified URN for the associated resource handler. A Tool Consumer may use the associated resource handler to obtain an icon to render in the gradebook for the line items of the given type, or to classify line items based on the resource type.

5.17.4 LocaleKey Attribute Description

Table 5.41 Description of the 'localeKey' attribute for the LineItemType class.

Descriptor	Definition
Attribute name	localeKey
Data type	Normalized String
Value space	A Normalized String: [1-4095 characters].
Multiplicity	0..1
Description	A key used to look up the language-specific display name for the LineItemType from a resource bundle.

5.17.5 DefaultDisplayName Attribute Description

Table 5.42 Description of the 'defaultDisplayName' attribute for the LineItemType class.

Descriptor	Definition
Attribute name	defaultDisplayName
Data type	Normalized String
Value space	A Normalized String: [1-4095 characters].
Multiplicity	0..1
Description	A default value for the display name of the LineItemType. This value will be used if the "localization" capability is not supported, or if a locale-specific display name cannot be found based on the localeKey.

5.18 Result Class Description

The PIM for the Result is shown in Figure 5.3.

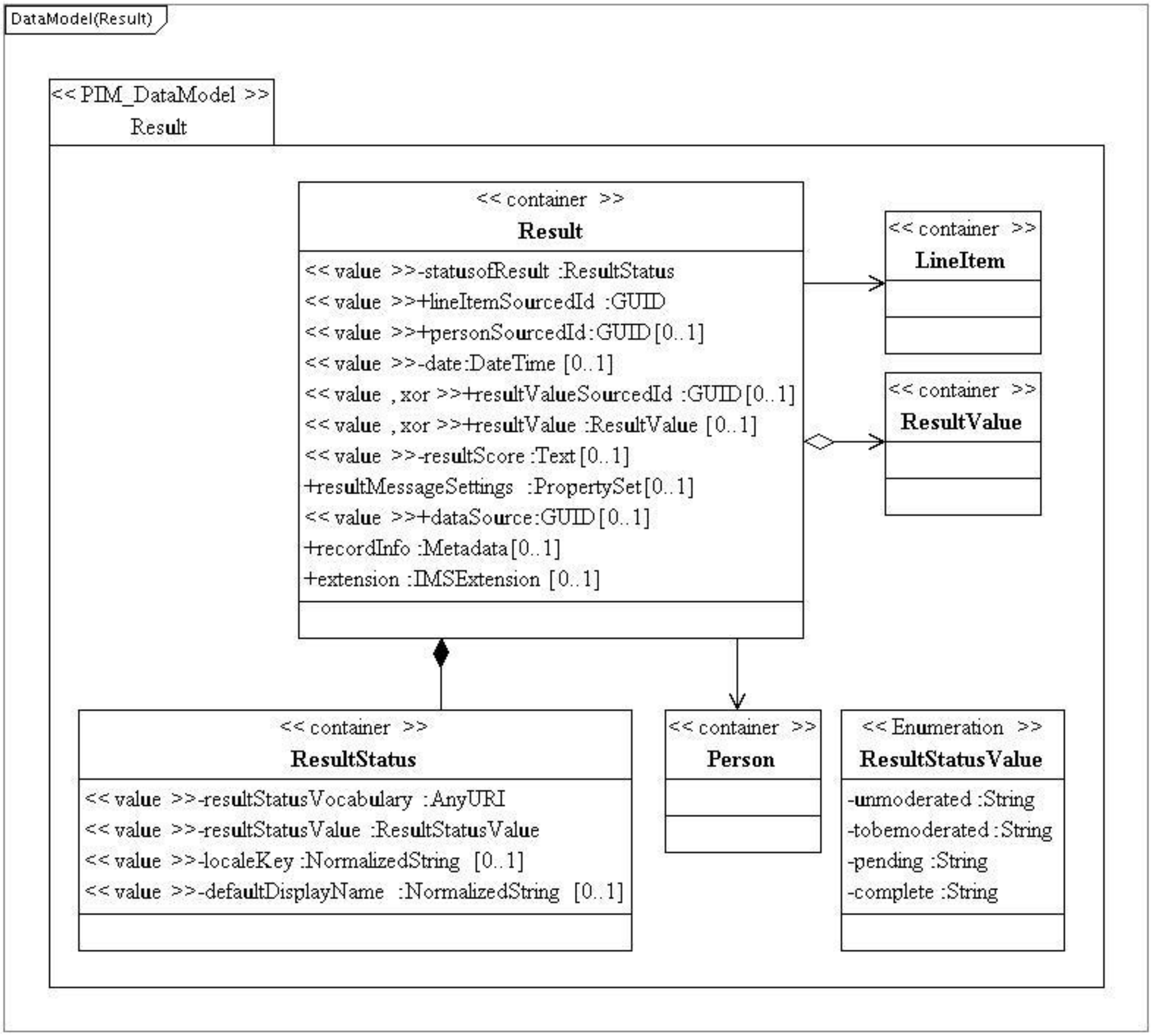

DataModel(Result)
<< PIM_DataModel >>
Result
<< container >>
Result
<< value >>-statusofResult :ResultStatus
<< value >>+lineItemSourcedId :GUID
<< value >>+personSourcedId:GUID [0..1]
<< value >>-date:DateTime [0..1]
<< value , xor >>+resultValueSourcedId :GUID [0..1]
<< value , xor >>+resultValue :ResultValue [0..1]
<< value >>-resultScore :Text [0..1]
+resultMessageSettings :PropertySet [0..1]
<< value >>+dataSource:GUID [0..1]
+recordInfo :Metadata [0..1]
+extension :IMSExtension [0..1]
<< container >>
LineItem
<< container >>
ResultValue
<< container >>
ResultStatus
<< value >>-resultStatusVocabulary :AnyURI
<< value >>-resultStatusValue :ResultStatusValue
<< value >>-localeKey :NormalizedString [0..1]
<< value >>-defaultDisplayName :NormalizedString [0..1]
<< container >>
Person
<< Enumeration >>
ResultStatusValue
-unmoderated :String
-tobemoderated :String
-pending :String
-complete :String

Figure 5.3 Result class diagram.

Table 5.43 Description of the Result class.

Descriptor	Definition
Class name	Result
Class type	container
Multiplicity	1
Parents	ResultRecord
Children	[statusofResult, lineItemSourcedId, personSourcedId, date, resultValueSourcedId, resultValue, resultScore, resultMesageSettings, dataSource, recordInfo, extension], ordered
Description	This is the container for the results. Result objects do not have to be associated with a LineItem.

5.18.1 StatusofResult Attribute Description

Table 5.44 Description of the 'statusofResult' attribute for the Result class.

Descriptor	Definition
Attribute name	statusofResult
Data type	ResultStatus
Value space	container
Multiplicity	1
Description	The container for the definition of the status of the result.

5.18.2 LineItemSourcedId Attribute Description

Table 5.45 Description of the 'lineItemSourcedId' attribute for the Result class.

Descriptor	Definition
Attribute name	lineItemSourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is the sourcedId of the LineItem to which the Result object is assigned.

5.18.3 PersonSourcedId Attribute Description

Table 5.46 Description of the ‘personSourcedId’ attribute for the Result class.

Descriptor	Definition
Attribute name	personSourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	This is the sourcedId of the Person for whom the Result object is assigned.

5.18.4 Date Attribute Description

Table 5.47 Description of the ‘date’ attribute for the Result class.

Descriptor	Definition
Attribute name	date
Data type	DateTime
Value space	ISO 8601 i.e., ‘YYYY-MM-DDTHH:MM:SS±TZ’
Multiplicity	0..1
Description	This is the dateTime at which the current status of the Result was recorded.

5.18.5 ResultValueSourcedId Attribute Description

Table 5.48 Description of the ‘ResultValueSourcedId’ attribute for the Result class.

Descriptor	Definition
Attribute name	ResultValueSourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	The sourcedId of the ResultValue object that sets the context for the values in the Result object. A result will have either a sourcedId pointer to the resultValue or the embedded resultValue object itself.

5.18.6 ResultValue Attribute Description

Table 5.49 Description of the ‘resultValue’ attribute for the Result class.

Descriptor	Definition
Attribute name	resultValue
Data type	ResultValue
Value space	container
Multiplicity	0..1
Description	This is the embedded ResultValue structure for the lineItem. A result will have either a sourcedId pointer to the resultValue or the embedded resultValue object itself.

5.18.7 ResultScore Attribute Description

Table 5.50 Description of the ‘resultScore’ attribute for the Result class.

Descriptor	Definition
Attribute name	resultScore
Data type	Text
Value space	A language dependent String [1-127 characters]. The default language is ‘en-US’.
Multiplicity	0..1
Description	Value of the result assigned to the member for participation in the collection. Ideally this would be one of the values from the results values list e.g., A+, 85%, Completed, etc.

5.18.8 ResultMessageSettings Attribute Description

Table 5.51 Description of the ‘resultMessageSettings’ attribute for the Result class.

Descriptor	Definition
Attribute name	resultMessageSettings
Data type	PropertySet
Value space	container
Multiplicity	0..1
Description	The container for information about tool that work with the Result information.

5.18.9 DataSource Attribute Description

Table 5.52 Description of the ‘dataSource’ attribute for the Result class.

Descriptor	Definition
Attribute name	dataSource
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	An identifier of the original source system of the Result object.

5.18.10 RecordInfo Attribute Description

Table 5.53 Description of the ‘recordInfo’ attribute for the Result class.

Descriptor	Definition
Attribute name	recordInfo
Data type	Metadata
Value space	container
Multiplicity	0..1
Description	The container for metadata about the Result object. No particular form of metadata is mandated.

5.18.11 Extension Attribute Description

Table 5.54 Description of the ‘extension’ attribute for the Result class.

Descriptor	Definition
Attribute name	extension
Data type	IMSExtension
Value space	container
Multiplicity	0..1
Description	The extension mechanism for the Result data model.

5.19 ResultStatus Class Description

The PIM for the ResultStatus class is shown in Figure 5.3.

5.19.1 ResultStatusVocabulary Attribute Description

Table 5.55 Description of the 'resultStatusVocabulary' attribute for the ResultStatus class.

Descriptor	Definition
Attribute name	resultStatusVocabulary
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted values for the result status. If this is a reference to a VDEX file it is the 'vocabIdentifier' of the vocabulary.

5.19.2 ResultStatusValue Attribute Description

Table 5.56 Description of the 'resultStatusValue' attribute for the ResultStatus class.

Descriptor	Definition
Attribute name	resultStatusValue
Data type	ResultStatusValue
Value space	Vocabulary-based. This core vocabulary is given in Appendix B.
Multiplicity	1
Description	This is used to define the status of the value of the Result. The lifecycle by which the status is changed for a Result is outside the scope of this specification. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model. The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.

5.19.3 LocaleKey Attribute Description

Table 5.57 Description of the 'localeKey' attribute for the ResultStatus class.

Descriptor	Definition
Attribute name	localeKey
Data type	Normalized String
Value space	A Normalized String: [1-4095 characters].
Multiplicity	0..1
Description	A key used to look up the language-specific display name for the Result from a resource bundle.

5.19.4 DefaultDisplayName Attribute Description

Table 5.58 Description of the 'defaultDisplayName' attribute for the ResultStatus class.

Descriptor	Definition
Attribute name	defaultDisplayName
Data type	Normalized String
Value space	A Normalized String: [1-4095 characters].
Multiplicity	0..1
Description	A default value for the display name of the Result. This value will be used if the "localization" capability is not supported, or if a locale-specific display name cannot be found based on the localeKey.

5.20 ResultValue Class Description

The PIM for the ResultValue is shown in Figure 5.4.

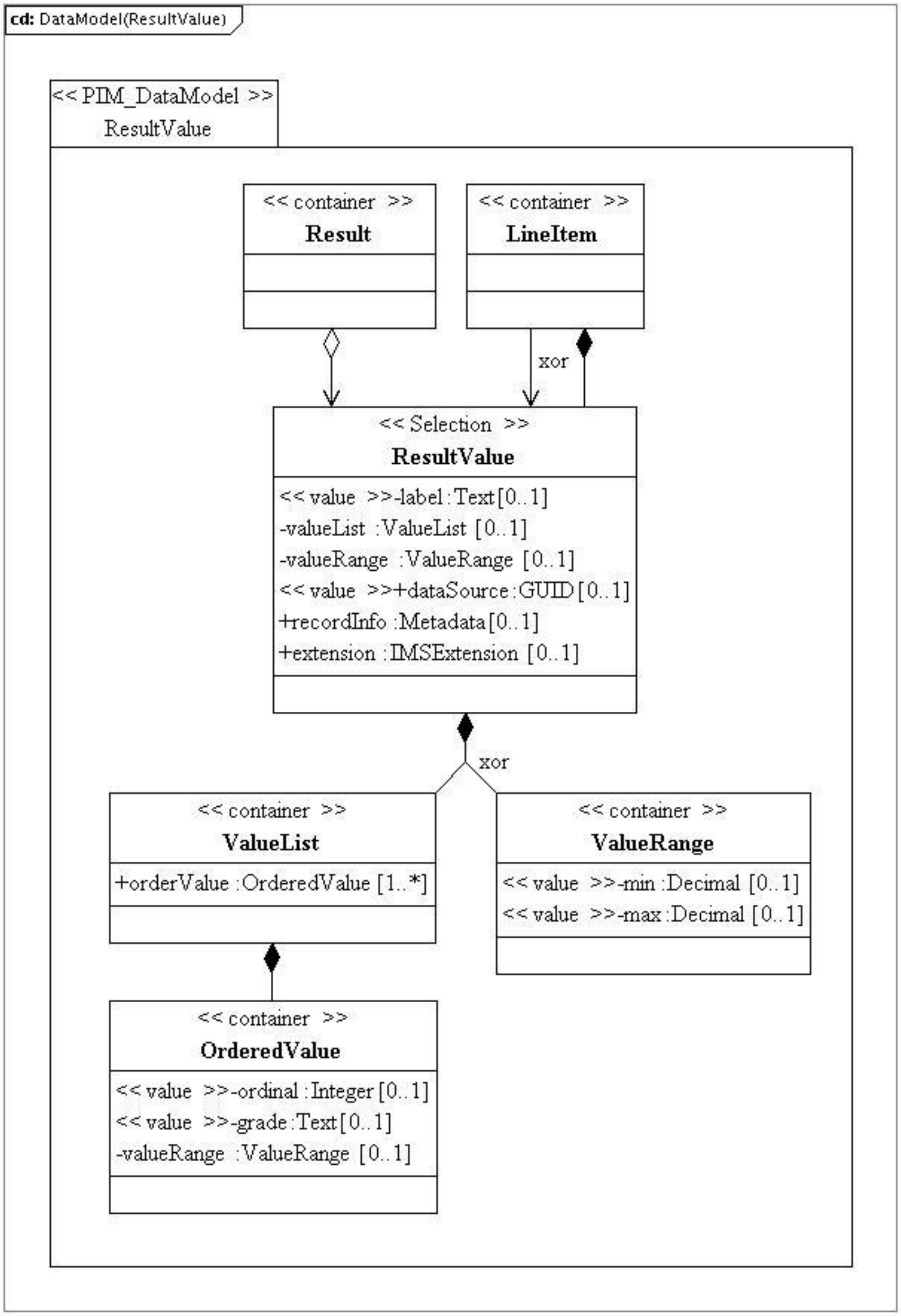

cd: DataModel(ResultValue)
<< PIM_DataModel >>
ResultValue
<< container >>
Result
<< container >>
LineItem
xor
<< Selection >>
ResultValue
<< value >>-label : Text [0..1]
-valueList : ValueList [0..1]
-valueRange : ValueRange [0..1]
<< value >>+dataSource : GUID [0..1]
+recordInfo : Metadata [0..1]
+extension : IMSExtension [0..1]
xor
<< container >>
ValueList
+orderValue : OrderedValue [1..*]
<< container >>
ValueRange
<< value >>-min : Decimal [0..1]
<< value >>-max : Decimal [0..1]
<< container >>
OrderedValue
<< value >>-ordinal : Integer [0..1]
<< value >>-grade : Text [0..1]
-valueRange : ValueRange [0..1]

Figure 5.4 ResultValue class diagram.

Table 5.59 Description of the ResultValue class.

Descriptor	Definition
Class name	ResultValue
Class type	container
Multiplicity	1
Parents	ResultValueRecord
Children	[label, valueList, valueRange, dataSource, metadata, extension], ordered
Description	The container for the information that defines the nature of the value contained in a Result value. For example this could define a list of grades, the minimum and maximum score, etc.

5.20.1 Label Attribute Description

Table 5.60 Description of the 'label' attribute for the ResultValue class.

Descriptor	Definition
Attribute name	label
Data type	Text
Value space	A language dependent String [1-63 characters]. The default language is 'en-US'.
Multiplicity	0..1
Description	Short descriptive name for interim result grading mode. Examples are: 'Letter Grade', 'Pass/Fail', etc.

5.20.2 ValueList Attribute Description

Table 5.61 Description of the 'valueList' attribute for the ResultValue class.

Descriptor	Definition
Attribute name	valueList
Data type	ValueList
Value space	container
Multiplicity	0..1 (conditional)
Description	The ordered list of permitted values for the Result e.g .'A', 'B', 'C', 'D' and 'E' and 'Fail'.

5.20.3 ValueRange Attribute Description

Table 5.62 Description of the 'valueRange' attribute for the ResultValue class.

Descriptor	Definition
Attribute name	valueRange
Data type	ValueRange
Value space	container
Multiplicity	0..1 (conditional)
Description	The definition of the minimum and maximum values for the Result value.

5.20.4 DataSource Attribute Description

Table 5.63 Description of the 'dataSource' attribute for the ResultValue class.

Descriptor	Definition
Attribute name	dataSource
Data type	GUID
Value space	See Table 5.1.
Multiplicity	0..1
Description	An identifier of the original source system of the ResultValue object.

5.20.5 RecordInfo Attribute Description

Table 5.64 Description of the 'recordInfo' attribute for the ResultValue class.

Descriptor	Definition
Attribute name	recordInfo
Data type	Metadata
Value space	container
Multiplicity	0..1
Description	The container for metadata about the ResultValue object. No particular form of metadata is mandated.

5.20.6 Extension Attribute Description

Table 5.65 Description of the 'extension' attribute for the ResultValue class.

Descriptor	Definition
Attribute name	extension
Data type	IMSExtension
Value space	container
Multiplicity	0..1
Description	The extension mechanism for the ResultValue data model.

5.21 ValueList Class Description

The PIM for the ValueList class is shown in Figure 5.4.

Table 5.66 Description of the ValueList class.

Descriptor	Definition
Class name	ValueList
Class type	container
Multiplicity	0..1
Parents	ResultValue
Children	[valueRange, orderedValue], ordered
Description	The list of ordered values that can be assigned to a Result object.

5.21.1 OrderValue Attribute Description

Table 5.67 Description of the 'orderValue' attribute for the ValueList class.

Descriptor	Definition
Attribute name	orderValue
Data type	OrderedValue
Value space	container
Multiplicity	1..unbounded, unordered
Description	The container for the list of ordered values that can be assigned to a Result object. An instance of the OrderValue class must support a least 5 instances of the orderValue attribute.

5.22 OrderedValue Class Description

The PIM for the OrderedValue class is shown in Figure 5.4.

Table 5.68 Description of the OrderedValue class.

Descriptor	Definition
Class name	OrderedValue
Class type	container
Multiplicity	1..unbounded, unordered
Parents	ValueList
Children	[ordinal, grade]
Description	A specific value in the list of permitted values.

5.22.1 Ordinal Attribute Description

Table 5.69 Description of the 'ordinal' attribute for the OrderedValue class.

Descriptor	Definition
Attribute name	ordinal
Data type	Integer
Value space	An integer in the range of: ±2^{16}-1.
Multiplicity	0..1
Description	A specific number that can be assigned to the Result object.

5.22.2 Grade Attribute Description

Table 5.70 Description of the 'grade' attribute for the OrderedValue class.

Descriptor	Definition
Attribute name	grade
Data type	Text
Value space	A language dependent String [1-15 characters]. The default language is 'en-US'.
Multiplicity	0..1
Description	A specific string that can be assigned to the Result object.

5.22.3 ValueRange Attribute Description

Table 5.71 Description of the 'valueRange' attribute for the OrderedValue class.

Descriptor	Definition
Attribute name	valueRange
Data type	ValueRange
Value space	container
Multiplicity	0..1 (conditional)
Description	The definition of the equivalent range of values for the associated grade.

5.23 ValueRange Class Description

The PIM for the ValueRange class is shown in Figure 5.4.

Table 5.72 Description of the ValueRange class.

Descriptor	Definition
Class name	ValueRange
Class type	container
Multiplicity	0..1
Parents	ResultValue, ValueList
Children	[min, max]
Description	Container for the minimum and maximum value permitted for the Result object.

5.23.1 Min Attribute Description

Table 5.73 Description of the 'min' attribute for the ValueRange class.

Descriptor	Definition
Attribute name	Min
Data type	Decimal
Value space	Real number in the range: ±32676.00.
Multiplicity	0..1
Description	The minimum value, inclusive, that can be assigned to the Result object. This value must be less that the maximum.

5.23.2 Max Attribute Description

Table 5.74 Description of the ‘max’ attribute for the ValueRange class.

Descriptor	Definition
Attribute name	Max
Data type	Decimal
Value space	Real number in the range: ±32676.00.
Multiplicity	0..1
Description	The minimum value, inclusive, that can be assigned to the Result object. This value must be greater than the minimum.

5.24 Common Classes Descriptions

The PIM for the common classes is shown in Figure 5.5.

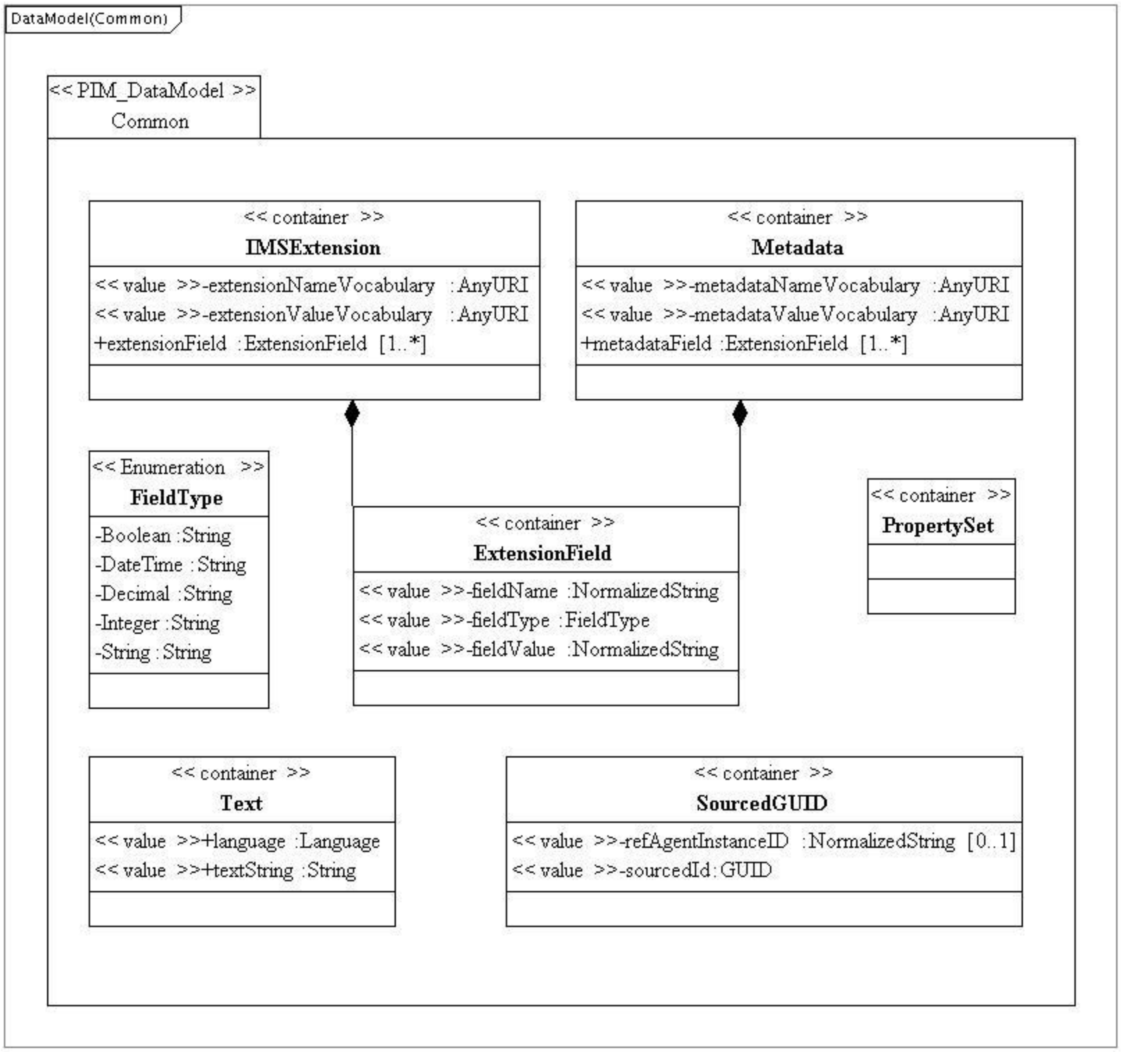

DataModel(Common)
<< PIM_DataModel >>
Common
<< container >>
IMSExtension
<< value >>-extensionNameVocabulary :AnyURI
<< value >>-extensionValueVocabulary :AnyURI
+extensionField :ExtensionField [1..*]
<< container >>
Metadata
<< value >>-metadataNameVocabulary :AnyURI
<< value >>-metadataValueVocabulary :AnyURI
+metadataField :ExtensionField [1..*]
<< Enumeration >>
FieldType
-Boolean :String
-DateTime :String
-Decimal :String
-Integer :String
-String :String
<< container >>
ExtensionField
<< value >>-fieldName :NormalizedString
<< value >>-fieldType :FieldType
<< value >>-fieldValue :NormalizedString
<< container >>
PropertySet
<< container >>
Text
<< value >>+language :Language
<< value >>+textString :String
<< container >>
SourcedGUID
<< value >>-refAgentInstanceID :NormalizedString [0..1]
<< value >>-sourcedId :GUID

Figure 5.5 Common class diagram.

5.24.1 Metadata Class Description

The PIM for the Metadata class is shown in Figure 5.5.

Table 5.75 Description of the Metadata class.

Descriptor	Definition
Class name	Metadata
Class type	container
Children	[metadataNameVocabulary, metadataTypeVocabulary, metadataField], ordered
Description	This is the container for meta-data about the corresponding object. The meta-data entries are supplied using a name/type/value triple based upon external vocabularies.

Table 5.76 Description of the 'metadataNameVocabulary' attribute for the Metadata class.

Descriptor	Definition
Attribute name	metadataNameVocabulary
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldName values. If this is a reference to a VDEX file it is the 'vocabIdentifier' of the vocabulary.

Table 5.77 Description of the 'metadataTypeVocabulary' attribute for the Metadata class.

Descriptor	Definition
Attribute name	metadataTypeVocabulary
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldType values. If this is a reference to a VDEX file it is the 'vocabIdentifier' of the vocabulary.

Table 5.78 Description of the ‘metadataField’ attribute for the Metadata class.

Descriptor	Definition
Attribute name	metadataField
Data type	ExtensionField
Value space	container
Multiplicity	1..unbounded, unordered
Description	The container for the triples that are used to define each extension data element.

5.24.2 IMSExtension Class Description

The PIM for the IMSExtension class is shown in Figure 5.5.

Table 5.79 Description of the IMSExtension class.

Descriptor	Definition
Class name	IMSExtension
Class type	container
Children	[extensionNameVocabulary, extensionTypeVocabulary, extensionField], ordered
Description	The container for the extension of the Group data model.

Table 5.80 Description of the ‘extensionNameVocabulary’ attribute for the IMSExtension class.

Descriptor	Definition
Attribute name	extensionNameVocabulary
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldName values. If this is a reference to a VDEX file it is the ‘vocabIdentifier’ of the vocabulary.

Table 5.81 Description of the 'extensionTypeVocabulary' attribute for the IMSExtension class.

Descriptor	Definition
Attribute name	extensionTypeVocabulary
Data type	AnyURI
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that is used to define the set of permitted fieldType values. If this is a reference to a VDEX file it is the 'vocabIdentifier' of the vocabulary.

Table 5.82 Description of the 'extensionField' attribute for the IMSExtension class.

Descriptor	Definition
Attribute name	extensionField
Data type	ExtensionField
Value space	container
Multiplicity	1..unbounded, unordered
Description	The container for the triples that are used to define each extension data element.

5.24.3 ExtensionField Class Description

The PIM for the ExtensionField class is shown in Figure 5.5.

Table 5.83 Description of the ExtensionField class.

Descriptor	Definition
Class name	ExtensionField
Class type	container
Children	None.
Description	The container for each triple that describes an extension field. Each triple consists of field name, field type and field value.

Table 5.84 Description of the 'fieldName' attribute for the ExtensionField class.

Descriptor	Definition
Attribute name	fieldName
Data type	NormalizedString
Value space	A language dependent Normalized String [1-127 characters]. Default language is 'en-US'.
Multiplicity	1
Description	The container for the name of the extension field. This is used to identify the full triple.

Table 5.85 Description of the 'fieldType' attribute for the ExtensionField class.

Descriptor	Definition
Attribute name	fieldType
Data type	Enumerated vocabulary.
Value space	Vocabulary-based. The core vocabulary is given in Appendix B.
Multiplicity	1
Description	This defines the data-type for the extension triple. The value space for this vocabulary is approved by IMS GLC and made available to the public as defined in [SDN11, 06]. The syntax and semantics of the approved list of terms shall be supported by all software components implementing this Information Model. The value space for the vocabulary may be extended. Such extending expressions may be created and used only when no approved IMS GLC value satisfies the expressive need of an implementing community to define the shape of a collection.

Table 5.86 Description of the 'fieldValue' attribute for the ExtensionField class.

Descriptor	Definition
Attribute name	fieldValue
Data type	NormalizedString
Value space	A language dependent Normalized String [1-127 characters]. Default language is 'en-US'.
Multiplicity	1
Description	The container for the data value itself. This is stored as a string but should be interpreted as per the data-type defined in the 'fieldType' part of the triple.

5.24.4 Text Class Description

Table 5.87 Description of the ‘Text’ class.

Descriptor	Definition
Class name	Text
Class type	container
Children	n/a
Description	Text to be stored. This is a language/string tuple.

Table 5.88 Description of the ‘language’ attribute for the Text class.

Descriptor	Definition
Attribute name	language
Data type	Enumerated.
Value space	RFC4646 language codes. The default value is: ‘en-US’. See Appendix B for more details on this vocabulary.
Multiplicity	1
Description	The language for the associated text string.

Table 5.89 Description of the ‘textString’ attribute for the Text class.

Descriptor	Definition
Attribute name	textString
Data type	String
Value space	String [1-4095 characters].
Multiplicity	1
Description	The container for the string.

5.24.5 SourcedGUID Class Description

Table 5.90 Description of the SourcedGUID class.

Descriptor	Definition
Class name	SourcedGUID
Class type	container
Children	[refAgentInstanceID, sourcedId], ordered
Description	This is a structured GUID that consists of an instance identifier and a sourcedId.

Table 5.91 Description of the 'refAgentInstanceID' attribute for the SourcedGUID class.

Descriptor	Definition
Attribute name	refAgentInstanceID
Data type	NormalizedString
Value space	Normalized string [1..31 characters].
Multiplicity	0..1
Description	This is an instance identifier used to differentiate, if necessary, between multiple end system reference agents.

Table 5.92 Description of the 'sourcedId' attribute for the SourcedGUID class.

Descriptor	Definition
Attribute name	sourcedId
Data type	GUID
Value space	See Table 5.1.
Multiplicity	1
Description	The sourcedId for the object. This should be a GUID.

5.24.6 PropertySet Class Description

Table 5.93 Description of the PropertySet class.

Descriptor	Definition
Class name	PropertySet
Class type	container
Children	See the IMS GLC LTI v1.0 Specification.
Description	See the IMS GLC LTI v1.0 Specification.

6 Extending and Profiling the Service

6.1 Proprietary Extensions

Proprietary extensions of the service are based upon two approaches:

a) The extension of the data models being manipulated by the current set of operations;

b) The inclusion of new operations to support new proprietary functionality.

It is NOT permitted to change the Behavior of the current set of operations. Such changes MUST be supported by the creation of new operations.

6.1.1 Proprietary Operations

The definition of new operations should follow the same format as adopted herein. The new operations should be defined using a new interface type. Every operation must result in the return of a status code that describes the final state of the request on the target end system.

An example of creating such an extension is given in the accompanying Best Practices document [LIS, 11c].

6.1.2 Proprietary Data Elements

Extensions to the data model are only permitted where the *IMSExtension* class is available. Within the Outcomes data model only the ‘LineItem’, ‘Result’ and ‘ResultValue’ classes can be extended. The extension takes the form of a Name/Type/Value triple. Many extension fields can be added but hierarchical structures must be emulated using the appropriate delimited notation in the ‘Name’ field. This triple consists of:

- Name – the name assigned to the extension field (this is a string that can support any naming convention);
- Type – the data-type that is to be used for the value (this is used for interpreting the associated value);
- Value – the data value for the extension (the value is supplied as a string).

6.2 Profiling the Service

This Service can be profiled. In general, Profiling is used to:

a) Refine which Interfaces are used and which operations are supported for each Interface;

b) Refine the data models (see the IMS GLC Application Profiling guidelines for more details on how data models can be profiled [APG, 05a][APG, 05b]).

Valid Profiles must be restrictive i.e., optional features can be removed or constraints increased but new features must not be added. A Profile of this service is made by annotating the UML supplied with the documentation for the specification.

Appendix A – Summary List of Service Status Codes

The summary list of status codes that can be returned by the different operations through the StatusInfo object is given in Table. A1.1. The key to the entries is: ‘Y’ denotes the code may be returned by that operation. A blank entry means that the code cannot be returned by that operation.

Table A1.1 Status codes for the service operations.

CodeMinor Status Code	create[1]	createByProxy[2]	delete[3]	read[4]	update[5]	replace[6]	discover[8]	changeIdentifier[7]
‘fullsuccess’	Y	Y	Y	Y	Y	Y	Y	Y
‘createsuccess’						Y		
‘nosourcedids’				Y				
‘idallocfail’		Y						
‘overflowfail’	Y	Y						
‘idallocinusefail’	Y							Y
‘invaliddata’	Y	Y			Y	Y	Y	
‘contextunknown’	Y	Y			Y	Y		
‘toomuchdata’				Y				
‘invalidlineitemtype’	Y	Y		Y	Y	Y		
‘gradingnotpermitted’	Y	Y			Y	Y		
‘incompletedata’	Y	Y		Y	Y	Y		
‘partialdatastorage’	Y	Y			Y	Y		
‘unknownobject’			Y	Y	Y	Y		Y
‘unknownquery’							Y	
‘deletefailure’			Y					
‘unknownvocabulary’	Y	Y		Y	Y	Y		
‘unknownmdvocabulary’	Y	Y			Y	Y		
‘targetreadfailure’				Y				
‘savepointerror’				Y				
‘savepointsyncerror’				Y				
‘unknownextension’	Y	Y		Y	Y	Y		
‘targetisbusy’	Y	Y	Y	Y	Y	Y	Y	Y
‘linkfailure’	Y	Y	Y	Y	Y	Y	Y	Y

CodeMinor Status Code	**create**[1]	**createByProxy**[2]	**delete**[3]	**read**[4]	**update**[5]	**replace**[6]	**discover**[8]	**changeIdentifier**[7]
‘unauthorizedrequest’	Y	Y	Y	Y	Y	Y	Y	Y
‘unsupportedLIS’	Y	Y	Y	Y	Y	Y	Y	Y
‘unsupportedLISoperation’	Y	Y	Y	Y	Y	Y	Y	Y

Notes:

[1] Used by operations: ‘createLineItem’, ‘createResult’ and ‘createResultValue’;

[2] Used by operations: ‘createByProxyLineItem’, ‘createByProxyResult’ and ‘createByProxyResultValue’;

[3] Used by operations: ‘deleteLineItem’, ‘deleteResult’ and ‘deleteResultValue’;

[4] Used by operations: ‘readLineItem’, ‘readResult’, ‘readResultValue’, ‘readAllLineItemIds’, ‘readLineItemIdsForPerson’, ‘readLineItemIdsForCourseOffering’, ‘readLineItemIdsForCourseSection’, ‘readLineItemIdsWithStatus’, ‘readAllResultIds’, ‘readResultIdsForPerson’, ‘readResultIdsForLineItem’, ‘readResultIdsForCourseOffering’, ‘readResultIdsForCourseSection’, ‘readLineItems’, ‘readResults’, ‘readResultValues’, ‘readResultIdsForCourseSectionWithStatus’, ‘readAllResultValueIds’, ‘readResultValueIdForLineItem’, ‘readResultValueIdForResult’, ‘readLineItemIdsFromSavePoint’, ‘readResultIdsFromSavePoint’, ‘readResultValueIdsFromSavePoint’, ‘readLineItemsFromSavePoint’, ‘readResultsFromSavePoint’ and ‘readResultValuesFromSavePoint’;

[5] Used by operations: ‘updateLineItem’, ‘updateResult’ and ‘updateResultValue’;

[6] Used by operations: ‘replaceLineItem’, ‘replaceResult’ and ‘replaceResultValue’;

[7] Used by operations: ‘changeLineItemIdentifier’, ‘changeResultIdentifier’ and ‘changeResultValueIdentifier’;

[8] Used by operations: ‘discoverLineItem’, ‘discoverResult’ and ‘discoverResultValue’.

There is a set of status codes that must be supported by each of the Outcomes Management Service operations. These codes are described in Table A1.2.

Table A1.2 Common status codes for the service operations.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unauthorizedrequest’	The source system is not authorized to make this request of the target. The reason for the refusal can be one of several causes.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=targetisbusy’	The target end-system received the request but is busy and cannot process the request. The request should be resubmitted.
‘CodeMajor=Failure’ ‘Severity=Error’ ‘CodeMinor=linkfailure’	There has been a failure in the end-to-end system communications mechanism and so the request has not been delivered.
‘CodeMajor=unsupportedLIS’ ‘Severity=Status’ ‘CodeMinor=*’	This service in LIS is not supported by the target system. Every system that implements any part of the LIS specification **must** return this status code for a service component in LIS that is not supported.
‘CodeMajor=unsupportedLISoperation’ ‘Severity=Status’ ‘CodeMinor=*’	This operation is not supported by the target system. Every system that implements any part of the LIS specification **must** return this status code for an operation that is not supported in a supported service.

Appendix B – Vocabularies

B1 Set of Defined Vocabularies

The vocabularies listed in Tables B1.1, B1.2 and B1.3 are the default set maintained under the IMS GLC Vocabulary Registry [SDN11, 06]. It is the responsibility of an implementation to ensure that it is using the correct and latest versions of the vocabulary files. Changes to the default vocabularies are permitted; this results in the creation of a new vocabulary that should be registered with IMS GLC. As part of a profiling process entirely new vocabularies may be defined to replace the default set.

The set of external vocabularies that are used in this information model are listed in Table B1.1, B1.2 and B1.3.

B1.1 LineItemType Vocabulary

The vocabulary for the 'lineItemType' is listed in Table B1.1.

Table B1.1 The lineItemType external vocabulary.

Vocabulary	Description
LineItem Class 'lineItemType' attribute	Data types that are permitted for the type/status of lineItem fields. An implementation must support the 'Final' and 'Interim' values. The core vocabulary is: • MidTerm • Interim • Final

Rule B.1-01: Midterm – to denote that the lineItem consists of a set of Midterm results for the course component;

Rule B.1-02: Interim – to denote that the lineItem consists of a set of Interim results for the course component;

Rule B.1-03: Final – to denote that the lineItem consists of a set of Final results for the course component.

B1.1 StatusofResult Vocabulary

The vocabulary for the 'statusofResult' is listed in Table B1.2.

Table B1.2 The result status external vocabulary.

Vocabulary	Description
Result Class 'statusofResult' attribute	Data types that are permitted for the status of the value of the Result. The core vocabulary is: • Unmoderated • Tobemoderated • Pending • Completed

Rule B.2-01: Unmoderated – to denote that the result is unmoderated;

Rule B.2-03: Tobemoderated – to denote that the result is to be moderated;

Rule B.2-03: Pending – to denote that the result is of an interim nature;

Rule B.2-04: Completed – to denote that the result is Final.

B1.3 ReplaceStatus Vocabulary

The vocabulary for the 'replaceStatus' is listed in Table B1.3.

Table B1.3 The replaceStatus external vocabulary.

Vocabulary	Description
ReplaceStatusCode Class 'replaceStatus' attribute	The status codes that may be returned to indicate the state of the attempted Result object replacement. Enumerated as: • Fullsuccess • Createsuccess • Unknownlineitem • Unknownperson • Contextunknown • Gradingnotpermitted • Invalidresult • Resultalreadyposted • Incompletedata • Partialdatastorage • Unknownvocabulary • Unknownmdvocabulary • Unknownextension

Rule B.3-01: Fullsuccess – the result replacement was successful;

Rule B.3-02: Createsuccess – a new result object was successfully created;

Rule B.3-03: Unknownlineitem – the LineItem object is unknown;

Rule B.3-04: Unknownperson – the associated Person object is unknown;

Rule B.3-05: Contextunknown – the associated context identification is unknown;

Rule B.3-06: Gradingnotpermitted – the associated Course object is not gradable;

Rule B.3-07: Invalidresult – grade is not valid for this enrollment;

Rule B.3-08: Resultalreadyposted – the received grade has already been posted to student enrollment

Rule B.3-09: Incompletedata – some mandatory part of the data has been detected as missing by the target system;

Rule B.3-10: Partialdatastorage – the target has stored a subset of the sent data record i.e., some of the optional data has not been stored (all mandatory data has been supplied and stored);

Rule B.3-11: Unknownvocabulary – one or more vocabulary values is unknown;

Rule B.3-12: Unknownmdvocabulary – one or more metadata vocabulary values is unknown;

Rule B.3-13: Unknownextension – the target cannot process the proprietary data model extensions used in the object.

B1.4 FieldType Vocabulary

The vocabulary for type of field in 'fieldType' is listed in Table B1.4.

Table B1.4 The fieldType external vocabulary.

Vocabulary	Description
ExtensionField Class 'fieldType' attribute	Data types that are permitted for the extension fields. These data-types reflect the permitted types for XML. Enumerated as: • Boolean • DateTime • Integer • Decimal • String

Rule B.4-01: Boolean – the data-type is equivalent to the definition of a 'Boolean' in XML;

Rule B.4-02: DateTime – the data-type is equivalent to the definition of a 'DateTime' in XML;

Rule B.4-03: Integer – the data-type is equivalent to the definition of an 'Integer' in XML;

Rule B.4-04: Decimal – the data-type is equivalent to the definition of a 'Decimal' in XML;

Rule B.4-05: String – the data-type is equivalent to the definition of a 'String' in XML.

B1.5 Language Vocabulary

The language code/country code combination used to identify the language for a piece of text is an enumerated external IMS GLC vocabulary that captures the full set of entries from RFC4646.

B2 Using Vocabularies for the Metadata Class

The Metadata class consists of attributes:

- metadataNameVocabulary – identifies the vocabulary that contains the reference set of fieldName values for the meta-data[2];
- metadataTypeVocabulary – identifies the vocabulary that contains the reference set of fieldType values for the meta-data. The value for this attribute is the same as the 'vocabIdentifier' of the VDEX instance for this vocabulary;
- metadataField – contains the set of triples (fieldName/fieldType/fieldValue) for each meta-data entry.

The value in the 'fieldName' must be from the vocabulary (identified using the metadataNameVocabulary attribute). The value in the 'fieldType' must be from the external vocabulary containing the permitted set of external field types (as listed in the metadataTypeVocabulary attribute). The value in the 'fieldValue' is the metadata value itself. Nested values are possible using a dot notation in the 'fieldName' e.g., for LOM this could be 'general.keyword.string', etc.

[2] The corresponding vocabulary must be defined. It is recommended that the vocabulary registered with IMS GLC made available as a VDEX file. If the vocabulary is defined as a VDEX file then the value for 'metadataNameVocabulary' should be the 'vocabIdentifier' of the VDEX instance.

B3 Using Vocabularies for the IMSExtension Class

The IMSExtension class consists of attributes:

- extensionNameVocabulary[3] – identifies the vocabulary that contains the reference set of fieldName values for the extension;
- extensionTypeVocabulary – identifies the vocabulary that contains the reference set of fieldType values for the extension. The value for this attribute is the same as the 'vocabIdentifier' of the VDEX instance for this vocabulary;
- extensionField – contains the set of triples (fieldName/fieldType/fieldValue) for each extension.

The value in the 'fieldName' must be from the vocabulary (identified using the extensionNameVocabulary attribute). The value in the 'fieldType' must be from the external vocabulary containing the permitted set of external field types (as listed in the extensionTypeVocabulary attribute). The value in the 'fieldValue' is the extension value itself. Nested values are possible using a dot notation in the 'fieldName' cf. for meta-data.

[3] The corresponding vocabulary must be defined. It is recommended that the vocabulary registered with IMS GLC made available as a VDEX file. If the vocabulary is defined as a VDEX file then the value for 'extensionNameVocabulary' should be the 'vocabIdentifier' of the VDEX instance.

Appendix C – File-based Data Exchange

The IMS GLC Bulk Data Exchange Management Service [BDEMS, 11] is used to exchange bulk Outcomes and related information. The Outcomes information is exchanged by placing multiple *LineItemRecord*, *ResultRecord* and *ResultValueRecord* structures within a bulk data container. Figure C1.1 shows the key class structure and the associated definitions are provided in Section 5 of this document.

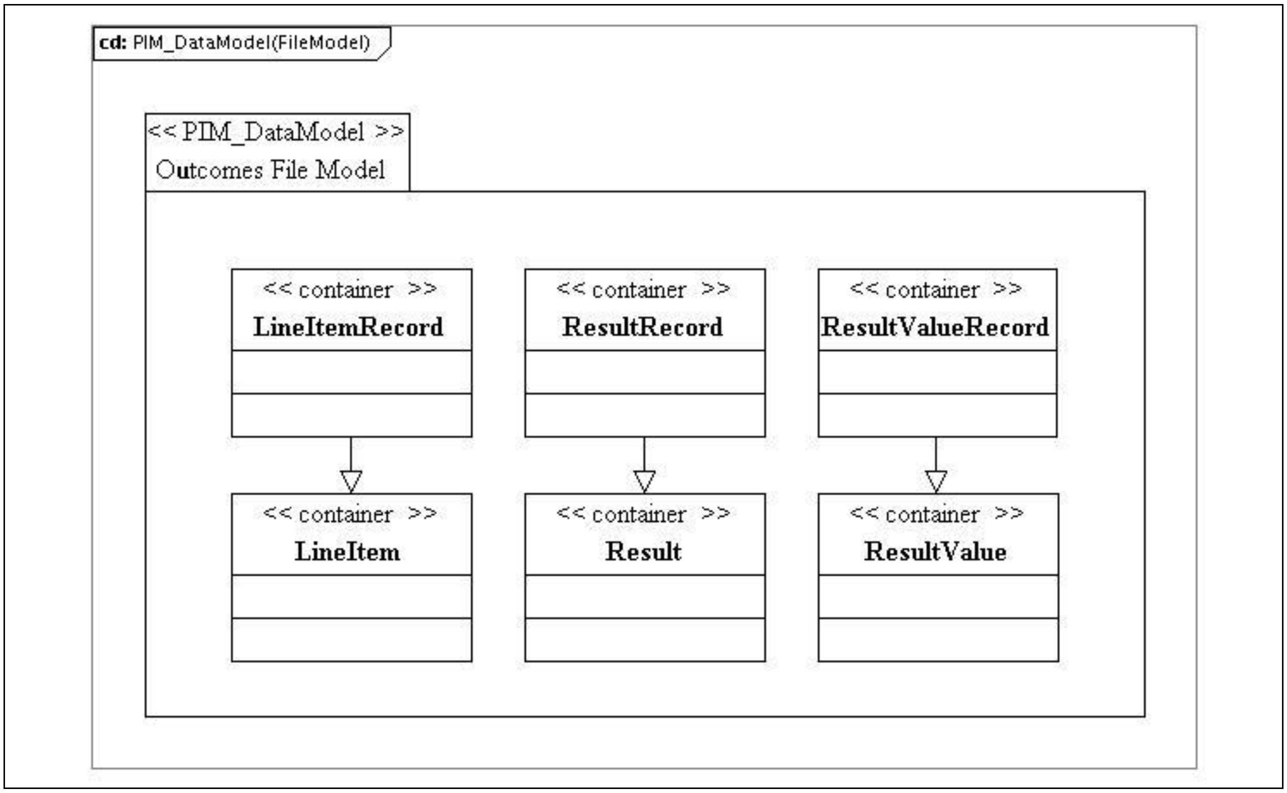


Figure C.1 Outcomes class diagram for file-based data exchange.

Note that separate binding instance validation files will be used for containing the description of LineItemRecords, ResultRecords and ResultValueRecords within a file. This ensures that only the required data structures are contained in the corresponding binding validation files.

About This Document

Title:	IMS GLC Outcomes Management Service Information Model
Editor:	Colin Smythe (IMS GLC)
Co-chairs:	Linda Feng (Oracle) and Bill Lee (Desire2Learn)
Version:	1.0
Version Date:	30 June 2011
Release:	Public Draft v2.0
Status:	**Public Draft**
Summary:	This document contains the IMS GLC Outcomes Management Service v1.0 Information Model. This service is used to exchange information about Outcomes. The Outcomes data model is based upon results that can be grouped together to reflect an assessment activity for a Course. The Outcomes data model is based upon LineItems, Results and ResultValues. The business transactions include the simple create, read, update, delete and simple searching of the Outcomes data model. This document contains the definition of the abstract application-programming interface for the Outcomes Management Service. This is a new service that is based upon the interim and final results parts of the data model in the IMS GLC Membership Management Service v1.0 specification.
Revision Information:	Original release.
Purpose:	This document is made available for review by the public community at large.
Document Location:	http://www.imsglobal.org/lis/

List of Contributors

The following individuals contributed to the development of this document:

Kerry Blinco	DEEWR (Australia)	Zack Leavitt	Pearson (USA)
Kirk Bunte	SungardHE (USA)	Bill Lee	Desire2Learn (Canada)
Angus Chan	Desire2Learn (Canada)	Richard Moon	SungardHE (USA)
Adam Cooper	JISC/JISC-CETIS (UK)	Mike Parkhill	Desire2Learn (Canada)
Michael Feldstein	Oracle (USA)	Colin Smythe	IMS GLC (UK)
Linda Feng	Oracle (USA)	Reinhold Staudinger	Blackboard (USA)
Chris Hatton	Pearson (USA)	Nick Terrible	University of Wisconsin (USA)
Jon Fontaine	Blackboard (USA)	Jason Zhong	SungardHE (USA)
Karen Kuffner	University of Michigan (USA)		

Revision History

Version No.	Release Date	Comments
Final Release v1.0	30 June 2011	The first formal release of the Final Release version of this document.

Index

IMS Global Learning Consortium, Inc. ("IMS GLC") is publishing the information contained in this document ("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 OMS Information Model v2.0 Public Draft v2.0

Date: 30 June 2011