



IMS GLC Bulk Data Exchange Management Service (BDEMS) Information Model Version 1.0

Final Release Version 1.0

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.
Documents Name: IMS GLC BDEMS Information Model – Revision: 30 June 2011

Table of Contents

LIST OF FIGURES.....	5
LIST OF TABLES.....	5
1 INTRODUCTION	9
1.1 BULK DATA EXCHANGE MANAGEMENT SERVICE OVERVIEW	9
1.2 SCOPE AND CONTEXT	9
1.3 STRUCTURE OF THIS DOCUMENT	10
1.4 NOMENCLATURE.....	10
1.5 REFERENCES	11
2 BULK DATA EXCHANGE MANAGEMENT SERVICE DESCRIPTION.....	12
2.1 AN ABSTRACT REPRESENTATION	12
2.2 BULK DATA EXCHANGE MANAGEMENT SERVICE ARCHITECTURE & SPECIFICATION MODEL.....	12
2.3 BULK DATA FILE	13
2.4 HANDLING THE SERVICE STATUS CODES	14
3 BEHAVIORAL MODEL	15
3.1 SERVICE DEFINITION	15
3.2 BULKDATAEXCHANGEMANAGER INTERFACE DESCRIPTION.....	16
3.2.1 <i>AnnounceBulkDataExchange() Operation.....</i>	<i>17</i>
3.2.2 <i>AnnounceFailureBulkDataExchange() Operation.....</i>	<i>19</i>
3.2.3 <i>ReportBulkDataExchange() Operation.....</i>	<i>20</i>
3.2.4 <i>RequestBulkDataExchange() Operation.....</i>	<i>21</i>
3.2.5 <i>IgnoreBulkDataExchange() Operation.....</i>	<i>22</i>
3.2.6 <i>CancelBulkDataExchange() Operation</i>	<i>23</i>
3.3 SERVICE CHOREOGRAPHY	24
4 INTERFACE DATA MODEL.....	27
4.1 BULKBLOCKMANIFEST CLASS DESCRIPTION	27
4.1.1 <i>BulkBlockManifestId Attribute Description.....</i>	<i>28</i>
4.1.2 <i>ExpiryDate Attribute Description</i>	<i>28</i>
4.1.3 <i>BulkBlockDataFile Attribute Description.....</i>	<i>29</i>
4.1.4 <i>BulkBlockDataFile Class Description</i>	<i>29</i>
4.1.5 <i>Url Attribute Description.....</i>	<i>29</i>
4.1.6 <i>Checksum Attribute Description.....</i>	<i>30</i>
4.1.7 <i>TotalSize Attribute Description.....</i>	<i>30</i>
4.1.8 <i>SavePoint Attribute Description</i>	<i>30</i>
4.1.9 <i>ServiceSet Attribute Description.....</i>	<i>31</i>
4.1.10 <i>ServiceSet Class Description</i>	<i>31</i>
4.1.11 <i>ServiceRecord Class Description.....</i>	<i>32</i>
4.1.12 <i>OperationSet Class Description.....</i>	<i>33</i>
4.2 BULKBLOCKREPORT CLASS DESCRIPTION	35

4.2.1	<i>ReportSummary Class Description</i>	37
4.2.2	<i>InterfaceSummaryReort Class Description</i>	39
4.2.3	<i>ReportFailureDetail Class Description</i>	40
4.2.4	<i>TransactionReport Class Description</i>	41
4.3	FAILUREREPORT CLASS DESCRIPTION	43
4.4	FILTEROBJECT CLASS DESCRIPTION	44
4.5	SEQUENCEIDENTIFIER CLASS DESCRIPTION.....	47
4.6	SUPPORTEDSERVICES CLASS DESCRIPTION	48
4.6.1	<i>PersonInterface Class Description</i>	52
4.6.2	<i>GroupInterface Class Description</i>	53
4.6.3	<i>MembershipInterface Class Description</i>	54
4.6.4	<i>TemplateInterface Class Description</i>	54
4.6.5	<i>OfferingInterface Class Description</i>	55
4.6.6	<i>SectionInterface Class Description</i>	56
4.6.7	<i>AssociationInterface Class Description</i>	56
4.6.8	<i>LineItemInterface Class Description</i>	57
4.6.9	<i>ResultInterface Class Description</i>	58
4.6.10	<i>ResultValueInterface Class Description</i>	58
4.7	STATUSINFO CLASS DESCRIPTION	59
5	BULK DATA FILE DATA MODEL	60
5.1	KEY TERMS AND CONCEPTS	60
5.2	BULKDATARECORD CLASS DESCRIPTION	63
5.2.1	<i>TransactionRecord Attribute Description</i>	64
5.3	TRANSACTIONRECORD CLASS DESCRIPTION	65
5.3.1	<i>TransactionIdentifier Attribute Description</i>	65
5.3.2	<i>ServiceName Attribute Description</i>	66
5.3.3	<i>InterfaceName Attribute Description</i>	66
5.3.4	<i>OperationName Attribute Description</i>	67
5.3.5	<i>ParameterSet Attribute Description</i>	67
5.4	PARAMETERSET CLASS DESCRIPTION.....	67
5.4.1	<i>ParameterRecord Attribute Description</i>	68
5.5	PARAMETERRECORD CLASS DESCRIPTION	68
5.5.1	<i>ParameterInvoc Attribute Description</i>	68
5.5.2	<i>ParameterName Attribute Description</i>	69
5.5.3	<i>ParameterType Attribute Description</i>	69
5.5.4	<i>ParameterValue Attribute Description</i>	70
5.6	PARAMETERVALUE CLASS DESCRIPTION	70
5.6.1	<i>Guid Attribute Description</i>	71
5.6.2	<i>GuidSet Attribute Description</i>	71
5.6.3	<i>QueryObject Attribute Description</i>	71
5.6.4	<i>SequenceIdentifier Attribute Description</i>	72
5.6.5	<i>Text Attribute Description</i>	72

5.6.6	<i>URL Attribute Description</i>	73
5.6.7	<i>PersonRecord Attribute Description</i>	73
5.6.8	<i>PersonCoreRecord Attribute Description</i>	73
5.6.9	<i>PersonRecordSet Attribute Description</i>	74
5.6.10	<i>GroupRecord Attribute Description</i>	74
5.6.11	<i>GroupRecordSet Attribute Description</i>	74
5.6.12	<i>MembershipRecord Attribute Description</i>	75
5.6.13	<i>MembershipRecordSet Attribute Description</i>	75
5.6.14	<i>CourseTemplateRecord Attribute Description</i>	75
5.6.15	<i>CourseTemplateRecordSet Attribute Description</i>	76
5.6.16	<i>CourseOfferingRecord Attribute Description</i>	76
5.6.17	<i>CourseSectionRecord Attribute Description</i>	76
5.6.18	<i>CourseSectionRecordSet Attribute Description</i>	77
5.6.19	<i>SectionAssociationRecord Attribute Description</i>	77
5.6.20	<i>SectionAssociationRecordSet Attribute Description</i>	77
5.6.21	<i>ResultRecord Attribute Description</i>	78
5.6.22	<i>ResultRecordSet Attribute Description</i>	78
5.6.23	<i>ResultValueRecord Attribute Description</i>	78
5.6.24	<i>ResultValueRecordSet Attribute Description</i>	79
5.6.25	<i>LineItemRecord Attribute Description</i>	79
5.6.26	<i>LineItemRecordSet Attribute Description</i>	79
5.6.27	<i>LineItemType Attribute Description</i>	80
5.6.28	<i>MembershipIdType Attribute Description</i>	80
5.6.29	<i>Relationship Attribute Description</i>	80
5.6.30	<i>Role Attribute Description</i>	81
5.6.31	<i>ResultStatus Attribute Description</i>	81
5.6.32	<i>Status Attribute Description</i>	81
6	FILE RETRIEVAL APPROACHES	82
6.1	HTTPS GET	82
6.2	OTHER PROTOCOLS.....	82
6.2.1	File Transfer Protocol over SSL (FTPS)	82
7	EXTENDING AND PROFILING THE SERVICE	83
7.1	PROPRIETARY EXTENSIONS.....	83
7.1.1	Proprietary Operations.....	83
7.1.2	Proprietary Data Elements	83
7.2	PROFILING THE SERVICE	83
	APPENDIX A SERVICE STATUS CODES	84
	APPENDIX B THE VOCABULARIES	86
B1	TRANSACTIONFAILSTATUS VOCABULARY	86
B2	PARAMETERTYPE VOCABULARY	87
B3	FILEVALUE VOCABULARY.....	89
B4	FAILURECODE VOCABULARY	90

ABOUT THIS DOCUMENT	91
LIST OF CONTRIBUTORS.....	91
REVISION HISTORY	92
INDEX	93

List of Figures

FIGURE 2.1 BULK DATA EXCHANGE MANAGEMENT SERVICE ARCHITECTURE MODEL.	12
FIGURE 2.2 COMPOSITION OF A BULK DATA FILE.	13
FIGURE 3.1 BULKDataExchangeManagementService INTERFACE DEFINITION.	15
FIGURE 3.2 SERVICE PROVIDER DRIVEN ASYNCHRONOUS SERVICE CHOREOGRAPHY.	24
FIGURE 3.3 SERVICE PROVIDER DRIVEN ASYNCHRONOUS SERVICE CHOREOGRAPHY.	25
FIGURE 4.1 BulkBlockManifest CLASS DIAGRAM.	27
FIGURE 4.2 BulkBlockReport CLASS DIAGRAM.	35
FIGURE 4.3 FilterObject CLASS DIAGRAM.	44
FIGURE 4.4 SupportedServices CLASS DIAGRAM.	48
FIGURE 5.1 BulkDataRecord CLASS DIAGRAM.	63

List of Tables

TABLE 3.1 SUMMARY OF OPERATIONS FOR BULKDataExchangeManager.	16
TABLE 3.2 STATUS CODES FOR THE ‘ANNOUNCEBulkDataExchange’ OPERATION.	17
TABLE 3.3 STATUS CODES FOR THE ‘ANNOUNCEFailureBulkDataExchange’ OPERATION.	19
TABLE 3.4 STATUS CODES FOR THE ‘REPORTBulkDataExchange’ OPERATION.	20
TABLE 3.5 STATUS CODES FOR THE ‘REQUESTBulkDataExchange’ OPERATION.	21
TABLE 3.6 STATUS CODES FOR THE ‘IGNOREBulkDataExchange’ OPERATION.	22
TABLE 3.7 STATUS CODES FOR THE ‘CANCELBulkDataExchange’ OPERATION.	23
TABLE 4.1 DESCRIPTION OF THE ‘BULKBlockManifest’ CLASS.	28
TABLE 4.2 DESCRIPTION OF THE ‘BULKBlockManifestId’ ATTRIBUTE FOR THE BULKBlockManifest CLASS.	28
TABLE 4.3 DESCRIPTION OF THE ‘EXPIRYDATE’ ATTRIBUTE FOR THE BULKBlockManifest CLASS.	28
TABLE 4.4 DESCRIPTION OF THE ‘BULKBlockDataFile’ ATTRIBUTE FOR THE BULKBlockManifest CLASS.	29
TABLE 4.5 DESCRIPTION OF THE ‘BULKBlockDataFile’ CLASS.	29
TABLE 4.6 DESCRIPTION OF THE ‘URL’ ATTRIBUTE FOR THE BULKBlockDataFile CLASS.	29
TABLE 4.7 DESCRIPTION OF THE ‘CHECKSUM’ ATTRIBUTE FOR THE BULKBlockDataFile CLASS.	30
TABLE 4.8 DESCRIPTION OF THE ‘TOTALSIZE’ ATTRIBUTE FOR THE BULKBlockDataFile CLASS.	30
TABLE 4.9 DESCRIPTION OF THE ‘SAVEPOINT’ ATTRIBUTE FOR THE BULKBlockManifest CLASS.	30
TABLE 4.10 DESCRIPTION OF THE ‘SERVICESET’ ATTRIBUTE FOR THE BULKBlockDataFile CLASS.	31
TABLE 4.11 DESCRIPTION OF THE ‘SERVICESET’ CLASS.	31
TABLE 4.12 DESCRIPTION OF THE ‘SERVICERECORD’ ATTRIBUTE FOR THE SERVICESET CLASS.	32
TABLE 4.13 DESCRIPTION OF THE ‘SERVICERECORD’ CLASS.	32

TABLE 4.14 DESCRIPTION OF THE ‘SERVICEName’ ATTRIBUTE FOR THE SERVICERecord CLASS.	32
TABLE 4.15 DESCRIPTION OF THE ‘INTERFACEName’ ATTRIBUTE FOR THE SERVICERecord CLASS.	33
TABLE 4.16 DESCRIPTION OF THE ‘OPERATIONSet’ ATTRIBUTE FOR THE SERVICERecord CLASS.	33
TABLE 4.17 DESCRIPTION OF THE ‘OPERATIONSet’ CLASS.	33
TABLE 4.18 DESCRIPTION OF THE ‘OPERATIONName’ ATTRIBUTE FOR THE OPERATIONSet CLASS.	34
TABLE 4.19 DESCRIPTION OF THE ‘BULKBlockReport’ CLASS.	36
TABLE 4.20 DESCRIPTION OF THE ‘BULKBlockManifestIdRef’ ATTRIBUTE FOR THE BULKBlockReport CLASS.	36
TABLE 4.21 DESCRIPTION OF THE ‘TRANSACTIONReportSummary’ ATTRIBUTE FOR THE BULKBlockReport CLASS.	36
TABLE 4.22 DESCRIPTION OF THE ‘TRANSACTIONReportDetail’ ATTRIBUTE FOR THE BULKBlockReport CLASS.	37
TABLE 4.23 DESCRIPTION OF THE ‘REPORTSummary’ CLASS.	37
TABLE 4.24 DESCRIPTION OF THE ‘NOOFTotalFullSuccess’ ATTRIBUTE FOR THE REPORTSummary CLASS.	37
TABLE 4.25 DESCRIPTION OF THE ‘NOOFTotalPartialSuccess’ ATTRIBUTE FOR THE REPORTSummary CLASS.	38
TABLE 4.26 DESCRIPTION OF THE ‘NOOFTotalFailure’ ATTRIBUTE FOR THE REPORTSummary CLASS.	38
TABLE 4.27 DESCRIPTION OF THE ‘INTERFACESummaryReport’ ATTRIBUTE FOR THE REPORTSummary CLASS.	38
TABLE 4.28 DESCRIPTION OF THE ‘INTERFACESummaryReport’ CLASS.	39
TABLE 4.29 DESCRIPTION OF THE ‘INTERFACEName’ ATTRIBUTE FOR THE INTERFACESummaryReport CLASS.	39
TABLE 4.30 DESCRIPTION OF THE ‘NOOFFullSuccess’ ATTRIBUTE FOR THE INTERFACESummaryReport CLASS.	39
TABLE 4.31 DESCRIPTION OF THE ‘NOOFTotalPartialSuccess’ ATTRIBUTE FOR THE INTERFACESummaryReport CLASS.	40
TABLE 4.32 DESCRIPTION OF THE ‘NOOFTotalFailure’ ATTRIBUTE FOR THE INTERFACESummaryReport CLASS.	40
TABLE 4.33 DESCRIPTION OF THE ‘REPORTFailureDetail’ CLASS.	40
TABLE 4.34 DESCRIPTION OF THE ‘FAILUREReport’ ATTRIBUTE FOR THE REPORTFailureDetail CLASS.	41
TABLE 4.35 DESCRIPTION OF THE ‘TRANSACTIONFailureReport’ CLASS.	41
TABLE 4.36 DESCRIPTION OF THE ‘TRANSACTIONOpIdentifier’ ATTRIBUTE FOR THE TRANSACTIONReport CLASS.	41
TABLE 4.37 DESCRIPTION OF THE ‘SERVICEName’ ATTRIBUTE FOR THE TRANSACTIONReport CLASS.	42
TABLE 4.38 DESCRIPTION OF THE ‘TRANSACTIONFailStatusVocabulary’ ATTRIBUTE FOR THE TRANSACTIONReport CLASS.	42
TABLE 4.39 DESCRIPTION OF THE ‘TRANSACTIONFailStatus’ ATTRIBUTE FOR THE TRANSACTIONReport CLASS.	42
TABLE 4.40 DESCRIPTION OF THE ‘FAILUREReport’ CLASS.	43
TABLE 4.41 DESCRIPTION OF THE ‘FAILURECode’ ATTRIBUTE FOR THE FAILUREReport CLASS.	43
TABLE 4.42 DESCRIPTION OF THE ‘FAILUREDescription’ ATTRIBUTE FOR THE FAILUREReport CLASS.	44
TABLE 4.43 DESCRIPTION OF THE ‘FILTERObject’ CLASS.	45
TABLE 4.44 DESCRIPTION OF THE ‘FILTERRule’ ATTRIBUTE FOR THE FILTERObject CLASS.	45
TABLE 4.45 DESCRIPTION OF THE ‘FILTERRule’ CLASS.	45
TABLE 4.46 DESCRIPTION OF THE ‘FILTERTypeVocabulary’ ATTRIBUTE FOR THE FILTERRule CLASS.	46
TABLE 4.47 DESCRIPTION OF THE ‘FILTERType’ ATTRIBUTE FOR THE FILTERRule CLASS.	46
TABLE 4.48 DESCRIPTION OF THE ‘FILTERValueVocabulary’ ATTRIBUTE FOR THE FILTERRule CLASS.	46
TABLE 4.49 DESCRIPTION OF THE ‘FILTERValue’ ATTRIBUTE FOR THE FILTERRule CLASS.	47
TABLE 4.50 DESCRIPTION OF THE ‘SUPPORTEDServices’ CLASS.	49
TABLE 4.51 DESCRIPTION OF THE ‘PERSONOperation’ ATTRIBUTE FOR THE SUPPORTEDServices CLASS.	49
TABLE 4.52 DESCRIPTION OF THE ‘GROUPOperation’ ATTRIBUTE FOR THE SUPPORTEDServices CLASS.	49

TABLE 4.53 DESCRIPTION OF THE ‘MEMBERSHIPOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.	50
TABLE 4.54 DESCRIPTION OF THE ‘TEMPLATEOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.	50
TABLE 4.55 DESCRIPTION OF THE ‘OFFERINGOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.....	50
TABLE 4.56 DESCRIPTION OF THE ‘SECTIONOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.....	51
TABLE 4.57 DESCRIPTION OF THE ‘ASSOCIATIONOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.....	51
TABLE 4.58 DESCRIPTION OF THE ‘LINEITEMOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.....	51
TABLE 4.59 DESCRIPTION OF THE ‘RESULTOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.	52
TABLE 4.60 DESCRIPTION OF THE ‘RESULTVALUEOPERATION’ ATTRIBUTE FOR THE SUPPORTEDSERVICES CLASS.....	52
TABLE 4.61 DESCRIPTION OF THE ‘PERSONINTERFACE’ CLASS.	52
TABLE 4.62 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE PERSONINTERFACE CLASS.	53
TABLE 4.63 DESCRIPTION OF THE ‘GROUPINTERFACE’ CLASS.....	53
TABLE 4.64 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE GROUPINTERFACE CLASS.	53
TABLE 4.65 DESCRIPTION OF THE ‘MEMBERSHIPINTERFACE’ CLASS.....	54
TABLE 4.66 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE MEMBERSHIPINTERFACE CLASS.	54
TABLE 4.67 DESCRIPTION OF THE ‘TEMPLATEINTERFACE’ CLASS.....	54
TABLE 4.68 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE TEMPLATEINTERFACE CLASS.....	55
TABLE 4.69 DESCRIPTION OF THE ‘OFFERINGINTERFACE’ CLASS.....	55
TABLE 4.70 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE OFFERINGINTERFACE CLASS.....	55
TABLE 4.71 DESCRIPTION OF THE ‘SECTIONINTERFACE’ CLASS.	56
TABLE 4.72 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE SECTIONINTERFACE CLASS.	56
TABLE 4.73 DESCRIPTION OF THE ‘ASSOCIATIONINTERFACE’ CLASS.	56
TABLE 4.74 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE ASSOCIATIONINTERFACE CLASS.	57
TABLE 4.75 DESCRIPTION OF THE ‘LINEITEMINTERFACE’ CLASS.	57
TABLE 4.76 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE LINEITEMINTERFACE CLASS.	57
TABLE 4.77 DESCRIPTION OF THE ‘RESULTINTERFACE’ CLASS.	58
TABLE 4.78 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE RESULTINTERFACE CLASS.	58
TABLE 4.79 DESCRIPTION OF THE ‘RESULTVALUEINTERFACE’ CLASS.	58
TABLE 4.80 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE FOR THE RESULTVALUEINTERFACE CLASS.	59
TABLE 5.1 CLASS DESCRIPTORS	60
TABLE 5.2 DESCRIPTION OF THE ‘BULKDATARECORD’ CLASS.....	64
TABLE 5.3 DESCRIPTION OF THE ‘TRANSACTIONRECORD’ ATTRIBUTE OF THE BULKDATARECORD CLASS.....	64
TABLE 5.4 DESCRIPTION OF THE ‘TRANSACTIONRECORD’ CLASS.	65
TABLE 5.5 DESCRIPTION OF THE ‘TRANSACTIONIDENTIFIER’ ATTRIBUTE OF THE TRANSACTIONRECORD CLASS.	65
TABLE 5.6 DESCRIPTION OF THE ‘SERVICENAME’ ATTRIBUTE OF THE TRANSACTIONRECORD CLASS.	66
TABLE 5.7 DESCRIPTION OF THE ‘INTERFACENAME’ ATTRIBUTE OF THE TRANSACTIONRECORD CLASS.	66
TABLE 5.8 DESCRIPTION OF THE ‘OPERATIONNAME’ ATTRIBUTE OF THE TRANSACTIONRECORD CLASS.	67
TABLE 5.9 DESCRIPTION OF THE ‘PARAMETERSET’ ATTRIBUTE OF THE TRANSACTIONRECORD CLASS.	67
TABLE 5.10 DESCRIPTION OF THE ‘PARAMETERSET’ CLASS.....	67
TABLE 5.11 DESCRIPTION OF THE ‘PARAMETERRECORD’ ATTRIBUTE OF THE PARAMETERSET CLASS.	68
TABLE 5.12 DESCRIPTION OF THE ‘PARAMETERRECORD’ CLASS.....	68
TABLE 5.13 DESCRIPTION OF THE ‘PARAMETERINVOC’ ATTRIBUTE OF THE PARAMETERRECORD CLASS.	68
TABLE 5.14 DESCRIPTION OF THE ‘PARAMETERNAME’ ATTRIBUTE OF THE PARAMETERRECORD CLASS.	69

TABLE 5.15 DESCRIPTION OF THE ‘PARAMETERType’ ATTRIBUTE OF THE PARAMETERRECORD CLASS.....	69
TABLE 5.16 DESCRIPTION OF THE ‘PARAMETERVALUE’ ATTRIBUTE OF THE PARAMETERRECORD CLASS.....	70
TABLE 5.17 DESCRIPTION OF THE ‘PARAMETERVALUE’ CLASS.....	70
TABLE 5.18 DESCRIPTION OF THE ‘GUID’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	71
TABLE 5.19 DESCRIPTION OF THE ‘GUIDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	71
TABLE 5.20 DESCRIPTION OF THE ‘QUERYOBJECT’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	71
TABLE 5.21 DESCRIPTION OF THE ‘SEQUENCEIDENTIFIER’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	72
TABLE 5.22 DESCRIPTION OF THE ‘TEXT’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	72
TABLE 5.23 DESCRIPTION OF THE ‘URL’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	73
TABLE 5.24 DESCRIPTION OF THE ‘PERSONRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	73
TABLE 5.25 DESCRIPTION OF THE ‘PERSONCORERECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	73
TABLE 5.26 DESCRIPTION OF THE ‘PERSONRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	74
TABLE 5.27 DESCRIPTION OF THE ‘GROUPRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	74
TABLE 5.28 DESCRIPTION OF THE ‘GROUPRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	74
TABLE 5.29 DESCRIPTION OF THE ‘MEMBERSHIPRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	75
TABLE 5.30 DESCRIPTION OF THE ‘MEMBERSHIPRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	75
TABLE 5.31 DESCRIPTION OF THE ‘COURSETEMPLATERECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	75
TABLE 5.32 DESCRIPTION OF THE ‘COURSETEMPLATERECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	76
TABLE 5.33 DESCRIPTION OF THE ‘COURSEOFFERINGRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	76
TABLE 5.34 DESCRIPTION OF THE ‘COURSESECTIONRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	76
TABLE 5.35 DESCRIPTION OF THE ‘COURSESECTIONRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	77
TABLE 5.36 DESCRIPTION OF THE ‘SECTIONASSOCIATIONRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS..	77
TABLE 5.37 DESCRIPTION OF THE ‘SECTIONASSOCIATIONRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	77
TABLE 5.38 DESCRIPTION OF THE ‘RESULTRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	78
TABLE 5.39 DESCRIPTION OF THE ‘RESULTRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	78
TABLE 5.40 DESCRIPTION OF THE ‘RESULTVALUERECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	78
TABLE 5.41 DESCRIPTION OF THE ‘RESULTVALUERECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	79
TABLE 5.42 DESCRIPTION OF THE ‘LINEITEMRECORD’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	79
TABLE 5.43 DESCRIPTION OF THE ‘LINEITEMRECORDSET’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	79
TABLE 5.44 DESCRIPTION OF THE ‘LINEITEMTYPE’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	80
TABLE 5.45 DESCRIPTION OF THE ‘MEMBERSHIPIDTYPE’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	80
TABLE 5.46 DESCRIPTION OF THE ‘RELATIONSHIP’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.....	80
TABLE 5.47 DESCRIPTION OF THE ‘ROLE’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	81
TABLE 5.48 DESCRIPTION OF THE ‘RESULTSTATUS’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	81
TABLE 5.49 DESCRIPTION OF THE ‘STATUS’ ATTRIBUTE FOR THE PARAMETERVALUE CLASS.	81
TABLE A1.1 STATUS CODES FOR THE BULKDATAEXCHANGEMANAGER INTERFACE OPERATIONS.	84
TABLE A1.2 COMMON STATUS CODES FOR THE SERVICE OPERATIONS.	85
TABLE B1.1 THE VOCABULARY FOR ‘TRANSACTIONFAILSTATUS’.....	86
TABLE B2.1 THE VOCABULARY FOR ‘PARAMETERType’.	87
TABLE B3.1 THE VOCABULARY FOR ‘FILTERType’.	89
TABLE B4.1 THE VOCABULARY FOR ‘FAILURECODE’.....	90

1 Introduction

1.1 Bulk Data Exchange Management Service Overview

The Bulk Data Exchange Management Service (BDEMS) specification is the definition of how systems manage the exchange of structured bulk information between learning information systems. The Bulk Data Exchange 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 – the Bulk Data Exchange Management Service focuses on the exchange of structured bulk data between learning information systems. There are no definitions in the specification on how the data is managed within the systems;
- Service-oriented – the Bulk Data Exchange 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 Bulk Data Exchange Management Service is combined with the Person Management Service, Group Management Service, Membership Management Service, Course Management Service and Outcomes Management Service to provide the Learning Information Services [LIS, 11a];
- Layering – the Bulk Data Exchange 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 Bulk Data Exchange Management Service is defined in terms of its 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 Bulk Data Exchange Management Service information model is defined using the Unified Modeling Language (UML). This enables reliable mapping of the information model into a range of different bindings. The binding of immediate importance is to the Web Services Description Language (WSDL);
- Adoption – whenever appropriate, the Bulk Data Exchange Management Service specification makes use of other IMS GLC and non-IMS GLC standards and specifications.

The aim of this specification is to enable systems to exchange structured bulk data. The defined service can be included in any type of system e.g., Student Information System, Human Resources System, etc. A ‘bulk data’ object can be composed of many different types of structured objects e.g., ‘person’, ‘group’, ‘course offering’, etc. Six operations are supplied that allow a service to announce the presence of the bulk data file and to report the retrieval of the bulk data file. A separate Web Service file exchange protocol is defined to obtain the bulk data file.

1.2 Scope and Context

This document is the IMS GLC Bulk Data Exchange Management Service Information Model v1.0 and as such it is used as the basis for the development of the following document:

- a) IMS GLC Bulk Data Exchange Management Service WSDL Binding v1.0 [BDEMS, 11] – the description of the WSDL binding of the Information Model.

The core uses-cases for the Bulk Data Exchange Management Service are described as a subset of the Learning Information Services Specification [LIS, 11a]. This is the first version of this service.

This information model defines Bulk Data Exchange Management Service Abstract Application Programming Interface (a-API). The Learning Information Services specification, of which the Bulk Data Exchange 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 Modeling Language (UML) [SDN07, 07].

1.3 Structure of this Document

The structure of this document is:

2. BULK DATA EXCHANGE MANAGEMENT SERVICE DESCRIPTION	The description of the overall structure and operation of the Bulk Data Exchange Management Service. This includes the description of the architectural model and the domain object model;
3. BEHAVIORAL MODEL	The definition of the operations of Bulk Data Exchange Management 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 Bulk Data Exchange Management Service End Systems. These are the parameters exchanged across the interoperability interface;
5. BULK DATA FILE DATA MODEL	The definition of the data models for the external bulk data file that must be obtained using the file transfer protocol;
6. FILE RETRIEVAL APPROACHES	A brief discussion on the approach to be used to obtain the external data file(s) identified by the URLs supplied to the service consumer;
7. EXTENDING AND PROFILING THE SERVICE	Identification of the ways in which the Bulk Data Exchange Management Service can be extended both in terms of the addition of new constituent services and proprietary extensions to a service;
APPENDIX A SERVICE STATUS CODES	A summary list of the status codes, and their causes, that can be returned by each of the operations forming the Bulk Data Exchange Management Service;
APPENDIX B SET OF VOCABULARIES	The set of vocabularies that have been defined for this service.

1.4 Nomenclature

API	Application Programming Interface
a-API	Abstract Application Programming Interface
BDEMS	Bulk Data Exchange Management Service
FTP	File Transfer Protocol
FTPES	FTP over Explicit TLS/SSL
FTPS	File Transfer Protocol over SSL
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAF	IMS GLC Abstract Framework
IMS GLC	IMS Global Learning Consortium Inc.
LIS	Learning Information Services
PIM	Platform Independent Model
PSM	Platform Specific Model
RFC	Request For Comment
SDN	Specification Development Note
SSH	Secure Shell
SSL	Secure Sockets Layer

SFTP	SSH File Transfer Protocol
TLS	Transport Layer Security
UML	Unified Modeling Language
URL	Uniform Resource Locator
WSDL	Web Services Description Language

1.5 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 WSDL Binding v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 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 GLC 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 Specification v1.0 Final Release*, L.Feng, W.Lee and C.Smythe, IMS Global Learning Consortium, June 2011.
- [LIS, 11b] *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, June 2011.
- [RFC2228] *FTP Security Extensions*, M.Horowitz and S.Lunt, IETF RFC 2228, October 2009.
- [RFC4217] *Securing FTP with TLS*, P.Ford-Hutchinson, IETF RFC 4217, October 2005.
- [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.
- [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 Bulk Data Exchange 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 a Bulk Data Exchange Management System. The breakdown of the service into its interface classes is a convenient way to document the set of behaviors. The objective for producing these interfaces is to identify and define the messages that are exchanged between the end-systems to realize the system behaviors required of the service.

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 Bulk Data Exchange Services 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 those 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 Bulk Data Exchange Management Service Architecture & Specification Model

The basic architectural model for the Bulk Data Exchange Management Service specification is shown in Figure 2.1. In this architecture the scope of the IMS GLC Bulk Data Exchange 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.

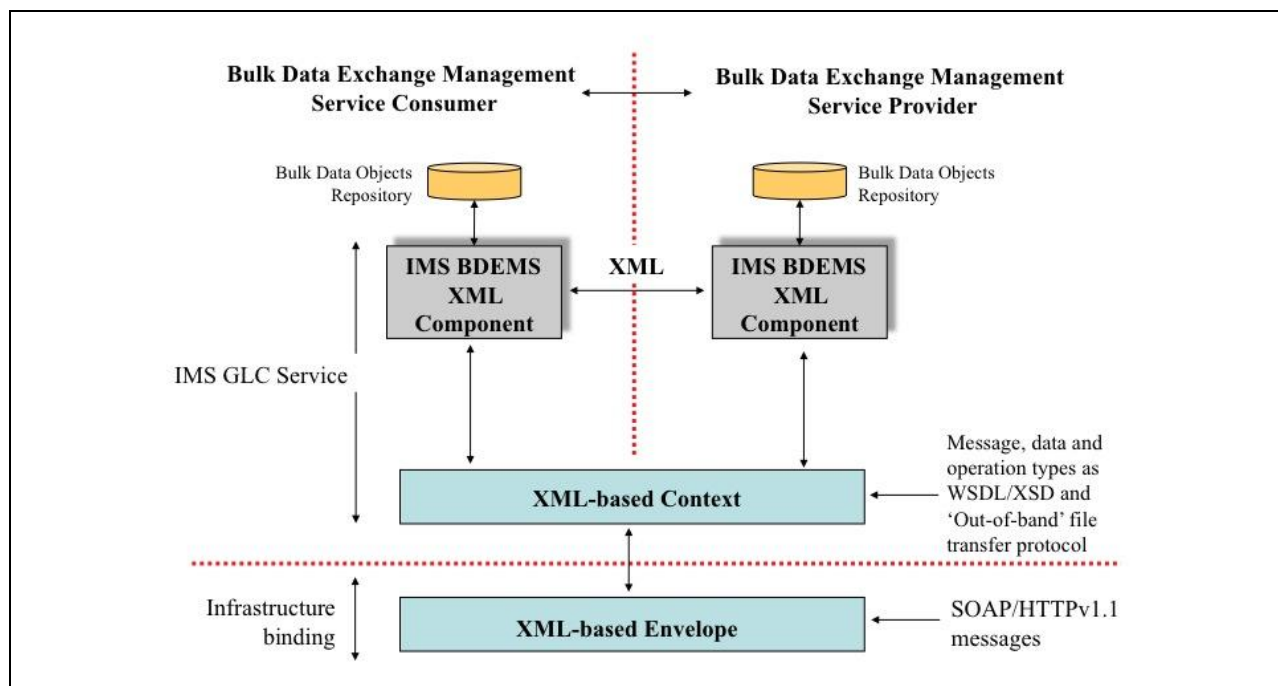


Figure 2.1 Bulk Data Exchange Management service architecture model.

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 bulk data 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 persistent consistent with the required behavior.

2.3 Bulk Data File

The core of the BDEMS is that two systems exchange bulk data using external data files (the number of physical files exchanged is an implementation issue). The logical structure of the bulk data file is shown in Figure 2.2b and the equivalent physical distribution to data files is shown in Figure 2.2a. The bulk data file is acquired using the Web Service (alternative file transfer protocols are also possible but not addressed in this specification).

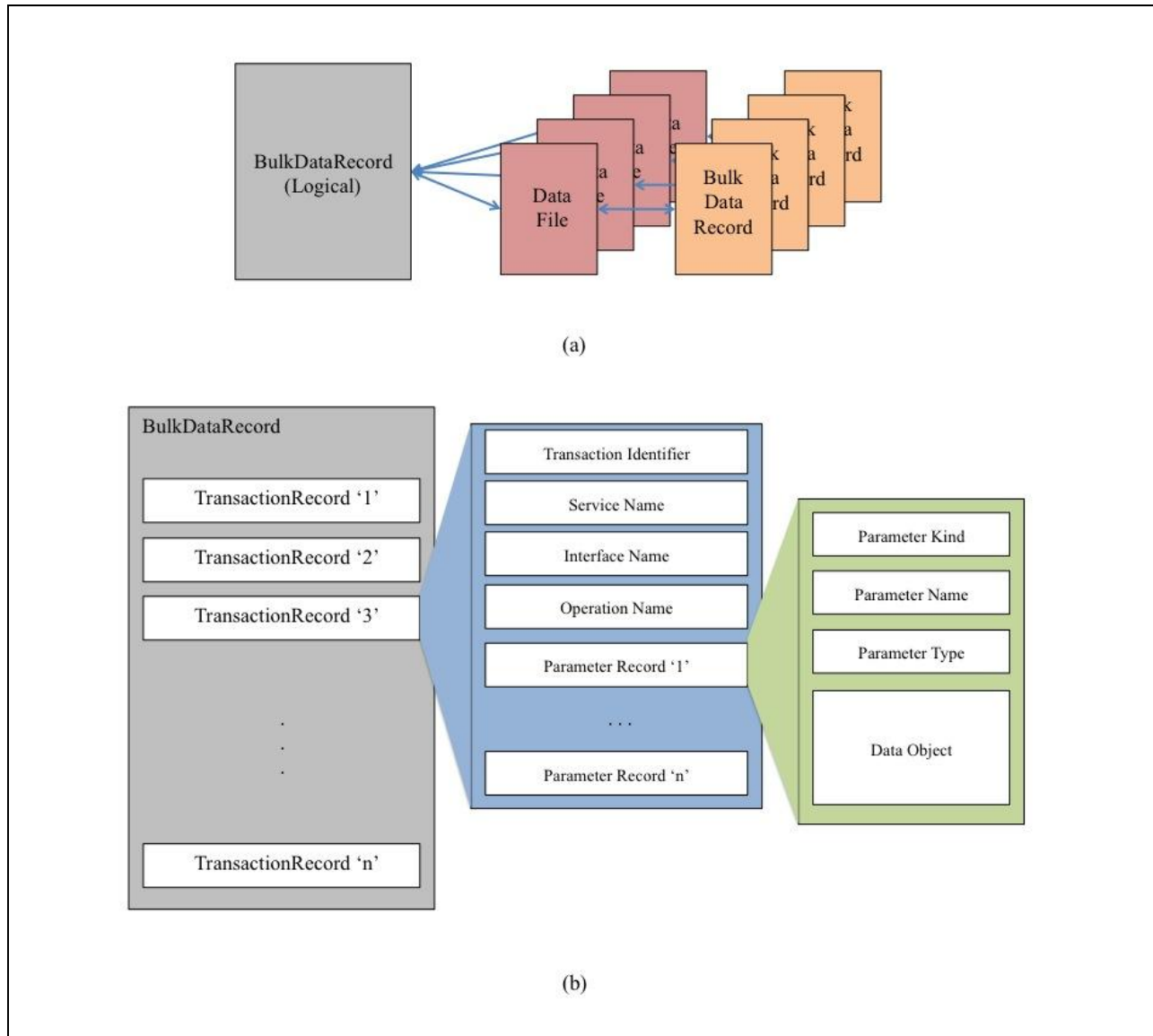


Figure 2.2 Composition of a bulk data file.

In Figure 2.2b, the top-level container for the bulk data file is the bulk data record that is composed of an unbounded number of Transaction Records. Each record is used to contain the operation for a particular service and interface combination. The order of the transaction records is significant however there is no explicit relationship between the transaction records.

2.4 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:

- 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 BulkDataExchangeManagementService is used to model the service responsible for manipulating information used between Learning Information Services systems. The BulkDataExchangeManagementService interface is shown in Figure 3.1.

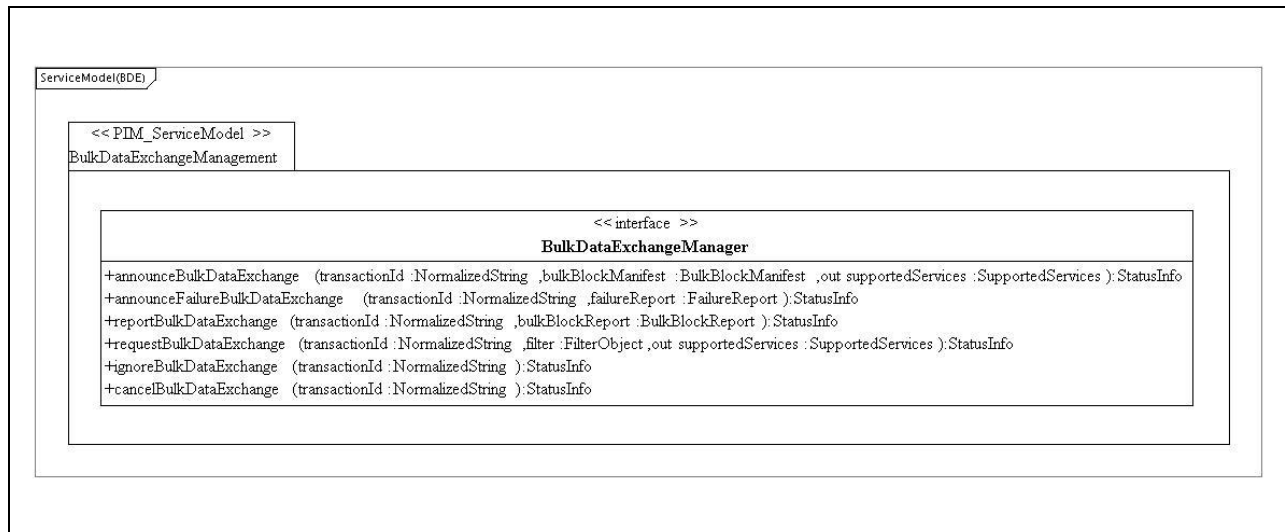


Figure 3.1 BulkDataExchangeManagementService interface definition.

The BulkDataExchangeManagementService has a single interface: BulkDataExchangeManager that supports the manipulation of a single bulk object.

NOTE: The operation used to download the data file(s) is not defined in Figure 3.1. Instead, the specific Web Service binding is defined in Section 6.

3.2 BulkDataExchangeManager Interface Description

The BulkDataExchangeManager interface class describes the operations that are permitted on 'bulk data' objects. The interface stereotype indicates that there are no attributes for this class. The set of operations are summarized in Table 3.1.

Table 3.1 Summary of operations for BulkDataExchangeManager.

Operation	Description
announceBulkDataExchange	To announce the availability of a bulk data object that is ready to be retrieved by service consumers. This call suspends real-time event notification
announceFailureBulkDataExchange	To inform the service consumer that a previously issued, and acknowledged, requestBulkDataExchange transaction cannot be serviced by the service provider.
reportBulkDataExchange	To report the retrieval of the bulk data object whose availability was announced previously. This signals real-time event processing should be restarted.
requestBulkDataExchange	Issued by the service consumer to request a bulk data exchange from the service provider.
ignoreBulkDataExchange	Issued by the service consumer to inform the service provider that a previously issued announce bulk data exchange request will be ignored. Real-time processing should be resumed by both consumer and server.
cancelBulkDataExchange	Issued by the service consumer to inform the service provider that a previously requested bulk data exchange request has now been cancelled. Real-time processing should be resumed by both consumer and server.

3.2.1 AnnounceBulkDataExchange() Operation

Name:	announceBulkDataExchange
Return Function Parameter:	<i>StatusInfo</i> – the status of the announce request. The permitted status codes are defined in Tables 3.2 and A.2.
Supplied (in) Parameters:	<p><i>transactionId:NormalizedString</i> – the unique identifier assigned to this transaction by the service provider.</p> <p><i>bulkBlockManifest:BulkBlockManifest</i> – ‘house-keeping’ information about the bulk data files that are available including the URLs.</p>
Returned (out) Parameters:	<i>servicesSupported:ServicesSupported</i> – a report of the services supported by the consumer responding to this announcement.
Behavior:	<p>When the service provider issues the ‘announceBulkDataExchange’ request the target must acknowledge the receipt of the announcement (this must be acknowledged to ensure that the service provider does not enter a retransmission mode, etc.). This is the first phase of the asynchronous service and all responses must return the transactionId supplied in this operation. The service consumer will then use an appropriate file retrieval protocol to obtain the file identified by the URL(s) given in the request message. The ‘house-keeping’ information is used to ascertain whether or not the consumer can make use of the information in the file.</p> <p>Once this call has been issued, the service provider will stop real-time event notifications. This will be restarted once the bulk exchange has either been completed or prematurely terminated.</p> <p>A service consumer receiving this announcement will provide, in the response, a description of the services and operations supported.</p>
Notes:	None.

Table 3.2 Status codes for the ‘announceBulkDataExchange’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The announcement request has been successfully received and the supplied data is valid.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invalidurl’	One of the received URLs is invalid.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=incompletedata’	Some mandatory part of the BulkBlockManifest object has been detected as missing by the service consumer.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	Some part of the BulkBlockManifest object is invalid.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=unsupportedservices’	One, or more, of the services named in the bulk block data file object is not supported by the service consumer.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=unsupportedoperations'	One, or more, of the services named in the bulk block data file object is not supported by the service consumer.

3.2.2 **AnnounceFailureBulkDataExchange() Operation**

Name:	announceFailureBulkDataExchange
Return Function Parameter:	<i>StatusInfo</i> – the status of the announce failure request. The permitted status codes are defined in Tables 3.3 and A.2.
Supplied (in) Parameters:	<p><i>transactionId:NormalizedString</i> – the unique identifier assigned to this transaction by the service provider.</p> <p><i>failureReport:FailureReport</i> – the failure report supplied by the service provider.</p>
Returned (out) Parameters:	None.
Behavior:	<p>When the service provider issues the ‘announceFailureBulkDataExchange’ request the service consumer must acknowledge its receipt (this must be acknowledged to ensure that the service provider does not enter a retransmission mode, etc.). The transactionId must be the same as that supplied by the service consumer allocated in the original ‘requestBulkDataExchange’ operation.</p> <p>In the failure report the following failure codes are permitted:</p> <ul style="list-style-type: none"> • ‘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. <p>Once this call has been issued and acknowledged, the service provider will resume real-time event notifications.</p>
Notes:	Other failure codes will be defined as required.

Table 3.3 Status codes for the ‘announceFailureBulkDataExchange’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The announce failure operation has been successfully received and processed at the service consumer.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invalidtransactionid’	The transactionId returned by the service provider is unknown at the service consumer. An error condition is logged at the service consumer but no further processing is undertaken.

3.2.3 ReportBulkDataExchange() Operation

Name:	reportBulkDataExchange
Return Function Parameter:	<i>StatusInfo</i> – the status of the report request. The permitted status codes are defined in Tables 3.4 and A.2.
Supplied (in) Parameters:	<p><i>transactionId:NormalizedString</i> – the unique identifier assigned to this transaction by the service provider's previous announcement of available bulk data.</p> <p><i>bulkBlockReport:BulkBlockReport</i> – the status report information on the retrieval of the bulk block data file.</p>
Returned (out) Parameters:	None.
Behavior:	<p>When the service consumer issues the 'reportBulkDataExchange' request the service provider must acknowledge its receipt. This report operation is in response to a previously sent announcement i.e., this is the second phase of the asynchronous service. This report provides details on the processing of the transactions in the bulk data record. Error information concerning any transaction processing that resulted in a failure condition is reported.</p> <p>Receipt of a valid report results in the real-time event notification to be restarted by the service provider.</p>
Notes:	None.

Table 3.4 Status codes for the 'reportBulkDataExchange' operation.

Status Code	Explanation of the Cause of the Code
'CodeMajor=Success' 'Severity=Status' 'CodeMinor=fullsuccess'	The report has been successfully received at the service provider and the supplied data is valid.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invalidtransactionid'	The transactionId returned by the service consumer is unknown at the service provider. An error condition is logged at the service provider but no further processing is undertaken.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=incompletedata'	Some mandatory part of the BulkBlockReport object has been detected as missing by the service provider.
'CodeMajor=Failure' 'Severity=Status' 'CodeMinor=invaliddata'	Some part of the BulkBlockReport object is invalid.
CodeMajor=Failure' 'Severity=Status' 'CodeMinor=expireddata'	The bulk data file 'expiryDate' shows that the validity of the file has expired.

3.2.4 RequestBulkDataExchange() Operation

Name:	requestBulkDataExchange
Return Function Parameter:	<i>StatusInfo</i> – the status of the bulk data request. The permitted status codes are defined in Tables 3.5 and A.2.
Supplied (in) Parameters:	<p><i>transactionId:NormalizedString</i> – the unique identifier assigned to this transaction by the service consumer and to be used in associated transaction responses.</p> <p><i>filter:FilterObject</i> – the filter (supplied as a type/value pair) that is to be applied by the service provider to create the bulk data exchange file required by the service consumer. The current filter instructions are:</p> <ul style="list-style-type: none"> ‘type=object’ – this is used to identify the data types are required e.g., Person objects, Group objects, etc. (the permitted range of values is defined in the associated vocabulary); ‘type=savepoint’ – the sequence identifier reference point from which the required data is to be supplied. <p>The filter instructions are composed using a logical ‘AND’.</p>
Returned (out) Parameters:	<i>servicesSupported:ServicesSupported</i> – a report of the services supported by the service provider responding to this announcement.
Behavior:	<p>Upon receipt and acceptance of this request, the service provider is expected to suspend real-time event notifications.</p> <p>A service provider receiving this request will provide, in the response, a description of the services and operations supported.</p>
Notes:	None.

Table 3.5 Status codes for the ‘requestBulkDataExchange’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The request has been successfully received and the target will be able to fulfill the request.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invaliddata’	The identified object data type(s) is/are unknown in the service provider.
‘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 synchronization data will be made available.

3.2.5 IgnoreBulkDataExchange() Operation

Name:	ignoreBulkDataExchange
Return Function Parameter:	<i>StatusInfo</i> – the status of the ignore transaction. The permitted status codes are defined in Tables 3.6 and A.2.
Supplied (in) Parameters:	<i>transactionId:NormalizedString</i> – the unique identifier assigned to this transaction by the service provider’s previous announcement of available bulk data.
Returned (out) Parameters:	None.
Behavior:	<p>The ‘ignoreBulkDataExchange’ operation is issued when the service consumer wishes to inform the service provider that a previously issued ‘announceBulkDataExchange’ transaction will be ignored.</p> <p>Furthermore real-time event processing is to be resumed by both the service provider and service consumer.</p>
Notes:	None.

Table 3.6 Status codes for the ‘ignoreBulkDataExchange’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The ignore request has been successfully received at the service provider.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invalidtransactionid’	The transactionId returned by the service consumer is unknown at the service provider. An error condition is logged at the service provider but no further processing is undertaken.

3.2.6 CancelBulkDataExchange() Operation

Name:	cancelBulkDataExchange
Return Function Parameter:	<i>StatusInfo</i> – the status of the cancel transaction. The permitted status codes are defined in Tables 3.7 and A.2.
Supplied (in) Parameters:	<i>transactionId:NormalizedString</i> – the unique identifier assigned to this transaction by the service consumer’s previous bulk data exchange request.
Returned (out) Parameters:	None.
Behavior:	<p>The ‘cancelBulkDataExchange’ operation is issued when the service consumer wishes to inform the service provider that a previously issued bulk data exchange request transaction is to be cancelled. The transactionId must be the same as that supplied in the previous ‘requestBulkDataExchange’.</p> <p>Furthermore real-time event processing is to be resumed by both the service provider and service consumer.</p>
Notes:	None.

Table 3.7 Status codes for the ‘cancelBulkDataExchange’ operation.

Status Code	Explanation of the Cause of the Code
‘CodeMajor=Success’ ‘Severity=Status’ ‘CodeMinor=fullsuccess’	The cancel request has been successfully received at the service provider.
‘CodeMajor=Failure’ ‘Severity=Status’ ‘CodeMinor=invalidtransactionid’	The transactionId returned by the service consumer is unknown at the service provider. An error condition is logged at the service provider but no further processing is undertaken.

3.3 Service Choreography

The Bulk Data Exchange Management Services is an asynchronous service. There are two basic patterns:

- Service provider driven announcements;
- Service consumer driven requests.

Figure 3.2 is a schematic representation of the service choreography for the service provider driven process; the announce/report service calls choreography.

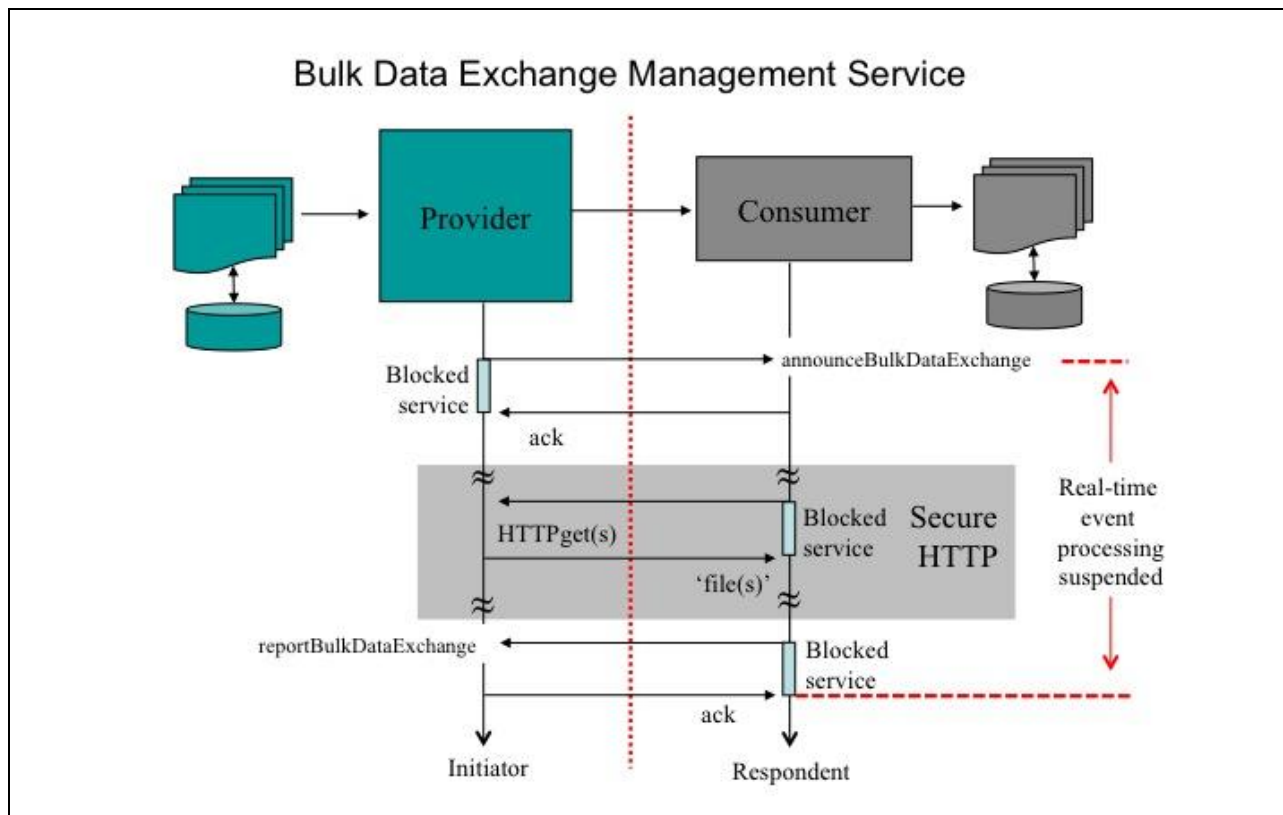


Figure 3.2 Service Provider driven asynchronous service choreography.

The two phases of the BDEMS are separated by the data file download protocol (using a secure HTTP message file retrieval i.e., HTTPSget). The first phase is established when the 'announceBulkDataExchange' request message is issued by the service provider. Once received by the service consumer the announcement acknowledgement is sent back to the service provider. At some later time, determined by the service consumer, the identified set of data files are retrieved using one or more 'HTTPSget' calls (the data files may, or may not, be stored on the service provider or some other data store); see Section 6. The contents of the data files are inspected. Once the data files have been processed, the service consumer issues the 'reportBulkDataExchange' message to the service provider. In response the service provider acknowledges receipt of the report message. Real-time event notifications and processing between these two systems is suspended during this choreography.

Each of the 'announceBulkDataExchange' and 'reportBulkDataExchange' request/acknowledgement message sequences is synchronous i.e., the service is blocked until the corresponding acknowledgement is received. It is the choreography between the two sets of message sequences, and the download protocol, that is asynchronous.

The service consumer driven choreography is shown schematically in Figure 3.3. The service consumer issues the 'requestBulkDataExchange' call. At some point later, the service provider issues an 'announceBulkDataExchange' request to inform the service consumer of the location of the requested data files. The choreography now follows the sequence as shown in Figure 3.2.

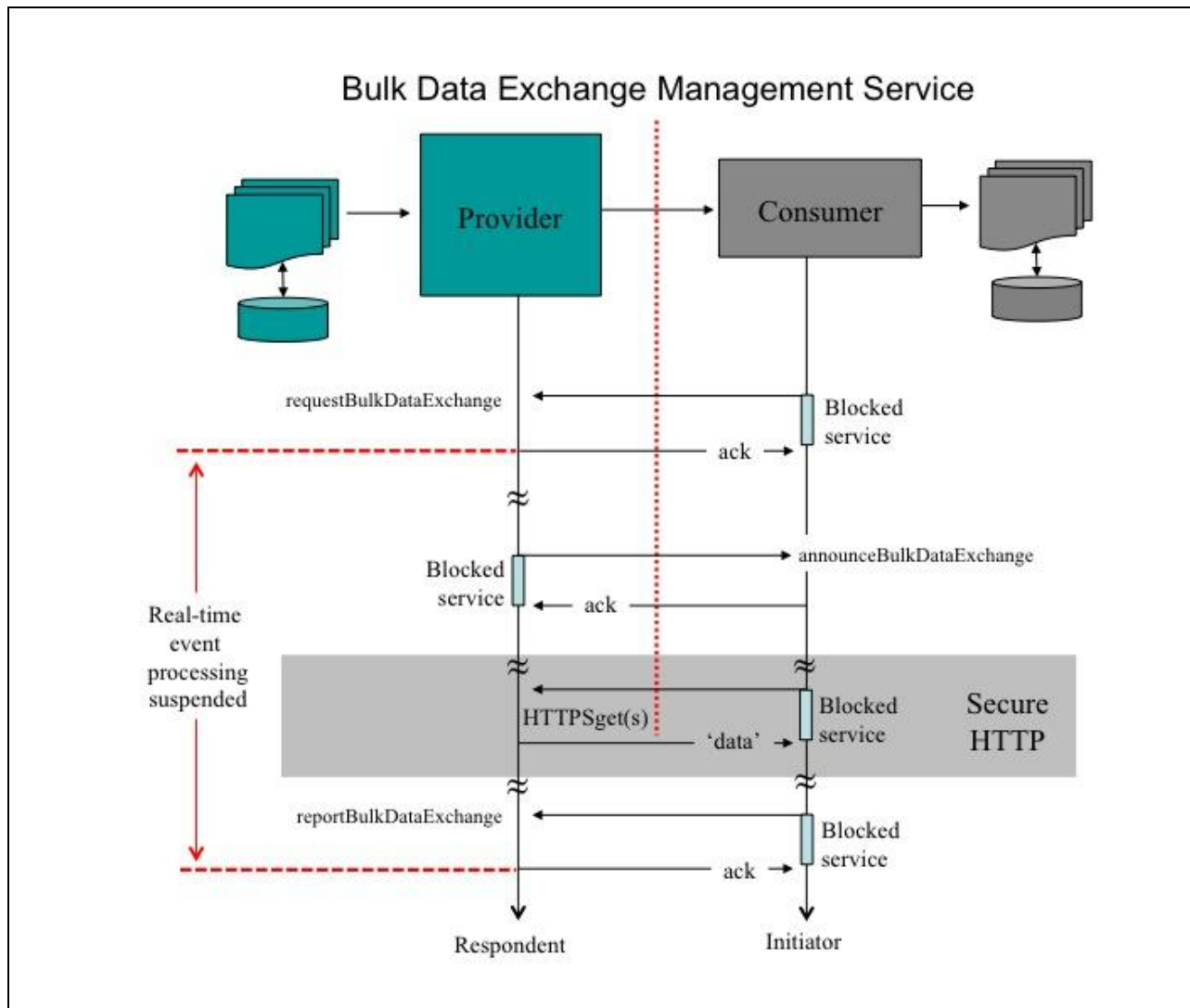


Figure 3.3 Service Provider driven asynchronous service choreography.

In Figures 3.2 and 3.3 the acknowledgement messages are empty messages but are tagged so that the original request message is identified. An appropriate status code is returned i.e., the acknowledgement may indicate that the request will not be completed.

The choreographies displayed in Figure 3.2 and 3.3 address the success sequence of events. Alternative choreographies are possible when failure conditions are detected or the service consumer wishes to prematurely end the process i.e., and return to real-time event processing. These alternate choreographies are supported by the following operations:

- In Figure 3.2 the service consumer can restart the real-time event processing without downloading the bulk data file(s) by issuing the 'ignoreBulkDataExchange'. Real-time event processing restarts once the service provider acknowledges receipt of the 'ignoreBulkDataExchange';
- In Figure 3.3 the service provider may issue the 'announceFailureBulkDataExchange' instead of the 'announceBulkDataExchange'. This occurs when the service provider is unable to service the original request. For example, the service provider may be unable to apply the required data retrieval filters;
- In Figure 3.3 the service consumer may prematurely cancel the bulk data exchange by issuing the 'cancelBulkDataExchange' operation. At this point the service provider will cancel the generation and storage

of the corresponding bulk data files. Real-time event processing is immediately restarted by both the service consumer and provider.

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. All parameters are mandatory.

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

4.1 BulkBlockManifest Class Description

The PIM for the BulkBlockManifest data model is shown in Figure 4.1.

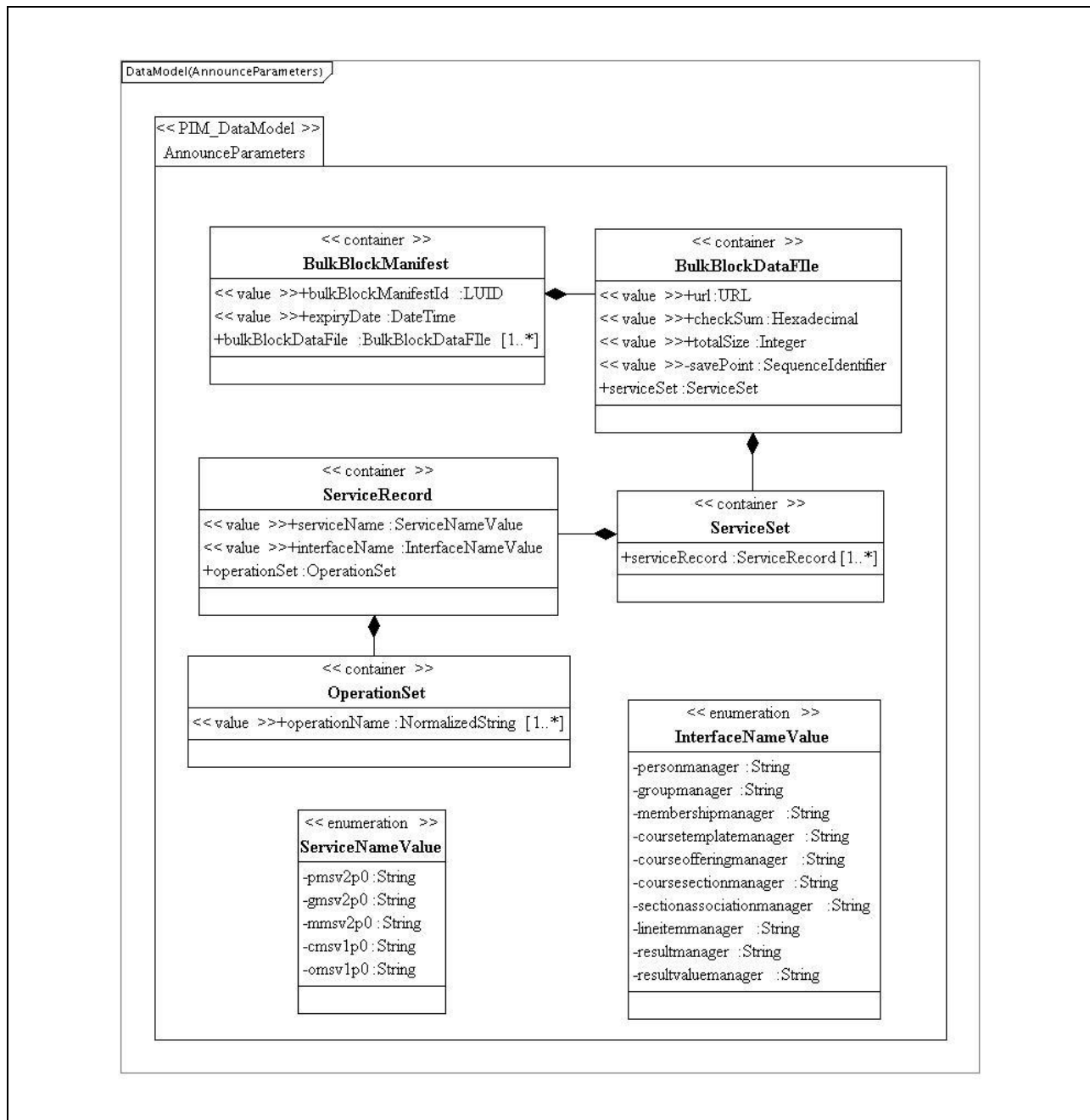


Figure 4.1 BulkBlockManifest class diagram.

Table 4.1 Description of the ‘BulkBlockManifest’ class.

Descriptor	Definition
Class name	BulkBlockManifest
Class type	container
Multiplicity	1
Parents	Parameter Root
Children	[bulkBlockManifestId, expiryDate, bulkBlockDataFile]
Description	This is the container for all of the information that describes the data that is to be found in the external bulk data files. This is a manifest for the set of external files i.e., the bulk data can be stored in more than one file. A unique transaction identifier is also included.

4.1.1 BulkBlockManifestId Attribute Description

Table 4.2 Description of the ‘bulkBlockManifestId’ attribute for the BulkBlockManifest class.

Descriptor	Definition
Attribute name	bulkBlockManifestId
Data type	LUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is a unique identifier that is used to mark this bulk block manifest transaction. All transaction responses to this request must refer to this identifier.

4.1.2 ExpiryDate Attribute Description

Table 4.3 Description of the ‘expiryDate’ attribute for the BulkBlockManifest class.

Descriptor	Definition
Attribute name	expiryDate
Data type	DateTime
Value space	See Table 5.1.
Multiplicity	1
Description	The date and time after which the data file set will be declared as expired.

4.1.3 BulkBlockDataFile Attribute Description

Table 4.4 Description of the ‘bulkBlockDataFile’ attribute for the BulkBlockManifest class.

Descriptor	Definition
Attribute name	bulkBlockDataFile
Data type	BulkBlockDataFile
Value space	container
Multiplicity	1..unbounded, ordered
Description	The information specific to a block data file.

4.1.4 BulkBlockDataFile Class Description

Table 4.5 Description of the ‘BulkBlockDataFile’ class.

Descriptor	Definition
Class name	BulkBlockDataFile
Class type	container
Multiplicity	1..unbounded, ordered
Parents	BulkBlockManifest
Children	[url, checkSum, totalSize, savePoint, serviceSet]
Description	The container for all the data specific to a block data file. File order is significant.

4.1.5 Url Attribute Description

Table 4.6 Description of the ‘url’ attribute for the BulkBlockDataFile class.

Descriptor	Definition
Attribute name	url
Data type	URL
Value space	See Table 5.1.
Multiplicity	1
Description	The URL of the data block file. This URL must be used by the appropriate file download protocol (defined in Section 6).

4.1.6 CheckSum Attribute Description

Table 4.7 Description of the ‘checksum’ attribute for the BulkBlockDataFile class.

Descriptor	Definition
Attribute name	checksum
Data type	Hexadecimal
Value space	A 128bit hexadecimal number in the range [‘x00000000000000000000000000000000’ - ‘xffffffffffffffffffffffffffffffff’)
Multiplicity	1
Description	The checksum of the file. The algorithm will be MD5 i.e., a 128-bit checksum is created.

4.1.7 TotalSize Attribute Description

Table 4.8 Description of the ‘totalSize’ attribute for the BulkBlockDataFile class.

Descriptor	Definition
Attribute name	totalSize
Data type	Integer
Value space	An integer in the range $[1-2^{32-1}]$.
Multiplicity	1
Description	This is the total length of the file in bytes.

4.1.8 SavePoint Attribute Description

Table 4.9 Description of the ‘savePoint’ attribute for the BulkBlockManifest class.

Descriptor	Definition
Attribute name	savePoint
Data type	SequenceIdentifier
Value space	See Sub-section 4.5 of this document.
Multiplicity	1
Description	The date and time after which the data file set will be declared as expired.

4.1.9 ServiceSet Attribute Description

Table 4.10 Description of the ‘serviceSet’ attribute for the BulkBlockDataFile class.

Descriptor	Definition
Attribute name	serviceSet
Data type	ServiceSet
Value space	container
Multiplicity	1
Description	The container for the set of information for all of the service operations. There is a separate sub-container for the information about each operation. This information is used to provide the service provider with a summary of the services and operations that are to be invoked.

4.1.10 ServiceSet Class Description

Table 4.11 Description of the ‘ServiceSet’ class.

Descriptor	Definition
Class name	ServiceSet
Class type	container
Multiplicity	1
Parents	BulkBlockDataFile
Children	[serviceRecord]
Description	<p>The container for the set of information for all of the service operations. There is a separate sub-container for the information about each service and interface combination. The order of this information is unimportant because it is not used to organize how the data file is interpreted. There may be several descriptions for a service.</p> <p>This information is used by a service consumer to determine whether or not it can handle all of the operations described in the bulk data file.</p>

Table 4.12 Description of the ‘serviceRecord’ attribute for the ServiceSet class.

Descriptor	Definition
Attribute name	serviceRecord
Data type	ServiceRecord
Value space	container
Multiplicity	1..unbounded, unordered
Description	The container for information about a single service and interface combination.

4.1.11 ServiceRecord Class Description**Table 4.13 Description of the ‘ServiceRecord’ class.**

Descriptor	Definition
Class name	ServiceRecord
Class type	container
Multiplicity	1..unbounded, ordered
Parents	ServiceSet
Children	[serviceName, interfaceName, operationSet]
Description	The container for information about a single service and interface combination.

Table 4.14 Description of the ‘serviceName’ attribute for the ServiceRecord class.

Descriptor	Definition
Attribute name	serviceName
Data type	Enumerated vocabulary.
Value space	The permitted vocabulary is: { pmsv2p0 gmsv2p0 mmsv2p0 cmsv1p0 omsv1p0 }.
Multiplicity	1
Description	The name of the service that owns the operation. This name is that allocated to the name of the ServiceModel Package in the service PIM.

Table 4.15 Description of the ‘interfaceName’ attribute for the ServiceRecord class.

Descriptor	Definition
Attribute name	interfaceName
Data type	Enumerated vocabulary.
Value space	The permitted vocabulary is: { personmanager groupmanager membershipmanager coursetemplatemanager courseofferingmanager coursesectionmanager sectionassociationmanager lineitemmanager resultmanager resultvaluemanager }.
Multiplicity	1
Description	The name of the interface that owns the operation. This name is that allocated to the name of the interface class in the ServiceModel Package in the service PIM.

Table 4.16 Description of the ‘operationSet’ attribute for the ServiceRecord class.

Descriptor	Definition
Attribute name	operationSet
Data type	OperationSet
Value space	container
Multiplicity	1
Description	The container for the set of named operations that must be supported for the successful completion of the data file read.

4.1.12 OperationSet Class Description

Table 4.17 Description of the ‘OperationSet’ class.

Descriptor	Definition
Class name	OperationSet
Class type	container
Multiplicity	1
Parents	ServiceRecord
Children	[operationName]
Description	The container for the set of named operations that must be supported for the successful completion of the data file read.

Table 4.18 Description of the ‘operationName’ attribute for the OperationSet class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	NormalizedString [1-255 characters].
Multiplicity	1..unbounded, unordered
Description	The names of the set of operations that are supported by the associated service/interface combination.

4.2 BulkBlockReport Class Description

The PIM for the BulkBlockReport data model is shown in Figure 4.2.

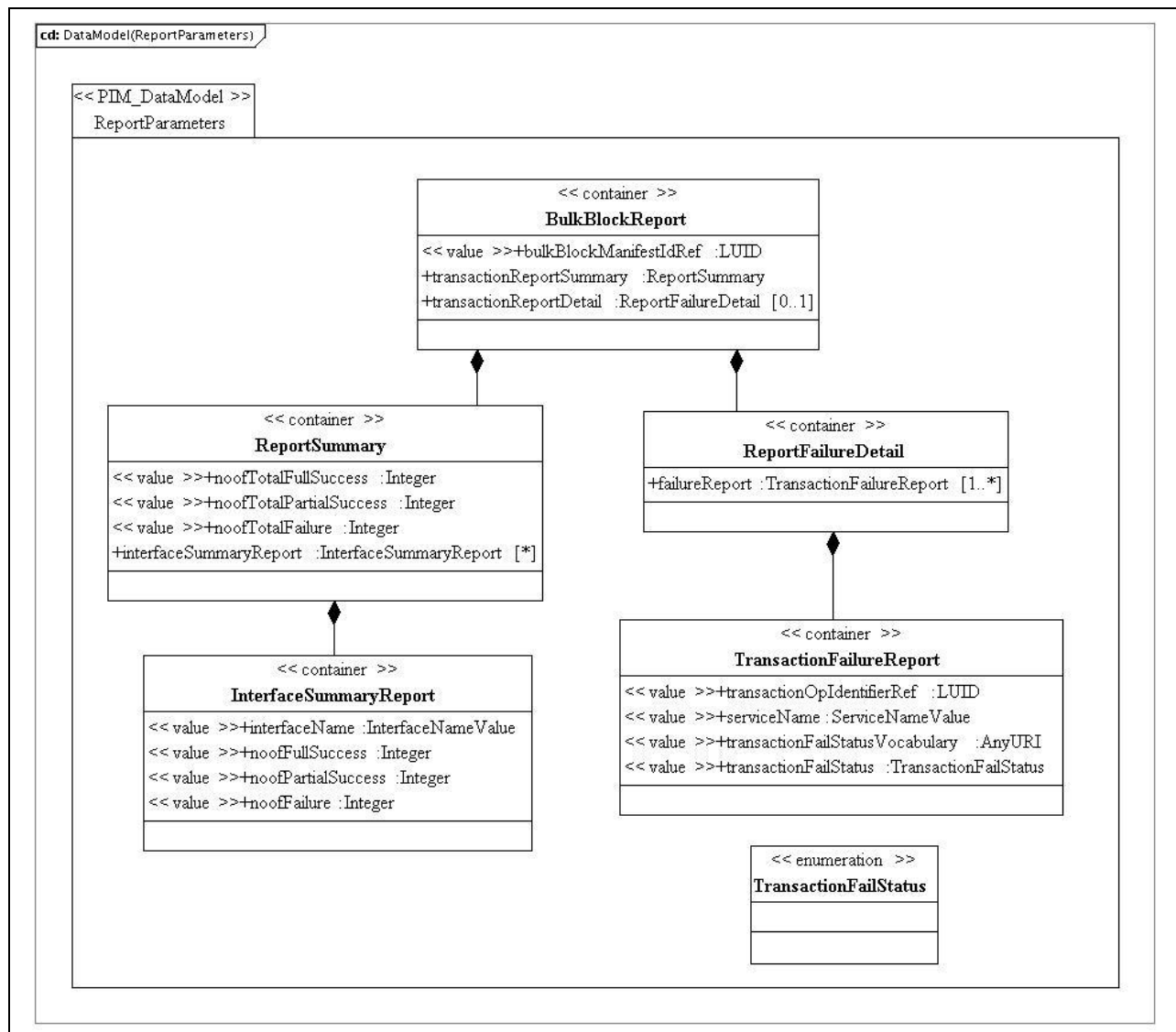


Figure 4.2 BulkBlockReport class diagram.

Table 4.19 Description of the ‘BulkBlockReport’ class.

Descriptor	Definition
Class name	BulkBlockReport
Class type	container
Multiplicity	1
Parents	Parameter Root
Children	[bulkBlockManifestIdRef, transactionReportSummary, transactionReportDetail]
Description	This is the container for the report information that is returned in the ‘reportBulkDataExchange’ operation. This report must be returned to the service provider that issued the original ‘announceBulkDataExchange’ operation.

Table 4.20 Description of the ‘bulkBlockManifestIdRef’ attribute for the BulkBlockReport class.

Descriptor	Definition
Attribute name	bulkBlockManifestIdRef
Data type	LUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is the unique identifier that was allocated to the BulkBlockManifest in the ‘announceBulkDataExchange’ request operation to which this report is a response. This enables the two block manifest operations to be correlated.

Table 4.21 Description of the ‘transactionReportSummary’ attribute for the BulkBlockReport class.

Descriptor	Definition
Attribute name	transactionReportSummary
Data type	ReportSummary
Value space	container
Multiplicity	1
Description	The container for the summary status report on all the transactions within the bulk data file actions undertaken by the service consumer

Table 4.22 Description of the ‘transactionReportDetail’ attribute for the BulkBlockReport class.

Descriptor	Definition
Attribute name	transactionReportDetail
Data type	FailureReportDetail
Value space	container
Multiplicity	0..1
Description	The container for the status reports on all the transactions within the bulk data file that were unsuccessfully completed by the service consumer. If no reports are contained then the ‘announceBulkDataExchange’ request has been successfully completed.

4.2.1 ReportSummary Class Description

Table 4.23 Description of the ‘ReportSummary’ class.

Descriptor	Definition
Class name	ReportSummary
Class type	container
Multiplicity	1
Parents	BulkBlockReport
Children	[noofTotalFullSuccess, noofTotalPartialSuccess, noofTotalFailure, interfaceSummaryReport]
Description	The container for the summary information about the processing of the bulk data files by the service consumer.

Table 4.24 Description of the ‘noofTotalFullSuccess’ attribute for the ReportSummary class.

Descriptor	Definition
Attribute name	noofTotalFullSuccess
Data type	Integer
Value space	See Table 5.1.
Multiplicity	1
Description	Reports the total number of successfully completed transactions (with a code minor value of ‘fullsuccess’ or ‘createsuccess’) as requested in the bulk data files i.e., for all of the services.

Table 4.25 Description of the ‘noofTotalPartialSuccess’ attribute for the ReportSummary class.

Descriptor	Definition
Attribute name	noofTotalPartialSuccess
Data type	Integer
Value space	See Table 5.1.
Multiplicity	1
Description	Reports the total number of partially completed transactions as requested in the bulk data files i.e., for all of the services.

Table 4.26 Description of the ‘noofTotalFailure’ attribute for the ReportSummary class.

Descriptor	Definition
Attribute name	noofTotalFailure
Data type	Integer
Value space	See Table 5.1.
Multiplicity	1
Description	Reports the total number of failed transactions requested in the bulk data files i.e., for all of the services.

Table 4.27 Description of the ‘interfaceSummaryReport’ attribute for the ReportSummary class.

Descriptor	Definition
Attribute name	interfaceSummaryReport
Data type	InterfaceSummaryReport
Value space	container
Multiplicity	0..unbounded, unordered
Description	Provides a summary report for each of the specific interfaces.

4.2.2 InterfaceSummaryReort Class Description

Table 4.28 Description of the ‘InterfaceSummaryReport’ class.

Descriptor	Definition
Class name	InterfaceSummaryReport
Class type	container
Multiplicity	0..unbounded, unordered
Parents	BulkBlockReport
Children	[interfaceName, noofFullSuccess, noofPartialSuccess, noofFailure]
Description	The container for the summary information about the processing of the bulk data files by the service consumer for a specific service interface.

Table 4.29 Description of the ‘interfaceName’ attribute for the InterfaceSummaryReport class.

Descriptor	Definition
Attribute name	interfaceName
Data type	Enumerated vocabulary.
Value space	The permitted vocabulary is: { personmanager groupmanager membershipmanager coursetemplatemanager courseofferingmanager coursesectionmanager sectionassociationmanager lineitemmanager resultmanager resultvaluemanager }.
Multiplicity	1
Description	The name of the interface for which he summary is being reported. This name is that allocated to the name of the interface class in the ServiceModel Package in the service PIM.

Table 4.30 Description of the ‘noofFullSuccess’ attribute for the InterfaceSummaryReport class.

Descriptor	Definition
Attribute name	noofFullSuccess
Data type	Integer
Value space	See Table 5.1.
Multiplicity	1
Description	Reports the total number of successfully completed transactions (with a code minor value of ‘fullsuccess’ or ‘createsuccess’) for the named interface as requested in the bulk data files.

Table 4.31 Description of the ‘noofTotalPartialSuccess’ attribute for the InterfaceSummaryReport class.

Descriptor	Definition
Attribute name	noofPartialSuccess
Data type	Integer
Value space	See Table 5.1.
Multiplicity	1
Description	Reports the total number of partially completed transactions as requested for the named interface as requested in the bulk data files.

Table 4.32 Description of the ‘noofTotalFailure’ attribute for the InterfaceSummaryReport class.

Descriptor	Definition
Attribute name	noofFailure
Data type	Integer
Value space	See Table 5.1.
Multiplicity	1
Description	Reports the total number of failed transactions for the named interface as requested in the bulk data files.

4.2.3 ReportFailureDetail Class Description

Table 4.33 Description of the ‘ReportFailureDetail’ class.

Descriptor	Definition
Class name	ReportFailureDetail
Class type	container
Multiplicity	0..1
Parents	BulkBlockReport
Children	[failueReport]
Description	The container for the set of reports that describe the transactions that resulted in the generation of failure status codes.

Table 4.34 Description of the ‘failureReport’ attribute for the ReportFailureDetail class.

Descriptor	Definition
Attribute name	failureReport
Data type	TransactionFailureReport
Value space	container
Multiplicity	1
Description	The container for the failure reports for a specific transaction.

4.2.4 TransactionReport Class Description

Table 4.35 Description of the ‘TransactionFailureReport’ class.

Descriptor	Definition
Class name	TransactionFailureReport
Class type	container
Multiplicity	0..unbounded, ordered
Parents	ReportFailureDetail
Children	[transactionOpIdentifierRef, serviceName, transactionFailStatusVocabulary, transactionFailStatus]
Description	The container for the status reports on all the transactions within the bulk data file that were unsuccessfully completed by the service consumer. If no reports are contained then the ‘announceBulkDataExchange’ request has been successfully completed.

Table 4.36 Description of the ‘transactionOpIdentifier’ attribute for the TransactionReport class.

Descriptor	Definition
Attribute name	transactionOpIdentifierRef
Data type	LUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is the unique identifier allocated to this transaction in the Bulk Data File. Only those transactions that have resulted in a failure condition are reported using this object.

Table 4.37 Description of the ‘serviceName’ attribute for the TransactionReport class.

Descriptor	Definition
Attribute name	serviceName
Data type	Enumerated vocabulary.
Value space	The permitted vocabulary is: { pmsv2p0 gmsv2p0 mmsv2p0 cmsv1p0 omsv1p0 }.
Multiplicity	1
Description	The name of the service that owns the operation. This name is that allocated to the name of the ServiceModel Package in the service PIM.

Table 4.38 Description of the ‘transactionFailStatusVocabulary’ attribute for the TransactionReport class.

Descriptor	Definition
Attribute name	transactionFailStatusVocabulary
Data type	AnyURI.
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that contains the permitted values for the attribute ‘transactionFailStatus’.

Table 4.39 Description of the ‘transactionFailStatus’ attribute for the TransactionReport class.

Descriptor	Definition
Attribute name	transactionFailStatus
Data type	Enumerated list.
Value space	Vocabulary-based – see Appendix B.
Multiplicity	1
Description	<p>The data that is used to report the failure condition of the transaction. The significance of the status value is dependent on the service and operation that has resulted in the failure. 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.</p> <p>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.</p>

4.3 FailureReport Class Description

This is the data-type for the reporting of the failure conditions associated with the ‘announceBulkDataExchange’ and reported in the ‘announceFailureBulkDataExchange’ transaction.

Table 4.40 Description of the ‘FailureReport’ class.

Descriptor	Definition
Class name	FailureReport
Class type	container
Multiplicity	1
Parents	Parameter Root
Children	[failureCode, failureDescription]
Description	The container for the failure reports by the service provider in response to a ‘requestBulkDataExchange’ transaction.

Table 4.41 Description of the ‘failureCode’ attribute for the FailureReport class.

Descriptor	Definition
Attribute name	failureCode
Data type	Enumerated list.
Value space	Vocabulary-based – see Appendix B.
Multiplicity	1
Description	<p>The data that is used to report the failure code for a ‘requestBulkDataExchange’ transaction. 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.</p> <p>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.</p>

Table 4.42 Description of the ‘failureDescription’ attribute for the FailureReport class.

Descriptor	Definition
Attribute name	failureDescription
Data type	NormalizedString
Value space	A Normalized String [1-255 characters].
Multiplicity	1
Description	A short, human readable, description of the cause of the failure of the ‘requestBulkDataExchnage’.

4.4 FilterObject Class Description

This is the data-type for the set of filter rules that are to be applied to the data objects controlled by the service provider. The PIM for the FilterObject data model is shown in Figure 4.3.

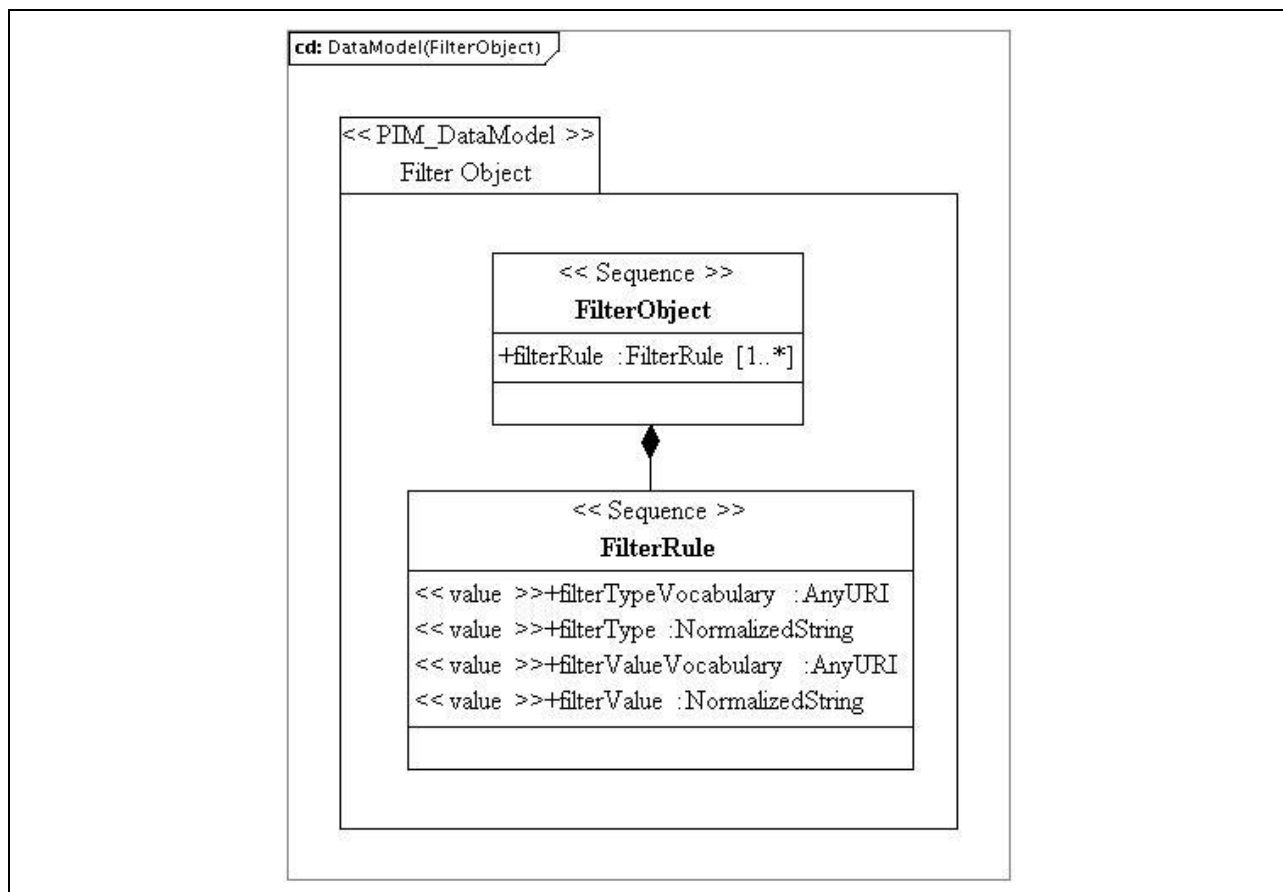
**Figure 4.3 FilterObject class diagram.**

Table 4.43 Description of the ‘FilterObject’ class.

Descriptor	Definition
Class name	FilterObject
Class type	container
Multiplicity	1
Parents	Parameter Root
Children	[filterRule]
Description	The container for the set of filter rules that are to be applied to the set of data objects. The set of rules are composed assuming a logical ‘AND’ relationship.

Table 4.44 Description of the ‘filterRule’ attribute for the FilterObject class.

Descriptor	Definition
Attribute name	filterRule
Data type	FilterRule
Value space	container
Multiplicity	1..unbounded, unordered
Description	The set of filter rules.

Table 4.45 Description of the ‘FilterRule’ class.

Descriptor	Definition
Class name	FilterRule
Class type	container
Multiplicity	1
Parents	FilterObject
Children	[filterTypeVocabulary , filterType, filterValueVocabulary, filterValue]
Description	The container for the type/value for each filter rule.

Table 4.46 Description of the ‘filterTypeVocabulary’ attribute for the FilterRule class.

Descriptor	Definition
Attribute name	filterTypeVocabulary
Data type	AnyURI.
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that contains the permitted values for the attribute ‘filterType’.

Table 4.47 Description of the ‘filterType’ attribute for the FilterRule class.

Descriptor	Definition
Attribute name	filterType
Data type	Enumerated list.
Value space	Enumerated as: <ul style="list-style-type: none"> • ‘object’ – provide data according to the identified type of data object; • ‘savepoint’ – provide data that has been changed at, or after, the supplied SequenceIdentifier reference point.
Multiplicity	1
Description	The data that is used to define the permitted set of filter rules.

Table 4.48 Description of the ‘filterValueVocabulary’ attribute for the FilterRule class.

Descriptor	Definition
Attribute name	filterValueVocabulary
Data type	AnyURI.
Value space	See Table 5.1.
Multiplicity	1
Description	The URI for the vocabulary that contains the permitted values for the attribute ‘filterValue’.

Table 4.49 Description of the ‘filterValue’ attribute for the FilterRule class.

Descriptor	Definition
Attribute name	filterValue
Data type	NormalizedString
Value space	<p>If ‘filterType=object’ then the value space is the enumerated set defined in corresponding vocabulary in Appendix B</p> <p>If ‘filterType=savepoint’ then the value space is defined by the data-type SequenceIdentifier.</p>
Multiplicity	1
Description	The rule that is to be applied to the data retrieval.

4.5 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 ‘1000-01-01T00:00:00.000’. 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 ‘group’ object.

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

4.6 SupportedServices Class Description

This is the data-type for the report of the set of LIS services that are supported by a specific end system. The PIM for the SupportedServices data model is shown in Figure 4.4.

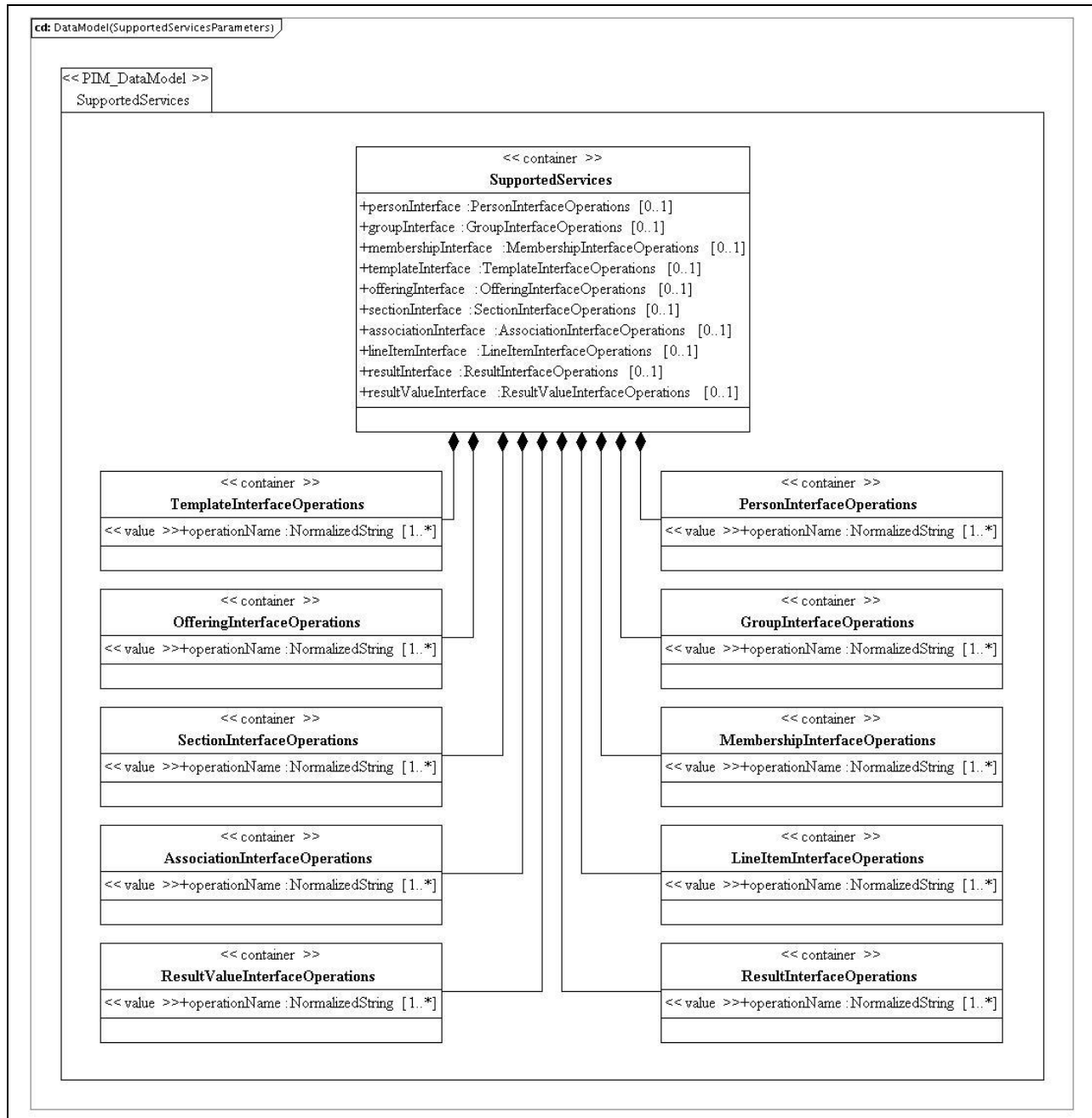


Figure 4.4 SupportedServices class diagram.

Table 4.50 Description of the ‘SupportedServices’ class.

Descriptor	Definition
Class name	SupportedServices
Class type	container
Multiplicity	1
Parents	Parameter Root
Children	[personInterface, groupInterface, membershipInterface, templateInterface, offeringInterface, sectionInterface, associationInterface, lineItemInterface, resultInterface, resultValueInterface]
Description	The container for the set of reports on the operations for each LIS interface that are supported by the end system providing the report.

Table 4.51 Description of the ‘personOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	personOperation
Data type	PersonInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the PersonManager interface.

Table 4.52 Description of the ‘groupOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	groupOperation
Data type	GroupInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the GroupManager interface.

Table 4.53 Description of the ‘membershipOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	membershipOperation
Data type	MembershipInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the MembershipManager interface.

Table 4.54 Description of the ‘templateOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	templateOperation
Data type	TemplateInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the CourseTemplateManager interface.

Table 4.55 Description of the ‘offeringOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	offeringOperation
Data type	OfferingInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the CourseOfferingManager interface.

Table 4.56 Description of the ‘sectionOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	sectionOperation
Data type	SectionInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the CourseSectionManager interface.

Table 4.57 Description of the ‘associationOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	associationOperation
Data type	AssociationInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the SectionAssociationManager interface.

Table 4.58 Description of the ‘lineItemOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	lineItemOperation
Data type	LineItemInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the LineItemManager interface.

Table 4.59 Description of the ‘resultOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	resultOperation
Data type	ResultInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the ResultManager interface.

Table 4.60 Description of the ‘resultValueOperation’ attribute for the SupportedServices class.

Descriptor	Definition
Attribute name	resultValueOperation
Data type	ResultValueInterfaceOperations
Value space	container
Multiplicity	0..1
Description	The container for the report of the operations supported in the ResultValueManager interface.

4.6.1 PersonInterface Class Description

Table 4.61 Description of the ‘PersonInterface’ class.

Descriptor	Definition
Class name	PersonInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the PersonManager interface that are supported by the end system.

Table 4.62 Description of the ‘operationName’ attribute for the PersonInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the PersonManager interface.

4.6.2 GroupInterface Class Description

Table 4.63 Description of the ‘GroupInterface’ class.

Descriptor	Definition
Class name	GroupInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the GroupManager interface that are supported by the end system.

Table 4.64 Description of the ‘operationName’ attribute for the GroupInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the GroupManager interface.

4.6.3 MembershipInterface Class Description

Table 4.65 Description of the ‘MembershipInterface’ class.

Descriptor	Definition
Class name	GroupInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the MembershipManager interface that are supported by the end system.

Table 4.66 Description of the ‘operationName’ attribute for the MembershipInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the MembershipManager interface.

4.6.4 TemplateInterface Class Description

Table 4.67 Description of the ‘TemplateInterface’ class.

Descriptor	Definition
Class name	TemplateInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the CourseTemplateManager interface that are supported by the end system.

Table 4.68 Description of the ‘operationName’ attribute for the TemplateInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the CourseTemplateManager interface.

4.6.5 OfferingInterface Class Description

Table 4.69 Description of the ‘OfferingInterface’ class.

Descriptor	Definition
Class name	OfferingInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the CourseOfferingManager interface that are supported by the end system.

Table 4.70 Description of the ‘operationName’ attribute for the OfferingInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the CourseOfferingManager interface.

4.6.6 SectionInterface Class Description

Table 4.71 Description of the 'SectionInterface' class.

Descriptor	Definition
Class name	SectionInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the CourseSectionManager interface that are supported by the end system.

Table 4.72 Description of the 'operationName' attribute for the SectionInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the CourseSectionManager interface.

4.6.7 AssociationInterface Class Description

Table 4.73 Description of the 'AssociationInterface' class.

Descriptor	Definition
Class name	AssociationInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the SectionAssociationManager interface that are supported by the end system.

Table 4.74 Description of the ‘operationName’ attribute for the AssociationInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the CourseOfferingManager interface.

4.6.8 LineItemInterface Class Description**Table 4.75 Description of the ‘LineItemInterface’ class.**

Descriptor	Definition
Class name	LineItemInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the LineItemManager interface that are supported by the end system.

Table 4.76 Description of the ‘operationName’ attribute for the LineItemInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the LineItemManager interface.

4.6.9 ResultInterface Class Description

Table 4.77 Description of the ‘ResultInterface’ class.

Descriptor	Definition
Class name	ResultInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the ResultManager interface that are supported by the end system.

Table 4.78 Description of the ‘operationName’ attribute for the ResultInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the ResultManager interface.

4.6.10 ResultValueInterface Class Description

Table 4.79 Description of the ‘ResultValueInterface’ class.

Descriptor	Definition
Class name	ResultInterface
Class type	container
Multiplicity	1
Parents	SupportedServices
Children	[operationName]
Description	The report of the set of operations for the ResultValueManager interface that are supported by the end system.

Table 4.80 Description of the ‘operationName’ attribute for the ResultValueInterface class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	See Table 5.1.
Multiplicity	1
Description	The name of an operation supported by the ResultValueManager interface.

4.7 StatusInfo Class Description

This is the container 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].

5 Bulk Data File Data Model

The bulk data model defines the logical structure of the data to be stored in the external data file(s) that is exchanged between BDEMS systems. 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 three abstract class types. These abstractions are bound to specific data structures for machine processing in the IMS GLC Bulk Data Exchange Management Services v1.0 WSDL Binding Version 1.0 [BDEMS, 11]. The three 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	<p>For value and characteristic classes, the allowed structure for valid values for the class.</p> <p>Valid data types are:</p> <p>Boolean: The primitive, two-valued data type that uses the keywords “true” and “false” to indicate the logical state of an object.</p> <p>DateTime: The DateTime represents a combined date and time in the format of ISO 8601 i.e., ‘YYYY-MM-DDThh:mm:ssTZD’. The time is denoted in Coordinated Universal Time (UTC) with TZD denoting the time zone offset in hours and minutes with respect to UTC.</p> <p>GUID: An identifier that is globally unique. This will be based upon the NormalizedString data-type that has a constrained value-space.</p> <p>Hexadecimal: A hexadecimal number.</p> <p>Integer: An integer.</p> <p>LUID: An identifier that is locally unique i.e., within the context of the two communicating systems. This will be based upon the NormalizedString data-type that has a constrained value-space.</p> <p>NormalizedString: A sequence of printable characters that does not contain carriage returns or tabs.</p>

Descriptor	Definition
	<p>String: A sequence of printable characters.</p> <p>Unspecified: The data type is not known or is not important.</p> <p>URL: A string that conforms to the constraints of a URL. A normalized string for the of [1-4095 characters].</p>
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	<p>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:</p> <ul style="list-style-type: none"> • '0..1' [optional; restricted] • '0..unbounded' [optional; unrestricted] • '1..1' [mandatory; restricted] • '1..unbounded' [mandatory; unrestricted] <p>Multiplicities may also appear in short-hand notation in the UML models. The short-hand equivalents shall be (exclusive of bracketed comments):</p> <ul style="list-style-type: none"> • '*' [optional; unrestricted] • '1' [mandatory; restricted] • '1..*' [mandatory; unrestricted] <p>Where multiplicity is greater than one, the importance of the ordering of siblings is also indicated by appending either ","ordered or "," unordered.</p> <p>Ordered specifies a sequence of siblings as listed, unordered specifies a collection or bag of siblings in which order is not important.</p>
Parents	Lists classes that may be parents of this class.
Children	<p>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 shall be separated by a comma.</p> <p>Where more than one child is listed, the importance of the ordering of siblings is also indicated by appending either ","ordered or "," unordered.</p> <p>Ordered specifies a sequence of siblings as listed, unordered specifies a collection or bag of siblings in which order is not important.</p>
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 support only 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 must 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 BulkDataRecord Class Description

The PIM for the BulkDataRecord data model is shown in Figure 5.1.

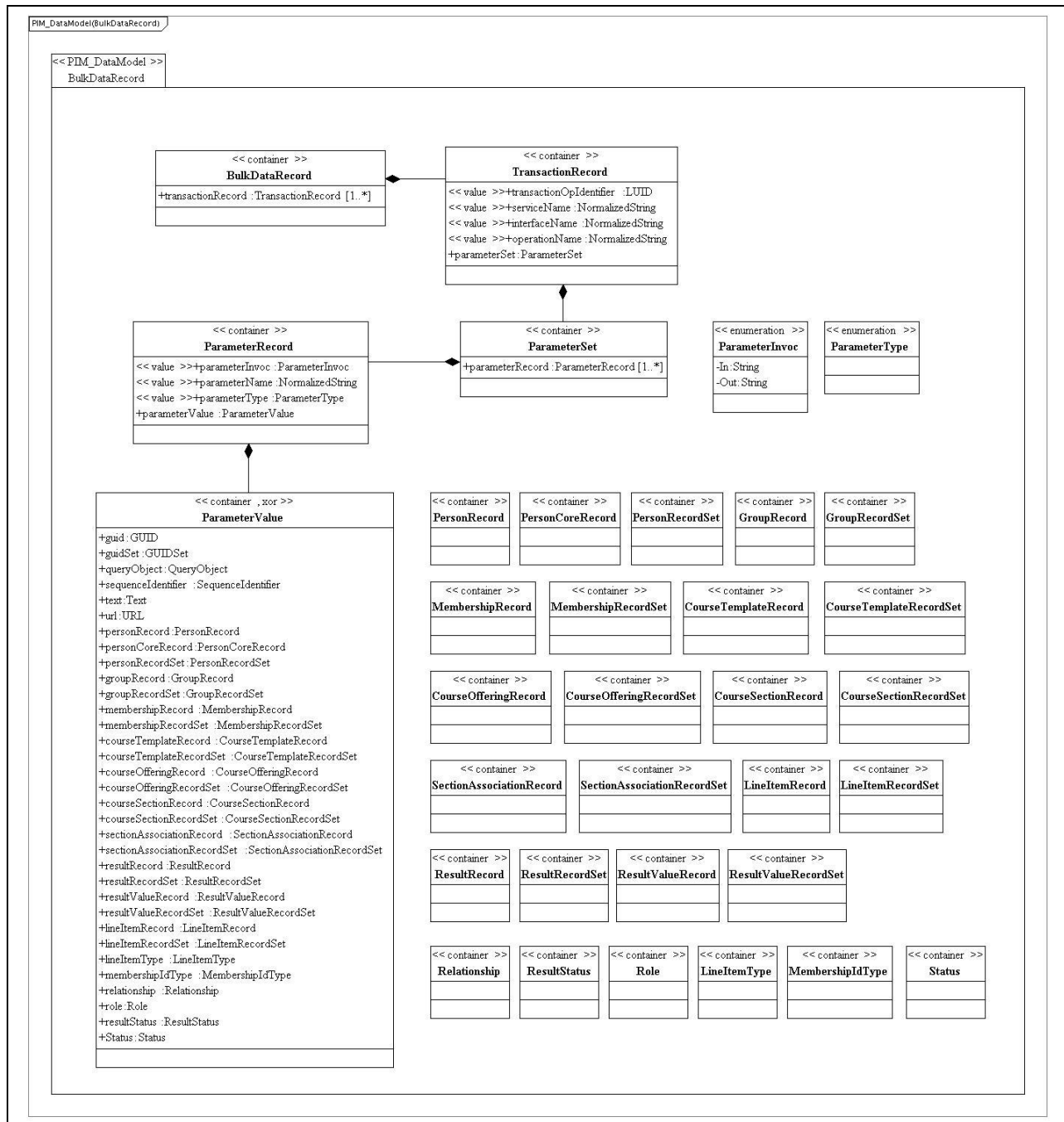


Figure 5.1 BulkDataRecord class diagram.

Table 5.2 Description of the ‘BulkDataRecord’ class.

Descriptor	Definition
Class name	BulkDataRecord
Class type	container
Multiplicity	1
Parents	Root
Children	[transactionRecord], ordered
Description	This is the top-level container for the set of transaction records that are contained within the bulk data record.

5.2.1 TransactionRecord Attribute Description

Table 5.3 Description of the ‘transactionRecord’ attribute of the BulkDataRecord class.

Descriptor	Definition
Attribute name	transactionRecord
Data type	TransactionRecord
Value space	container
Multiplicity	1..unbounded, ordered
Description	Each transaction has its own data structure. The order of the transactionRecords is important and must be maintained. A bulk data file must be able to store at least 100,000 transactionRecords.

5.3 TransactionRecord Class Description

Table 5.4 Description of the ‘TransactionRecord’ class.

Descriptor	Definition
Class name	TransactionRecord
Class type	container
Multiplicity	1..unbounded, ordered
Parents	BulkDataRecord
Children	[transactionOpIdentifier, serviceName, interfaceName, operationName, parameterSet], ordered
Description	This is the container for each of the transactions to be processed by a BDEMS. Each ‘transactionRecord’ is self-contained but the consequences of its processing will depend on the position it occurs within the sequence of transactionRecords.

5.3.1 TransactionIdentifier Attribute Description

Table 5.5 Description of the ‘transactionIdentifier’ attribute of the TransactionRecord class.

Descriptor	Definition
Attribute name	transactionOpIdentifier
Data type	LUID
Value space	See Table 5.1.
Multiplicity	1
Description	This is the unique identifier assigned to this transactionRecord within the set of bulk data records (each BDEMS call could involve several URLs, each pointing to a different file each of which contains a bulk data record structure).

5.3.2 ServiceName Attribute Description

Table 5.6 Description of the ‘serviceName’ attribute of the TransactionRecord class.

Descriptor	Definition
Attribute name	serviceName
Data type	NormalizedString
Value space	A Normalized String [1-255 characters].
Multiplicity	1
Description	The name of the service that owns the operation described in the transactionRecord. This name is that allocated to the name of the ServiceModel Package in the service PIM.

5.3.3 InterfaceName Attribute Description

Table 5.7 Description of the ‘interfaceName’ attribute of the TransactionRecord class.

Descriptor	Definition
Attribute name	interfaceName
Data type	NormalizedString
Value space	NormalizedString [1-255 characters].
Multiplicity	1
Description	The name of the interface that owns the operation described in the transactionRecord. This name is that allocated to the name of the interface class in the ServiceModel Package in the service PIM.

5.3.4 OperationName Attribute Description

Table 5.8 Description of the ‘operationName’ attribute of the TransactionRecord class.

Descriptor	Definition
Attribute name	operationName
Data type	NormalizedString
Value space	NormalizedString [1-255 characters].
Multiplicity	1
Description	The name of the operation that must be invoked by the end system. The name allocated to operation in the interface class in the ServiceModel Package in the service PIM.

5.3.5 ParameterSet Attribute Description

Table 5.9 Description of the ‘parameterSet’ attribute of the TransactionRecord class.

Descriptor	Definition
Attribute name	parameterSet
Data type	ParameterSet
Value space	container
Multiplicity	1
Description	The container for the set of parameters that is associated with the operation.

5.4 ParameterSet Class Description

Table 5.10 Description of the ‘ParameterSet’ class.

Descriptor	Definition
Class name	ParameterSet
Class type	container
Multiplicity	1
Parents	TransactionRecord
Children	[parameterRecord], unordered
Description	This is the container for the set of parameterRecord descriptions. Each parameter of the operation has its own ‘parameterRecord’ description.

5.4.1 ParameterRecord Attribute Description

Table 5.11 Description of the ‘parameterRecord’ attribute of the ParameterSet class.

Descriptor	Definition
Attribute name	parameterRecord
Data type	ParameterRecord
Value space	container
Multiplicity	1..unbounded, unordered
Description	This is the container for the set of parameterRecord descriptions. Each parameter of the operation has its own ‘parameterRecord’ description. An implementation must be capable of supporting at least ten parameterRecords.

5.5 ParameterRecord Class Description

Table 5.12 Description of the ‘ParameterRecord’ class.

Descriptor	Definition
Class name	ParameterRecord
Class type	container
Multiplicity	1..unbounded, unordered
Parents	ParameterSet
Children	[parameterInvoc, parameterName, parameterType, parameterValue]
Description	This is the container for each parameter associated with the operation.

5.5.1 ParameterInvoc Attribute Description

Table 5.13 Description of the ‘parameterInvoc’ attribute of the ParameterRecord class.

Descriptor	Definition
Attribute name	parameterInvoc
Data type	Enumerated.
Value space	Enumerated as {In, Out}. ‘In’ refers to a parameter sent to the service provider and ‘Out’ for a parameter sent from the service provider.
Multiplicity	1
Description	Identifies whether the parameter is input to, or output from, the service.

5.5.2 ParameterName Attribute Description

Table 5.14 Description of the ‘parameterName’ attribute of the ParameterRecord class.

Descriptor	Definition
Attribute name	parameterName
Data type	NormalizedString
Value space	NormalizedString [1-255 characters].
Multiplicity	1
Description	The name of the parameter being described. This name is that allocated to the parameter in the ServiceModel Package in the service PIM.

5.5.3 ParameterType Attribute Description

Table 5.15 Description of the ‘parameterType’ attribute of the ParameterRecord class.

Descriptor	Definition
Attribute name	parameterType
Data type	Enumerated list.
Value space	Vocabulary-based – see Appendix B.
Multiplicity	1
Description	<p>This identifies the data-type of the parameter and covers all of the data-types used in the set of services used to create the LIS. 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.</p> <p>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.</p>

5.5.4 ParameterValue Attribute Description

Table 5.16 Description of the ‘parameterValue’ attribute of the ParameterRecord class.

Descriptor	Definition
Attribute name	parameterValue
Data type	ParameterValue
Value space	container
Multiplicity	1
Description	This is the actual data structure that is passed in the parameter.

5.6 ParameterValue Class Description

Table 5.17 Description of the ‘ParameterValue’ class.

Descriptor	Definition
Class name	ParameterValue
Class type	container
Multiplicity	1
Parents	ParameterRecord
Children	[guid, guidSet, objectQuery, sequenceIdentifier, text, url, personRecord, personCoreRecord, personRecordSet, groupRecord, groupRecordSet, membershipRecord, membershipRecordSet, courseTemplateRecord, courseTemplateRecordSet, courseOfferingRecord, courseOfferingRecordSet, courseSectionRecord, courseSectionRecordSet, sectionAssociationRecord, sectionAssociationRecordSet, resultRecord, resultRecordSet, lineItemRecord, lineItemRecordSet, lineItemType, membershipIdType, relationship, role, resultStatus, status], xor
Description	This is the container for the data structure that is passed in the parameter. An instance of this container will have only one of the children listed.

5.6.1 Guid Attribute Description

Table 5.18 Description of the ‘guid’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	guid
Data type	GUID
Value space	See Table 5.1.
Multiplicity	XOR conditional
Description	The unique identifier of the object.

5.6.2 GuidSet Attribute Description

Table 5.19 Description of the ‘guidSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	guidSet
Data type	GUIDSet
Value space	container
Multiplicity	XOR conditional
Description	A set of GUIDs. See the appropriate service specification.

5.6.3 QueryObject Attribute Description

Table 5.20 Description of the ‘queryObject’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	queryObject
Data type	QueryObject
Value space	NormalizedString [1..4095] characters.
Multiplicity	XOR conditional
Description	This is the data-type for the query instruction. This is a string with the smallest permitted maximum length of 4095 bytes. 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.

5.6.4 SequenceIdentifier Attribute Description

Table 5.21 Description of the ‘sequenceIdentifier’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	sequenceIdentifier
Data type	SequenceIdentifier
Value space	DateTime: YYYY-MM-DDTHH:MM:SS.NNN
Multiplicity	XOR conditional
Description	<p>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).</p> <p>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 ‘group’ object.</p> <p>All values are to be rounded down at the level of greatest resolution.</p>

5.6.5 Text Attribute Description

Table 5.22 Description of the ‘text’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	text
Data type	Text
Value space	Language dependent String [1-127 characters]. The default language is ‘en-US’.
Multiplicity	XOR conditional
Description	A language dependent string container.

5.6.6 URL Attribute Description

Table 5.23 Description of the ‘url’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	url
Data type	URL
Value space	See Table 5.1.
Multiplicity	XOR conditional
Description	This is the URL of the external data file that is used to contain the relevant objects. An implementation must be capable of supporting URLs of at least 1024 bytes in length i.e., the shortest permitted maximum length.

5.6.7 PersonRecord Attribute Description

Table 5.24 Description of the ‘personRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	personRecord
Data type	PersonRecord
Value space	container
Multiplicity	XOR conditional
Description	A person record. See the IMS GLC Person Management Service Information Model v2.0 specification for more details on the related object structure.

5.6.8 PersonCoreRecord Attribute Description

Table 5.25 Description of the ‘personCoreRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	personCoreRecord
Data type	PersonCoreRecord
Value space	container
Multiplicity	XOR conditional
Description	A core person record. See the IMS GLC Person Management Service Information Model v2.0 specification for more details on the related object structure.

5.6.9 PersonRecordSet Attribute Description

Table 5.26 Description of the ‘personRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	personRecordSet
Data type	PersonRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of person records. See the IMS GLC Person Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.10 GroupRecord Attribute Description

Table 5.27 Description of the ‘groupRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	groupRecord
Data type	GroupRecord
Value space	container
Multiplicity	XOR conditional
Description	A group record. See the IMS GLC Group Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.11 GroupRecordSet Attribute Description

Table 5.28 Description of the ‘groupRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	groupRecordSet
Data type	GroupRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of group records. See the IMS GLC Group Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.12 MembershipRecord Attribute Description

Table 5.29 Description of the ‘membershipRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	membershipRecord
Data type	MembershipRecord
Value space	container
Multiplicity	XOR conditional
Description	A membership record. See the IMS GLC Membership Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.13 MembershipRecordSet Attribute Description

Table 5.30 Description of the ‘membershipRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	membershipRecordSet
Data type	MembershipRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of membership records. See the IMS GLC Membership Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.14 CourseTemplateRecord Attribute Description

Table 5.31 Description of the ‘courseTemplateRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	courseTemplateRecord
Data type	CourseTemplateRecord
Value space	container
Multiplicity	XOR conditional
Description	A course template record. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.15 CourseTemplateRecordSet Attribute Description

Table 5.32 Description of the ‘courseTemplateRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	courseTemplateRecordSet
Data type	CourseTemplateRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of course template records. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.16 CourseOfferingRecord Attribute Description

Table 5.33 Description of the ‘courseOfferingRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	courseOfferingRecord
Data type	CourseOfferingRecord
Value space	container
Multiplicity	XOR conditional
Description	A course offering record. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.17 CourseSectionRecord Attribute Description

Table 5.34 Description of the ‘courseSectionRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	courseSectionRecord
Data type	CourseSectionRecord
Value space	container
Multiplicity	XOR conditional
Description	A course section record. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.18 CourseSectionRecordSet Attribute Description

Table 5.35 Description of the ‘courseSectionRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	courseSectionRecordSet
Data type	CourseSectionRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of course section records. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.19 SectionAssociationRecord Attribute Description

Table 5.36 Description of the ‘sectionAssociationRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	sectionAssociationRecord
Data type	SectionAssociationRecord
Value space	container
Multiplicity	XOR conditional
Description	A section association record. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.20 SectionAssociationRecordSet Attribute Description

Table 5.37 Description of the ‘sectionAssociationRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	sectionAssociationRecordSet
Data type	SectionAssociationRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of section association records. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.21 ResultRecord Attribute Description

Table 5.38 Description of the ‘resultRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	resultRecord
Data type	ResultRecord
Value space	container
Multiplicity	XOR conditional
Description	A result record. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.22 ResultRecordSet Attribute Description

Table 5.39 Description of the ‘resultRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	resultRecordSet
Data type	ResultRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of result records. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.23 ResultValueRecord Attribute Description

Table 5.40 Description of the ‘resultValueRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	resultValueRecord
Data type	ResultValueRecord
Value space	container
Multiplicity	XOR conditional
Description	A resultValue record. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.24 ResultValueRecordSet Attribute Description

Table 5.41 Description of the ‘resultValueRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	resultValueRecordSet
Data type	ResultValueRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of resultValue records. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.25 LineItemRecord Attribute Description

Table 5.42 Description of the ‘lineItemRecord’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	lineItemRecord
Data type	LineItemRecord
Value space	container
Multiplicity	XOR conditional
Description	A lineItem record. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.26 LineItemRecordSet Attribute Description

Table 5.43 Description of the ‘lineItemRecordSet’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	lineItemRecordSet
Data type	LineItemRecordSet
Value space	container
Multiplicity	XOR conditional
Description	A set of lineItem records. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.27 LineItemType Attribute Description

Table 5.44 Description of the ‘lineItemType’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	lineItemType
Data type	LineItemType
Value space	Enumerated list.
Multiplicity	XOR conditional
Description	A lineItemType. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.28 MembershipIdType Attribute Description

Table 5.45 Description of the ‘membershipIdType’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	membershipIdType
Data type	MembershipIdType
Value space	Enumerated list.
Multiplicity	XOR conditional
Description	A membership type. See the IMS GLC Membership Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.29 Relationship Attribute Description

Table 5.46 Description of the ‘relationship’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	relationship
Data type	Relationship
Value space	container
Multiplicity	XOR conditional
Description	The relationship between two Groups. See the IMS GLC Group Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.30 Role Attribute Description

Table 5.47 Description of the ‘role’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	role
Data type	Role
Value space	Enumerated list.
Multiplicity	XOR conditional
Description	A role type. See the IMS GLC Membership Management Service v2.0 Information Model specification for more details on the related object structure.

5.6.31 ResultStatus Attribute Description

Table 5.48 Description of the ‘resultStatus’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	resultStatus
Data type	ResultStatus
Value space	Enumerated list.
Multiplicity	XOR conditional
Description	The status of a Result. See the IMS GLC Outcomes Management Service v1.0 Information Model specification for more details on the related object structure.

5.6.32 Status Attribute Description

Table 5.49 Description of the ‘status’ attribute for the ParameterValue class.

Descriptor	Definition
Attribute name	status
Data type	Status
Value space	Enumerated list.
Multiplicity	XOR conditional
Description	The status of a Course component. See the IMS GLC Course Management Service v1.0 Information Model specification for more details on the related object structure.

6 File Retrieval Approaches

6.1 HTTPS Get

Every implementation of the LIS BDEMS must provide the use of HTTPSget calls to access the data files that are created by a service provider. The call contains the URL provided by the bulk data manifest object in the corresponding ‘announceBulkDataExchange’ message from the service provider (note that several URLs may be supplied and so several HTTPSget calls may be made by the service consumer).

Hypertext Transfer Protocol Secure (HTTPS) is a combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encryption and secure identification of the server. As default, it uses port 443.

The internal logical content structure of the file is as shown in Figure 5.1. The physical format follows the corresponding binding. The parameter ‘transactionId’ may be supplied depending on the implementation requirements.

6.2 Other Protocols

Apart from the file download approach mandated i.e., HTTPSget calls, the bulk data files may be obtained using any protocol that has been agreed (out-of-band) by the two systems.

6.2.1 File Transfer Protocol over SSL (FTPS)

File Transfer Protocol over Secure Sockets Layer (FTPS), commonly referred to as FTP/SSL, is a name used to encompass a number of ways in which FTP software can perform secure file transfers. Each way involves the use of a SSL/TLS layer below the standard FTP protocol to encrypt the control and/or data channels. This is NOT the same as either SSH File Transfer Protocol (SFTP), or FTP over SSH.

AUTH TLS, Explicit FTPS or FTPES, named for the command issued to indicate that TLS security should be used. This is the preferred method according to RFC 4217. The client connects using clear text on port 21 and may negotiate a secure TLS connection during the FTP setup or at any time thereafter. The server may allow non-encrypted FTP should negotiation fail. If the control channel is unencrypted, any subsequent data channels must also be unencrypted; if the control channel is encrypted, the subsequent data channels may be in clear or encrypted (this is a requirement of the *AUTH* mechanism defined under RFC 2228). If the command channel is not encrypted, the protocol is said to be using a clear command channel. If the data channel is not encrypted, the protocol is said to be using a clear data channel. Encrypted data channels and encryption on the control channel can be set up and torn down by the client at any time.

7 Extending and Profiling the Service

7.1 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.

7.1.1 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, 11b].

7.1.2 Proprietary Data Elements

Extensions to the data model are only permitted where the *IMSExtension* class is available i.e., within the data models for the objects contained in the data file (see the corresponding Information Models for how to extend a specific data model of Person, Group, Membership, Course and Outcomes). It is not possible to extend the other parts of the data file or the service parameters.

7.2 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 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 BulkDataExchangeManager interface operations.

CodeMinor Status Code	announce	announceFailure	report	request	ignore	cancel
‘fullsuccess’	Y	Y	Y	Y	Y	Y
‘invaliddata’	Y		Y	Y		
‘invalidtransactionid’		Y	Y		Y	Y
‘incompletedata’	Y		Y			
‘invalidurl’	Y					
‘unsupportedservices’	Y					
‘unsupportedoperations’	Y					
‘expireddate’			Y			
‘savepointerror’				Y		
‘savepointsyncerror’				Y		
‘targetisbusy’	Y	Y	Y	Y	Y	Y
‘unauthorizedrequest’	Y	Y	Y	Y	Y	Y
‘linkfailure’	Y	Y	Y	Y	Y	Y
‘unsupportedLIS’	Y	Y	Y	Y	Y	Y
‘unsupportedLISoperation’	Y	Y	Y	Y	Y	Y
‘unknownservice’	Y	Y	Y	Y	Y	Y
‘unknownoperation’	Y	Y	Y	Y	Y	Y

There is a set of status codes that must be supported by each of the Bulk Data Exchange 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 provided service.

Appendix B The Vocabularies

The vocabularies listed in the following Tables 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.

B1 TransactionFailStatus Vocabulary

The vocabulary for 'transactionFailStatus' is listed in Table B1.1.

Table B1.1 The vocabulary for 'transactionFailStatus'.

Vocabulary Association	Description of the Vocabulary
Type of status codes returned in the 'transactionFailStatus' string.	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> • unsupportedLISservice • unsupportedLISoperation • invaliddata • incompletdata • unauthorisedrequest • idallocfail • overflowfail • idallocinusefail • partialdatastorage • unknownobject • deletefailure • targetreadfailure • savepointerror • savepointsyncerror • unknownquery • unknownvocabulary • unknownmdvocabulary • unknownextension • unknownservice • unknownoperation • toomuchdata

- Rule B.1-01: unsupportedLISservice – the code returned if the target does not support the identified service;
- Rule B.1-02: unsupportedLISoperation – the code returned if the target does not support the LIS operation;
- Rule B.1-03: invaliddata – the code returned if the target detects that some of the supplied data record contains invalid data;
- Rule B.1-04: incompletdata – the code returned if the target detects that some of the required data record has not been supplied;
- Rule B.1-05: unauthorisedrequest – the code returned by the target if the source does not have the authority to issue the request;
- Rule B.1-06: idallocfail – code returned if the target cannot create the new object as no GUIDs are available;

- Rule B.1-07: overflowfail – the code returned by the target if the write request cannot be completed because there is insufficient database storage;
- Rule B.1-08: idallocinusefail – the code returned if the target cannot create the new object because the GUID is already allocated to another object;
- Rule B.1-09: partialdatastorage – the code returned if the target has been able to store some, but not all, of the data record;
- Rule B.1-10: unknownobject – the code returned if the target cannot locate the identified object;
- Rule B.1-11: deletefailure – the code returned if the target cannot delete the identified object;
- Rule B.1-12: targetreadfailure – the code returned if the target has been unable to read the required data record;
- Rule B.1-13: savepointerror – the code returned by the target if an error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database;
- Rule B.1-14: savepointsyncerror – the code returned if the save point reference from the source was later than that of the target system;
- Rule B.1-15: unknownquery – the code returned by the target if the requested query string cannot be interpreted and actioned;
- Rule B.1-16: unknownvocabulary – the code returned by the target if a vocabulary or vocabulary term is unrecognized;
- Rule B.1-17: unknownmdvocabulary – the code returned by the target if a metadata vocabulary or vocabulary term is unrecognized;
- Rule B.1-18: unknownextension – the code returned by the target if an extension vocabulary or vocabulary term is unrecognized;
- Rule B.1-19: unknownservice – the code returned by the target the service is not part of LIS;
- Rule B.1-20: unknownoperation – the code returned by the target if the operation is unknown i.e., it is not a valid LIS operation.
- Rule B.1-21: toomuchdata – the code returned by the target if the read operation has attempted to return more data than can be sustained by the message transport system.

B2 ParameterType Vocabulary

The vocabulary for 'parameterType' is listed in Table B2.1.

Table B2.1 The vocabulary for 'parameterType'.

Vocabulary Association	Description of the Vocabulary
The data-type for the parameter in the 'parameterType' string.	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> • GUID • GUIDSet • QueryObject • SequenceIdentifier • Text • URL • CourseOfferingRecord • CourseOfferingRecordSet • CourseSectionRecord • CourseSectionRecordSet

Vocabulary Association	Description of the Vocabulary
	<ul style="list-style-type: none"> • CourseTemplateRecord • CourseTemplateRecordSet • GroupRecord • GroupRecordSet • LineItemRecord • LineItemRecordSet • MembershipRecord • MembershipRecordSet • PersonCoreRecord • PersonRecord • PersonRecordSet • ResultRecord • ResultRecordSet • ResultValueRecord • ResultValueRecordSet • SectionAssociationRecord • SectionAssociationRecordSet • LineItemType • MembershipIdType • Relationship • Role • ResultStatus • Status

- Rule B.2-01: GUID – denotes that the parameter is for a GUID;
- Rule B.2-02: GUIDset – denotes that the parameter is for a set of GUIDs;
- Rule B.2-03: QueryObject – denotes that the parameter is for a QueryObject;
- Rule B.2-04: SequenceIdentifier – denotes that the parameter is for a Sequence Identifier;
- Rule B.2-05: Text – denotes that the parameter is for a Text string;
- Rule B.2-06: URL – denotes that the parameter is for a URL;
- Rule B.2-07: CourseOfferingRecord – denotes that the parameter is for a CourseOffering record;
- Rule B.2-08: CourseOfferingRecordSet – denotes that the parameter is for a set of CourseOffering records;
- Rule B.2-09: CourseSectionRecord – denotes that the parameter is for a CourseSection record;
- Rule B.2-10: CourseSectionRecordSet – denotes that the parameter is for a set of CourseSection records;
- Rule B.2-11: CourseTemplateRecord – denotes that the parameter is for a CourseTemplate record;
- Rule B.2-12: CourseTemplateRecordSet – denotes that the parameter is for a set of CourseTemplate records;
- Rule B.2-13: GroupRecord – denotes that the parameter is for a GroupRecord;
- Rule B.2-14: GroupRecordSet – denotes that the parameter is for a set of Group records;
- Rule B.2-15: LineItemRecord – denotes that the parameter is for a LineItem record;
- Rule B.2-16: LineItemRecordSet – denotes that the parameter is for a set of LineItem records;
- Rule B.2-17: MembershipRecord – denotes that the parameter is for a Membership record;

- Rule B.2-18: MembershipRecordSet – denotes that the parameter is for a set of Membership records;
- Rule B.2-20: PersonCoreRecord – denotes that the parameter is for a PersonCore record;
- Rule B.2-21: PersonRecord – denotes that the parameter is for a Person record;
- Rule B.2-22: PersonRecordSet – denotes that the parameter is for a set of Person records;
- Rule B.2-23: ResultRecord – denotes that the parameter is for a Result record;
- Rule B.2-24: ResultRecordSet – denotes that the parameter is for a set of Result records;
- Rule B.2-25: ResultValueRecord – denotes that the parameter is for a ResultValue record;
- Rule B.2-26: ResultValueRecordSet – denotes that the parameter is for a set of ResultValue records;
- Rule B.2-27: SectionAssociationRecord – denotes that the parameter is for a SectionAssociation record;
- Rule B.2-28: SectionAssociationRecordSet – denotes that the parameter is for a set of SectionAssociation records;
- Rule B.2-29: LineItemType – denotes that the parameter is for a LineItemType;
- Rule B.2-30: MembershipIdType – denotes that the parameter is for a MembershipIdType;
- Rule B.2-31: Relationship – denotes that the parameter is for a Relationship;
- Rule B.2-32: Role – denotes that the parameter is for a Role;
- Rule B.2-33: ResultStatus – denotes that the parameter is for a ResultStatus;
- Rule B.2-34: Status – denotes that the parameter is for a Status.

B3 FileValue Vocabulary

The vocabulary for ‘fileValue’ (when ‘filterType=object’) is listed in Table B3.1.

Table B3.1 The vocabulary for ‘filterType’.

Vocabulary Association	Description of the Vocabulary
The permitted entries for the ‘filterValue’ string when ‘filterType=object’.	<p>The core vocabulary is:</p> <ul style="list-style-type: none"> • All • Person • Group • Membership • CourseTemplate • CourseOffering • CourseSection • SectionAssociation • LineItem • Result • ResultValue • AllCourse • AllOutcomes

- Rule B.3-01: All – used to denote that synchronization data for all of the LIS records is required;
- Rule B.3-02: Person – used to denote that synchronization data for all of the LIS Person records is required;

- Rule B.3-03: Group – used to denote that synchronization data for all of the LIS Group records is required;
- Rule B.3-04: Membership – denotes that synchronization data for all of the LIS Membership records is required;
- Rule B.3-05: CourseTemplate – used to denote that synchronization data for all of the LIS CourseTemplate records is required;
- Rule B.3-06: CourseOffering – used to denote that synchronization data for all of the LIS CourseOffering records is required;
- Rule B.3-07: CourseSection – denotes that synchronization data for all of the LIS CourseSection records is required;
- Rule B.3-08: SectionAssociation – used to denote that synchronization data for all of the LIS SectionAssociation records is required;
- Rule B.3-09: LineItem – denotes that synchronization data for all of the LIS LineItem records is required;
- Rule B.3-10: Result – used to denote that synchronization data for all of the LIS Result records is required
- Rule B.3-11: ResultValue – denotes that synchronization data for all of the LIS ResultValue records is required
- Rule B.3-12: AllCourse – used to denote that synchronization data for all of the LIS Course (CourseTemplate, CourseOffering, CourseSection and SectionAssociation) records is required;
- Rule B.3-13: AllOutcomes – used to denote that synchronization data for all of the LIS Outcomes (LineItem, Result and ResultValue) records is required.

B4 FailureCode Vocabulary

The vocabulary for ‘failureCode’ is listed in Table B4.1.

Table B4.1 The vocabulary for ‘failureCode’.

Vocabulary Association	Description of the Vocabulary
The permitted entries for the ‘failureCode’.	The core vocabulary is: <ul style="list-style-type: none"> • savepointerror

- Rule B.4-01: savepointerror – the code returned by the service provider if an error has occurred in the processing of the save-point identifier information making it impossible to read the correct objects from the database.

About This Document

Title:	IMS GLC Bulk Data Exchange Management Service (BDEMS) 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:	Final 1.0
Status:	Final Release
Summary:	This document contains the IMS GLC Bulk Data Exchange Management Service v1.0 Information Model. This asynchronous service is used to exchange structured bulk data records. The core of the definition is a simple announcement/request service that provides a description and location of the file containing the structured bulk data. A Web Service file retrieval protocol is used to obtain the bulk data record. Once the data retrieval has been completed the corresponding status information is returned to the service provider.
Revision Information:	Original Final Release
Purpose:	This document is made available for adoption by the public community at large.
Document Location:	http://www.msglobal.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	Cengage (USA)	Colin Smythe	IMS GLC (UK)
Linda Feng	Oracle (USA)	Reinhold Staudinger	Blackboard (USA)
Chris Hatton	Pearson (USA)	Nick Terrible	University of Wisconsin (USA)
John 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

A

Abstract Framework .. 10, 11, 12
 API..... 11, 13, 15
 Attributes
 BDEMS
 bulkBlockDataFile ... 29, 30
 bulkBlockManifest . 18, 20,
 21, 22, 23, 24
 bulkBlockReport..... 21, 22
 operationSet 33, 34
 parameterRecord..... 68, 69
 parameterSet 66, 68
 parameterValue 69, 71
 serviceRecord 32, 33
 serviceSet..... 30, 32
 transactionRecord ... 65, 66,
 67
 transactionReport.... 37, 38,
 39, 40, 41, 42
 Course
 association 78
 courseOffering 77
 courseSection..... 77
 courseTemplate..... 76
 offering 10, 77
 section..... 61, 77, 78
 sectionAssociation 78
 template 76, 77
 groupRecord 71, 75
 lineItem..... 71, 80
 lineItemRecord 71, 80
 Membership
 membership 76, 81
 membershipRecord 71, 76
 personRecord.. 71, 74, 75, 76,
 77, 78
 resultRecord..... 71, 79, 80
 resultValue..... 79, 80
 resultValueRecord 79

B

Binding technologies
 WSDL..... 10, 12, 61
 Bulk Data Exchange
 Management Service 1, 10,
 11, 12, 13, 14, 25, 61, 66, 83,
 86, 92

C

Classes
 BDEMS

BulkBlockDataFile . 30, 31,
 32
 BulkBlockManifest . 18, 20,
 21, 22, 23, 24, 28, 29,
 30, 31, 37
 BulkBlockReport 21, 36,
 37, 38, 40, 41, 42, 44,
 45, 46, 49, 50, 53, 54,
 55, 56, 57, 58, 59
 BulkDataRecord. 64, 65, 66
 OperationSet 33, 34, 35
 ParameterRecord68, 69, 70,
 71
 ParameterSet 66, 68, 69
 ParameterValue. 69, 71, 72,
 73, 74, 75, 76, 77, 78,
 79, 80, 81, 82
 ServiceRecord 32, 33, 34
 ServiceSet 30, 32, 33
 TransactionRecord .. 65, 66,
 67, 68
 TransactionReport... 37, 38,
 39, 40, 41, 42, 43, 44,
 45, 46, 47, 48, 50, 51,
 52, 53, 54, 55, 56, 57,
 58, 59, 60
 Course
 Association... 87, 88, 90, 91
 CourseOffering . 77, 89, 90,
 91
 CourseSection ... 77, 89, 90,
 91
 CourseTemplate 76, 89, 90,
 91
 Offering..... 71, 88
 Section .. 16, 25, 28, 30, 62,
 71, 88
 SectionAssociation.. 78, 90,
 91
 Template 71, 88
 Group .. 10, 22, 75, 81, 84, 89,
 90, 91
 Relationship . 71, 81, 88, 90
 GroupRecord.... 71, 75, 88, 89
 LineItem..... 80, 88, 89, 90, 91
 LineItemRecord 71, 80, 88, 89
 MembershipRecord 71, 76, 88,
 89
 Outcomes
 ResultRecord 71, 79, 80, 88,
 90
 ResultValue..... 90, 91
 ResultValueRecord . 79, 88,
 90

Person.. 10, 22, 74, 75, 76, 77,
 78, 84, 90
 Result 71, 82, 90, 91
 Common Services 10
 Conformance 13
 Course 10, 76, 77, 78, 82, 84, 91
 Course Management Service. 10,
 76, 77, 78, 82
 Course Structures
 Association..... 87, 88, 90, 91
 CourseOffering. 77, 89, 90, 91
 CourseSection .. 77, 89, 90, 91
 CourseTemplate 76, 89, 90, 91
 Offering 71, 88
 Section. 16, 25, 28, 30, 62, 71,
 88
 SectionAssociation. 78, 90, 91
 Template..... 71, 88

G

Group Management Service.. 10,
 75, 81

I

Interface Class
 BulkDataExchangeManager
 16, 17, 85

L

Learning Information Services
 10, 11, 12, 14, 16
 Learning Information Services
 Best Practices &
 Implementation Guide 12
 LIS
 Bulk Data Exchange
 Management Service . 1, 10,
 11, 12, 13, 14, 25, 61, 66,
 83, 86, 92
 Course Management Service
 10, 76, 77, 78, 82
 Group Management Service
 10, 75, 81
 Membership Management
 Service..... 10, 76, 81, 82
 Outcomes Management
 Service.... 10, 79, 80, 81, 82
 Person Management Service
 10, 74, 75, 76, 77, 78

M

Membership Management
Service 10, 76, 81, 82

O**Operations****BDEMS**

announceBulkDataExchange
.....17, 18, 20, 22, 23, 25,
26, 37, 38, 42, 44

announceFailureBulkDataE
xchange..... 17, 20, 26, 44

cancelBulkDataExchange
..... 17, 24, 26

ignoreBulkDataExchange
..... 17, 23, 26

reportBulkDataExchange
.....17, 20, 21, 22, 23, 24,
25, 37

requestBulkDataExchange
.....17, 20, 22, 24, 25, 44

Outcomes Management Service
..... 10, 79, 80, 81, 82

P

Person Management Service. 10,
74, 75, 76, 77, 78

Profiling 11, 84

S

SectionAssociation..... 78, 90, 91
Services

Bulk Data Exchange

Management 10, 11, 12, 13,
25, 61, 86, 92

Course Management Service

..... 10, 76, 77, 78, 82

Group Management

10, 75, 81
Membership Management. 10,
76, 81, 82

Outcomes Management

Service ... 10, 79, 80, 81, 82

Person Management.... 10, 74,
75, 76, 77, 78

Specifications**IMS****Enterprise Service****Best Practices &****Implementation**

Guide 12

Status Codes..... 85

Status Codes.. 11, 15, 18, 20, 21,
22, 23, 24, 85, 86

deletefailure 87, 88, 90

expireddate..... 85

fullsuccess. 18, 20, 21, 22, 23,
24, 85

idallocfail 87, 90

idallocinusefail..... 87, 88, 90

incompletedata 18, 20, 21, 23,
24, 85, 87, 90

invaliddata. 18, 21, 22, 85, 87,
90

invalidurl 18, 85

linkfailure 85, 86

overflowfail 88

partialdatastorage ... 87, 88, 90

savepointerror.. 20, 22, 85, 87,
88, 90, 91

savepointsyncerror 22, 85, 87,
88

targetisbusy 85, 86

targetreadfailure 87, 88, 90

unauthorizedrequest 85, 86

unknownobject 87, 88, 90

unknownquery 87, 88

unknownvocabulary 87, 88

unsupported 85

unsupportedoperation .. 19, 85,
87, 90

unsupportedservice 18, 85, 87,
88, 89, 90, 91

V

Vocabularies..... 11, 87

W

WDSL 10, 12, 61

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 GLC BDEMS Information Model v1.0 Final Release v1.0

Date: 30 June 2011