170-WP-018-001

Catalog Interoperability Protocol Catalog Translator Design Document for the ECS Project

White Paper

White Paper--Not intended for formal review or government approval.

May 1998

Prepared Under Contract NAS5-60000

RESPONSIBLE ENGINEER

A. V. Gupte /s/ 5/18/98

Ajay V. Gupte, Senior Systems Engineer Date
EOSDIS Core System Project

SUBMITTED BY

George Percivall /s/ 5/18/98

George S. Percivall, Lead Systems Engineer Date
EOSDIS Core System Project

Raytheon Systems Company
Upper Marlboro, Maryland

This page intentionally left blank.

Abstract

This design document defines and describes the Catalog Translator (Translator) component of the Catalog Interoperability Protocol (CIP) Prototype. The Translator translates the CIP requests to ECS Services and converts the ECS responses to CIP responses. The design is based on the CIP Protocol, Release B Specification and System Design Document (SDD) and ECS Release B. However, the description of some sections of the prototype, which use previously existing software modules, reference CIP Release A and V0-to-ECS gateway. These differences are highlighted whenever they occur.

Note that while the high level design is based on ECS Release B, design of the phase I, reflected in this version of the prototype is constrained by re-use of available components, specifically the CIP-A Demonstrator (CIP-AD). It is expected that some of the interfaces and objects will change during the next design iteration.

Keywords: CIP, ICS, ECS, Prototype, Catalog Translator, Translator, CIP-A Demonstrator

This page intentionally left blank.

Contents

Abstract

1. Introduction

1.1 Purpose 1-1

1.2 Organization 1-1

1.3 Acknowledgments 1-1

1.4 Review and Approval 1-1

2. CIP-ECS Catalog Translator Architecture Overview

2.1 CIP-ECS Catalog Translator Architecture 2-1

2.2 CIP and ICS 2-2

2.3 ECS Subsystems 2-2

3. CIP-ECS Catalog Translator Catalog Translator

3.1 CIP/ECS Catalog Translator Overview 3-1

3.1.1 CIP Server Objects 3-3

3.1.2 Search and Response Processing 3-3

3.1.3 ECS Interfaces 3-6

3.1.4 Persistent Data 3-7

3.1.5 Translator/Controller 3-8

3.2 Translator vs. ECS Object Interaction Event Trace Diagrams 3-8

3.2.1 Search Request/Response 3-9

3.2.2 Present Request/Response 3-9

4. Re-usable/COTS Components

4.1 Re-usable/COTS Components 4-1

4.2 Candidate Software Components 4-1

4.2.1 Z39.50 Implementations 4-1

4.2.2 CIP Release A Demonstrator (CIP-AD) 4-2

4.2.3 ECS Data Sub-System 4-2

4.2.4 Relational Database Management Systems (RDBMS) 4-3

4.3 Chosen Software Components 4-3

4.3.1 Development Environment 4-3

4.3.2 CIP A Demonstrator 4-4

4.3.3 Mini SQL 4-4

4.3.4 DBV OSI II Toolkit 4-4

4.3.5 ECS Data Sub-System 4-5

Figures

2.1-1 Catalog Translator Architecture 1

3.1-1 Functional Model of the CIP-ECS Catalog Translator Catalog Translator 1

3.1-2 Catalog Translator Object Model 2

3.1-3 CIP Server Interface Object Diagram 3

3.1-4 Search and Response Object Model 4

3.1-5 CIP-ECS Query Mapping Object Model 5

3.1-6 CIP-ECS Schema Mapping Object Model 5

3.1-7 Session Data Object Model 6

3.1-8 ECS Interface Object Model 7

3.1-9 Persistent Data Object Model 8

3.1-10 Translator Object Model 8

3.2-1 CIP-Domain: Search Request/Response Event Trace 10

3.2-2 ECS-Domain: Inventory Search Request/Response Event Trace 11

3.2-3 CIP-Domain: Present Request/Response Event Trace 12

3.2-4 ECS-Domain: Inspect Request/Response Event Trace 13

Appendix A. CIP/ECS Prototype Requirements

Appendix B. CIP/ECS Attribute Mapping

Appendix C. Universal Resource to Item Descriptor ID Mapping

Abbreviations and Acronyms

This page intentionally left blank.

1. Introduction

1.1 Purpose

The Catalog Interoperability Protocol/EOSDIS Core System Catalog Translator Prototype is a proof-of-concept software system, designed to provide translation of services between ICS and ECS domains for directory queries (catalog searches), inventory queries, search requests and browse requests. The Translator prototype translates the ICS requests to ECS services and converts the ECS responses to ICS responses. This functionality allows the users in the ICS domain to interrogate ECS without specific ECS knowledge.

This design document describes the design and functionality of the Catalog Translator.

1.2 Organization

This paper is organized as follows:

- Section 1 introduces and describes the purpose and organization of the paper.
- Section 2 provides an overview of the architecture of the prototype.
- Section 3 describes the design of the catalog translator which is used to translate queries from ICS, originating at a CIP client, into ECS queries.
- Section 4 describes existing software components which are the basis for the construction of the prototype and the choices that were made.
- Appendix A lists the requirements that dictated the construction of the CIP-ECS Catalog Translator.
- Appendix B presents the attributes mapping tables that shows the compound attributes the elements in CIP and ECS domains and the mappings from CIP-to-ECS.
- Appendix C describes the ECS notion of Universal Resource and the CIP domain equivalent of Item Descriptor ID and then details the mapping from one to other.

1.3 Acknowledgments

This document was developed with input from Ananth Rao, Eric S. Martin and Janet Hylton of Raytheon Systems Company.

1.4 Review and Approval

This White Paper is an informal document approved at the Office Manager level. It does not require formal Government review or approval; however, it is submitted with the intent that review and comments will be forthcoming.

The ideas expressed in this White Paper are valid from April 1998 through October 1998. Questions regarding technical information contained within this Paper should be addressed to the following ECS and/or GSFC contacts:

- ECS Contacts
  - Ajay V. Gupte, Senior Systems Engineer, (301) 883-4178, agupte@eos.hitc.com
- GSFC Contacts
  - Yonsook Enloe, ESDIS, (301)614-5122, yonsook@harp.gsfc.nasa.gov

Questions concerning distribution or control of this document should be addressed to:

Data Management Office
The ECS Project Office
Raytheon Systems Company
1616 McCormick Drive
Upper Marlboro, Maryland 20774-5372

2. CIP-ECS Catalog Translator Architecture Overview

2.1 CIP-ECS Catalog Translator Architecture

The Catalog Translator provides a ICS interface for interrogating the ECS domains. In this system, CIP queries requesting access to ECS products, originating from CIP clients in the ICS domain are sent to the Translator, which translates the incoming query into ECS query format, maps CIP attributes into attributes in the ECS domain and submits it to the ECS server for execution. See Section 3.2 for a detailed scenario in form of event traces.

The architecture and the location of the Catalog Translator within the ICS single site system is shown in Figure 2.1-1.

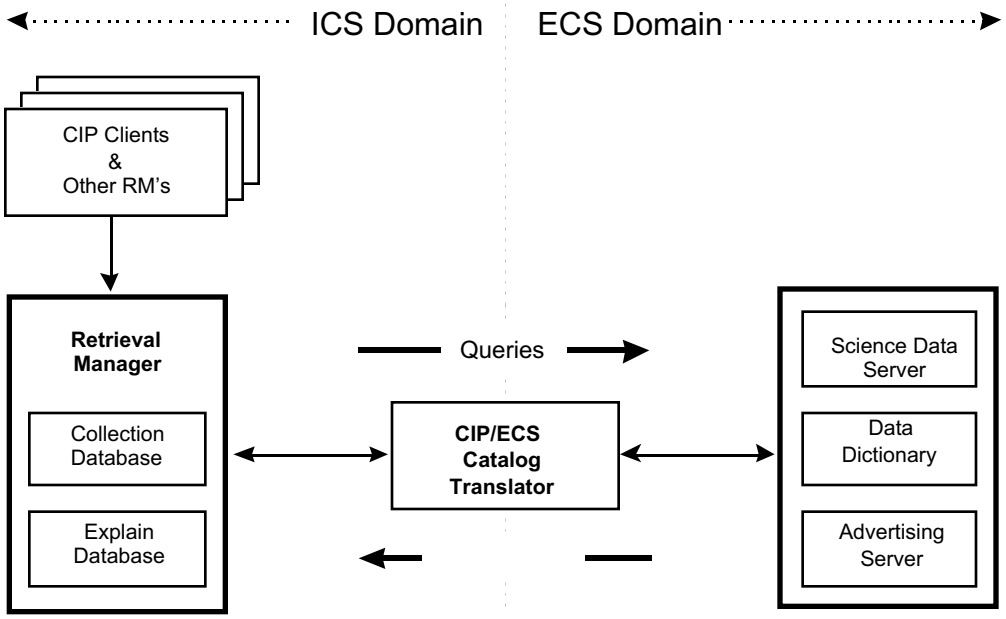

Figure 2.1-1. Catalog Translator Architecture

A portion of the Catalog Translator re-uses components developed by CEO as the CIP A Demonstrator (CIP-AD). The CIP-AD, described in Sections 4.2.2 and 4.3.2, consists of implementations of the CIP Client and the Retrieval Manager. Catalog Translator provides one of the three components that are needed for interoperability between the ECS and CIP domains. The remaining components of the prototype required to show interoperability are:

- Order Handling System (OHS) Translator
- ECS-ICS Gateway

This document only addresses the design and the architecture of the CIP-ECS Catalog Translator. Separate documents will address the other components. In addition, this document also addresses the mapping of the CIP and the ECS attributes that lend to the interoperation of the two domains, the event traces to show the object level interaction among the objects of the catalog translator.

2.2 CIP and ICS

The Catalog Interoperability Protocol (CIP) is described in the CEOS CIP Specification CEOS/WGISS/PTT/CIP-B, Issue 2.2, March 1997. The INFEO modifications and adaptations have been defined in the CEO Architecture Design Document and the System Requirements Document that apply to INFEO.

The Interoperable Catalogue System (ICS) is described in the CEOS System Design Document (SDD) CEOS/WGISS/PTT/SDD, Issue 1.2, March 1997.

It is anticipated that as the documents evolve, the prototype will also be modified to conform to the System Design and Specification Documents. Updates to the System Design Document and CEOS CIP Specification are planned for release in May 1998.

2.3 ECS Subsystems

ECS interfaces are defined in the EOSDIS Core System (ECS) Application Programming Interface (API) Interface Definition Document (IDD) for the ECS Project 819-RD-001-003, Issued October 1996.

ECS Subsystem interfaces are also defined in Version 2.0 Segment/Design Specification, 305-CD-100-001, Issued March 1998.

Science Data Server Interfaces are defined in Release B Science Data Server (SDSRV) Public Interface Overview for the ECS project, Issued March 1998.

Science Data Server Interfaces and Design are defined in ‘Change Pages for Release-B SDPS Data Server Subsystem Design Specification’, 430-TP-008-001, Issued June 1996.

The Design Patterns used in the ECS subsystem use the designs from ‘Design Patterns: Elements of Reusable Object-Oriented Software’ by Erich Gamma, Richard Helm, et al. Published by Addison-Wesley, 1995, ISBN 0201633612.

Other ECS Documents may be used if needed.

3. CIP-ECS Catalog Translator Catalog Translator

3.1 CIP/ECS Catalog Translator Overview

The Catalog Translator is used to translate incoming queries from CIP format to ECS format and also maps incoming search attributes from values in the CIP domain to attribute values in the ECS domain. After execution of the search request at the ECS server, the Catalog Translator also translates result sets from ECS format to CIP format before relaying them back to the client. For browse type requests, where image data is returned, the image is returned directly back to the CIP server and no format translation is done. Figure 3.1-1 shows the Functional Model of the Catalog Translator` and Figure 3.1-2 shows the Object Model of the Catalog Translator.

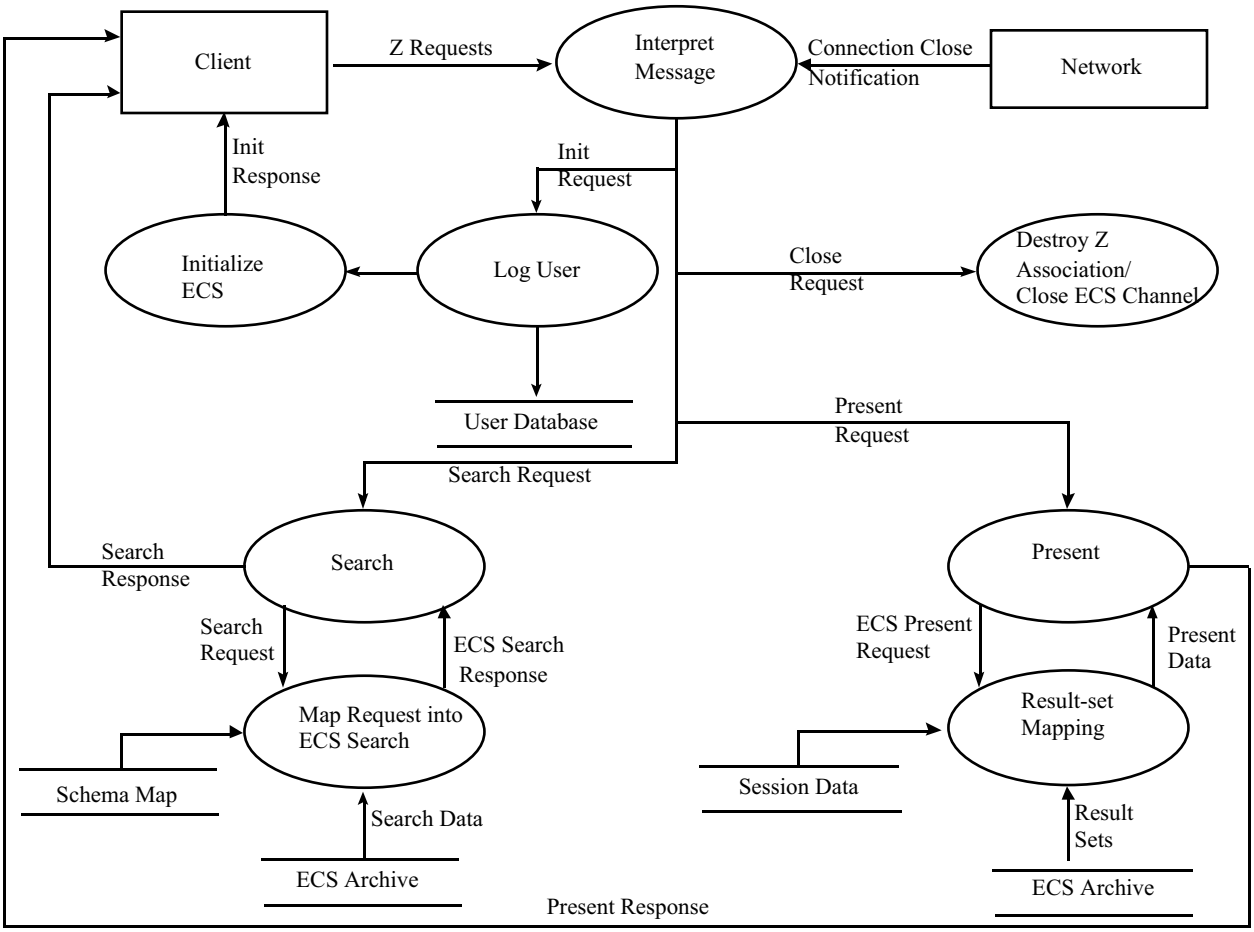

Figure 3.1-1. Functional Model of the CIP-ECS Catalog Translator Catalog Translator

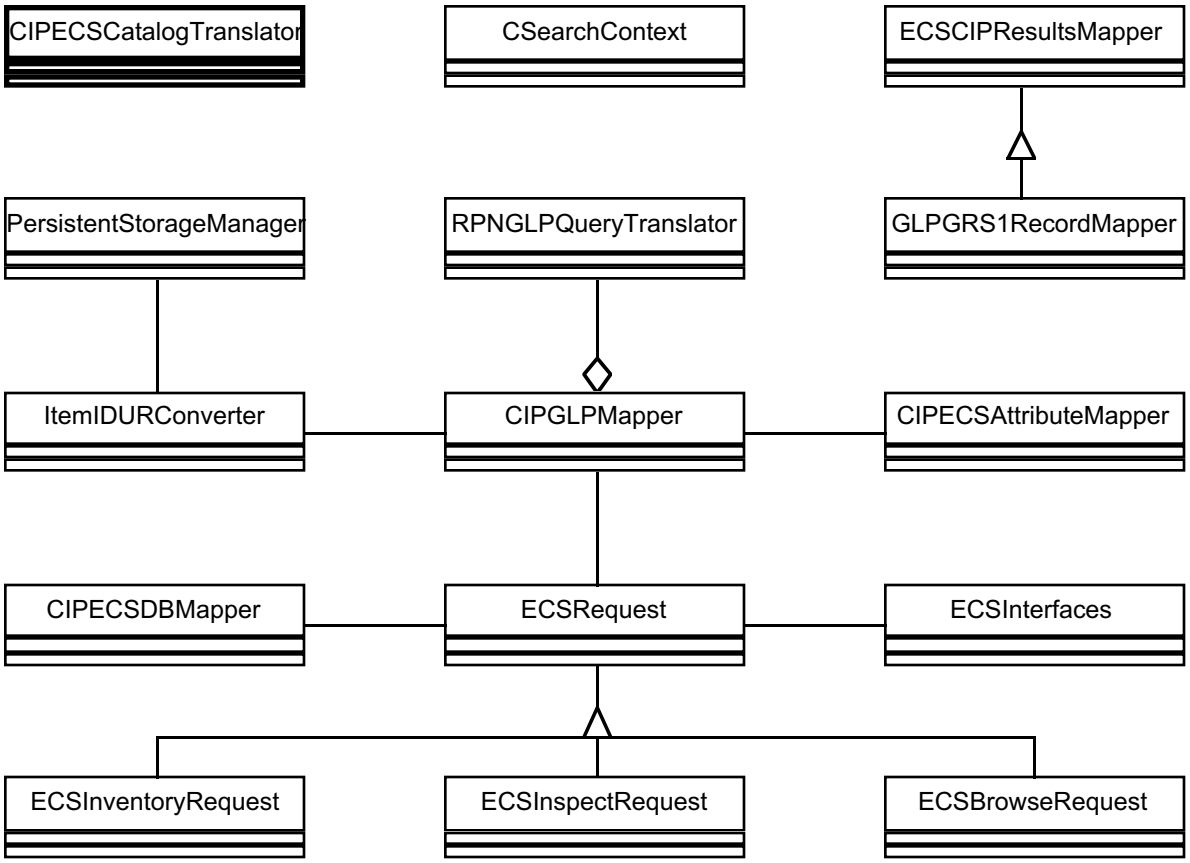

Figure 3.1-2. Catalog Translator Object Model

The Catalog Translator classes can be grouped into areas that define specific functionality. Multiple classes from the Catalog Translator object model in Figure 3.1-2, cooperate to carry through a specific function, since each class is related to others in some manner. Hence the following sections, define the objects by the functions that they perform instead of individual classes. The functional areas internal to the Catalog Translator, needed for mapping CIP Request to the ECS Domain and for mapping the ECS Results back to CIP Domain are as follows:

- CIP Server Objects
- Search and Response Processing
- ECS Interfaces
- Persistent Data
- Translator/Controller

3.1.1 CIP Server Objects

The CIP Server functional area contains the classes that listen for new incoming connection requests from Retrieval Managers on a well known port, and create new processes to service each incoming connection request. The classes in this module interface directly to interface functions in the Z39.50 communications tool-kit which provides the underlying TCP/IP communications mechanisms. The classes that comprise the CIP Server, shown below in Figure 3.1-3, are re-used from the CIP-A Demonstrator and interface directly with the DBV-OSI II object libraries.

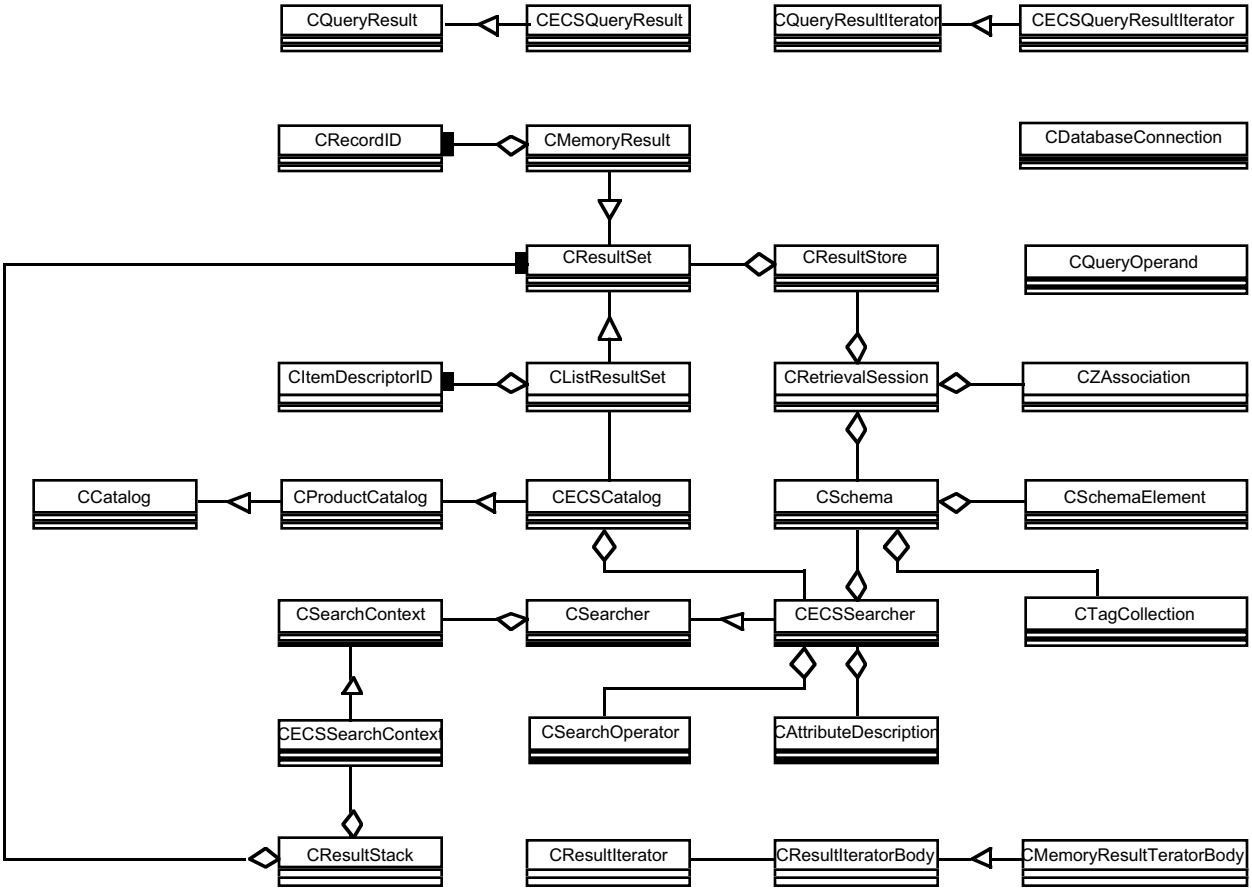

Figure 3.1-3. CIP Server Interface Object Diagram

3.1.2 Search and Response Processing

This Search and Response Processing function comprises of the classes containing the search requests served by the Catalog Translator, and the corresponding result classes. CIP search requests can be of two types: Collection Search and Product Search. CIP Collection type search

requests are resolved by the Retrieval Manager, in conjunction with pre-populated local persistent databases and are not forwarded to the Catalog Translator. Product Search request types are serviced by the Catalog Translator. The CIP-ECS Query Mapping and request processing is performed by the classes RPNGLPQueryTranslator, CIPGLPMapper, CIPECSAttributeMapper and ECSRequest. The Result Records and Response processing is performed by CIPECSAttributeMapper, GLPGRS1RecordMapper and ECSCIPResultsMapper Classes. All these classes are shown below in Figure 3.1-4 which shows the Search and Response Object Model.

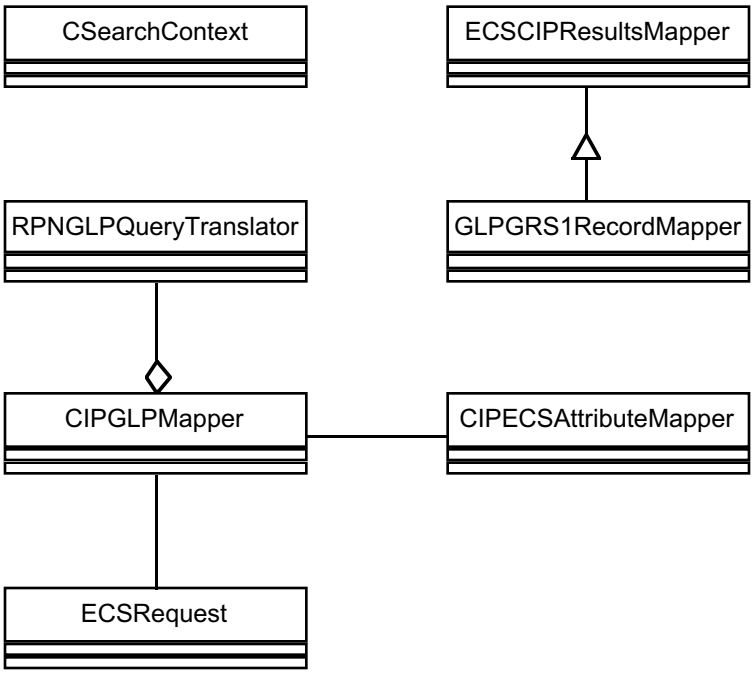

Figure 3.1-4. Search and Response Object Model

3.1.2.1 CIP-ECS Query Mapping

The CIP-ECS Query Mapping although part of the Search Processing is separately defined here due to the complexity of the task. It is also considered a separate area because another group of high-level classes cooperate to perform this function as shown in Figure 3.1-5. The CIP protocol is built on top of the ANSI Standard Z39-50 protocol, the primary underlying query language of which is Reverse Polish Notation (RPN). Thus all incoming queries into the Catalog Translator are formatted in RPN. Since the primary query language for the ECS domain is a parameterized attribute list – the GL-Parameter List, the RPN Queries must be converted into GL-Parameter List form. The CIPGLPMapper class is the central class which converts the queries with the RPNGLPQueryTranslator, CIPECSAttributeMapper and CIPECSDBMapper working as helper classes in syntax translation. The ItemIDURConverter provides the service of locating the products within the ECS domain.

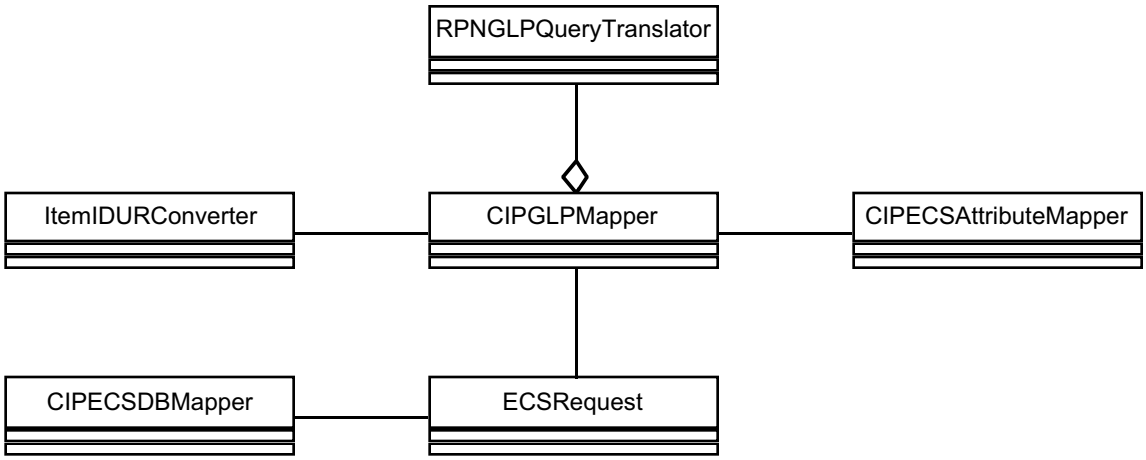

Figure 3.1-5. CIP-ECS Query Mapping Object Model

3.1.2.2 CIP-ECS Schema Mapping

In addition to query mapping as described above, the Catalog Translator also needs to map CIP attributes into the equivalent ECS attributes and vice-versa. The schema mapping table is shown in Appendix B. In order to support a flexible mapping paradigm, this mapping table is stored in an external relational database table. The database is constructed by the RM administrator (RMA) with the appropriate privileges, using the CIP/ECS attribute map. This scheme allows modification to the mapping tables, without requiring any code re-compilation. For attributes that cannot be mapped through a simple one-to-one <CIP-ECS> pair, this module supports more complex mappings. These might be required for cases such as units conversions, date type conversions and integer to character conversions. Figure 3.1-6 shows the CIP-ECS Schema Mapping Object Model. The classes that cooperate for this function are ItemIDURMapper, CIPGLPMapper, CIPECSAttributeMapper and CIPECSDBMapper.

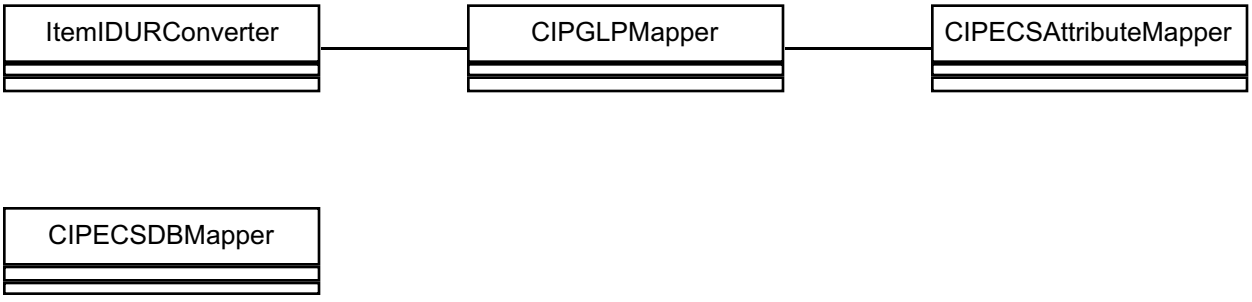

Figure 3.1-6. CIP-ECS Schema Mapping Object Model

3.1.2.3 Session Data

The classes that perform this function provides an interface to session data that may either be specified by the CIP Client at initialization time such as *preferredMessageSize* and *exceptionalRecordSize*, or result-set mapping identifiers that are created during result-set generation. This session data is deleted at the end of the Z39.50 session. The Persistent Data is also managed by these classes and that fucntionality is described in Section 3.1.4. The following classes in conjunction with other lower level classes will manage session specific data that is deleted at the end of one user session. The Figure 3.1-7 shows the Session Data Mapping Object Model.

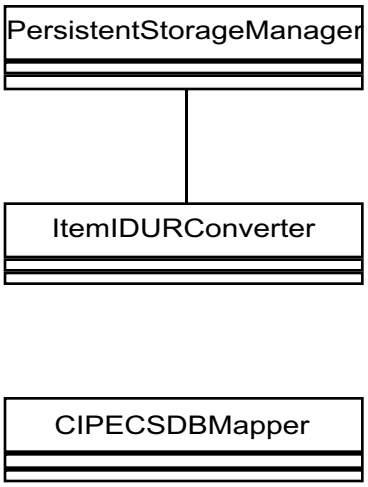

Figure 3.1-7. Session Data Object Model

3.1.3 ECS Interfaces

The ECS Interfaces function represents the classes that make up the ECS Science Data Server client proxy interface. Once the CIP Domain Requests are converted to those required in ECS domain, the ECS Interface classes are used to communicate the request to ECS sub-systems. The results and status information returned by ECS via the interface is then converted to that required in CIP domain. The Science data server can be accessed and queried by these interfaces. The event traces in Section 3.2 expand on the use of these classes. The Figure 3.1-8 shows the ECS Interface Object Model.

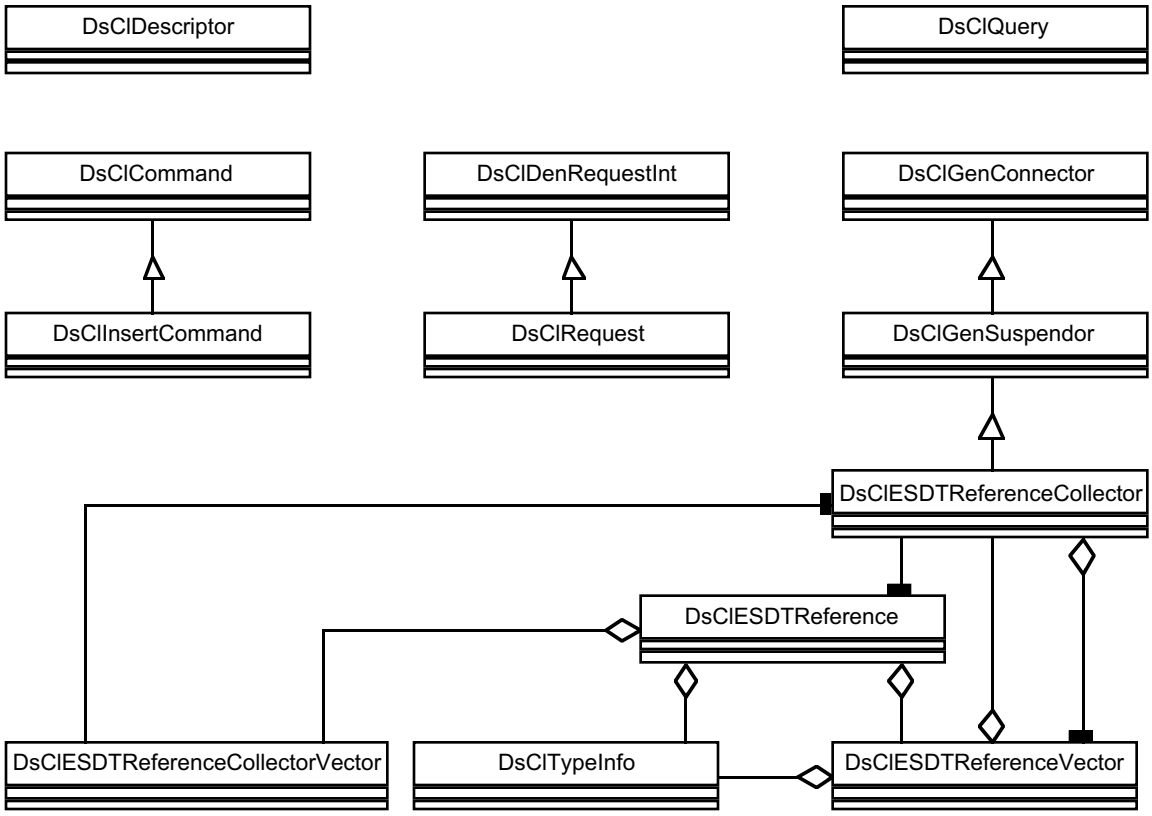

Figure 3.1-8. ECS Interface Object Model

3.1.4 Persistent Data

The function of persistent translator data management is performed by the classes, in Figure 3.1-9, in conjunction with the COTS (or shareware) database later described in Section 4.2.4. All database related functionality (opening, closing, querying etc.) is encapsulated in these classes. The data that is used for conversion of ItemID to Universal Resource is stored and managed by the ItemIDURConverter and PersistentStorageManager classes. The persistent data that is used for the CIP to ECS Database Mapping is managed by CIPECSDBMapper class. The CIPECSAttributeMapper class stores and manages the attribute mapping tables. As is implied, the data is usually stored in RDBMS and retrieved and updated by the different classes.

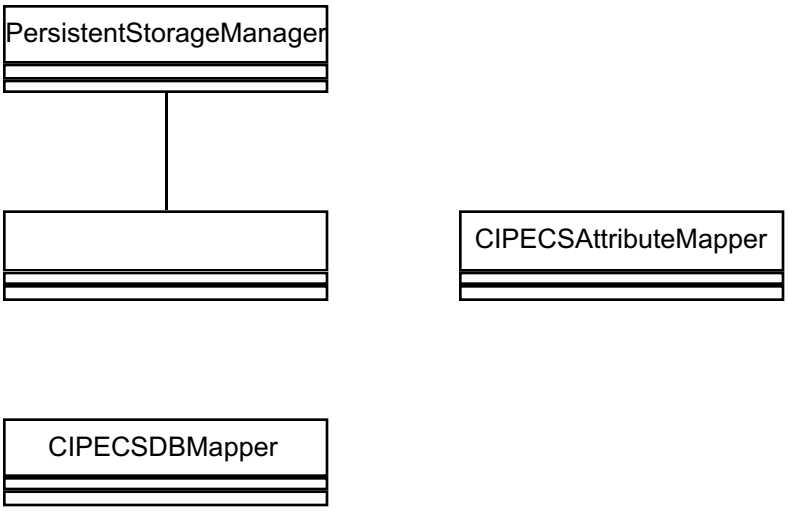

Figure 3.1-9. Persistent Data Object Model

3.1.5 Translator/Controller

The Translator object is a controller class that manages all the other objects including interactions with the other external processes. All the classes serve as helper classes to this controller class. The Translator object model of this class is shown in Figure 3.1-10. The CSearchContext is the primary controller class that is invoked by the catalog Translator and instantiates all the other objects as needed.

Figure 3.1-10. Translator Object Model

3.2 Translator vs. ECS Object Interaction Event Trace Diagrams

The following figures show the object level trace diagrams for the interaction among the High level Translator Objects and those in the ECS, specifically the Science Data Server.

The Search Request and the Present Requests for the CIP domain are mapped to the corresponding Inventory Search Request and the Inspect Request in the ECS domain.

3.2.1 Search Request/Response

Figure 3.2-1 shows the Object level interaction for the Search Request and the return of the Search Response, where only the Result set count and the status of the operation are the expected results. When the search is executed, the Collections to be targeted are already identified and provided as part of the query. The remaining search parameters and the qualifiers are entered by the user. Figure 3.3-2 shows the interaction of the controller class object in the translator with the objects of the Science Data Server in greater detail.

3.2.2 Present Request/Response

Figure 3.3-3 shows the object level interaction for the Present Request and the construction and return of the Present Response. The results that were created in the Search Request within the working collection are returned. Figure 3.3-4 shows the interaction of the controller class object in the translator with the objects of the Science Data Server in greater detail that help in the construction of the result set. The collector, a proxy for the working collections on the client side, i.e. the translator side, can be accessed in a number of ways including serially to build the result set. Since the collector exists at the translator, it manages all the communication and efficient network use and data storage for the granules that are requested by the users. The collector can be accessed by the DsClESDTReferenceCollector class.

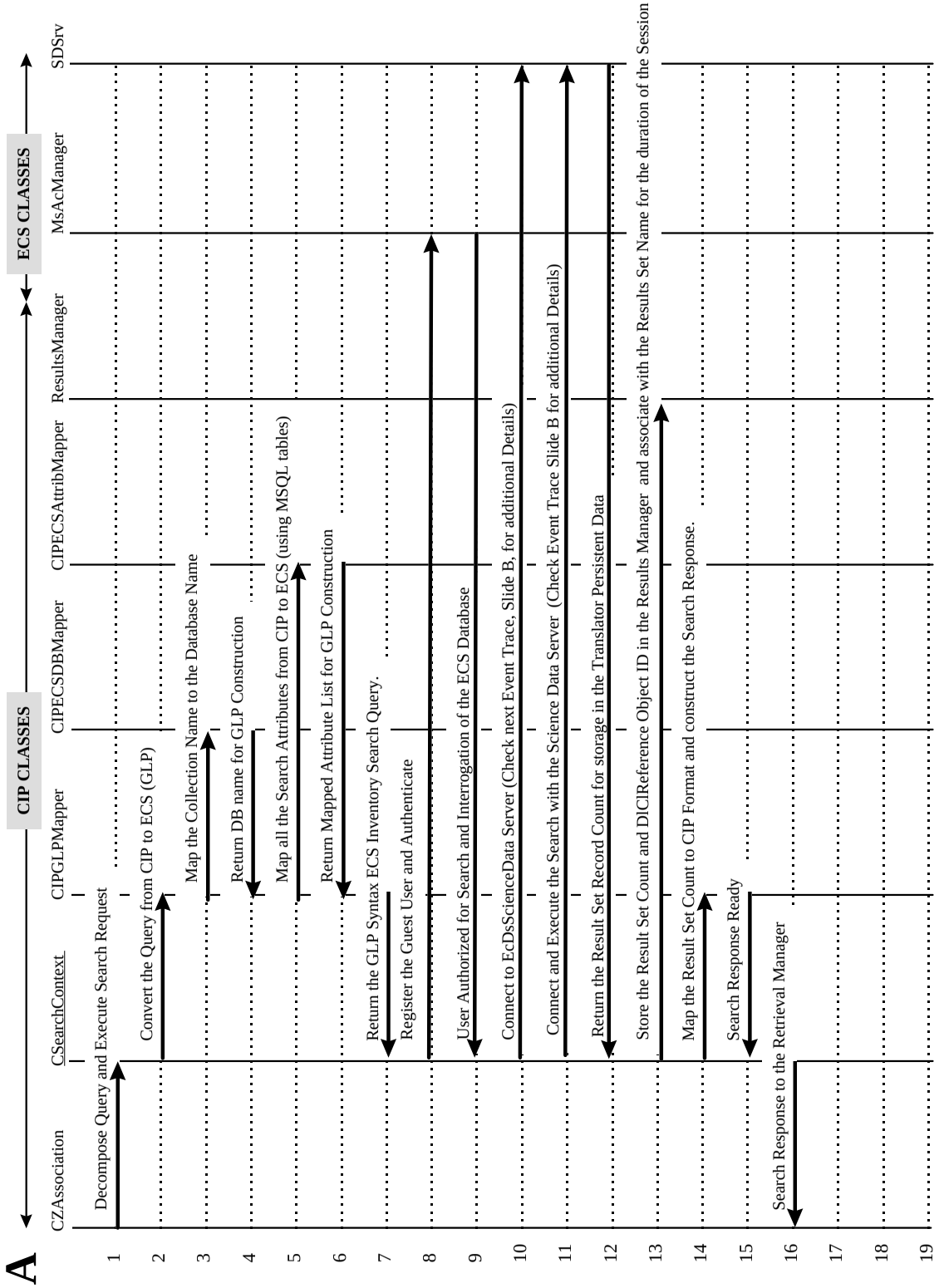

Figure 3.2-1. CIP-Domain: Search Request/Response Event Trace

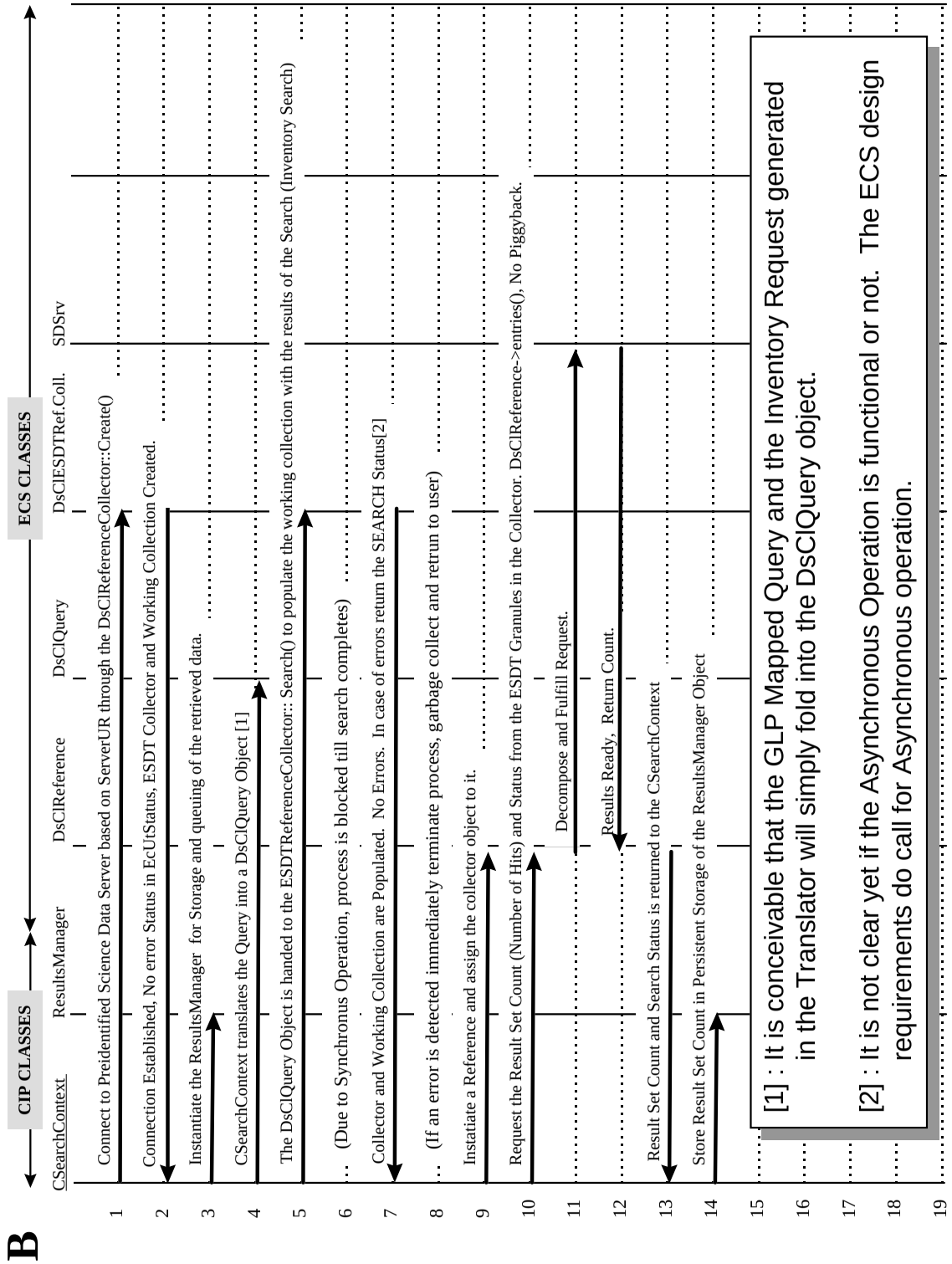

Figure 3.2-2. ECS-Domain: Inventory Search Request/Response Event Trace

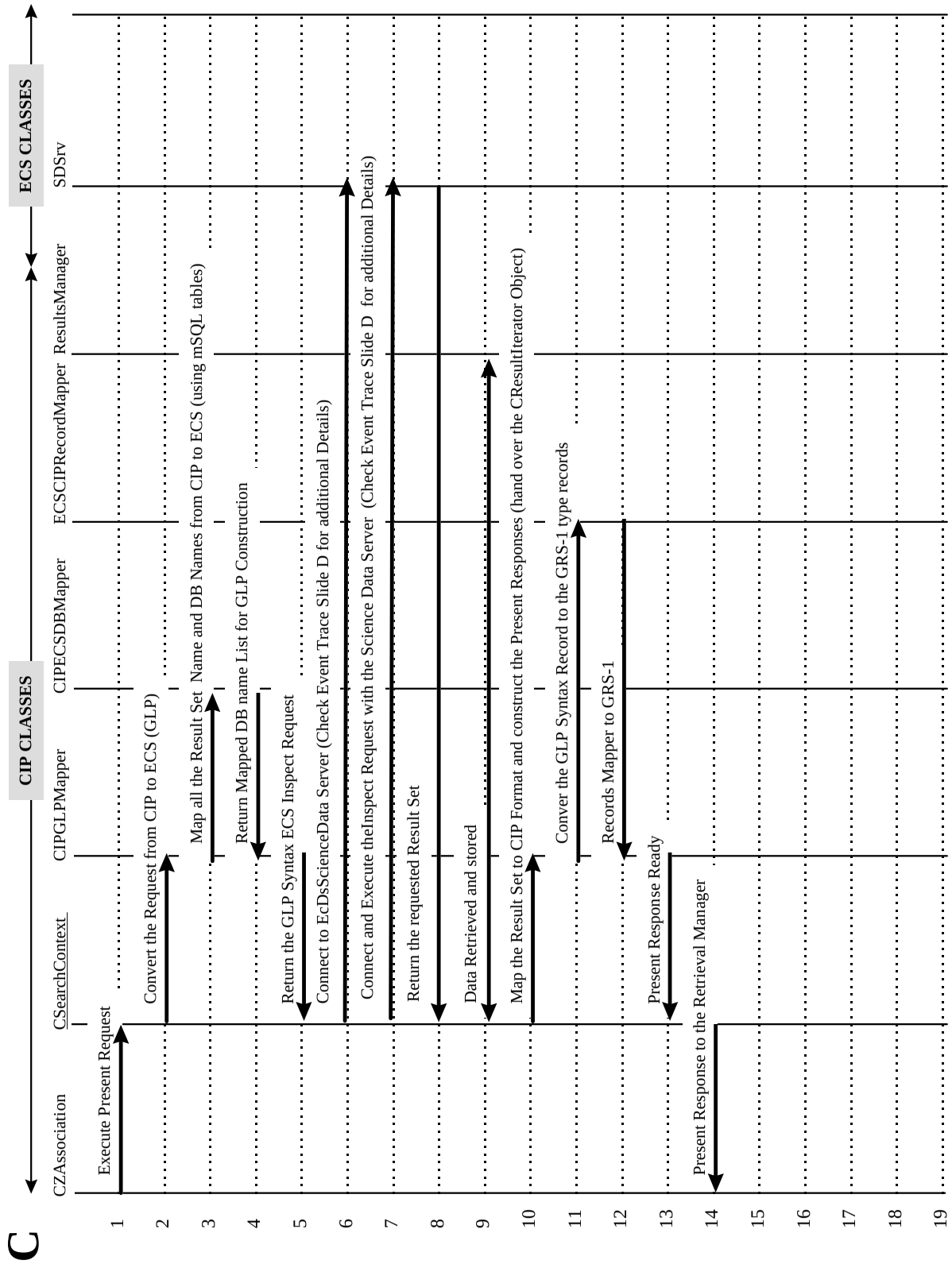

Figure 3.2-3. CIP-Domain: Present Request/Response Event Trace

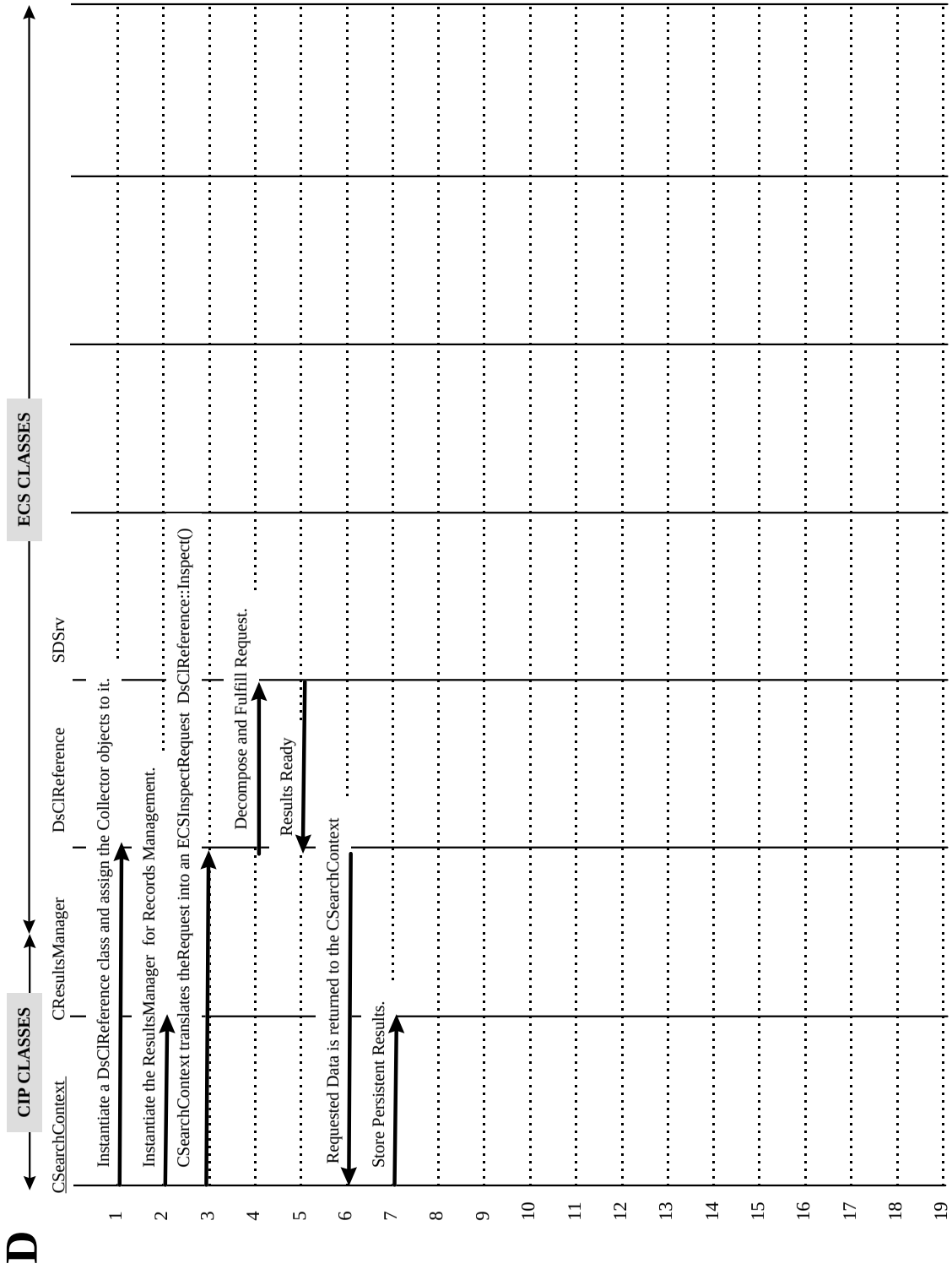

Figure 3.2-4. ECS-Domain: Inspect Request/Response Event Trace

This page intentionally left blank.

4. Re-usable/COTS Components

4.1 Re-usable/COTS Components

This section describes the options that are available for the re-usable and the commercial off the shelf (COTS) components that were evaluated for building the Catalog Translator. The candidate software packages are first described, following which the chosen software is identified. The candidate software packages are:

- Z39.50 Implementations
  - Isite
  - YAZ Toolkit
  - DBV-OSI II Toolkit
- CIP Release A Demonstrator Prototype
- ECS Data Subsystem API
- Database Management Systems
  - Sybase
  - Illustra
  - Mini SQL

4.2 Candidate Software Components

4.2.1 Z39.50 Implementations

Z39.50 is an information retrieval protocol which specifies formats and procedures concerning message exchange between a client and a server. It enables a client to request that a data server searches an archive and locates data meeting specified criteria and then to retrieve some or all of the identified data. It has been specified as a standard (ANSI/NISO Z39.50) and has been selected as the base protocol for CIP.

4.2.1.1 Isite

Isite is an integrated Internet publishing software package which is freely distributed by the Clearinghouse for Networked Information Discovery and Retrieval (CNIDR), through the National Science Foundation Cooperative Agreement NCR-9216963. Support is received from the Microelectronics Center of North Carolina (MCNC) who continues to develop and maintain *Isite*. *Isite* integrates database systems with other open Internet systems and protocols such as the World Wide Web, Gopher, electronic mail and, primarily, Z39.50. The primary protocol,

Z39.50 - Version 2, offers a variety of search and retrieval facilities suitable for complex database operations. *Isite* is developed using the C++ programming language and includes all Z39.50 communications applications as well as an HTTP to Z39.50 gateway and a complete text search system called *Isearch*. *Isite* supports a subset of the GEO Profile.

4.2.1.2 YAZ Toolkit

YAZ is a public domain programmers toolkit supporting the development of Z39.50 - Version 3 and SR protocol clients and servers. YAZ was developed using the C programming language and was designed for portability. YAZ has successfully been compiled on many UNIX® systems when provided an ANSI C or GNU compiler. YAZ readily supports the international standard SR protocol and enables communication using either conventional TCP/IP or an OSI stack.

4.2.1.3 DBV-OSI II Toolkit

The DBV-OSI II API is a Z39.50 - Version 3, service provider software package developed by Crossnet Systems Limited for the German DBV-OSI II library project. The API is an extremely versatile and adaptable package for anyone wishing to implement Z39.50 client-server applications. The API provides a comprehensive subset of the Z39.50 Version 3 protocol services and supports both TCP/IP and a full OSI association environment by utilizing the ISODE package. The DBV-OSI II package has been extended to include support for Explain, Access Control, Resource Control, Trigger Resource Control and the Access Control Prompt 1. Crossnet Systems Limited is also responsible for the continued development and maintenance of the DBV-OSI II API. This toolkit is freeware and is publicly available through FTP, and is a dependency of the CIP Release A Demonstrator (CIP-AD) discussed below. The software has been released in two versions -- Release 1.9.3 and Release 2.1.

4.2.2 CIP Release A Demonstrator (CIP-AD)

The CIP-AD is a software development effort performed by Logica UK Ltd. under contract to the Center for Earth Observation (CEO). The development effort was undertaken in order to provide a common distributed interface to Earth Observation (EO) catalogs. The development effort was performed using the C++ programming language and is based on the CIP-A specification for catalog interoperability, and uses the ANSI/NISO standard Z39.50 - Version 3 protocol for information retrieval, implemented with the DBV-OSI II Toolkit. The architecture of CIP-AD is described in the CIP Release A Demonstrator Architectural Design Document, LUK.503.EC.22426/AD001. The

4.2.3 ECS Data Sub-System

The ECS Release B (Version 2.0) Data Sub-System (DSS) consists of three different servers – The Science Data Server (SDSrv), The Data Distribution Server (DDist) and the Storage Management Server (STMGT). The DSS effectively hides the implementation of the Sub-system by providing the clients with a proxy interface to SDSrv. The interface provides a set of well defined classes that can be instantiated to Query, Search and Retrieve data objects, the

Product Descriptors, from the backend system. The proxy interface also assumes all the responsibility of the secure communications and the extraneous details of the distributed searches.

In addition to the DSS, the Data Dictionary Server (DDict) is also needed to access the Directory information that is required to pose qualified search queries. The Data Dictionary Server is accessed by automatic tools that are used to populated the persistent data and the Collection Data Base.

4.2.4 Relational Database Management Systems (RDBMS)

4.2.4.1 Sybase

The ECS systems Release A and Release B (Release 2.0) use the Sybase 11 as the underlying relational database engine, to implement and maintain the Data Dictionary Services, the Management Sub-System Servers and the Data Sub-System Servers. The Science Data Server uses the Sybase RDBMS

4.2.4.2 Illustra

Illustra 3.2 is an currently in use by the Java Earth Science Tool (JEST) and is the best known Object Oriented Database Management System (OODBMS) in the industry. However, it is only a consideration for the Catalog Translator due to its use in an existing ECS system.

4.2.4.3 Mini SQL (mSQL)

Mini SQL is a lightweight relational database engine designed to provide fast access to stored data with low memory requirements. As its name implies, it offers a subset of ANSI SQL as its query interface. Although it only supports a subset of SQL (no views, sub-queries etc.) all supported query functionality is in accordance with the ANSI SQL specification. The mSQL package includes the database engine, an interactive terminal "monitor" program, a database administrative program, a schema viewer and a C language API. The API and the database engine have been designed to work in a client-server environment over TCP/IP. The package is publicly available and is shareware to commercial companies (ECS has purchased a site license for the deployment of mSQL). The CIP-A Demonstrator uses mSQL as the underlying database engine for their implementation of the Collection and Explain Databases.

4.3 Chosen Software Components

The following components have been chosen to build the first phase of the prototype.

4.3.1 Development Environment

The following COTS packages will be used in the first phase of prototype development. These versions were the result of specific choices, as described in subsequent sections.

- Solaris 2.5

- DBV OSI II Z39.50 Toolkit, Version 2.1.0, Patch-Level 2
- C and C++ Programming Language
- GNU C++ Compiler
- MiniSQL Relational Database System, Version 2.0 b7.
- Apache Web Server, Version 1.2.6
- CIP Version 2.2 with INFEO Modifications.

4.3.2 CIP A Demonstrator

This is the core component which will provide much of the ICS architectural framework for purposes of interoperability. The Demonstrator will provide the front end WWW Client interface as well as much of the Retrieval Manager middleware component, which will act as the client to the CIP Catalog Translator. As the name implies, CIP-A Demonstrator is based on CIP Specification Release A, and it must be upgraded to CIP Specification Release B to interoperate with other ICS components. Catalog Translator component will be upgraded to CIP specification Release B, Version 2.2 with INFEO modifications. The purpose is to allow immediate interoperability with the INFEO Retrieval Managers. It may be upgraded to CIP-Specification Release B, Version 2.4 at a later point.

4.3.3 Mini SQL

The Demonstrator uses this package as the underlying database engine and we perceive this interface to remain unchanged. It will also be used in the CIP-ECS Catalog Translator and will allow low-cost re-use of the Translator. Currently the most current version of MSQL is the Version 2.2 b7; as newer version of the mSQL become available, they will be considered based upon evaluation of benefits to the translator prototype.

4.3.4 DBV OSI II Toolkit

The DBV OSI II, Version 2.1.0, Toolkit has been selected as the underlying Z39.50 implementation. This toolkit will be re-hosted to the target environment along with the CIP A Demonstrator. The DBV Toolkit was chosen as it supports Z39.50 Version 3. The CIP-A Demonstrator was developed with the Version 1.9.3 of DBV-OSI II and hence the CIP-ECS prototype also used the same version. The future needs of the prototype now requires upgrade to version 2.1. These upgrades must be done to the CIP Server Objects of the Catalog Translator. In addition, the Retrieval manager must be upgraded as well for international interoperability.

Future phases of prototype development will evaluate the YAZ Toolkit in more detail, since the YAZ product has a smaller memory footprint and it should interoperate at the protocol level with any system compliant with Z39.50 Version 3 Protocol.

4.3.5 ECS Data Sub-System

The ECS Catalog Translator will communicate with the ECS subsystem via the client proxy interface to the Science Data Server (SDSrv). The interface allows a client to access Searching and accessing facilities of the SDSrv.

The Drop 4 version of ECS software is currently being used in the development, assuming that the ECS Interfaces remain constant, newer version of the ECS systems can be easily integrated into the future release of the Catalog Translator.

This page intentionally left blank.

Appendix A. CIP/ECS Prototype Requirements

This appendix lists the requirements for the CIP/ECS Prototype.

A.1 Requirements

The following requirements are met by the CIP-ECS Catalog Translator preliminary design.

1. The prototype shall support product search requests against a target Provider Archive Collection (Single-Type Collection only) for ECS Data.
2. The prototype shall support the retrieval of browse images. The image URL will be communicated to the user for display.
3. Mapping of query attributes (metadata) shall be done in the Translator.
4. The Catalog Translator shall map the Item Descriptor ID to Universal Resource and vice versa.
5. The Catalog Translator shall map the ECS database names to CIP names and vice versa.
6. The Catalog Translator shall allow result sets to be created by name for the duration of the session. Queries against the result set are not supported in this version of the prototype.

This page intentionally left blank.

Appendix B. CIP/ECS Attribute Mapping

B.1 Introduction

This appendix maps the Attributes from the CIP domain to the ICS and ECS domains. The table B.2-1 is sorted by the CIP compound attributes followed by the CIP Element Name. The Goals is show the one-to-one mapping and the mapping of the compound attributes from one domain to the other. In ideal case, the ECS domain has a corresponding compound Attribute (defined as an ECS class) and the element types and names match from CIP to ECS domain. Since all attributes o not map in this manner, the table also provides a synopsis for solving the problem of compound attribute mapping.

B.2 CIP/ECS Attribute Mapping

Collection, Product and Granule Attribute Description Mappings (current as of November 1997)

Table B.2-1 Attribute Description Mappings

CIP Compound	CIP Element Name	CIP Prod /Coll	ECS Class	ECS Element Name	ECS Coll /Gran
	Albert's Conical Equal Area			SEE NOTES	
	Any	**C P**			
	Authoritative	***C***		SEE NOTES	
	Authoritative	***P***			
	Band Mode				
	Browse Compression				
	Browse Data				
	Browse Delivery Options				
	Browse Format				
	Browse Retrieval Options				
	Browse Service Options				
	Catalogue ID				
	Category Specific Info				
	Central Meridian Longitude			SEE NOTES	
	Central Meridian Scale factor			SEE NOTES	
	Collection Descriptor				
	Data Centre			SEE NOTES	
	Delivered Algorithm Package				
	Document Compression				
	Document Data				
	Document Delivery Options				
	Document Retrieval Options				
	Document Service Options				
	Document Variant				
	False Easting				
	False Northing				
	GridCoordinateSystemDefinition				
	Guide Collection Specific				
	Guide Descriptor				
	Included Collection				

CIP Compound	CIP Element Name	CIP Prod /Coll	ECS Class	ECS Element Name	ECS Coll /Gran
	Descriptors				
	Included Guide Descriptors				
	Included Item Descriptors				
	Included Product Descriptors				
	Investigator			Notes	
	Item Data				
	Item Delivery Method				
	Item Word Size				
	Landsat Number				
	Language				
	Locality				
	Locality Description				
	Locality Type				
	Map Projection			SEE NOTES	
	Map Projection Definition			SEE NOTES	
	Ordering Centre ID	***C P***			
	Pass Type				
	Polar Stereographic			SEE Notes	
	Postal Code	C	ContactAddress	PostalCode	C
	Processing Centre	***C P***		SEE Notes	
	Processing Type				
	Product Compression				
	Product Delivery Options				
	Product Descriptor				
	Product Order Options				
	Product Processing Option				
	Product Processing Options				
	Product Service Options				
	ProductType				
	Projection Origin Latitude				
	Publication				
	Publication Place				
	Purpose				
	QA Collection Statistics				
	Related Collections Descriptors				
	Related Guide Descriptors				
	Related Item Descriptor				
	Related Item Descriptors				
	Related Product Descriptors				
	Relation Description				
	Reprocessing				
	Scene selection Options				
	Science Review				
	Science Review Status				
	Sensor				
	Space Oblique Mercator			SEE Notes on MapProjection	
	Spatial Coverage				
	Spatial Reference				
	Standard Parallel			SEE Notes on Map Projection	
	Straight Vertical Pole Longitude			SEE Notes on Map Projection	
	Technical			SEE Notes onRole	
	Transverse Mercator			SEE Notes MapProjection	
	Universal Transverse Mercator			SEE Notes MapProjection	
	Use Constraints				
	User Descriptor				
	User ID				
	Vertical Co-ordinate System				
	WRSGRS Pass				
	WRSGRS Scene				

CIP Compound	CIP Element Name	CIP Prod /Coll	ECS Class	ECS Element Name	ECS Coll /Gran
Acquisition	Acquisition	P			
Acquisition	Acquisition Station	P			
Acquisition	Acquisition Status	P			
Address	Address	C			
Address	City	C	ContactAddress	City	C
Address	Country	C	ContactAddress	Country	C
Address	State	C	ContactAddress	StateProvience	C
Address	StreetAddress	C	ContactAddress	StreetAddress	C
Administrator	Administrator	C		SEE NOTES	
AlititudeSystem	Altitude Datum Name	C&P	AltitudeSystemDefinition	AltitudeDatumName	C
AltitudeSystem	Altitude Resolution	C&P	AltitudeSystemDefinition	AltitudeResolution	C
AltitudeSystem	Altitude System	C&P			
BoundingRectangle	**East Bounding Co-ordinate**	**C&P**	GranuleBoundingRectangle	East Bounding Co-ordinate	G
			CollectionBoundingRectangle	**CollectionEast Bounding Co-ordinate**	**C**
BoundingRectangle	**North Bounding Co-ordinate**	**C**	**CollectionBoundingRectangle**	**CollectionNorthBoundingCoordinate**	**C**
	North Bounding Co-ordinate	**P**	GranuleBoundingRectangle	NorthBoundingCoordinate	G
BoundingRectangle	**South Bounding Co-ordinate**	**C**	**CollectionBoundingRectangle**	**CollectionSouthBoundingCoordinate**	**C**
		P	GranuleBoundingRectangle	SouthBoundingCoordinate	G
BoundingRectangle	**West Bounding Co-ordinate**	C	**CollectionBoundingRectangle**	**CollectionWestBoundingCoordinate**	**C**
	West Bounding Co-ordinate	P	GranuleBoundingRectangle	WestBoundingCoordinate	G
Browse	Browse	C&P		See Notes	
Browse	Browse ID	C&P	SummaryProduct	SummaryProductID	C
BrowseData	Browse Pointer		GranuleSummaryProduct	GranuleSummaryPointer	G
			SummaryProduct	SummaryProductPointer	C
BrowseDeliveryOptions	ItemByteSize	G	GranuleSummaryProduct	GranuleSummarySize	G
DocumentDeliveryOptions		C	DocumentDetails	DocumentSize	C
ProductDeliveryOptions		C&P	**ECSDataGranule**	**ECSDataGranuleSize**	**G**
Circle	Circle	C&P	GranuleCircle		G
			CollectionCircle		C
Circle	***Radius Value***	***C***	CollectionCircle	CollectionRadiusValue	C
		P	GranuleCircle	RadiusValue	G
CollDescriptor	Access constraints	C		Access Constraints	
Collection Type	**Collection Hierarchy Category**	**C**			
Collection Type	**Collection Hierarchy Position**	**C**			
Collection Type	Collection Type	C			
CollectionDescriptor	External Publication Citation	C	SingleTypeCollection	CitationForExternalPublication	C
CollectionDescriptor	**Item Descriptor ID**	**C**			
ProductDescriptor	**Item Descriptor ID**	**P**			
CollectionDescriptor	**Item Descriptor Name**	**C**	**CollectionDescriptionClass**	**ShortName**	**C**
					P
CollectionDescriptor	**Progress**	**C**	**SingleTypeCollection**	**CollectionState**	**C**

CIP Compound	CIP Element Name	CIP Prod /Coll	ECS Class	ECS Element Name	ECS Coll /Gran
CollectionDescriptor	***Project Name***	***C***	Campaign	CampaignShortName;CampaignLongName	C&G
CollectionDescriptor	Version ID	C	**ECSCollection**	**VersionID**	**C**
GuideDescriptor		C	**Document**	**DocumentVersion**	**C**
DeliveredAlgorithmPackage		C	AlgorithmPackage	AlgorithmPackageVersion	C
CollectionType	**Collection Category**	**C**	ECSCollection	CollectionType SEE NOTES	C
Compound	Processing	C&P			
Compound	Processing Level	C			
Compound	Product Collection Specific	C			
Contact	Contact	C	Contact		C
Contact	Email Address	C	Email	ElectronicMailAddress	C
Contact	Fax Number	C	Telephone	TelephoneNumberType TelephoneNumver	C
Contact	Hours Of Service	C	Contact	HoursofService	C
Contact	Instructions	C	Contact	ContactInstructions	C
Contact	Job Position	C	PersonOrganizationXref	ContactJobPosition	C
Contact	**Organisation Name**	**C**	**ContactOrganization**	**OrganizationLongName**	**C**
Contact	**Person Name**	**C**	**ContactPerson**	**FirstName**	**C**
		P	ContactPerson	MiddleName	C
			ContactPerson	**LastName**	**C**
Contact	Role	C	CollectionContactXref	Role	C
Contact	Telephone Number	C	Telephone	Telephone Number TelephoneNumberType	C
Data Centre	***Data Centre Name***	***C***		SEE NOTES	
DataOriginator	**Mission ID**	**C**			
DataOriginator	***Mission ID***	***P***			
DataOriginator	**Originator**	**C**			
	Originator	***P***			
DataOriginator	**Source Name**	**C**	PlatformModel & Instrument	PlatformShortName&InstrumentShortName	C
DataOriginator	***Source Name***	***P***			
Delivered AlgoPackage	***Algorithm Package Name***	***C***	AlgorithmPackage	Algorithm Package Name	C
DepthSystem	Depth Datum Name	C&P	DepthSystemDefinition	DepthDatumName	C
DepthSystem	Depth Resolution	C&P	DepthSystemDefinition	DepthResolution	C
DepthSystem	Depth System	C&P			
Discipline	Discipline	C			
Discipline	**Discipline Keywords**	**C**	**ECSKeyword**	**ECSDisciplineKeyword**	**C**
DliveredAlgorithmPackage	SW Package Pointer	C	AlgorithmPackage	AlgorithmPackageID	C
DocumentData (Guide and Ref Paper)	Document Pointer	C	Document	DocumentPointer	C
DocumentDeliveryOptions	Document Format	C	DocumentDetails	DocumentFormat	C
DocumentVariant(Guide)	Document Name	C	Document	DocumentTitle	C
G Polygon Exclusion G Ring	G Polygon Exclusion G Ring	C&P	CollectionExclusionGRing		C
			GranuleExclusionGRing		G
G Polygon Outer G Ring	G Polygon Outer G Ring	C&P	CollectionOuterGRing		C
			GranuleOuterGRing		G
GeographicSystem	Geographic System	C&P	GeographicCoordinateSystem		C
GeographicSystem	Latitude Resolution	C&P	GeographicCoordinateSystem	LatitudeResolution (SEE NOTES)	C
GeographicSystem	Longitude Resolution	C&P	GeographicCoordinateSystem	LongitudeResolution (SEE NOTES)	C
Gpolygon	G Polygon	C&P	CollectionGPolygon		C
			GranuleGPolygon		G
Grid Co-ordinate System	Grid Co-ordinate System	C&P	GridCoordinateSystem		C
Grid Co-ordinate System	Grid Co-ordinate System Name	C&P	Grid Co-ordinate System	GridCoordinateSystemName	C
Guide	*Document Type*	C	Guide	GuideType	
Guide	Guide	C	Guide		C
GuideDescriptor					
CollectionDescriptor	**Creation Date**	**C**			

CIP Compound	CIP Element Name	CIP Prod /Coll	ECS Class	ECS Element Name	ECS Coll /Gran
DeliveredAlgoPackage Guide Descriptor				**Document Created**	**C**
GuideDescriptor		**C**	**Document**		
DeliveredAlgorithmPackage		**C**			
CollDescriptor	**Abstract**	**C**			
Browse		**P**	**ECSCollection**	**CollectionDescript**	**C**
			GranuleSummaryProduct	GranuleSummaryDescripti	G
GuideDescriptor ReferencePaper	Author			SEE NOTES	
hBoundingRectangle	Bounding Rectangle	C	CollectionBoundingRectangle		
		P	GranuleBoundingRectangle		
Horizontal Co-ordinate System	Horizontal Co-ordinate System	C&P	HorizontalCoordinateSystem		C
Keywords	***General Keyword***	***C***	ThematicKeywordClass	ThemeKeyword	C
Keywords	Keywords	C			
Keywords	***Spatial Keyword***	***C***	SpatialKeywordClass	SpatialKeyword	C
Keywords	***Temporal Keyword***	***C***	TemporalKeywordClass	TemporalKeyword	C
MapProjection	Map Projection Name	C&P	MapProjection	MapProjectionName	
PlanarSystem	Planar System	C&P			
Point	***Latitude***	***C***	CollectionCircle	CollectionCenterLatitude	C
	Latitude	***P***	GranuleCircle	CircleCenterLatitude	G
			CollectionPoint	CollectionPointLatitude	C
			CollectionExclusionGRingPoint	ExclusionGRingPointLatitude	C
			GranuleEXclusionGRingPoint		G
			Collection OuterGRingPoint	OuterGRingPointLatitude	C
			GranuleOuterGRingPoint		G
			GranulePoint	PointLatitude	G
Point	***Longitude***	***C***	CollectionCircle	CollectionCenterLongitude	C
		P	GranuleCircle	CircleCenterLongitude	G
			CollectionPoint	CollectionPointLongitude	C
			CollectionExclusionGRingPoint	ExclusionGRingPointongitude	C
			GranuleEXclusionGRingPoint		
			Collection OuterGRingPoint	OuterGRingPointLongitude	G
			GranuleOuterGRingPoint		C
			GranulePoint	PointLongitude	G
Point	Point	C&P			
ProcessingLevel	Processing Level Description	C	ProcessingLevel	ProcessingLevleDescription	C
ProcessingLevel	Processing Level ID	C	ProcessingLevel	Processing Level ID	C
Product Collection Specific	**Archiving Centre Id**	**C**		**Archive Center**	**C**
ProductDescriptor	***Archiving Centre Id***	***P***			
ProductCollectionSpecific	StorageMedium	C	CollectionStorageMedium	CollectionStorageMedium	C
ProductCollectionSpecific	Temporal Resolution	C	**ECSCollection**	**TemporalRangeType**	**C**
ProductDeliveryOptions	Product Format	C&P	CSDT Description	PrimaryCSDT	C
ProductDeliveryOptions	***Product Medium***	***C***			
	Product Medium	***P***			
ProductDescriptor	Data Originator	**P**			
ProductCollectionSpecific		C			
ProductOrderOptions	***Group ID***	***C***			
		P			
Publication	***Publication Date***	***C***	ReferencePasper	DAteOfReferencePaperPublication	C
QACollStats	QA Quality Explanation	C	QAFlags	SEE Notes	G
QACollStats	QA Rating	C	QAFlags		G
QAProductStatistics	***QA Missing Data Percentage***	***P***	QA Stats	QA Missing Data Percentage	G
QAProductStatistics	***QA Out of Bounds Data Percentage***	***P***	QA Stats	QA Out of Bounds Data Percentage	G

CIP Compound	CIP Element Name	CIP Prod /Coll	ECS Class	ECS Element Name	ECS Coll /Gran
QAProductStatistics	***QA Percent Cloud Cover***	***P***	QA Stats	QAPercentCloudCover	G
QAProductStatistics	***QA Percent Interpolated Data***	***P***	QA Stats	QA Percent Interpolated Data	G
QAProductStatistics	QA Product Statistics	P			
ReferencePaper	Abstract Pointer	C	ReferencePaper	Abstract Pointer	C
ReferencePaper	Citation	C	ReferencePaper	ReferencePaperID	C
ReferencePaper	Reference paper				
ReferencePaper	**Reference Type**	**C**	ReferencePaper	ReferencePaperType	C
Reprocessing	Reprocessing Actual	P	ECSDataGranule	ReprocessingActual	G
Reprocessing	Reprocessing Planned	P	ECSDataGranule	ReprocessingPlanned	G
Review	***Future Review Date***	***C***			
Review	Review	C			
Revision	Revision	C			
Revision	**Revision Date**	**C**	**ECSCollection**	**RevisionDate**	**C**
		G	**Document**	**Document Updated**	**C**
Revision (Guide, Collection, DelAlgoPackage)	Update Frequency	C	**SingleTypeCollection**	**Maintenance&UpdateFrequency**	**C**
ScienceReview	***Science Review Date***	***C***	ECSCollection	LastReviewDate	C
Sensor	Sensor Mode	C&P	CollectionInstrumentOperationMode	OperationMode	C
			GranuleInstrumentXref	OperationMode	G
Sensor	**Sensor Name**	C	Sensor	SensorShortName;SensorLongName	C&G
Sensor	***Sensor Name***	***P***			
Temporal Coverage	Temporal Coverage	C&P			
Temporal Period	Temporal Period	C			
Temporal Range	Temporal Range	C			
TemporalPeriod	Period Cycle	C&P	RegularPeriodic	PeriodCycleDurationValue	C
TemporalPeriod	Period Cycle Unit	C&P	RegularPeriodic	PeriodCycleDurationUnit	C
TemporalPeriod	Period Duration	C&P	RegularPeriodic	PeriodDurationValue	C
TemporalPeriod	Period Duration Unit	C&P	RegularPeriodic	PeriodDurationUnit	C
TemporalPeriod	Period Name	C&P	RegualrPeriodic	PeriodName	C
TemporalRange	***End Date***	***C***	**CollectionDateTime**	**CollectionEndingDate**	**C**
			CollectionDateTime	**CollectionEndingTime**	**C**
TemporalRange	**End Date**	**P**	**ECSDataGranule**	**RangeEndingDate**	**G**
			ECSDataGranule	**RangeEndingTime**	**G**
TemporalRange	***Start Date***	***C***	**CollectionDateTime**	**CollectionBeginningDate**	**C**
	Start Date	**P**	**ECS DataGranule**	**CollectionBeginningTime**	**G**
				RangeBeginingDate & Range BeginingTime	
Term	Term				
Term	**Term Keyword**	**C**	**ECSKeyword**	**ECSTermKeyword**	**C**
Term	**Variable Keyword**	**C**	**ECSVariable**	**ECSVariableKeyword**	**C**
Topic	Topic	C			
Topic	**Topic Keyword**	**C**	ECSKeyword	ECSTopicKeyword	C
Universal Transverse Mercator	UTM Zone Number	C	ZoneIdentifierClass	ZoneIdentifier SEE Notes	C
User	Network Address				
WRSGRSSCENE	***Frame***	***C***			
		P			
WRSGRSSCENE	***Track***	***C***			
		P			

LEGEND

C : Collection

P : Product

G : Granule

CIP bolded attributes	: Mandatory Use Attributes
CIP bolded italic attributes	: Optional Use Attributes
ECS bolded attributes	: Intermediate Mandatory Use Attributes

NOTE: The rows where Compound Name and Element Names match (identical), are to be ignored.

This page intentionally left blank.

Appendix C. Universal Resource to Item Descriptor ID Mapping

This appendix identifies the Universal Resource (UR) and the Item Descriptor Identifier (Item ID) and describes the mapping from the UR to the Item ID.

C.1 Introduction

The UR provides a mechanism to access ECS object instances through logical references. It allows users to access object that can exist anywhere in the system. For this functionality, a UR encapsulates identifiers specific to particular object classes, it accommodates nesting of identifiers and supports multiple formats. The UR structure also facilitates uniqueness across DAAC's.

Similarly, the Item Descriptor Identifier are needed in the CIP domain to locate or to refer to any item descriptor (i.e. for collection, product or guide) within and across Retrieval Managers. The structure of the Item ID uniquely identifies each item owned by the Retrieval Manager.

The translator will access a table, that is maintained by external automatic components, that will map from ECS UR to CIP Item ID's and from Item ID's to ECS UR's. The two way mapping will allow the user to order data and for unique association of the Item Id to the ECS UR.

C.2 Universal Resource (UR) Definition

The following definition of UR shows the structure of a UR and demonstrates the concept with an example. Finally, there is an externalized version of the UR that is generated when a UR structure is flattened for communication across network.

The granule UR is a string of up to 89 characters, and contains information about the specific Science Data Server where the granule resides and a unique granule ID.

UR:<serverID>:<GranuleID>

<serverID> :== refers to the identification of the logical server that contains the inventory metadata. The ,serverID> has a maximum of 50 bytes, including colon separators and is broken up as follows. The structure is:

[length]:<ESDT_UR>:UR:[length]:<Server_UR>:[length]:<ServiceToken>

where

length :== A 2-byte value indicating the number of bytes in the string that appears between the following two colons.

ESDT_UR :== a maximum of 10-bytes indicating that the UR is associated with an ESDT.

Server_UR :== A maximum of 15-bytes indicating that the UR is associated with a Science Data Server.

ServiceToken :== A 13-byte combination of DCE cell name and the Science Data Server Name containing the Data Granule.

<GranuleID> :== has a maximum length of 35-bytes, including the colon separators, and is defined as:

[length]:<BaseType>:<SubType>:<databaseID>

where

length :== A 2-byte value indicating the number of bytes in the string that appears between the following two colons.

BaseType :== a 2 -byte value that gives the type of the data, e.g.

BR	Browse
SC	Science
QA	QA
DP	Delivered Algorithm Package (DAP)
AP	Algorithm Package
AC	Science Software Archive Package Components (SSAPC)
PH	Production History
LM	LIMITED
CL	COLLECTION
GD	Guide
AD	Algorithm Description
PR	Processing Report
PP	Production Plan
RP	Reference Paper

SubType :== A 5-12 byte combination ‘ShortName.VersionID’, and is human readable. VersionID is fixed 3 byte long, zero padded.

DatabaseID :== A 1-16 byte number that uniquely distinguishes between granules in the same Science Data Server.

NOTE: The ShortName, VersionID and the DatabaseID are the only parts of the UR that are entered into the Science Data Server Inventory Tables as part of the meta-data for the data granule. Also, the structure of the UR may change and so the translator cannot parse the UR to extract any information and must treat the UR as a character stream that identifies the granules and no more.

EXAMPLE:

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[RBD:DSSDSRV]:20:SC:AST_LIBT.001:1308

The Underlined potion is the GranuleID and where

ESDT_UR =	DsShESDTUR
Server_UR =	DsShSciServerUR
ServiceToken =	[RBD:DSSDSRV]
Basetype =	SC
SubType =	AST_LIBT.001
databaseID =	1308

C.3 Item Descriptor Identifier (Item ID) Definition

The Item ID's are needed in the CIP domain to locate or to refer to any item descriptor (i.e. for collection, product or guide) within and across Retrieval Managers. The structure of the Item ID uniquely identifies each item owned by the Retrieval Manager. The two versions of CIP Specification define the Item Descriptor Id in two different format. Since it is anticipated that sometime in future, the Catalog Translator will be upgraded to CIP Specification, Release 2.4, both the definitions are given here. CIP Release 2.2 defines the Item ID as a alpha numeric string of 3 alphabets followed by 8 digits. The Release 2.3 defines the Item ID as a 128 byte string.

C.3.1 CIP Specification Version 2.2 and CIP Specification Version 2.2 (INFEO)

CIP Specification Version 2.2 in Appendix D section D.3, defines the Item ID as a alphanumeric string of the type 'aaa_ddddddd' where 'aaa' denotes a alphabet string and the 'ddddd…' denotes the ASN.1 GeneralString portion of the string.

where

aaa: is an indicator of the type of the descriptor

'CID' = Collection Descriptor

'PID' = Product Descriptor

'GID' = Guide Descriptor

dddd… : an ASN.1 GeneralString of EXACTLY 8 characters which is unique within the scope of all items owned of this type by a SINGLE Retrieval Manager.

C.3.1.1 EXAMPLE OF Item ID:

PID_MS010108

This may denote a Product Descriptor of **M**odi**S** data captured on 01/01/2008.

C.3.2 CIP Specification Version 2.4 (based upon interim version 2.3)

CIP Specification Version 2.4 in Appendix E.3, defines the Item ID in different formats for the collection_descriptor_id, product_descriptor_id. They are both more detailed than the ECS UR and should not pose any problem with mapping.

collection_decsriptor_id = "z39.50s://" + host + [":" + port] + "/" + [collection_type_prefix | persistent_result_set_prefix] + [u_chars_max128 | origin_timestamp]

product_descriptor_id = "PID" + u_chars_max128

C.4 Mapping or UR to Item ID

The mapping of the Universal Resource to the Item Descriptor ID must be performed in different manners for the CIP Specification Version 2.2 Item ID and for the CIP Specification Version 2.4. Both the methods are presented below. It is anticipated that in future upgrades of the Catalog Translator, it will be made conformant to the CIP Specification Version 2.4 and hence both the mapping methods are defined here.

C.4.1 CIP Specification Version 2.2 and CIP Specification Version 2.2 (INFEO)

A mapping table is created in RDBMS that persistently stores the 12-Byte Item ID and the corresponding UR in a table that is indexed both on Item ID and the UR should be adequate in supporting any mapping requirements of the Catalog and the OHS Translators. This table will be populated by outside automatic tools that will manage the data. In the following figure shows that if an Item ID (A) is entered searched in the mapping tables, you would get the corresponding UR (A') and a UR (B') is entered the corresponding Item ID (B) can be found.

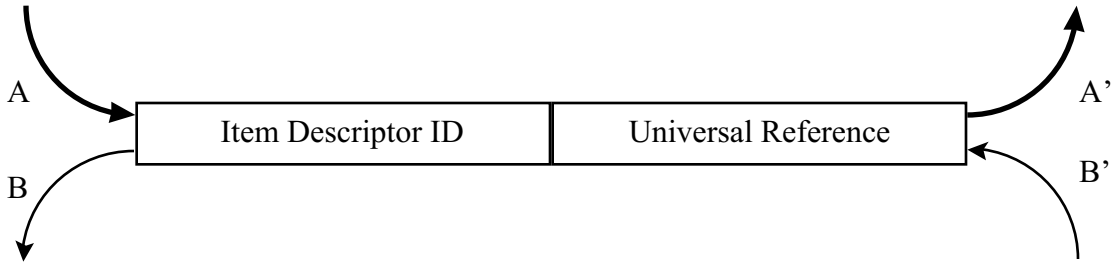

Figure C.4-1. Mapping table row to map the UR to Item ID (for Version 2.2)

C.4.2 MAPPING UR to Item ID (of Version 2.4)

The mapping process is substantially simpler when the system is updated to Release 2.4. The entire UR can be easily folded into the u_chars_max128 area of the collection_descriptor_id and the product_descriptor_id. Even after this the field would have 39 bytes left (128 minus 89) to stuff other information. The UR information can be extracted simply by parsing and locating the beginning of the u_chars_max128 field and then extracting all the information until the UR is complete. There is NO need for a separate marker for end of the string.

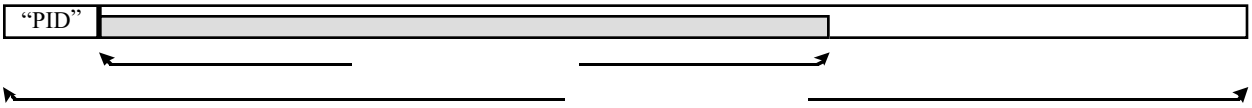

Figure C.4-2. Mapping the UR to Item ID (Version 2.4)

This page intentionally left blank.

Abbreviations and Acronyms

API	Application Programming Interface
Catalog Translator	CIP/ECS Catalog Translator
CEO	Center for Earth Observation (European Commission)
CIP	Catalog Interoperability Protocol
CIP-AD	CIP Release A Demonstrator
COTS	Commercial Off The Shelf
ECS	EOSDIS Core System
EOSDIS	Earth Observation System Data Information System
ICS	Interoperable Catalog System
JEST	Java Earth Science Tool
OHS	Order Handling System
RM	Retrieval Manager
RMA	Retrieval Manager Administrator
RPN	Reverse Polish Notation
SDD	System Design Document
WWW	World Wide Web

This page intentionally left blank.