



Digital Business Ecosystem

WP36: Composed Service Accounting Architecture Definition

D36.4: A set of Accounting Building Blocks for Automatically Composed Services



Information Society
Technologies

Project funded by the European Community
under the "Information Society Technology"
Programme

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable N°: D36.4
Due date: 06/2006
Delivery Date: 02/2007

Short Description:

This document describes a set of software building blocks that deliver accounting functionality to the DBE. The primary objectives are to develop open and adaptable models for: mediation, rating, correlation and aggregation, and billing for composed services.

Partners owning: WIT

Partners contributed: WIT

Made available to: Public

Versioning			
Version	Date	Author, Organization	Description
0.1	22/02/2007	CHRISTOPHER FOLEY - WIT	Initial Document Creation

Quality check

Internal Reviewer:

Accounting Software Building Blocks

Table of Contents:

1.EXECUTIVE SUMMARY.....	5
2.INTRODUCTION.....	6
3.METERING.....	7
3.1METERING FILTER.....	7
3.1.1Description	7
3.1.2Functional Implementation.....	8
3.2METERING SENSOR.....	9
3.2.1Description	9
3.2.2Functional Implementation.....	9
3.3DBE METERING SERVICE.....	10

3.3.1Description	10
3.3.2Functional Implementation.....	10
3.4DBE METERING USAGE DATA	11
3.4.1Description	11
3.4.2Functional Implementation.....	11
4.MEDIATION.....	12
4.1DBE IPDR.....	13
4.1.1Description.....	13
4.1.2Functional Implementation.....	13
4.2DBE IPDR RECORDER.....	14
4.2.1Description	14
4.2.2Functional Implementation.....	14
4.3DBE IPDR TRANSMITTER.....	16
4.3.1Description.....	16
4.3.2Functional Implementation.....	16
4.4DBE IPDR STORE.....	17
4.4.1Description.....	17
4.4.2Functional Implementation.....	17
5.RATING.....	18
5.1DBE RATING SERVICE.....	19
5.1.1Description.....	19
5.1.2Functional Implementation.....	19
5.2CREST RATING SERVICE.....	20
5.2.1Description.....	20
5.2.2Functional Implementation.....	20
5.3CREST RATING ENGINE.....	21
5.3.1Description.....	21
5.3.2Functional Implementation.....	21
5.4CREST PERSISTENCE.....	22
5.4.1Description.....	22
5.4.2Functional Implementation.....	22
5.5CREST WORKBOOKENGINE.....	23
5.5.1Description.....	23
5.5.2Functional Implementation.....	23
6.ACCESSING SOURCE CODE.....	25

List of Figures:

FIGURE1: NETWORK SERVICE ACCOUNTING SYSTEM.....	6
FIGURE 2: METERING REFERENCE MODEL	7
FIGURE 3: DBE METERING FILTER CLASS DIAGRAM.....	8
FIGURE 4: DBE METERING SENSOR CLASS DIAGRAM.....	9
FIGURE 5: DBE METERING SERVICE CLASS DIAGRAM.....	10
FIGURE 6: DBE USAGE DATA CLASS DIAGRAM.....	11
FIGURE 7: IPDR NMD-U REFERENCE MODEL.....	12
FIGURE 8: DBE IPDR SCHEMA.....	13
FIGURE 9: DBE IPDR RECORDER CLASS DIAGRAM.....	14
FIGURE 10: IPDR HELPER CLASS DIAGRAM.....	15
FIGURE 11: DBE TRANSMITTER CLASS DIAGRAM.....	16
FIGURE 12: DBE STORE CLASS DIAGRAM.....	17
FIGURE 13: RATING SERVICE.....	18

1. Executive Summary

The aim of this document is to describe the software components which are delivered as part of Work package 36 . The document will give a general overview of accounting architecture and describes the design and functionality of the software components developed in each layer of the architecture. It will also specify the technologies and standards adopted.

2. Introduction

An objective of WP36 is to define an accounting architecture to support the OSS accounting requirements of providers delivering next generation composed services. The components developed to achieve this follow the standard network service accounting system as shown in Figure 1.

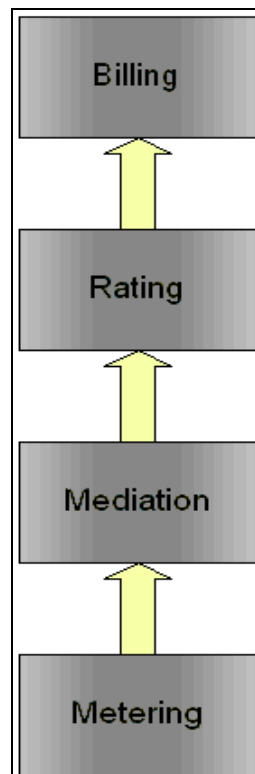


Figure1: Network Service Accounting System

Following research performed in Deliverable 36.1¹, the NMD-U(Network Data Management – Usage) reference model has been adopted as the basis for much of the DBE accounting architecture. A metering layer has been developed to record service usage and expose this to the next layer, Mediation. Also, a rating and charging solution will be provided in the current phase of the project.

The following sections will discuss the design, specification and functionality of the accounting software components for each layer of the accounting system.

3. Metering

Metering is provided as a DBE core service. It is a generic solution which will enable the SME to record usage of their services deployed in the DBE for charging or other measurement purposes. Based on the web service metering reference model discussed in W36.1, the metering architecture takes advantage of the filter functionality provided by the ServENT² where all messages passing through the ServENT can be interpreted by a customised filter. A metering filter acts as an interceptor of service requests and responses, extracting suitable logging information and exporting this metered data to a metering service. After the handler is finished extracting logging information, the engine passes the message to the next filter in the chain.

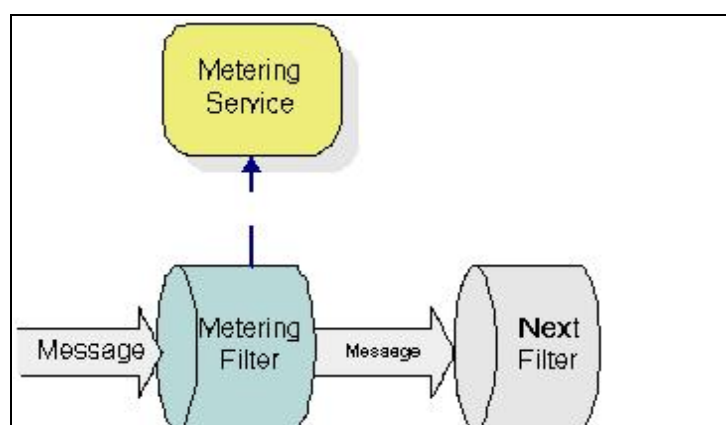


Figure 2: Metering Reference Model

The metered usage data is represented using a generic DBE data model. This data is persisted in the DBE Distributed Storage System (DSS) or the local file system where it is made available for mediation.

3.1 Metering Filter

3.1.1 Description

A metering filter can be included in a DBE service filter chain as part of the deployment of the service into the DBE. The Metering filter is a standard ServENT filter and implements the ServENT ServiceFilter interface. The method and method parameters of the DBE service to be metered are specified as part of the deployment of the service and are contained as in the deployment.properties file for the service. The Metering filter intercepts requests and responses and passes them directly to the DBE Servent Sensor where they are converted to DBE usage data.

3.1.2 Functional Implementation

The Metering filter is implemented in the DBE through the DBEMeteringFilter class which implements the ServiceFilter interface. Through the ServiceFilter interface, the Metering Filter accepts invocation requests and responses from the DBE ServENT.

Technologies used:

- DBE ServENT.

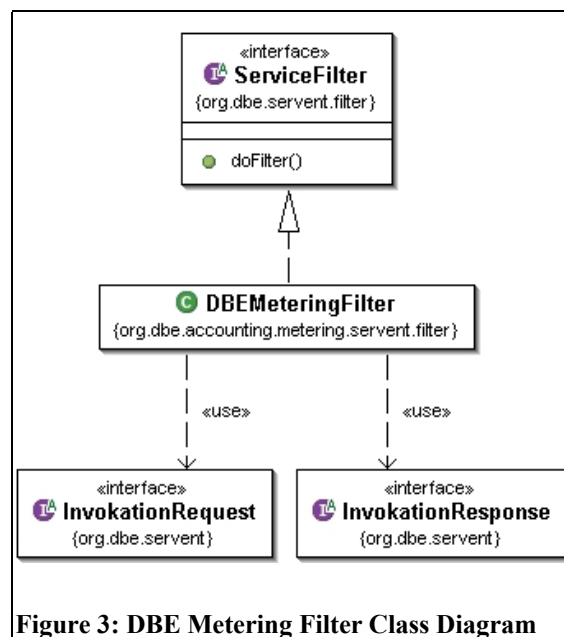


Figure 3: DBE Metering Filter Class Diagram

3.2 Metering Sensor.

3.2.1 Description

The DBE Metering Sensor accepts requests and responses from a metering filter and converts them to standard DBE Usage data. The Usage data can then be passed to the Metering Service for persistence.

3.2.2 Functional Implementation

Technologies used in the implementation of the Sensor are:

- DBE ServENT.
- Eclipse Modeling Framework

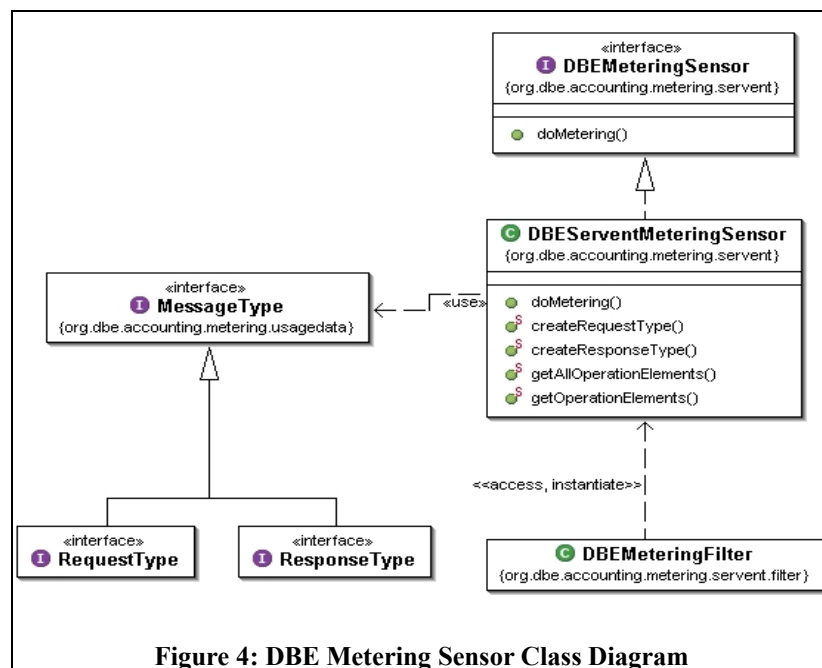


Figure 4: DBE Metering Sensor Class Diagram

3.3 DBE Metering Service

3.3.1 Description

The Metering Service completes metering by persisting and presenting usage data for mediation. The data can be persisted either in the DSS or on a local file system. The Metering service is deployed as a core service and is configured to persist data periodically in timestamped files. The Usage data received by the metering service is serialized into XMI using Resource functionality provided by the Eclipse Modeling Framework.

3.3.2 Functional Implementation

Technologies used in the implementation of the Sensor are:

- DBE ServENT.
- Eclipse Modeling Framework
- XMI (persisting of Data)

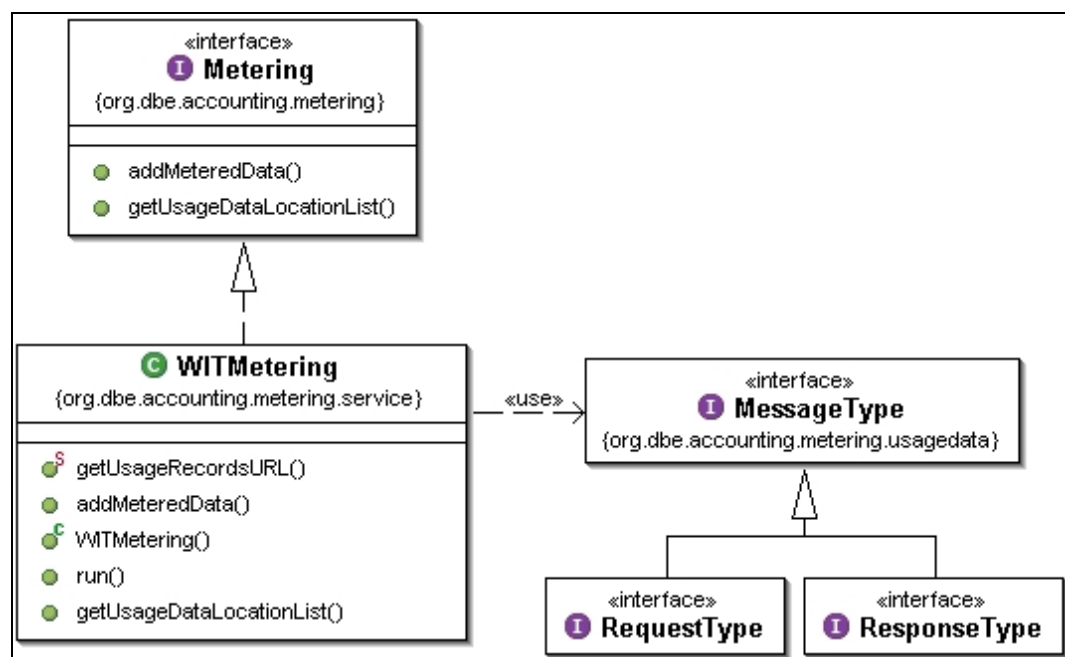


Figure 5: DBE Metering Service Class Diagram

3.4 DBE Metering Usage Data

3.4.1 Description

The Usage Data Model is a generic design to represent DBE service usage. It was modeled and created using the Eclipse Modeling Framework³ which provides code generation and serialization/deserialization features. The aim of this approach is to provide a shared common view of the underlying data.

3.4.2 Functional Implementation

As shown in the class diagram in figure 6, the Usage Data Model is contained in the implementation of `ServiceUsageType` which contains `RequestTypes` and `ResponseTypes`. `RequestType` and `ResponseType` extend `MessageType` which defines the required attributes to represent usage data for the DBE. Operation element types contain data specific to the operation being metered.

Technologies used in the implementation of the Sensor are:

- Eclipse Modeling Framework(Modeling and Code Generation)

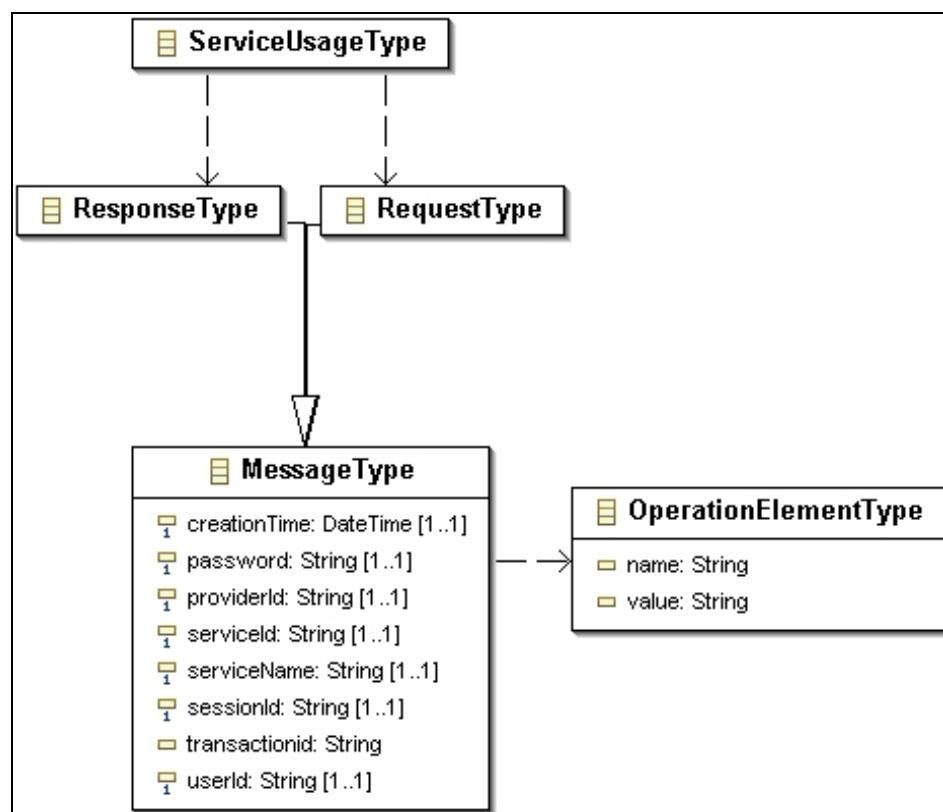


Figure 6: DBE Usage Data Class Diagram

4. Mediation

As already stated, mediation complies with IPDR.org NDM-U⁴ which provides a general reference model and defines the actors and interfaces involved in the transfer of the usage data. This is shown in the NDM-U reference model diagram and is described in greater detail in DBE deliverable 36.1.

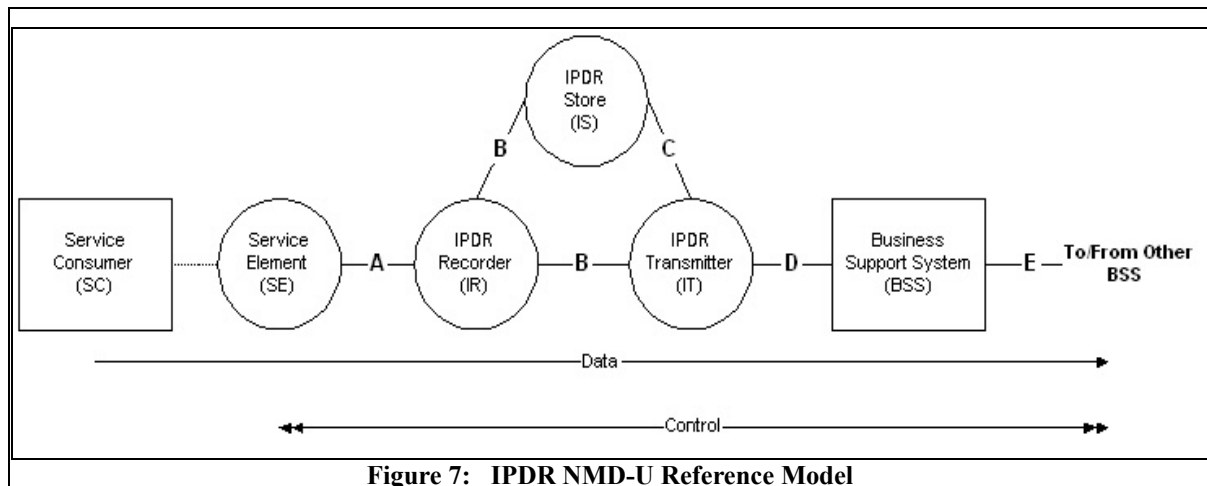


Figure 7: IPDR NDM-U Reference Model

Using the IPDR reference model, a flexible DBE IPDR record format has been developed that satisfies the needs of the DBE mediation. The elements involved in this model have been developed as DBE services and can be used by SMEs as the building blocks for a DBE mediation solution.

4.1 DBE IPDR

4.1.1 Description

The DBE IPDR Type is an extension of the general IPDRType Schema v3.6 provided in IPDR.org NDM-U. Java types representing the model were created using Eclipse EMF code generation facility and are used throughout the mediation components.

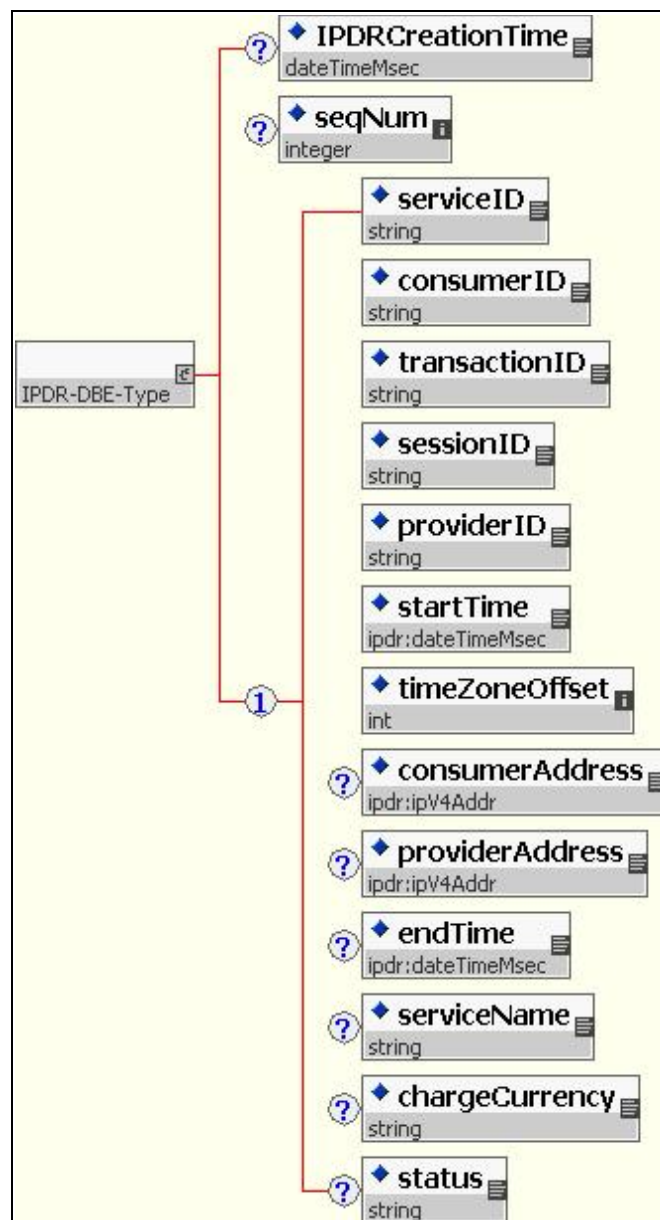


Figure 8: DBE IPDR Schema

4.1.2 Functional Implementation

A set of IPDR implementation classes generated from the schema in figure 8. The implementation classes produced are modified to implement the Serializable interface as this is required for all parameters used in DBE services.

4.2 DBE IPDR Recorder

4.2.1 Description

The DBE IPDR Recorder retrieves DBE Usage Data records from a specified location. It then converts these into IPDRDocs forwards these to IPDR Store and/or an IPDR Transmitter. Multiple service meters can be connected to a single IPDR Recorder.

4.2.2 Functional Implementation

The implementation of the IPDR Recorder is contained in the RecorderImpl class which implements IPDRRecorder_A_Interface. The RecorderImpl accesses the Usage Data records using a DataReader.

A service specific schema can be loaded into the Recorder which extends the generic DBE schema. A set of EMF resources is generated from the loaded schema and these resources are used to create the IPDRDoc.

Technologies used in the implementation of the Sensor are:

- DBE Servent
- Eclipse Modeling Framework

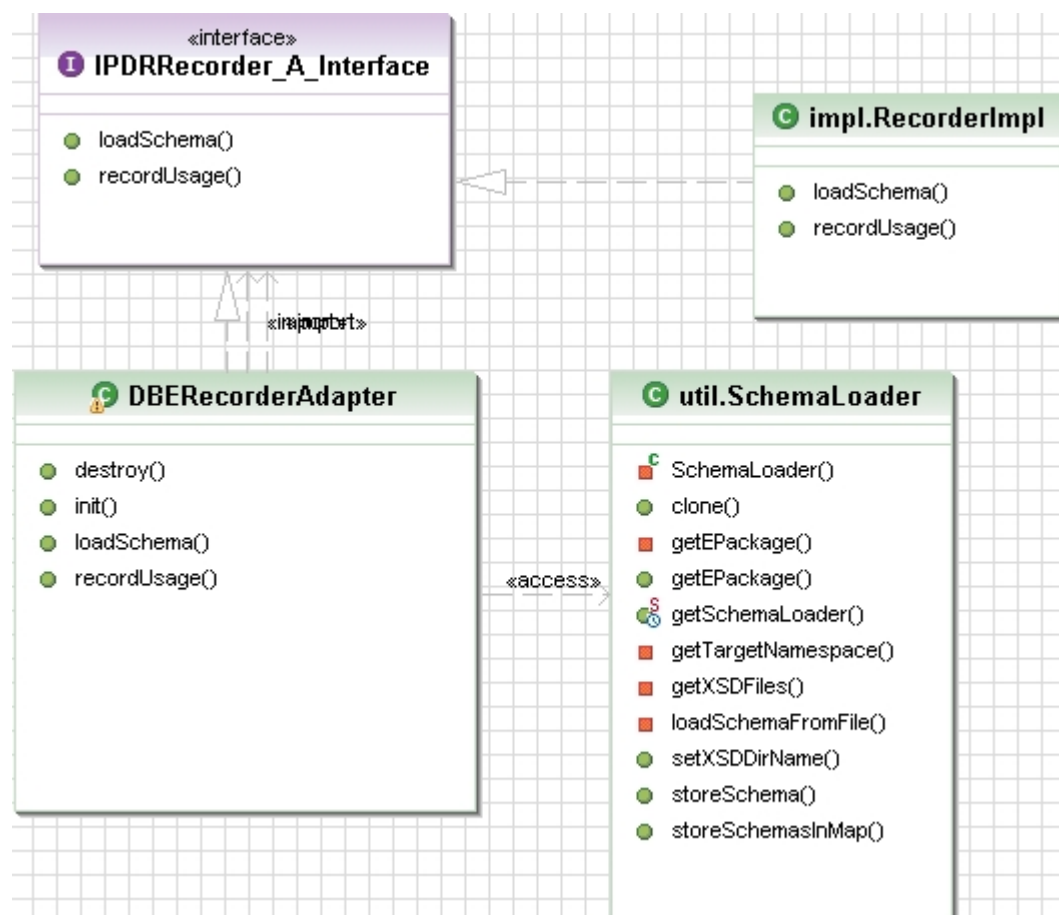


Figure 9: DBE IPDR Recorder Class Diagram

IPDRDocs are created using the `GenericIPDRHelper` class which can interrogate DBE usage data and use the available EMF resource sets.

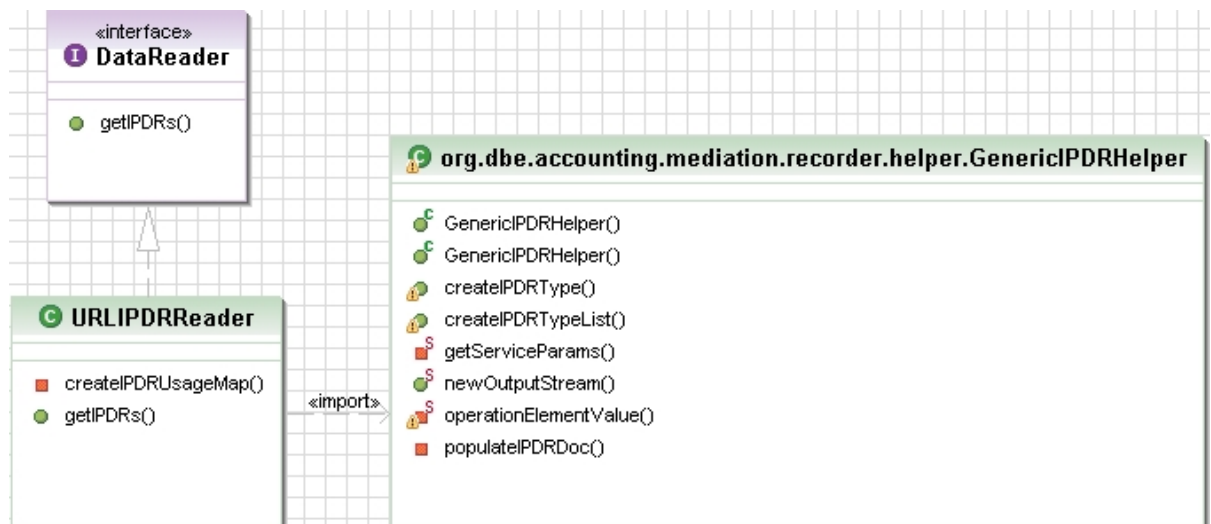


Figure 10: IPDR Helper Class Diagram

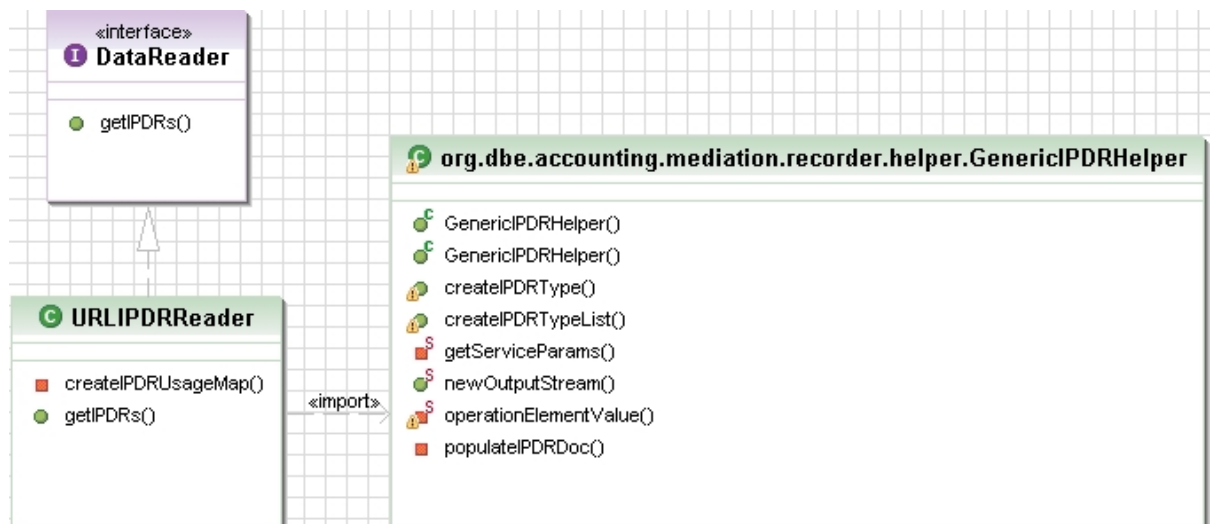


Figure 10: IPDR Helper Class Diagram

4.4 DBE IPDR Store

4.4.1 Description

The DBE IPDR Store is responsible for persisting IPDR Documents in the system.

4.4.2 Functional Implementation

The DBE IPDR Store is implemented through the DBEStoreImpl class that implements IPDR_B_Interface. The Transmitter accepts IPDRDocs from an IPDR Recorder and persists them.

Technologies used in the implementation of the Sensor are:

- DBE Servent
- Eclipse Modeling Framework(EMF)

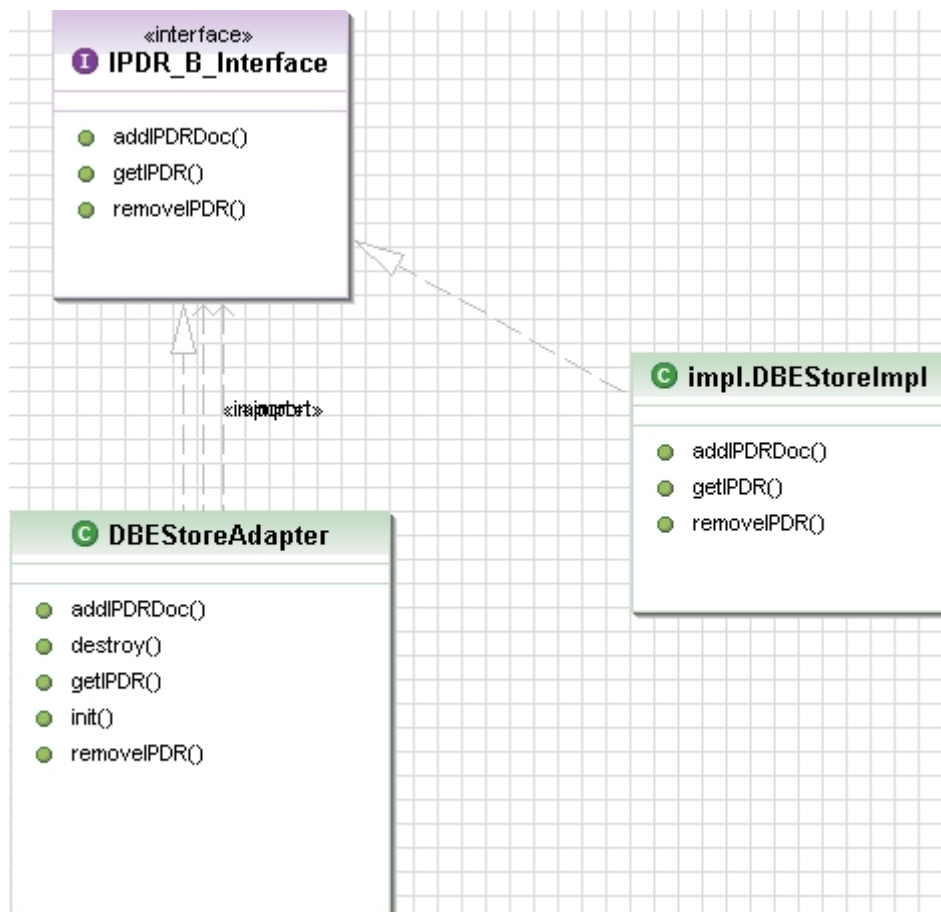


Figure 12: DBE Store Class Diagram

5. Rating

Rating is provided as a DBE service. It is a generic solution which allows SMEs to apply charging information to IPDR Documents which have been received from the Mediation layer. The charging algorithms are realized as OpenOffice spreadsheets. It calculates the charge and appends it to the IPDR document. This is then stored in a DB. These IPDR Documents, complete with the calculated charge, can then be accessed by a Billing system.

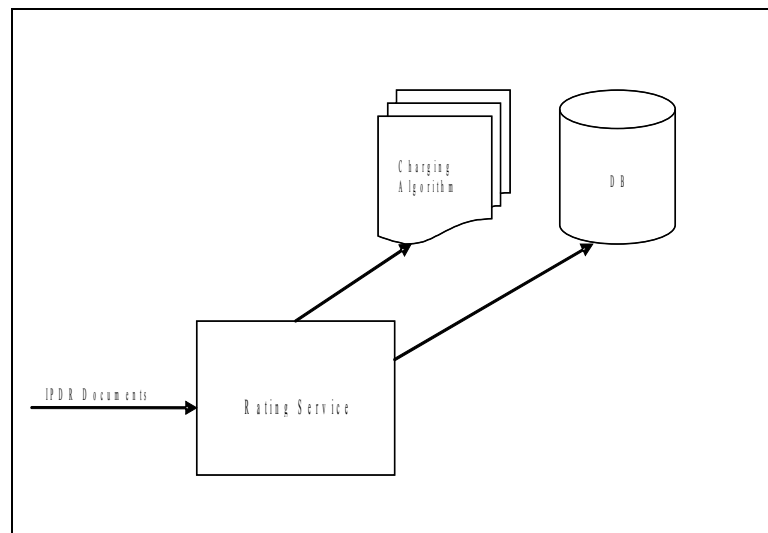


Figure 13: Rating Service

The rating engine has been called CREST - a Component-orientated Rating Engine for all Service Types. A rating engine that primarily utilises the IPDR record format and spreadsheet technology to rate service usage instances and produce rated IPDR documents. However, the CREST has been designed with maximum flexibility in mind to ensure ease of extension to other usage record formats. The CREST also incorporates features such as persistence and queuing, each of which are again easily extended to various providers other than those provided as defaults.

5.1 DBE Rating Service

5.1.1 Description

The Rating Service is the DBE representation of the rating functionality. It is the complete rating functionality(CREST) wrapped in a DBE service. It encompasses the charging and the storage of the IPDR Documents.

5.1.2 Functional Implementation

Technologies used in this implementation are:

- DBE ServENT

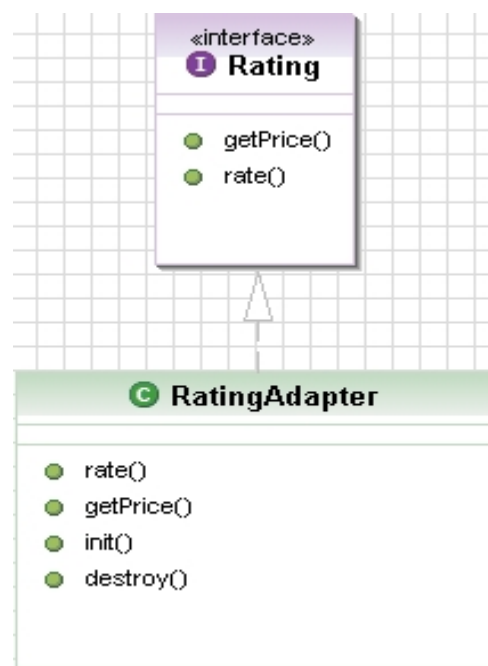


Figure 13: Class Diagram of package org.dbe.accounting.rating

5.2 CREST Rating Service

5.2.1 Description

This is the core CREST Rating Service.

5.2.2 Functional Implementation

The CRatingService class provides a rate() method which performs the rating of the incoming IPDR documents. It loads the relevant charging algorithms by using CAlgorithmFactory and passes this to the CRatingEngine in a separate thread.

CRatingService exposes a getPrice() method which performs the rating but doesn't store any of the charged IPDR Documents. It a query mechanism which can be used by another service without persisting any information.

The rating service also triggers the persistence of the rated and unrated versions of the IPDR Documents.

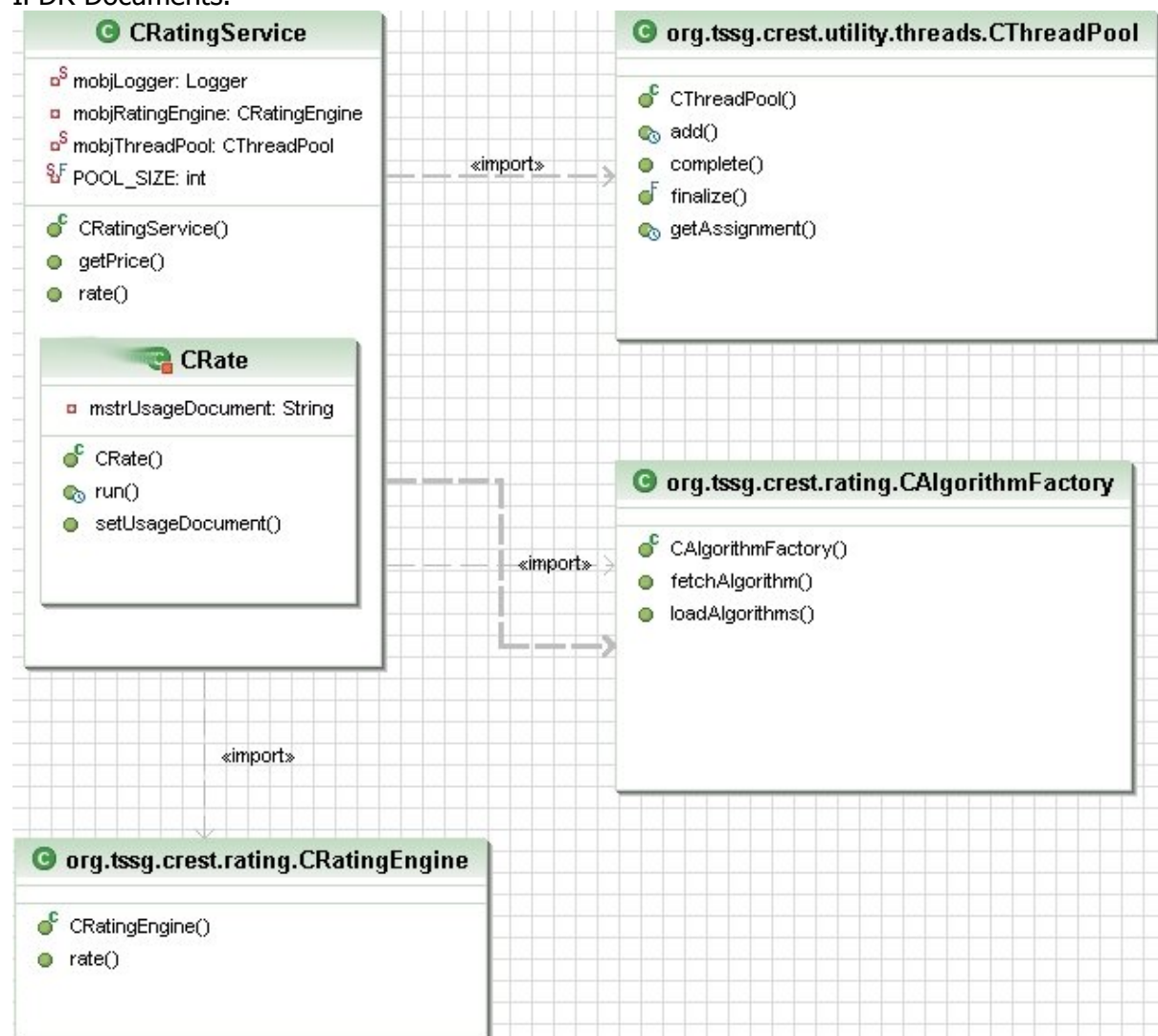


Figure 14: Class Diagram of package org.tssg.crest.rating.service

5.3 CREST Rating Engine

5.3.1 Description

The Rating Engine loads the rating algorithms and stores them in memory. It then performs the rating on the IPDR Documents.

5.3.2 Functional Implementation

Each charging algorithm is represented by the CAlgorithm class. The CAlgorithmFactory provides a facility to load the algorithms from disk into memory.

The CRatingEngine class performs the rating using the relevant algorithm and returns an IPDR Document complete with the charging information.

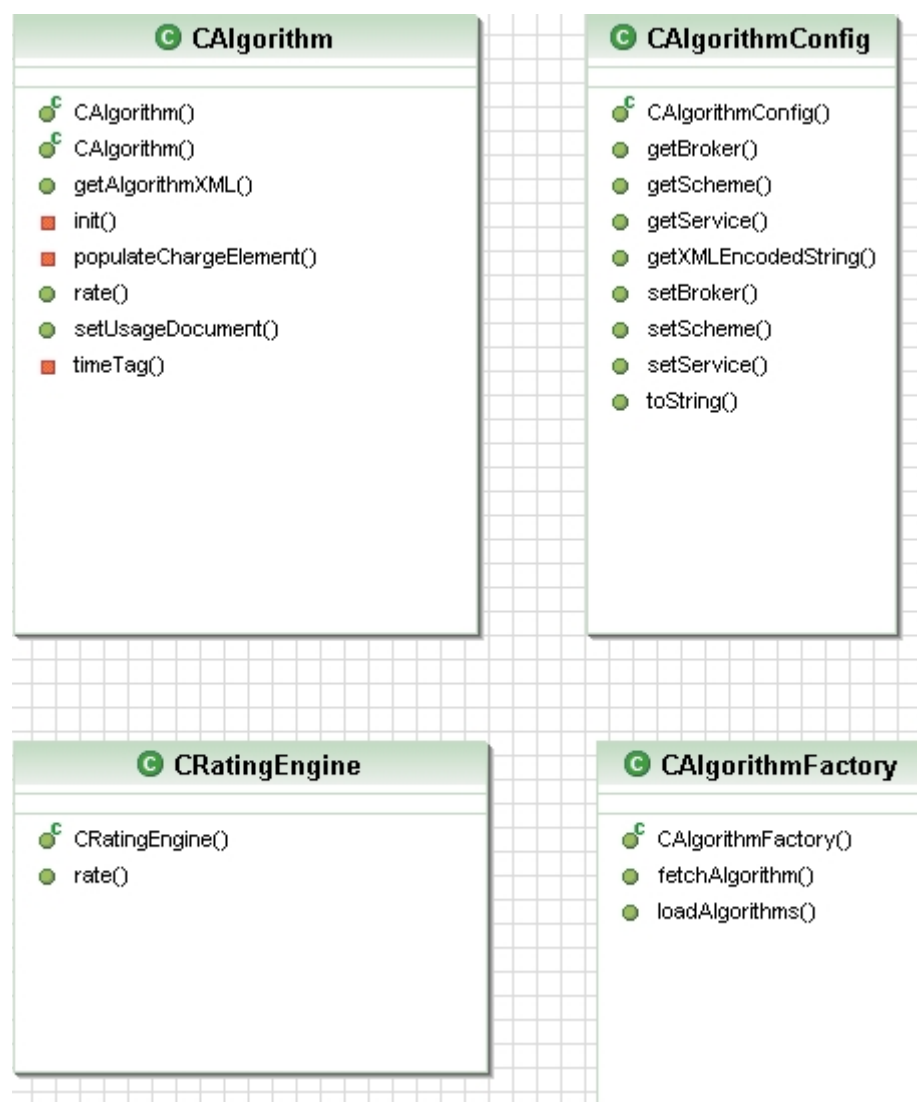


Figure 15: Class Diagram of package org.tssg.crest.rating

5.4 CREST Persistence

5.4.1 Description

The persistence package within CREST provides functionality for storage and querying of data. The DB which is used is eXist (i.e. An opensource XML DB).

5.4.2 Functional Implementation

The CPersist class exposes an insert() method which stores the document in the correct collection within the eXist DB. CQuery provides a query mechanism for the eXist DB where the IPDR Documents are stored.

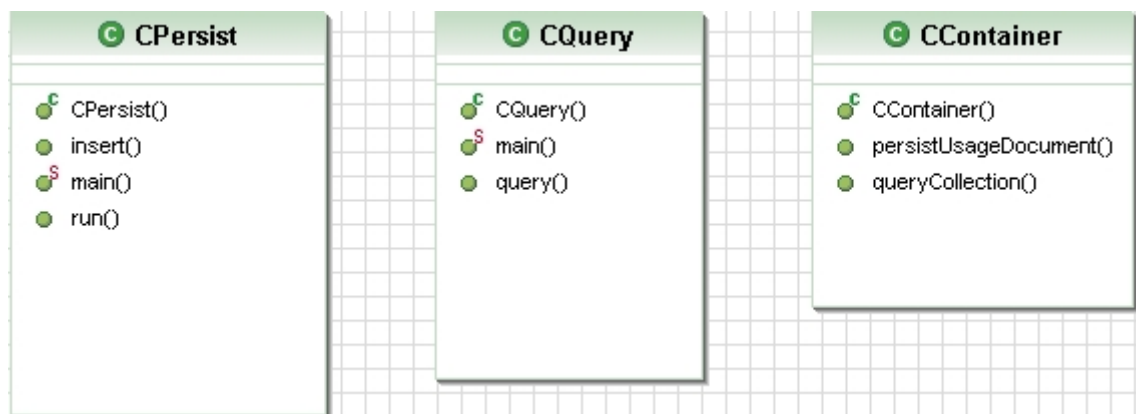


Figure 16: Class Diagram of package org.tssg.crest.persistence.exist

5.5 CREST WorkbookEngine

5.5.1 Description

The workbook engine package is responsible for interfacing with the algorithms which are Open Office workbooks.

5.5.2 Functional Implementation

The ISpreadsheetApp and IWorkbook interfaces provide methods which allow interaction with the OpenOffice workbooks. It provides functionality like getting and setting values of workbook cells, loading and saving of workbooks.

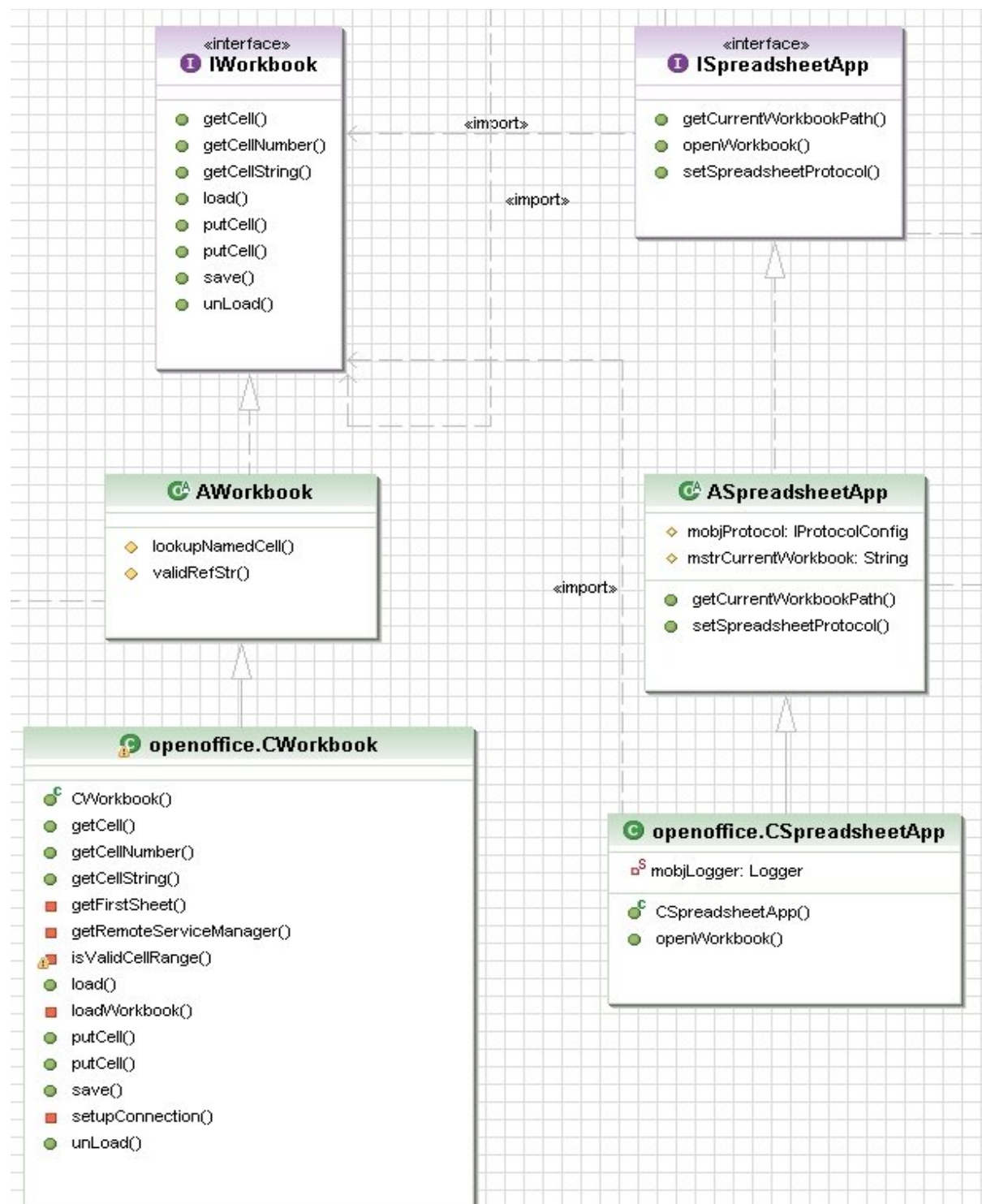


Figure 16: Class Diagram of package org.tssg.crest.workbook

6. Accessing Source Code

The source code for the components described in this document is currently contained on CVS (openacc.cvs.sourceforge.net). The modules relating to the accounting components are shown in Figure 17.

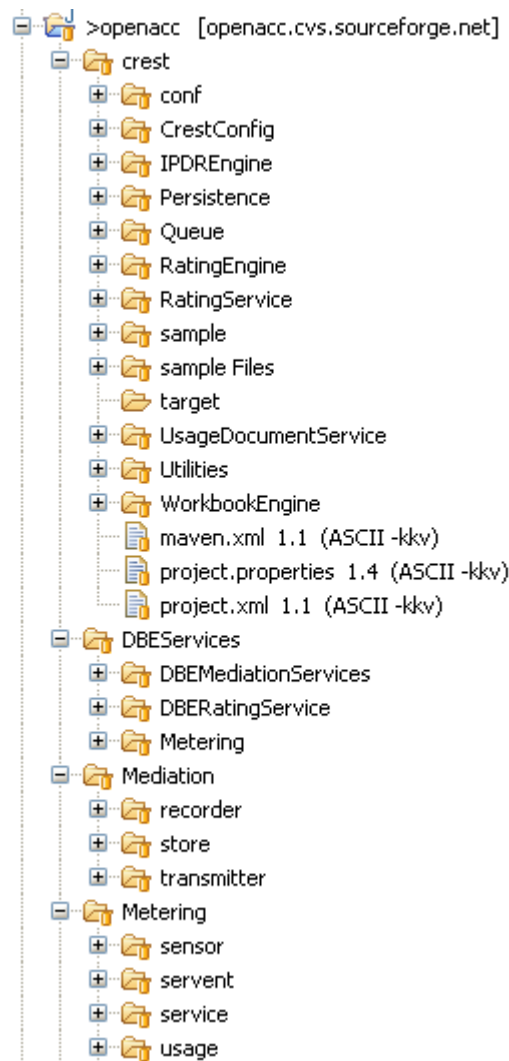


Figure 17: Snapshot of openacc.cvs.sourceforge.net structure

The 'crest' , 'Mediation' and 'Metering' folders contain the building blocks for an accounting system. The 'DBEServices' folder contain the DBE versions of the services. The 'Metering' folder within 'DBEServices' includes the Metering Filter and the DBEMetering Service. This 'Metering' folder is also available within the swallow project located at swallow.cvs.sourceforge.net(See Figure 18).

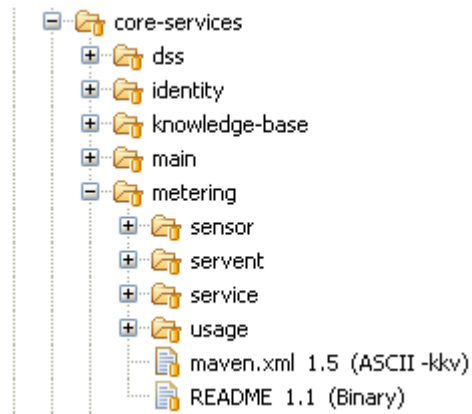


Figure 17: Snapshot of `swallow.cvs.sourceforge.net` structure

- 1 *Deliverable 36.1 - State of the Art in Accounting for Composed Services*
- 2 *ServENT User Guide* - http://gaudi.techideas.info/dist/user_guide
- 3 *Eclipse EMF user guide version 1.0* -
http://download.eclipse.org/tools/emf/scripts/docs.php?doc=docs/UG/EMF_v1.0_Users_Guide.html
- 4 *Network Data Management – Usage (NDM-U) For IP-Based Service, Version 3.1.1, October 9 2002, IPDR Organisation*, http://www.ipdr.org/documents/NDM-U_3.1.1.pdf
- 5 *eXist – Open source Native XML Database* <http://exist.sourceforge.net>