



Digital Business Ecosystem

Contract n° 507953

## **WP36: Composed Service Accounting Architecture Definition**

### **D36.2: A Set of Accounting Software Building Blocks**



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

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

**Deliverable N°:** D36.2  
**Due date:** 30/04/2005  
**Delivery Date:** 17/06/2005

**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:** All project partners and the EC

Versioning			
Version	Date	Author, Organization	Description
0.1	17/06/05	FRANK WALSH - WIT	Initial Document Creation
1.0	22/06/05	FRANK WALSH - WIT	Final Version

**Quality check**

N/A

## Accounting Software Building Blocks

### Table of Contents:

<b>1.EXECUTIVE SUMMARY.....</b>	<b>5</b>
<b>2.INTRODUCTION.....</b>	<b>6</b>
<b>3.METERING.....</b>	<b>7</b>
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
<b>4.MEDIATION.....</b>	<b>12</b>
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.....	15
4.3.1Description.....	15
4.3.2Functional Implementation.....	15
4.4DBE IPDR STORE.....	16
<b>5.RATING &amp; BILLING.....</b>	<b>17</b>
<b>6.ACCESSING SOURCE CODE.....</b>	<b>18</b>
<b>REFERENCES.....</b>	<b>19</b>

## List of Figures:

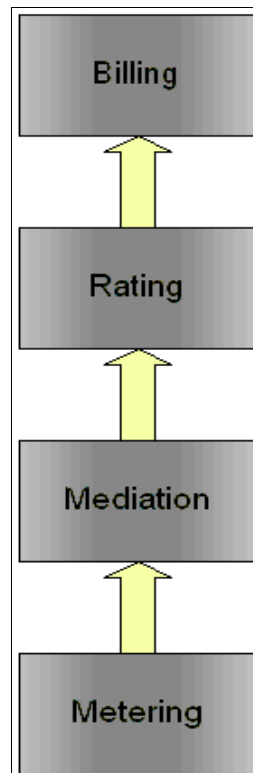
<b>FIGURE1: NETWORK SERVICE ACCOUNTING SYSTEM.....</b>	<b>6</b>
<b>FIGURE 2: METERING REFERENCE MODEL .....</b>	<b>7</b>
<b>FIGURE 3: DBE METERING FILTER CLASS DIAGRAM.....</b>	<b>8</b>
<b>FIGURE 4: DBE METERING SENSOR CLASS DIAGRAM.....</b>	<b>9</b>
<b>FIGURE 5: DBE METERING SERVICE CLASS DIAGRAM.....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

## **1. Executive Summary**

The aim of this document is to describe the software components to be 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 thus far 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 36.1<sup>1</sup>, 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 ServENT<sup>2</sup> 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.

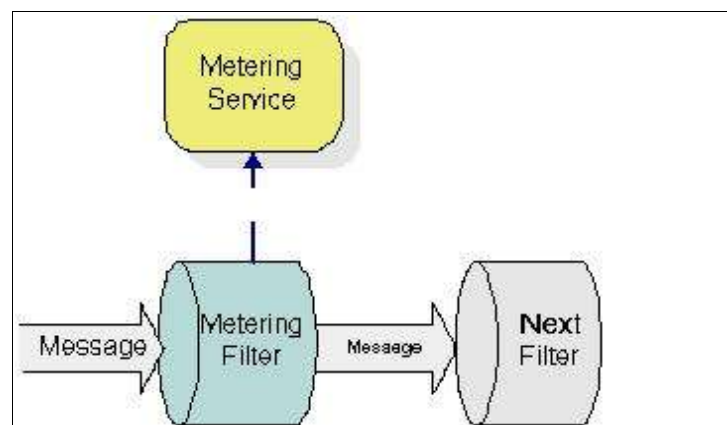


Figure 2: Metering Reference Model

The metered usage data is represented using a generic DBE data model. This data is persisted in the DBE Distributed Storage System (DSS) 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.

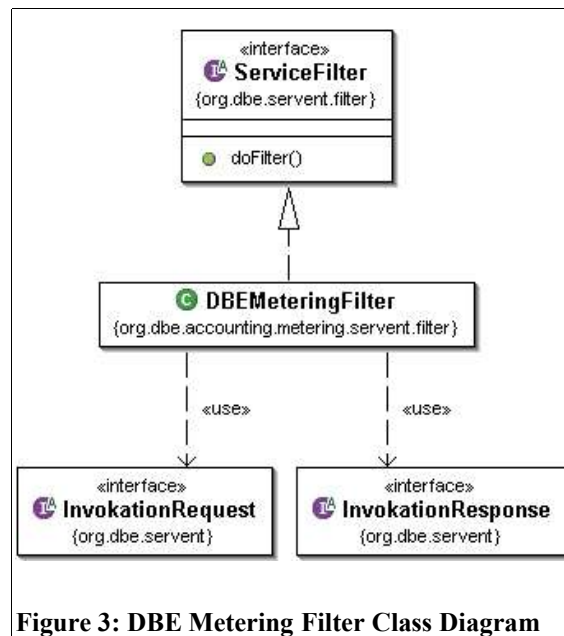


Figure 3: DBE Metering Filter Class Diagram

The source of this implementation is currently available on DBE collabnet (CVS)



## 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

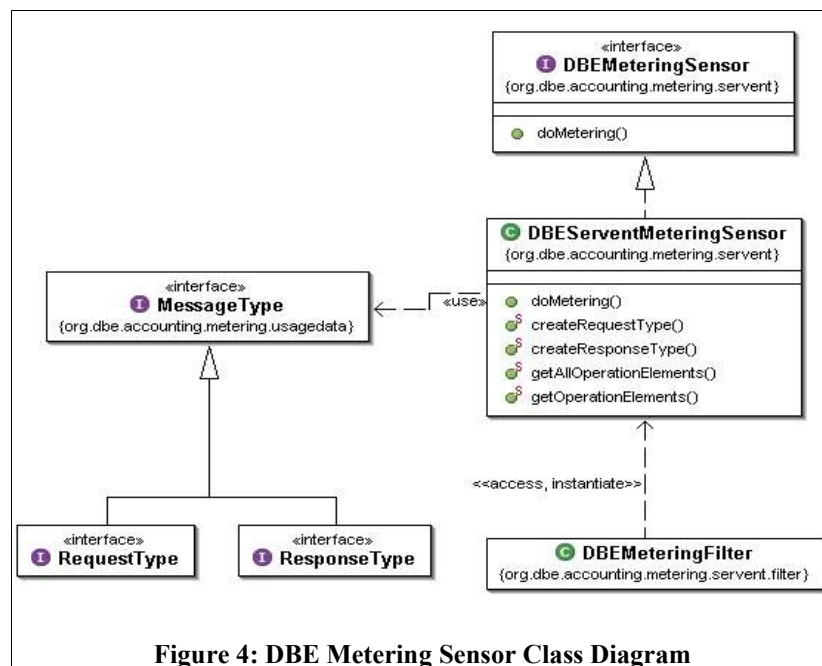


Figure 4: DBE Metering Sensor Class Diagram

The source of this implementation is currently available on DBE collabnet (CVS)

### 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)

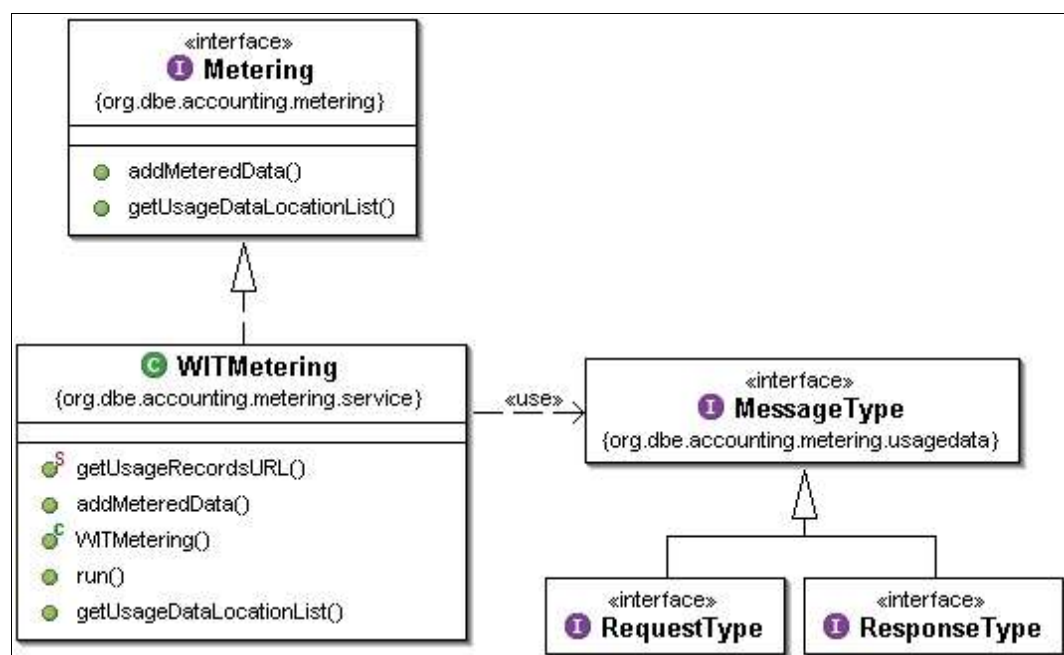


Figure 5: DBE Metering Service Class Diagram

## 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 Framework<sup>3</sup> 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)

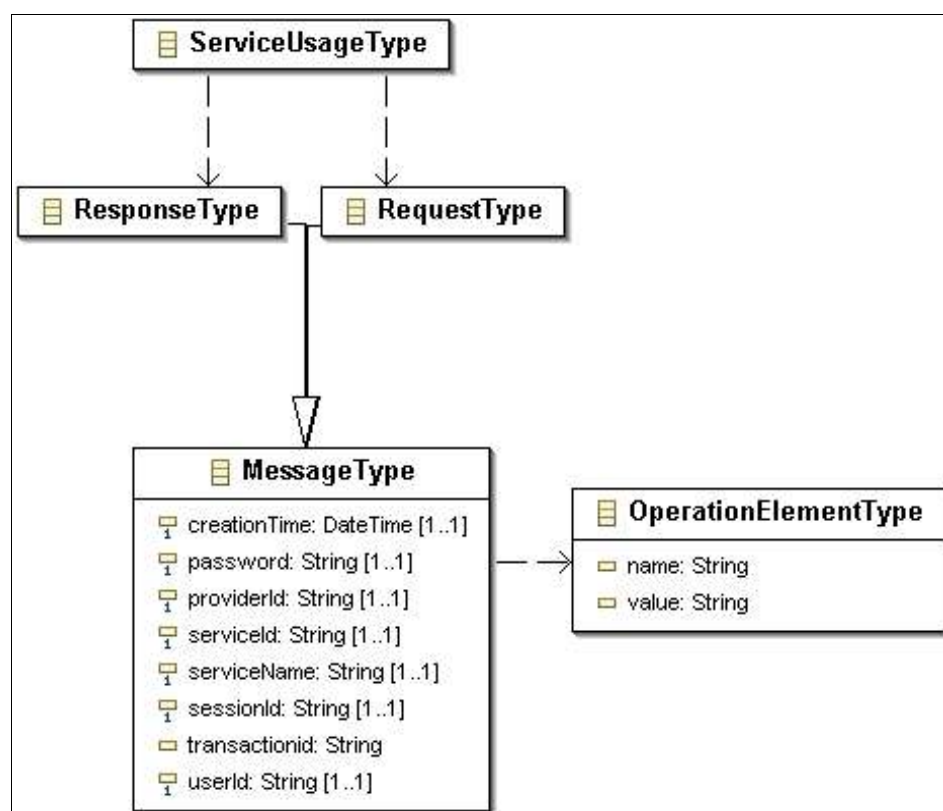
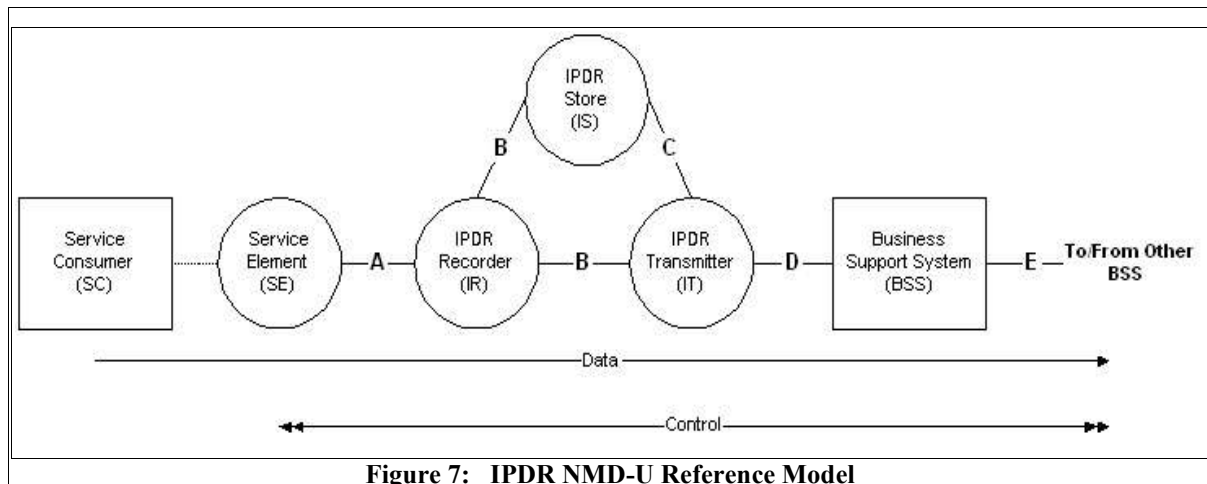


Figure 6: DBE Usage Data Class Diagram

## 4. Mediation

As already stated, mediation complies with IPDR.org NDM-U<sup>4</sup> 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.



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.

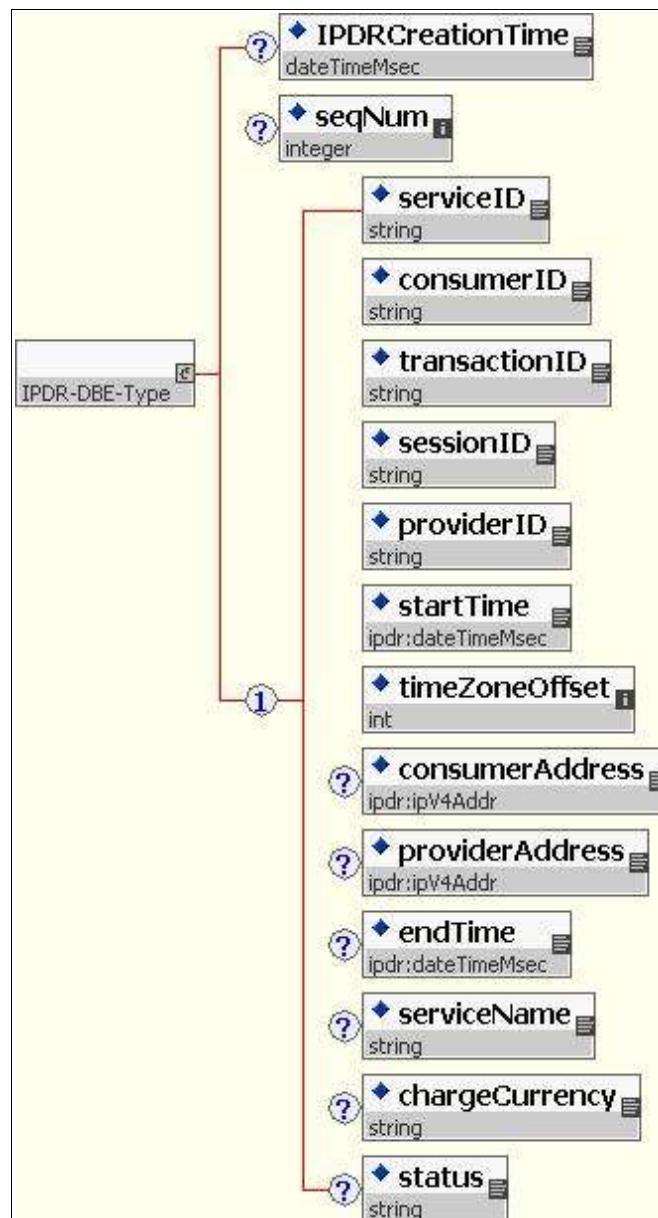


Figure 8: DBE IPDR Schema

### 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 packages DBE Usage Data into DBE IPDRs. presents a stream of IPDRs to an IPDR Store and/or an IPDR Transmitter. Multiple service meters can be connected to a single IPDR Recorder. The IPDR recorder permits both polling and pushing of usage data. Correlation and normalisation of DBE Service requests and responses is also provided in the DBERecorder

### 4.2.2 Functional Implementation

The implementation of the IPDR Recorder is contained in the DBEIPDRRecorder class which implements DBEIPDRRecorder\_A\_Interface. The DBEIPDRRecorder accepts ServiceUsageTypes and converts the usage data into IPDRTypes using a DataReader.

Technologies used in the implementation of the Sensor are:

- DBE Servent
- Eclipse Modeling Framework

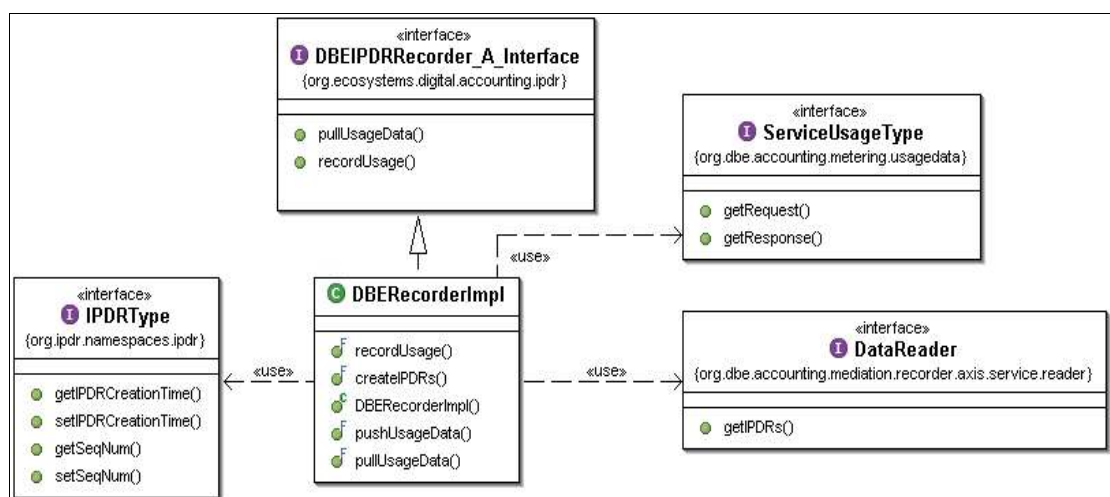


Figure 9: DBE IPDR Recorder Class Diagram

IPDRs are created using a IPDRHelper class which can interrogate DBE usage data and create a corresponding DBE IPDRType.

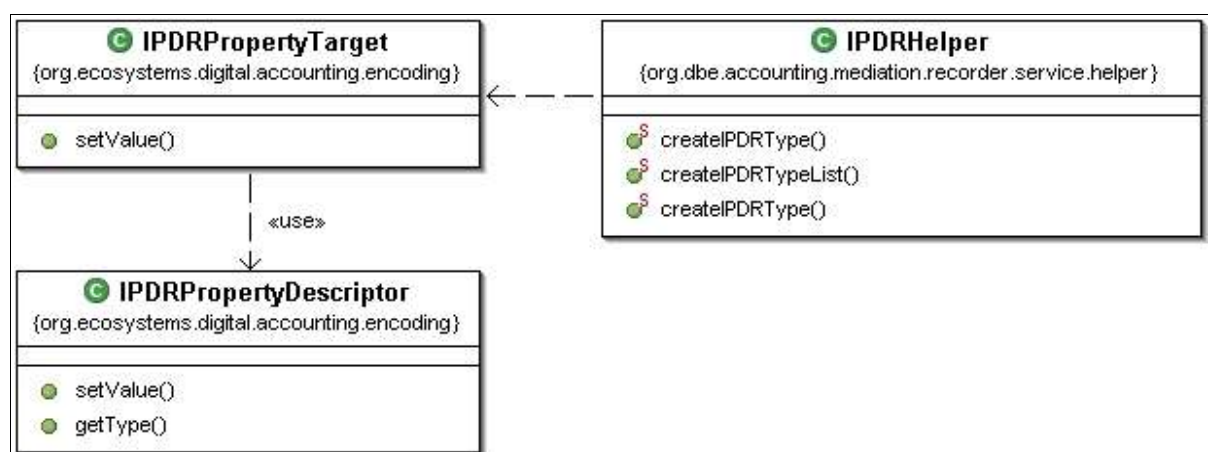


Figure 10: IPDR Helper Class Diagram

## 4.3 DBE IPDR Transmitter

### 4.3.1 Description

The DBE IPDR Transmitter is responsible for packaging IPDR of the same service type into IPDR Documents and transmitting IPDR Documents to a business support system. In the components provided, the business support system is the rating layer shown in figure 1.

### 4.3.2 Functional Implementation

The DBE IPDR Transmitter is implemented through the DBETransmitter class that implements DBEMediationIPDR\_B\_Interface. The Transmitter accepts IPDRTypes from an IPDR Recorder or IPDRStore, packages the IPDRs into IPDRDocuments (IPDRDocType) and transmits them to a Business Support System for Rating.

Technologies used in the implementation of the Sensor are:

- DBE Servent
- Eclipse Modeling Framework(EMCore)

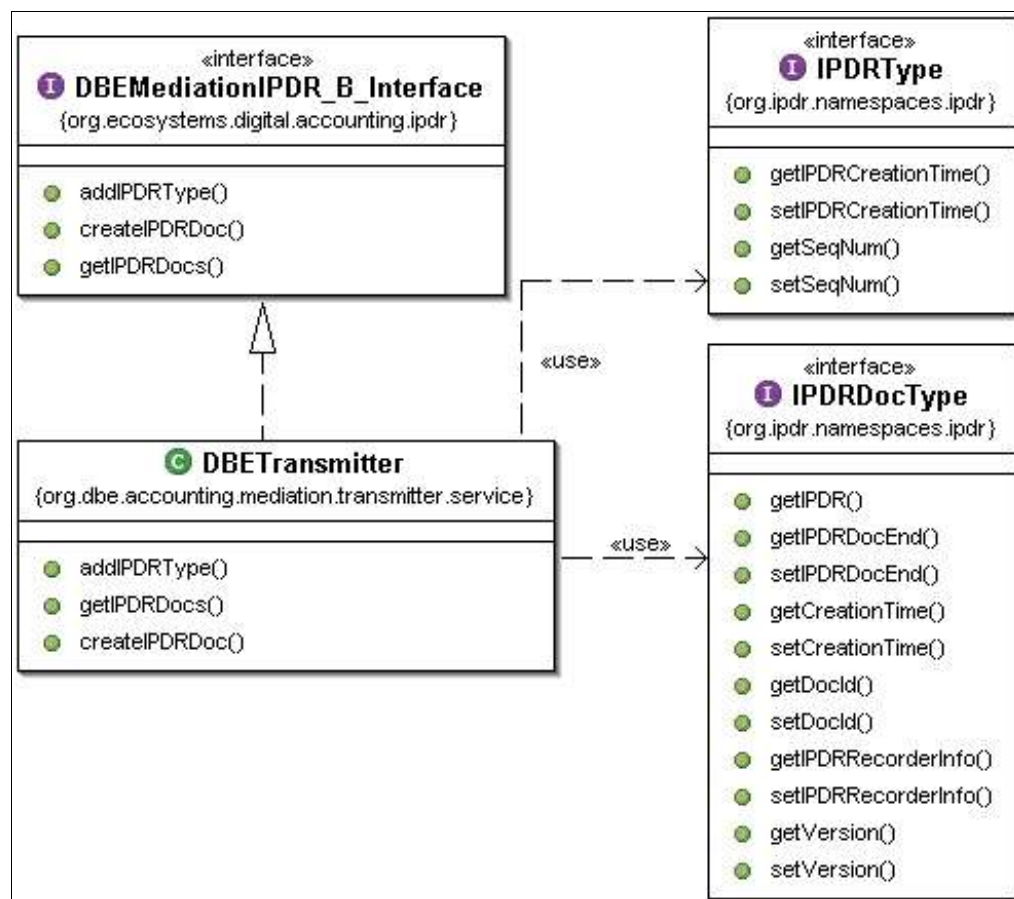


Figure 11: DBE Transmitter Class Diagram

#### **4.4 DBE IPDR Store**

The DBE IPDR store is in the process of development. It is envisioned that the IPDR Store will persist IPDR Documents in the DSS for retrieval by an IPDR Transmitter.

No functional components are available to date.



## **5. Rating & Billing**

At this stage of the project, the Rating and Billing part of the accounting functionality is provided by a previously developed IPDR compliant rating and charging engine, the Rating Bureau Service (RBS)<sup>5</sup>. This component set is not open source and is only be used at this stage as a 'black box' functional block. These components are currently exposed as axis web services and will be re-developed during the second part of the project using open source technologies providing the same functionality as the RBS in an open source fashion.

## **6. Accessing Source Code**

The source code for the components described in this document is currently contained on DBE CVS ([cvs.digital-ecosystem.net:/cvs](http://cvs.digital-ecosystem.net:/cvs)) and is accessible with a valid DBE Collabnet User Id and Password. The modules relating to the accounting components are as follows:

- `src/accounting/mediation`
- `src/accounting/ipdr`
- `src/accounting/metering`
- `src/accounting/rbs`

All DBE source code will be available on SourceForge under the project name Swallow. The date for the release to SourceForge is predicted to be the end of June, 2005.

## References

- 1 *Deliverable 36.1 - State of the Art in Accounting for Composed Services*
- 2 *ServENT User Guide* - [http://gaudi.techideas.info/dist/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](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](http://www.ipdr.org/documents/NDM-U_3.1.1.pdf)
- 5 *Mediation, Rating, Billing in an IP Services Environment* - <http://www.ist-albatross.org/AccountingWhitePaper.pdf>