Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE3.1.2.2.15
Specification of AXMEDIS accounting and reporting, first update of public part of DE9.1.1 and part of DE3.1.2E

Version: 1.6
Date: 14 April 2006
Responsible: EXITECH (revised and closed by coordinator)

Project Number: IST-2-511299 Project Title: AXMEDIS Deliverable Type: report Visible to User Groups: yes Visible to Affiliated: yes Visible to the Public: yes
Deliverable Number: DE3.1.2.2.15 Contractual Date of Delivery: M18 Actual Date of Delivery: 16/4/2006 Title of Deliverable: Specification of AXMEDIS accounting and reporting, first update of public part of DE 9.1.1 and part of DE3.1.2E Work-Package contributing to the Deliverable: WP3.1 Task contributing to the Deliverable: WP3, WP2 Nature of the Deliverable: report Author(s): EXITECH

Abstract: this part includes the specification of components, formats, databases and protocol related to the AXMEDIS Framework area accounting and reporting including CAMART and Administrative Information Integrator

Keyword List: XML, XSLT, Administrative information, CRM, CMS

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. **DEFINITIONS**

   i. "**Acceptance Date**" is the date on which these terms and conditions for entering into possession of the document have been accepted.

   ii. "**Copyright**" stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.

   iii. "**Licensor**" is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.axmedis.org

   iv. "**Document**" means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.

   v. "**Works**" means any works created by the licensee, which reproduce a Document or any of its part.

2. **LICENCE**

   1. The Licensor grants a non-exclusive royalty free license to reproduce and use the Documents subject to present terms and conditions (the **License**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
   2. In consideration of the Licensor granting the License, licensee agrees to adhere to the following terms and conditions.

3. **TERM AND TERMINATION**

   1. Granted License shall commence on Acceptance Date.
   2. Granted License will terminate automatically if licensee fails to comply with any of the terms and conditions of this License.
   3. Termination of this License does not affect either party's accrued rights and obligations as at the date of termination.
   4. Upon termination of this License for whatever reason, licensee shall cease to make any use of the accessed Copyright.
   5. All provisions of this License, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this License and shall continue in full force and effect.
   6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.

4. **USE**

   1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:
      i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;
      ii. change or remove the title of a Document;
      iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or
      iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor

5. **COPYRIGHT NOTICES**

   1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.

6. **WARRANTY**

1. The Licensor warrants the licensee that the present license is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.
2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.
3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfillment of any of his obligations in respect of this License.

7. **INFRINGEMENT**

1. Licensee undertakes to notify promptly the Licensor of any threatened or actual infringement of the Copyright which comes to licensee notice and shall, at the Licensor's request and expense, do all such things as are reasonably necessary to defend and enforce the Licensor's rights in the Copyright.

8. **GOVERNING LAW AND JURISDICTION**

1. This License shall be subject to, and construed and interpreted in accordance with Italian law.
2. The parties irrevocably submit to the exclusive jurisdiction of the Italian Courts.

Please note that:

- You can become affiliated with AXMEDIS. This will give you the access to a huge amount of knowledge, information and source code related to the AXMEDIS Framework. If you are interested please contact P. Nesi at nesi@dsi.unifi.it. Once affiliated with AXMEDIS you will have the possibility of using the AXMEDIS specification and technology for your business.
- You can contribute to the improvement of AXMEDIS documents and specification by sending the contribution to P. Nesi at nesi@dsi.unifi.it
- You can attend AXMEDIS meetings that are open to public, for additional information see WWW.axmedis.org or contact P. Nesi at nesi@dsi.unifi.it

Table of Content

1 Executive Summary and Report Scope

The full AXMEDIS specification document has been decomposed in the following parts:

DE number	Deliverable title	responsible
DE3.1.2.2.1	Specification of General Aspects of AXMEDIS framework, first update of DE3.1.2 part A AXMEDIS-DE3-1-2-2-1-Spec-of-AX-Gen-Asp-of-AXMEDIS-framework-upA-v1-0.doc	DSI
DE3.1.2.2.2	Specification of AXMEDIS Command Manager, first update of DE3.1.2 part B AXMEDIS- DE3-1-2-2-2-Spec-of-AX-Cmd-Man-upB-v1-0.doc	DSI
DE3.1.2.2.3	Specification of AXMEDIS Object Manager and Protection Processor, first update of DE3.1.2 part B AXMEDIS-DE3-1-2-2-3-Spec-of-AXOM-and-ProtProc-upB-v1-0.doc	DSI
DE3.1.2.2.4	Specification of AXMEDIS Editors and Viewers, first update of DE3.1.2 part B AXMEDIS-DE3-1-2-2-4-Spec-of-AX-Editors-and-Viewers-upB-v1-0.doc	DSI
DE3.1.2.2.5	Specification of External AXMEDIS Editors/Viewers and Players, first update of DE3.1.2 part B AXMEDIS-DE3-1-2-2-5-Spec-of-External-Editors-Viewers-Players-upB-v1-0.doc	EPFL
DE3.1.2.2.6	Specification of AXMEDIS Content Processing, first update of DE3.1.2 part C AXMEDIS-DE3-1-2-2-6-Spec-of-AX-Content-Processing-upC-v1-0.doc	DSI
DE3.1.2.2.7	Specification of AXMEDIS External Processing Algorithms AXMEDIS-DE3-1-2-2-7-Spec-of-AX-External-Processing-Algorithms-v1-0.doc	FHGIGD
DE3.1.2.2.8	Specification of AXMEDIS CMS Crawling Capabilities, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-8-Spec-of-AX-CMS-Crawling-Capab-v1-0.doc	DSI
DE3.1.2.2.9	Specification of AXMEDIS database and query support, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-9-Spec-of-AX-database-and-query-support-v1-0.doc	EXITECH
DE3.1.2.2.10	Specification of AXMEDIS P2P tools, AXEPTool and AXMEDIS, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-10-Spec-of-AXEPTool-and-AXMEDIA-tools-v1-0.doc	CRS4
DE3.1.2.2.11	Specification of AXMEDIS Programme and Publication tools, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-11-Spec-of-AX-Progr-and-Pub-tool-v1-0.doc	UNIVLEEDS
DE3.1.2.2.12	Specification of AXMEDIS Workflow Tools, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-12-Spec-of-AX-Workflow-Tools-v1-0.doc	IRC
DE3.1.2.2.13	Specification of AXMEDIS Certifier and Supervisor and networks of AXCS, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-13-Spec-of-AXCS-and-networks-v1-0.doc	DSI
DE3.1.2.2.14	Specification of AXMEDIS Protection Support, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-14-Spec-of-AX-Protection-Support-v1-0.doc	FUPF
DE3.1.2.2.15	Specification of AXMEDIS accounting and reporting, first update of part of DE3.1.2 AXMEDIS-DE3-1-2-2-15-Spec-of-AX-Accounting-and-Reporting-v1-0.doc	EXITECH

1.1 This document concerns

Specification of AXMEDIS accounting and reporting, first update of part of DE3.1.2.

1.2 List of Modules or Executable Tools Specified in this document

A module is a component that can be or it is reused in other cases or points of the AXMEDIS framework or of other AXMEDIS based solutions.
The modules/tools have to include effective components and/or tools and also testing components and tools.

Module/tool Name	Module/Tool Description and purpose, state also in which other AXMEDIS area is used	Standards exploited if any
CAMART	Core Accounting Manager and Reporting Tool is a part of the framework and is comprised of 2 webservices that are able to gather data from AXCS and SuperAXCS and to generate statistics on the fly or to store in the local DB information related to the transaction made on object pertinent to the factory issuing the query.	
AII	Administrative Information Integrator is a tool capable of collecting data gathered from CAMART and to produce administrative reporting accorting to a standard compliant with different CMSes	

1.3 List of Formats Specified in this document

A format can be (i) an XML content file for modeling some information, (ii) a file format for storing information, (iii) a format that is manipulated by the tools described in this document, etc...

Format Name	Format Description and purpose, state also in which other modules is used	Standards exploited if any
AII exchange format	XML format for sending data to company CRM	

1.4 List of Databases Specified in this document

Database Name	database Description and purpose, state also in which other AXMEDIS area is using	Standards exploited if any
AXDB	the AXDB part that is related to CAMART and AII	

1.5 List of Protocols Specified in this document

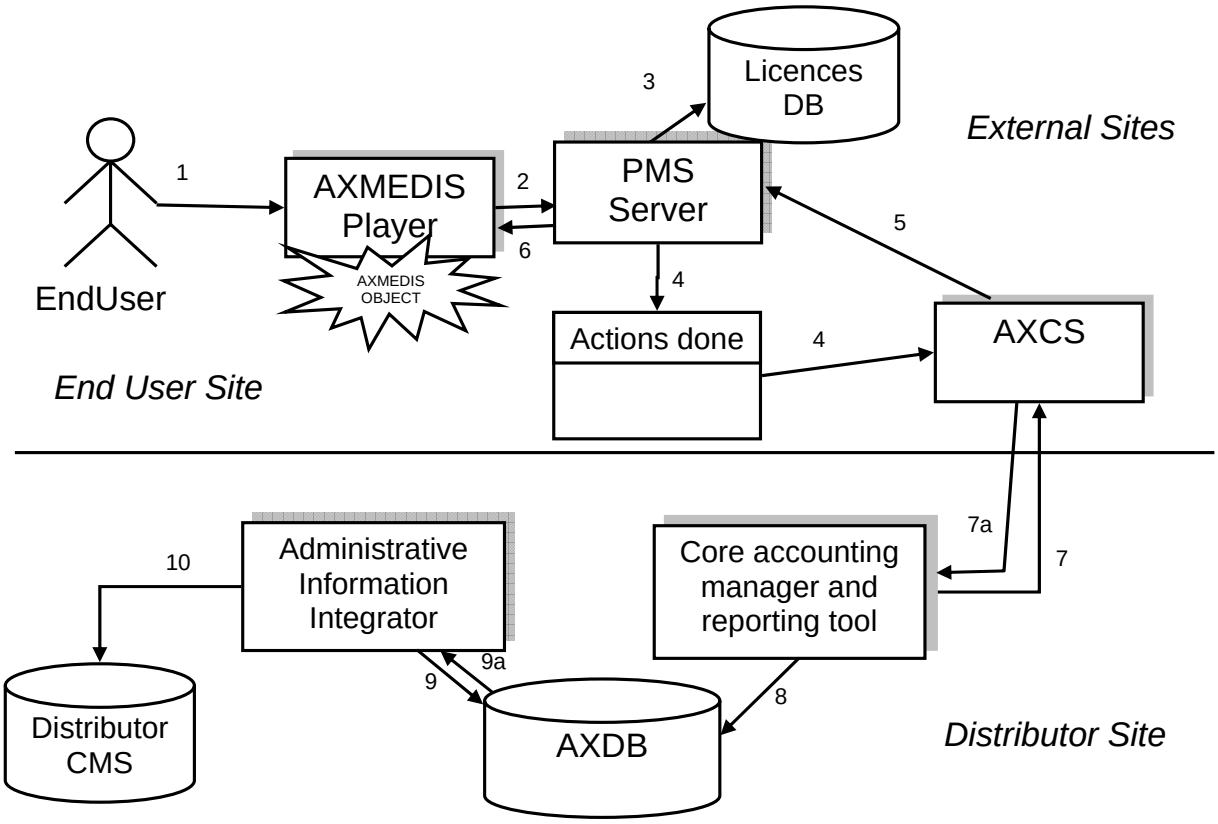
A protocol is a communication modality among distinct processes that can be located or not on different computers.

Protocol Name	protocol Description and purpose, state also in which other modules is used	Who is the master and who is the slave	Standards exploited if any

2 General Use Cases and scenarios

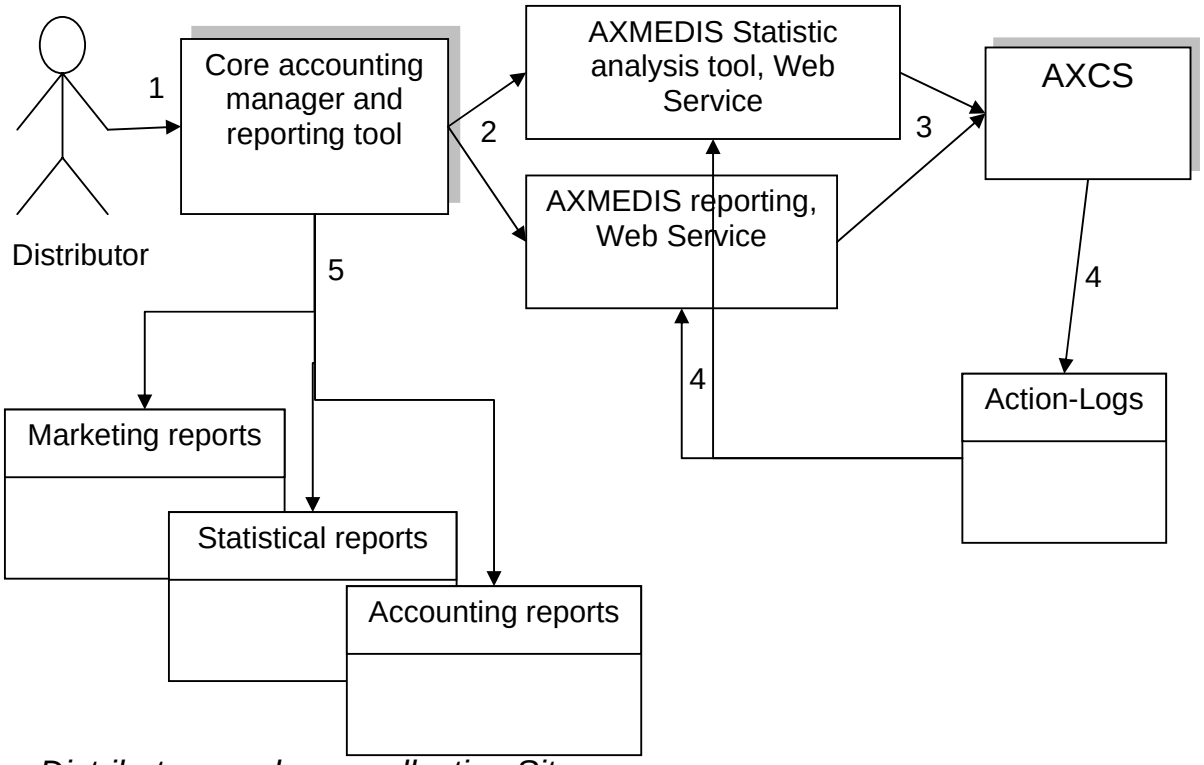
- Mapping of Administrative Information
- Distributor wants Administrative Information
- Creator or Collecting Society wants Administrative Information
- Administrative Information Integrator
- Administrative Information Integrator as seen from Collecting society perspective
- Relationships between CAMART and AII

2.1 Mapping of Administrative Information



1. End User requests to perform an action on an AXMEDIS Protected Object
2. AXMEDIS Player asks PMS to perform an Action (assuming client has been already certified)
3. PMS checks in the Licence DB if the Action is allowed (assuming OK)
4. PMS sends AXCS the action performed
5. AXCS returns the key to access the content (if necessary)
6. PMS grants the access to the content and possibly returns the key to the AXMEDIS Player
7. CAMART retrieves from AXCS the actions performed by all the End Users on objects distributed by the distributor
8. CAMART stores the transactions into the AXDB
9. Administrative Information Integrator gets transactions performed from the AXDB
10. Administrative information are mapped into the Distributor CMS

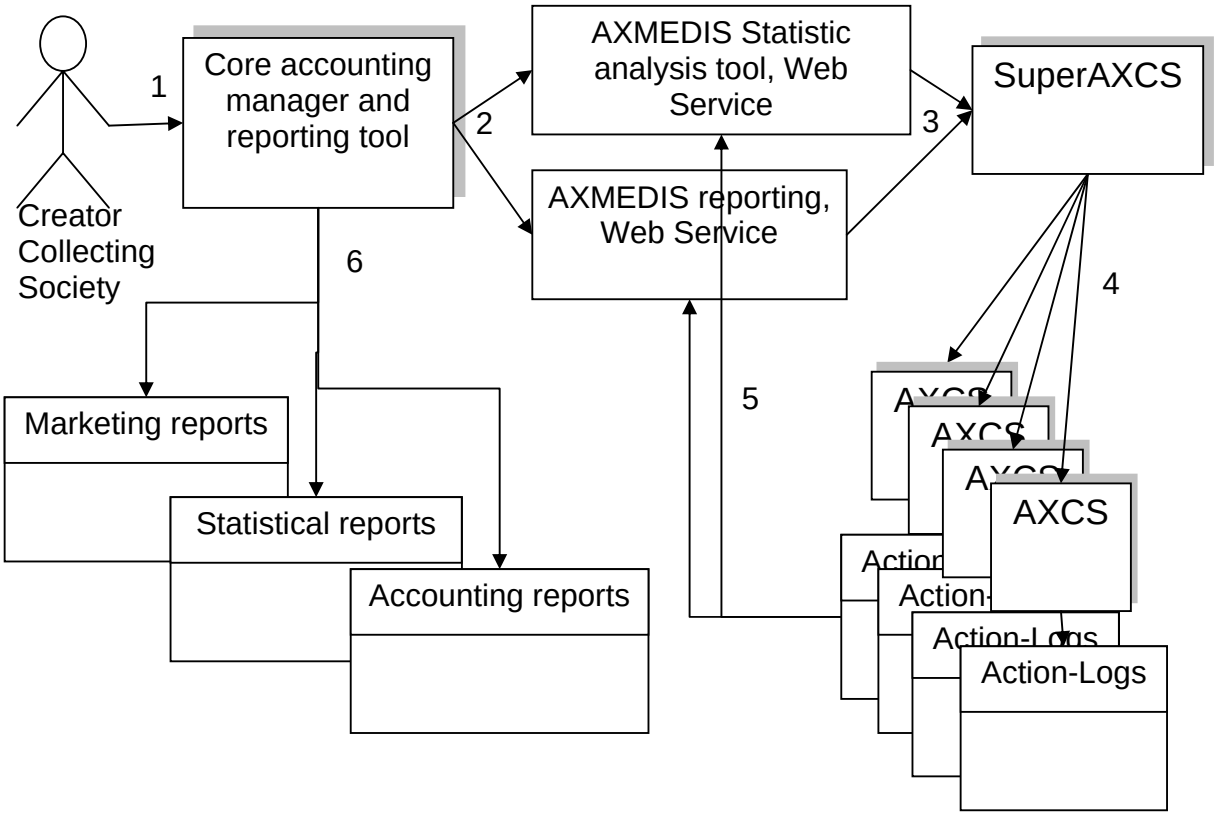
2.2 Distributor wants Administrative Information



Distributor, producer, collecting Site

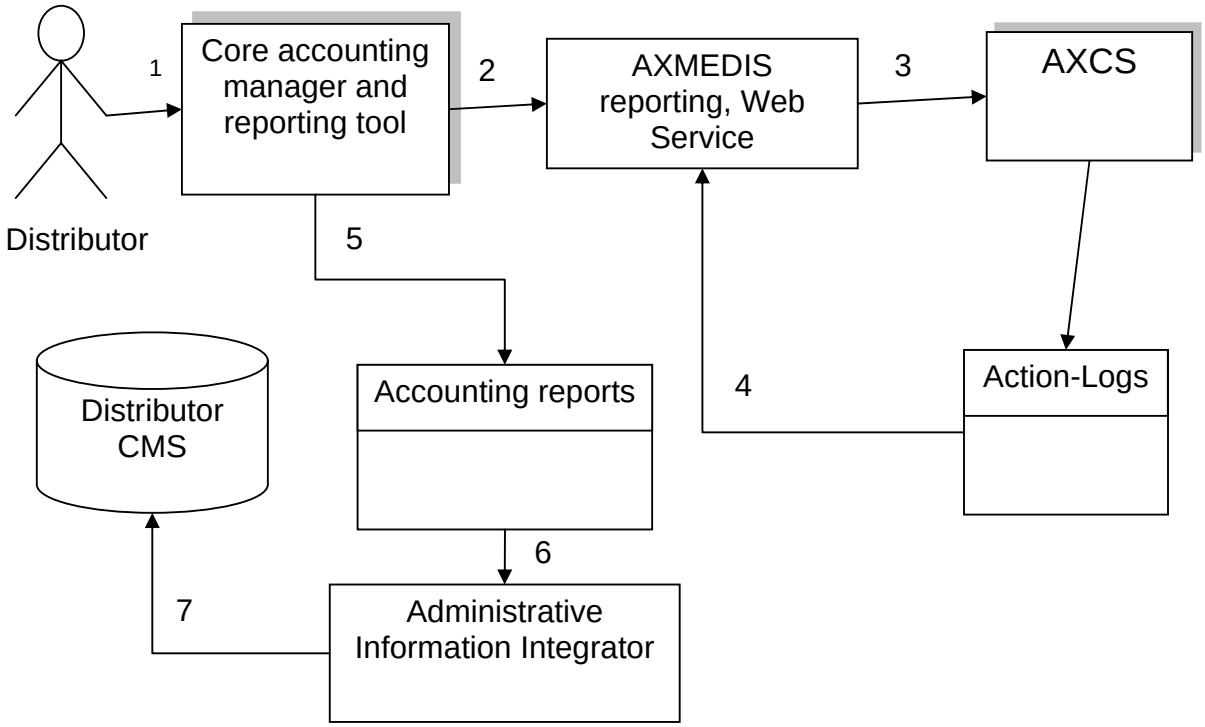
1. A Distributor wants to obtain information on actions performed on the objects he has rights for.
2. CAMART queries the correct tool for obtaining the Action-Logs in the correct form (anonymous or not, aggregated or not, etc)
3. AXMEDIS Statistic or reporting tools query AXCS
4. AXCS extracts the required Action-Logs and communicate them to the tools that perform actions to return results in the desired form
5. Different reports are generated on the basis of the information collected.

2.3 Creator or Collecting Society wants Administrative Information



1. An Actor, that is collecting society or creator, wants to recover information on actions performed on the objects he has rights.
2. Core accounting manager and reporting tool query the correct tool for obtaining the Action-Logs in the correct form (anonymous or not, aggregated or not, etc)
3. AXMEDIS Statistic or reporting tools query the SuperAXCS
4. SuperAXCS recover information from the different AXCSs
5. The different AXCSs extract the required Action-Logs and communicate them to the tools that perform actions to return results in the desired form
6. Different reports are generated on the basis of the information collected.

2.4 Administrative Information Integrator



1. A Distributor wants to recover information on actions performed on the objects he has rights.
2. CAMART queries the reporting web service to obtain the Action-Logs
3. AXMEDIS Statistic or reporting tools queries AXCS
4. AXCS extracts the required Action-Logs and communicate them to the reporting tool
5. Accounting report is generated.
6. Accounting report is passed to the Administrative Information Integrator
7. Data are loaded into the Distributor CMS

2.5 Administrative Information Integrator as seen from Collecting society perspective

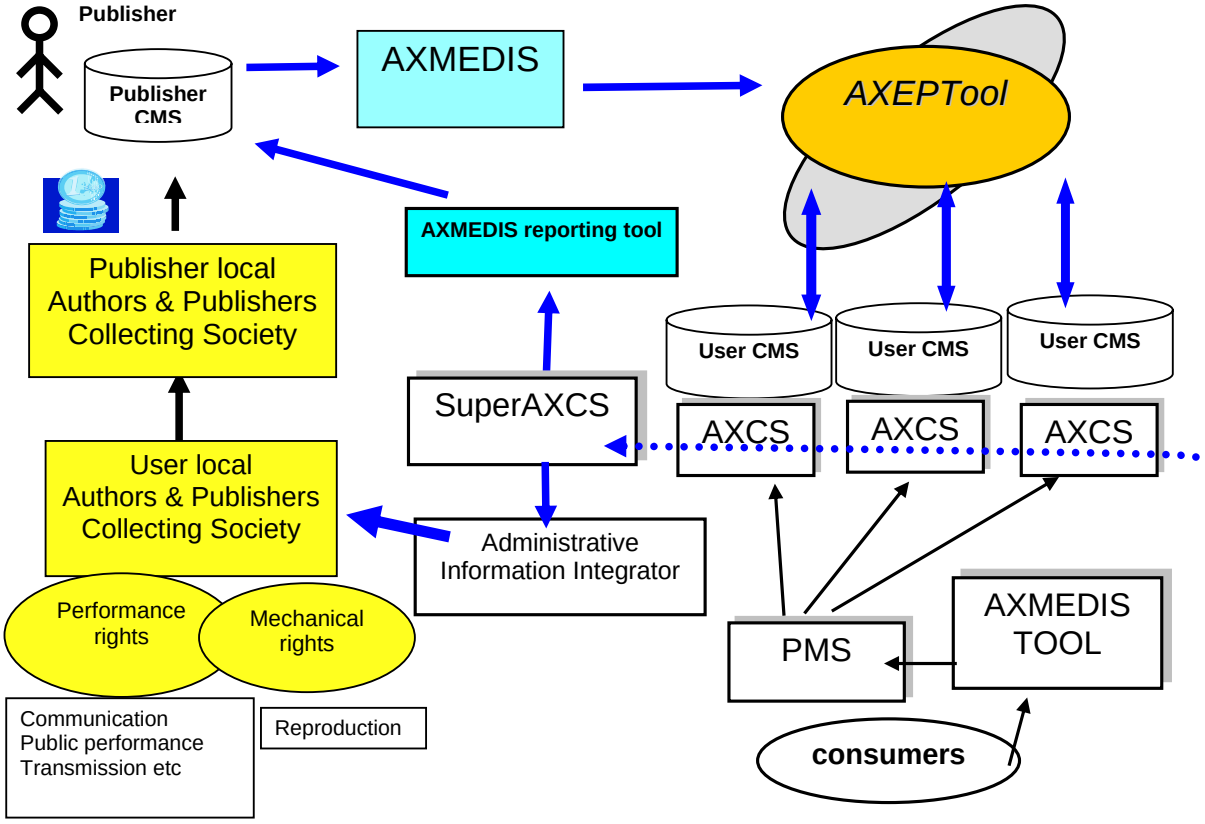
The following three scenarios describe how AXMEDIS provides a service to the Collecting Societies, supporting them in gathering reporting information on the use of protected objects so to enhance the management, ease the administration administration and enforce the rights they are vested in or represent.

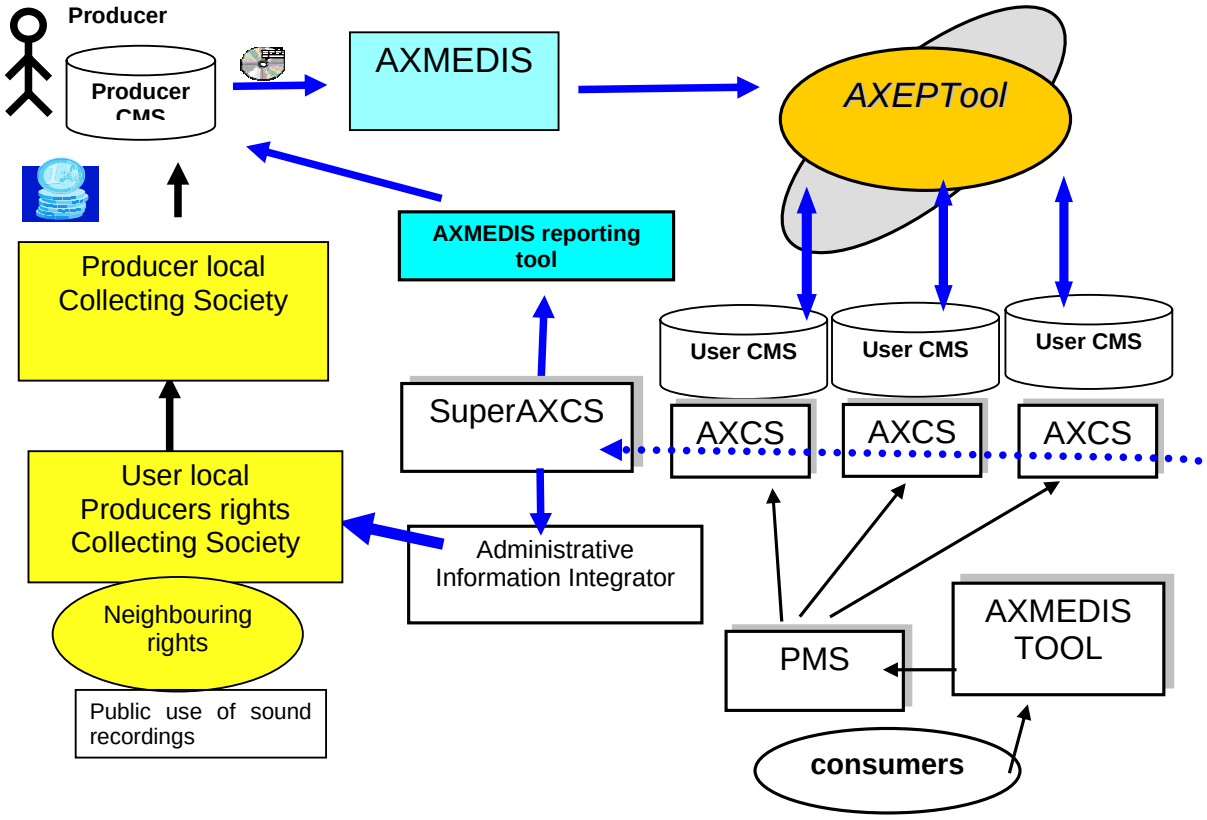
Collecting societies administer a wide range of rights on behalf of copyright owners for a wide range of uses and users. They collect and distribute to right owners royalty income and equitable remuneration in relation to the exercise of these rights. In addition to these core functions, there are many other functions carried out by all or many Collecting societies such as enforcement, monitoring and auditing activities, particularly important in view of the increasingly uses of copyrighted content in the AXMEDIS context.

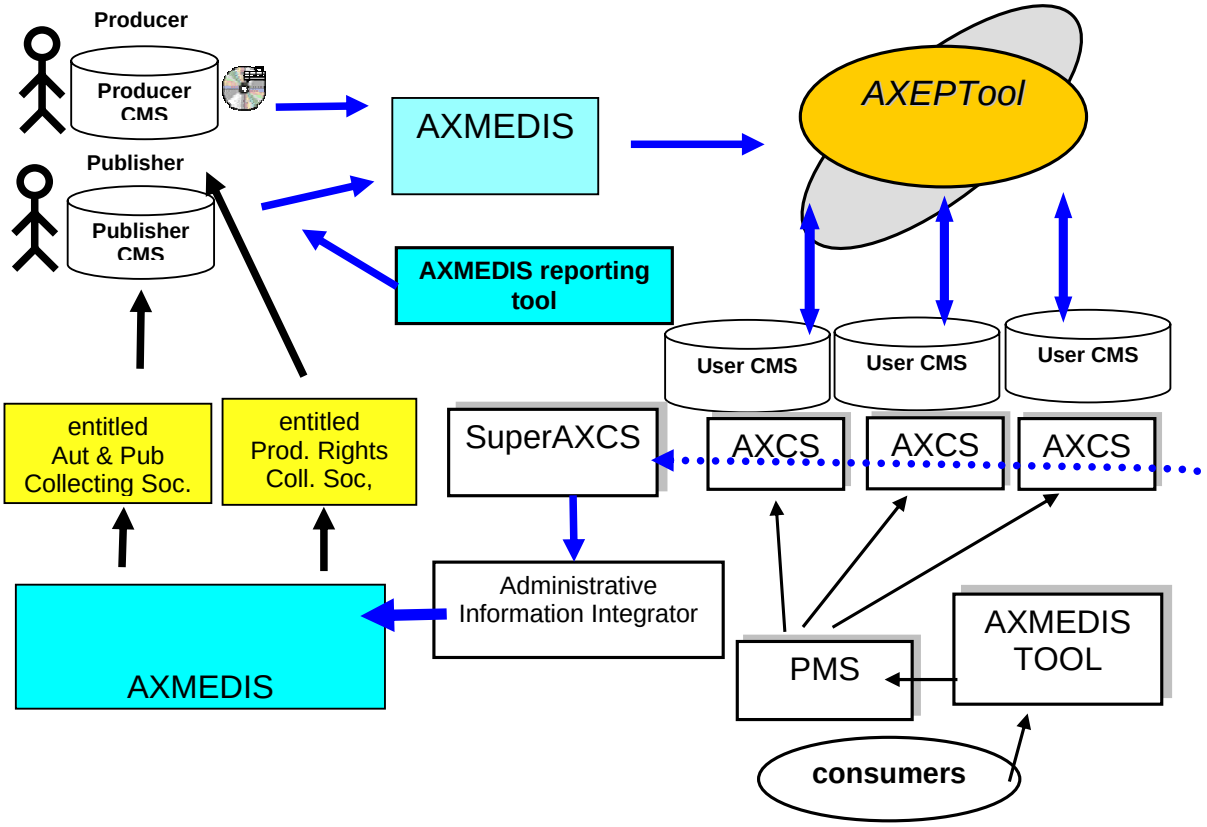
It has to be underline that these scenarios only refer to the use of music whose exploitation rights are granted to the original publisher and the producer. Their content, once governed as an AXMEDIS object, are ready to be exploited within the AXEPTOOL and distributed accordingly to the DRM and license terms provided. All actions (uses) or events performed on these AXMEDIS objects are recorded in the AXCS of each user and then reported, along with other relevant data, to the super AXCS. The super AXCS tool will then interact with the AXMEDIS reporting tool and with the Administrative Information Integrator that will respectively report relevant information into the right owners' database and into the database of the entitled collecting societies. It has to be underline that the link between AXMEDIS tools and the collecting societies should be implemented by taking into account new tools and network developed by collecting societies themselves such as the FastTrack project. This project aims at realizing a global interconnected network of databases on

musical and audiovisual works, rights owners, contracts and data on sound recording to support diary operations of the societies involved such as identification of works and distribution of royalties

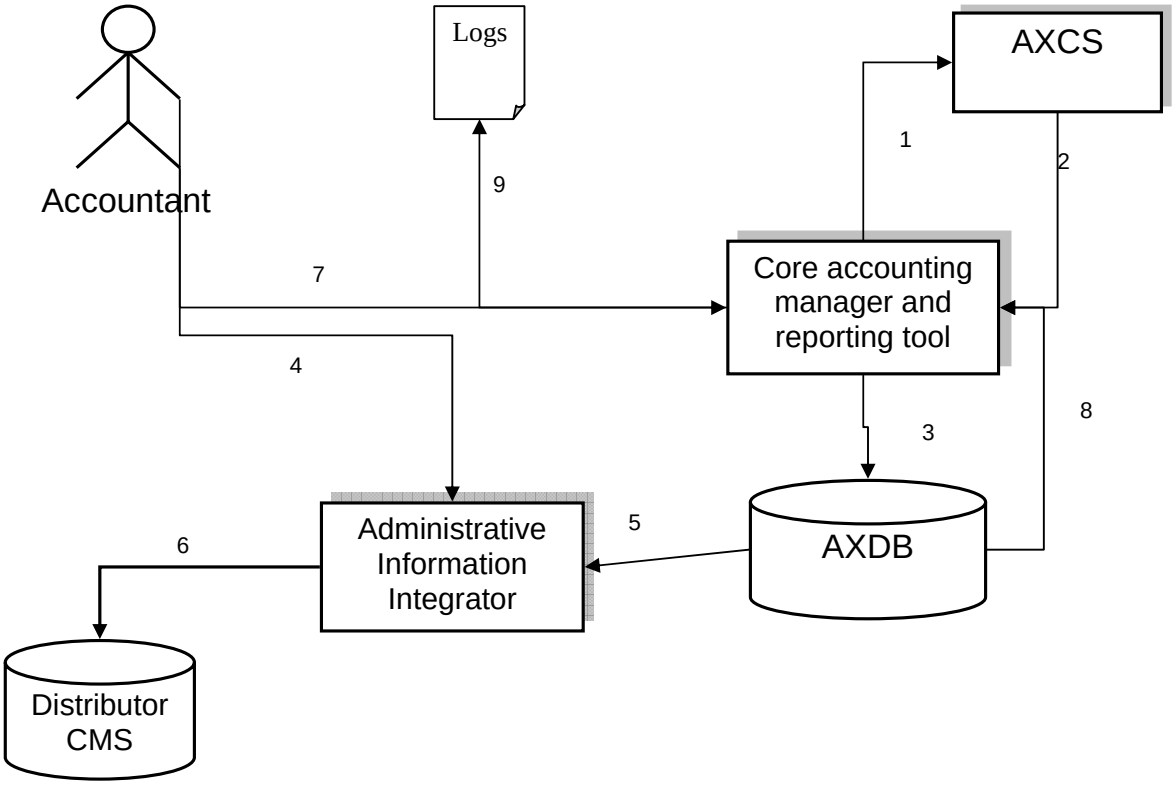
Independently of the ways and methodologies rights are granted to users (e.g. compulsory license, individual license) the Administrative Information integrator tool should provide Collecting societies with data needed to check, verify and monitor the use of the AXMEDIS objects in conformity with the rights granted by the relevant license and with information necessary to identify right owners including identification standard codes already developed (such as the ISRC) as well as those under development.
An AXMEDIS object will involve a multiplicity of rights owners (there could be many just in one musical work) and many different collecting societies (such as public performance rights societies, mechanical rights societies, producers rights societies, performers rights societies etc). Its multi distribution channels will allow multiple reproductions, transmissions and retransmissions until it reaches the end user/consumer.
Due to this issue and to the complexity of the different rules that govern the collection and distribution of royalties for the exploitation of multimedia contents and compounded objects in the digital environment, the control of the correct use of the rights granted and the consequent collection of the royalties due become a complex task and one of the main concern of right holders community.




Producer
Producer CMS
AXMEDIS
AXEPTool
AXMEDIS reporting tool
Producer local Collecting Society
User CMS
User CMS
User CMS
SuperAXCS
AXCS
AXCS
AXCS
User local Producers rights Collecting Society
Administrative Information Integrator
Neighbouring rights
Public use of sound recordings
PMS
AXMEDIS TOOL
consumers


Producer
Producer CMS
Publisher
Publisher CMS
AXMEDIS
AXEPTool
AXMEDIS reporting tool
User CMS
User CMS
User CMS
entitled Aut & Pub Collecting Soc.
entitled Prod. Rights Coll. Soc,
SuperAXCS
AXCS
AXCS
AXCS
Administrative Information Integrator
AXMEDIS
PMS
AXMEDIS TOOL
consumers

2.6 Relationships between CAMART and AII

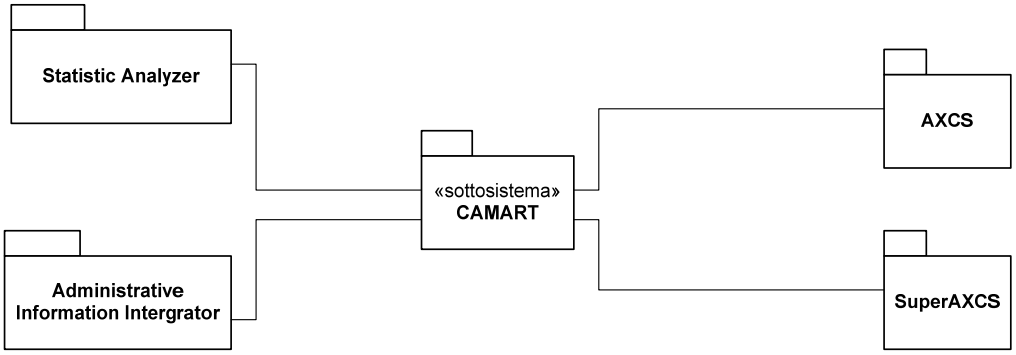


1. CAMART contact AXCS web service for having fresh logs
2. AXCS give back to CAMART the Logs
3. CAMART store logs on AXDB
4. Accountant configure AII to put export logs toward CMS
5. AII get logs from AXDB
6. AII export logs to CMS
7. Accountant can optionally query directly the CAMART for logs filtered by AXOID, AXUID, timestamp etc
8. In that case CAMART will get the log from AXDB
9. CAMART will generate a raw XML report of Logs

3 General architecture and relationships among the modules produced

The AXMEDIS accounting area is comprised of CAMART module that interacts with Webservices provided by AXCS and SuperAXCS and by a couple of module that have in charge the statistical analysis and the reporting toward CMS (Administrative Information Integrator).

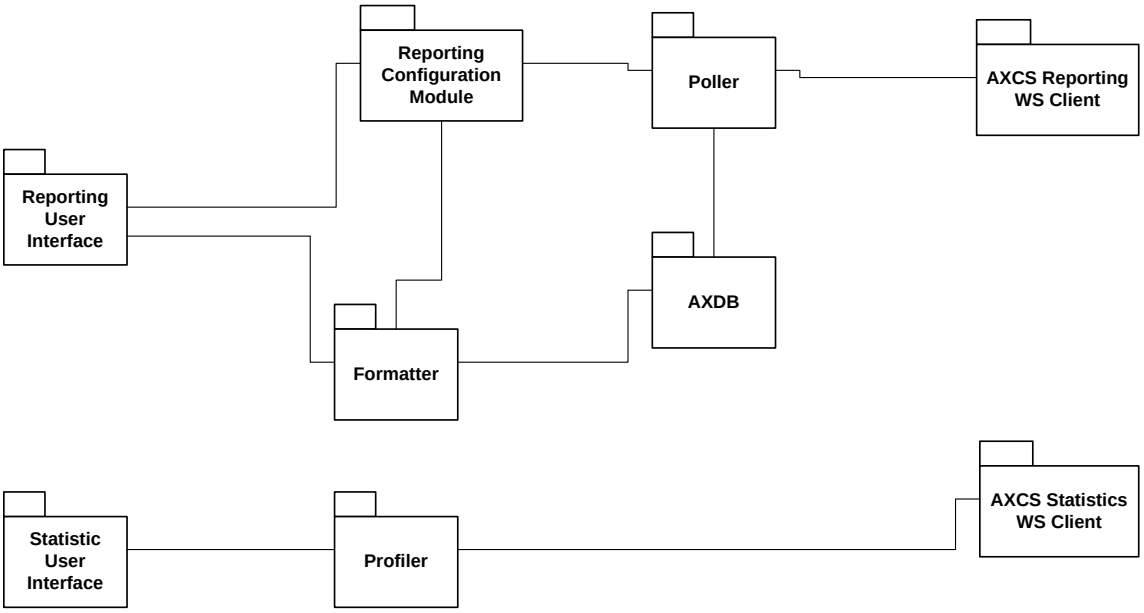
Accounting and Reporting Area				
Acoounting Reporting	1.0	General Overview of Accounting and Reporting Area	04/04/2006	FF



It is important to detail something more the CAMART module that is comprised of several sub-modules. At the first glance we note that Statistic and reporting engines are separated since they are connected to different web services and it is also relevant that only the reporting part is connected to the AXDB, while the statistical part operates as a producer/consumer chain with a intermediate stage of profiling.
The module in the following diagrams are then:

- *AXCS Reporting WS Client*, that is the client of the AXCS Reporting Web Service that extracts logs from AXCS
- *Poller*, that is a module that reads logs from the previous module controlling also the timing and stores such logs in the AXDB
- *Reporting Configuration Module*, that configures the user profile assigning to each user a different location for the storing of the results and a different formatting of the output
- *Formatter*, that gets the logs from the AXDB and according to the configuration creates an output on the basis on the user needs
- *Reporting User Interface*, controls the configuration management and allow the direct view of the results
- *AXCS Statistics WS Client*, that is the client of the AXCS Statistics Web Service that extracts logs from AXCS
- *Profiler*, that elaborates high level statistics on the data provided by the web service client such as top ten and bottom ten, top distributors and bottom distributors and so on
- *Statistics user interface*, allows the user to choose the high level statistics needed and the filtering to be performed on the statistical data

CAMART Module				
CAMART	1.0	CAMART module of Accounting and Reporting Area	04/04/2006	FF



4 Module or Executable Tool Core Accounting Manager and Reporting Tools (CAMART)

Module/Tool Profile		
CAMART		
Responsible Name	Villoresi	
Responsible Partner	EXITECH	
Status (proposed/approved)	approved	
Implemented/not implemented	implemented	
Status of the implementation	under development	
Executable or Library/module (Support)	Web Application	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
Platforms supported	All supported by JAVA	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/Framework/source/camart/	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/repos/Framework/bin/camart/	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user and Passwd) if any		
Test cases (present/absent)	absent	
Test cases location	none	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

4.1 General Description of the Module

4.1.1 Core Accounting Manager and Reporting Tools interface toward the system

The role of Core Accounting manager and Reporting Tool (CAMART) is strictly bound with database for logs (provided by AXCS) since it has to:

- collect information from AXMEDIS Certifier and Supervisor about Action Log and store them into the AXMEDIS database, homogenizing fragmented information the different tools sent to the AXCS. This is very important as the AXCS or the SuperAXCS will not store forever data related to each user, but will use policies to remove detailed data, after a certain amount of time or after some triggered operations that have to be defined. **Only data for statistical purposes will be kept**;
- collect information regarding the B2B. For example to have a table of Business counterparts that distribute, integrate, etc. AXMEDIS Objects. They should have a contract with the AXMEDIS Factory that host the AXMEDIS database (AXDB). This information is useful to interpreting the Action Log Information coming from the AXCS databases, when available in the AXCS.

The information will be collected on schedule and CAMART will act as a client of the AXCS Reporting Web Service.

AXMEDIS system has to be scalable and therefore we will have to deal with the fact that some installation can have AXDB, AXCS and other supporting tools on different machines, while others can be less distributed due to a lesser need for speed or storage capacity.

The core accounting manager is a sort of Client side of the bridge between the AXDB and the AXCS databases in order to allow AXCS to be independent by the database. The server side in the AXCS is the Web Service: AXMEDIS Reporting Web Service. The CAMART can be interpreted as a part of the

AXMEDIS Database Interface, since is the part of the system that allows writing data related to Action-Logs into the DB.

This aspect is clarified by the scenario reported in Section 2.1.

4.1.2 Core Accounting Manager and Reporting Tools interface toward the user

Before drafting a detailed specification for the second aspect of this tool, reviewing practical scenarios will highlight the interaction between the user and the tool and show how the other AXMEDIS parts are involved. These scenarios are summarized by the use cases in Sections 2.2 and 2.3.

The actor working on the CAMART user interface can be any administrative and management user that has interest in making queries and browsing the information related to the usage of AXMEDIS objects. For example:

- A Distributor could be interested in seeing the list of Action Logs related to a given second distributor, integrator, etc.
- An Integrator could be interested in seeing the list of Action Logs related to a given AXMEDIS Object, etc.
- A Distributor could be interested in seeing the list of second Distributors that have exploited some specific AXMEDIS object, etc.
- A Distributor could be interested to see how many transactions have been registered on its AXMEDIS objects in the last two months.

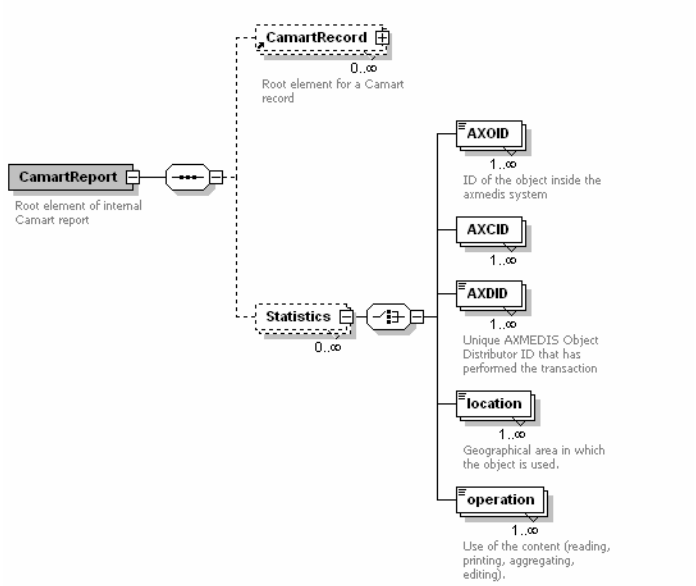
These examples can be used by the Actor (account manager) to extract the information and move it into the Administrative Database of the CMS by means of the AXMEDIS tool called Administrative Information Integrator.

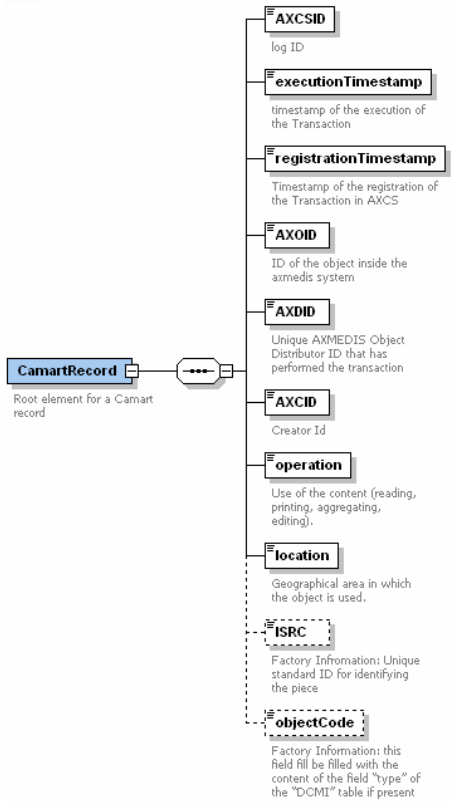
Statistics recovered by the AXCS are only general information and trends. This means that are not contextualized in terms of specific Object ID, etc.

The general role of CAMART with respect to the end user that uses it is to provide a web interface for making reporting and statistical queries directly onto the system.
We have to consider in a different way Reporting and statistical queries since:

- Reporting queries are always executed on the AXMEDIS Database Interface for collecting information that are contained in the local database that in turn had been gathered from the AXCS. In this way the actor sees only the data for which is authorized;
- Statistical queries will always be resolved by the SuperAXCS by the means of Statistic Analysis Tool, Web Service. CAMART will act as the client side of the Web Service and will display results to the user in the web interface.

The user interface, if needed, will be web based in order to cover the needs of interoperability, usability and maintainability that are expected from AXMEDIS in the large sense.
The web interface presented in section 4.3 is capable of calling the AXCS Statistics Web Service in order to extract anonymous logs in a certain period of time with some filters applied on them.
Moreover it is possible to generate also high level statistics to be published on a web portal or used by internal company departments such as marketing, planning and so on.
The format that is provided to the user is generated according to the XML schema reported in section 8. A graphic view of the schema is reported below:





4.1.3 CAMART interface with AXCS

CAMART needs to get fresh logs from AXCS and therefore it is a client of the web service interface that AXCS will expose to have access to log information.

In this section it is defined the interface in order to specify in terms of WSDL this communication protocol that will be a synchronous polling from CAMART to AXCS.

AXCS will have to implement two web services and CAMART will be a client for them, reading at predefined time interval the logs from AXCS and storing them in the local AXDB related to Logs for the reporting part and reading on demand for the statistical part.

The reporting process is completely automatic and is managed by a CAMART daemon that will periodically extract from AXCS all the logs for which the user is entitled and store them in the AXDB.

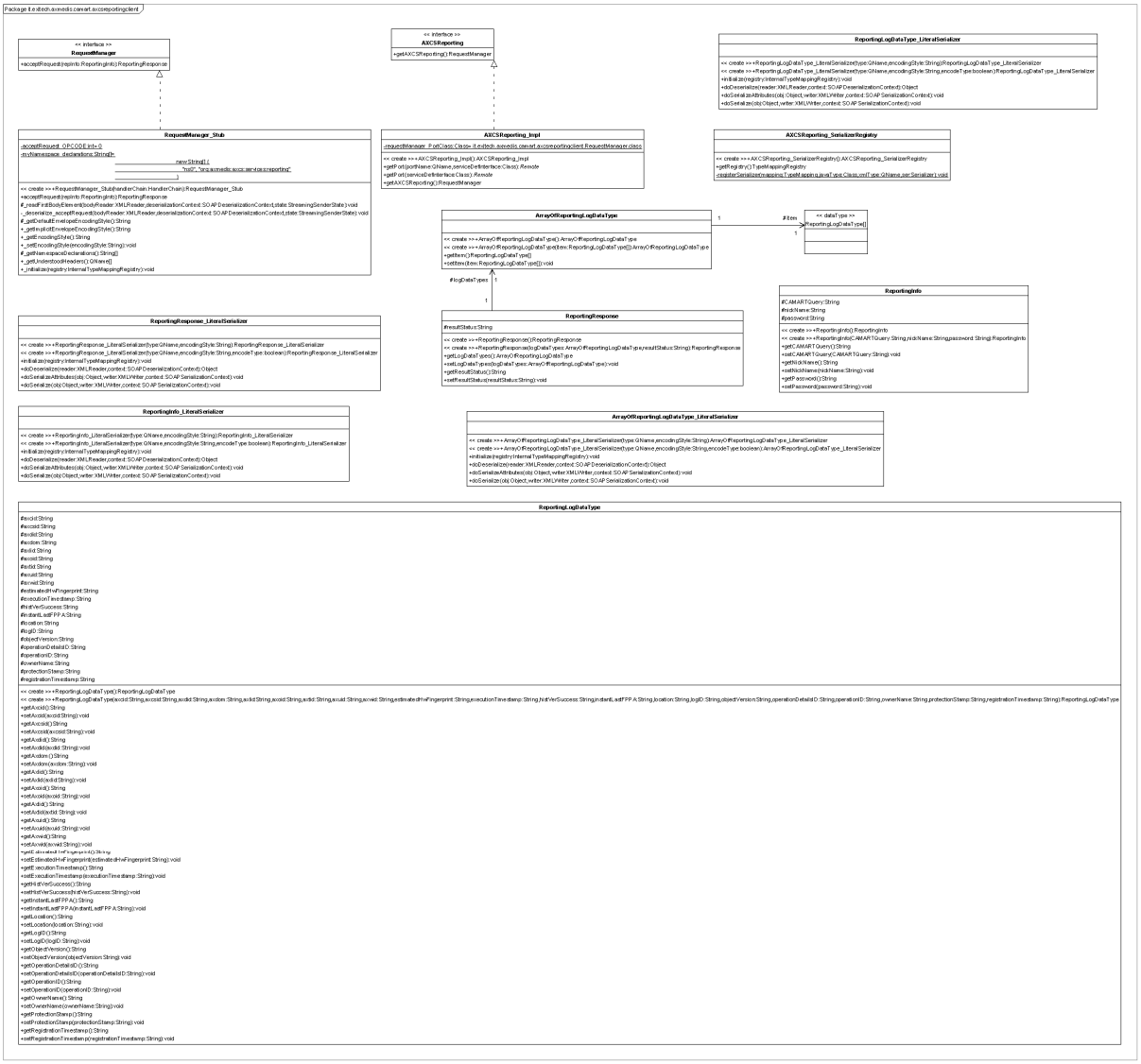
The web services implemented by AXCS/SuperAXCS are described in details in DE 3.1.2.2.13 while in the following a summary is reported for both the Statistical and the Reporting part.

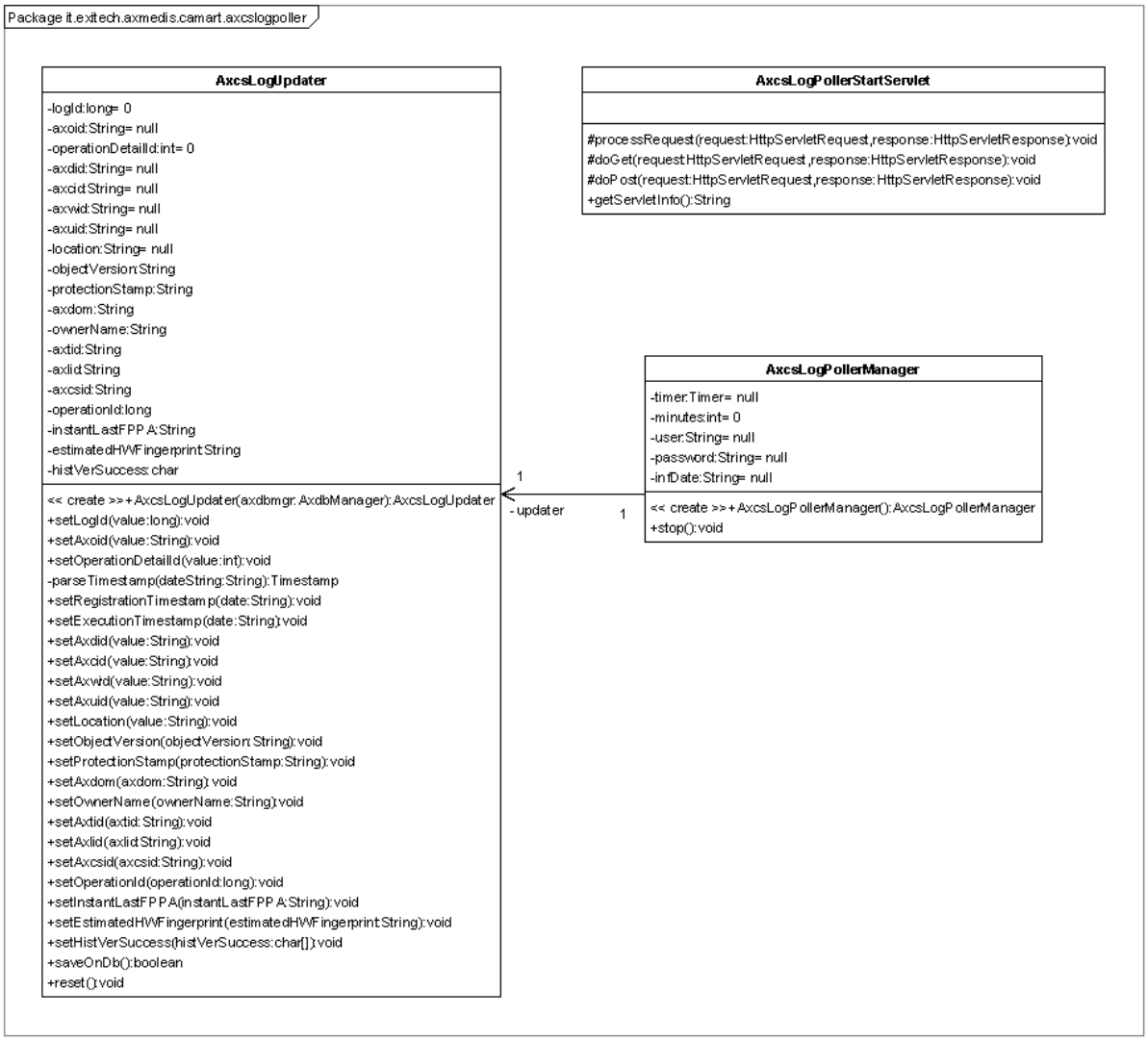
AXCSStatistics	
Method	***Description***
acceptRequest	This method, exposed by AXCS, will be invoked by CAMART to get all the logs that respects the characteristics in the CAMARTQuery parameter.

AXCSReporting	
Method	***Description***
acceptRequest	This method, exposed by AXCS, will be invoked by CAMART to get all the logs that respects the characteristics in the CAMARTQuery parameter.

4.2 Module Design in terms of Classes

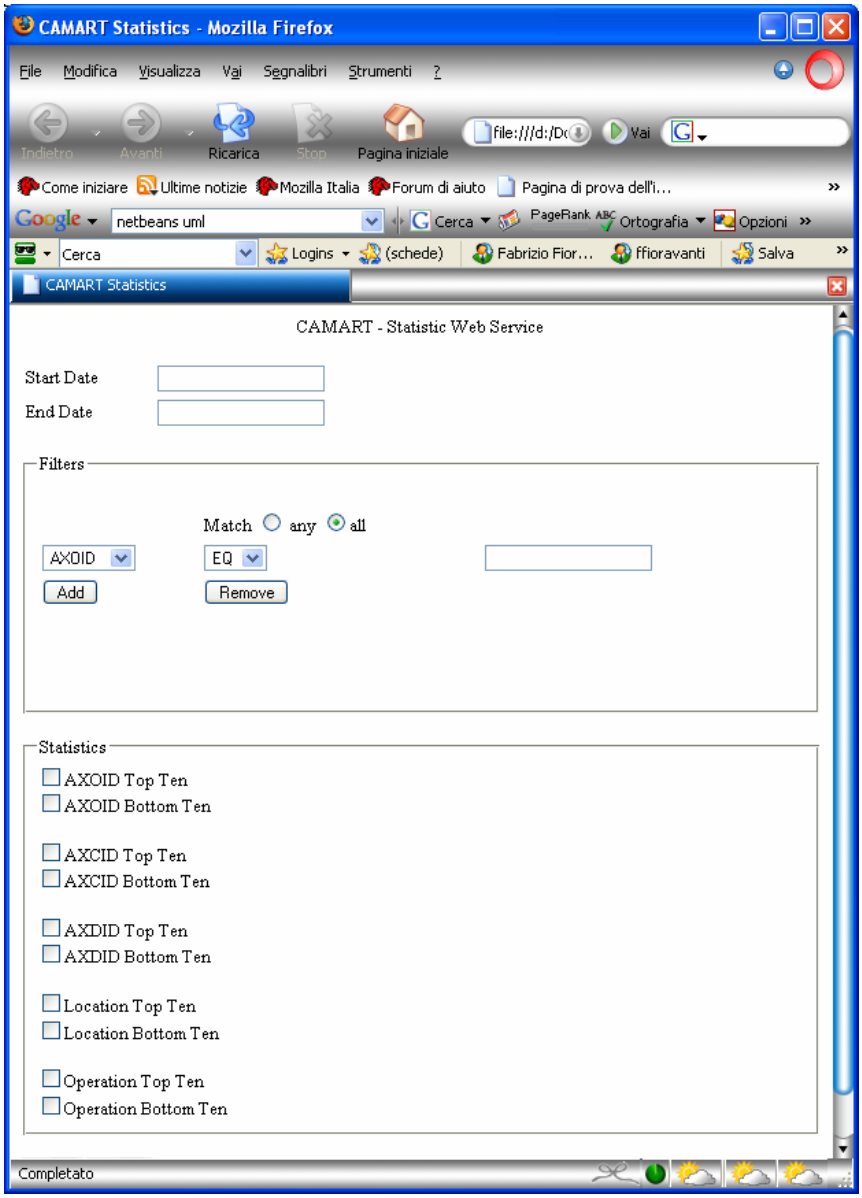
A detailed view of the module in terms of classes is reported in the following pictures:


Package it.eitech.axmedis.conwrt.axcreportingclient
<< interface >>
RequestManager
+acceptRequest(reqInfo:ReportingInfo):ReportingResponse
<< interface >>
AXCSReporting
+getAXCSReporting():RequestManager
ReportingLogDataType_LiteralSerializer
RequestManager_Stub
AXCSReporting_Impl
AXCSReporting_SerializerRegistry
ArrayOfReportingLogDataType
<< dataType >>
ReportingLogDataType[]
ReportingInfo
ReportingResponse
ReportingResponse_LiteralSerializer
ReportingInfo_LiteralSerializer
ArrayOfReportingLogDataType_LiteralSerializer
ReportingLogDataType



4.3 User interface description

CAMART in its general view is a back-office service while the statistical extraction of log has a web user interface to customize the search and the statistics to be displayed and or saved.



This interface allows extracting toward tools that are not the CMS (for which the operation is performed by Administrative Information Integrator) of statistical data for a further analysis. This operation is not bound to the log extraction from AXCS that is performed always in an automatic manner.

4.4 Technical and Installation information

To deploy the Camart Service, a PC with Java2 1.5.0 Runtime environment and Apache Tomcat 5.5 is required. The service will be distributed as a WAR file to be deployed in %TOMCAT_ROOT%/webapps.
If the MySQL which contains AxDB is not installed in the same PC with the default port, modifications in context file are needed after deployment. This XML file is located in %TOMCAT_ROOT%/ conf/Catalina/localhost with the same name of the WAR file. Modifications should be done only in the following attributes (XPath syntax is used):

- `/Context/Resource`
  - `@url` the URL pointing to MySQL instance (example: jdbc:mysql://my-host/axdb?jdbcCompliantTruncation=false)

- `@username` the user ID that has remote access to AxDB privileges granted
- `@password` the password associated to previous user ID

4.5 Draft User Manual

Regarding the interface for statistics, the user manual is very simple since we have 3 sections that are:

- Dates: in this first part the range of date/time we want to analyze must be imposed. If the final date is left empty it is assumed the current date/time
- Filters: this section allows to filters the data for a variety of parameters such as AXOID, AXCID, AXDID, location etc, in order to have for example statistics only for a certain location (say Italy and for a certain creator)
- Statistics: The user can ask to the system to have high level statistics such as top ten and bottom ten based on different parameters

4.6 Integration and compilation issues

Since Java and Apache Tomcat technologies, combined with W3C standard protocol (HTTP) and mark-up language (XHTML), were used, there is not any known issue related to interoperability and integration in different context.

4.7 Configuration Parameters

Config parameter	Possible values
connectionPool: the endpoint to connection pool linked to AxDB	java:comp/env/jdbc/AxDBResource
maxPushRequests: maximum number of concurrent active push request	From 32 to Java Integer.MAX_VALUE
maxConcurrentUsers: maximum number of concurrently connected users to AII web portal	From 10 to Java Integer.MAX_VALUE

4.8 Errors reported and that may occur

Error code	Description and rationales
AccountException	Happens if user has not the rights to access the service, or if the HTTP session was terminated by inactivity time-out. AIIServlet should gently-fail upon this.

5 Module or Executable Tool Administrative Information Integrator (AII)

Module/Tool Profile		
AII		
Responsible Name	Villoresi	
Responsible Partner	EXITECH	
Status (proposed/approved)	approved	
Implemented/not implemented	implemented	
Status of the implementation	under development	
Executable or Library/module (Support)	Web Application	
Single Thread or Multithread	Multithread	
Language of Development	JAVA	
Platforms supported	All supported by JAVA	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/Framework/source/camart/	
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/repos/Framework/bin/camart/	
Reference to the AXFW location of the demonstrator executable tool for public download		
Address for accessing to WebServices if any, add accession information (user and Passwd) if any		
Test cases (present/absent)	absent	
Test cases location	none	
Usage of the AXMEDIS configuration manager (yes/no)	no	
Usage of the AXMEDIS Error Manager (yes/no)	no	
Major Problems not solved	-- --	
Major pending requirements	-- --	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format (protected or not, etc.)
Formats Used	Shared with	format name or reference to a section

Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not

5.1 General Description of the Module

Administrative Information integrator is a critical part of the AXMEDIS system since it is the real bridge between the AXMEDIS world and the world of company's CMS and CRM for taking in account administrative and legal aspects (such reclaim for payment not done and so on).
This component has also a double face since it can operate in a dual manner: used for polling information from AXMEDIS system when needed by distributor for example, or user for pushing information in the CMS as soon as they are available for example in the case of collecting societies.
The operating mode is determined by accounting people during the installation/configuration of the system when it will be established whose fields have to be exported from the DB to the CMS and the frequency of exporting. When a frequency is set, the Administrative Information Integrator will work in push mode, pushing information in the CMS import area, otherwise it operates in polling mode by starting the update in the CMS by a link to a web page.

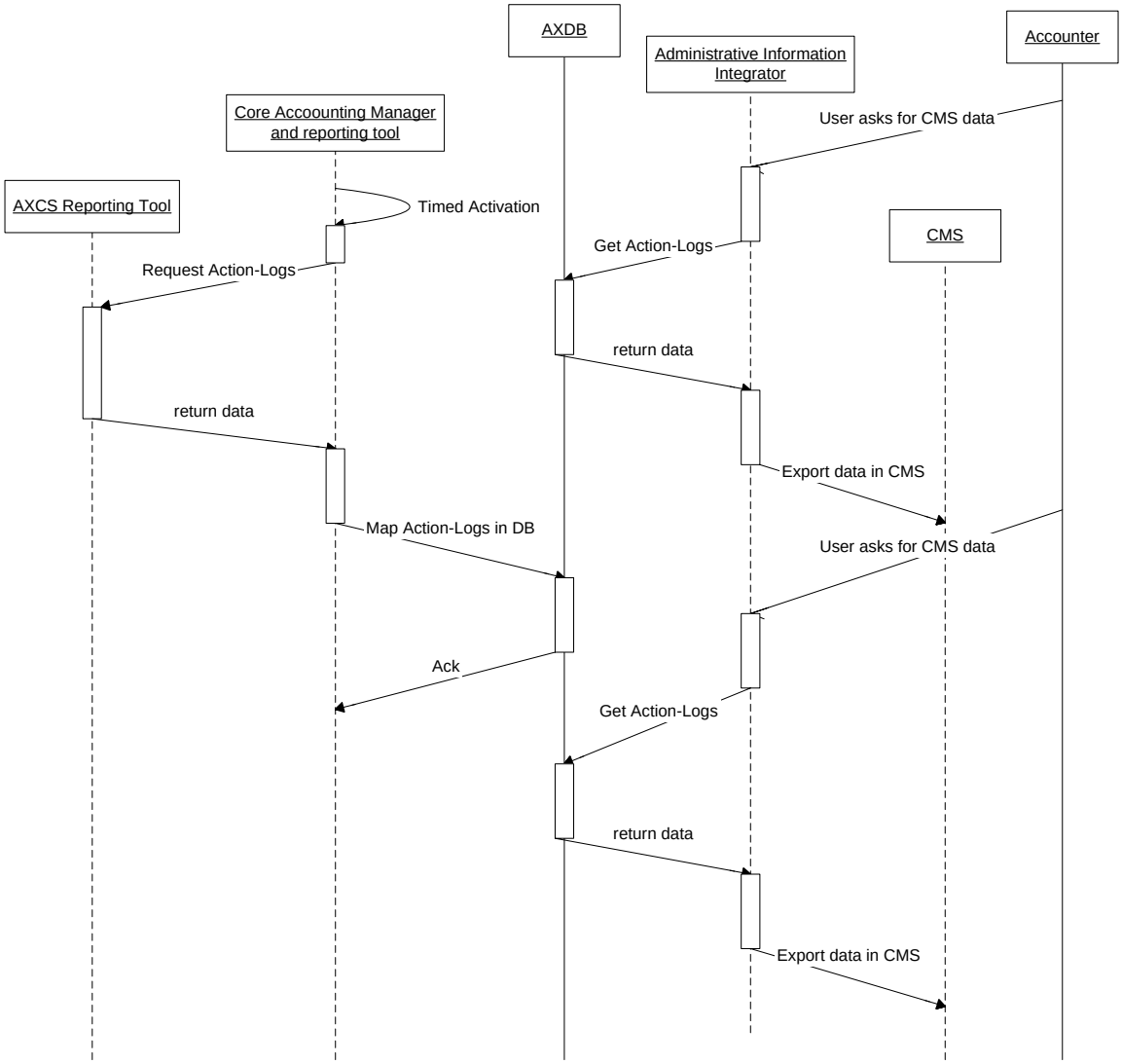
The principal specification arises, as always, from requirements. Administrative Information Integrator has to:

- ***interface with different CMS technologies***: this means that administrative information integrator has to export its reports in a portable format such as an XML that will be defined in the detailed specification. The generated XML can then be parsed and transformed by the way of standard mechanisms such as XSL transformation;
- ***store administrative information into the Content Provider database***: this operation will be possible in two ways, or by preparing a file to be put in the import area on the content provider database or by interfacing with the services offered by the CMS of the Content provider by standard mechanisms such as web services or other remote interface available to the Administrative Information Integrator;

- ***communicate with the AXDB to get administrative information related to a specific Content Provider***: this is the minimum security requirement that have to be established in order to be sure to distribute information to entitled persons only.
- ***guarantee privacy of sensitive data via protection mechanisms***: apart from what has already been stated, if a network connection is necessary to transfer data and the CMS of the Content Provider permits a connection on a secure channel, then the data will be sent encrypted.

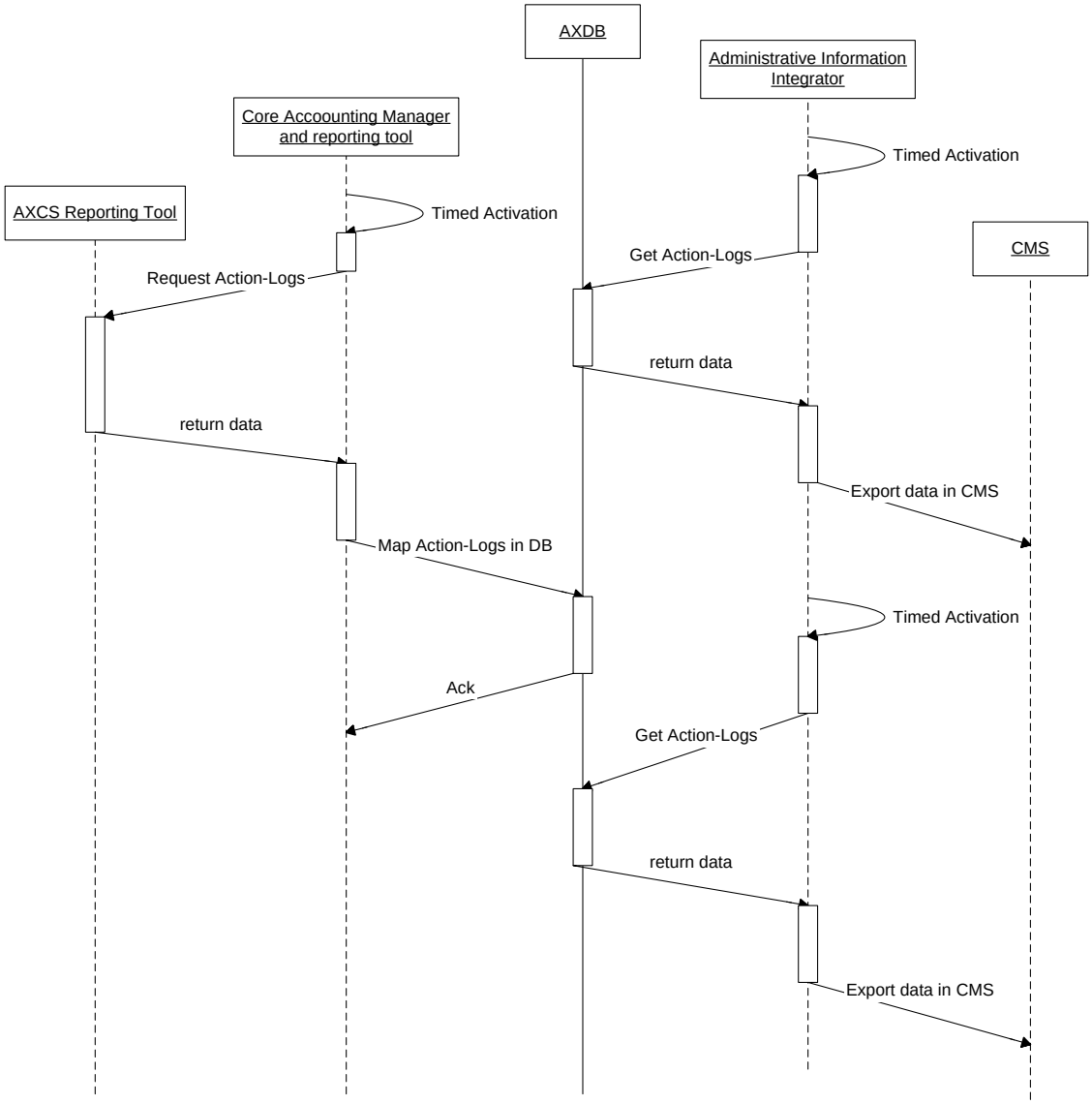
5.1.1 Administrative Information Integrator in polling mode

This is the operational mode when an automatic update of the data in the CMS is not set. In this mode the Accountant connects to a web page and issue a simple command to the Administrative Information Integrator that visualizes in the web page or exports in the CMS Import area the information that have not been already exported according to the format defined during the installation and configuration process.
The process that happens in the Accounting Area when the Administrative Information Integrator is in polling mode can be modeled by the following diagram.

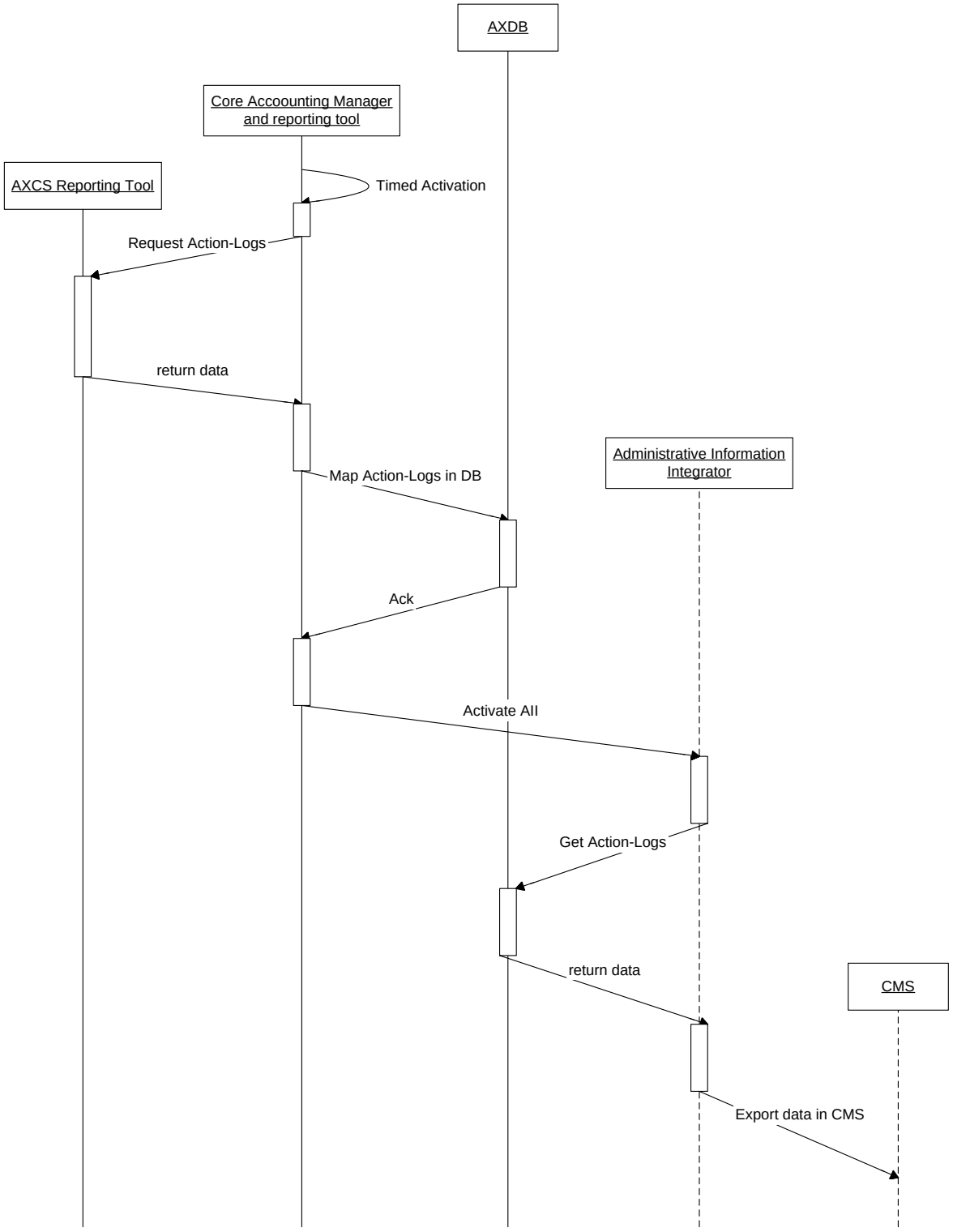


5.1.2 Administrative Information Integrator in push mode

All the times that during the configuration of the Administrative Information Integrator the timed insertion of administrative information has been selected, the Administrative Information Integrator is enabled to put in the import area of the CMS all the information that the CMS accountant has selected. The updating of the CMS should be also synchronized by a trigger sent by Core Accounting Manager and reporting tool.
In the following diagram the timed activation is reported, by which the Administrative Information Integrator and the CAMART operates asynchronously on a timed basis.



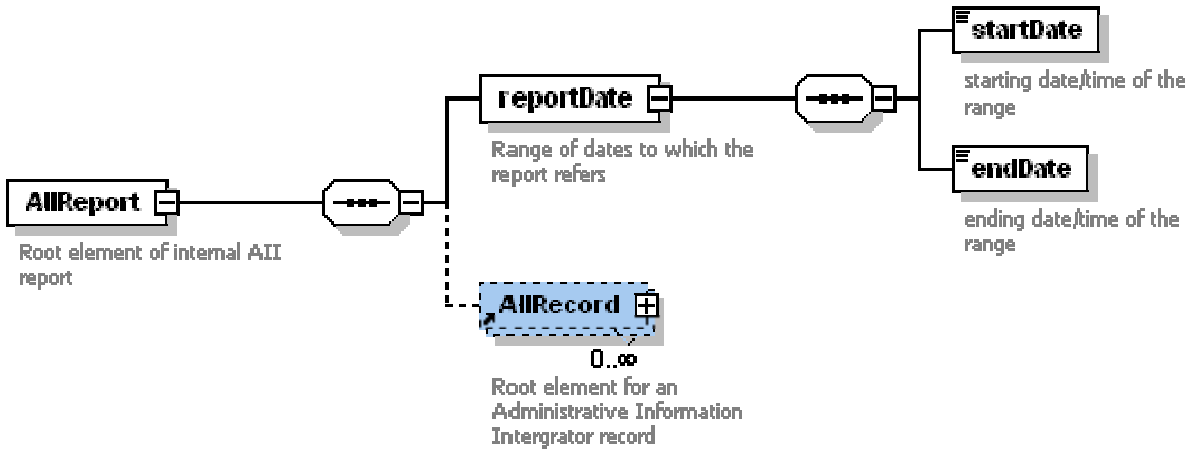
In the following diagram the timed activation is reported, by which the Administrative Information Integrator is triggered by the CAMART.

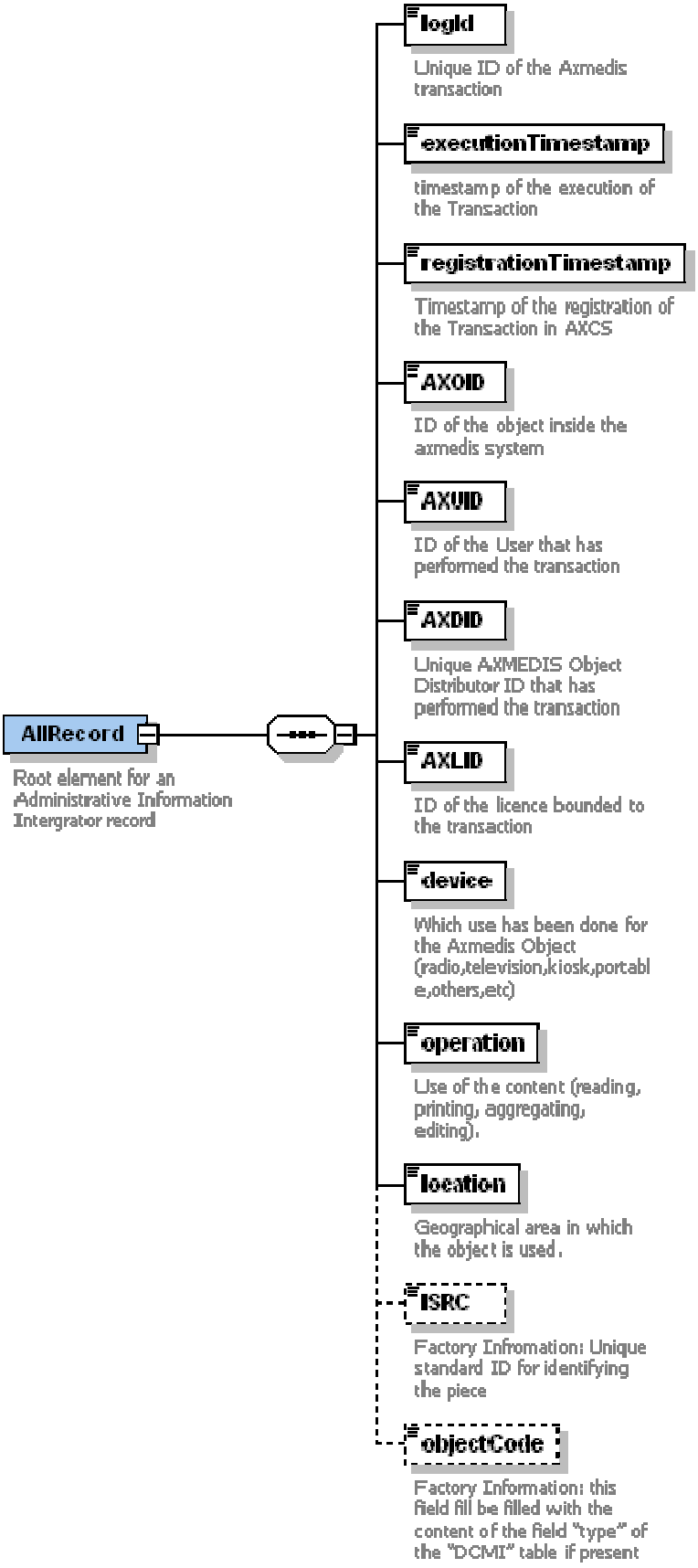


5.1.3 Administrative Information Integrator and CAMART integration

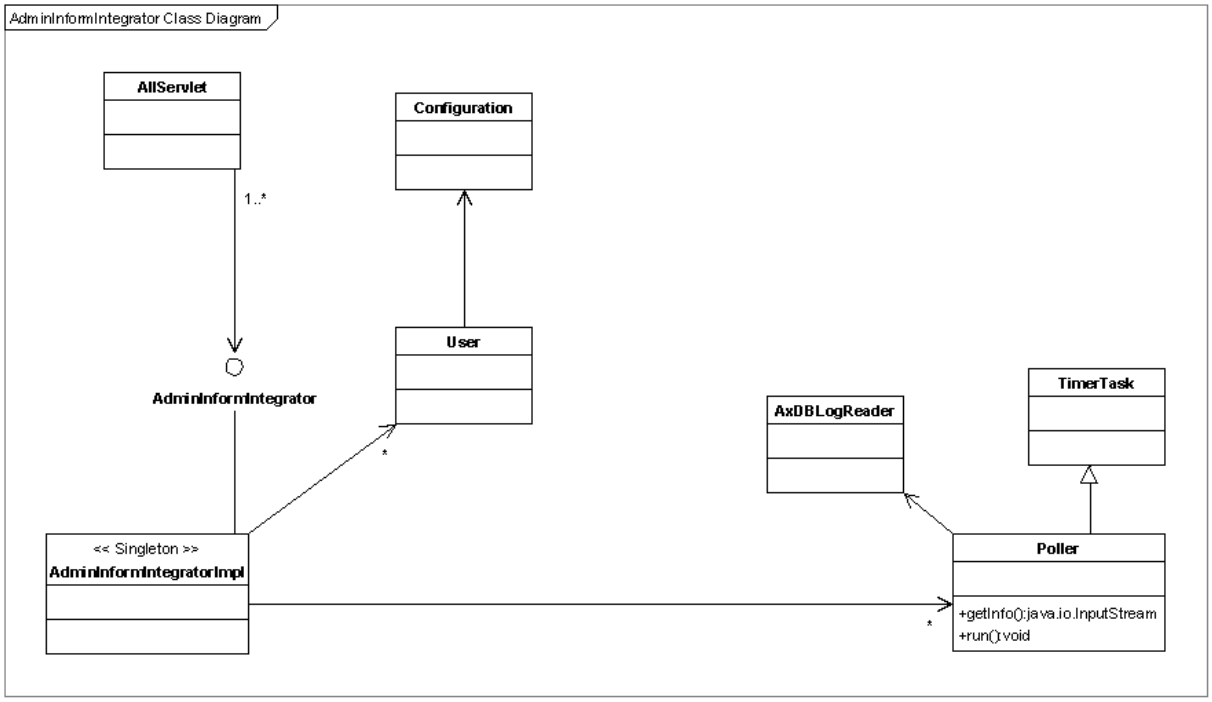
It is necessary to point out how the CAMART and AII are related in order to have the right tool doing the right work.
CAMART is the only interface toward the AXCS, while AII will read the Logs that are already in the AXDB; CAMART has the duty to put log on AXDB with a predefined scheduled period. CAMART will offer to the user an interface for getting logs that are on AXDB independently of the CMS exporting.
All the activities can be summarized in the scenario reported in section 2.6.

On the basis of the consideration exposed in DE 9.1.1, in this section an XML schema for the general format of the AII record file will be defined.
For each partner, on the basis of the requirements shown in DE 9.1.1, an XSL will be created to be applied to the XML in order to obtain a document in the user target format. This process can be repeated for each new company that joins AXMEDIS if the general XML file is not a supported format.
In the XML file, the Business Information will not appear and the factory operation will be considered optional, since it is not known if they can be provided or not.
The general schema is reported below in graphical form, while the XML of the XSD is reported in Section 7.




AllRecord
Root element for an Administrative Information Intergrator record
logId
Unique ID of the Axmedis transaction
executionTimestamp
timestamp of the execution of the Transaction
registrationTimestamp
Timestamp of the registration of the Transaction in AXCS
AXOID
ID of the object inside the axmedis system
AXUID
ID of the User that has performed the transaction
AXDID
Unique AXMEDIS Object Distributor ID that has performed the transaction
AXLID
ID of the licence bounded to the transaction
device
Which use has been done for the Axmedis Object (radio,television,kiosk,portable,others,etc)
operation
Use of the content (reading, printing, aggregating, editing).
location
Geographical area in which the object is used.
ISRC
Factory Infromation: Unique standard ID for identifying the piece
objectCode
Factory Information: this field fill be filled with the content of the field "type" of the "DCMI" table if present

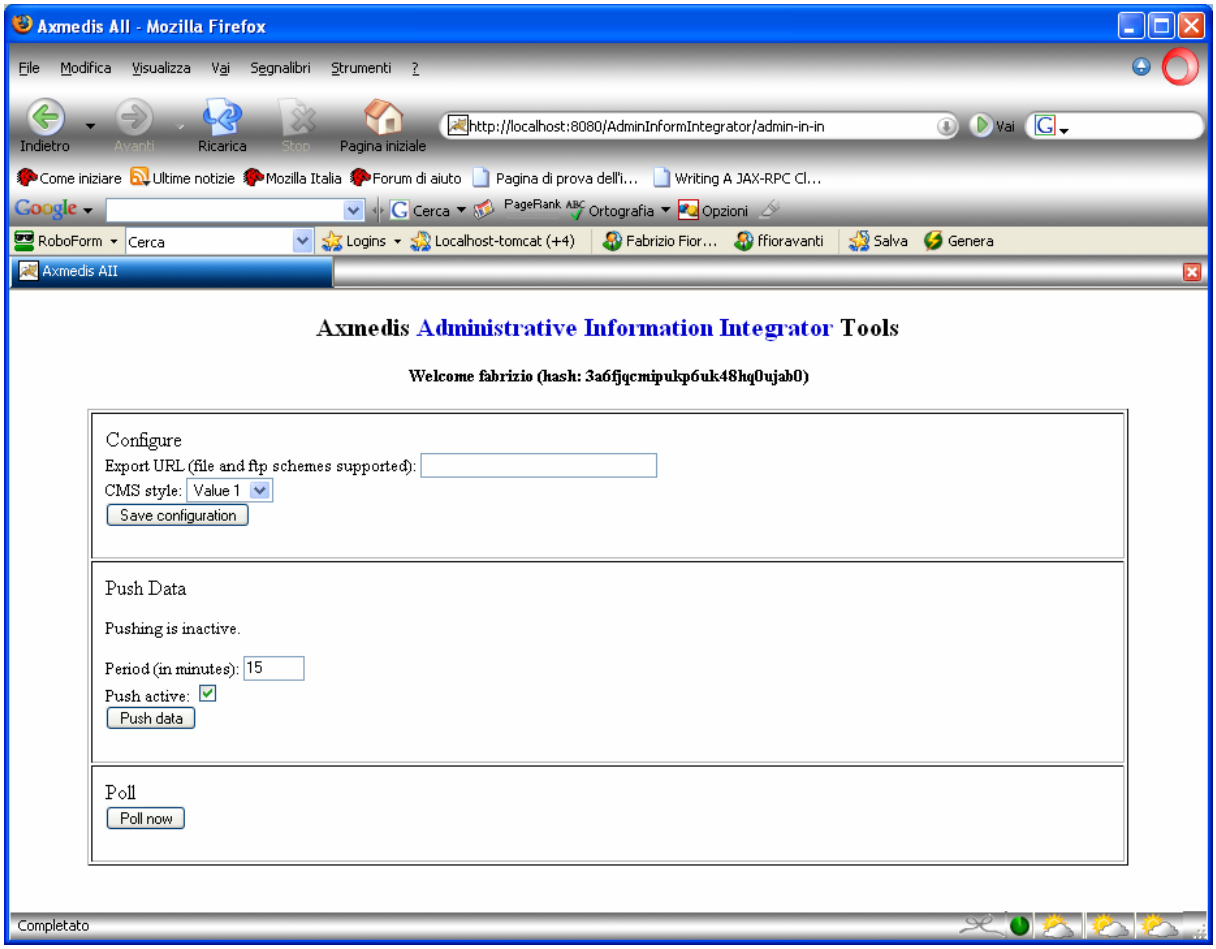
5.2 Module Design in terms of Classes



- AdminInformIntegrator: it is the main interface of the service, the business logic is implemented in AdminInformIntegratorImpl;
- AdminInformIntegratorImpl is the controller class of Administrative Information Integrator. It keeps track of the users and the various asynchronous Poller tasks;
- Poller is a class wrapping a polling task, which can be use synchronously or asynchronously. It works synchronously when Administrative Information Integrator is in polling mode, and asynchronously when the service is in pushing mode. In the last case the class is used as a TimerTask that polls data periodically and sends them back to controller class, which publish them at the exporting path selected by user;
- AxDBLogReader is the class in charge to read data from AxDB;
- User is the class that keeps track of all info about logged user, from user ID to configuration for exporting/formatting the AxDB results.

5.3 AdminInformIntegratorImplUser interface description

The user interface for the Administrative Information Integrator is able to configure the tool in the different modes (pushing or polling mode). In the case of pushing mode the frequency at which the push will be made can be set, while in polling mode, the file can be displayed or downloaded. In any case the formatting style can be selected among those present in the database. Refer to the Specification Part E for the support database needed by the Administrative Information Integrator.



In the case of pushing mode, a daemon will be created, that for each user enabled to get data from Administrative Information Integrator will automatically download in the configured path the fresh logs. System will take care or keeping in the database the timestamp of last download (both for polling and pushing) in order to have for each user a stored profile with the last logs downloaded.
This user interface is for configuring how the AII will interact with CMS in terms of periodic update of the file to be imported by the CMS, the style that have to be applied in order to a have multiplatform capability for different CMS. The timestamp of the extraction will be registered automatically from the system and there is no need for the user to take care of that. This information can be displayed in the previous web page.

5.4 Technical and Installation information

To deploy the Administrative Information Integrator service a PC with Java2 1.5.0 Runtime environment and Apache Tomcat 5.5 is required. The service will be distributed as a WAR file to be deployed in %TOMCAT_ROOT%/webapps.
If the MySQL which contains AxDB is not installed in the same PC with the default port, modifications in context file are needed after deployment. This XML file is located in %TOMCAT_ROOT%/ conf/Catalina/localhost with the same name of the WAR file. Modifications should be done only in the following attributes (XPath syntax is used):

- `/Context/Resource`
  - `@url` the URL pointing to MySQL instance (example: jdbc:mysql://my-host/axdb?jdbcCompliantTruncation=false)
  - `@username` the user ID that has remote access to AxDB privileges granted
  - `@password` the password associated to previous user ID

5.5 Draft User Manual

The configuration interface appears after the login that allows to map the user with the chosen configuration. If the user do not select any particular configuration, data will be returned in the neutral format specified in this document, while if the user selects one of the predefined CMS styles the data will be formatted accordingly.
Logs can be extracted in the current time instant with the "Poll Now" button or scheduled every n minutes by the Push section.
The Export URL in the first section is the ftp url where the data formatted according to the CMS style have to be put each time the pushing action is activated.

5.6 Integration and compilation issues

Since Java and Apache Tomcat technologies, combined with W3C standard protocol (HTTP) and mark-up language (XHTML), were used, there is not any known issue related to interoperability and integration in different context.

5.7 Configuration Parameters

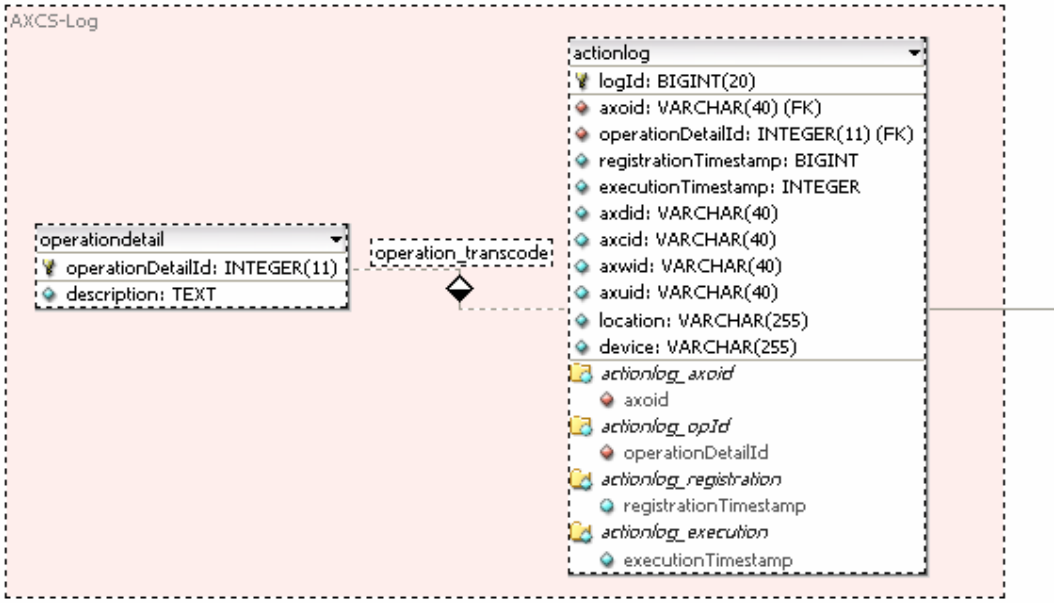
Config parameter	Possible values
connectionPool: the endpoint to connection pool linked to AxDB	java:comp/env/jdbc/AxDBResource
maxPushRequests: maximum number of concurrent active push request	From 32 to Java Integer.MAX_VALUE
maxConcurrentUsers: maximum number of concurrently connected users to AII web portal	From 10 to Java Integer.MAX_VALUE

5.8 Errors reported and that may occur

Error code	Description and rationales
AccountException	Happens if user has not the rights to access the service, or if the HTTP session was terminated by inactivity time-out. AIIServlet should gently-fail upon this.

6 Table description for database AXDB (CAMART and AII related tables)

In this section the AXDB table involved with CAMART and AII are described.
The table for storing the information provided by AXCS Reporting Web service is reported below with the description of the fields.



AXCS-Log

actionlog

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
logId	**BIGINT(20)**	PK	NN				AI
axoid	VARCHAR(40)		NN				
operationDetailId	INTEGER(11)		NN	UNSIGNED			
registrationTimestamp	BIGINT		NN				
executionTimestamp	INTEGER		NN	UNSIGNED			
axdid	VARCHAR(40)						
axcid	VARCHAR(40)						
axwid	VARCHAR(40)						
axuid	VARCHAR(40)		NN			general user of the system that has performed the operation: not an internal user	
location	VARCHAR(255)						
device	VARCHAR(255)						

IndexName	IndexType	Columns
PRIMARY	PRIMARY	logId
actionlog_axoid	Index	axoid
actionlog_opId	Index	operationDetailId

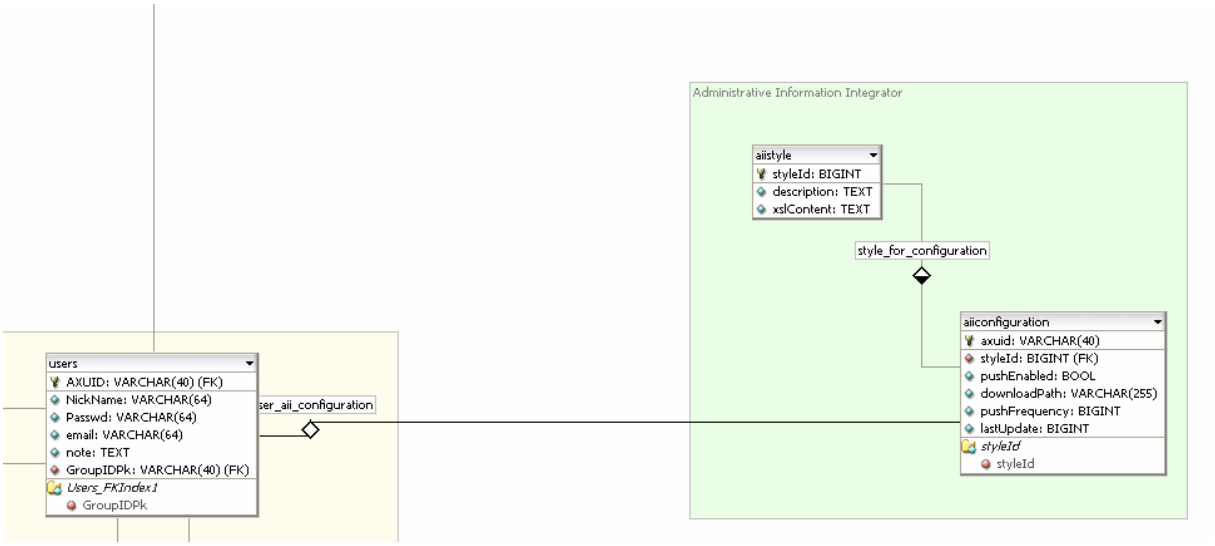
actionlog_registration	Index	registrationTimestamp
actionlog_execution	Index	executionTimestamp

operationdetail

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
operationDetailId	**INTEGER(11)**	PK	NN	UNSIGNED			AI
description	TEXT		NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	operationDetailId

The table for storing the information needed by AII below with the description of the fields.



Administrative Information Integrator

aiistyle

ColumnName	DataType	PrimaryKey	NotNull	Flags Default Value	Comment	AutoInc
styleId	**BIGINT**	PK	NN			AI
description	TEXT		NN			
xslContent	TEXT		NN			

IndexName	IndexType	Columns
PRIMARY	PRIMARY	styleId

aiiconfiguration

ColumnName	DataType	PrimaryKey	NotNull	Flags Default Value	Comment	AutoInc
axuid	**VARCHAR(40)**	PK	NN			
styleId	BIGINT		NN			
pushEnabled	BOOL		NN			

downloadPath	VARCHAR(255)		NN				
pushFrequency	BIGINT		NN				
lastUpdate	BIGINT		NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	axuid
styleId	Index	styleId

UserGroup

users

ColumnName	DataType	PrimaryKey	NotNull	Flags	Default Value	Comment	AutoInc
AXUID	**VARCHAR(40)**	PK	NN				
NickName	VARCHAR(64)		NN				
Passwd	VARCHAR(64)		NN				
email	VARCHAR(64)		NN				
note	TEXT						
GroupIDPk	VARCHAR(40)		NN				

IndexName	IndexType	Columns
PRIMARY	PRIMARY	AXUID
Users_FKIndex1	Index	GroupIDPk

7 Formal description of format AII Record

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
	<!-- version 1.1 (2-Mar-2006)
-->
	<xs:element name="AIIRecord">
		<xs:annotation>
			<xs:documentation>Root element for an Administrative Information Intergrator
record</xs:documentation>
		</xs:annotation>
		<xs:complexType>
			<xs:sequence>
				<xs:element name="logId" type="xs:string">
					<xs:annotation>
						<xs:documentation>Unique ID of the Axmedis
transaction</xs:documentation>
					</xs:annotation>
				</xs:element>
				<xs:element name="executionTimestamp" type="xs:dateTime">
					<xs:annotation>
						<xs:documentation>timestamp of the execution of the
Transaction</xs:documentation>
					</xs:annotation>
				</xs:element>
				<xs:element name="registrationTimestamp" type="xs:dateTime">
					<xs:annotation>
						<xs:documentation>Timestamp of the registration of the Transaction in
AXCS</xs:documentation>
					</xs:annotation>
				</xs:element>
				<xs:element name="AXOID">
					<xs:annotation>
						<xs:documentation>ID of the object inside the axmedis
system</xs:documentation>
					</xs:annotation>
					<xs:simpleType>
						<xs:restriction base="xs:string">
							<xs:maxLength value="40"/>
							<xs:whiteSpace value="collapse"/>
						</xs:restriction>
					</xs:simpleType>
				</xs:element>
				<xs:element name="AXUID">
					<xs:annotation>
						<xs:documentation>ID of the User that has performed the
transaction</xs:documentation>
					</xs:annotation>
					<xs:simpleType>
						<xs:restriction base="xs:string">
							<xs:maxLength value="40"/>
							<xs:whiteSpace value="collapse"/>
						</xs:restriction>
					</xs:simpleType>
				</xs:element>
				<xs:element name="AXDID">
					<xs:annotation>
						<xs:documentation>Unique AXMEDIS Object Distributor ID that has
performed the transaction</xs:documentation>
					</xs:annotation>
					<xs:simpleType>
						<xs:restriction base="xs:string">
							<xs:maxLength value="40"/>
							<xs:whiteSpace value="collapse"/>
						</xs:restriction>
					</xs:simpleType>
```

```
                                        </xs:element>
                                        <xs:element name="AXLID">
                                                <xs:annotation>
                                                        <xs:documentation>ID of the licence bounded to the
transaction</xs:documentation>
                                                </xs:annotation>
                                                <xs:simpleType>
                                                        <xs:restriction base="xs:string">
                                                                <xs:maxLength value="40"/>
                                                                <xs:whiteSpace value="collapse"/>
                                                        </xs:restriction>
                                                </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="device" type="xs:string">
                                                <xs:annotation>
                                                        <xs:documentation>Which use has been done for the Axmedis Object
(radio,television,kiosk,portable,others,etc)</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="operation" type="xs:string">
                                                <xs:annotation>
                                                        <xs:documentation>Use of the content (reading, printing, aggregating,
editing).</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="location" type="xs:string">
                                                <xs:annotation>
                                                        <xs:documentation>Geographical area in which the object is
used.</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="ISRC" type="xs:string" minOccurs="0">
                                                <xs:annotation>
                                                        <xs:documentation>Factory Infromation: Unique standard ID for
identifying the piece</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="objectCode" type="xs:string" minOccurs="0">
                                                <xs:annotation>
                                                        <xs:documentation>Factory Information: this field fill be filled with the
content of the field “type” of the “DCMI” table if present </xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
                <xs:element name="AIIReport">
                        <xs:annotation>
                                <xs:documentation>Root element of internal AII report</xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element name="reportDate">
                                                <xs:annotation>
                                                        <xs:documentation>Range of dates to which the report
refers</xs:documentation>
                                                </xs:annotation>
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element name="startDate" type="xs:dateTime">
                                                                        <xs:annotation>
                                                                                <xs:documentation>starting date/time of the
range</xs:documentation>
                                                                        </xs:annotation>
                                                                </xs:element>
                                                                <xs:element name="endDate" type="xs:dateTime">
                                                                        <xs:annotation>
```

```
                                                                    <xs:documentation>ending date/time of the
range</xs:documentation>
                                                                </xs:annotation>
                                                            </xs:element>
                                                        </xs:sequence>
                                                    </xs:complexType>
                                                </xs:element>
                                                <xs:element ref="AIIRecord" minOccurs="0" maxOccurs="unbounded"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
</xs:schema>
```

8 Formal description of format for statistics record

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
        <!-- version 1.0 (30-Mar-2006)
-->
        <xs:element name="CamartRecord">
                <xs:annotation>
                        <xs:documentation>Root element for a Camart record</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="AXCSID" type="xs:string">
                                        <xs:annotation>
                                                <xs:documentation>log ID</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="executionTimestamp" type="xs:dateTime">
                                        <xs:annotation>
                                                <xs:documentation>timestamp of the execution of the
Transaction</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="registrationTimestamp" type="xs:dateTime">
                                        <xs:annotation>
                                                <xs:documentation>Timestamp of the registration of the Transaction in
AXCS</xs:documentation>
                                        </xs:annotation>
                                </xs:element>
                                <xs:element name="AXOID">
                                        <xs:annotation>
                                                <xs:documentation>ID of the object inside the axmedis
system</xs:documentation>
                                        </xs:annotation>
                                        <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                        <xs:maxLength value="40"/>
                                                        <xs:whiteSpace value="collapse"/>
                                                </xs:restriction>
                                        </xs:simpleType>
                                </xs:element>
                                <xs:element name="AXDID">
                                        <xs:annotation>
                                                <xs:documentation>Unique AXMEDIS Object Distributor ID that has
performed the transaction</xs:documentation>
                                        </xs:annotation>
                                        <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                        <xs:maxLength value="40"/>
                                                        <xs:whiteSpace value="collapse"/>
                                                </xs:restriction>
                                        </xs:simpleType>
```

```
                                        </xs:element>
                                        <xs:element name="AXCID" type="xs:string">
                                                <xs:annotation>
                                                        <xs:documentation>Creator Id</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="operation" type="xs:string">
                                                <xs:annotation>
                                                        <xs:documentation>Use of the content (reading, printing, aggregating,
editing).</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="location" type="xs:string">
                                                <xs:annotation>
                                                        <xs:documentation>Geographical area in which the object is
used.</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="ISRC" type="xs:string" minOccurs="0">
                                                <xs:annotation>
                                                        <xs:documentation>Factory Infromation: Unique standard ID for
identifying the piece</xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                        <xs:element name="objectCode" type="xs:string" minOccurs="0">
                                                <xs:annotation>
                                                        <xs:documentation>Factory Information: this field fill be filled with the
content of the field “type” of the “DCMI” table if present </xs:documentation>
                                                </xs:annotation>
                                        </xs:element>
                                </xs:sequence>
                        </xs:complexType>
                </xs:element>
                <xs:element name="CamartReport">
                        <xs:annotation>
                                <xs:documentation>Root element of internal Camart report</xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="CamartRecord" minOccurs="0" maxOccurs="unbounded"/>
                                        <xs:element name="Statistics" minOccurs="0" maxOccurs="unbounded">
                                                <xs:complexType>
                                                        <xs:sequence>
                                                                <xs:element name="AXOID" minOccurs="0"
maxOccurs="unbounded">
                                                                        <xs:annotation>
                                                                                <xs:documentation>ID of the object inside
the axmedis system</xs:documentation>
                                                                        </xs:annotation>
                                                                        <xs:complexType>
                                                                                <xs:simpleContent>
                                                                                        <xs:extension base="xs:string">
                                                                                                <xs:attribute
name="count" type="xs:int" use="required"/>
                                                                                        </xs:extension>
                                                                                </xs:simpleContent>
                                                                        </xs:complexType>
                                                                </xs:element>
                                                        </xs:sequence>
                                                        <xs:attribute name="type" use="required">
                                                                <xs:simpleType>
                                                                        <xs:restriction base="xs:string">
                                                                                <xs:enumeration value="TopChart"/>
                                                                                <xs:enumeration value="BottomChart"/>
                                                                        </xs:restriction>
                                                                </xs:simpleType>
                                                        </xs:attribute>
                                                        <xs:attribute name="cardinality" type="xs:int"/>
```

```
                                                <xs:attribute name="restrictBy" type="xs:string"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
</xs:schema>
```

9 Bibliography (mandatory)

- DE 2.1.1 Use Case and Test case
- DE 3.1.2 AXMEDIS Framework Specification
- DE 9.1.1 Specification of CMS integration and feedback

10 Glossary (mandatory)

AII: Administrative Information Integrator
CAMART: Core Accounting Manager and Reporting Tool