Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE3.1.2.2.11
Specification of AXMEDIS
Programme and Publication Tools

Version: 1.6
Date: 13-04-2006
Responsible: UNIVLEEDS (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: DE3.1.2.2.11 Contractual Date of Delivery: M18 Actual Date of Delivery: 16/04/2006 Title of Deliverable: Specification of AXMEDIS Programme and Publication Tools Work-Package contributing to the Deliverable: WP1, WP2, WP3, WP4, WP5, WP11 Task contributing to the Deliverable: T5.4.5 Nature of the Deliverable: Report Author(s): UNIVLEEDS

Abstract: this part includes the specification of components, formats, databases and protocol related to the AXMEDIS Framework area Programme and Publication (P&P) including the P&P Model, P&P Editor, P&P Engine and P&P Engine Monitor.

Keyword List: Programme, Publication, Distribution, Channel, Content, Formatting, On-Demand, Workflow

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 licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.
   2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.

3. **TERM AND TERMINATION**

   1. Granted Licence shall commence on Acceptance Date.
   2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.
   3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.
   4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.
   5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence 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 licence 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-fulfilment 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 Licence 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 Overview of this Part of the AXMEDIS Specification Document

This document, DE3-1-2-2-11 of the above list, is concerned with the Programme and Publication (P&P) area which includes:

- P&P Editor: A GUI for users to create, edit save, and activate P&P Programmes for publication.
- P&P Engine: To process and manage distribution of P&P Programmes based upon the distribution rules
- P&P Engine Monitor: A GUI front end to monitor the P&P Engine
- P&P Repository for P&P Programmes

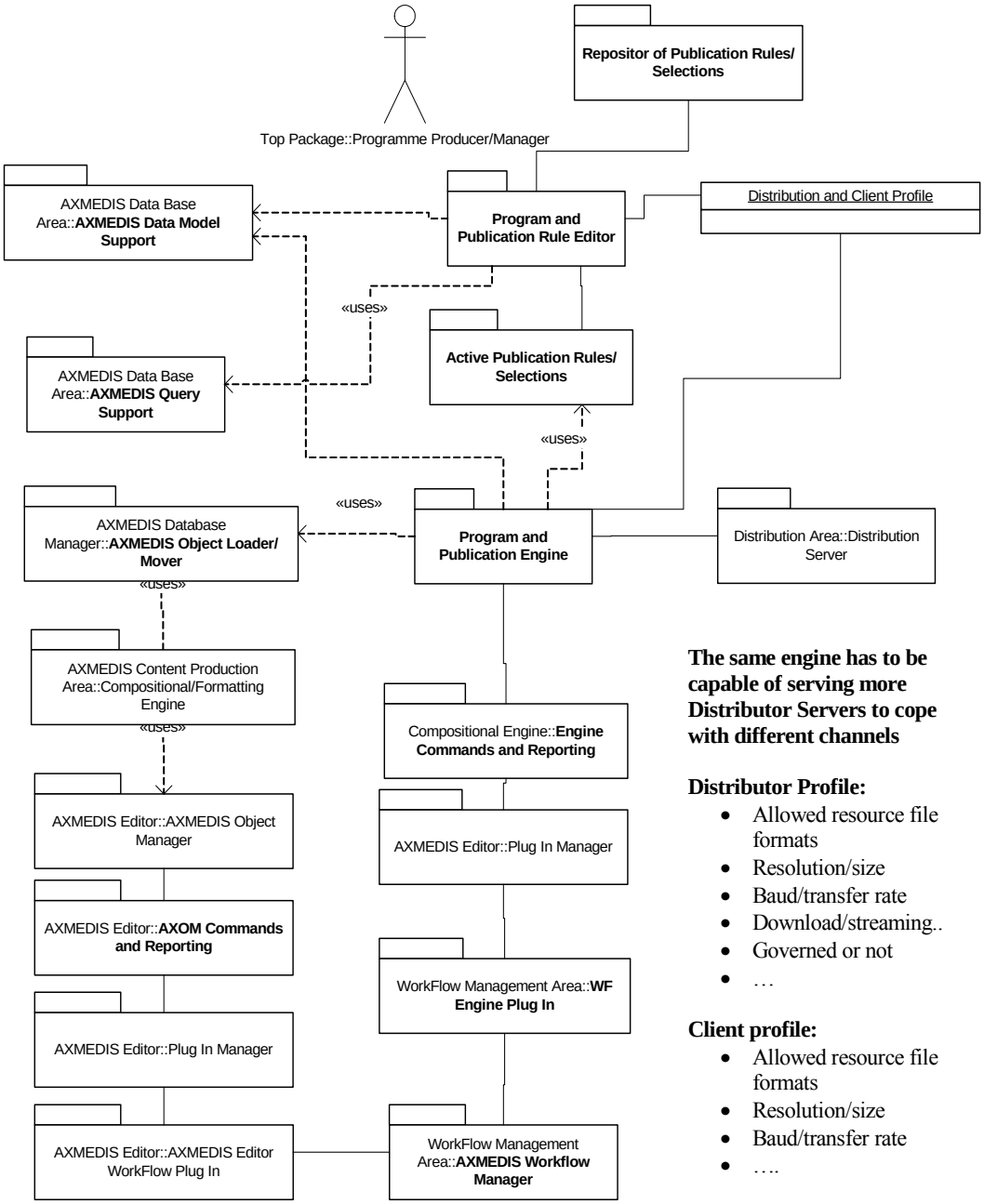


Figure: Programme and Publication Area

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
P&P Model	The P&P Model utilises XERCES and	AXMEDIS common and XERCES v2-6 libraries
P&P Editor	The P&P Editor is an executable GUI for creating, editing, saving, testing and activating P&P Programme for publication. Other AXMEDIS Modules include ▪ P&P Model ▪ AXMEDIS Configuration Manager for application ▪ AXMEDIS Plugin Manager for Workflow ▪ AXMEDIS Query Support	P&P Exploits the P&P Rule Model which utilises AXMEDIS common and XERCES v2-6 libraries (see above)
P&P Engine	The P&P Engine ▪ P&P Model ▪ AXMEDIS Configuration Manager ▪ AXMEDIS Plugin Manager ▪ AXMEDIS AXOM for loading AXMEDIS objects	AXMEDIS Workflow with Webservices AXDB with Webservices
P&P Engine Monitor	The P&P Engine Monitor is used to monitor P&P Programmes currently	

1.3 List of Formats Specified in this document

A format can be (i) an XML content file for modelling 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
axpnprule.xsd	Based on the Rule_AXMEDIS used is the content processing rules (AXCP). The Programme is divided into three rule parts. 1. The header section including the programme Rule ID (AXRID), 2. The Schedule section specifying when the P&P Programme is to run 3. The Distribution section where the parameters are specified for the what (which object), how (distribution channel, where (the URI) and when (time the object is to be distributed).	XERCES v2-6 and the AXMEDIS common libraries

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
AXMEDIS Database	The AXDB is accessed using the AXOM loader and saver tools	Webservices

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
WebServices	Webservice calls are used to pass on the workflow request coming from the request adaptor and converting them into a generalised function call towards AXMEDIS Framework using Webservices	Request Gateway – Master AXMEDIS Tools – Slave (P&P Editor and P&P Engine)	Webservice
WebServices	Webservice calls are also used to pass on the request coming from the AXMEDIS Tools towards the workflow. The calls are received by the response gateways.	Response Gateway – Slave AXMEDIS Tools – Master (P&P Editor and P&P Engine)	Webservice
Socket	TCP/IP	P&P Engine – Master P&P Editor - Slave	TCP/IP

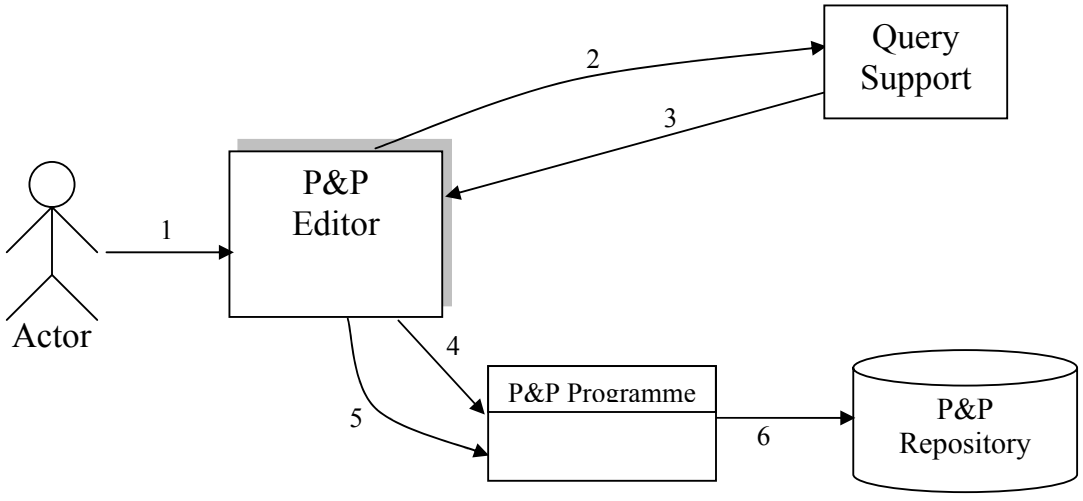
2 Use Cases and scenarios

The use cases for the P&P Area include:

- Use Case 10.1: Programme and Publication Rules Production
- Use Case 10.2: Programme and Publication Rules Editing
- Use Case 10.3: Activation of Programme and Publication Rules
- Use Case 10.4: Launch of Programme and Publication Rules from Workflow
- Use Case 10.5 Trial Pre-activation of Programme and Publication Rules

2.1 Use Case 11.1: Programme and Publication Programme Production (UNIVLEEDS)

This section describes how the P&P Programme are produced by a Programme Manager (the Actor) to create a new programme for the publication of AXMEDIS Digital Objects over multiple distribution channels such as PDA, PC and satellite.



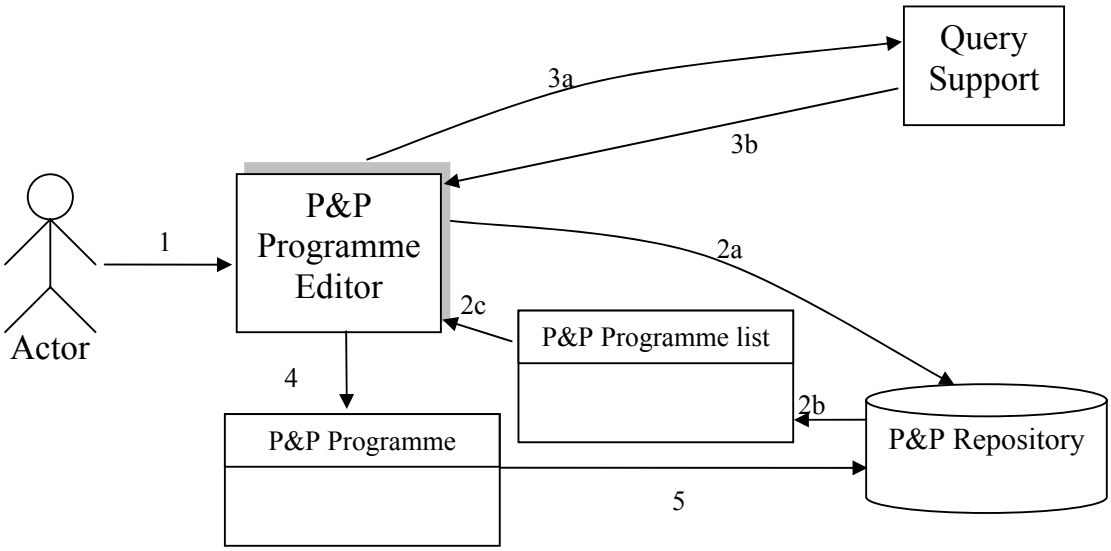
1. Actor initiates GUI in the Programme and Publication Editor

2. The Actor submits queries to Query Support for a list of AXMEDIS objects
3. Query Support returns a list of AXOIDs (see UCs in section 3.1.2)
4. The Programme and Publication Editor GUI allows the actor to select AXOIDs The Actor specifies what (AXOID), where (channel), when (schedule), etc.
5. The Programme is saved in the Programme and Publication Repository which can be re-used

By default the Programme is "inactive" at the end of the programme production, until the Actor activated/published it. The actor is required to use the Query Support interface to select AXMEDIS objects.

2.2 Use Case 11.2: Programme and Publication Programme Editing (UNIVLEEDS)

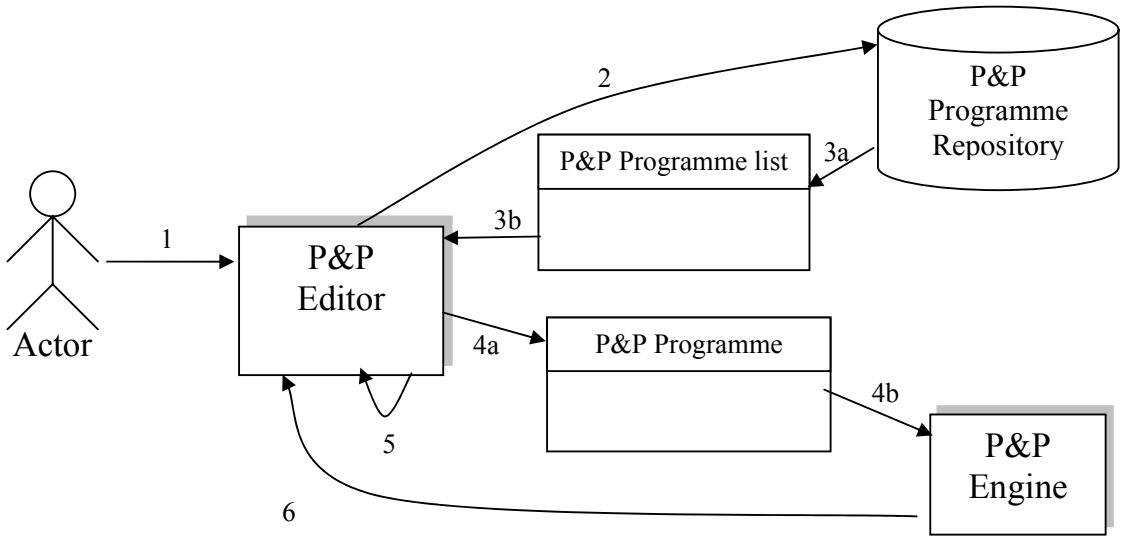
This section describes how the Programme and Publication Programme are manipulated by a programme manager to edit the various distribution and schedule parameters specified within the P&P Programme.



1. Actor initiates the P&P GUI
2. (2a) the user browse the existing programmes in the P&P Repository and (2b) selects P&P Programme for a Programme list for (2c) load the selected P&P Programme
3. (3a) Actor queries for additional objects using Query Support (3b).Query Support returns a list of AXOID
4. Actor edits the Programme (e.g. time, channel)
5. The revised P&P Programme is saved to the Programme and Publication Repository as a new Programme or overwriting the original Programme

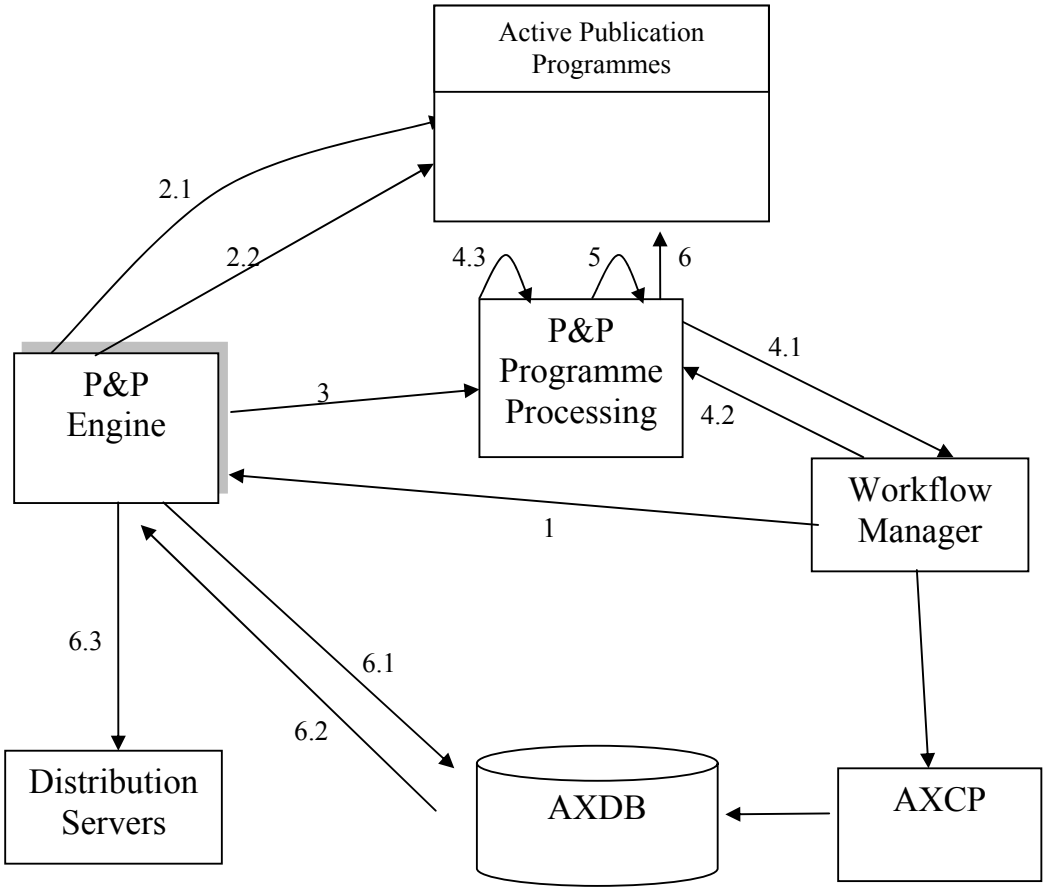
2.3 Use Case 11.3: Activation of Programme and Publication Programme (UNIVLEEDS)

This section describes how the Programme and Publication Programmes are activated for the publication of AXMEDIS Objects. In this use case the result is the P&P Programme is sent to the P&P Engine which processes the parameters and objects to deliver correctly formatted AXMEDIS Object for the Actor's specified distribution channel.



1. The actor uses the Programme and Publication Editor GUI
2. If the programme has not been loaded, the user can select and load the programme, for final checking
3. The programme/schedule is returned from the repository
4. A GUI to allow the user to activate/publish the programme (4a) to the P&P Engine (4b)
5. The P&P Editor checks and reports an invalid programme
6. A confirmation on the success of the publication

2.4 Use Case 11.4: Launch of Programme and Publication Programme from Workflow (IRC, UNIVLEEDS)



In this section, the P&P engine is running monitoring the system clock to ensure that one or more scheduled programmes are delivered in time for the actual consumption

1. Workflow sends a P&P programme to be activated
2. The Engine checks if the P&P Programme is already running
   - 2.1. Engine searches active programmes
   - 2.2. The matching programme is removed if found
3. check source and target format
4. if profiles mismatch
   - 4.1. send request to AXCP via Workflow
   - 4.2. receive new object ID from workflow
   - 4.3. update the programme with new object ID
5. calculate distribution time based on distribution schedule and distribution server profile
6. programme is scheduled for distribution
7. at their scheduled distribution time
   - 7.1. request Objects from AXDB
   - 7.2. Receive objects (from AXDB according to Scenarios v3.6)
   - 7.3. Send to Distribution Servers

In this use case there is a variation to the above scenario for content production On-Demand. The On-Demand scenario is sent via workflow and in this case the content is produced only for the requesting

distribution server/terminal and to be processed and distributed immediately. In addition to "Activate" request, Workflow can send:

1. A "Kill" request. In this case, the Engine will remove the scheduled Programme
2. A "List" request. In this case, the Engine will return a list of scheduled programmes
3. A "Status" request. In this case, the Engine will return the status of the relevant scheduled programme.
4.

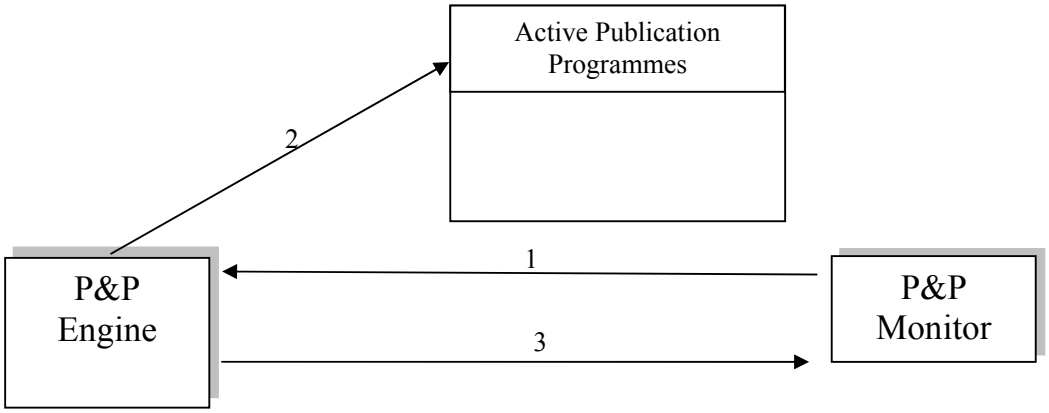
2.5 Use Case 11.5: Trial Pre-activation of Programme and Publication Programme (UNIVLEEDS)

This section describes how the P&P Programmes are pre-activated to simulate, test and be prepared.

1. Steps 1-4 of the UC11.3 (see section 2. 3 Activation of P&P Programme above)
2. The P&P Editor GUI to allow the user to activate/publish the Programme as a trial (quick trial or full trial)
3. A confirmation on the success of the trial-run when completed

There are two trials available, the Quick trial and the Full trial. A quick trial would complete each stage for publication without requiring the engines such as the formatting engine to actually format the object but simply acknowledge if it can format the object from the source object to a given target representation. The full trial completes a publication without final distribution.

2.6 Use Case 11.6: Monitoring of Programme and Publication Engine (UNIVLEEDS)



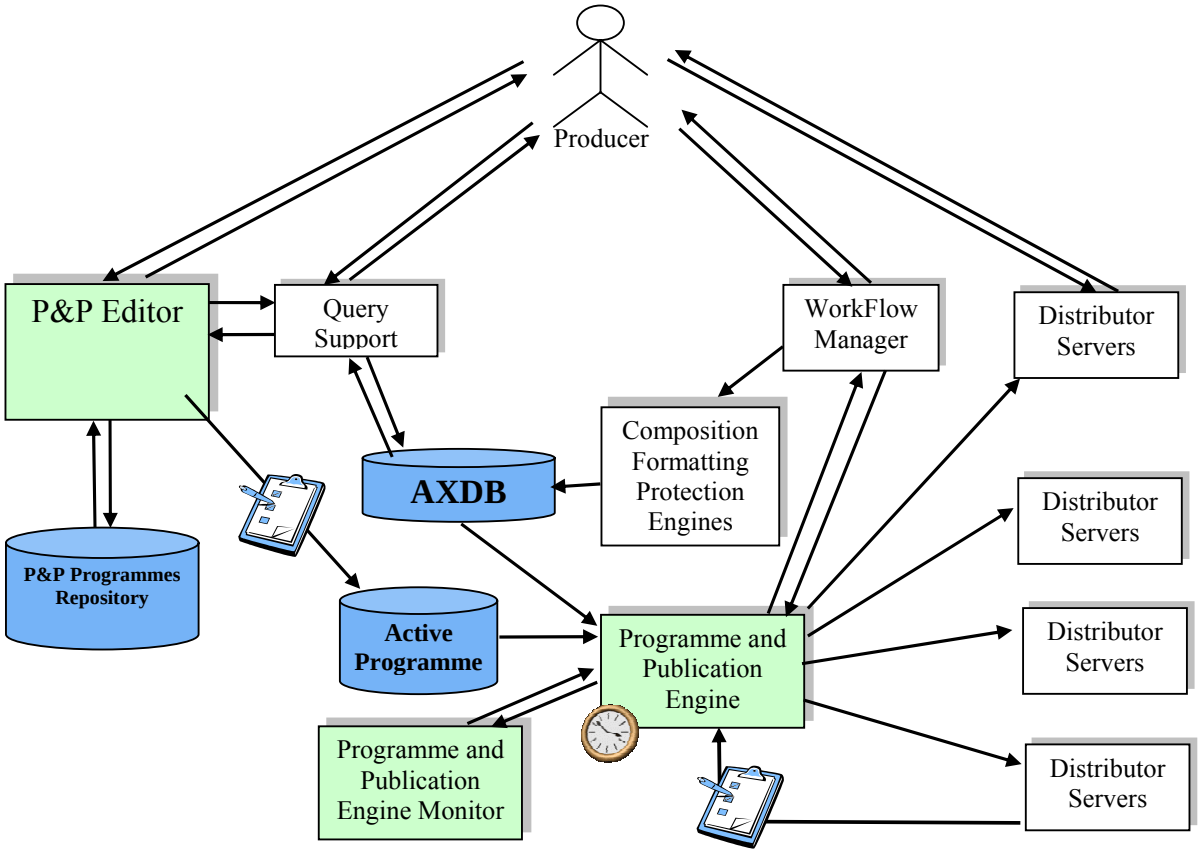
This use case begins when a programme manager wishes to monitor programmes running on the P&P Engine. The monitor application is launched and the actor can view and manage active P&P programmes.

1. The Actor starts the P&P Engine Monitor which connects to a P&P Engine
2. The P&P Monitor requests a list of active Programmes
3. The P&P Engine returns the current list of Programmes
4. The P&P Monitor updates the GUI window

The variation to this Use Case is that the Actor may select an active P&P programme from the list and "Kill" it. The P&P Engine aborts the Programme and notifies the Workflow that the specific P&P programme has been killed.

3 General architecture and relationships among the modules produced

The AXMEDIS Programme and Publication Area are comprised of three main components plus it utilises the Workflow manager. The P&P Editor is designed for creating, editing and saving P&P Programmes. A P&P Programme consists of the parameters and distribution rules for delivering AXMEDIS objects. The P&P Engine to execute the scheduled P&P Programme and the P&P Monitor for monitoring the P&P engine.



1. **Workflow Manager** – performs the role of supervisor and controlling the P&P Programme and P&P Editor activities.
2. **P&P Programme Editor** – is an editor for scheduling the distribution of AXMEDIS objects. It is supported by a P&P Programme Repository and interacts with the P&P Engine. The editor also uses
3. **P&P Engine** – performs the distribution activities for the scheduled objects in the activated programmes. The P&P engine distributes AXMEDIS objects to the distributor server using their API's or other defined methods. The P&P Engine interacts with the AXCP to request formatting of AXMEDIS objects if they are not compatible with the distribution channel specified and with the AXDB to retrieve the AXMEDIS objects for distribution.
4. **P&P Monitor** – interacts with the P&P Engine and provides a graphical user interface for actors to monitor the P&P Engine including killing activated P&P Programmes if necessary.
5. **AXDB** – AXMEDIS database is accessed using the AXDB manager object loader to retrieve the AXMEDIS object requested for distribution.
6. **AXCP** – AXMEDIS Content Processing is used by the P&P Engine to request the formatting of Objects to fit the profile for the distribution channel specified for the distribution of the AXMEDIS Object.
7. **Repository P&P Programmes**: It is a simple repository of programmes; it is the file systems with programmes described by means of an XML schema (axpnprule.xsd).
8. **Active Programmes**: They are programmes that are scheduled to be run by the P&P Engine.

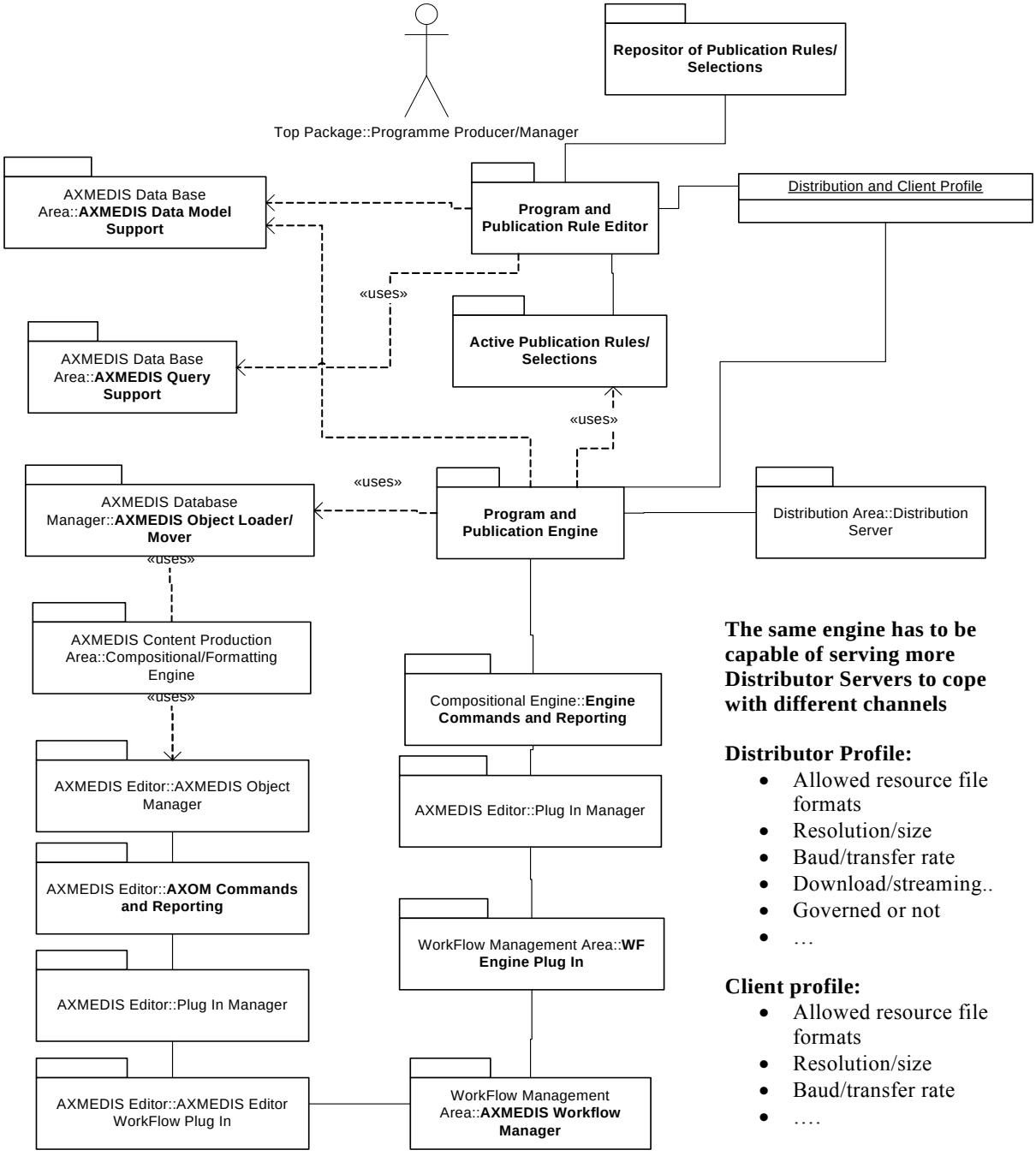
9. **AXMEDIS Database Manager** – It allows the P&P Engine to retrieve AXMEDIS objects involved in the execution of a programme after the request for formatting (if required). The P&P Engine uses the AXMEDIS Object Loader.
10. **AXMEDIS Query Support** – It allows the P&P Editor to submit queries to the AXMEDIS Database Manager to specify objects details including the AXOID to add to a P&P Programme.

3.1 P&P Programme and Publication Area UML Decomposition

This section describes the UML decomposition of the AXMEDIS Programme and Publication Area. It represents the complete picture for the Programme and Publication Area.
According t the UML diagram, the AXMEDIS Programme and Publication Area includes:

- **P&P Rule Editor:** A graphic editor that allows the creation, saving and editing of P&P Programmes.
- **Repository of Publication Rules:** It is a simple file system where the Programme and Publication rules (P&P Programme) are described by means of an XML schema (axpnprule.xsd)
- **Active Publication Rules:** They are P&P Programmes scheduled to be run by the Programme and Publication Engine
- **Compositional Engine:** Allows the P&P Engine to request formatting changes to an AXMEDI Object to fit the distribution profile for the distribution channel specified in the P&P Programme. This process uses AXMEDIS Workflow Manager and the AXMEDIS Object Manager (AXOM) see DE3-1-2-2-6 Specification of AXMEDIS Content Processing for more details.
- **AXMEDIS Query Support:** It allows the P&P Editor to submit queries to the AXMEDIS Database Manager
- **Distribution Area:** Is the various distribution servers the P&P engine can use through their APIs or other specified means for providing the AXMEDIS Object.
- **Distribution and Client Profile:** Is the information available on the various devices and distribution servers to provide information on how to calculate the distribution time and formatting of the AXMEDIS Object.
- **AXMEDIS Database Manager** – It allows the P&P Engine to retrieve the AXMEDIS objects involved in the execution of the P&P Programme using the AXMEDIS Object Loader.

Programme and Publication Area



The same engine has to be capable of serving more Distributor Servers to cope with different channels

Distributor Profile:

- Allowed resource file formats
- Resolution/size
- Baud/transfer rate
- Download/streaming..
- Governed or not
- …

Client profile:

- Allowed resource file formats
- Resolution/size
- Baud/transfer rate
- ….

4 Executable Tool Programme and Publication Engine (WP5.4.5 UNIVLEEDS)

Module/Tool Profile	
Programme and Publication Engine	
Responsible Name	Kia Ng and Garry Quested
Responsible Partner	UNIVLEEDS
Status (proposed/approved)	Proposed
Implemented/not implemented	Implemented
Status of the implementation	Version 1.0
Executable or Library/module (Support)	Executable
Single Thread or Multithread	Multithread
Language of Development	C++
Platforms supported	Microsoft Windows XP Win32
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/Application/axpnpeng/source https://cvs.axmedis.org/repos/Application/axpnpeng/include
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/repos/Application/axpnpeng/bin/win32/axpnpeng.exe
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	Currently running on local system as multiple engines
Test cases (present/absent)	Present (by running the P&P Editor and creating or loading an existing programme from the repository, the Engine can be tested by activating the P&P Programme)
Test cases location	TC10.3, TC10.4, TC10.5, TC10.6
Usage of the AXMEDIS configuration manager (yes/no)	Yes
Usage of the AXMEDIS Error Manager (yes/no)	No
Major Problems not solved	Optimisation of object delivery according to channel
Major pending requirements	Connection to distribution channels such as using the EUTELSAT API for distribution. Full integration with monitor to enable management of Programmes

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.)
P&P Model	AXCP, AXWF, AXOM, AXDB	Webservices
Formats Used	Shared with	format name or reference to a section
XML	P&P Editor	axpnprule.xsd (see P&P model)
Protocol Used	Shared with	Protocol name or reference to a section
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Simple CLI on Engine outputs status messages on STDOUT	C++	WinXP
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
wxWidgets	wxWidgets-2.4	wxWidgets licence (http://www.wxwidgets.org/newlicen.htm)
Xerces	XERCES version 2-6	Apache Software License, Version 2.0. http://www.apache.org/licenses/LICENSE-2.0.html

4.1 General Description of the Module

The Programme and Publication Engine will be developed exploiting the work performed for the Publication tool in WP4.4. This will allow the reception of specific commands (requests) for creating content produced by exploiting the capabilities of the AXMEDIS formatting engine. In addition, the Programme and Publication Engine will also have the capabilities for the publication the Programme based on the specific rules.

The active engine is continuous running software accessing the system clock to process a list of programmes, which consists of "rules" to make available AXMEDIS objects to the specified destination channels at the correct time, taking into account the transfer and/or formatting (if required) time. This is achieved by the input of *activated* P&P Programme for scheduled distribution.

The active Programme and Publications Engine's main function is to continually run looking for active publication rules and make the objects in the rules available for distribution. The main points to consider:

- Access to correct system clock
- Keep a track of newly activated P&P Programmes to add to the delivery system
- The API to the AXMEDIS formatting engine to request an appropriate format for distribution and retrieving the correct object from the AXDB
- Providing the AXMEDIS objects to a AXMEDIS distribution server allowing for delivery time

On Demand for the user is to be done by the Active Engine during the "transfer" schedule, just before the actual AXMEDIS object is to be delivered to the distribution server. A check has to be carried out to see if the object in question is "compatible" with the destination profile. The check could be done when new Programme is "activated" by the actor so that the Engine knows these additional requirements and can allow

for necessary formatting time before actual physical delivery. Non-compatible objects require the Programme and Publication Engine to interface with the Formatting engine to provide the appropriate processing to reformat the object for distribution. For example, the AXMEDIS object in the database could be for HDTV and the Programme producer has requested it to be used on a PDA. The Programme and Publication Engine provides the Formatting engine a reference to the AXMEDIS object and the destination profile.

The Formatting engine is expected to

- Take a copy of the AXMEDIS object
- Process/convert/etc the object so that it is compatible to the destination profile
- Deliver it to a Distribution Server or destination

4.1.1. Configuration management

Configuration is handled by the AXMEDIS Configuration Manager. The port that the server listens on is configured here. Otherwise it will default to 3000. The server also listens on the next port up (3001) for monitor requests.

4.1.2. Please verify if some your components can be produced customizing a component produced by other partners or can used by other partners in other tools

There are no components that could be used of use to other partners at this time

4.1.3. Interoperability on different platforms,

The project is built and tested on Windows XP but uses only portable libraries and code so could easily be made cross platform (provided the code from other partners also supports this).

4.1.4. Print capability of the information manipulated

No support for this. This is not required for this specific module.

4.1.5. Protection aspects (registration, certification, operation control, access to certifier, DRM, etc.), please consult protection experts

No support. This is not required for this specific module.

4.1.6. Help to support the users,

The server is not interactive so it can be started and stopped. No help is needed. The monitor has limited built in help in the GUI which will be expanded as its functions become more complex

4.1.7. Multilingual support of the user interface and of the help

There is currently not scheduled for implemented but the P&P Engine code can be internationalised with the wxWidgets macros

4.1.8. Undo support that could be obtained with controlling all commands

This is not appropriate and not required for this specific module.

4.1.9. Workflow and cooperative work support to be integrated with the Workflow tools that will be selected for AXMEDIS, etc.

The P&P Engine is integrated with workflow with communication and process control such as activation, deletion, and reporting.

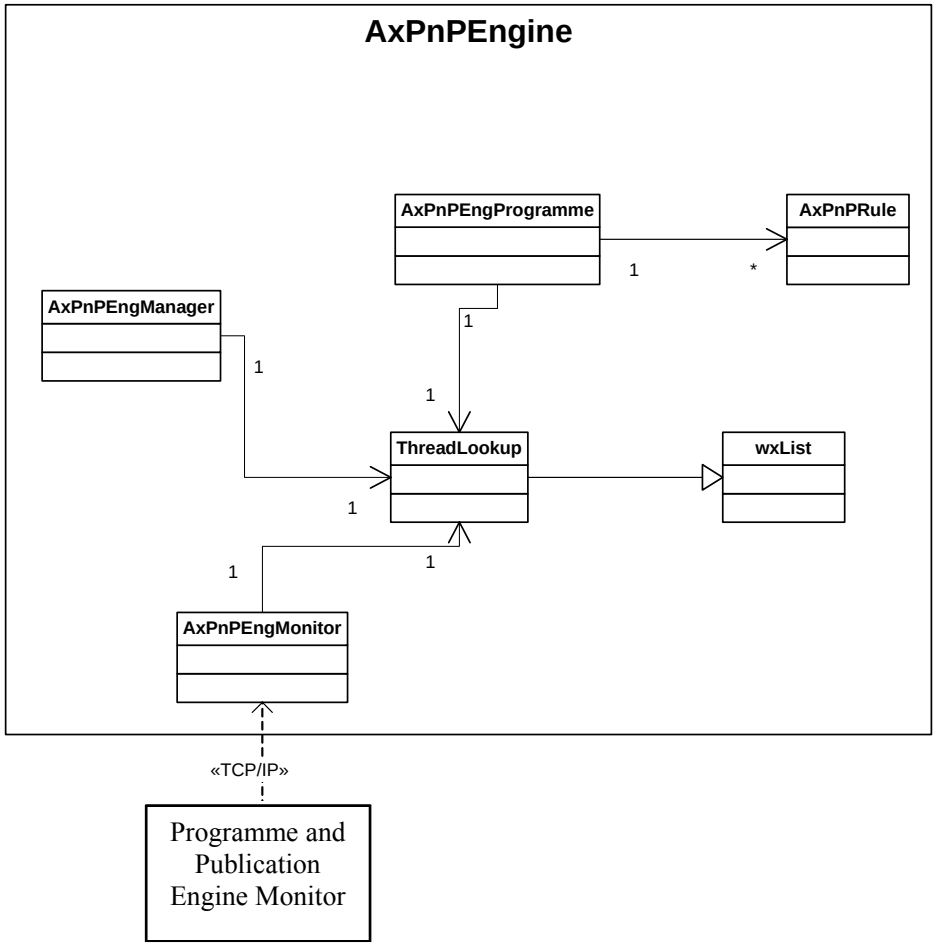
4.1.10. Insert an about for citing, copyright, AXMEDIS projects and EC in a proper manner, as will be defined later.

The P&P Engine does not directly interact with the user (but via the Editor and the Monitor) and hence no need to have the “About” dialog as in GUI interface. However, the “About” will be included as command options and start-up option.

4.1.11. Refer to used standards providing references and documents for the other partners. These documents will be made accessible to all via WEB.
The P&P Engine uses the AXMEDIS P&P schema (axpnprule.xsd) which is accessible to all via the web portal. The communications protocol between the P&P Engine and the WF is also accessible to all via the web portal.

4.1.12. Declare any library and tools that you are going to use and the license level/type/cost for that tools/libraries, etc. According to the CA you have to be very carefully in using:
wxWidget, Xerces

4.2 Module Design in terms of Classes



4.3 User interface description

4.3.1 Engine

The engine when running is non-interactive. It outputs status messages to the console. See the P&P Engine Monitor concerning the user interface for the P&P Engine.

4.4 Technical and Installation information

The compiled versions of axpnpeng and axpnpengmonitor are easy to install.
To run the engine, a copy of the executable axpnpeng.exe must be present along with the following components:

- xerces-c_2_6.dll

- libcurl.dll
- editor-configuration.xml
- configuration.xsd
- AXPNPRULES/axppnrule.xsd
- AXPNPRULES/Selection_v1.6.xsd

4.5 Draft User Manual

See next subsection ("Examples of usage") on how to run the P&P Engine.
See the Draft User Manual for the P&P Engine Monitor (section 5.5 Draft User Manual) for the usage of the Engine which works with the P&P Engine Monitor.

4.6 Examples of usage

Run the executable from within the OS (developed and tested under WinXP) by either double clicking the exe file or by typing the path and filename in a DOS command shell.

It will now listen for programme requests from the P&P Editor and Workflow. On receiving a Programme request, the P&P Engine will schedule delivery of the AXMEDIS objects to ensure on-time delivery. The Engine will retrieve the object from the AXDB and deliver it using an appropriate protocol for the target, e.g. FTP. The programme can be aborted at any point by Workflow sending an abort request.

4.7 Integration and compilation issues

The .NET preferences are set to find the following libraries and requires the environment variables to be set for wxWidget, LibCurl and WsdlPull and Xerces. Control Panel -> System -> Advanced -> Environment Variable

- $(WXWIN) = where wxWindows-2.4.2 is installed
- $(LIBCURL) = where libcurl-devel-7.15.1 is installed
- $(WSDLPULL) = where wsdlpull-1.9 is installed
- $(XERCES) = where xerces-c_2_6_0 is installed

NB if you are building Xerces from source you will have to set XERCESCROOT to where you have installed the source code for Xerces. Building the libraries can be done as a batch build or build individually in the build menu for the Debug and Release libraries.

4.8 Configuration Parameters

Config parameter	Possible values
AXMEDIS_PLUGIN_MANAGER ▪ PLUGINS_PATH	<*path-to-executable*>/plugin
DATABASE ▪ queryUrl ▪ user ▪ passwd ▪ LoaderWSEndPoint	These parameters are user definable in the configuration. There are no default parameters. <*address_ to_database_loader_endpoint* >
WORKFLOW ▪ workflowUrl ▪ gatewayUrl	http://<*workflow-address*>:8080/OpenFLow/index_html http://<*workflow-address*>:8080/responseGateway/reChannel/reChannel.asmx

4.9 Errors reported and that may occur

Error code	Description and rationales
`Could not create new wxSocketServer`	The engine could not bind to the specified port on start-up
`Accept 0 failed`	Socket Accept method failed for programme port
`Accept 1 failed`	Socket Accept method failed for monitor port

`Socket Error: Bad on-demand message: content length incorrect`	The programme thread has received a message which it can't handle
`Socket Error: Bad on-demand message format (no space char)`	The programme thread has received a message which it can't handle
`Thread %d requested unavailable file: %s`	A programme refers to an AXObject that is not available
`Thread %d can't open file: %s`	The AXObject cannot be opened (appears to be damaged)
`Socket Error: Bad pnp programme message format: content length too short`	A P&P Programme was received but was not the same length as the header stated it would be
`Error loading pnp programme`	The AxPnPRule object could not parse the P&P Programme
`Error: No object available, unable to execute`	The AXObject could not be obtained

5 Executable Tool Programme and Publication Monitor (WP5.4.5 UNIVLEEDS)

Module/Tool Profile	
Programme and Publication Engine Monitor	
Responsible Name	Kia Ng and Garry Quested
Responsible Partner	UNIVLEEDS
Status (proposed/approved)	proposed
Implemented/not implemented	Implemented
Status of the implementation	Version 1.0
Executable or Library/module (Support)	Executable
Single Thread or Multithread	Multithread
Language of Development	C++
Platforms supported	Microsoft Windows XP Win32
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/Application/axpnpeng/source https://cvs.axmedis.org/repos/Application/axpnpeng/include
Reference to the AXFW location of the demonstrator executable tool for internal download	https://cvs.axmedis.org/repos/Application/axpnpeng/bin/win32/axpnpenginemonitor.exe ACTIVITIES/WP4 and WP5http://www.axmedis.org/attivita/documenti/download.php?area_id=4&attivita_id=17&l_s=struttura&gruppo=0&order_by=data&asc_desc=desc/General on AXFW and Guidelines/ http://www.axmedis.org/documenti/view_documenti.php?doc_id=1617
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)	Present (testing is achieved in conjunction with the axppeng software)
Test cases location	TC10.6
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	management of programmes as well as monitoring

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.)
P&P Engine	None	TCP/IP
Formats Used	Shared with	format name or reference to a section
Protocol Used	Shared with	Protocol name or reference to a section
Simple application layer networking protocol to send requests to the engine over TCP/IP	axpnpengine	TCP/IP
Used Database name		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
Graphical user interface provides a tree view of programmes currently active on the server	C++	wxWidgets, WinXP
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
wxWidgets	wxWidgets-2.4	wxWindows licence (http://www.wxwidgets.org/newlicen.htm)
Xerces	XERCES version 2-6	Apache Software License, Version 2.0. http://www.apache.org/licenses/LICENSE-2.0.html

5.1 General Description of the Module

This module provides a graphical interface to enable the user to monitor active programmes on the programme and publication engine. The engine is a server and so outputs status messages locally (to screen). The monitor may be on a local or remote machine and can query the engine over the network using TCP/IP.

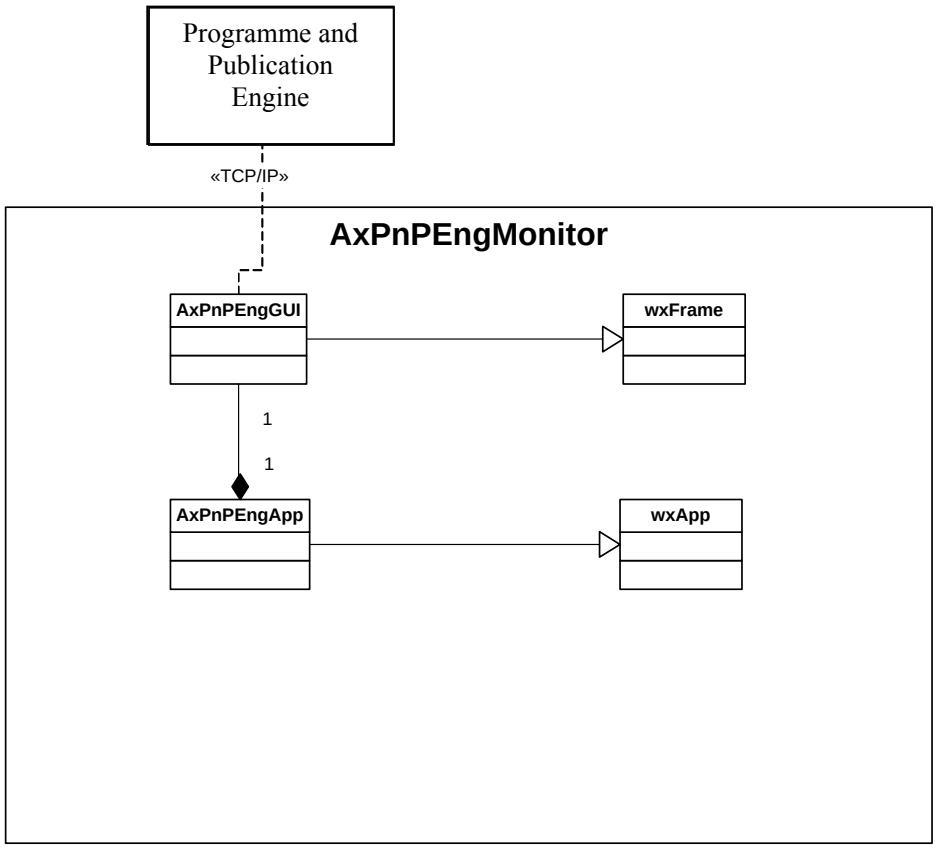
The project is built and tested on Windows XP but uses only portable libraries and code so could easily be made cross platform. The P&P Engine Monitor does not use the configuration management system. This is handled by the P&P Engine and only the Engine and P&P Programme monitoring is handled. As a front end GUI for the P&P Engine, the P&P Engine Monitor is not customisable for other partners and other AXMEDIS tools.

Currently, the monitor has limited built in help in the GUI which will be expanded as its functions become more complex. Multilingual support is not implemented but can be internationalized with wxwidgets macros

There is no integration with Workflow as this is handled by the P&P Engine for command and reporting. This may be needed when the monitor is able to manage as well as monitor active programmes. The P&P Engine is to support an ‘About’ for citing, Copyright etc. required for AXMEDIS projects and EC in a proper manner, as will be defined later. This however has not been implemented yet.

The P&P Engine Monitor uses the wxWidget 2.4 libraries with the wxWindows licence (see (http://www.wxwidgets.org/newlicen.htm) and the Xerces 2.6 libraries with the Apache Software License, Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0.html)

5.2 Module Design in terms of Classes



5.3 User interface description

5.3.1 Monitor

The Monitor provides a GUI to the P&P Engine. It is a separate process and can be run on a different machine to the P&P Engine.

When a user runs the P&P Engine Monitor, an application window appears. There is a menu of functions and two tabs which take up the rest of the frame. The first tab is for status messages (connection successful etc.) and the second is for programme management (kill programmes etc)

Menus

There are two menus. The action menu which includes:

- Connect (connect to an engine
- Disconnect
- Save Programmes (not yet implemented but will save current programme details to file)

The second menu is the help menu:

- Index (will open the help document)
- About (opens a dialog box with information about the software)

5.4 Technical and Installation information

To run the P&P Engine Monitor the following files are currently required:

- axpnpengmonitor.exe
- axmedis_pnp_logo.bmp

In the release mode, no other libraries or components are required.

5.5 Draft User Manual

On start-up of the P&P Engine monitor, the user connects to the P&P Engine by selecting the Action menu and the connect menu item.

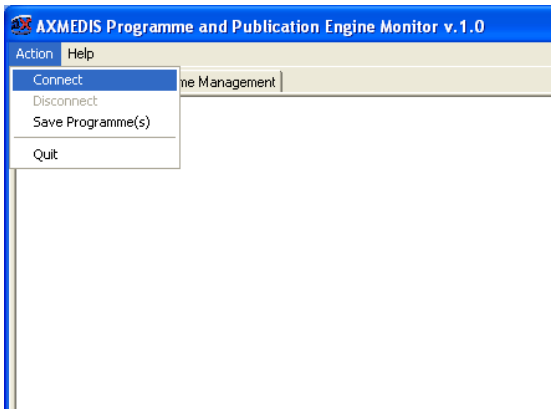


Figure: P&P Engine Monitor Action Menu

On connection the user is presented with two dialog boxes, the first is to select the address of the server (P&P Engine) (shown in the Figure below) and the second is the port for P&P Engine has designated for the P&P Engine Monitor.

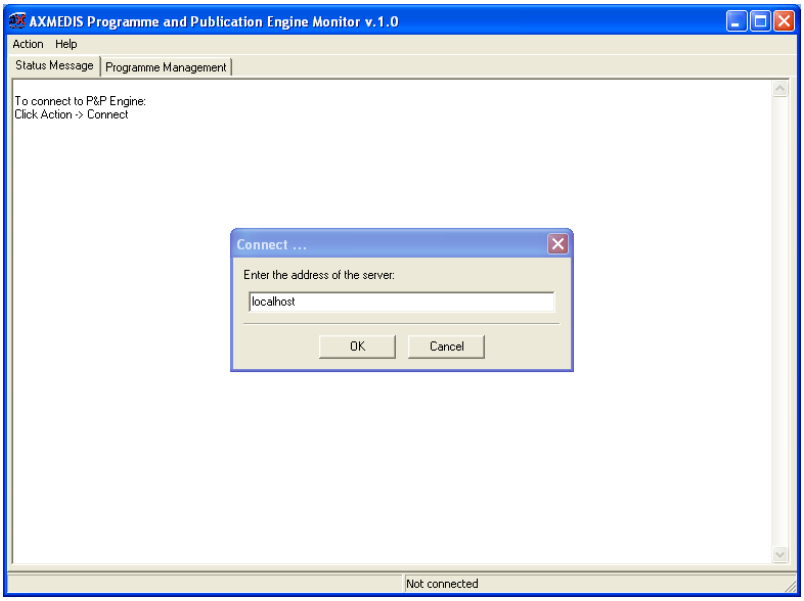


Figure: P&P Engine Monitor with connection Dialog box and Status Message Tab selected

On connection, the user can view the active P&P Programmes by selecting the Programme Management tab as shown in the Figure below.

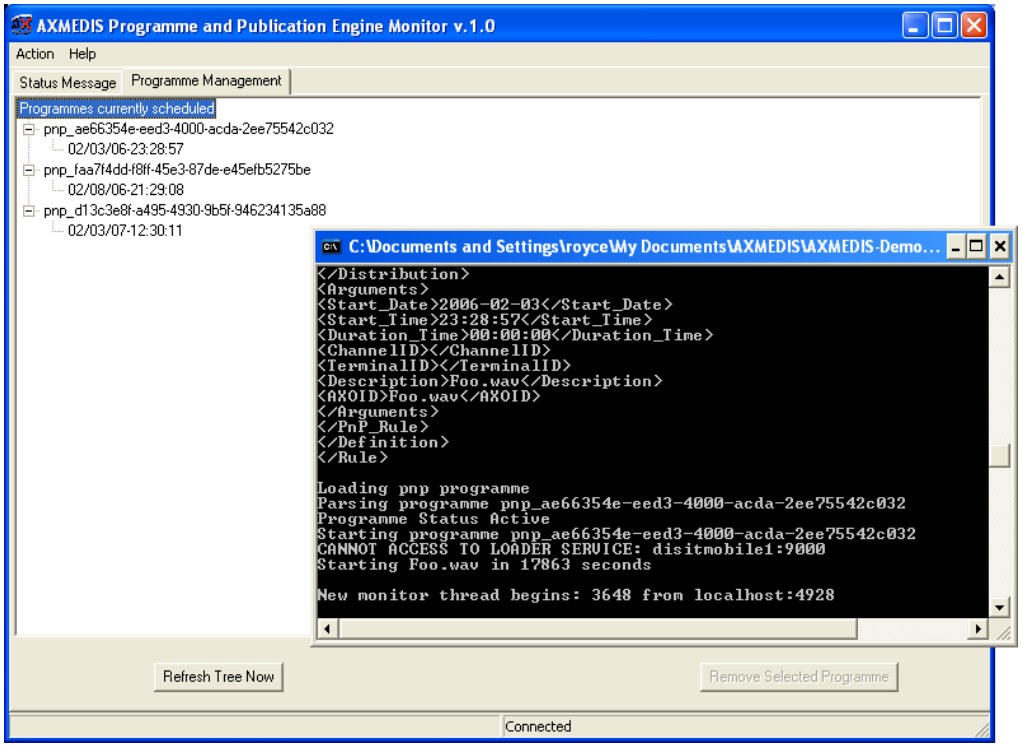


Figure: P&P Engine Monitor Programme Management View with P&P Engine Command prompt running in the foreground.

5.6 Examples of usage

Run the executable from within your OS and connect to an engine (this defaults to localhost and can be set to a different hostname at connection time.

To view currently running programmes on the P&P Engine, the user connects to the P&P Engine by selecting the "action" menu. On connection, the user is presented with a tree view of all the currently activated P&P Programmes currently running on the P&P Engine.

5.7 Integration and compilation issues

The P&P Engine Monitor is self contained and is used to connect to the P&P Engine. Therefore there are no integration issues within the AXMEDIS Framework and the P&P Engine Monitor. Compilation requires the use of wxWidget. The preferences are set to find the wxWidget libraries and requires the environment variables to be set for wxWidget and Xerces. Control Panel -> System -> Advanced -> Environment Variable

- $(WXWIN) = where wxWindows-2.4.2 is installed

5.8 Configuration Parameters

Currently, there are no mandatory configuration parameters, but standard default parameters and the use of history to preserve the last use are intended to be added to the implementation.

5.9 Errors reported and that may occur

Error code	Description and rationales
`Could not create new wxSocketServer`	The engine could not bind to the specified port on start-up
`Accept 0 failed`	Socket Accept method failed for programme port
`Accept 1 failed`	Socket Accept method failed for monitor port
`Socket Error: Bad on-demand message: content length incorrect`	The programme thread has received a message which it can't handle
`Socket Error: Bad on-demand message format (no space char)`	The programme thread has received a message which it can't handle
`Thread %d requested unavailable file: %s`	A programme refers to an AXObject that is not available
`Thread %d can't open file: %s`	The AXObject cannot be opened (appears to be damaged)
`Socket Error: Bad pnp programme message format: content length too short`	A P&P Programme was received but was not the same length as the header stated it would be
`Error loading pnp programme`	The AxPnPRule object could not parse the P&P Programme
`Error: No object available, unable to execute`	The AXObject could not be obtained

6 Executable Tool Programme and Publication Editor

Module/Tool Profile Programme and Publication Editor	
Responsible Name	Kia Ng and Royce Neagle
Responsible Partner	UNIVLEEDS
Status (proposed/approved)	proposed
Implemented/not implemented	Implemented
Status of the implementation	Version 1.0
Executable or Library/module (Support)	Executable and Library
Single Thread or Multithread	Multithread
Language of Development	C++
Platforms supported	Microsoft Windows XP (Tested)
Reference to the AXFW location of the source code demonstrator	Executable: https://cvs.axmedis.org/repos/Application/pnpeditor/source https://cvs.axmedis.org/repos/Application/pnpeditor/include Library: https://cvs.axmedis.org/repos/Framework/pnpeditor/source https://cvs.axmedis.org/repos/Framework/pnpeditor/include
Reference to the AXFW location of the demonstrator executable tool for internal download	Debug: https://cvs.axmedis.org/repos/Application/pnpeditor/bin/win32/axpnpeditor-debug.exe Release: https://cvs.axmedis.org/repos/Application/pnpeditor/bin/win32/axpnpeditor-release.exe ACTIVITIES/WP4 WP5http://www.axmedis.org/attivita/documenti/download.php?area_id=4&attivita_id=17&l_s=s on AXFW and Guidelines/ http://www.axmedis.org/documenti/view_documenti.php?doc_id=1617
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)	present
Test cases location	TC10.1, TC10.2, TC10.3, TC10.5
Usage of the AXMEDIS configuration manager (yes/no)	Yes

Usage of the AXMEDIS Error Manager (yes/no)	No	
Major Problems not solved		
Major pending requirements	Connection to distribution channels and specific interfaces such as the EUTELSAT API for dis parameters to specify within the P&P Programme the channel and terminal information.	
Interfaces API with other tools, named as	Name of the communicating tools References to other major components needed	Communication model and format
Query Support	AxSelectionEditor	Webservices (password protected)
Workflow	Workflow Plugin	Webservices
Formats Used	Shared with	format name or reference to a secti
XML plus schema ▪ `axpnprule.xsd`	P&P Engine Executable	Programme and Publication Forma
Protocol Used	Shared with	Protocol name or reference to a sec
WebServices	AXMEDIS Workflow - Master	See DE3-1-2-2-12 on Workflow Tools Spe
WebServices	AXMEDIS Workflow - Slave	See DE3-1-2-2-12 on Workflow Tools Spe
Socket (TCP/IP)	P&P Engine	See section 1.5 "List of Protocols Specifie
Used Database name		
AXDB		
User Interface	Development model, language, etc.	Library used for the development,
	Visual Studio 7 .NET	AXMEDIS Plug-in Manager
	C++	AXMEDIS Configuration Manage
		AXMEDIS common
		AXMEDIS pnpmodel
		AXMEDIS AxSelection
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK,
Xerces	XERCES version 2-6	Apache Software License, Version http://www.apache.org/licenses/LI
wxWidget	wxWindows version 2.4	wxWindows licence (http://www.wxwidgets.org/newlic

6.1 General Description of the Module

A Programme Manager can use a GUI to interact with the Query Support User Interface to make selections from the Query results in order to schedule some programmes (e.g. day, week, month, and year) with the following rules:

- WHAT: the AXMEDIS object of interest
- WHERE: destination channel, where to publish e.g. iTV or kiosk or other, and "where" profile
- WHEN: date, time, slot, duration
- HOW: direct transfer, reference or require formatting engine

The representation of the above rules could be represented using XML.

The Programme and Publication Programme Editor will provide the following functionality:

- Create: a Programme Manager uses a GUI to create some P&P Programmes using the Query Support User Interface to browse the AXMEDIS database, to select, to schedule, and to return with a list of relevant objects

- Edit: a Programme Manager make changes to the programme rules selected from a list in the GUI read from the P&P Programme Repository. Changes can also include using the Query Support User Interface to browse the AXMEDIS database to select and to return with a list of relevant objects
- Save: send the P&P Programme to a P&P for archiving (configured local file)
- Test: Test the P&P Programme through the P&P Engine
- Activate: send the P&P Programme to the P&P Engine

The Configuration is handled by the AXMEDIS Configuration Manager to set the configuration of the P&P Editor including the GUI related parameters, Workflow gateway parameters, P&P Engine connection address and AXMEDIS Plug-in parameters to access Workflow. The project is built and tested on Windows XP but uses only portable libraries and code so could easily be made cross platform (provided the code from other partners also supports this). Currently the wxWidget and Xerces libraries are being used.

- Xerces licence: Apache Software Licence, Version 2.0.
- wxWidget licence:

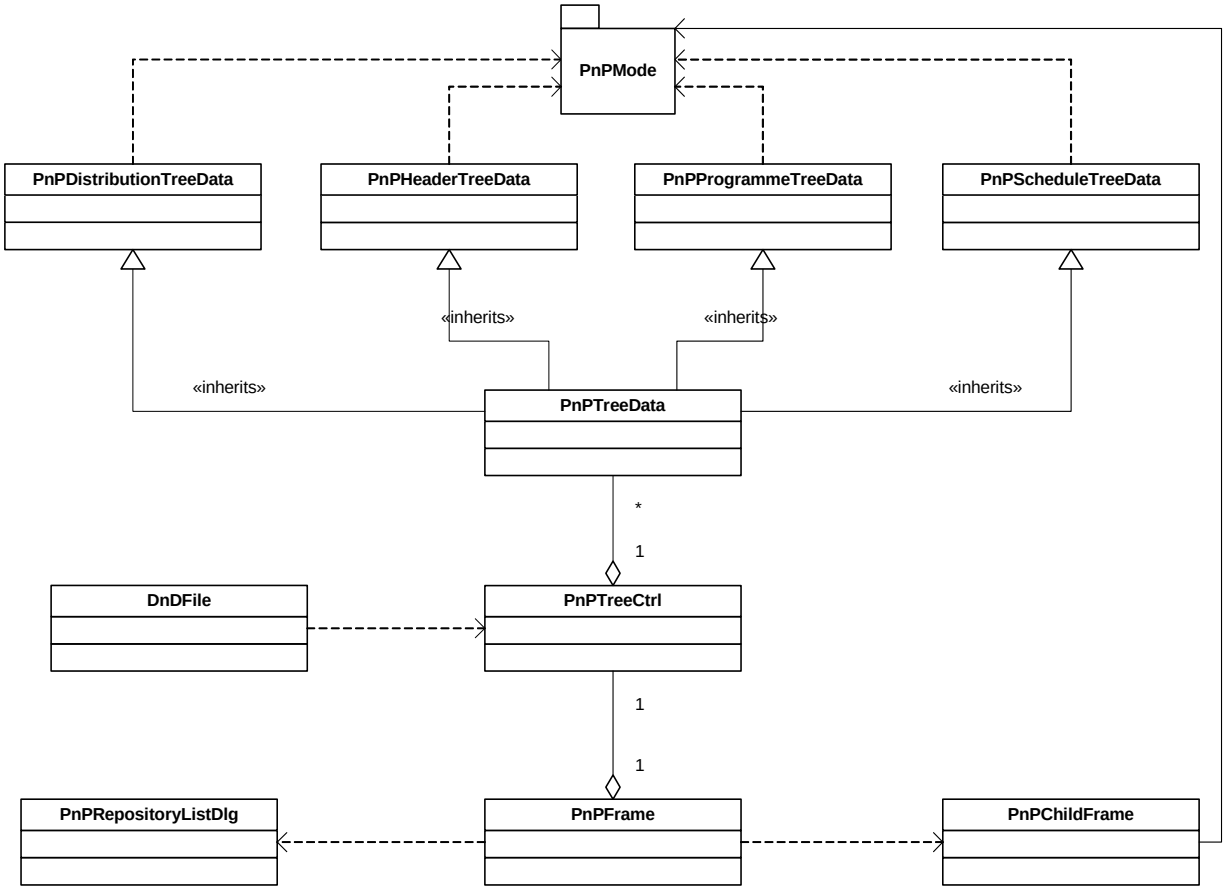
There are no components that could be of use to other partners at this time. The P&P Editor was specifically designed as a front end for P&P Programme creation and editing and currently the P&P Editor has not been implemented with multilingual support for the GUI interface. However, the wxWindows macro has been used for future development to begin implementation for multilingual support.

Functionalities to be implemented and are currently included are the printing of P&P XML Programme Rule. Future releases will also have the P&P Editor connecting to HTML help documentation from the Help Menu. The 'About' for citing is currently implemented without the citing, copyright etc. information inserted for display.

The P&P editor is integrated with AXMEDSI Workflow. Workflow can request editing of a new P&P Programme to send a P&P Programme with the request for editing. Workflow can also request the activation of a programme through the P&P Editor. These requests when completed send a notification for completion back to the AXMEDIS Workflow completed the request loop.

The P&P Engine Monitor uses the wxWidget 2.4 libraries with the wxWindows licence (see http://www.wxwidgets.org/newlicen.htm) and the Xerces 2.6 libraries with the Apache Software License, Version 2.0 (http://www.apache.org/licenses/LICENSE-2.0.html)

6.2 Module Design in terms of Classes



6.3 User interface description

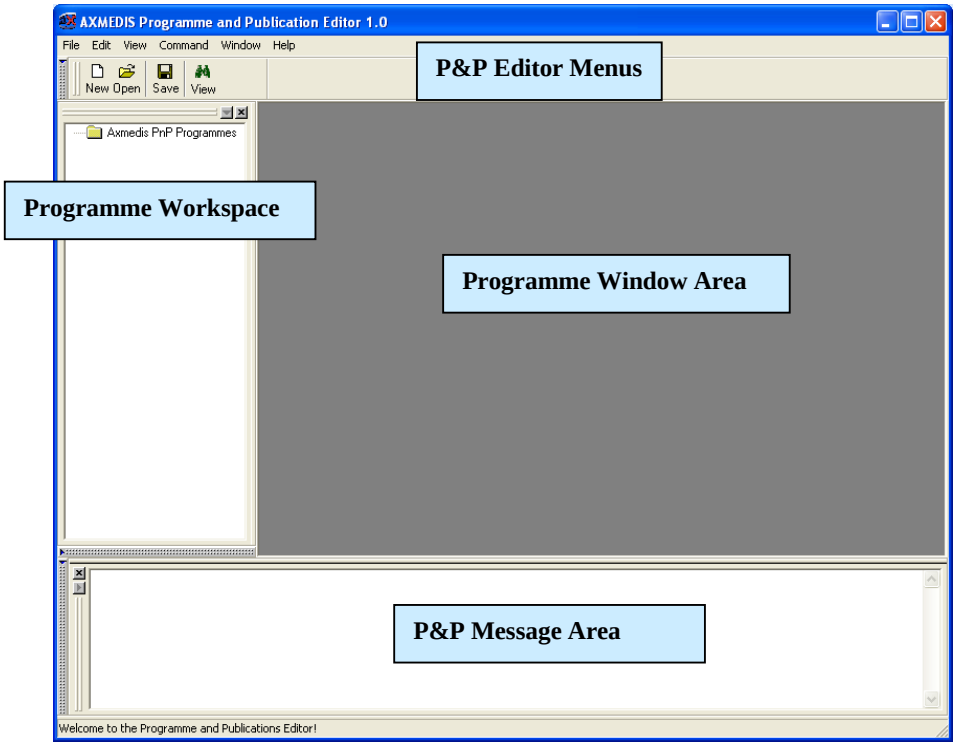


Figure: Initial P&P Editor View

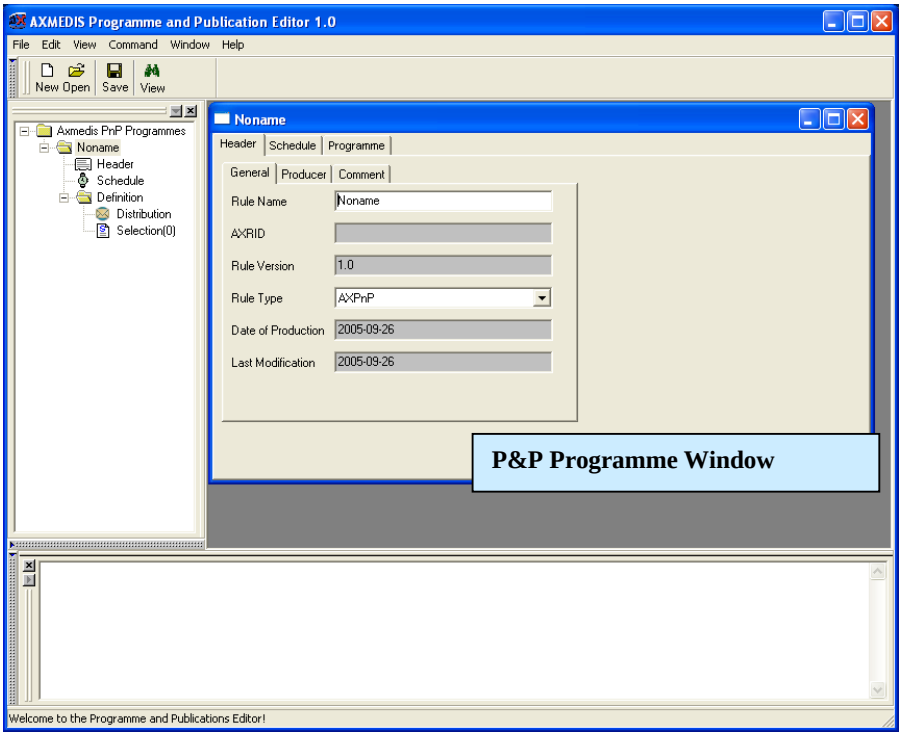


Figure: P&P Editor Displaying Programme Window and Workspace with Tree View

6.4 Technical and Installation information

- Installation capability, it has to be installable in a very easy manner
- Manual support for technical and user point of views

Installing the AXMEDIS Programme and Publication Editor is simply installing the executable axpnpeditor.exe into a directory and double clicking to run the application. The following libraries and file requirements are necessary to run the applications.

Libraries Required

- xerces-c_2_6.dll (Xerces - http://xml.apache.org/xerces-c/)

Files Required

- configuration.xsd
- axpnprule.xsd
- Selection-v1-6.xsd
- pnpeditor-configuration.xml (configuration file)
- plugin/rePlugin.dll (workflow rule editor Plug-in)
- plugin/rePlugin.xml (workflow rule editor Plug-in configuration file)

References to other major components needed	▪ P&P Engine to upload a programme is needed to be running to activate and test a P&P Programme. ▪ Query Support User Interface must be running ▪ AXMEDIS Workflow can be running
Problems not solved	▪
Configuration and execution context	▪ Set the path for the Query Support User interface ▪ Set the Path for the Plug-in Directory to load the Workflow Plug-in ▪ Set the gateway for the Workflow

	▪ Set the Engine and Port Address for the P&P Engine

6.5 Draft User Manual

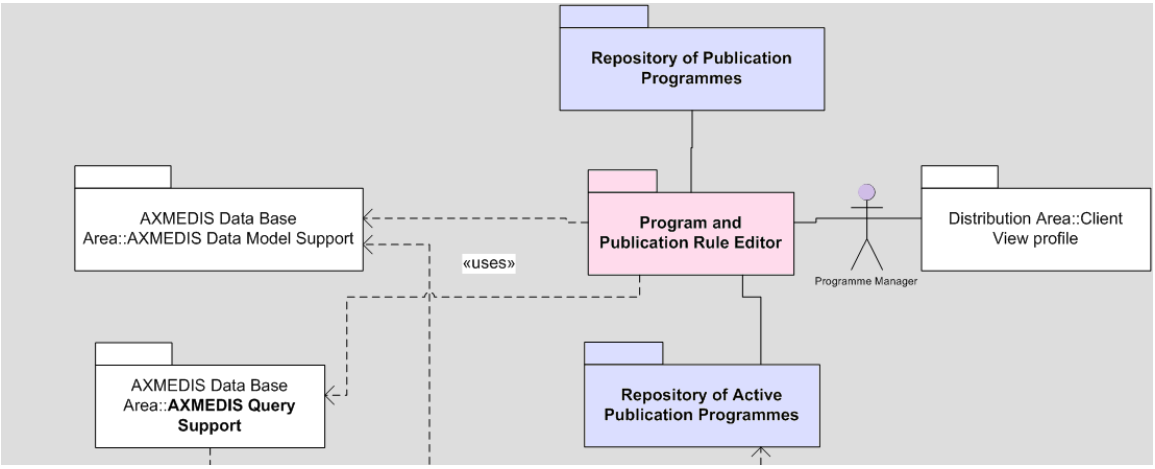


Figure: Scenario 1 – Distribution from a P&P Programme

Scenario a: New P&P programme

1. The actor opens the GUI (either through WF or directly) and selects a new P&P Programme
2. The programme manager uses query support to add AXObjects and/or Selections to the P&P Programme
3. The programme manager edits P&P Programme
4. The programme manager saves the P&P Programme to the P&P Repository

Scenario b: Loading and editing existing P&P Programme:

1. The programme manager opens the GUI and loads a P&P Programme from the P&P Programme Repository
2. The programme manager uses query support to add AXObjects and/or Selections to edit the P&P Programme
3. The programme manager saves the rule to the P&P Repository

Scenario c: Activating or testing an existing P&P programme:

1. The programme manager opens the GUI and loads a P&P Programme from the P&P Programme Repository
2. The programme manager activates/full trials/quick trials the P&P Programme
3. The programme is sent to the P&P Engine using the Active P&P Programme Repository

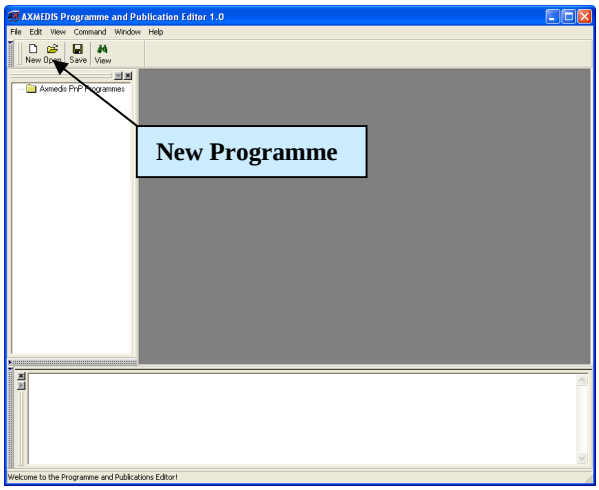
Programme and Publication Programme Production

A programme producer or manager wishes to create a programme of multimedia objects for viewing on a selected distribution channel. This can be achieved by using the P&P Editor to create a distribution programme which when complete is activated.

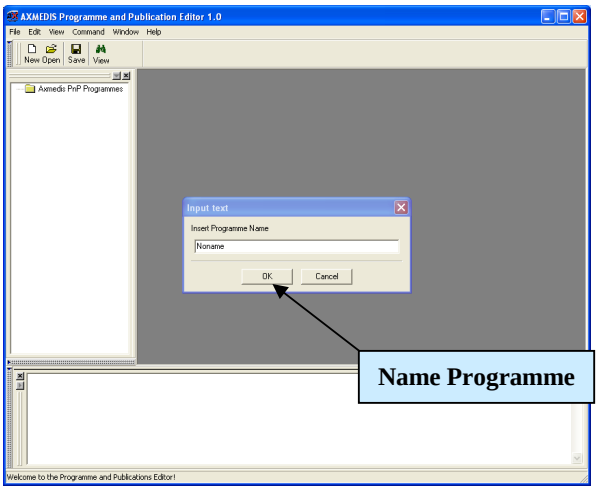
By opening the P&P Editor the programme manager can create a programme to specify the time and destination channel of the objects to be distributed. Once a programme has been created, the programme manager can save it for re-editing at a future time or test the programme. Testing the programme enables the programme manager to check if there are going to be any problems in sending each object and allow the manager within the editor to make changes to successfully distribute a programme. On completion of a complete programme, the programme manager activates the programme and during the programme life cycle the objects will be distributed until the programme has completed or the programme is removed from being activated.

Creating a programme

To create a programme, the programme manager starts the P&P Editor and from the opening screen creates a new programme by selecting "*New*" from the tool bar or using the File menu (File → New) or using the keyboard short cut "Ctrl-N" (see Screenshot 1). On requesting a new programme, the dialog box pops up and the programme managers enters the name of the draft programme and selects "OK" (see Screenshot 2).

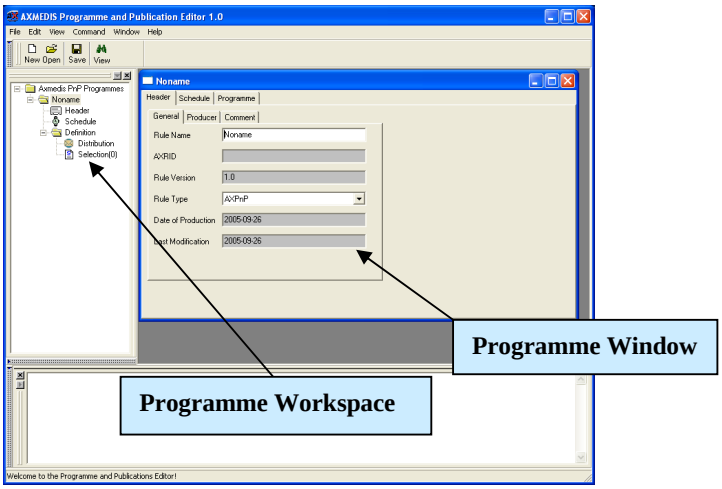


Screenshot 1



Screenshot 2

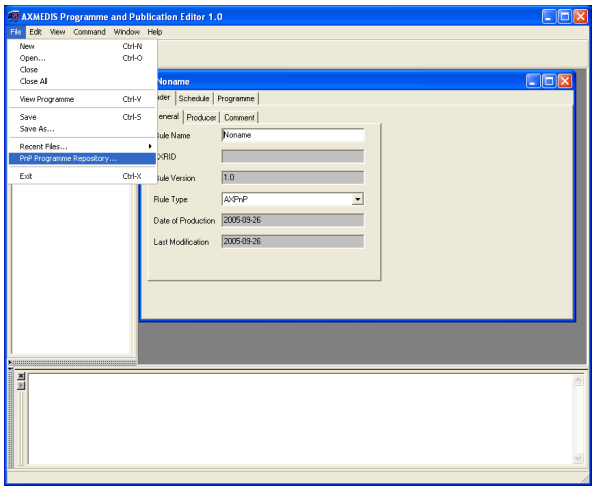
After the programme name has been OK'd, the new programme is ready for editing with the '*tree view*' used as a workspace and the programme window for editing the programme details and making a programme schedule (see Screenshot 3).

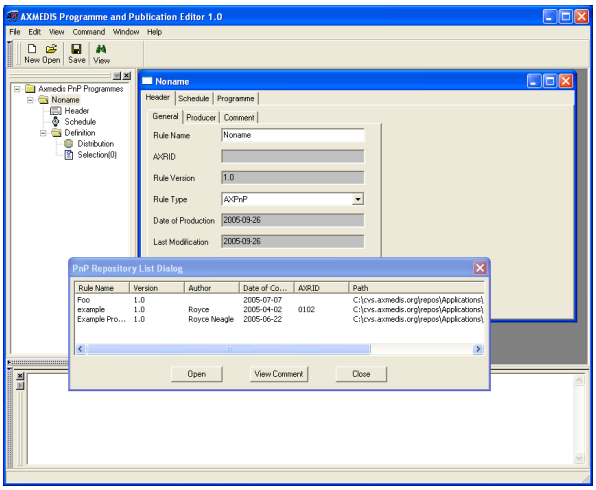


Screenshot 3

Loading and existing programme

Instead of creating a new programme, the programme manager may wish to edit an existing draft programme. The programme manager may have saved it as a file somewhere or saved it in the P&P Repository. By selecting '*open*' or '*P&P Programme Repository*', as seen in screenshot 4 and 5, the programme manager can load an existing draft programme from the Repository or elsewhere on the system.

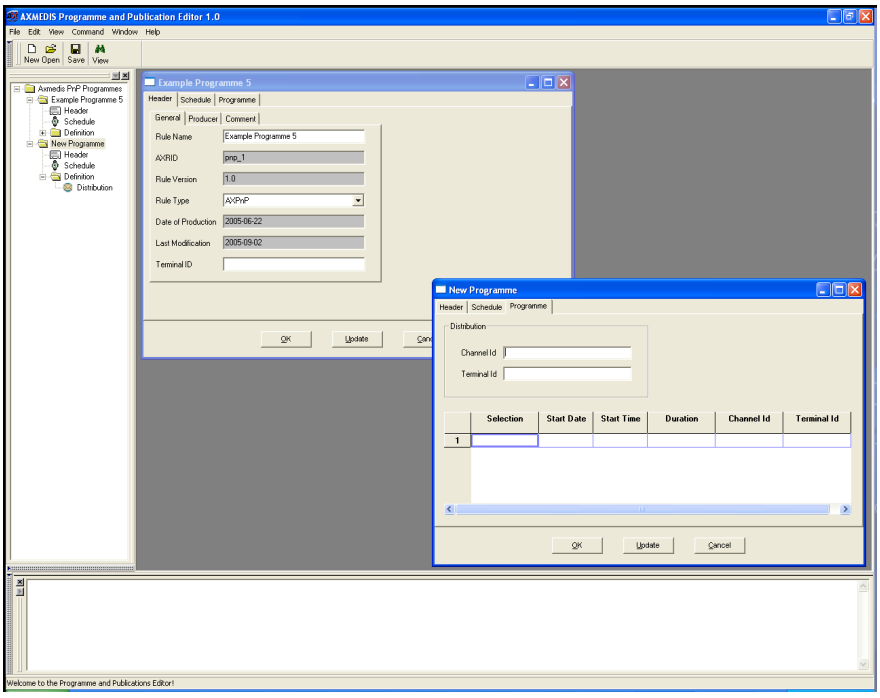




Screenshot 4 **Screenshot 5**

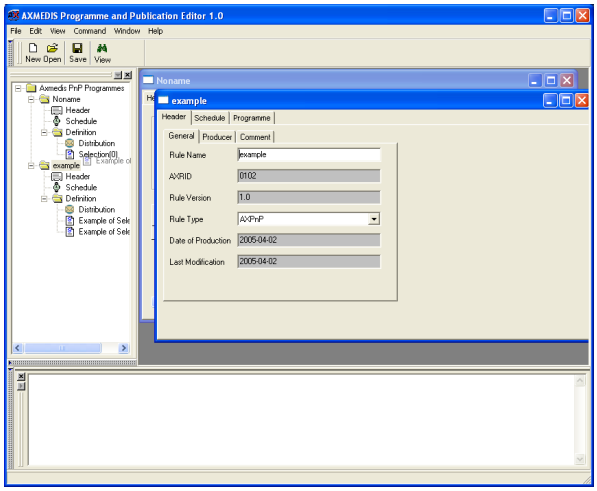
Editing a programme

The programme manager can use the editor to edit multiple programmes (as shown in screenshot 6), and edit programme data such as general, producer and content information; set the schedule for activating the programme; and set the specifications for delivering a multimedia object such as channel and terminal as well as distribution date and time, duration, channel if a different channel for a particular multimedia object is required, and terminal if a different terminal is required for a particular multimedia object.

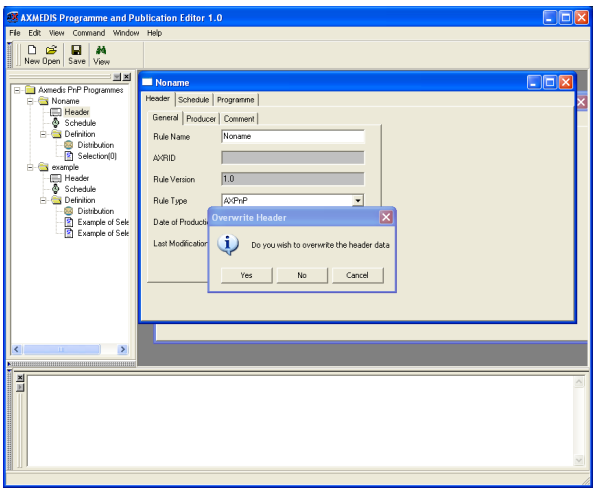


Screenshot 6

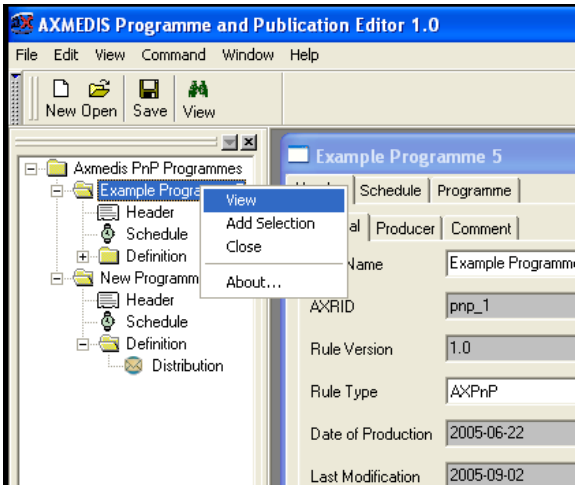
Editing functionalities include being able to drag and drop information from one programme to another using the workspace tree area as shown in the tree view in Screenshot 7 and the popup box in Screenshot 8 or using drop down menus by right clicking within the tree area as shown in Screenshot 9.



Screenshot 7



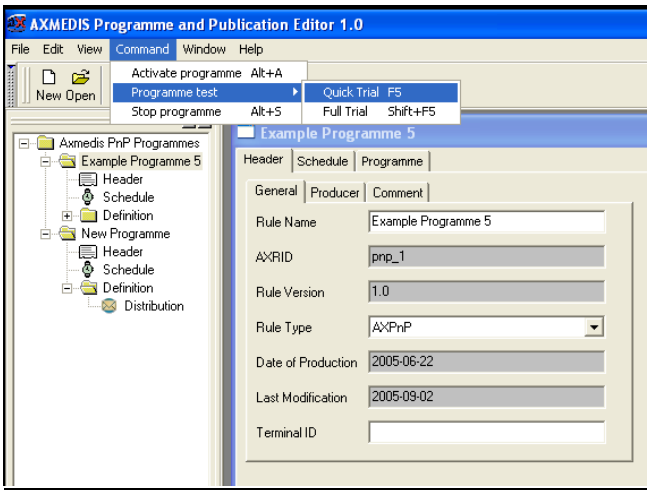
Screenshot 8



Screenshot 9

Testing a completed programme

Now the programme manager has created and edited his programme and believes it is ready for distribution, the programme manager has two options: the first is to run a quick trial; this ensures the programme devised is valid without any of the processing taking place (see Screenshot 10). This means request are made to AXMEDIS tools to make sure tasks can be accomplished for processing the programme. A message is return to the P&P Editor informing the user if the trial was successful or not. The second option is to run a full trial which enforces the P&P engine to request all the processing to be run if required with the exception of the final distribution to the distribution area. The processing jobs are requested and executed and on completion the results are returned. The final action of distribution is the only processing not requested in the full trial.



Screenshot 10

Activating and stopping a programme

When the programme manager is happy with his programme and validated it with a quick and/or full trial, the final option is to activate the programme. This sends the programme to the P&P Engine which processes the programme information and at the specified times distributes the correctly adapted multimedia object to the specified channel. During the lifespan of the active programme, the programme manager can also stop the programme from the P&P Editor by selecting "*Stop Programme*" in the Command menu. This can be seen in Screenshot 10.

Behind the scenes

On programme activation, selected objects to be sent to a particular channel will be checked to see if the channel can use the object. If the object is fine, then nothing is done and the object is ready to be sent via the specified distribution channel. However, if there is a mismatch, the AXMEDIS tools in the AXMEDIS Factory are used to resize or format the object so that it can be distributed – providing the rights allow the object to be adapted of course. With the object (new or reformatted) that can now be sent, the P&P Engine uses an internal clock to distribute the object at the correct time, as specified in the programme by the programme manager.

Conclusion

Using the P&P Editor, the programme manager can setup a programme that sends objects through a selected distribution channel. The programme manager can create a programme and edit it to set when and where the objects are to be delivered and also test the programme to check that everything is valid for distribution. Once checked, the activated programme is sent to the P&P Engine which uses AXMEDIS tools to automatically adapt the objects to be valid for the distribution channel and at the specified time sends the correctly adapted object to the distribution servers.

Learning Outcomes

At the end of this tutorial session, the programme managers will understand:

- How the P&P Area fits within the AXMEDIS architecture
- How to use the P&P Editor to create a P&P programme to distribute objects on selected channels.
- A programme manager will be able to make a programme and edit existing programme schedules;
- Test a programme
- Activate the programme to deliver objects.
- Stop a programme that has been activated through the P&P Editor.

6.6 Examples of usage

Double click on axpnpeditor.exe to run the P&P Editor application.

Set the configuration file if necessary (Menu File->Configuration...) to specify the parameters for Workflow Plug-in and the P&P Engine address.

6.7 Integration and compilation issues

6.7.1 How to get the source and place in the correct directories

The recommended way of getting the P&P Model Source is to obtain it directly from the AXMEDIS CVS repository <URL:https://cvs.axmedis.org/repos/>. The directory structure is important for compilation with the .NET project file. The recommended method is to checkout the Framework directory. The required header files, source files, libraries, .NET project file and required dll's to use the P&P Model are located within the repository as shown below.

```
Framework
  |- include
  |    |-pnpeditor
  |         |-*.h
  |- project
  |    |-pnpeditor
  |         |-win32
  |             |-axpnpeditorlib.vcproj
  |             |-HowToBuild.txt
  |- lib
  |    |-pnpeditor
  |         |-win32
  |             |- axpnpeditor.lib (Release)
  |             |- axpnpeditor_d.lib (Debug)
  |- source
       |-pnpeditor
            |-*.cpp
            |-bitmaps
               |- *.xpm
               |- *.bmp
```

6.7.2 Other requirements

The P&P Editor GUI requires wxWidget 2.4.2 and depends on the P&P Model and P&P Editor libraries. To download and install wxWidget, please refer to the AXMEDIS documentation or the following website

- wxWidget C - http://www.wxwindows.org/

The P&P Model libraries (axpnpmodel.lib and axpnpmodel_d.lib) and the P&P Editor (axpnpeditor.lib and axpnpeditor_d.lib) libraries and the respective include header files can be located in the CVS as shown in the following diagram.

```
Framework
  |- include
  |    |-pnpeditor
  |    |     |-*.h
  |    |-pnpmodel
  |         |-*.h
  |- lib
       |- pnpmodel
       |     |- win32
       |- pnpmodel
            |- win32
```

To build the libraries from source, see HowToBuild.txt in the CVS Framework

```
Framework
  |- project
```

```
|- pnpmodel
|    |- win32
|         |- HowToBuild.txt
|- pnpeditor
     |- win32
          |- HowToBuild.txt
```

6.7.3 Building the Source using Microsoft Visual C++

The .NET preferences are set to find the files in the directory structure as shown in part 1 and requires the environment variables to be set for wxWidget and Xerces. Control Panel -> System -> Advanced -> Environment Variable

- $(WXWIN) = where wxWindows-2.4.2 is installed
- $(LIBCURL) = where libcurl-devel-7.15.1 is installed
- $(MOZILLA) = where wxMozilla\mozilla is installed
- $(WSDLPULL) = where wsdlpull-1.9 is installed
- $(XERCES) = where xerces-c_2_6_0 is installed

NB if you are building Xerces from source you will have to set XERCESCROOT to where you have installed the source code for Xerces. Building the libraries can be done as a batch build or build individually in the build menu for the Debug and Release libraries.

6.7.4 Running the P&P Editor application

To run the application the following files need to be installed in the same directory:

- Xerces dll's depending on whether the last compilation was Debug or Release
  - xerces-c_2_6.dll
  - xerces-c_2_6D.dll
- schemas (required to validate the P&P XML files (.pp))
  - axpnprule.xsd
  - Selection-v1-6.xsd
- Sample PnP Programmes (not necessary)
  - AXPnPProgramme.pp
  - AXPnPProgramme2.pp
  - AXPnPProgramme3.pp

6.8 Configuration Parameters

Config parameter	Possible values
AXMEDIS_PLUGIN_MANAGER ▪ PLUGINS_PATH	 <*path-to-executable*>/plugin
DATABASE ▪ queryUrl ▪ user ▪ passwd ▪ LoaderWSEndPoint	These parameters are user definable in the configuration. There are no default parameters. <*address_ to_database_loader_endpoint* >
AXMEDIS_PNP_ENGINE ▪ SERVER_ADDRESS	 localhost:3000
WORKFLOW ▪ workflowUrl ▪ gatewayUrl	 http://<*workflow-address*>:8080/OpenFLow/index_html http://<*workflow-address*>:8080/responseGateway/reChannel/reChannel.asmx
PNP_REPOSITORY ▪ REPOSITORY_PATH	 <*path-to-pnp_repository*>

6.9 Errors reported and that may occur

Error code	Description and rationales

Connection Error: Unable to connect to P&P Engine	Called when P&P Programme is activated and the connection to the P&P Engine has failed
Report: Workflow is currently not present	Called when P&P Editor is run independent of Workflow

7 Module Tool Programme and Publication Rule Model

Module/Tool Profile		
Programme and Publication Rule Model		
Responsible Name	Kia Ng and Royce Neagle	
Responsible Partner	UNIVLEEDS	
Status (proposed/approved)	proposed	
Implemented/not implemented	Implemented	
Status of the implementation	Version 1.0	
Executable or Library/module (Support)	Library	
Single Thread or Multithread	Multithread	
Language of Development	C++	
Platforms supported	Microsoft Windows XP (Tested)	
Reference to the AXFW location of the source code demonstrator	https://cvs.axmedis.org/repos/Framework/pnpmodel/source https://cvs.axmedis.org/repos/Framework/pnpmodel/include	
Reference to the AXFW location of the demonstrator executable tool for internal download	Debug: https://cvs.axmedis.org/repos/Framework/pnpmodel/bin/win32/axpnpmodel_d.lib Release: https://cvs.axmedis.org/repos/Framework/pnpmodel/bin/win32/axpnpmodel.lib ACTIVITIES/WP4 and WP5http://www.axmedis.org/attivita/documenti/download.php?area_id=4&attivita_id=17&l_s=struttura&gruppo=0&order_by=data&asc_desc=desc/General on AXFW and Guidelines/ http://www.axmedis.org/documenti/view_documenti.php?doc_id=1617	
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)	present	
Test cases location	https://cvs.axmedis.org/repos/Framework/doc/test/pnpmodel/	
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	Name of the communicating tools	Communication model and format

tools, named as	References to other major components needed	(protected or not, etc.)
Formats Used	Shared with	format name or reference to a section
XML plus schema ▪ `axpnprule.xsd`	P&P Engine Executable	Programme and Publication Format with extension .pp
Protocol Used	Shared with	Protocol name or reference to a section
None		
Used Database name		
None		
User Interface	Development model, language, etc.	Library used for the development, platform, etc.
	Visual Studio 7 .NET	AXMEDIS Plug-in Manager
	C++	AXMEDIS Configuration Manager
		AXMEDIS common
		AXMEDIS pnpmodel
Used Libraries	Name of the library and version	License status: GPL. LGPL. PEK, proprietary, authorized or not
Xerces	XERCES version 2-6	
wxWidget	wxWindows version 2.4	

7.1 General Description of the Module

The P&P Rule Model provides the methods to manipulate the P&P Programme data. The data is separated into three sections. These sections are : the header (AxRuleHeader); the schedule (AxRuleSchedule); and the Arguments comprising of the AxPnPRule and AxRuleDistribution data.

P&P Programme Header (AxRuleHeader)

This section holds the following data

- P&P Programme Name (type: String) specifies the name of the programme
- P&P Programme ID (type: String) this is a unique value of 40 bytes with the first four bytes ("pnp_")
- P&P Programme Version (type: String) specifying which P&P Programme Rule version is used to create a P&P Programme
- P&P Rule (type: String) is always defined as AxPnP
- Software Name (type: String) storing the name of the software used to create the P&P Programme e.g. P&P Editor
- Software Version (type: String) storing the version of the software used to create the P&P Programme
- Production date when the P&P Programme was first created
- Author (type: String), Affiliation (type: String) and URL (type: String) to specify metadata information on the author of the P&P Programme
- Last Modification specifying the last time the P&P Programme was edited and saved
- Terminal ID (type: String) specifying the user's terminal the programme was last edited on.
- Cost (type: String). This is a redundant field used by the AXCP Rule.
- Work Item ID (type: String) is the field specifying the Workflow Job Request for a Programme. This includes request for a new P&P Programme, editing an existing Programme, and activating a P&P Programme. This is used when sending a notification of completion back to workflow
- Comment (type: String) is the field where the author of the P&P Programme can add comments.

P&P Programme Schedule (AxRuleSchedule)

- Firing Date is the start date for the P&P Programme to be activated by the P&P Engine
- Firing Time is the start time for the P&P Programme to be activated by the P&P Engine
- Expiration Date is the end date for the P&P Programme to be killed by the P&P Engine
- Expiration Time is the end time for the P&P Programme to be killed by the P&P Engine
- Periodicity includes the period (type: String) and units when a programme needs to repeat itself and how often
- Status (type: String) specifies the current status of the P&P Programme i.e. active, inactive, quick trial or full trial

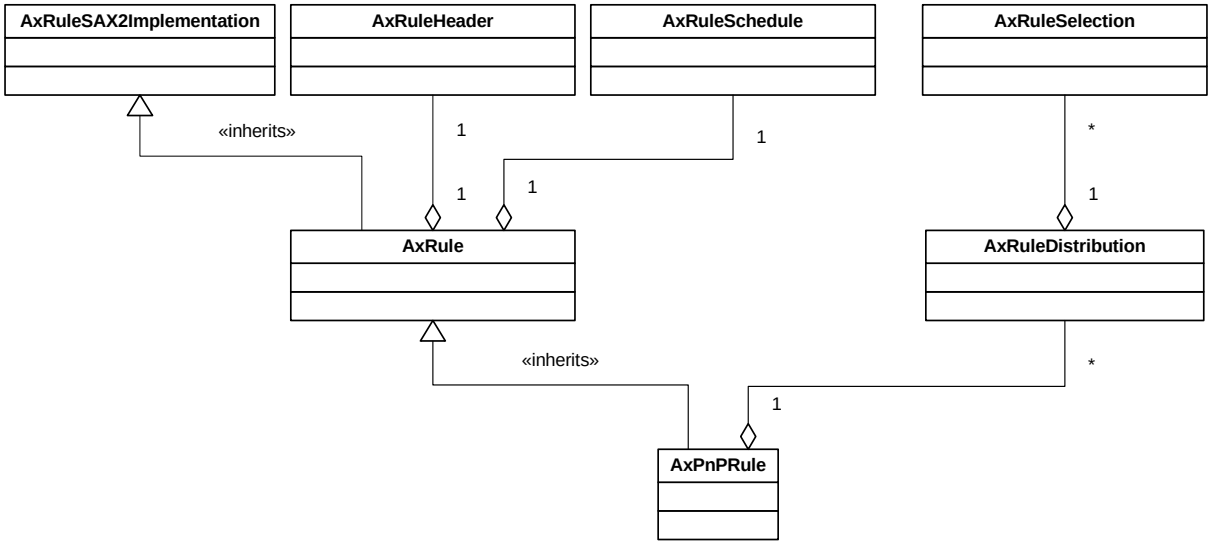
P&P Programme Arguments (AxPnPRule)

- Distribution List (P&P Programme Distribution) is a list of AXMEDIS objects to be distributed including information such as when, where, and how
- Global Distribution Channel ID (type: String) is the distribution channel to be used by the P&P Programme
- Global Distribution Terminal ID (type: String) is the distribution terminal to be used by the P&P Programme

P&P Programme Distribution (AxRuleDistribution)

- Start Date (type: Date) is the start date for the distribution of the AXMEDIS Object
- Start Time (type: Time) is the start time for the distribution of the AXMEDIS Object
- Duration Time is the length of time for the distribution of the AXMEDIS Object
- Distribution Channel ID is the channel the object is to be distributed if different from the global distribution specified in the section above
- Distribution Terminal ID (type: String) is the terminal the object is to be distributed if different from the global distribution specified in the section above
- Selection is an AXMEDIS Object or group of AXMEDIS Objects to be distributed
- AXOID is a single AXMEDIS Object to be distributed
- Description is the field where the author of the P&P Programme can add a description for each object to be distributed. See the comment filed in the Header section for adding comments concerning a P&P Programme.

7.2 Module Design in terms of Classes



7.3 User interface description

This is a library module that can be used by all partners for creating, editing and saving P&P Programmes. There is no user interface.

7.4 Technical and Installation information

References to other major components needed	▪ P&P Engine to upload a programme is needed to be running to activate and test a P&P Programme. ▪ Query Support User Interface must be running ▪ AXMEDIS Workflow can be running
Problems not solved	▪
Configuration and execution context	▪ Set the path for the Query Support User interface ▪ Set the Path for the Plug-in Directory to load the Workflow Plug-in ▪ Set the Engine and Port Address for the P&P Engine

7.5 Examples of usage

```
<?xml version="1.0" encoding="UTF-8"?>
<Rule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="axpnprule.xsd">
  <Header>
    <Rule_Name>New Programme</Rule_Name>
    <AXRID>pnp_84746afd-e022-4f5a-b834-6d9afd8e41e7</AXRID>
    <Rule_Version>1.0</Rule_Version>
    <Rule_Type>AXPnP</Rule_Type>
    <Software_Name>PnP Editor</Software_Name>
    <Version_of_software>1.0</Version_of_software>
    <Date_of_production>2006-03-15</Date_of_production>
    <Author></Author>
    <Affiliation></Affiliation>
    <URL></URL>
    <Comment></Comment>
    <Last_Modifications>2006-03-15</Last_Modifications>
    <Terminal_ID></Terminal_ID>
    <Cost></Cost>
    <Work_Item_ID></Work_Item_ID>
  </Header>
  <Schedule>
    <Run>
       <Date>2006-03-15</Date>
       <Time>14:53:38</Time>
       <Periodicity Unit="Day">0</Periodicity>
       <Expiration_Date>2006-03-15</Expiration_Date>
       <Expiration_Time>14:53:38</Expiration_Time>
     </Run>
     <Status>FTrial</Status>
   </Schedule>
  <Definition>
    <PnP_Rule>
      <Distribution>
        <DistributionChannelID></DistributionChannelID>
        <DistributionTerminalID>localhost</DistributionTerminalID>
      </Distribution>
      <Arguments>
        <Start_Date>2006-03-15</Start_Date>
        <Start_Time>14:54:04</Start_Time>
```

```
      <Duration_Time>00:00:00</Duration_Time>
      <ChannelID></ChannelID>
      <TerminalID></TerminalID>
      <Description>4142e15e-43a0-45ca-89d0-9d1d17725c87</Description>
      <AXOID>4142e15e-43a0-45ca-89d0-9d1d17725c87</AXOID>
    </Arguments>
  </PnP_Rule>
 </Definition>
</Rule>
```

7.6 Integration and compilation issues

7.6.1 How to get the source and place in the correct directories

The recommended way of getting the P&P Model source is to obtain it directly from the AXMEDIS CVS repository <URL:https://cvs.axmedis.org/repos/>. The directory structure is important for compilation with the .NET project file. The recommended method is to checkout the Framework directory. The required header files, source files, libraries, .NET project file and required dll's to use the P&P Model are located within the repository as shown below.

```
Framework
   |- include
   |    |- model
   |         |-*.h
   |- project
   |    |-pnpmodel
   |         |-win32
   |             |-axpnpmodel.vcproj
   |             |-HowToBuild.txt
   |- lib
   |    |-pnpmodel
   |         |-win32
   |             |- axpnpmodel.lib (Release)
   |             |- axpnpmodel_d.lib (Debug)
   |             |- xerces-c_2_6.dll
   |             |- xerces-c_2_6D.dll
   |- source
        |-pnpmodel
             |-*.cpp
```

7.6.2 Other requirements

The P&P Model requires wxWidget 2.4.2 and Xerces 2.6. To download and install please refer to the AXMEDIS documentation or the following websites

- wxWidget C - http://www.wxwindows.org/
- Xerces C++ - http://xml.apache.org/xerces-c/

7.6.3 Building the Source using Microsoft Visual C++

The preferences are set to find the files in the directory structure as shown in part 1 and requires the environment variables to be set for wxWidget and Xerces. Control Panel → System → Advanced → Environment Variable

- $(WXWIN) = where wxWindows-2.4.2 is installed
- $(XERCES) = where Xerces is installed

NB if you are building Xerces from source you will have to set XERCESCROOT to where you have installed the source code for Xerces.

Building the libraries can be done as a batch build or build individually in the build menu for the Debug and Release libraries.

7.7 Errors reported and that may occur

Error code	Description and rationales
`warning`	SAX Error handler specifying warning messages during the parsing of the P&P Programme from file or XML string. The XML P&P Programme is still parsed.
`Error`	SAX Error handler specifying error messages during the parsing of the P&P Programme from file or XML string.
`fatalError`	SAX Error handler specifying fatal error messages during the parsing of the P&P Programme from file or XML string.

7.8 Formal description of algorithm to sort the P&P Programme Distribution List

P&P Model	
Method	sortDistributionList
Description	This function sorts a P&P Distribution list for a P&P Programme into an order by specifying a distribution field e.g. start time, start date, duration, channel id etc.
Input parameters	**AxSortDistId** Identifier for the distribution field in a P&P Programme to be sorted • ID_SORTDATE: identifier for the distribution start date • ID_SORTTIME: identifier for the distribution start time • ID_SORTDURATION: identifier for the distribution duration • ID_SORTCHANNEL: identifier for the distribution channel • ID_SORTTERMINAL: identifier for the distribution terminal • ID_SORTNAME: identifier for the distribution name
Output parameters	No output parameters, this function sorts the distribution rule in the P&P Programme model. When saved or written to file, the P&P programme will have the new order for the distribution

This method utilises the wxWindows sort function for the wxList class and a P&P Distribution list compare function returning -1, 0 or 1 is implemented for the P&P programme distribution fields returning

- 0 is the fields are the same
- -1 where the first field is compared to be ordered before the second field
- and 1 where the first field is compared to be ordered after the second field

7.9 Formal description of algorithm to add a Distribution Rule to the P&P Programme

P&P Model	
Method	appendDistribution
Description	This function adds a new P&P Distribution rule to the P&P Programme.
Input parameters	**AxRuleDistribution** A distribution rule that is created
Output parameters	No output parameters

To utilise this method, the method requires a new distribution to be created. The pseudo code is:

1. Create new distribution rule
2. Add rules

3. Append distribution to the P&P Programme

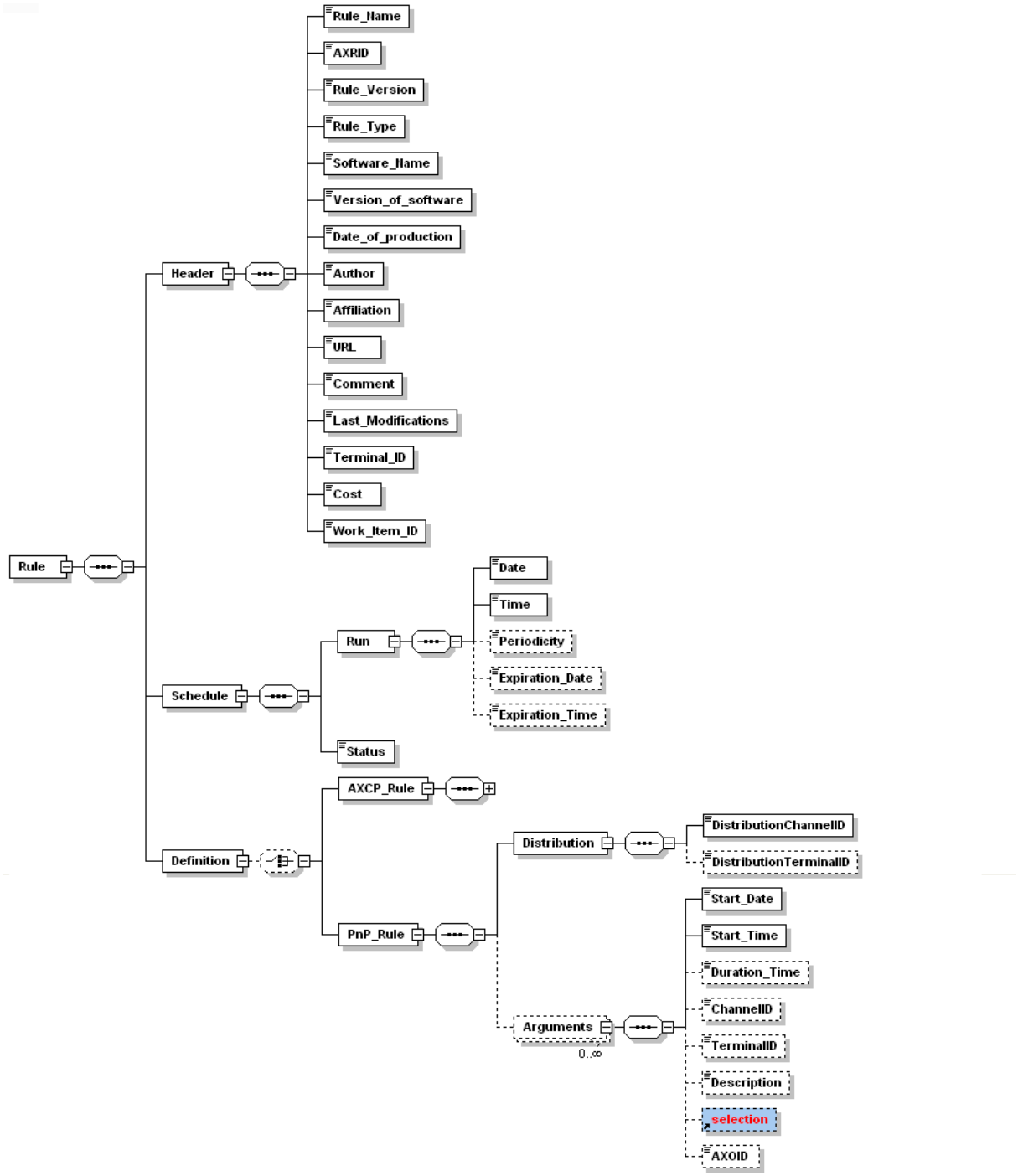
Code Example

```
AxRuleDistribution* newDistribution = new AxRuleDistribution();

newDistribution->setStartDate((wxDateTime)date_to_start_distribution);
newDistribution->setStartTime((wxDateTime)time_to_start_distribution);
…

pnpProgramme->appendDistribution(newDistribution);
```

8 Formal description of format P&P Programme

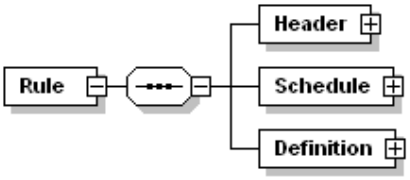


8.1 P&P Programme XML Schema

The following XML schema refers to the axpnprule.xsd

element **Rule**

diagram



children **Header** **Schedule** **Definition**

source

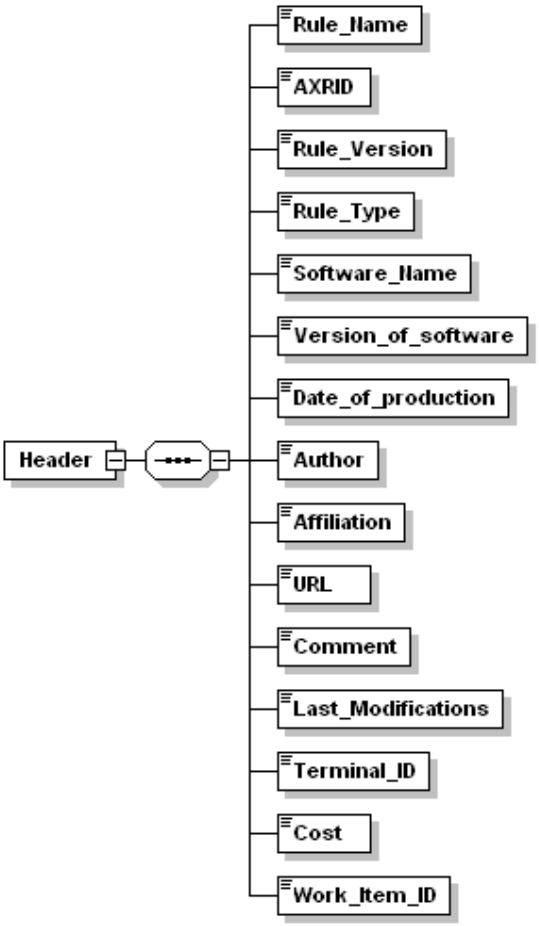
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 3 U (http://www.xmlspy.com) by UNIVLEEDS -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include schemaLocation="Selection-v1-6.xsd"/>
  <xs:simpleType name="periodunit">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Day"/>
      <xs:enumeration value="Month"/>
      <xs:enumeration value="Week"/>
      <xs:enumeration value="Year"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="Rule">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Rule_Name" type="xs:string"/>
              <xs:element name="AXRID" type="xs:string"/>
              <xs:element name="Rule_Version" type="xs:string"/>
              <xs:element name="Rule_Type">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="AXCP"/>
                    <xs:enumeration value="AXPnP"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Software_Name" type="xs:string"/>
              <xs:element name="Version_of_software" type="xs:string"/>
              <xs:element name="Date_of_production" type="xs:date"/>
              <xs:element name="Author" type="xs:string"/>
              <xs:element name="Affiliation" type="xs:string"/>
              <xs:element name="URL" type="xs:anyURI"/>
              <xs:element name="Comment" type="xs:string"/>
              <xs:element name="Last_Modifications" type="xs:date"/>
              <xs:element name="Terminal_ID" type="xs:string"/>
              <xs:element name="Cost" type="xs:string"/>
              <xs:element name="Work_Item_ID" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Schedule">
          <xs:complexType>
            <xs:sequence>
```

```
<xs:element name="Run">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Date" type="xs:date"/>
   <xs:element name="Time" type="xs:time"/>
   <xs:element name="Periodicity" minOccurs="0">
    <xs:complexType>
     <xs:simpleContent>
      <xs:extension base="xs:integer">
       <xs:attribute name="Unit" type="periodunit"/>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
   <xs:element name="Expiration_Date" type="xs:date" minOccurs="0"/>
   <xs:element name="Expiration_Time" type="xs:time" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="Status">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="Active"/>
   <xs:enumeration value="QTrial"/>
   <xs:enumeration value="FTrial"/>
   <xs:enumeration value="Inactive"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Definition">
 <xs:complexType>
  <xs:choice minOccurs="0">
   <xs:element name="PnP_Rule">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="Distribution">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="DistributionChannelID" type="xs:string"/>
         <xs:element name="DistributionTerminalID" type="xs:string" minOccurs="0"/>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
      <xs:element name="Arguments" minOccurs="0" maxOccurs="unbounded">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="Start_Date" type="xs:date"/>
         <xs:element name="Start_Time" type="xs:time"/>
         <xs:element name="Duration_Time" type="xs:time" minOccurs="0"/>
         <xs:element name="ChannelID" type="xs:string" minOccurs="0"/>
         <xs:element name="TerminalID" type="xs:string" minOccurs="0"/>
         <xs:element name="Description" type="xs:string" minOccurs="0"/>
         <xs:element ref="selection" minOccurs="0"/>
         <xs:element name="AXOID" type="xs:string" minOccurs="0"/>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
```

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

element **Rule/Header**

diagram



children Rule_Name AXRID Rule_Version Rule_Type Software_Name Version_of_software Date_of_production Author Affiliation URL Comment Last_Modifications Terminal_ID Cost Work_Item_ID

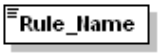
source

```
<xs:element name="Header">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Rule_Name" type="xs:string"/>
   <xs:element name="AXRID" type="xs:string"/>
   <xs:element name="Rule_Version" type="xs:string"/>
   <xs:element name="Rule_Type">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="AXCP"/>
      <xs:enumeration value="AXPnP"/>
```

```
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Software_Name" type="xs:string"/>
    <xs:element name="Version_of_software" type="xs:string"/>
    <xs:element name="Date_of_production" type="xs:date"/>
    <xs:element name="Author" type="xs:string"/>
    <xs:element name="Affiliation" type="xs:string"/>
    <xs:element name="URL" type="xs:anyURI"/>
    <xs:element name="Comment" type="xs:string"/>
    <xs:element name="Last_Modifications" type="xs:date"/>
    <xs:element name="Terminal_ID" type="xs:string"/>
    <xs:element name="Cost" type="xs:string"/>
    <xs:element name="Work_Item_ID" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

description This section contains metadata related to general information associated with a rule

element **Rule/Header/Rule_Name**

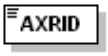
diagram 

type **xs:string**

source `<xs:element name="Rule_Name" type="xs:string"/>`

description It defines the name of the rule, e.g. “Audio Collection”

element **Rule/Header/AXRID**

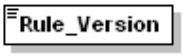
diagram 

type **xs:string**

source `<xs:element name="AXRID" type="xs:string"/>`

description It defines the AXMEDIS Rule ID

element **Rule/Header/Rule_Version**

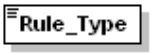
diagram 

type **xs:string**

source `<xs:element name="Rule_Version" type="xs:string"/>`

description It defines the version of the rule, e.g. “1.0”

element **Rule/Header/Rule_Type**

diagram 

type restriction of **xs:string**

facets
enumeration AXCP
enumeration AXPnP

source
```
<xs:element name="Rule_Type">
 <xs:simpleType>
  <xs:restriction base="xs:string">
```

```
      <xs:enumeration value="AXCP"/>
      <xs:enumeration value="AXPnP"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

description It defines the type of rule, AXCP rules identifies all rules related to the Content Processing Area, whereas the AXPnP rules are the rule of the P&P area

element **Rule/Header/Software_Name**

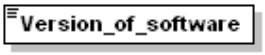
diagram 

type **xs:string**

source `<xs:element name="Software_Name" type="xs:string"/>`

description It specifies the name of software used, e.g. “AXMEDIS Rule Editor”

element **Rule/Header/Version_of_software**

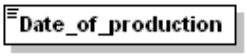
diagram 

type **xs:string**

source `<xs:element name="Version_of_software" type="xs:string"/>`

description It defines the version of software used., e.g. “2.0”

element **Rule/Header/Date_of_production**

diagram 

type **xs:date**

source `<xs:element name="Date_of_production" type="xs:date"/>`

description It defines when the rule has been created

Note This item now embeds the item Time_of_Production defined in the DE3-1-2C-AXFW_Spec-(content-production)-Part-C document.

:

element **Rule/Header/Author**

diagram 

type **xs:string**

source `<xs:element name="Author" type="xs:string"/>`

description It defines the name of author who has created the rule

element **Rule/Header/Affiliation**

diagram 

type **xs:string**

source `<xs:element name="Affiliation" type="xs:string"/>`

description It defines the name of Affiliation

element **Rule/Header/URL**

diagram 

type **xs:anyURI**

source <xs:element name="URL" type="xs:anyURI"/>

description It defines the Internet address/URL of the Affiliation

element **Rule/Header/Comment**

diagram 

type **xs:string**

source <xs:element name="Comment" type="xs:string"/>

description Of the P&P Programme

element **Rule/Header/Last_Modifications**

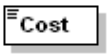
diagram 

type **xs:date**

source <xs:element name="Last_Modifications" type="xs:date"/>

description It is used to report the last modified date

element **Rule/Header/Cost**

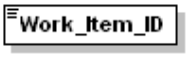
diagram 

type **xs:string**

source <xs:element name="Cost" type="xs:string"/>

description Estimation of Cost

element **Rule/Header/Work_Item_ID**

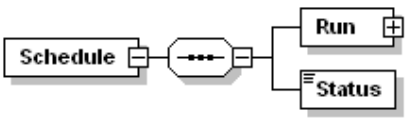
diagram 

type **xs:string**

source <xs:element name="Work_Item_ID" type="xs:string"/>

description External reference, for instance the commitment

element **Rule/Schedule**

diagram 

children **Run Status**

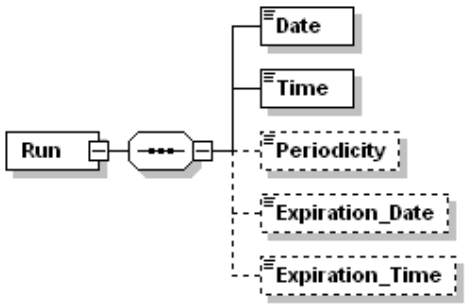
source <xs:element name="Schedule">

```
<xs:complexType>
 <xs:sequence>
  <xs:element name="Run">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="Date" type="xs:date"/>
     <xs:element name="Time" type="xs:time"/>
     <xs:element name="Periodicity" minOccurs="0">
      <xs:complexType>
       <xs:simpleContent>
        <xs:extension base="xs:integer">
         <xs:attribute name="Unit" type="periodunit"/>
        </xs:extension>
       </xs:simpleContent>
      </xs:complexType>
     </xs:element>
     <xs:element name="Expiration_Date" type="xs:date" minOccurs="0"/>
     <xs:element name="Expiration_Time" type="xs:time" minOccurs="0"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  <xs:element name="Status">
   <xs:simpleType>
    <xs:restriction base="xs:string">
     <xs:enumeration value="Active"/>
     <xs:enumeration value="QTrial"/>
     <xs:enumeration value="FTrial"/>
     <xs:enumeration value="Inactive"/>
    </xs:restriction>
   </xs:simpleType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
</xs:element>
```

description This section contains the sequence of metadata for programming the activation of a programme

element **Rule/Schedule/Run**

diagram



children **Date Time Periodicity Expiration Date Expiration_Time**

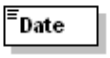
source

```
<xs:element name="Run">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Date" type="xs:date"/>
   <xs:element name="Time" type="xs:time"/>
   <xs:element name="Periodicity" minOccurs="0">
    <xs:complexType>
     <xs:simpleContent>
      <xs:extension base="xs:integer">
       <xs:attribute name="Unit" type="periodunit"/>
      </xs:extension>
     </xs:simpleContent>
    </xs:complexType>
   </xs:element>
   <xs:element name="Expiration_Date" type="xs:date" minOccurs="0"/>
   <xs:element name="Expiration_Time" type="xs:time" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
```

```
</xs:element>
```

description It defines a subsection of metadata that describes information needed for scheduling the execution of the rule.

element **Rule/Schedule/Run/Date**

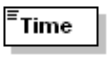
diagram 

type **xs:date**

source
```
<xs:element name="Date" type="xs:date"/>
```

description It defines when the rule has to be executed by the engine in terms of day, month and year.

element **Rule/Schedule/Run/Time**

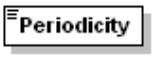
diagram 

type **xs:time**

source
```
<xs:element name="Time" type="xs:time"/>
```

description It defines when the rule has to be executed by the engine in term of time clock.

element **Rule/Schedule/Run/Periodicity**

diagram 

type extension of **xs:integer**

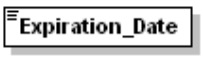
attributes

Name	Type	Use	Default	Fixed	Annotation
Unit	periodunit				

source
```
<xs:element name="Periodicity" minOccurs="0">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="Unit" type="periodunit"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

description It defines if a rule has to be executed periodically, every “Unit” “periodunit” e.g. “2” “Week”

element **Rule/Schedule/Run/Expiration_Date**

diagram 

type **xs:date**

source
```
<xs:element name="Expiration_Date" type="xs:date" minOccurs="0"/>
```

description The date to stop the periodicity

element **Rule/Schedule/Run/Expiration_Time**

diagram 

type **xs:time**

source
```
<xs:element name="Expiration_Time" type="xs:time" minOccurs="0"/>
```

description The time to stop the periodicity

element **Rule/Schedule/Status**

diagram

type restriction of **xs:string**

facets
enumeration Active
enumeration QTrial
FTrial
enumeration Inactive

source

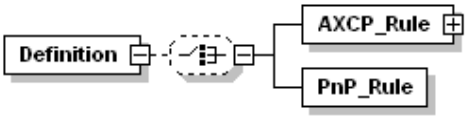
```
<xs:element name="Status">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="Active"/>
   <xs:enumeration value="QTrial"/>
   <xs:enumeration value="FTrial"/>
   <xs:enumeration value="Inactive"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
```

description It defines if a rule is:

- "active" and can be executed
- "quick trail" not to be executed but to ascertain that the programme can be run at execution time
- "full trial" is to executed all parts of the programme with the exception of distribution
- "inactive"

element **Rule/Definition**

diagram



children **AXCP_Rule PnP_Rule**

source

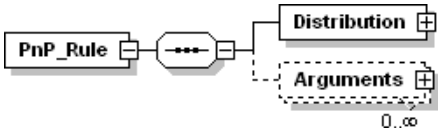
```
<xs:element name="Definition">
 <xs:complexType>
  <xs:choice minOccurs="0">
   <xs:element name="PnP_Rule">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="Distribution">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="DistributionChannelID" type="xs:string"/>
         <xs:element name="DistributionTerminalID" type="xs:string" minOccurs="0"/>
        </xs:sequence>
       </xs:complexType>
      </xs:element>
      <xs:element name="Arguments" minOccurs="0" maxOccurs="unbounded">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="Start_Date" type="xs:date"/>
         <xs:element name="Start_Time" type="xs:time"/>
         <xs:element name="Duration_Time" type="xs:time" minOccurs="0"/>
         <xs:element name="ChannelID" type="xs:string" minOccurs="0"/>
         <xs:element name="TerminalID" type="xs:string" minOccurs="0"/>
         <xs:element name="Description" type="xs:string" minOccurs="0"/>
         <xs:element ref="selection" minOccurs="0"/>
         <xs:element name="AXOID" type="xs:string" minOccurs="0"/>
```

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
 </xs:choice>
</xs:complexType>
</xs:element>
```

description This section includes the section containing the procedural description of an AXCP or a PnP rule

element **Rule/Definition/PnP_Rule**

diagram



children **Distribution Arguments**

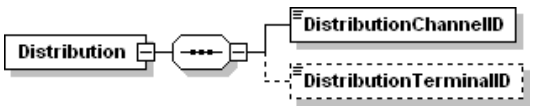
source

```
<xs:element name="PnP_Rule">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Distribution">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="DistributionChannelID" type="xs:string"/>
      <xs:element name="DistributionTerminalID" type="xs:string" minOccurs="0"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="Arguments" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="Start_Date" type="xs:date"/>
      <xs:element name="Start_Time" type="xs:time"/>
      <xs:element name="Duration_Time" type="xs:time" minOccurs="0"/>
      <xs:element name="ChannelID" type="xs:string" minOccurs="0"/>
      <xs:element name="TerminalID" type="xs:string" minOccurs="0"/>
      <xs:element name="Description" type="xs:string" minOccurs="0"/>
      <xs:element ref="selection" minOccurs="0"/>
      <xs:element name="AXOID" type="xs:string" minOccurs="0"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

description This section includes the section containing the procedural description of an PnP rule

element **Rule/Definition/PnP_Rule/Distribution**

diagram



children **DistributionChannelID DistributionTerminalID**

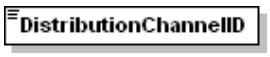
source

```
<xs:element name="Distribution">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="DistributionChannelID" type="xs:string"/>
```

```
    <xs:element name="DistributionTerminalID" type="xs:string" minOccurs="0"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
```

description This section includes the section containing the distribution channel identifiers for the PnP Rule

element **Rule/Definition/PnP_Rule/Distribution/DistributionChannelID**

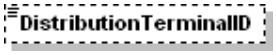
diagram 

type **xs:String**

source
```
<xs:element name="DistributionChannelID" type="xs:string"/>
```

description This section includes the section containing the distribution channel identifier string

element **Rule/Definition/PnP_Rule/Distribution**

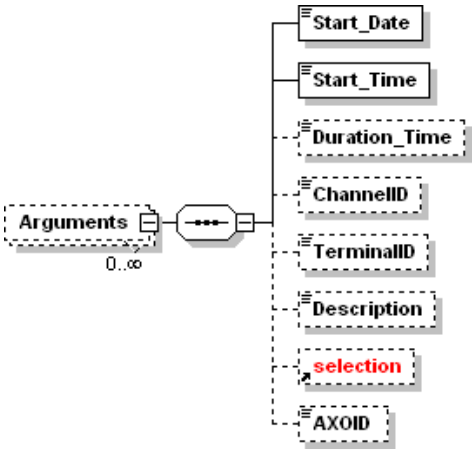
diagram 

type **Xs:String**

source
```
<xs:element name="DistributionTerminalID" type="xs:string" minOccurs="0"/>
```

description This section includes the section containing the distribution terminal identifier address for the PnP Rule. This is most likely to be an IP Address

element **Rule/Definition/PnP_Rule/Arguments**

diagram 

children **Start_Date Start_Time Duration_Time ChannelID TerminalID Description selection AXOID**

source
```
<xs:element name="Arguments" minOccurs="0" maxOccurs="unbounded">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="Start_Date" type="xs:date"/>
   <xs:element name="Start_Time" type="xs:time"/>
   <xs:element name="Duration_Time" type="xs:time" minOccurs="0"/>
   <xs:element name="ChannelID" type="xs:string" minOccurs="0"/>
   <xs:element name="TerminalID" type="xs:string" minOccurs="0"/>
   <xs:element name="Description" type="xs:string" minOccurs="0"/>
   <xs:element ref="selection" minOccurs="0"/>
   <xs:element name="AXOID" type="xs:string" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
```

description This section includes the section containing the description of a distribution argument specifying how, where and when an AXMEDIS object is to be distributed

element **Rule/Definition/PnP_Rule/Arguments**

diagram 

Type **xs:Date**

source <xs:element name="Start_Date" type="xs:date"/>

description This defines the start date for the distribution of an AXMEDIS Object or Selection

element **Rule/Definition/PnP_Rule/Arguments**

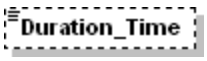
diagram 

Type **xs:Time**

source <xs:element name="Start_Time" type="xs:time"/>

description This defines the start time for the distribution of an AXMEDIS Object or Selection

element **Rule/Definition/PnP_Rule/Arguments**

diagram 

Type **xs:Time**

source <xs:element name="Duration_Time" type="xs:time" minOccurs="0"/>

description This defines the duration time for the distribution of an AXMEDIS Object or Selection

element **Rule/Definition/PnP_Rule/Arguments**

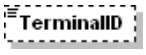
diagram 

Type **xs:String**

source <xs:element name="ChannelID" type="xs:string" minOccurs="0"/>

description This defines the distribution channel if different from the global distribution for the P&P Programme

element **Rule/Definition/PnP_Rule/Arguments**

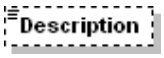
diagram 

Type **xs:String**

source <xs:element name="TerminalID" type="xs:string" minOccurs="0"/>

description This defines the distribution terminal if different from the global terminal id for the P&P Programme

element **Rule/Definition/PnP_Rule/Arguments**

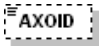
diagram 

Type **xs:String**

source <xs:element name="Description" type="xs:string" minOccurs="0"/>

description This element holds a description of the object defined by the user to describe the object such as name of the object etc

element **Rule/Definition/PnP_Rule/Arguments**

Diagram 

Type **xs:String**

Source <xs:element name="AXOID" type="xs:string" minOccurs="0"/>

description This element holds a description of the object defined by the user to describe the object such as name of the object etc

9 Formal description of communication protocol between the P&P Editor and Workflow

9.1.1 Program Publication User Interface method (Rule Editor Channel)

GET{ServiceURI}? Credentials="Credential string"
&AXRQID="Request ID string"
&Program_Name="Program Name string"
&URI="UserListenerService string"

Program_Name contains the P&P Programme XML to be edited
Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
URI is the URI where the Response Adapter waits for notifications from the Response Gateway

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Rule_editor_Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:boolean"/>
        <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
        <xs:element name="errorcode" type="xs:int"/>
        <xs:element name="programid" type="xs:string" minOccurs="0" maxOccurs="20"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

PnP User Interface	
Method	bool PnPUI();
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to invoke the PnP Program Editor to edit a program as specified by its name.
Input parameters	char *AXRQID, char *userCredentials, char *programName, char *URI
Output parameters	bool

Request Sample Message	`<message name="PnPUIRequest">`
	`<part name="ProgramName" element="s0:ProgramName"/>`
	`<part name="UserCredentials" element="s0:UserCredentials"/>`
	`<part name="AXRQID" element="s0:AXRQID"/>`

	<part name="REListenerService" element="s0:REListenerService"/> </message>
Response Sample Message	<xs:element name="REResult"> <xs:complexType> <xs:sequence> <xs:element name="Result" type="xs:boolean"/> <xs:element name="ErrorMSG" type="xs:string" minOccurs="0"/> <xs:element name="ErrorCode" type="xs:string"/> <xs:element name="ProgramList" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:complexType> </xs:element>

9.1.2 Activate Program Publication method (Rule Editor Channel)

GET{ServiceURI}? Credentials=”Credential string” &AXRQID=”Request ID string”
&ProgramID=”ProgramID string” &URI=”EngineListenerService string”
ProgramID contains the ID of the program to be activated.
Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
URI is the URI where the Response Adapter waits for notifications from the Response Gateway

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
 <xs:element name="Rule_editor_Response">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="result" type="xs:boolean"/>
    <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
    <xs:element name="errorcode" type="xs:int"/>
    <xs:element name="programid" type="xs:string" minOccurs="0" maxOccurs="20"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Activate PnP	
Method	bool activatePnP();
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to activate a program inside the PnP Engine through PnP Editor.
Input parameters	char *AXRQID, char *userCredentials, char* programID, char* URI
Output parameters	Bool
Request Sample Message	<message name="ActivatePnPRequest"> <part name="ProgramID" element="s0:ProgramID"/> <part name="UserCredentials" element="s0:UserCredentials"/> <part name="AXRQID" element="s0:AXRQID"/>

	<part name="ReListenerService" element="s0:REListenerService"/> </message>
Response Sample Message	<xs:element name="REResult"> <xs:complexType> <xs:sequence> <xs:element name="Result" type="xs:boolean"/> <xs:element name="ErrorMSG" type="xs:string" minOccurs="0"/> <xs:element name="ErrorCode" type="xs:string"/> <xs:element name="ProgramList" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:complexType> </xs:element>

9.1.3 Notify Completion Program Publication method (Rule Editor Response Channel)

Notify Completion	
Method	bool notifyCompletion();
Description	This method is invoked through a Webservice call going to the Workflow Request Gateway. This method will allow the P&P Editor to notify the completion of a job
Input parameters	char* AXRQID, char* URI, char* endpURL, bool result=true, char* errorCode, char* errorMSG, char* ProgramID, char* xmlRuleSchema
Output parameters	bool
Request Sample Message	<message name="RENotificationRequest"> <part name="AXRQID" element="s0:AXRQID"/> <part name="URI" element="s0:URI"/> <part name="Result" element="s0:Result"/> <part name="ErrorCode" element="s0:ErrorCode"/> <part name="ErrorMSG" element="s0:ErrorMSG"/> <part name="RuleSchema" element="s0:RuleSchema"/> </message>
Response Sample Message	<xs:element name="MethodResponse"> <xs:complexType> <xs:sequence> <xs:element name="Result" type="xs:boolean"/> <xs:element name="ErrorCode" type="xs:string" minOccurs="0"/> <xs:element name="ErrorMSG" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:complexType> </xs:element>

10 Formal description of communication protocol between the P&P Engine and Workflow

10.1.1 Status Request to Program and Publication method (Engine Channel)

GET{ServiceURI}? Credentials=”Credential string”&AXRQID=”Request ID string”
&ProgramID=”Program ID string”

Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
ProgramID is the ID of the Program whose status is to be returned

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Engine_Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:boolean"/>
        <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
        <xs:element name="errorcode" type="xs:int"/>
        <xs:element name="ruleid" type="xs:string" minOccurs="0" maxOccurs="20"/>
        <xs:element name="status" type="xs:string" minOccurs="0"/>
        <xs:element name="xml_rule_schema" type="xs:string" minOccurs="0"/>
        <xs:element name="rulelog" type="xs:string" minOccurs="0" maxOccurs="100"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Call name	
Method	bool StatusRequestPnP();
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to retrieve the status of the PnP engine.
Input parameters	char *AXRQID, char *userCredentials, char *ProgramID, std::string &status
Output parameters	bool
Request Sample Message	
Response Sample Message	

10.1.2 Suspend Program and Publication method (Engine Channel)

GET{ServiceURI}? Credentials=”Credential string”&AXRQID=”Request ID string”
&ProgramID=”Program ID string”

Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
ProgramID is the ID of the Program to be suspended

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Engine_Response">
    <xs:complexType>
      <xs:sequence>
```

```
      <xs:element name="result" type="xs:boolean"/>
      <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
      <xs:element name="errorcode" type="xs:int"/>
      <xs:element name="ruleid" type="xs:string" minOccurs="0" maxOccurs="20"/>
      <xs:element name="status" type="xs:string" minOccurs="0"/>
      <xs:element name="xml_rule_schema" type="xs:string" minOccurs="0"/>
      <xs:element name="rulelog" type="xs:string" minOccurs="0" maxOccurs="100"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
</xs:schema>
```

10.1.3 Abort Program Publication method (Engine Channel)

GET{ServiceURI}? Credentials=”Credential string”&AXRQID=”Request ID string”
&ProgramID=”Program ID string”
Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
ProgramID is the ID of the Program to be aborted

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="Engine_Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:boolean"/>
        <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
        <xs:element name="errorcode" type="xs:int"/>
        <xs:element name="ruleid" type="xs:string" minOccurs="0" maxOccurs="20"/>
        <xs:element name="status" type="xs:string" minOccurs="0"/>
        <xs:element name="xml_rule_schema" type="xs:string" minOccurs="0"/>
        <xs:element name="rulelog" type="xs:string" minOccurs="0" maxOccurs="100"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Call name	
Method	bool AbortPnPProgram();
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to abort a program in the PnP engine.
Input parameters	char *AXRQID, char *userCredentials,char *ProgramID
Output parameters	bool
Request Sample Message	
Response	

Sample Message	

10.1.4 Resume Program Publication method (Engine Channel)

GET{ServiceURI}? Credentials=”Credential string”&AXRQID=”Request ID string”
&ProgramID=”Program ID string”&URI=”EngineListenerService string”
Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
ProgramID is the ID of the Program to be resumed
URI is the URI where the Response Adapter waits for notifications from the Response Gateway

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Engine_Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:boolean"/>
        <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
        <xs:element name="errorcode" type="xs:int"/>
        <xs:element name="ruleid" type="xs:string" minOccurs="0" maxOccurs="20"/>
        <xs:element name="status" type="xs:string" minOccurs="0"/>
        <xs:element name="xml_rule_schema" type="xs:string" minOccurs="0"/>
        <xs:element name="rulelog" type="xs:string" minOccurs="0" maxOccurs="100"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Call name	
Method	bool ResumePnPProgram( );
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to resume a program in the PnP engine.
Input parameters	char *AXRQID, char *userCredentials, char *ProgramID, char *URI
Output parameters	bool
Request Sample Message	
Response Sample Message	

10.1.5 Activate Program Publication method (Engine Channel)

GET{ServiceURI}? Credentials=”Credential string”&AXRQID=”Request ID string”
&ProgramID=”Program ID string”&URI=”EngineListenerService string
Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
ProgramID is the ID of the Program to be activated
URI is the URI where the Response Adapter waits for notifications from the Response Gateway

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Engine_Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:boolean"/>
        <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
        <xs:element name="errorcode" type="xs:int"/>
        <xs:element name="ruleid" type="xs:string" minOccurs="0" maxOccurs="20"/>
        <xs:element name="status" type="xs:string" minOccurs="0"/>
        <xs:element name="xml_rule_schema" type="xs:string" minOccurs="0"/>
        <xs:element name="rulelog" type="xs:string" minOccurs="0" maxOccurs="100"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Call name	
Method	bool ActivatePnPProgram();
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to activate a program in the PnP engine.
Input parameters	char *AXRQID,char *userCredentials, char*ProgramID, char *URI
Output parameters	bool
Request Sample Message	
Response Sample Message	

10.1.6 Workflow Notification method (Engine Response Channel)

GET{EngineListenerServiceURI}? Credentials=”Credential string”&AXRQID=”Request ID string”
&Status=”Status string”

Credentials contains user Credentials.
AXRQID is the unique identifier to the Request, to be returned in following notifications from the Response Gateway
Status is a string containing the Notification from the WorkFlow process

The response to the invoked method is sent via an http GET response. The response is XML coded, following the schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Engine_Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="result" type="xs:boolean"/>
        <xs:element name="errormsg" type="xs:string" nillable="true" minOccurs="0"/>
        <xs:element name="errorcode" type="xs:int"/>
        <xs:element name="ruleid" type="xs:string" minOccurs="0" maxOccurs="20"/>
        <xs:element name="status" type="xs:string" minOccurs="0"/>
        <xs:element name="xml_rule_schema" type="xs:string" minOccurs="0"/>
        <xs:element name="rulelog" type="xs:string" minOccurs="0" maxOccurs="100"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Call name	
Method	bool WFNotification();
Description	This method is invoked through a Webservice call coming from Workflow Request Gateway. This method will allow the workflow engine to send the notification to the PnP engine for the previously issues request to activate a process.
Input parameters	char *AXRQID, char *userCredentials, char*status
Output parameters	bool
Request Sample Message	
Response Sample Message	

11 Bibliography (mandatory)

[1] K. Ng, B. Ong, R. Neagle, P. Ebinger, M. Schmucker, I. Bruno, and P. Nesi, "AXMEDIS Framework for Programme and Publication and On-Demand Production" in *Proceedings of the 1st International Conference on Automated Production of Cross Media Content for Multi-channel Distribution* (AXMEDIS2005), December 2005, Florence Italy, pp. 247-250.

[2] K. Ng, B. Ong, P. Nesi, and P. Bellini "Automated Multi-Channel Cross-Media Production and Distribution" in *Proceedings of the EVA 2005 London Conference,* July 2005, London, pp25-29.

[3] W3C CC/PP Information Page, July 2004 http://www.w3.org/Mobile/CCPP/, Last accessed April 2006

[4] W3C, Web Services Activity, March 2006, http://www.w3.org/2002/ws/, Last accessed April 2006.

[5] P. Bellini, I. Bruno, and P. Nesi, "A Distributed Environment for Automatic Multimedia Content Production based on GRID" in *Proceedings of the 1st International Conference on Automated Production of Cross Media Content for Multi-channel Distribution* (AXMEDIS2005), December 2005, Florence Italy, pp 134-142.

[6] M. Butler, DELI: A Delivery context Library for CC/PP and UAProf, Technical Report HPL-2001-269, August 2002, http://www.hpl.hp.com/personal/marbut/ DELIUserGuideWEB.htm. Last accessed April 2006.

[7] Openflow Workflow homepage www.openflow.it last accessed April 2006

[8] Microsoft BizTalk server homepage http://www.microsoft.com/biztalk/ last accessed April, 2006

[9] A. Badii, M. Sailor, C. Marangoni, A. Neglia, L. Pearce, "Object Life Cycles & Interaction Environments for Cross Media Production & Distribution" in *Proceedings of the 1st International Conference on Automated Production of Cross Media Content for Multi-channel Distribution* (AXMEDIS2005), December 2005, Florence Italy, pp 279-286.

[10] A. Badii, M. Sailor, C. Marangoni, A. Neglia, L. Pearce, "Workflow Data Exchange Semantics, Object Discovery & Integration Architecture for Cross Media Production & Distribution" in *Proceedings of the 1st International Conference on Automated Production of Cross Media Content for Multi-channel Distribution* (AXMEDIS2005), December 2005, Florence Italy, pp 101-108.

[11] ISO/IEC 21000-7:2004, "Information Technology — Multimedia Framework (MPEG-21) — Part 7: Digital Item Adaptation," 2004

[12] A. Badii, M. Sailor, R. Nair, Profiling Management for Personalised Multimedia Delivery On-Demand within the AXMEDIS Framework, to appear in *Proceedings of the 2nd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution* (AXMEDIS2006), December 2006, University of Leeds, UK.

12 Glossary

12.1 List of AXMEDIS Acronyms for Identification

Term	Meaning
AXOID	unique AXMEDIS Object ID
AXPnP	Defines the AXRID as a P&P Programme
AXRID	unique AXMEDIS Rule ID

12.2 List of AXMEDIS Acronyms for Tools

Term	Meaning
AXCP	AXMEDIS Content Processing
AXDBM	AXMEDIS Database Manager
AXDOM	AXMEDIS Domain (associated to the PMS Home managing the Domain)
AXFW	AXMEDIS Framework
AXMEDIS	the project
AXOB	AXMEDIS Object, they does not contain the license. An AXOB with a license is presently called Governed AXOB.
AXOM	AXMEDIS Object Manager
AXQS	AXMEDIS Query Support
AXWFM	AXMEDIS workflow Manager
GAXOB	Governed AXMEDIS Object
P&P	AXMEDIS Programme and Publication

12.3 AXMEDIS Programme and Publication (P&P) Glossary of Terms

Term	Meaning
P&P Editor	AXMEDIS Programme and Publication Editor for creating, manipulating and saving Programme and Publication programmes
P&P Engine	AXMEDIS Programme and Publication Engine for the processing of activated P&P Programmes for distribution. It is connected to the AXMEDIS distribution servers and uses AXMEDIS Factory resources for AXMEDIS Content Processing area and for retrieving AXMEDIS Objects
P&P Engine Monitor	A GUI tool for monitoring the processes of the P&P Engine, finding activated and pending P&P Programmes.
P&P Programme	Programme and Publication rule(s) in XML format specifying a set of naccessary parameters for the distribution of the specifiied objects such as start time and date, AXMEDIS object ID, distribution channel, etc.
XML	Extensible Markup Language

12.4 AXMEDIS Workflow Management Systems Glossary of Terms

Term	Meaning
AXRQID	A composite parameter as a subtype of WF-Exchange_ID that can be passed between the AXWF and relevant AXMEDIS components to help identify a particular WF-Exchange_ID.
Exchange_Instance	A particular exchange messaging act or communication between any relevant AXMEDIS components and AXWF within a workflow-instance.
Response_ID	A composite parameter as a subtype of WF-Exchange_ID that can be passed from any relevant AXMEDIS components to help identify a particular WF-Exchange_ID as a response to a particular AXRQID.