An Integrated Framework for Automating Content Enrichment, Packaging and Distribution with DRM support

Pierfrancesco Bellini, Ivan Bruno, Paolo Nesi, Davide Rogai
DISIT-DSI, Distributed Systems and Internet Technology Lab
Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze
Via S. Marta, 3, 50139 Firenze, Italy
http://www.AXMEDIS.org, nesi@dsi.unifi.it, http://www.dsi.unifi.it/~nesi
Office: +39-055-4796523, Fax: +39-055-4796363

Abstract

Content enrichment is one of the most expensive activities related to the integration and augmentation of information of content elements. This activity is typically performed for cultural and educational content and recently also for edutainment and infotainment. AXMEDIS Framework aims at providing technical solutions and tools for reducing costs by automating content production, protection and distribution, thus supporting activities of content enrichment. The AXMEDIS solution is based on GRID technology and provides an efficient management of content processing and finalization as well as for dynamic service discovery and composition, distributed resource management and adaptive media delivery. AXMEDIS is a large IST FP6 Integrated Project of Research and Development, in which a set of enabling technologies are developed to cope with the above mentioned problems.

1. Introduction

The market of digital content for entertainment and infotainment is urging new models for integrating different content aspects and in particular simple resources with semantic and knowledge. These activities are in many cases called of content enrichment or augmentation, and in details include metadata processing, annotation, extraction of descriptors, definition of relationships, etc., and then of content formatting and layouting to allow exploiting and rendering the aggregated information in some suitable manner and not only for the indexing into a database and search, but also for the usage from the final user that can be interested to see and navigate on semantic annotations. Complete the above process, the activities of content packaging, protection and distribution of the final content based product to the final users and/or to other actors for further enrichment or for exploiting the enriched content for creating more valuable content products in a business to business (B2B) environment.

Therefore, in the view of creating high value content, most of the above mentioned activities may be iterative and may be very expensive to be performed for the presence of the humans in the loop and for the computational complexity of processing huge about of digital data such video, audio files, documents, multimedia, images, etc. One of the major costs and problems, that lead to involve the human intervention, are related to the management of the IPR (Intellectually Property Right) management and clearance. For example the activity of requesting and obtaining the authorization to use some content elements to enrich other content: a singer biography for an audio track, an image to enrich an audio file, a textual description of an image, etc.

For this reason, content providers, aggregators and distributors need flexible solutions for reducing production costs. These goals can be reached by acting on a set of factors that impact on: content modeling, content processing, content protection, digital rights management, interoperability of content, component based architecture, etc.

In order to manage the complexity of content enrichment and processing for massive content production, suitable architectures have to be set up to allow the fast and dynamic allocation of processes and their supervision. On this purpose a GRID technology can be used [1], [2], [10]. Although, there is no widely agreed definition what GRID means, the consequences for scientific and research work, and also for the private users are relevant. GRID computing is a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations. GRID is already being successfully used in many scientific applications where huge amounts of data have to be processed in

reasonable time and/or stored. According to the work described in this paper, the GRID Computing is declined to the production and management of multimedia content. It is probable that a large number of multimedia services (music, video, radio, television, etc.) are changing towards real-time on-demand services: VOD, IPTV, DVB-H, I-TV, etc. Crucial technical issues of providing access to such services are user-friendliness, universal access to services, as well as an efficient service GRID middleware that enables the dynamic service discovery and composition, distributed resource management and adaptive media production and delivery according to the user needs and context.

This paper describes how the AXMEDIS Framework and solution can address the above mentioned problems in an integrated manner. AXMEDIS framework provides a set of innovations, technical solutions and tools to allow the automation of cross media content production, packaging, protection and distribution, including enrichment and integration [3], [7], [8], [9]. AXMEDIS is based on and extended a set of standards: MPEG-21 [5], [6], XML, WSDL, XSLT, SMIL, OMA, etc.

AXMEDIS is a large Integrated Project of Research and Development partially founded by the European Commission in IST FP6 and includes more than 35 partners (core and affiliated) among them (in any specific order): University of Florence, HP, EUTELSAT, TISCALI, University Politecnic of Catalogna, University of Leeds, Telecom Italia, BBC, Telecom Lituania, Giunti Ilabs, SIAE, SDAE, ELION, ETRI, EPFL, FHGIGD, ACIT, AFI, STRATEGICA, EXITECH, XIM, University of Reading, Accademia Nazionale di Santa Cecilia, etc. More technical information including about how to make registration and affiliation to AXMEDIS can be recovered on www.axmedis.org In addition, AXMEDIS is going to be used in VARIAZIONI eContentPlus project on content enrichment. The VARIAZIONI project includes many other cultural institutions such as ALBENIZ, Sibelius Academy, etc., and technical partners GERMINUS, University Pompeo Fabra, University of Florence, etc. (see [4]).

2. AXMEDIS General Overview

In Figure 1, the technical architecture of AXMEDIS is reported. The whole architecture is based on a set of facilities organized in the so called the **AXMEDIS model supports and plugins**. It includes a set of core technologies and support modules at the basis of the whole set of the AXMEDIS tools for authoring, administration, content distribution, content playing, and content sharing via P2P and other traditional channels. Almost all the AXMEDIS tools (for content authoring, distribution and processing) can be (i) manually used by end-users and/or (ii) remotely controlled by means of external tools via dedicated Web Services (WSs) to automate their activities. This permits to create a large variety of architectures, including also those based on some supervising control based on (i) Workflow Management Systems such as Open Flow and in the future also with BizTalk or (ii) some higher level AXMEDIS GRID, also called AXMEDIS Content Processing, AXCP.

An **AXMEDIS Content Factory** can be built on the basis of AXMEDIS tools in a scalable and flexible manner. Tuning for example, AXCP GRID size, database size and type, number of authoring tools, number and type of tools/algorithms and libraries for content processing, licenses, integration support based on Workflow or not, etc.

In the rest of this section, the most relevant areas of AXMEDIS are described while the AXCP aspects and capabilities are better described in the following sections.

AXMEDIS Distribution tools for automating the content publication and acquisition/sharing in the business area allowing the interconnection of AXMEDIS Content Production Factories by means of the AXEPTools which is a secure and legal P2P tool. Among connected AXMEDIS Content Production Factories, it is also possible to make distributed queries to search for content, and to automatically publish and acquire/update content from/to the business partners, etc. The tools in this area also allow scheduling content distribution and publication towards external web services for example those of front end distribution servers. The distribution tools include the AXMEDIA P2P for distributing and sharing content among final users by means of secure P2P network by using BitTorrent technology.

AXMEDIS Players for content playing and execution on several different platforms, to built specific and customized content players for PC, PDA, and mobiles, and as Active X or plugins for other tools for enforcing the DRM.

AXMEDIS DRM Servers which are protection and supervising tools for registering users, certificating users, authenticating devices and tools, monitoring all the activities on the AXMEDIS content on AXMEDIS players and tools, processing licenses, managing black lists, and collecting and reporting the information about content usage and rights exploitation, etc.

Workflow Management tools include a set of micro tools and interfaces which are pervasively connected to all the AXMEDIS tools and plug-ins to allow

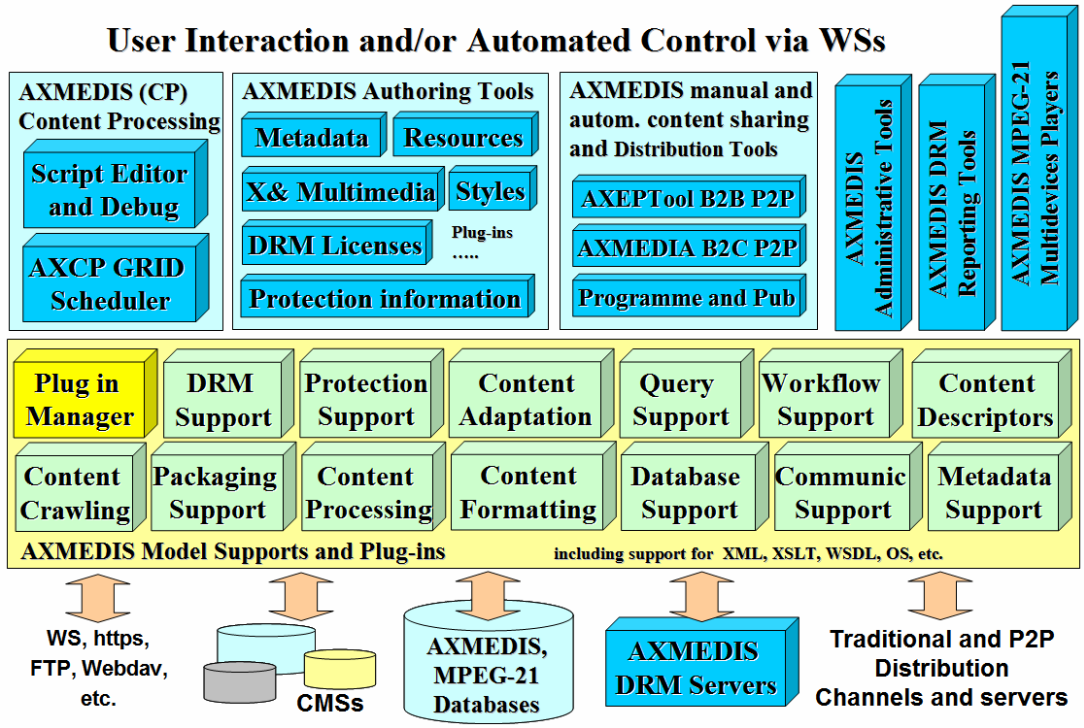

Fig.1 – AXMEDIS Technical Architecture.

interfacing the whole content factory to Workflow tools such as Open Flow and BizTalk.

AXMEDIS Database includes the AXMEDIS/MPEG-21 database model, supporting the storage and access to AXMEDIS content via a set of metadata for each object called AXInfo that can be customized. The database also includes produced licenses for the objects, history of performed actions on content, potentially available rights for each digital resource, models of contracts, etc.

AXMEDIS Accounting tools allow content producers, distributors or collecting societies to collect administrative information and reports about their content in order to gathering information about the list of rights that have been exploited on their AXMEDIS objects by the final users and by the business users.

3. Content Enrichment problems

On the basis of the reported technical architecture (see Fig.1), the AXMEDIS framework can be used to set up several different configurations of AXMEDIS Factories for content enrichment and production.

In Fig. 2, a simplified flow of a possible content enrichment process is depicted to put in evidence the problems and the differences of performing the content enrichment on protected and non protected content elements.

In Fig.2, the flow starts with the collection/ingestion of non protected content and metadata. In AXMEDIS, as well as in other architectures, the content ingestion can be performed automatically collecting/crawling content and metadata from legacy CMSs (Content Management Systems). Once the content is ingested, the content elements and information are indexed and the AXMEDIS Factory can be used for:

- producing content in some integrated format and/or package (integrating/relating metadata, annotations, and digital resources, each other) and indexing, storing/organizing them in some database,
- early content enrichment by adding manually and/or automatically: (i) semantic descriptors, (ii) extracting content descriptors for video, audio, document, multimedia, images, etc. (performing adaptation and content processing when needed), (iii) annotations and associating them to content and its internal structure and information (segmentation, labeling, etc.),

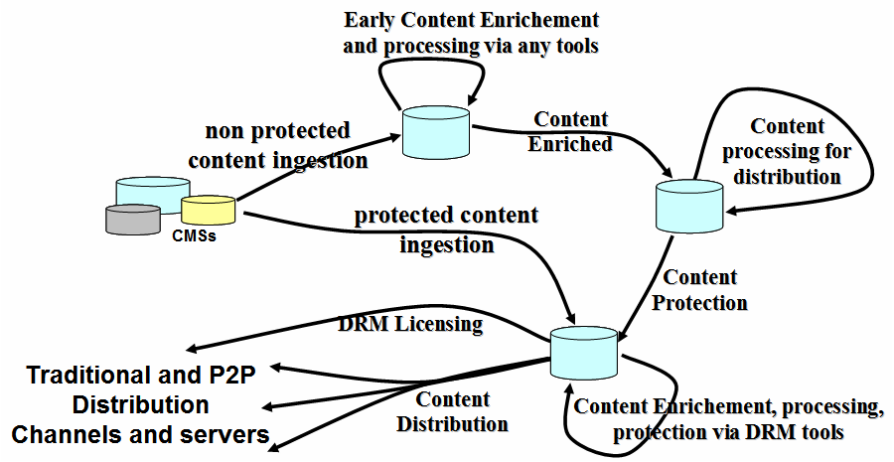

Fig.2 – Enrichment and distribution of protected and non protected content

- storage of the enriched content in some database,
- addition of some other clued content for preparing the enriched content for usage, for example synchronization with SMIL, SVG or integration with HTML, etc.
- preparing content for distribution according to different preferred formats (adapting the content format and changing its layout according to distribution channel profile, device profile, network profile and also user profile), for one or more different:
    - devices/tools such as: PC, PDA, Mobiles, STBs, etc.,
    - distribution channels such as: satellite, internet, mobile, P2P, etc.,
- Content Protection: protecting content according to some Protection Information model that may depend on the channel and device and user (MPEG-21 IPMP, Window Media, etc.),
- DRM Licensing: applying DRM (Digital Rights Management) rules according to IPR information and business models obtained from some administrative CMS (Content Management System), defining and issuing licenses (may be in different formats) for its usage on AXMEDIS tools (i.e., processing, authoring, distribution and players tools),
- Content Distribution: performing the effective distribution: broadcasting, P2P content sharing, etc.

In the above described process, it has to be noted that the ingested content can be early enriched with annotations, semantic information, multimedia and metadata information, including also IPR and licensing information. While the respect of IPR information on content distribution is enforced by means of content protection technologies and formalized in some DRM licenses. This allows to specify for each specific content element who is authorized to do what and with which constraints (time, location, etc.). This approach allows creating a package that can be shared among different peoples and experts for manual and/or semiautomatic content enrichment in the respect of the IPR.

Thus, the protected content can be manipulated and further enriched only by authenticated and certified tools and according to the produced license that formalize the actions/rights that can be performed on each specific content element by each specific actor of the content enrichment process. Thus the ingestion of protected content has to follow a differed path (see Fig.2).

AXMEDIS solves the above problem providing an integrated framework and a number of tools for content processing and enrichment that are intrinsically supported by DRM tools. The AXMEDIS framework is mainly based on MPEG-21 [5], [6], and OMA as cross-media models, for packaging and DRM, and provides tools for content sharing and distribution at B2B level, and it is capable to work with a multichannel architecture for B2C distribution for the production of content on demand [8], [9]. The AXMEDIS cross media content model attempts to wrap any digital resource in a container to make them ready for delivering by using a large range of business and transaction models and supporting them with some DRM.

4. Content Processing Language Features

The AXMEDIS Content Processing Area and the related language have been designed to provide a set of features and capabilities for automating content production and processing.
The processing algorithms are specified in terms of script code (Spider Monkey) allowing the manipulation of complex AXMEDIS data types and simple digital resources and content in general, and for the direct access to the AXMEDIS database and processing queries with the help of the AXMEDIS Query Support. The available data types, operators and accessible algorithms allow manipulation of any digital resources in a large number of formats for: images, audio files, video, documents, multimedia (including MPEG-4, HTML, SMIL, LOM, etc.), plus MPEG-21 aspects, MPEG-21 DI, DID, REL, IPMP, etc.

As regard the processing capabilities, an AXCP Rule formalises in its language features to perform the following activities on content.

Content Retrieval and Storage, Ingestion and Gathering from

- Content Management Systems, from file system, or protocols: ODBC, MySQL, ORACLE, MS-SQL, etc., files systems, http and ftp accesses;
- Many different CMSs via Focuseek Crawler;
- AXMEDIS MPEG-21 database;
- other AXMEDIS content Factories by means of the P2P tools, namely AXEPTool;

Content and Metadata Processing such as

- metadata manipulation, mapping and adaptation;
- content adaptation, extraction of descriptors, transcoding, synchronisation, estimation of fingerprint, watermarking, indexing, document summarization, keywords extraction;
- audio-video synchronisation, segmentation, labelling of content and definition of relationships;

Content Composition and Enrichment for

- processing resources and coupling/annotating them with metadata and semantic information;
- packaging content elements in AXMEDIS, MPEG-21 and OMA formats;
- creation of content components or objects as a combinations of raw assets such as Text, Images, Audio, Video, Animation, metadata, descriptors, licenses, multimedia objects such as MPEG-4, HTML, SMIL, macromedia tool files, games, etc.;
- creation of content as linear, nested or hierarchical combination of content components;

Content Formatting

- gluing content elements together by means of SMIL based templates and applying style sheets (XSLT) to define the usage interface (format, layout) of the content collection and the interested content usage paradigms (leaving open some parameters). For example, karaoke, collection browsing, selection menus, sliding presentation, stable background with a window with live video, animated text moving on an image, running text, etc.;
- optimization of styling parameters left open or defining them manually to arrange for example: best fitting of images in the screen, optimizing the amount of text in the page, best time fitting, etc.;

Content Protection such as:

- Protection of digital resources and full objects with their complex structure;
- creation of Protection Information parameters (IPMP of MPEG-21, OMA);
- Applying Protection Information model to AXMEDIS object such as, segmenting digital resources, applying encryption, scrambling;
- Posting specific protection information for each given object to the DRM servers;

Content Licensing

- Generating license from license model and additional information, storing licenses, posting them on license server automatically;
- Transcoding/translating licenses from MPEG-21 REL to ODRL OMA formats;
- verification of licenses against available rights to simulate the usage from the user site;

Content Publication and Distribution towards

- any distribution channel, producing programme and its schedule;
- P2P network of other AXMEDIS Factories of content integrators, producers, and distributors.

Have to be highlighted that the AXCP Language provides a way to use C++ plug-ins and to call generic WebServices. These features can be profitably used in content enrichment, for example calling external WebServices (e.g. search engines, metadata archives, etc.) that can be used to add new metadata or to add new content. Moreover specific plug-ins can analyze the content using specific algorithms (image analysis, audio analysis) to infer properties that can be used to enrich the content itself.

5. AXMEDIS Content Processing Tools

In order to exploit the above features, the AXMEDIS Content Processing Area is governed by a set of tools as depicted in Fig.3 and described as follows.

AXMEDIS Rule Editor (see Fig.4): it is capable to produce, debug, test and validate the executable AXCP Rules. An AXCP Rule is:

- written using the AXCP language for content production which is an extension of Java script;
- tested, debugged and validated on the AXMEDIS Rule Editor;
- activated for content processing on any AXCP GRID Node or on a single computer;
- used/parameterized for producing content on demand or to be integrated in your content factory;
- activated from your Workflow Manager engine via web service;

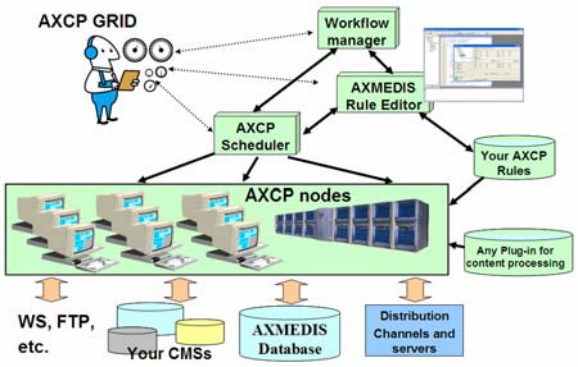

Fig. 3 -- AXCP Architecture

The AXCP GRID consists of a set of general purpose or specialized computers (called AXCP GRID Nodes or Engines) to execute AXCP Rules governed by the AXCP Rule Scheduler.

The processing capabilities and functionalities reported in the previous section and exploited from AXCP Tools and Rules are expandable by realizing or installing a set of additional plug-ins, by using the AXMEDIS Plug-in descriptor and solution.

5.1 AXCP Rule Scheduler

The AXCP Scheduler performs scheduling of AXCP rules in GRID nodes according to the content production and processing needs in terms of time and resources. It performs the rule firing/activation, rule executor discovering and management and rules dispatching. It is capable to receive commands for remote control and provide reporting (notifications, exceptions, logs, etc…) via a web service.

The AXCP Rule Scheduler manages AXCP rules and the available AXCP GRID Nodes. Each of them has a corresponding counterpart model on the Scheduler side to represents its capabilities and status. Knowing the availability and capabilities of each GRID Node is mandatory to verify the suitability of the computer that will execute the rule. To this end, the association of Rule to a GRID Node is based on the analysis of their profiles/capabilities.

Each AXCP GRID Node provides its profile/capabilities descriptor during its first connection and discovery/connection in the GRID network, containing:

- Execution status: running, idle, etc
- Identity and information of the executor: computer name, IP address, location, etc.
- Capabilities: CPU, RAM, Clock, Disk Space, OS, available workload, network cost for the communication with the database, etc.
- Tools Provided: AXMEDIS Plug-Ins installed and External tools Plug-In installed.

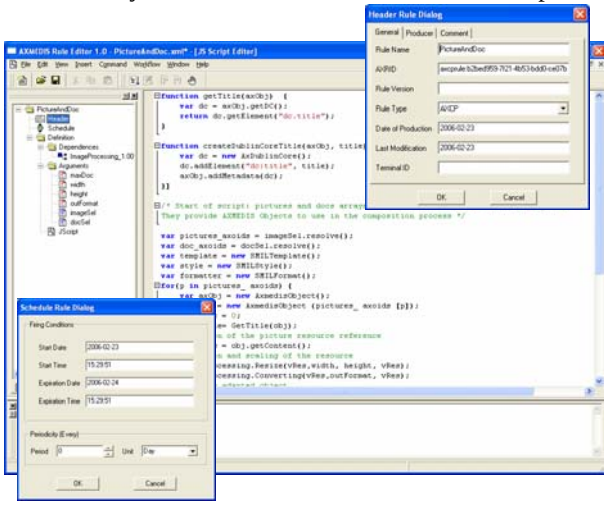

Fig. 4 – AXCP Rule Editor

5.2 AXCP GRID Node

The AXCP GRID Node is an Executor of Rules running on each GRID Node. It receives the Rule code to be executed from the Scheduler and provides to it the notifications, exceptions and errors messages at runtime. It is a computational unit in the distributed environment where rules are executed. It hosts the Spider Monkey Javascript engine for JavaScript code execution and the GRID interface for the network communication. The architecture of the AXCP GRID Node is mainly constituted by the following main components:

- **Rule Executor Manager** to control the *Script Executor*;
- **Rule Executor** to execute rules that include an instance of Spider Monkey Javascript Engine (called JS Engine).

The **Rule Executor Manager** is the interface between the Scheduler and the Rule Executor. The GRID Peer Interface allows the communication in the distribute environment and provides the support for:

1. receiving commands, messages, requests and files from the Scheduler;
2. sending messages, notifications, exceptions and files to the Scheduler;
3. being discovered by the Scheduler during the discovering phase.

The Rule Executor Manager analyses the Node to recover the information to build its profile that sends it to the Scheduler, when it is discovered. It stays in an idle status until the Scheduler engages it asking to put in execution a rule. In this case, the rule is received as an XML file, and then it is fired by the *Launcher*. The Rule Executor Manager provides to the *Rule Executor* the communication support. In this way, errors, status and the end of execution are notified to the Scheduler. Finally, it provides remote control commands to kill, pause, resume, of rules and to get the status of the executor.

The **Rule Executor** has been designed to host the JS Engine and run a Rule also in the *debugging* and *check* modalities. The former modality allows debugging and monitoring the execution of a Rule by means of traps in the code corresponding to breakpoints (interrupting the execution), watching variables and managing the stack of function calls. The latter modality allows verifying the syntax of the rule before to run definitively in the AXCP Rule Engine, and estimating by partial execution some parameters related to the complexity of the rule such as: requested time computation, amount of disk space, data transferring time, etc.
This last modality is obtained by executing the script without the real execution of the procedures that are the real cost of processing, with the aim of extracting the main parameters that are going to influence the execution time of the algorithm depending on the context (Executor) in which it is going to the executed.

The Spider Monkey JavaScript engine's built-in objects were extended by wrapping AXMEDIS data types (AXMEDIS JS Classes), functions (JS Functions) and AXCP plug-in tools deriving from the AXMEDIS Framework data types. They provide direct program services, or they can serve as interfaces to program's services. For example, an AXMEDIS JS Class that provides direct service might be one that handles all of an application's network access, or might serve as an intermediary broker of database services.

6. AXCP Rules and Tools

An AXCP is comprised of three main sections (see Figs. 5). The *Header* contains general metadata such as: rule name, AXRID (Rule ID), rule version, rule type, software name, version of software, date of production, time of production, author, affiliation, URL, comment, last modification date and time. *Schedule* section contains temporal constraints describing the rule status ("*active*" or "*inactive*") and conditions for firing it such as: start date, start time, periodicity (monthly, daily, weekly, etc.), expiration date and expiration time. Such information is used by the AXCP Rule Scheduler in the GRID environment for planning the activity and associating active rules with available computational resources. *Definition* is the section that contains the rule signature in terms of list of arguments (parameters and selections), list of dependences (required AXMEDIS plug-ins) and the script code to run. Dependences define constrains on AXMEDIS plug-ins that has to be installed in a GRID node.
A XML formalization can be obtained from Fig. 5. An AXCP rule can be used for different purpose in the Content Processing Area.

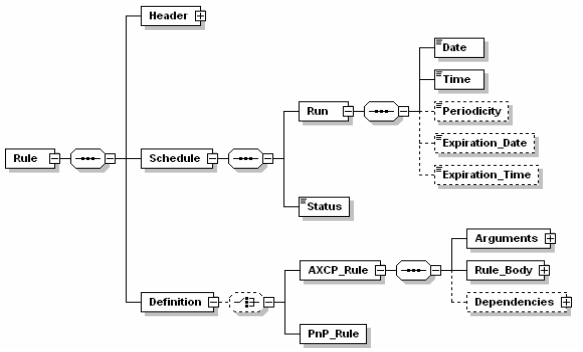

Fig. 5 – XML schema of the AXCP Rule

For instance, in an AXMEDIS Content Factory three main different rules categories are used:

- ***Publication Rule*:** executed to automating the process of publishing AXMEDIS objects from the database to the distribution channel or P2P network;
- ***Downloading Rule*:** executed to download a Selection of objects from the P2P network and

move them into the temporary database for further inspection and Load according to a Loading Rule.

- ***Loading Rule*:** executed to load into the AXMEDIS Database with a selection of objects retrieved from the P2P network;

Formally, the signature of the AXCP Rule is:

$$R = f(S_1,S_2,..,S_n,P_1,\ldots,P_m)$$

Where:

- S_i is a sequence of queries, to be sent to the AXMEDIS Database to retrieve digital object (content) IDs or a set of object IDs to AXMEDIS objects or a mix of them;
- P_i is a parameter (basic type as integer, common string, XML string, Boolean, etc.);
- f is the identifier of rule (e.g., the ID of rule);
- R is the consumptive result of the rule application.

Other results are the resulting objects processed and/or produced during the execution of the Rule and they can be directly posted into the file system or database.

7. Conclusions

Content enrichment is one of the most expensive activities related to the integration and augmentation of information of content elements. This activity is typically performed for cultural and educational content and recently also for edutainment and infotainment. AXMEDIS Framework aims at providing technical solutions and tools for reducing costs by automating content production, protection and distribution, thus supporting activities of content enrichment. In this paper, the aims and the capabilities of the AXMEDIS Content Processing, AXCP, architecture have been described. This AXCP is a flexible and scalable core subsystem of the AXMEDIS Framework and architecture. Such subsystem is involved in the automatic content production, protection, formatting, metadata adaptation, etc. The adopted solution is based on GRID Computing and on a script language for coding Rules to be executed. The AXCP Rule Language extends the Javascript by adding a descriptor for those rules and a large set of object types modelling AXMEDIS data types for content processing: digital resources, metadata, descriptors, ID in UUID, licenses in MPEG-21 REL, protection information in MPEG-21 IPMP, XML, SMIL, HTML, WSDL, etc. In addition, rule capabilities for the rule description, GRID Node capabilities, and plug-in component descriptors have been defined in order to enable the dynamic allocation and verification of AXCP Rules on GRID Nodes. The model defined for describing the components allows extending the capabilities of the AXCP Language and the automatic installation of new components via the AXCP Scheduler if needed. The full documentation can be recovered on AXMEDIS portal. AXMEDIS is an open platform in the sense that the tools, the specifications and the formats are accessible and royalty free. The access to the source code of all the tools can be easily obtained joining the AXMEDIS community.

8. Acknowledgments

The authors would like to thanks to all AXMEDIS project partners including the Expert User Group and all affiliated members, for their contributions, funding and collaborations. A specific acknowledgment to European Commission IST FP6 DG INFSO for partial funding of AXMEDIS project. A warm thanks to all AXMEDIS people that have worked in the project and sorry if they have not been involved in the paper and have not been mentioned.

9. References

[1] D. W. Erwin and D. F. Snelling. UNICORE: A GRID computing environment. Lecture Notes in Computer Science, 2150, 2001.

[2] Foster. The anatomy of the GRID: Enabling scalable virtual organizations. Lecture Notes in Computer Science, n.2150, 2001.

[3] WWW.AXMEDIS.ORG

[4] www.variazioniproject.org

[5] MPEG Group, "Introducing MPEG-21 DID", www.chiariglione.org/mpeg/technologies/mp21-did/

[6] MPEG Group, "Introducing MPEG-21 IPMP Components", www.chiariglione.org/mpeg/technologies/mp21-ipmp/

[7] P. Bellini and P. Nesi, "An Architecture of Automating Production of Cross Media Content for Multi-channel Distribution", in Proc. AXMEDIS 2005, Florence, Italy, 30 Nov. - 2 Dec., IEEE Press, pp 123-133

[8] P. Nesi, Rogai, D., Vallotti, A., "A Protection Processor for MPEG-21 Players", Proc. of the IEEE International Conference on Multimedia and Expo, Toronto, Canada, 9-12 July 2006.

[9] P. Bellini, Nesi, P.; Rogai, D.; Vallotti, A., "AXMEDIS tool core for MPEG-21 authoring/playing", Proc. of the first International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, IEEE Computer Soc. Press, Nov. 2005, Florence, Italy, AXMEDIS 2005.

[10] D. De Roure, N. R. Jennings, N. Shadbolt, The Semantic Grid: Past, Present and Future, Proceedings of the IEEE, Vol.93, N.3, March 2005.