Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE4.9.1.2
The Usability Issues for the AXMEDIS Production Tools, First update of D4.9.1

Version: 1.2
Date: 11/9/2006
Responsible: Elke-Maria Melchior, ACIT (revised and accepted by coordinator)

Project Number: IST-2-511299
Project Title: AXMEDIS
Deliverable Type: PU
Visible to User Groups: Yes
Visible to Affiliated: Yes
Visible to Public: Yes

Deliverable Number: DE4.9.1.2
Contractual Date of Delivery: M24 (end of August 2006)
Actual Date of Delivery:
Work-Package contributing to the Deliverable: WP4, subWP4.9
Task contributing to the Deliverable: T4.9.1, T4.9.2
Nature of the Deliverable: report
Author(s): ACIT, input from DSI and user partners

Keyword List: Usability, Quality, Methods, Measures, Content Production, 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 Licence.

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 Contents

1 Executive Summary and Report Scope

AXMEDIS is researching and producing innovative tools for content production which will be integrated in an open P2P tool (AXEPTool) at which any CMS could be joined for content production and distribution in the B2B environment of AXMEDIS. The added value of the AXMEDIS tools will be reduced costs and allowing production and formatting on demand. A number of sustainable demonstrators and take up actions will demonstrate a content production process respecting the business of SMEs, permitting access to relevant content and producing content at reasonable cost.

The objective of this work package is to assure that the AXMEDIS project develops tools and applications which in the end correspond to the needs of prospective users. Competence is developed in the consortium early and proactively so that user issues can be integrated into the development from the start rather than carried out as a separate activity.

The user aspects in AXMEDIS will cover the entire innovation cycle- not just RTD, and the activities carried out will be specific and appropriate to the different phases of this lifecycle. The development process is user-centered and iterative with participation of users in all phases. Measures will show the success in terms of user quality and acceptance.

The principle pursued is to involve users early, to analyze their needs and requirements (using as input the result from WP2 Continuous Requirements Analysis, and feeding back results from user needs analysis), then to embed usability testing into all development activities to provide early feedback in such as way that intermediate design and development results can be improved. In order to reach the implementation and trial phase early and effectively, users – especially professional users – will be prepared by workshop activities and training where they will be introduced to new technology and the possible impact and planning for their processes and existing systems, as well as to the procedure of and to the selected methods for user validation.

Training must show that users enter into trials well prepared. The training activities will address a large audience, including potential future participants of take-up actions and related projects. It complements the technical system development and the dissemination activities.

Chapter 2 illustrates the state of the art of user validation, the technical implementation of a user-centered development process. The purpose of user validation is to assure that the result of the project – the AXMEDIS framework and tools – is in agreement with the needs and requirements of customers and users, and is accepted by them in the end. User validation, including such topics as analysis of user needs, contextual inquiry and design, usability inspection, usability testing, or user preferences and satisfaction measurement, is a mature approach now, based on scientific knowledge, and proven and tested methods. There are a number of sources of information, including textbooks and practical guidance, which help to introduce the approach into development teams and projects.

We refer to the introductory information, handbook, and extensive information on methods used in user validation which is available on www.vnet5.org. The site introduces to the procedures and methods which we consider sound and proven, and which cover all aspects of user validation – although this is by no means the only view of best practice in the field, and others offer similar information. This is the main source for further information to most of the methods which we consider valid and recommend. Brief descriptions are given for the practitioner who has to plan user validation in a development project, and links to further information.

A question that may be raised is whether the same methods which were developed in the context of office software, transaction processing, communication devices or navigation tools are applicable to products which focus on the production of cross media content for multi-channel distribution, and which are highly innovative. We argue that this is a domain with some additional challenges, which are met by carefully adapting the methods and approach used, but it largely presents the same problems for user validation as any other software development project.

The innovative challenge will be met by using care in the approach, and awareness of the fact that comparison with existing integrated applications and the use of previous experience is not possible. Specific challenges for user tests in the AXMEDIS project are the following: AXMEDIS integrates different challenging tools. The technical risk of this approach may be reduced by investigating and understanding the preferences of users for new functionality and the value of new functionality to users. Some expectations for workflow and process efficiency can only be realized after relatively large scale implementations of demonstrators have been in operation for a while.

Chapter 3 introduces to user-centered development and user validation to familiarize the project partners with the terminology. The specific challenges for the AXMEDIS project are identified in this chapter:

- Efficiency of workflow and process
- Cost/benefit and added value
- Usability of a scripting language for content processing
- Usefulness of metadata for information retrieval.

Chapter 4 describes the approach to user-centred development and user validation in the AXMEDIS project in detail.

In chapter 5 relevant evaluation methods for AXMEDIS are described. Traditional usability evaluation methods focus on the evaluation of the quality of use of user interfaces. New methods may have to be developed for the evaluation of the specific challenges listed above.

2 Introduction

AXMEDIS is researching and producing innovative tools for content production which will be integrated in an open P2P tool (AXEPTool) at which any CMS could be joined for content production and distribution in the B2B environment of AXMEDIS. The added value of the AXMEDIS tools will be reduced costs and allowing production and formatting on demand. A number of sustainable demonstrators and take up actions will demonstrate a content production process respecting the business of SMEs, permitting access to relevant content and producing content at reasonable cost.

The objective of this work package is to assure that the AXMEDIS project develops tools and applications which in the end correspond to the needs of prospective users. Competence is developed in the consortium early and proactively so that user issues can be integrated into the development from the start rather than carried out as a separate activity.

The user aspects in AXMEDIS will cover the entire innovation cycle- not just RTD, and the activities carried out will be specific and appropriate to the different phases of this lifecycle. The development process is user-centered and iterative with participation of users in all phases. Measures will show the success in terms of user quality and acceptance.

The principle pursued is to involve users early, to analyze their needs and requirements (using as input the result from WP2 Continuous Requirements Analysis, and feeding back results from user needs analysis), then to embed usability testing into all development activities to provide early feedback in such as way that intermediate design and development results can be improved. In order to reach the implementation and trial phase early and effectively, users – especially professional users – will be prepared by workshop activities and training where they will be introduced to new technology and the possible impact and planning for their processes and existing systems, as well as to the procedure of and to the selected methods for user validation.

Training must show that users enter into trials well prepared. The training activities will address a large audience, including potential future participants of take-up actions and related projects. It complements the technical system development and the dissemination activities.

<u>State of the art</u>

User validation is the technical implementation of a user-centered development process. The purpose of user validation is to assure that the result of the project – the AXMEDIS framework and tools – is in agreement with the needs and requirements of customers and users, and is accepted by them in the end. User validation, including such topics as analysis of user needs, contextual inquiry and design, usability inspection, usability testing, or user preferences and satisfaction measurement, is a mature approach now, based on scientific knowledge, and proven and tested methods. There are a number of sources of information, including textbooks and practical guidance, which help to introduce the approach into development teams and projects.

We refer to the introductory information, handbook, and extensive information on methods used in user validation which is available on www.vnet5.org. The site introduces to the procedures and methods which we consider sound and proven, and which cover all aspects of user validation – although this is by no means the only view of best practice in the field, and others offer similar information. This is the main source for further information to most of the methods which we consider valid and recommend. Brief descriptions are given for the practitioner who has to plan user validation in a development project, and links to further information.

A question that may be raised is whether the same methods which were developed in the context of office software, transaction processing, communication devices or navigation tools are applicable to products which focus on the production of cross media content for multi-channel distribution, and which are highly innovative. We argue that this is a domain with some additional challenges, which are met by carefully adapting the methods and approach used, but it largely presents the same problems for user validation as any other software development project.

The innovative challenge will be met by using care in the approach, and awareness of the fact that comparison with existing integrated applications and the use of previous experience is not possible. Specific challenges for user tests in the AXMEDIS project are the following: AXMEDIS integrates different challenging tools. The technical risk of this approach may be reduced by investigating and understanding the preferences of users for new functionality and the value of new functionality to users. Some expectations for workflow and process efficiency can only be realized after relatively large scale implementations of demonstrators have been in operation for a while.

Chapter 3 introduces to user-centered development and user validation to familiarize the project partners with the terminology. The specific challenges for the AXMEDIS project are identified in this chapter:

- Efficiency of workflow and process
- Cost/benefit and added value
- Usability of a scripting language for content processing
- Usefulness of metadata for information retrieval.

Chapter 4 describes the approach to user-centred development and user validation in the AXMEDIS project in detail.

In chapter 5 relevant evaluation methods for AXMEDIS are described. Traditional usability evaluation methods focus on the evaluation of the quality of use of user interfaces. New methods may have to be developed for the evaluation of the specific challenges listed above.

The results of usability evaluation will be input to the assessment and evaluation data in WP 10.

3 Purpose of user-centered development

The purpose of user validation in RTD projects is to assure that the result of the project - here the AXMEDIS framework and integrated tools - is in agreement with the needs and requirements of users and customers, and is accepted by these in the end. User-centered development, including activities such as analysis of user needs, usability testing and user satisfaction measurement, is a mature field today, based on scientific knowledge, proven and tested methods. A number of sources of information, including text books and practical guides help to introduce the approach into development teams and projects. We refer to VNET5 (www.vnet5.org) as the main source for further information to most of the methods which we consider valid and recommend. VNET5 contains introductory information, a handbook, and extensive information on the selection of appropriate methods for user validation. The site introduces to the procedures and methods which we consider sound and proven, and which cover all aspects of user validation. Although, this is not the only view of best practice in the field. Other sources offer similar information (Dumas & Redish 1993, Hix & Hartson 1993, Isensee & Rudd 1996, Mayhew 1999, Nielsen 1993 and 1994, Preece 1994, Vredenburg et al 2001).

The tools which are developed in AXMEDIS include functionality for the production of cross media content and for multi channel distribution. The innovative aspects are that the AXMEDIS tools will be integrated in an open P2P tool (AXEPTool) at which any CMS could be joined for content production and distribution in the B2B environment of AXMEDIS. The added value of the AXMEDIS tools will be reduced costs, increased efficiency of the production process, and allowing production and formatting on demand. The AXMEDIS tools will carry out some of the work procedures automatically and these innovative functions will only be partly visible to users.

A question which has been raised frequently in connection with other innovative and advanced applications is whether the same validation methods which were developed in the context of predominantly procedural applications such as office software, transaction processing, or devices for communication and navigation are applicable to innovative products with new functionality.

We argue that these are separate issues, each to be dealt with an appropriate approach and valid and applicable methods:

- A large part of the AXMEDIS framework and tools present the same problems for user validation as any other software development project, and suggest that proven and well known methods for user tests and validation are used. The main objective is to assure that the tools correspond to the needs and requirements of users (eg. friendly and robust user interface, easy to use).
- Innovative challenges are the evaluation of workflow and process efficiency, the analysis of cost/benefit and added value, usability of a scripting language for content processing and the evaluation of the usefulness of metadata for information retrieval.

A question which will be asked is whether the AXMEDIS framework with integrated tools improves significantly user performance and the subjective user experience. The total system performance, including the effects of user behaviour and user experience, and the resulting business benefits is assessed by looking at global cost and performance parameters and the value of innovative and automated features, and does not specifically evaluate individual aspects of the underlying innovative technology.

3.1 Continuous collaboration with users and customers to assure acceptance

User-centered development, that is the continuous collaboration with users in the analysis and evaluation of the technical concepts and results during the entire development process, has produced consistently positive results in the development of successful new products and services. The main aim of user centered development is to avoid the disappointment of development projects which do not meet their great expectations. It attempts to achieve this in two ways:

- By ensuring that the solution which is developed is free from obvious defects and is as close as possible to user needs,
- By providing sound and reliable information about the value and applicability of technology, also in comparison to competing solutions (benchmarking).

Independent of the type of product, service, or industry considered, there is only one reliable approach to assure that at the end of the development process the result is accepted by users: To involve users from the start of the development process in an effective manner. Effective means that valid - correct and relevant - information is collected from prospective users and other stakeholders, and that it is used to improve the solution under development.

It is important to distinguish between the different types of stakeholders:

Users are the individuals who in the course of their work interact directly with the product or service which is developed. The acceptance criterion of users is that they are able to carry out the intended work tasks efficiently and successfully, and without undue problems or stress, and in addition that their subjective assessment of the innovation is positive. The positive subjective user experience becomes increasingly important in situations where the user is free to make use of the innovative features, and integrate them into his normal working procedures, or to disregard and not use them. Professional users will use the AXMEDIS framework and tools to produce content for consumer end-users. There will be also be ‘prosumers’, users who create and consume content.

Customers are organizations (or individuals) which decide about the investment (eg into the AXMEDIS framework and tools), and which must be persuaded of the value of a new technology. The acceptance criterion of customers is the total cost/benefit advantage obtained by the introduction of innovative technology. This also includes non-monetary factors such as the consequences for human resource management and the positive acceptance by the personnel affected.

Other stakeholders are individuals or organizations who are neither direct users nor owners or customers of the product or service under development, but may have an interest in the development results. In AXMEDIS these other stakeholders include artists, authors, content owners, production and distribution chain managers, production designers, production and distribution chain technicians, marketing professionals, legal experts, distribution partners, researchers, standards bodies, collecting societies, industry associations, sponsors, etc.

Many traditional industries - such as the food or automotive industries - have developed effective and reliable processes to assure that the results of development activities correspond to customer needs. In these industries thorough and disastrous product failures are quite rare today. In established industries the needs of customers, and the criteria according to which they make their choices, are fairly well known, and testing of product quality is securely integrated into the development process. Even the training and education of the developers includes this element. The situation is different in the electronics and software industries where long standing experience does not yet exist, because there is an abundance of new ideas and products. As a consequence a critical reflection of the right approach for user and customer involvement is needed at the initiation of each new project - which we do here for the AXMEDIS project.

3.2 The specific demands and challenges in the AXMEDIS project

A frequent argument is that innovative technology can not be guided by user- and market-analysis at all, because the prospective users will not be able to answer questions relating to the innovative product before full scale prototypes and the required infrastructure are available and can be demonstrated in full. This is an argument worth taking into account, but the response should be to adapt the approach to this special challenge, and not to rely on the anticipation of user needs by experts entirely. Experience has shown that the likelihood of error of technology focused expert predictions is quite large, and hopes for grand successes of new products and large markets are often not fulfilled. We regard it as advisable and necessary to apply user centered design principles to innovative development projects.

AXMEDIS is an application oriented RTD project where some of the development activities will be less amenable to precise evaluation than others, but the need to assess the value of the prototype applications and demonstrators with users is still present. The methods used will have to take these specific needs into account.

The specific challenges for user tests in the AXMEDIS project (derived from DoW and D2.1.1) are

- Efficiency of workflow and process
- Cost/benefit and added value
- Usability of a scripting language for content processing
- Usefulness of metadata for information retrieval

Traditional usability evaluation methods focus on the evaluation of the quality of use of user interfaces. New methods may have to be developed for the evaluation of the specific challenges listed above.

3.2.1 Efficiency of the workflow and process

The expectations for increased efficiency of the workflow and process can only be realized after relatively large scale implementations have been in operation for a while and users have invested some effort to learn how to use the AXMEDIS tools. The cross-media production and distribution workflow is likely to be a determining factor for user acceptance. This includes the specific issue of Digital Rights Management (DRM), the flexibility of DRM, increased safety and reliability of protection models.

3.2.2 Cost/benefit and added value

The AXMEDIS tools and the approach to integrate the tools into the AXMEDIS framework are new, and a considerable amount of technical risk exists. The risk may be reduced by understanding which functionality is of value to users, which functions do users prefer to other functions.

3.2.3 Usability of a scripting language for content processing

AXMEDIS is developing a language for scripting rules for processing content. The rationale behind scripts is

- to provide automation of complex and at times repetitive tasks in the content production process
- to adapt the AXMEDIS system to user needs.

What is performed in code has to be similar to what is performed on the authoring side on the GUI of the authoring tools. The question is which is the best way of programming those rules?

3.2.4 Usefulness of metadata for information retrieval

Efficient communication of metadata is essential for different groups of users using different types of content (text, images, audio, video, etc.), different types of AXMEDIS objects for different purposes. The opportunity of metadata is high quality (re-)usable content, selectable on the basis of metadata (features, context of use, technical aspects). For composition and formatting metadata play a crucial role. They can help to produce more usable and more accessible content at all levels from editing to fruition. For the selection of AXMEDIS objects users will exploit available metadata to the maximum. It is evident that dealing with both B2B and B2C environment the metadata set has to be extremely variegated and structured. This is particularly true when taking into account that in the B2B segment we may have to face publishers acquiring content for editing or aggregating, distributors acquiring content for rental, other professional actors acquiring content for different purposes like education, training, advertisement, company image communication. This requires that the object metadata set combines the simplicity of Dublin Core (small number of fields) for an easy search with the complexity of LOM and other metadata sets to ensure proper professional usage at all levels from editing to fruition.

3.3 Approach and methods

The approach defines the goals and conceptual background for user validation, while the methods applied to implement the approach are selected from a repertoire to fit the specific parameters of the AXMEDIS project. The general approach for the project is described here. The selection of specific methods will be made when all relevant conditions are known which have to be taken into account, such as the precise question addressed in analysis and testing (which depends on design options), the availability of subjects, and other factors.

Here the main phases of user validation are defined, and a portfolio of the applicable methods is listed which are considered for the project. The user validation plan for the AXMEDIS project is developed on the basis of the format developed in VNET5 and will be updated as needed and as the project progresses. A template for reporting of results is also proposed.

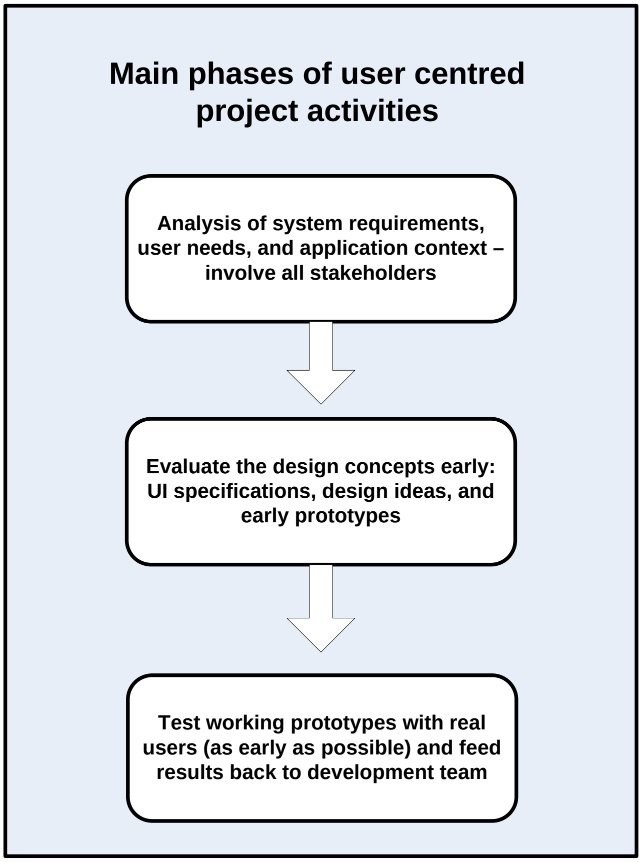
One of the objectives of this user validation approach is to use common methods, and to look for complementary results. Some of the experience may be applicable to other projects (eg take-up actions).

The detailed planning of methods to use, and the commitment to carry out a defined program of user testing and validation is set out in the attached user validation plan. Later we may decide to split this plan into one plan each for each demonstrator and take-up action. The plan is defined in detail up to month 18. The process and methods to follow from month 19 until the end of year 4 will be elaborated by that time, and documented as an extension to this user validation plan. The ability to respond to findings, and the principle of iterative testing and redesign cycles requires that sufficient flexibility is maintained in the process up to the end of the project.

3.4 The main phases of user centered development

There are three main phases of user centered design, which correspond to project phases. In all phases the objective of user centered design is to generate information by user analysis which guides the design and development activities.

It is essential that the results of user validation are communicated towards the individuals and groups who are able to use and implement them to improve design quality. (Design refers to the entire software and other relevant features which determine the user experience when interacting with the applications developed, i.e. coverage of functions, graphical and navigation design, and also quality factors such as performance, robustness, learning effort, etc.)


Main phases of user centred project activities
Analysis of system requirements, user needs, and application context – involve all stakeholders
Evaluate the design concepts early: UI specifications, design ideas, and early prototypes
Test working prototypes with real users (as early as possible) and feed results back to development team

4 Approach to user validation in AXMEDIS

User validation (including user needs analysis, usability testing, user satisfaction measurement and other methods) is a mature and well documented discipline. For specific domains and development groups the approach and methods to be used are quite well established, and need not be introduced in detail. The issues specific to the AXMEDIS project will be considered and appropriate approaches proposed.

4.1 Activities in subsequent phases of the project

The common approach to user validation as described at the VNET5 web site (www.vnet5.org) is representative for the organization of user centered activities in development projects and has been adopted widely. A key property is that the actual process of user validation is tailored for each project in response to the specific goals and parameters of the project.

The following table describes the user validation activities for the AXMEDIS project. User validation is a process running in parallel to other project activities and closely interlinked with the development of the AXMEDIS framework and tools, with the development of demonstrators and, at a later stage, with the take-up actions. The timescales and activities correspond to the current state of planning, but will be updated regularly in response to project progress.

Activity	**Comment**	**Corresponding AXMEDIS activities / results**	**Corresponding AXMEDIS timescales**
Develop a vision for the application and obtain the support of all stakeholders which will be involved in the project. Agree on the strategic goals of the project.	This is the condition for starting user validation, and the first milestone	Project proposal, DoW	
Collect user needs and requirements, elaborate scenarios of use and use cases, define test cases.	Experienced users and user representatives should be closely involved in this activity, and the results should be critically reviewed (e.g. in user group meetings and workshops)	DE2.1.1a/b User requirements and use cases DE2.2.1 Test cases and content description DE2.3 User group meetings	M1-8
Analyze the context of use of the AXMEDIS framework and tools	Physical, technical, legal, safety, privacy and other requirements and constraints of users are identified.	Platform requirements and constraints, migration and integration aspects (DE2.1.1a)	M1-8 Repeated analysis may be indicated at the time of demonstrator development and at the start of take-up actions.
Define non-functional requirements (quantitative, where possible), taking the expected competitive situation into account. Select applicable measures and benchmarks if possible.		Some non-functional requirements are included in DE2.1.1a.	M1-8
Adopt or develop a style guide.	Minimum requirements are defined in standards for software interface and interaction (e.g. ISO 9241). Platform vendors have developed usability principles, interface and interaction guidelines, most of them are available on the web. Company internal style guides from user partners may be applicable.	See chapters 5.2.1 and 5.2.2 DE3.1.1 Guidelines and specification of research	A decision on the adoption of a style guide has to be made when user interface development activities start.

Activity	**Comment**	**Corresponding AXMEDIS activities / results**	**Corresponding AXMEDIS timescales**
Perform regular design reviews, inspect and test technical components and first prototypes during the entire development phase.	Initially tests are carried out by the development team or by user interface experts. Only technically sound results should be presented to decision makers and to users - usually this is the case later during the development cycle.	Mock-ups presented to user group and to external experts	M12-18 Usability inspection by users and experts during AXMEDIS project meetings and during the 1st AXMEDIS conference in December 2005
Foresee frequent test & evaluate & redesign cycles. Perform usability inspections of components and first prototypes. User tests are carried out as soon as mature components are available.	The re-design of components as a result of tests which reveal shortcomings must be anticipated.	AXMEDIS Tools (Editors and Viewers, Query Support, Workflow Manager, and other components with user interface developed in WP4).	M12-M18 Usability workshop 1st AXMEDIS conference in December 2005
Inspection, testing, and review of integrated prototype. User tests are carried out as soon as mature prototypes are available.	Critically dependent on the availability of a working prototype.	When the first stable integrated prototype of the AXMEDIS framework is assembled, a user testing workshop will be held where the prototype is inspected and where the user testing methods which will be used subsequently are practiced.	Usability inspection of AXMEDIS Tools by users and experts during AXMEDIS project meetings in March and July 2006. M18-24: Continuous usability inspection and informal user testing of AXMEDIS Tools (chapter 6 contains the results of this activity)
Update of the requirements with the results from inspection of prototypes with application experts and first tests with selected users.	Partly an informal process. To be documented when important modifications are made.		Continuously until end of year 4. M18 DE2.1.1.2.1.User requirements, first update DE2.1.1.2.2 Use cases and scenarios, first update DE2.2.1.2 Test cases and content description, first update

Activity	Comment	Corresponding AXMEDIS activities / results	Corresponding AXMEDIS timescales
Inspection, testing, and review of demonstrators			M18-24 Usability inspection by users and experts during AXMEDIS project meetings in March and July 2006.
User tests of the AXMEDIS demonstrators in realistic application environments. Assessment of workflow and process efficiency and user performance as well as subjective user experience and preferences.	Involve user representatives and real users as soon as stable demonstrators are available. Avoid the introduction of a demonstrator to users as long as uncertainties about the technical maturity of the implementation still exist.		M24-M36
Tests of the AXMEDIS framework and tools as part of Take-Up Actions.	Involve user representatives and real users as soon as stable demonstrators are available. Avoid the introduction of a demonstrator to users as long as uncertainties about the technical maturity of the implementation still exist.		Year 4 To be specified in detail at the end of year 3.
Assessment of the added value and business benefit of specific features of the AXMEDIS framework and tools, of workflow and process efficiency and user performance as well as subjective user experience.	Large-scale trials involving real users should only be initiated when a stable system allowing meaningful productive use is available.		Year 4 Internal trial M37-40 External trial M41-48 To be specified in detail at the end of year 3.

Milestone results of the user validation activities

The sequence of activities does not have to be strictly sequential. Four critical and essential milestones should be reached, where the feasibility of the user validation plan should be reviewed.

	Achievement	
1	Common vision of user needs is agreed between development teams, and customer and user representatives	Achieved M6
2	A tested and working initial AXMEDIS framework implementation is completed which has passed expert reviews and informal usability tests.	Expected M18 Achieved M24
3	User tests have indicated workflow and process efficiency, and demonstrated user acceptance and the feasibility of the AXMEDIS framework. They may have indicated further needs for correction of weaknesses, and the potential of the application for introduction as a product.	Expected M36
4	User tests have demonstrated workflow and process efficiency, added value, and user acceptance of the AXMEDIS framework. They may have indicated further needs for correction of weaknesses, and the potential of the application for introduction as a product (incl. Take-up actions).	End of project: M48

Other results, for example requirements documents, are not milestones, because the iterative mode of system development does not define a single point in the project plan where the requirements specifications are fixed. Requirements documents are updated when severe new demands have been identified and need to be communicated.

As a general rule, the development team should test and refine the AXMEDIS software - possibly with the help of experts - until it is satisfied that no further defects exist in the software, and that the quality is likely to meet the expectations of users. Only at that moment should real users be involved. Presenting users with faulty software wastes their time and motivation, and is unlikely to deliver much information beyond the fact that users are not satisfied.

The specific need of assessing user response to process-oriented functionality (workflow) requires that appropriate methods will be developed. These should be available during the field tests in the fourth year of the project.

The elicitation of user requirements and the design reviews and user tests during development are under the responsibility of the development teams (WP4 and WP5), and must be seen as part of the development activity. The role of T4.9 is to set up a user validation plan, to recommend appropriate methods, to inspect initial prototype components and to support the development team and the user partners in the execution of the user validation plan.

The final assessment of the AXMEDIS framework and tools and the data analysis is coordinated by T4.9.

4.2 Evaluation of the usability of a scripting language

AXMEDIS is developing a language for scripting rules for processing content. What is performed in code has to be similar to what is performed on the authoring side on the GUI of the authoring tools. The question is which is the best way of programming those rules?

The rationale behind scripts is

- to automate complex and repetitive tasks in the content production process
- to adapt the AXMEDIS system to user needs

A script is a list of commands which are executed automatically as if they had been entered one by one by the user. Scripts may also include sequencing instructions such as loops, conditionals, calls to predefined functions and other scripts.

By giving users the possibility of automating and sequencing any series of actions, scripts allow for interaction with the AXMEDIS system at a higher level of abstraction. A complex set of actions can be made an atomic object, which can then be played back at will in different contexts. By writing the appropriate script, a user can for example format all documents that belong to a specific set to be delivered to end-users with a single operation.

A scripting system represents a kind of high-level macro facility, a useful improvement already provided by many applications (office systems, excel, GIS, hypertext).

The objectives (and constraints) should be to bring the benefits of additional programming language features to the AXEMDIS framework and tools which per se is not recognised as a programming language by the users:

- to maintain compatibility of the language for scripting rules with the new AXMEDIS tools and with the language of tools users have been using so far, and
- to maintain the usability advantage. The commercial success of the scripts will be largely due to the fact that users find them more usable than programming languages for programming like tasks.

The analytic approach to the design of a language for scripting rules suggested here is “emphasize the cognitive requirements of end users as a primary design criterion”. This approach is based on recent developments in the study of programming usability which is applicable to the design of languages for scripting rules.

Users use computers to get their work done. Often they are not interested in programming per se. The work of AXMEDIS users, eg content processing, consists of repetitive and complex tasks. Usability is a fundamental concern in AXMEDIS. Therefore a mechanism, the ability to define re-usable abstractions, is needed. Principles from HCI research can be applied to the design of a language for scripting rules. However, it is hard to find practical useful guidance for the design of a language for scripting rules among the research results related to the psychology of programming and from empirical studies of programmers (Brown & Gould, 1987, Corritore & Wiedenbeck, 2000; Dumas & Parsons, 1995; Green, Petre & Bellamy, 1991; Hoadley, Linn, Mann & Clancy, 1996). There is no empirical evidence that usefully informs design decisions for a language for scripting rules for content production and processing. A structured approach to considering human issues in programming which has been used successfully in other recent programming language design projects is the Cognitive Dimensions approach which will be described in chapter 5.2.3.

4.3 Evaluation of visionary research aimed at breakthroughs as opposed to engineering solutions to defined problems

It has frequently been stated by leading innovators that there is no need to evaluate highly innovative research results, because demand for them is so obvious and their value for applications is so gigantic, that the only hurdle which has to be taken for success is the ability to deliver the technology. In the face of many instances where these expectations were never fulfilled, this argument should be regarded with suspicion. It is however a fact that it is difficult to foresee the applications for radically new technology, and it does in fact not make sense to insist too early to use innovations for particular applications before the technology is fully developed.

The AXMEDIS framework and tools are developed in a goal directed process, starting from the analysis and specification of requirements, context of use, guidelines and system specification, and added value and cost/benefit targets have been stated in quantitative terms:

- Reduction of the costs of cross media production in the order of 30 % in the production of "automatic content" and for on-demand production, and distribution in the order of 15 %.
- Increased accessibility to content for final consumers (implementation of new services; number of distribution channels and access devices)
- Increased visibility/accessibility of content with P2P tools of AXMEDIS for B2B content sharing
- Increased safety and reliability with protection models
- New business opportunities and higher involvement of SMEs.

4.4 Analysis of user needs and user requirements

The requirements and needs of users have been analyzed by detailed studies of the application context, based on the involvement of expert users from the AXMEDIS user partners. Needs of users have been elicited in work group meetings and with questionnaires. Further requirements were elicited from expert users in user group meeting.

As part of a systematic requirements analysis in traditional terms the information needs of users have been investigated. Part of this analysis was directed towards the understanding of the role of the AXMEDIS framework and tools in this process, and the expected specific benefit to be derived from the use of AXMEDIS tools.

Some quality criteria of users have been defined at this stage. These are also relevant from the customers' point of view:

- Efficiency of the workflow and process
- Best coverage of functions (adapted to tasks)
- User friendly GUIs
- Ease of use
- Robustness
- Accessibility
- Visibility

More quality criteria may be relevant. These will be considered in relationship to technical quality criteria for the AXMEDIS framework, and in connection with the assessment of the quality of media content from the user perspective. It may be the case that the quality of some algorithms included in the AXMEDIS framework are neither visible to users, nor of direct concern for them: It may just be the total quality which is assessed by the user.

The approach proposed is to identify the functionality which depends on the new and automatic algorithms and to investigate the value of this functionality according to user assessments elicited by ratings (see discussion of "tradeoffs" below).

4.5 Assessing components (AXMEDIS tools) and the value of the combined solution (AXMEDIS framework)

The AXMEDIS framework is presented to the user as a package where the quality and benefit of individual tools is not directly visible, and cannot be assessed separately by the user. The underlying functionality and the user interface of the AXMEDIS framework, for example, are not visible as separate tools to the user. An inadequate user interface could make excellent underlying functionality unusable. In some cases it would be desirable to identify both the cause of user dissatisfaction due to the shortcomings of specific tools, and to identify the value which specific functional components contribute to the perceived value of the entire application, the AXMEDIS framework with integrated tools.

An experimental test of the contribution of individual tools to the performance and quality achieved with the complete framework would be carried out by isolating tools, i.e. by switching certain tools on and off, and comparing user performance and user satisfaction. This procedure only makes sense if the remaining functionality still leaves a fully functional framework. This would be the preferable approach from a methodological point of view, but it may not be possible to implement it.

To separate the value of tool functionality based on the judgments of users in an analytic manner is a challenging task. If the mapping of tools to framework functionality is clear, it may be possible to separate the value of specific functionality to user satisfaction (for example metadata). Provided that the conditions are met, it may make sense to carry out an analysis of user choice and preferences for the separate features and functions of AXMEDIS framework. One critical condition is that a sufficiently large number of users experienced with the AXMEDIS framework can be studied.

It should be noted that the technical performance of system components is measured separately and with different methods, refer to the Measurement Manual (WP5.1) for this issue.

4.6 User needs and preferences

The needs of users are not fixed in the sense that precise requirements, constraints, and preferences are maintained under all conditions, but there is a certain amount of "elasticity" such that the user is prepared to trade one attribute for another one. A critical trade-off in complex IT applications is between ease-of-use and learning requirements: It is not possible to make complex procedures simple for the user beyond a certain level without the need for a certain amount of initial learning effort. Some functions may simply be too complex in a given context, and might not provide sufficient value for users and customers. Another critical trade-off can be observed with performance parameters, where speed and accuracy often compete in work processes, and high security often interferes with easy to use interfaces.

The final configuration of the AXMEDIS framework, and the long-term marketing should take these trade-offs into account. If a certain set of functionality is too complex to use, or takes too much learning effort, it must either be simplified in order to become acceptable to users, or excluded in the final version of the AXMEDIS framework. It is possible to study these trade-offs with methods taken from microeconomics and market research, but these methods are applicable only at a stage when the AXMEDIS framework is fairly well developed as a demonstrator or working prototype (in year 2-4 of the AXMEDIS project).

A trade-off analysis is based on the fact that users who have some experience with the AXMEDIS framework are quite capable to answer questions which permit the analysis of user preferences in terms of trade-offs. A meaningful (quantitative) analysis demands that data from a substantial number of representative users are available. If this is not the case, then interviews and rating techniques may allow the collection of data which give an indication of the trade-offs which users consider when selecting products for use and purchase.

Given the high cost of conducting a trade-off analysis, these will be considered towards the later stages of the project where feasible, and if the expected results justify the effort. This would be the case if it appears desirable to know the value of offering specific configurations of AXMEDIS framework to users, and when there are precise hypotheses regarding specific trade-offs.

The guiding question will be to understand the contribution which the use of innovative algorithms and functionality makes to the value offered by the AXMEDIS framework to the user.

The result would allow estimates of the value of adding specific functionality to the AXEMDIS framework, and would indicate which main features users would like to see integrated into the framework.

4.7 Quality criteria which the users will apply

User validation of the AXMEDIS tools will refer to quality dimensions which are relevant for prospective users. The main dimensions of usability quality which users apply to assess the value of ICT applications are quite well known. Quality dimensions can be measured, however, no measures exist which measure any quality dimension perfectly. The specific quantitative values of these in the particular AXMEDIS context are not known, because quality is not the same for different users, and in a given situation some attributes are more important for a user than others. Therefore it is useful to let users rate the importance of quality dimensions.

A measurement of quality refers to either an object or to a product attribute or feature. For example, performance of a product may refer to the hardware/software, to the skill of the user or to the content provided by the product. A system may be efficient for one task, but not for another. It has become good practice to discriminate between the factors system, user, context and task.

The quality dimensions are usually summarized as

- **Effectiveness** is the ability to actually carry out tasks successfully. **Efficiency** is the cost in terms of time and other factors to carry out the task.

- **Robustness** describes how well a system can cope with user errors (e.g. undo function provided, understandable error messages). **Effort for error correction** describes the cost in terms of time and other factors to recover from errors.

- **Learning effort** required to be able to use a product efficiently. Learning is a function of training and practice. It reduces the cost of task performance for the user. The AXMEDIS tools should be easy to learn so that users can rapidly start getting work done, and they should allow users to reach an acceptable performance level within a specified time. Initial performance is a key quality issue because any user of a system must, at some point, use it for the first time. Training and practice can improve performance continuously. For a good product the performance of mature users should increase considerably.

- **Added value** can be measured by counting the new features offered by a product. The AXMEDIS tools will be of economic value for the customer if the **cost / benefit** relation is positive compared to the traditional way of work. The added value of professional and commercial systems usually is increased productivity and the ability to execute work processes faster, more flexibly and with higher quality. However, economic value for the customer need not be added value for the user. From a users point of view the added value can be improved quality of life.

- **Integration into a product family and migration**. Many software products enter the market as part of product families (e.g. Microsoft Office), have a corporate identity, common look and feel, run on the same platform. The integration of a new product can be achieved through consistency of all components with existing products. This will save the customer cost for training. For the user the advantage of using an integrated product will be reduced learning effort through transfer of knowledge. If the user knows already how to operate products of a product family, it will be easier to use another product of that family due to consistent dialogue structures. From the user point of view consistency is highly desirable. Producing consistency requires an extensive program of agreed styles for common appearance, structure, navigation and other features. Guidelines, standards and tests for consistency.

- **Subjective assessment** (affect) of the quality of the AXMEDIS tools by users. Positive user experience, user satisfaction with a product and acceptance of a product is important for market success. (Measurement results for efficiency, effectiveness, and user satisfaction may differ!)

A more detailed discussion of quality factors which describe "quality of use" can be found at www.VNET5.org. For the various quality factors there are measurement instruments available (such as checklists, questionnaires, rating procedures, experiments etc) which can be selected such that the best and most cost-effective method is used for any given project. The choice of appropriate methods depends upon business objectives, quality objectives, and project constraints.

The preferred approach in human-computer interaction is to let users carry out defined tasks, with a defined objective and end point, under controlled conditions, and to measure user behaviour quantitatively. This may be complemented by the measurement of subjective user assessments. The proven and widely used methods for usability evaluation and user assessment do, however, not include means to measure added value and integration into product family and provide no means for the evaluation of the specific challenges for the AXMEDIS project. To measure these quality factors, new methods have to be developed.

4.8 Benchmarking and business benefit

Benchmarking of new products against established or competitive ones is the essential indicator for the likely success of the product on the market. It guides management decisions on future investment, marketing, and regarding the direction of further development effort (including the exploitation of research results). Benchmarking relies on a systematic comparison of system features according to the criteria which the future users and customers will apply. While established products must make sure that they do not show deficits in respect to new competitors, new competitors must demonstrate clear advantages and benefits in comparison with existing products. In addition, benchmarking helps to indicate the strengths of new applications, and to identify the most promising application domains and markets.

In the case of AXMEDIS, we can benchmark the production process today, or before AXMEDIS, but it would be interesting (also for exploitation reasons) to research other collaborative production tools.

The measures of user performance employed for user testing are applied, in addition to an estimate of the business benefits in quantitative terms (cost, speed etc), and non-monetary terms (quality, image, attractiveness). The result is either an estimate of the total cost of the business processes analyzed under different conditions. The most critical condition for meaningful benchmarking is the right choice of standards and comparable systems to test against. Selecting appropriate competitive solutions will be a challenge for AXMEDIS as the nature of the project is to create a unique framework that does not face simple equivalent competition. More likely, the choice of benchmarks to compare AXMEDIS benefits would need to come from a combination of different production tools, collaborative tools and content management solutions. This kind of competitive benchmarking will be explored, but may not provide sufficiently robust evidence at this stage of the market to be of use to AXMEDIS.

Assessment of business benefit is the attempt to quantify the benefit for the customer in relation to the total cost of ownership. This process is simpler than the benchmarking of competing solutions, as it simply benchmarks the cost-benefits of AXMEDIS introduction versus the status quo, or “do nothing” scenario. The results should provide basic parameters to explain the value of the introduction of AXMEDIS technology in terms of return on investment. This is a standard approach in market research and product strategic planning. It will be applied in a selective manner to the most critical aspects of AXMEDIS framework and tools. The results should be among the main arguments presented to users and customers for the uptake of the new technology.

The procedure which we suggest to use for benchmarking can be outlined as follows:

- Identify the main business processes.
- Estimate the high-level process cost for the key processes.
- Estimate the high-level value added by each process in order to model the value chain.
- Provide comparative estimates for alternative and competing applications (at minimum, the ‘do nothing’ scenario, ie. the costs before introduction of AXMEDIS).
- Identify the non-monetary benefits, and rank the value of these for the owner of the system.

The result gives an estimate of the total cost of the application, and allows cost versus benefits sample calculations for future deployments of the technology in comparison with alternatives.

In addition to cost estimates based on total cost of ownership, an investigation into non-monetary cost, benefits, and constraints (such as privacy, traceability of decisions, DRM, etc), can be carried out for the user partners / demonstrators. The information used in these investigations will be derived from users of the AXMEDIS framework and tools, who can supply details about desirable and useful features and properties of the new applications which they have been using. The information supplied by users becomes more realistic after demonstrators have been available for field trials.

An approach for AXMEDIS about the level of detail required in cost analysis, the differences between differing content forms and distribution networks, and also the value add of reselling to new channels will be proposed in chapter 5.5.

The analysis of the business benefits is done in order to provide quantitative parameters and arguments for the adoption of AXMEDIS technology. For this purpose the application processes which can benefit most will be indicated, and the strategic objectives which are served.

A number of steps are carried out in sequence to arrive at an analysis of the business benefit of new information services:

1. Define the tools which will be produced, and their potential impact on the modification of business processes.
2. Analyze the competitive situation. Both existing and future tools are taken into account.
3. Develop a cost model which focuses on process cost estimates for the most frequently executed user processes, including realistic estimates of personnel cost and training cost.
4. Assess the strategic benefits. This is done by scoring and relative rating by strategic planning experts from the foreseen user organizations. The data obtained are ratings rather than objective measures.
5. Produce weights for the cost factors and the strategic objectives and calculate results.

The cost estimates for specific processes (such as composing specific types of objects, searching for a specified set of objects in the data base) can be fairly precise. The strategic benefits are only represented by momentary rating scores with a strong subjective basis, and can only be interpreted as such. The value of such ratings is therefore critically dependent upon the participation of the decision makers who will in the end make the decisions on the basis of the available data.

A factor to be considered will be the risks and requirements associated with the introduction of new technology, and the learning cost for the personnel and the organization affected.

In the next chapter 5 the methods which we propose for adoption up to month 18 are presented in detail. A workshop with representatives from all user partners will be conducted when initial integrated prototypes of the AXMEDIS tools are available. A coordinated procedure will be agreed among all participants and set into practice by using common materials.

5 Methods

A sizable portfolio of methods for user tests and validation with similar aims, but different strengths and weaknesses exists for practical implementation of the approach outlined above. Rather than deciding early on particular methods, we consider it better to be open to select the most appropriate methods for the research questions at hand when all the constraints for the AXMEDIS tools, demonstrators, take-up actions have been identified in full.

We discuss the applicable methods for the AXMEDIS user validation and the topics where some work has to be done to extend or adapt methods. The user validation plan in the appendix lists the relevant parameters and constraints which suggest the choice of the methods for user validation. This is not a fixed plan for execution. It represents the set of preferred options among which a choice according to the constraints at the time of testing can be made.

The elaboration of additional methods to be used for user validation in the test of demonstrators, and for business benchmarking after the first 18 months, are subject of the work carried out in the years 2 to 4.

5.1 Requirements analysis phase

The results of the requirements analysis phase primarily provide input for the developers, but are also closely linked to user validation because the user tasks for testing (test cases) and the evaluation criteria are derived from the requirements. Requirements are also subject to testing, because documented requirements may deviate significantly from the actual requirements which users apply in their productive use. As a consequence the documented requirements do not become fixed, but will be updated with feedback and comments from the expert user group and as experience with the new AXMEDIS tools is collected in increasingly realistic settings.

Because of the innovative character of the AXMEDIS tools the emphasis was placed upon a thorough and profound understanding of the application context, and experienced domain experts from the AXMEDIS user partners are available who are able to inform the development team about all relevant aspects which will determine the acceptance by users.

Appropriate approaches to requirements analysis have been used in the first 6 months of the project (scenarios in narrative form with diagrams and use cases). These results have been tested with the user partners and a group of external expert users. Further user group meetings are planned.

The WP2 activities have analyzed the context of use to a considerable extent:

- User groups and roles were identified.
- Traditional and desired workflow procedures were described on a high level.
- Technical environments implemented at the premises of the user partners were described.
- Functional requirements of users were defined in detail (DE2-1-1a/-User-requirements and Use-Cases)
- Some non-functional requirements are more implicitly included in DE2-1-1a: eg. user friendly interface, easy to use, faster to use, robustness, best coverage of functions.

Non-functional requirements include the description of user characteristics such as prior knowledge and experiences, special needs, subjective preferences of users and customers and the description of the work environment. Further non-functional requirements must be derived from cost constraints. Non-functional requirements should be studied in more detail.

One focus of user-centred development is on minimising the risk through the development process. Risks can be reduced by understanding the different segments of the customer base. An in-depth analysis of the context of use will help to manage the risk to fail, because the context of use tells us who is going to use the product, what for, and under what circumstances. Innovative products must first be adopted by niche customers who find the product a total solution to their specific needs.

The physical and organizational context in which a product will be used represents a large number of constraints. Many of these are not spectacular issues, but have to be kept in mind when a new product is developed, because minor oversights and omissions can create considerable cost and delays at a later stage.

The concept of " Context of Use Analysis " maintains that the usability of a product is affected not only by the features of the product itself, but also by the characteristics of the users, the tasks they are carrying out, and the technical, organizational and physical environment in which the product is used. Context includes all factors which affect the usability of the product, excluding the functionality of the product itself. For practical purposes only parts of the analysis can be carried out and documented.

Context of Use Analysis is a structured technique for eliciting this information about the context in which a product will be used. The results are descriptions of

- characteristic groups of prospective users
- the tasks different groups of users intend to perform with the product
- the organizational and physical environments in which the product will be used.

This information will be collected with a Context of Use Questionnaire from the AXMEDIS user partners and maybe other potential customers. These data will be used to draw up profiles of the user groups. We also need to understand users' goals, needs and values. Users have lots of needs, some of them are more important (and more profitable to serve) than others. The aim is to spot product opportunities.

5.2 Usability inspection and user tests in the development phase

In the initial development standards and user interfaces guidelines for the design of information presentation and navigation structures, and samples which illustrate these, are highly effective. When used in a more rigorous form, style guides must be complemented by an inspection and review process which tests for the adherence to the guidelines. This is usually done by expert review - but not to be carried out by the developers themselves.

Inspection and design reviews during the early development phase should be carried out by system experts who are not involved in the development effort. They use checklists and test the system according to the defined use cases, assuming the role of a user. They report the results directly to the developers, and possibly involve the developers directly in the design review.

Effective methods are heuristic evaluation and cognitive walkthrough, where test experts systematically follow scenarios and use cases to exercise the system. This may be done in approximate form even with paper prototypes and specifications. The number of defects found is expected to be initially quite large. Experience shows that the less mature an implementation is, the faster will defects be found. Different expert evaluators, however, do not find the same defects, and not in the same order of sequence. It is therefore advisable to use two or three test experts to evaluate a design. In later development stages longer test sessions should be foreseen.

5.2.1 Relevant Standards

Agreed formal standards provide only a minimum quality requirement, but are a useful reference against which to compare new applications.

Organizations with a certified quality system such as ISO 9000 will usually have references to user interface development standards in their quality system. The procedures used, and the documentation standards should correspond to the quality system (which would also make procedures most efficient). A common requirement is the documentation of the tests which were carried out, and of the test results obtained.

This section provides a quick reference to the most important standards related to usability and user-interfaces.

Poorly designed user interfaces may cause problems and difficulties for the user. The objective of the **European Directive 90/270/EEC on minimum health and safety requirements for work with display screen equipment** is to prevent that health hazards are generated for workers and employees through the use of display screen units. The directive applies to display screens, regardless of the technology used, and workstations, which are defined as assemblies comprising display screen equipment, input devices, software and interfaces. The directive is addressed at employers and employees and to their representatives, and also to manufacturers and distributors of hardware and software. The principles are:

- Software must be suitable for the task.
- Software must be easy to use and, where appropriate, adaptable to the user's level of knowledge or experience. No quantitative or qualitative checking facility may be used without the knowledge of the workers.
- Systems must provide feedback to workers on their performance.
- Systems must display information in a format and at a pace which are adapted to users.
- The principles of software ergonomics must be applied, in particular to human-data processing.

The European Directive overlaps to a considerable degree with part 110, Dialogue Principles, of the **ISO 9241 The ergonomics of human system interaction** (previously called: Ergonomic requirements for office work with visual display terminals) and has been implemented in various ways in the member states as legislation relating to safety at work and accident prevention.

The European Directive is relevant for ICT products which are used at work. The main mechanism for testing the compliance of an application with the European Directive is the obligation of the employer to check and analyze that workplaces are not in conflict with these requirements. It is advisable that developers and vendors take into account the needs and requirements of their users and customers to comply with the European Directive in their application.

The software ergonomic qualities are not measurable by accepted objective measurement procedures. The current state of the art is the target against which the ergonomic quality of an application will be measured. Checklists which contain all relevant items are used for this purpose. It is also obvious that the level of quality of applications is evolving, and that the demands of customers and users are rising accordingly.

ISO 9241 part 151 Software ergonomics for World Wide Web user interfaces (was ISO 23973 Web Usability Standard) uses a reference model for web design comprising design, process and evaluation. The Standard itself is focused on the design domain aspects only. These are: purpose and strategy, content and functionality, navigation and interaction, and presentation and media design. These aspects can be seen as different levels of the overall design process.

The other two parts of the model represent the process domain and the evaluation domain. These constitute important aspects for the user-centred development of Web applications but they are not elaborated in this International Standard.

ISO 9241 part 200 Human system interaction processes (was ISO 13407 Human Centred Design for Interactive Systems**)** provides the framework for the design process. The principles of a human-centred process are:

- Active involvement of users (or those who speak for them) and clear understanding of user and tasks requirements.
- Appropriate allocation of function (making sure human skill is used properly)
- Iteration of design solutions (therefore allow time in project planning)
- Multi-disciplinary design (but beware over large design teams)

Active involvement of users and clear understanding of user and task requirements is taking place with regard to the professional users of the AXMEDIS framework and tools. Active involvement of end users must be seen as a challenge because not all potential groups of users are known in advance and the requirements and preferences of these potential groups of users concerning the AXEMEDIS system are not known.

Iteration of design solutions is implemented in AXMEDIS, some iterations have been done, for example:
Creation of usage scenarios –> Validation and feedback by expert users –> Improvement of scenarios
Creation of early prototypes –> Validation and feedback by expert users –> Amendment of prototypes.

Multi-disciplinary design is implemented in AXMEDIS, where partners with various relevant expertise are involved, for example computer scientists, legal experts, domain experts (the user partners) and usability experts.

The standard also recommends four key human-centred design activities:

- Understand and specify the context of use (make it explicit: avoid assuming it is obvious)
- Specify user and organisational requirements (there will be a variety of different viewpoints and individuality)
- Produce design solutions (plural, multiple designs encourage creativity)
- Evaluate designs against requirements (involves real user testing not just convincing demonstrations).

More specific guidance can be found in

- **ISO/IEC 9126 parts 2 and 3 contain criteria for the evaluation of user interfaces**
- **ISO 14915 and IEC 61997 contain recommendations for multi-media interfaces**
- **ISO 10741 Dialogue Interaction**
- **ISO/IEC 11581, Icon symbols and functions**
- **ISO/IEC FCD 18021: Information Technology - User Interface for mobile tools (2001)**
- **ISO 9241 part 171 Guidance on software accessibility** (was ISO 16071)
- **ANSI/HFES-200-199x - Ergonomic Requirements for Software User Interfaces**

5.2.2 User Interface Design Principles and Style Guides

Style guides containing user interface design principles are described in documents, and often are also embedded in user interface toolkits. In addition to general style guides, which propose good practice in user interface design and suggest a common style, there are style guides for users with special needs and non-PC devices (eg. at www.w3c.com). For the basic design for alternative access devices a search among the most recent style guides for non-PC devices will be performed. Several general style guides which are available online are summarized below and the corresponding links are provided with the references.

5.2.2.1 Apple Computer, Inc. Macintosh Human Interface Guidelines

Metaphors: Take advantage of people's knowledge of the world by using metaphors to convey concepts and features of your application. Use metaphors that represent concrete, familiar ideas, and make the metaphors obvious, so that users can apply a set of expectations to the computer environment. Metaphors should suggest a use for a particular element, but that use does no have to limit the implementation of the metaphor. It is important to strike a balance between the metaphor's suggested use and the computer's ability to support and extend the metaphor. For example, the number of items a user puts in the Trash is not limited to the number of items a physical wastebasket could hold.

Explicit and Implied Actions: In the first step of a manipulation, the user sees the desired object on screen. In the second step, he selects or designates that object. In the final step, he performs an action, either using a menu command or by direct manipulation of the object with mouse or other device. This leads to two paradigms for manipulating objects: explicit and implicit actions:

- Explicit actions clearly state the result of manipulating an object. For example, menus list the commands that can be performed on the currently selected object. Explicit actions do not require the user to memorize the commands that can be performed on a given object.
- Implied actions convey the result of an action through visual cues or context. A drag-and-drop operation is a common example of an implied action. For implied actions to be apparent, the user must be able to recognize the objects involved, the manipulation to be performed, and the consequences of the action.

Direct Manipulation: Make users feel they are controlling the objects represented on the display. An on screen object should remain visible while a user performs an action on it, and the impact of the action should be immediately visible. For example, with drag-and-drop users can move a file by dragging its icon from one location to another.

User Control: Allow the user to initiate and control actions. Use progressive disclosure to present users with the most appropriate actions but offer alternatives when they exist. The key is to provide users with the capabilities they need and at the same time to provide warnings to avoid irreversible actions (for example the deletion of data by accident).

Feedback and Communication: Keep users informed about what's happening. Provide appropriate feedback, for example that user input was received and is processed. Use a progress indicator for lengthy operations. Provide feedback in a direct, simple and understandable way. Error messages should be precise. For example "There's not enough space on the disk to save the document") and possible actions the user can take to rectify "Try saving the document in another location".

Consistency in the interface allows users to transfer their knowledge and skills from one application to another. What You See Is What You Get (WYSIWYG). Where users can format data for printing, publishing to the web, or writing to film, DVD, or other formats, make sure there are no significant differences between what users see on screen and what they receive in the final output, or use a preview function if necessary.

Forgiveness: Encourage people to explore your application by building in forgiveness, by making most actions easily reversible. People need to feel that they can try things without damaging the system or jeopardizing their data. Anticipate common problems and alert users to potential side effects.

Perceived Stability: To give users a conceptual sense of stability, the interface should provide a clear, finite set of objects and set of actions to perform on those objects. For example, when a menu command does not apply to a selected object or to the object in its current state, the command is dimmed rather than omitted.
To help convey the perception of stability, preserve user-modifiable settings such as window dimensions and locations. When a user sets up his or her environment to have a certain layout, the settings should stay that

way until the user changes them. Providing status and feedback also contributes to perceived stability by letting users know that the application is performing the specified task.

Aesthetic Integrity: Information should be well-organized and consistent with principles of good visual design. Your product should look pleasant on the screen, even when viewed for a long time. Keep graphics simple, and use them only to enhance usability. Do not overload windows and dialogues with icons or buttons. Arbitrary symbols to represent concepts may confuse or distract users. Match a graphic element with a user's likely expectations of its behaviour. Do not change the meaning or behaviour of standard items.

Modelessness: Avoid using modes that lock users into one operation and prevent them from working on anything else until that operation is completed. If an application uses modes, there must be a clear visual indicator of the current mode, and it should be very easy for users to get in to and out of the mode. For example, in many graphics applications, the pointer can look like a pencil, a cross, a paintbrush, or an eraser, depending on the function (the mode) the user selects.

5.2.2.2 IBM Corporation. Object-Oriented Interface Design: IBM Common User Access Guidelines

Simplicity: Don't compromise usability for function. Keep the interface simple and straightforward. Basic functions should be immediately apparent, while advanced functions may be less obvious to new users.

Support: Place the user in control and provide proactive assistance. Give users control over the system. Enable them to accomplish tasks using any sequence of steps that they would naturally use. Do not limit them by artificially restricting their choices to your notion of the "correct" sequence. Ideally, assistance should provide users with knowledge that will allow them to accomplish their tasks quickly.

Familiarity: Build on users' prior knowledge. Allow users to build on prior knowledge, especially knowledge they have gained from experience in the real world. A small amount of knowledge, used consistently throughout an interface, can empower the user to accomplish a large number of tasks.

Avoid the tendency to employ consistency without understanding your users, their tasks, and their shared experiences. When choosing a dimension within which to be consistent, seek to understand what the user expects and be consistent with those expectations. Providing a familiar experience is the ultimate use of consistency in which a truly intuitive interface will result.

Obviousness: Make objects and their controls visible and intuitive. Use real-world representations in the interface. Real-world representations and natural interactions (direct action) give the interface a familiar look and feel and can make it more intuitive to learn and use. Icons and windows were early attempts to draw on user experiences outside the computing domain. The controls of the system should be clearly visible and their functions identifiable. Allow users to interact directly with objects and minimize the use of indirect techniques.

Encouragement: Make actions predictable and reversible. A user's actions should cause the results the user expects. In order to meet those expectations, the designer must understand the user's tasks, goals, and mental model. Users should feel confidant in exploring, knowing they can try an action, view the result, and undo the action if the result is unacceptable. Users feel more comfortable with interfaces in which their actions do not cause irreversible consequences. Even seemingly trivial user actions, such as deselecting or moving objects, should be reversible. Avoid bundling actions together, because the user may not anticipate the side effect.

Satisfaction: Create a feeling of progress and achievement. Allow the user to make uninterrupted progress and enjoy a sense of accomplishment. Reflect the results of actions immediately; any delay intrudes on users' tasks and erodes confidence in the system. Immediate feedback allows users to assess whether the results were what they expected and to take alternative action immediately. For example, when a user chooses a new font, the font of all applicable text, or of sample text, should change immediately. The user can then decide if the effect is what was desired and, if not, can change it before switching attention to something else.

Availability: Make all objects available at all times. Users should be able to use all of their objects in any sequence and at any time. Avoid the use of modes, those states of the interface in which normally available actions are no longer available, or in which an action causes different results than it normally does.

Safety: Keep the user out of trouble. Protect them from making errors. Eliminate the opportunity for user error and confusion. The burden of keeping the user out of trouble rests on the designer. The interface should provide visual cues, reminders, lists of choices, and other aids, either automatically or on request. Contextual and hover help, as well as agents, can provide supplemental assistance. Users should never have to rely on their own memory for something the system already knows, such as previous settings, file names, and other interface details. If the information is in the system in any form, the system should provide it. Two-way communication may be necessary at times to allow users to clarify or confirm requests, or to remedy a problem, and may be used to help users reach their goals.

Versatility: Support alternate interaction techniques. Allow users to choose the method of interaction that is most appropriate to their situation. Interfaces that are flexible in this way are able to accommodate a wide range of user skills, physical abilities, interactions, and usage environments. Providing a range of interaction techniques recognizes that users are individuals with different abilities and situations. The differences include disabilities, preferences, and work environments.

Personalization: Allow users to customize. The interface should be tailored to individual users' needs and desires. Customization can help make an interface comfortable and familiar for a user. In an environment where one user may be using several computers, or another environment with many users with different "profiles", personalization will be a mean to tailor to the needs of each single user.

Affinity: Bring objects to life through good visual design. The goal of visual user interface design should be to develop an intuitive and familiar representation that is second nature to users. The following visual design principles promote clarity and simplicity in the interface:

- Subtractive design **-** reduce clutter by eliminating any visual element that doesn't contribute directly to visual communication.
- Visual hierarchy **-** by understanding the importance of users' tasks, establish a hierarchy of these tasks visually. An important object can be given extra visual prominence. Relative position and contrast in colour and size can be used.
- Affordance **-** when users can easily determine the action that should be taken with an object, that object displays good affordance. Objects with good affordance usually mimic real world objects.
- Visual scheme **-** design a visual scheme that maps to the user model and lets the user customize the interface. Do not eliminate extra space in your image just to save space. Use white space to provide visual "breathing room."

5.2.2.3 Microsoft Corporation. The Windows Interface Guidelines for Software Design: An Application Design Guide

User in Control of the software. The user plays an active rather than reactive role. Because of widely varying skills and preferences, users should be able to personalize interfaces. Software should be as interactive and responsive as possible. Avoid modes where possible.

Directness. Software should be designed such that users can directly manipulate software representations of information. Familiar metaphors provide a direct and intuitive interface for user tasks. By allowing users to transfer their knowledge and experience, metaphors make it easier to predict and learn the behaviours of software-based representations. Metaphors support recognition rather than recall. Meanings associated with familiar objects are remembered more easily than command names.

Consistency allows users to transfer existing knowledge to new tasks, learn new things more quickly, and focus attention to tasks. Consistency should be designed into an application, the operating environment, into metaphors.

Forgiveness. Users like to explore an interface and often learn by trial and error. An effective interface allows for interactive discovery and that users make mistakes. Effective design avoids error prone situations. It also accommodates potential user errors and makes it easy for the user to recover.

Feedback to user's actions helps confirm that the software is responding to user input and communicates details that distinguish the nature of the action. Effective and good feedback is timely and presented close to the users interaction.

Aesthetics. Visual design is an important part of an application's interface. Visual attributes provide valuable impressions and communicate important cues to the interactive behaviour of particular objects.

Simplicity. An interface should be simple (but not simplistic), easy to learn, and easy to use. It must also provide access to all functionality of an application. Maximizing functionality and maintaining simplicity work against each other in the interface. An effective design balances these objectives and is one of the purposes in defining project guidelines. Simplicity also correlates with familiarity; things that are familiar often seem simpler. Whenever possible, try to build connections that draw on your users' existing knowledge and experiences.

5.2.2.4 Web Content Accessibility Guidelines

The Web Content Accessibility Guidelines explain how to make web content accessible to people with disabilities. The primary goal of these guidelines is to promote accessibility. Following them will also make web content more available to all users, whatever user agent they are using (e.g., desktop browser, voice browser, mobile phone, personal computer, etc.) or constraints they may be operating under (such as noisy surroundings, under- or over-illuminated rooms, in a hands-free environment, etc.). Following these guidelines will help people find information on the web more quickly. These guidelines do not discourage content developers from using images, video, etc., but rather explain how to make multimedia content more accessible to a wide audience.

5.2.2.5 Other relevant guidelines

Other relevant guidelines may be those of Sun Microsystems for Java and Open Look, those of the Open Software Foundation, and principles and techniques to enhance the visual quality and usability of products described by Mullet & Sano (1995).

Company-internal style guides from AXMEDIS user partners may be applied where applicable. These issues were addressed in detail in the AXMEDIS deliverable DE3.1.3 Content selection guidelines, in the context of content rather than applications.

5.2.2.6 Relevant principles for AXMEDIS User Interface Design

The design principles presented in the previous chapters have much in common and share a user-centred approach. The table below shows the overlap of the three most relevant UI design guidelines.

Apple	**Microsoft**	**IBM**
Metaphors	Directness	Familiarity: Build on users' prior knowledge Obviousness: Make objects and their controls visible and intuitive
Direct Manipulation	Directness	
User Control	User in Control	Support: Place the user in control and provide proactive assistance
Perceived Stability		Personalization: Allow users to customize
Modelessness	User in Control	Personalization: Allow users to customize
Modelessness		Availability: Make all objects available at all times
Feedback and Communication	Feedback	
Consistency	Consistency	Familiarity: Build on users' prior knowledge
Forgiveness	Forgiveness	Safety: Keep the user out of trouble
Forgiveness	Forgiveness	Encouragement: Make actions predictable and reversible
Aesthetic Integrity	Aesthetics	Affinity: Bring objects to life through good visual design
Aesthetic Integrity	Simplicity	Simplicity: Don't compromise usability for function
		Satisfaction
		Versatility
Explicit and Implied Actions		

The most important concepts of the presented guidelines are summarized here to be considered for design and development of the AXMEDIS framework and tools:

- **Be consistent.** Consistency should be maintained across multiple platforms, operating systems, AXMEDIS tools. Users may need to switch between tools. Consistency is also achieved by creating device independent I/O methodologies.

- **Provide shortcuts** for frequent users. This rule takes into account the need for simplicity and user control with personalization. Time is often critical for AXMEDIS users. Reducing the number of operations needed to perform tasks with scripts and/or macro functionality is a key factor to increase efficiency.

- **Design for error recovery**. Error prevention and simple error handling, for example Undo functions, are essential.

- **Informative feedback**. The tools should offer informative feedback to establish a communication with the user.

- **Personalization**. Users differ with respect to the set of tasks they have to perform, skill levels, usage patterns, preferences. Users like a personal view of the application: specific colors, fonts, arrangement of icons on the desktop.

Multilingualism is another relevant issue. The AXMEDIS framework and tools will facilitate work across national, linguistic, and cultural boundaries and is intended to be installed all over Europe, where different languages and cultures are joined with the same idea in mind. We have to cope with the fact that not all AXMEDIS users will be familiar with the English language.

A number of additional factors may affect the design of AXMEDIS tools from technical requirements, legal requirements, to the requirements of different groups of users. The AXMEDIS project aims to integrate tools in a framework to make work tasks more efficient and effective. A challenge for developers is how to address tools that can be accessed through a variety of terminals, platforms with a number of different CMS, and that may vary in terms of interaction resources (screen size, processing power, CMS, platform).

There is no simple solution when a design trade-off occurs. For usability inspection the following factors should be considered:

- Additional features may affect user performance, complexity, stability, maintenance, and support costs.
- Simplicity is not the same as simplistic. Making the user interface simple to use is often hard work.
- Security may interfere with usability. AXMEDIS should provide a secure execution context to ensure IPR and copyright.

5.2.3 Cognitive Dimensions

The Cognitive Dimensions framework (Li, Grundy, Amor & Hosking 2002; Beckwith, Burnet & Cook, 2002; Burnett, Arredondo-Castro & Atwood, 2002) is supposed to help developers to assess the usability of a language for scripting rules at design time. The benefit lies in having an explicit list of design attributes – summarizing empirical evidence about programming language or environment attributes important to human problem solving - which can be checked and referred to throughout the design project.

This analysis can be carried out on its own or as part of a usability inspection described in the next subsection.

Design principles	**How to check whether design principles have been taken into account**
Abstraction gradient	What are the minimum and maximum levels of abstraction? Can fragments be encapsulated?
Consistency	When some of the language has been learnt, how much of the rest can be inferred?
Error-proneness	Does the design of the notation induce ‘careless mistakes’?
Hidden dependencies	Is every dependency overtly indicated in both directions? Is the indication perceptual or only symbolic?
Premature commitment	Do programmers have to make decisions before they have the information they need?
Progressive evaluation	Can a partially-complete program be executed to obtain feedback on “How am I doing”?
Role-expressiveness	Can the reader see how each component of a program relates to the whole?
Viscosity	How much effort is required to perform a single change?
Visibility and juxtaposability	Is every part of the code simultaneously visible (assuming a large enough display), or is it at least possible to compare any two parts side-by-side at will? If the code is dispersed, is it at least possible to know in what order to read it?

5.2.4 Usability Inspection

Usability inspection is not user testing. It is an activity based on rigorous testing procedures carried out by experts. The scope of inspection is to identify severe defects in application design and usability problems, to detect the nature of these problems and to suggest recommendations for possible solutions.

Usability inspection is applied best in the early phases of the development process when mock-ups or prototypes of a new application are available. Usability inspection follows similar principles as code inspection for the assurance of minimum quality of a software product (assuring that it works as specified, and corresponds to the minimum quality requirements plus additional ones which were specified, for example the adherence to style guides, standards - including internal requirements).

The general process is the following:

- Recruit inspectors (usability experts)
- Pre-evaluation training to familiarize the experts with the objectives of the inspection
- Several experts inspect a mock-up or prototype, simulate the performance of tasks, and record defects and usability problems.
- After the inspection the collected data are elaborated and the severity of usability problems is rated.
- Finally the results and recommendations for improvement are communicated to the development team.

The main benefits of inspection are that it is cost-effective because no test users are needed and that it can serve as a preparatory phase for a more focused user testing. Certain tests- such as the adherence to styles - can not be carried out with users. These require inspection methods to be carried out by experts who are familiar with the implementation of the styles required.

Several types of inspection methods are suggested:

- Heuristic Evaluation
- Cognitive Walkthrough
- Pluralistic Walkthrough
- Formal Usability Inspection

Heuristic Evaluation has been developed and applied systematically by Jacob Nielsen (1994), who has derived 10 usability heuristics from a factor analysis of 249 usability problems. Heuristic evaluation based on these principles can be regarded as a very well founded method, which is supported by a number of empirical studies about its effectiveness. The usability heuristics are (Nielsen 1994, p. 30).

1. "*The system should always keep users informed about what is going on, through appropriate feedback within reasonable time."*

2. *"The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order."*

3. *"Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo."*

4. *"Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions."*

5. *"Even better than good error messages is a careful design which prevents a problem from occurring in the first place."*

6. *"Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate."*

7. *"Accelerators* - *unseen by the novice user- may often speed up the interaction for the expert user to such an extent that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions."*

8. *"Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility."*

9. *"Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution."*

10. *"Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focussed on the user's task, list concrete steps to be carried out, and not be too large.*"

Ideally 3-5 evaluators inspect the user interface systematically with regard to these basic heuristics describing the characteristics of usable interfaces. The inspection takes 1-2 hours depending on the size of the application. The evaluators examine the interface and judge its compliance with the recognised usability principles (the heuristics). After the inspection the evaluators elaborate and summarize the problems they have found. In a debriefing session the results from all inspectors are discussed and aggregated, and the severity of the defects and usability problems is rated.

The severity of usability problems may vary:

- No user would have problems; Some users might have difficulties; Most users would have problems
- Frequency with which a defect or problem occurs: Always, often or rarely
- Impact of the defect or problem if it occurs: Easy or difficult for users to overcome the problem

The result of the debriefing session will be a list of defects and usability problems ordered according to severity, and recommendations for improvements.

The process of carrying out one of the other inspection methods (Cognitive or Pluralistic Walkthrough, Formal Usability Inspection) is similar to Heuristic Evaluation, except that these alternative inspection methods also include users, domain experts and developers and instead of the 10 heuristic evaluation principles other guidelines and more formal rules for carrying out the inspection are used.

It is important to be aware of the fact that inspection methods will render a decreasing number of defects as testing continues. In addition, different inspectors do not home in on quite the same set of defects. It is essential therefore is to use several inspectors.

In the initial testing cycles both a larger number of problems, and more severe defects will be found. It does not make sense to be exhaustive in the initial tests, but to focus on a sound assessment of the major defects, and on the formulation of the change requests handed to the development team. In the next test cycle a somewhat different set of defects will be identified, and the severity of defects found will decrease at the same time as the visible quality of the solution improves. (At this stage it makes sense to think about the involvement of users, who should be tested only with a reasonably stable application which allows them to carry out some tasks successfully.)

A workshop will be held to introduce members of the development teams to user interface guidelines and inspection methods applied in the early phases (AXMEDIS conference, Florence, 2 December, 2005).

5.3 Field trials and acceptance test

Users who are presented with incomplete and defective software become frustrated very quickly and can not provide much constructive feedback. Users should only be involved in tests as soon as the development team is confident of the quality of the result of development.

User tests are expensive in terms of effort expended, and it may be difficult to find a sufficient number of motivated and informed users. User tests should be well planned, and must correspond to minimum methodological constraints. Experts should help with the planning of test sessions. In order to demonstrate shortcomings of the application and problems of users and to convince developers, it may be useful to videotape relevant episodes of test sessions for later review and presentation to the developers.

There must be an understanding that the tests are carried out with the aim to identify as many problems as possible, and to find better solutions immediately. The number of subjects to use for testing can be small initially, but really conclusive tests require a minimum of about 8 users, but the numbers required can be considerably larger.

A valid and conclusive acceptance test with users provides the information for subsequent management decisions. This is the domain of realistic user trials, and field trials. Careful planning helps considerably to obtain interpretable and valid results at the end. The test conditions, instruction and training of users, data analysis procedures and benchmarks for comparison have to be defined. It is quite disappointing to arrive at un-interpretable results after considerable effort for user tests has been expended.

The selection of the measures, methods, and the planning of the field trials should be done by experts in the field. At this stage the user test activities are planned in detail up to month 18, when the first integrated prototypes are available to be inspected. A workshop will be held to introduce personnel from user partners to the test methods applied in the subsequent phases.

Field tests will be initiated in connection with the development of demonstrators. The methods to use **-** including logging and analysis of user data **-** will be elaborated according to the context in which the user partners will carry out their field tests.

5.3.1 Using the SUMI questionnaire to measure user satisfaction

SUMI is a validated instrument for measuring user satisfaction. It is designed to be used with end users of a software product being evaluated. SUMI enables experts concerned with the usability of a product (eg project manager, software developers, and other stakeholders), to obtain objective and trustworthy data about the subjective assessment of the product by users.

SUMI was designed to be used primarily to evaluate those systems which are generally known as 'office software' but in practice it has been used to measure a wide variety of software (from space station control systems to games) Computer users are likely to implicitly compare their level of satisfaction with any kind of software to the standard office software suites of which they have routine experience. When using SUMI to assess the usability of a prototype during development, a SUMI 'profile' can indicate the weak aspects of the prototype.

Users normally require about ten minutes to complete the inventory after having used the software for at least an hour.

User Satisfaction is one of the three key aspects of usability as defined by the ISO 9241 standard, part 11 (Efficiency, Effectiveness, Satisfaction). Satisfaction is an important variable. Low satisfaction scores inevitably mean that usage of the product either is or will be accompanied by feelings of stress with the end users. The most obvious signs of stress are lack of concentration, a tendency to make elementary mistakes, leading to increased use of help, and eventually the users refuse to use the software.

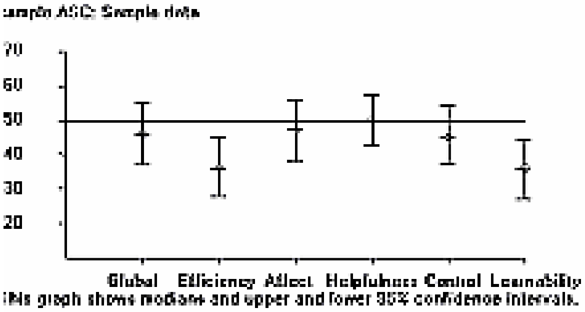
User Satisfaction can be subdivided into five aspects (measured with SUMI):

- Efficiency refers to the user's feeling that the software enables them to perform the task(s) in a quick, effective and economical manner.
- Affect is a psychological term for emotions. It refers to the positive user feeling of the user being mentally stimulated and pleased as a result of interacting with the software.
- Helpfulness refers to the user's perceptions that the software communicates in a helpful way and assists in the resolution of operational problems.
- Control refers to the feeling that the software is responding in an expected and consistent way to input and commands.
- Learnability refers to the feeling that the user has that it is relatively straightforward to become familiar with the software.

Sample Size: SUMI yields reliable information when used with appropriate sample sizes. A sample of ten or more users per system being evaluated is required to obtain statistically reliable results. Although SUMI has been used on samples as small as 3 or 4, its use in these circumstances was primarily for diagnostic purposes. On the other hand, sometimes it may only be possible to get a small handful of users. A small amount of information is better than no information at all, but results from small samples must be interpreted cautiously and critically with common sense.

The statistical analysis can be carried out with the scoring program SUMISCO. The output of SUMISCO can be divided into three components: Scale scores (Profile Analysis), User scores, and Item Consensual Analysis.

Example of SUMI Output



5.4 Evaluation of the usefulness of metadata for information retrieval

Metadata (data about data) constitute the information that enables the effective, efficient, and accurate retrieval and use of AXMEDIS objects. There is an increasing awareness of the need for improvement the quality of metadata. The usability of metadata depends on the quality of the metadata.

A consistent capturing, updating and utilization of metadata throughout the content creation, production and distribution processes is the basic condition for an optimization of the workflow, and thus for working in a more economic and cost-effective manner. The technology developed in AXMEDIS (e.g. descriptor /

metadata extractor for audio and video content) may not be inexpensive, but is supposed to pay off by an extensive automation and a virtually constant availability, and will ensure at the same time useful metadata and related contents.

The usefulness of the content of the AXMEDIS database will be directly influenced by the consistency and accuracy of the harvested metadata. High quality metadata supports the functional requirements of the AXMEDIS framework and tools (fit for purpose). The quality of metadata will affect the efficiency of technology users and the quality of products and services offered to end-users.

The evaluation of the usefulness of metadata is not part of the available portfolio of instruments for usability testing (which is largely focused on procedural and display aspects of user interaction). There are no empirically validated assessment methods for metadata usability. A solution to assess the usability of metadata will be to create a questionnaire for this purpose.

5.5 Cost/benefit analysis, value analysis, analysis of business benefits and benchmarking

The main objective of the field trials of AXMEDIS tools with real users is to inform about the quality of the solutions which were developed. This information is needed by customers, investors, and by decision makers who are involved in the implementation / purchase of new solutions.

The data to be provided should inform about

- Impact on the efficiency and effectiveness of the workflow investigated
- Quality of the result obtained by using AXMEDIS tools
- Motivating effect on users
- Direct and indirect value for the owner of the application system (decrease of production cost)

The data will be collected in carefully planned field trials.

The level of detail required in cost data will need to be sensitive to the user organisation’s wishes for competitive confidentiality and their ability to practically measure costs in their processes.

The data from each user will need to be categorized by:

- Nature of user organization, including, stages of the workflow covered such as authoring, packaging, integration, distribution, etc.
- Details of which AXMEDIS tools are used
- Type of content (simple text, audio, complex multimedia)
- Volume of content produced (the efficiencies and benefits will differ substantially between low-volume/high complexity content processes and high-volume/low complexity processes)
- Nature of main distribution channel
- Also, traditional distribution channel pre-AXMEDIS, and details of new channels enabled by use of AXMEDIS tools/framework.

In addition to measures indicating quality factors, the elasticity of demand exhibited in the behavior of users may be an important aspect to observe: The trade-off between development complexity versus efficiency in use as an influence on the design of future AXMEDIS tools or between the quality of the result and cost in performing tasks.

The field tests should provide initial data, which may be analyzed and discussed with decision makers. As a result some of the important trade-offs will be recognized: A question raised may be "How much more efficient must the work procedure be in order to justify an investment of xxx Euro?"

Simple procedures to investigate these relations are rating scales or "positioning", where experts are directly asked to assess these tradeoffs. A more reliable and more costly method is based on conjoint measurement, for which efficient computer based data collection methods are available, which can even be used via the WWW.

Another useful tool to be used will be a 'benefits register' which can be developed incrementally throughout the project. A starting point for AXMEDIS is shown in the table below. This can be updated by users leading to a comprehensive checklist of benefits at the end of the project that can be used as part of dissemination and exploitation. Where possible, the suggested quantifications can be measured during the field trials in order to provide actual data.

AXMEDIS benefits register

Benefit	Beneficiary	Quantification?	Strategic importance
Lead time reduction for content authoring	Content creators	Lead time in days when using AXMEDIS tools/ Lead time in days before use of AXMEDIS	High
Production cost reduction for content authoring	Content creators	Average unit cost with AXMEDIS/ Average unit cost before AXMEDIS	High
Access to wider range of content elements thanks to P2P for B2B	Content creators and integrators	Available library size with AXMEDIS/ Available library size before AXMEDIS	Medium

Costing of implementation

In order to complete the cost-benefit analysis, users will be required to record the costs of implementing AXMEDIS tools. These will include:

- Staff time
- Training costs
- Additional hardware and software
- Additional networking, communications costs
- Additional infrastructure requirements
- Etc.

In turn, this will allow for an estimation of investment requirement for AXMEDIS implementation, which in conjunction with the benefits quantification will allow for an estimation of return-on-investment.

As a conclusive analysis of the value of AXMEDIS technology this procedure will be considered. A favorable condition in the AXMEDIS project is the involvement of a substantial number of domain experts for content production and different types of distribution.

5.6 Useful Methods for AXMEDIS

Methods in the requirements phase		**Remarks**	**Further information**
Contextual Inquiry	In-depth analysis and observation of user behaviour in a realistic setting representative for the future application context. Close cooperation with users and various methods of analysis and forms of documentation may be used.	Can be resource-intensive. It may be difficult to involve users. Background knowledge in empirical behavioural research is useful.	Beyer & Holtzblatt (1998) www.incent.com
Volere requirements			Robertson & Robertson (2004)
Focus groups	Structured and directed meetings with representative users and application domain experts. The goal is to elicit needs and requirements from the user representatives.	A background in group management (mediation) is recommended, otherwise the validity of results may suffer.	Caplan (1990)
Context of use checklist	Checklists for analyzing all relevant aspects of the physical and organisation context of use.	Efficient and recommended to complement other analyses.	Bevan & MacLeod (1994)
Use Cases and Scenarios	Specification of user procedures and processes in semi-formal diagrams or narrative form	Also part of requirements specification methods	Constantine (1999) www.foruse.com

Methods in the design and implementation phase		**Remarks**	**Further information**
Heuristic evaluation	Inspection of prototypes by experts centered on ”heuristics” identified from a large number of design evaluations (eg. following user interface design guidelines, web design guidelines, accessibility guidelines, cognitive dimensions guidelines). The aim is to identify problems early and quickly and to assess the usability of a scripting language for content processing.	To be carried out by 1-3 experts unconnected to the development team. Efficient, but somewhat limited to finding design defects early in the development phase.	Nielsen (1992). www.useit.com

Cognitive walkthrough	Systematic inspection of the cognitive processes of users in the execution of specified task scenarios.		Nielsen & Mack (1994)
Style guides	Various style guide exist for application environments or minimum requirements.	May be partly applicable.	See References
Usability tests of prototypes	A participative test with users under controlled conditions. Data on user problems is collected by observations and "thinking aloud".	Experience in experimental behaviour research desirable. A more controlled and rigorous approach is used in systems tests.	VNET5 → resources
Prototyping	Used early in the design phase, either with paper prototypes, mock-ups, or early prototypes. Various forms of user tests may be applied.		VNET5 → resources
Methods in the systems test and field trials phase		**Remarks**	**Further information**
Performance measurement	Controlled and systematic measurement of user performance. This is not one method, but rather an approach which may be implemented in different ways. The aim is to demonstrate efficiency of workflow and process and benefits/added value for the customer. May include the measurement of learning time.	Can be resource intensive. Control of conditions and correct selection of samples of users are essential.	Dumas & Redish (1993) Rubin (1994)
User satisfaction measurement with SUMI or QUIS questionnaires	Complementary to performance measurement. The subjective assessment of system quality, which may be dissociated from objective parameters.	Cost effective, but not a replacement for performance measurement. Note that these questionnaires are validated measurement instruments, not to be confused with ad-hoc opinion surveys.	www.ucc.ie/SUMI
Questionnaires	Tailor-made questionnaires for the analysis of the Usefulness of metadata for information retrieval, for the analysis of cost/benefit and added value.		

6 Usability Inspection of the AXMEDIS Tools: Work performed

This chapter describes the usability inspection of the AXMEDIS Tools versions 1 to 2.5 performed from April to September. It includes a summary of high priority recommended actions and makes recommendations for future development.

The priorities of development were to quickly deliver basic AXMEDIS Tools that support the AXMEDIS business objectives: Something that works, looks good, and that people can and will use for content production and distribution because it speeds up and optimizes work through means of automating, accelerating and restructuring the process.

The aim of the work reported here was to:

- To inspect the usability of the AXMEDIS Tools
- To recommend specific changes to the user interface to help meet its usability goals better
- To achieve essential improvements of the user interface design quickly and with minimum iteration and disruption
- To make recommendations for subsequent user testing of the AXMEDIS Tools.

The chapter is structured into four sections:

- What Usability means for the AXMEDIS Tools,
- Method and procedure of assessment and evaluation,
- Detailed description of how good the AXMEDIS Tools are.
- Summary of the results, highlights barriers to use, and proposal of quick changes to improve the usability of the AXMEDIS Tools.
- Future activities.

6.1 What usability means for the AXMEDIS Tools

6.1.1 Supporting Business Goals

The AXMEDIS Tools should help users and customers to increase efficiency and thereby reduce cost visibly to justify the investment to switch to AXMEDIS Tools. The new functionality should add visible value for the users, notably: Access to more content, Web usability. Therefore the AXEMDIS Tools should be easy to learn, easy to use, enable users to carry out their work tasks, leading to high acceptance in the initial phase of use of each user.

6.1.2 Achieving the right Balance of Usability Goals

The following panel summarizes relevant usability goals for the AXMEDIS Tools. Many of these goals pull the design in different directions. Achieving a good balance involves making trade-offs and requires iterations to get it right. Here we are seeking to achieve a good and consistent design of the different tools as quickly as possible.

Visibility of System Status
The system should always keep the user informed about what is going on through appropriate feedback within reasonable time. The interface should help the user understand what is happening.

Match Between System and the Real World
The system should speak the user’s language, with words, phrases and concepts familiar to the user, rather than system-oriented, technical terms.

User Control and Freedom
Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. When users make a "mistake" or change their minds, the system must allow for "emergency exits" and backtracking. The system should support undo and redo.

Consistency and Standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. The system should follow platform conventions when they exist, and pay attention to consistency of format, icons, etc.

Help Users Recognize, Diagnose, and Recover From Errors
Error messages should be expressed in plain language. The UI should guide users in helpful ways using cues and messages expressed in plain language or using icons.

Error Prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Recognition Rather Than Recall
The user should not have to remember information from one part of the dialogue to another. Therefore objects, actions, options, and instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility
The interface should be flexible so that it is appropriate for novices and experts. Accelerators **-** unseen by the novice user **-** should speed up the interaction for the expert user.

Aesthetic and Minimalist Design
Interfaces should only contain relevant information. Extra information competes with relevant information and diminishes their relative visibility.

Help and Documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out. "Help" should be accomplished through good interface design and context-sensitive information.

6.2 Methods and Procedure of Usability Evaluation and Assessment

Usability Inspection during the early development phase is carried out by experts who are not involved in the development effort. (Certain tests - such as the adherence to styles - can not be carried out with users. These require inspection methods to be carried out by experts who are familiar with the implementation of the styles required.) They test the system according to defined use and test cases assuming the role of a user.

The number of defects found is expected to be initially quite large. Experience shows that the less mature an implementation is, the faster will defects be found. Different experts, however, do not find the same defects, and not in the same order of sequence. It is therefore advisable to use two or three test experts to evaluate a design. In later development stages longer test sessions should be foreseen.

Usability Inspection is not user testing. It is an activity based on rigorous testing procedures carried out by usability experts. The scope of inspection is to identify severe defects in application design and usability problems, to detect the nature of these problems and to suggest recommendations for possible solutions.

Usability inspection follows similar principles as code inspection for the assurance of minimum quality of a software product (assuring that it works as specified, and corresponds to the minimum quality requirements plus additional ones which were specified, for example the adherence to style guides, standards - including internal requirements).

The general procedure of the usability inspection which was carried out was the following:

1. Brief demonstration of AXMEDIS Tools by the developers. At the AXMEDIS meeting in Rome in March 2006 the AXMEDIS Tools were under development, with a lot of good technical thinking, a logical structure, workable user interfaces for the various tools. However, some of the Tools (P2P, Query) looked cluttered and uninviting, with a mix of visual styles. It had far too much text in some places, and a lack of simple instructions in others. It had some confusing screen flows, with some barriers to use the functionality offered. In short, aspects of the user interface design seemed awkward for many ordinary people to understand.

2. Exploration of AXMEDIS Tools (version 1, 2.2 and 2.3) guided by read me files, user guidelines, demonstration videos, to familiarize with the tools. A walkthrough of the AXMEDIS Tools using test case descriptions was carried out by a usability expert (who is not a task domain expert). This involved putting oneself in the role of a user, trying to achieve the main things such as a user will want to do. The evaluator notes problems encountered, and looks for good solutions. This is not a substitute for user testing, but can give rapid results in finding anticipated problems and generating practical solutions.

3. Usability inspection of the AXMEDIS Tools (version 1 to 2.4) with a checklist, simulation of task performance, and recording of design deficiencies, usability problems and defects. After the inspection the collected data was elaborated and the severity of usability problems was rated.

4. Repeated usability inspection of the AXMEDIS Tools version 2.5 and comparison with the recordings of design deficiencies, usability problems and defects found previously.

6.3 How usable are the AXMEDIS Tools version 2.5? - Deficiencies and usability problems in detail

The following sections describe in more detail the findings for each of the AXMEDIS Tools version 2.5:

- AXMEDIS DRM Editor
- AXMEDIS DRM Viewer
- AXMEDIS Editor
- AXMEDIS Player
- AXMEDIS Player on.net
- AXMEDIS Cont Proc Rule Editor
- AXMEDIS Cont Proc Rule Scheduler
- AXMEDIS Prog and Pub Editor
- AXMEDIS Prog and Pub Engine
- AXMEDIS Prog and Pub Monitor
- Workflow tools

6.3.1 Installation of AXMEDIS Tools and Workflow Manager

The installation of the AXMEDIS Tools up to v2.2 had to be performed separately for each tool. This was time-consuming, however, the installation guidelines and readme files provided sufficient help for most of the tools. Beginning with version 2.3 the available AXMEDIS Tools install shield speeded up the installation considerably.

The installation of the Workflow Manager is very complex, very technical, very time consuming. It cannot be expected that a non technical user will be able to carry out the installation. Although the workflow tutorial contains many good details about the installation of Zope, Python and Openflow, a technical expert is needed to do the installation.

Most of the tools need to be configurated ("Configuration" window, "Preferences") is difficult. The available user guidelines give some hints how to do the configuration, but not all modules that can be configurated are described well.

6.3.2 AXMEDIS DRM Editor and Viewer

6.3.2.1 Visibility of system status

Redundant information. Several windows display a lot of redundant information. For example, in the "Grant Group" window, "Grant Group" occurs in the border of the window and as a title in the upper middle of the window. The names of most data entry field labels (Principal , Resource, Right, Interval, Fee , Territory, Number) occur twice (figure 1), in the border which surrounds each data element and as a field label. *Avoid redundancy to increase visibility.*

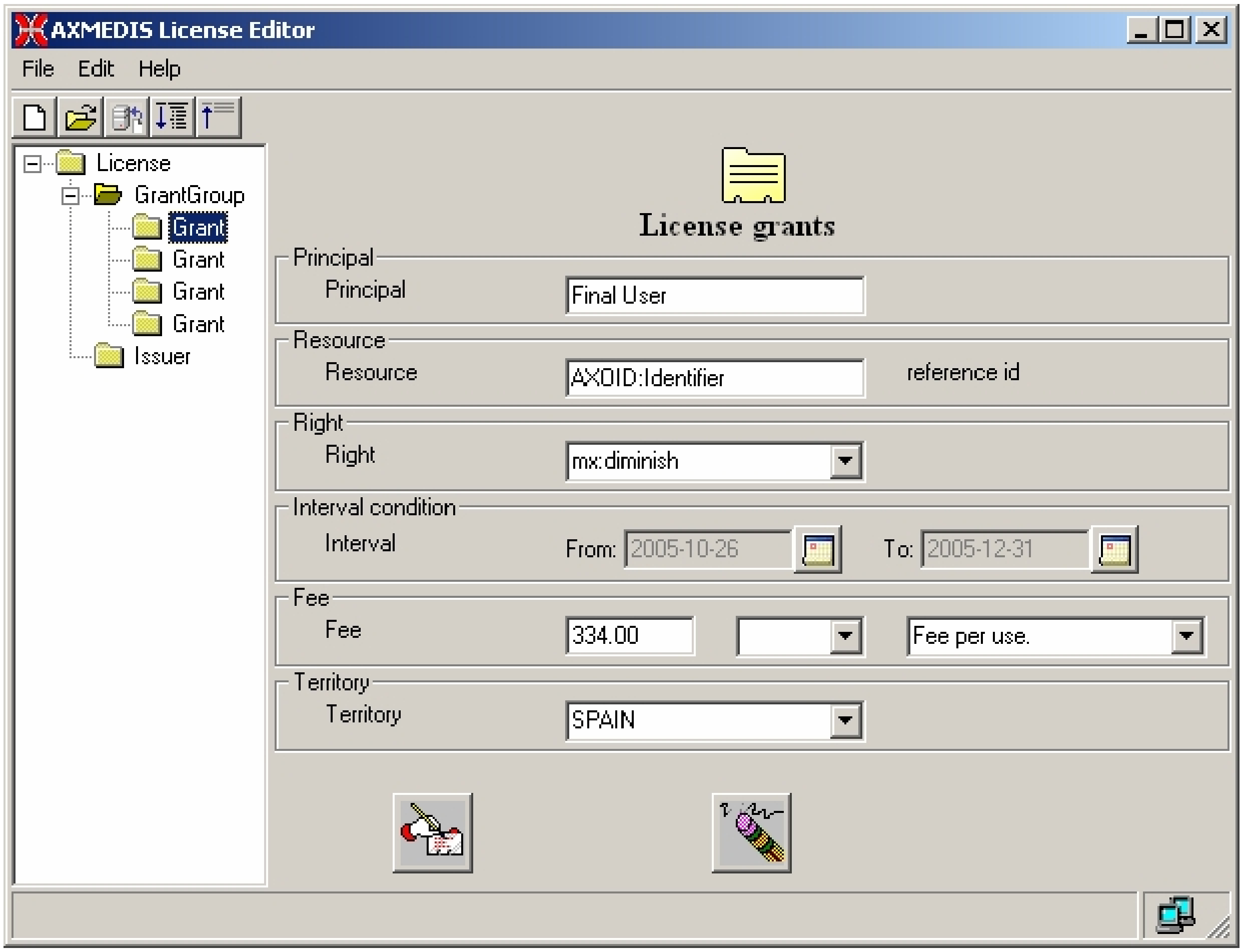
Some commands and icons cannot be applied in specific system states. E.g. "save to file" and "save to server" can be selected when no file is open. *Grey out commands and icons in system states where they cannot be applied to increase visibility for the user and to help the user recognize commands which can be applied in a specific state.*

The relationship between icons and commands is not visible. The menus in the top menu bar list commands but do not show the relationship to icons. *Visibility will be increased when the menus show the related icon in front of a command* (see example from Microsoft Word in figure 2).

Visibility of calendar icon. The calendar icon (in figure 1 in the "Interval condition" part) is neither well visible nor is the name of the icon displayed with mouse-over.

Visibility of license elements. Several grants in one license appear with the same name "grant" in the license tree (figure 1). *Numbering the grants will increase visibility. A user may find the opportunity to name the grants even better.*

Misleading spelling. The "License Grants" window shows only **one "License Grant"** at a time (figure 1). *Better change the name of the window to "License Grant".*



License grants

Principal
Principal: Final User

Resource
Resource: AXOID:Identifier reference id

Right
Right: mx:diminish

Interval condition
Interval From: 2005-10-26 To: 2005-12-31

Fee
Fee: 334.00 Fee per use.

Territory
Territory: SPAIN

Figure 1: AXMEDIS DRM Editor, “License grants” window

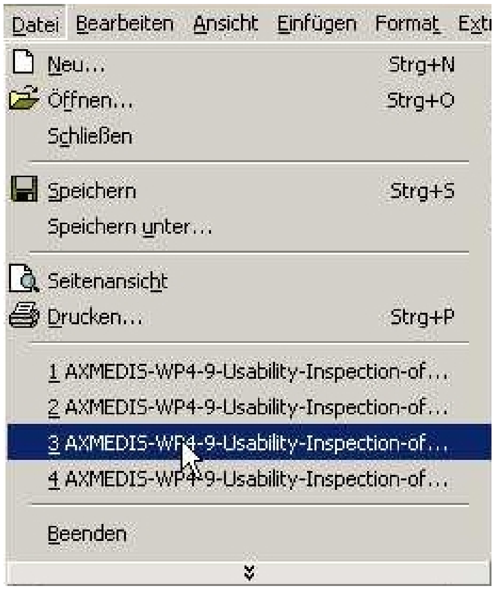
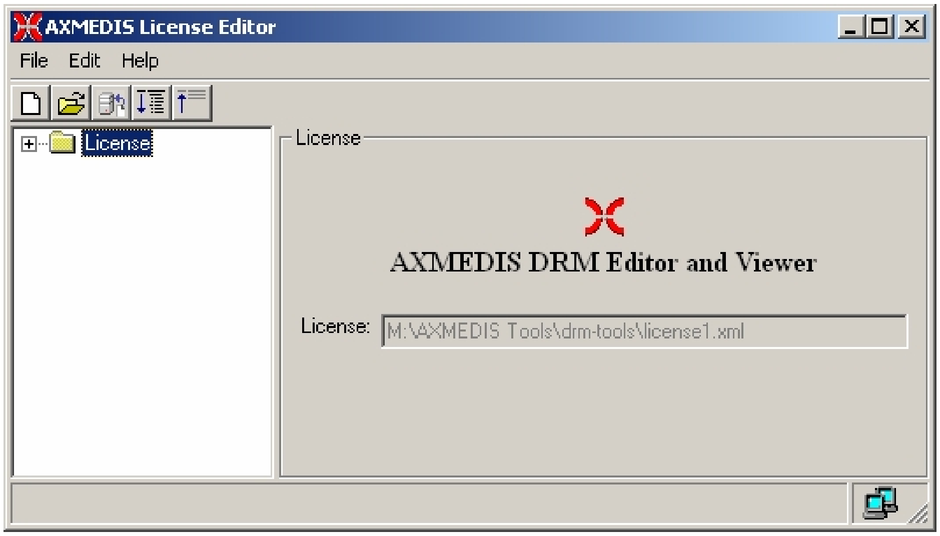

Datei
Bearbeiten
Ansicht
Einfügen
Format
Extr
Neu… Strg+N
Öffnen… Strg+O
Schließen
Speichern Strg+S
Speichern unter…
Seitenansicht
Drucken… Strg+P
1 AXMEDIS-WP4-9-Usability-Inspection-of…
2 AXMEDIS-WP4-9-Usability-Inspection-of…
3 AXMEDIS-WP4-9-Usability-Inspection-of…
4 AXMEDIS-WP4-9-Usability-Inspection-of…
Beenden

Figure 2: “File” menu in Microsoft Word

Visibility of the progress of task completion. During the creation of a new license it is not visible how much of the work has been done and what else needs to be done because only one element in the licence tree is displayed at a time. *It may be useful to add another level to the license tree which shows the elements that have been defined so far (a sit was implemented in version 1).*

Visibility of icon names. The names of the "ExpandAll" and "CollapseAll" icons will be more readable with a space between the words: "Expand All", "Collapse All".

Visibility of open file name. The name of an open file is only visible when the "license" window (figure 3) is displayed (see also section 6.3.2.4 about consistency). After "close" the "license" window is not refreshed and still shows the name of the previously closed file.



License: M:\AXMEDIS Tools\drm-tools\license1.xml

Figure 3: AXMEDIS License Editor

6.3.2.2 Match between the system and the real world

Terminology used in the license description and in the "Search" form is different: "Principal, Resource, Right, Interval, Fee, Territory, Number of times" vs. "AXOID, Person, Right, Territory, Usage, Date". *Use the same terminology in different locations.*

Multiple add of conditions may speed up the editing of a license.

Data entry forms. It is not obvious whether all fields in the data entry forms (e.g. "Search" form, "License Grant" form) must be completed, if the completion of a field is optional or obligatory. Field labels are not all familiar, brief, descriptive.

6.3.2.3 User Control and Freedom

Search. The "Search" command does not seem to work.
Version 1: Search for elements in a license did not work well because the search seemed to be case sensitive (input of terms with capital letters yielded a different result).

- Version 2.3 “Search“ sometimes works, sometimes no result was displayed although the search item was available in the license. Does “Search” only search the saved file and not take into account modifications which have not been saved?
- Version 2.5 “Search” does not seem to work correctly. Search in the example license1.xml for “condition: territory, spain” yields the result “2 grants found; Grant 1 : Right: Conditions: ; Grant 2 : Right: Conditions: “. Search for “condition: territory: SPAIN” displays the message “no matches found”.

Undo/Redo. There is no “Undo” and “Redo” functionality provided.

6.3.2.4 Consistency and Standards

Layout of window titles is inconsistent with the other tools. The titles of windows in the DRM Editor and Viewer are centred and a picture looking like an icon is provided together with the title. This is different compared with all the other AXMEDIS Tools. *Consistency across tools will increase visibility for the user.*

Inconsistent terminology. Different names for the tool can be found in different locations (see table 1, p. 50). *Be consistent and use the same name for the tool in all different places.* The best solution seems to be the one for the AXMEDIS Editor.

Inconsistent terminology. The command “Edit → Search” opens a window with the title “Check DRM conditions”. *The title of the window should be consistent with the command “Search”.*

Inconsistent style of some new icons. The icons “Creates a new grant”, “Deletes this grant”, “Add a new condition to the grant”, and “Deletes a condition of the grant” are not consistent with the icons used for create, delete in the other AXMEDIS Tools.

Inconsistent style of command names. The command names “Creates a new grant”, “Deletes this grant”, “Add a new condition to the grant”, “Deletes a condition of the grant” (figure 4) are too long. *A better solution may be “Create new grant”, “Delete grant”, “Add new condition”, “Delete condition”*. The size of all these icons is not the same.

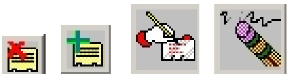


Figure 4: Command names “Creates a new grant”, “Deletes this grant”, “Add a new condition to the grant”, “Deletes a condition of the grant” (from left to right)

Inconsistent “delete” procedure. The procedure to delete a grant is not consistent with the procedure for deletion of a condition of a grant. While a grant is deleted with click to the delete icon which is visible in front of the grant, deletion of a condition is done by click to the delete icon first and then the condition to be deleted is selected from a menu. *Make the “delete” procedure consistent within the DRM Editor and with the other AXMEDIS Tools.*

Inconsistency of calendar icon. The calendar icon (figure 1 in the “Interval condition” part) is not consistent with the calendar icon used in the other AXMEDIS tools.

Inconsistent location for the file name. The AXMEDIS DRM Editor and Viewer show the name of an open file in the license window (figure 3). To be consistent with other AXMEDIS Tools and with standard software the name of an open file should appear in the title bar of the tool, e.g. “license1.xml - AXMEDIS DRM Editor”.

Table 1: Inconsistent use of tool name in different locations - AXMEDIS Tools version 2.5

AXMEDIS Tools menu	**AXMEDIS DRM Editor**
Message during start	AXMEDIS
Title bar	AXMEDIS License Editor
License Window	AXMEDIS DRM Editor and Viewer
Help –> About AXMEDIS License Editor …	Title bar: AXMEDIS i-icon DRM License Editor. v2.1.9
AXMEDIS Tools menu	**AXMEDIS DRM Viewer**
Message during start	AXMEDIS
Title bar	AXMEDIS License Viewer
License Window	AXMEDIS DRM Editor and Viewer
Help –> About AXMEDIS License Viewer …	Title bar: AXMEDIS i-icon DRM License Viewer. v2.1.9
AXMEDIS Tools menu	**AXMEDIS Editor**
Message during start	AXMEDIS Editor
Title bar	AXMEDIS Editor
Help –> About … SHIFT+F1	Title bar: About i-icon AXMEDIS Editor version: 0.4 date: Jun 22 2006 http://www.axmedis.org
AXMEDIS Tools menu	**AXMEDIS Player**
Message during start	AXMEDIS Player
Title bar	AXMEDIS Player
	No “Help → About” command
AXMEDIS Tools menu	**AXMEDIS Cont Proc Rule Editor**
Message during start	AXCP Rule Editor 1.0
Title bar	AXMEDIS Rule Editor 1.0
Help –> About … SHIFT+F1	Title bar: Help
AXMEDIS Tools menu	**AXMEDIS Cont Proc Rule Scheduler**
Message during start	AXMEDIS
Title bar	AXMEDIS Rule Scheduler
? –> About F2	Title bar: Error Not implemented yet.
AXMEDIS Tools menu	**AXMEDIS Prog and Pub Editor**
Message during start	AXMEDIS Programme Editor v. 1.0
Title bar	AXMEDIS Programme and Publication Editor 1.0
Help –> About … SHIFT+F1	Title bar: About Programmes and Publication

	i-icon This is about AXMEDIS Programme and Publication Editor http://www.axmedis.org
AXMEDIS Tools menu	**AXMEDIS Prog and Pub Engine**
Title bar	AXMEDIS Prog and Pub Engine
AXMEDIS Tools menu	**AXMEDIS Prog and Pub Monitor**
Message during start	AXMEDIS AxPnP Engine v.1.0
Title bar	AXMEDIS Programme and Publication Engine Monitor v.1.0
Help –> About …	Title bar: P&P Engine Monitor i-icon AXMEDIS Programme & Publication Engine Monitor v.1.0 http://www.axmedis.org

6.3.2.5 Help Users recognize, diagnose, and recover from errors

Wrong entries in data entry forms. The AXMEDIS DRM Editor does not detect wrong entries in fields of the “search/check DRM conditions” form. For example, numbers can be entered in the “territory” field, letters can be entered for “number of times” or “data” field, etc.

6.3.2.6 Error Prevention

“Save” command is missing. The user can only choose between “Save to file” or “Save to server” which opens a dialog window and asks for the file name. Selection of an existing file name will overwrite the existing file without asking the user to confirm overwrite.

6.3.2.7 Recognition rather than Recall

Data entry forms. Optional data fields are not marked.

6.3.2.8 Flexibility

Shortcuts are not provided. Conditions are added one by one. A multiple “add condition” function may be useful.

6.3.2.9 Aesthetic and Minimalist Design

Icon design. The design of some new icons is neither visually nor conceptually distinct (figure 1). In addition some pictures on the screen look very similar to icons, but do not represent a command.

6.3.2.10 Help and Documentation

User guidelines are available and very helpful.
On-line help. A “Help” menu is available but “Help” text is not implemented yet.
On-line instructions which guide the user are missing (e.g. completion instructions for data entry forms).

6.3.3 AXMEDIS Editor

6.3.3.1 Visibility of system status

The AXMEDIS Tools contain several editors. Because the AXMEDIS Editor is a tool used to create and edit AXMEDIS objects a better name would be to call it AXMEDIS Object Editor.

Window titles are not always precise. The AXMEDIS Editor consists of many windows with short titles. Longer titles may help understanding, eg.
AXMEDIS → AXMEDIS Object Hierarchy View
MEPG21 → MPEG21 Object Hierarchy View
View Metadata → Metadata Editor or Metadata Viewer or Metadata Editor / Viewer, etc.
The title of some windows does not make clear if the window contains an editor or a viewer or both.

“Protection View” window. “Move Up”, “Move Down” in the “Protection View” window looks corrupted.

Some icons are not meaningful. The icons for “add embedded resource”, “add object reference” and “add embedded object” (figure 5) are meaningless and the use of red and blue colour to distinguish two of the icons is not helpful. Avoid the design of icons for abstract concepts.

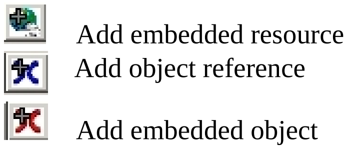


Figure 5: Meaningless icons

The icons “Go back”, “Go forward” and “Go to” are not well designed (figure 6). Neither the green colour nor the square or circle which surrounds the arrow do help to distinguish the different functions of the icons. The darker colour of the “Go to” icon gives the impression that this icon has been selected. Just the arrow would be sufficient to indicate “Go back” and “Go forward”.

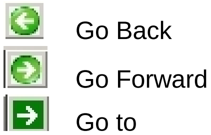


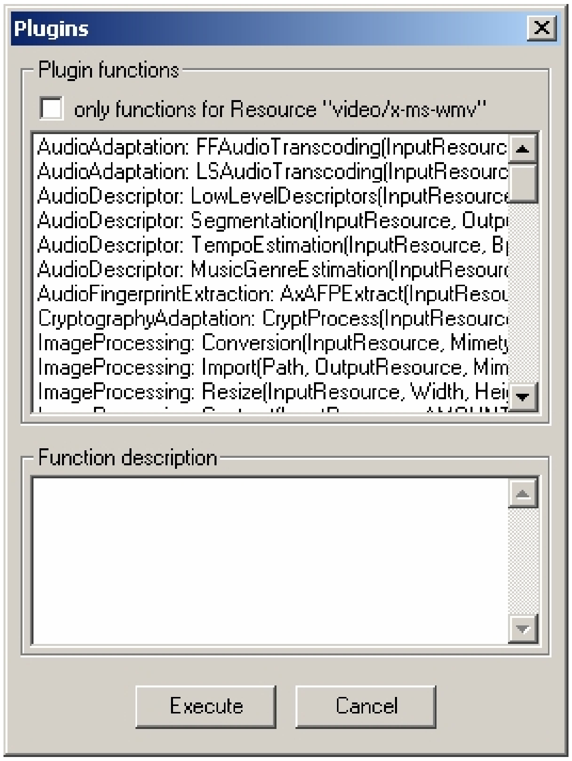
Figure 6: Inappropriate design of icons

The relationship between icons and commands in menus is not visible. The menus in the top menu bar list commands together with their shortcuts but do not show the icons for the commands. Visibility will be increased when the menus also show the icons for the commands (see example in figure 2).

It is not visible that some commands and icons cannot be applied in a specific state of the system. (e.g. when no file is open). When commands and icons are always visible it is up to the user to recall the commands which can be applied in a specific state. Exploration of commands which cannot be applied in a specific state often caused a crash of the tool. *Grey out commands and icons in system states where they cannot be applied.*

Icons not labelled. While the icons located in the top menu bar show labels with mouse over the icons in the "Resource View" window and other "View" windows do not have labels.

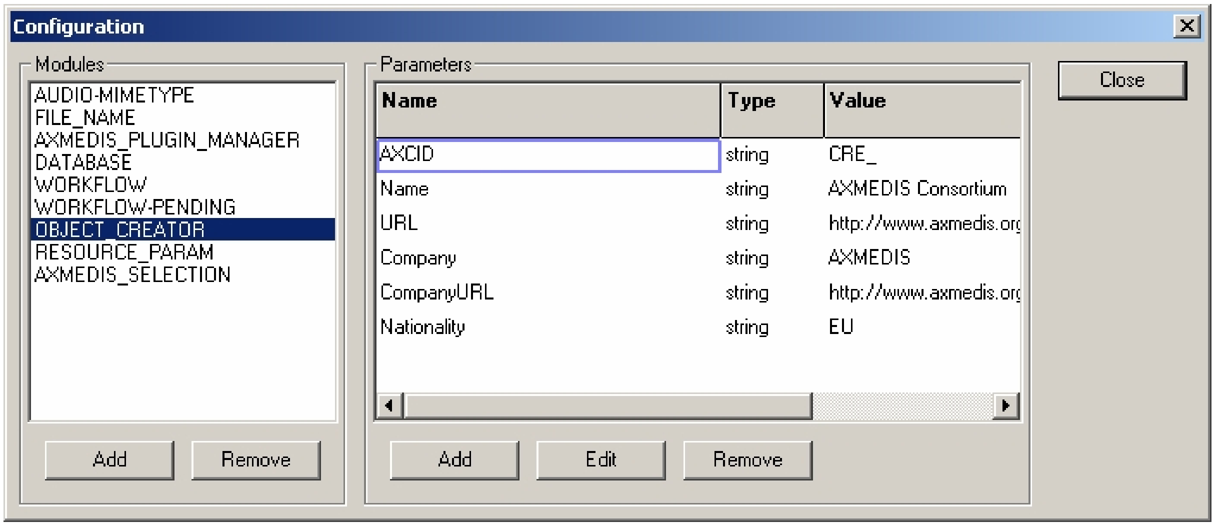
Width of windows. The width of several windows, e.g. "Plugins" (figure 7), "Configuration" (figure 8), is not wide enough when these windows are opened. The user has to increase the width of the window to be able to read the items which can be selected. *Increase window width such that the text in the window is readable when the window is opened.*



□ only functions for Resource "video/x-ms-wmv"

AudioAdaptation: FFAudioTranscoding(InputResourc
AudioAdaptation: LSAudioTranscoding(InputResourc
AudioDescriptor: LowLevelDescriptors(InputResource
AudioDescriptor: Segmentation(InputResource, Outp
AudioDescriptor: TempoEstimation(InputResource, B
AudioDescriptor: MusicGenreEstimation(InputResour
AudioFingerprintExtraction: AxAFPExtract(InputResou
CryptographyAdaptation: CryptProcess(InputResourc
ImageProcessing: Conversion(InputResource, Mimety
ImageProcessing: Import(Path, OutputResource, Mim
ImageProcessing: Resize(InputResource, Width, Hei

Figure 7: Insufficient width of "Plugins" window



AUDIO-MIMETYPE
FILE_NAME
AXMEDIS_PLUGIN_MANAGER
DATABASE
WORKFLOW
WORKFLOW-PENDING
OBJECT_CREATOR
RESOURCE_PARAM
AXMEDIS_SELECTION

Name	Type	Value
AXCID	string	CRE_
Name	string	AXMEDIS Consortium
URL	string	http://www.axmedis.org
Company	string	AXMEDIS
CompanyURL	string	http://www.axmedis.org
Nationality	string	EU

Figure 8: Insufficient width of “Configuration” window

Processing is not visible. Loading of resources and objects can take time (5-10 seconds and more), but this is not visible. *A delay indicator icon showing that the system is processing is needed.*

Progress of a task is not visible. When the user creates or edits an object it is not visible how much of the work has been done and what else needs to be done. *Context maps or procedural maps showing the process of work would be helpful.*

6.3.3.2 Match between the system and the real world

Terminology. The names of many commands and icons (e.g. icons “add embedded resource”, “add object reference” and “add embedded object” and commands in context menus) AXMEDIS specific, technical and unfamiliar to the users. *On-line help and instructions would be helpful here.*

Editors/Viewers. Selection of “Editors/Viewers” opens a “Viewer” window.

“View” menu title does not communicate. The commands in the “View” menu change depending on the context, i.e. on the open “View” window. This menu contains editing, search and view commands which can be applied to elements in the open “View” window, but that is not immediately visible to the user. *Is it possible to move the “View” menu into the open “View” window?*

Context menus are not visible at all. If users do not read the user guide or do not learn about context menus during training, they may never find out that context menus exist. In addition, recognition of context menus is easier than recall.

6.3.3.3 User Control and Freedom

Undo/Redo commands are available in the edit menu of the top menu bar but not implemented yet.

It would be helpful if users could set and save their own session properties for open viewers.

6.3.3.4 Consistency and Standards

Inconsistent use of tool name in different locations (see Table 1, page 49). Make the name of the tool used in different places consistent for all AXMEDIS Tools. Within the AXMEDIS Editor tool the name of the tool is already used consistently.

Inconsistent shortcuts used to exit the tool (table 2). A shortcut for exiting a tool is not needed, because this command is not used so frequently.

Table 2: Inconsistent use of the command for leaving a tool

Tool	Command name	Shortcut
AXMEDIS Cont Proc Editor	EXIT	Alt - X
AXMEDIS DRM Editor	EXIT	-
AXMEDIS DRM Viewer	EXIT	-
AXMEDIS Editor	EXIT	Ctrl + E
AXMEDIS Player	CLOSE	-
AXMEDIS Prog and Pub Editor	EXIT	Ctrl + X
AXMEDIS Prog and Pub Monitor	QUIT	-
AXMEDIS Cont Proc Rule Scheduler	EXIT	Alt - X

Inconsistent style of icons. The icon design is not always consistent with other AXMEDIS Tools (figure 9). For example, three different styles are used for “New” (AXMEDIS Cont Proc Editor, v1; AXMEDIS DRM Editor and Viewer and AXMEDIS Editor, v2; AXMEDIS Prog and Pub Editor, v3). Four different styles are used for “Save” (AXMEDIS Cont Proc Editor, v1; AXMEDIS DRM Editor and Viewer, v2; AXMEDIS Editor, v3; AXMEDIS Prog and Pub Editor, v4). In the AXMEDIS Prog and Pub Editor the name of the icon is always visible below the icon while in all the other tools the icon name is displayed with mouse over. “Open from Database” and “Upload into Database” icons as well as Cut”, “Copy” and “Paste” icons look different in AXMEDIS Cont Proc Editor and in AXMEDIS Editor.

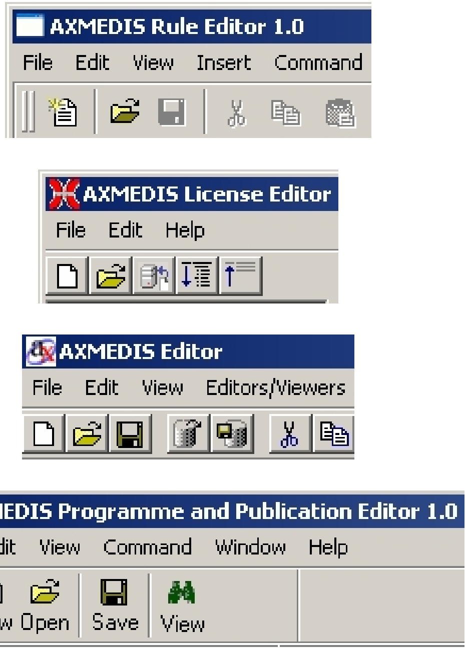

AXMEDIS Rule Editor 1.0
File Edit View Insert Command
AXMEDIS License Editor
File Edit Help
AXMEDIS Editor
File Edit View Editors/Viewers
AXMEDIS Programme and Publication Editor 1.0
File Edit View Command Window Help
New Open Save View

Figure 9: Different style of “New” and “Save” icon in different editors

Inconsistent command names for commands in menus and for command displayed for icons
The “Players” menu contains the command “start”. The related icon is named: “play”. *This could be made consistent by using the same name (either start or play) in both places and assuring that the same name is also used by the other AXMEDIS Tools.*

Menu titles do not communicate. The “Players” menu contains the commands “Start”, Stop”, “Pause”, “Continue”. However, a first time user may expect to find audio and video players in the “Players” menu. These players, however, are accessible via “View → Player Viewers”. Again, the title “View” menu is more general compared to a menu title “Players”. In addition, the “View” menu contains an “Edit” command which a user would try to find in a menu with the title “Edit”. *Improve menu titles.*

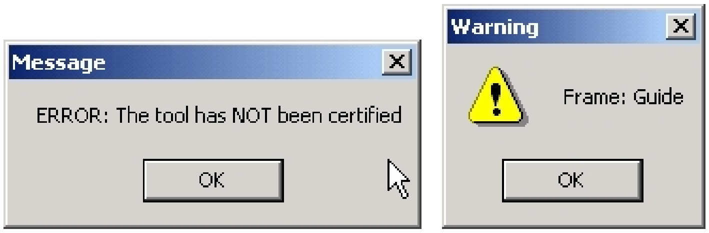
Inconsistent output in the “Help → About” menu (table 1, page 49). *Make the “Help” output consistent for all tools*. The solution for the AXMEDIS Editor is good, however, should be adopted by the other AXEMDIS Tools.

6.3.3.5 Help Users recognize, diagnose, and recover from errors

Help commands are available in the “Help” menu of the top menus bar, but not implemented yet.

Wrong entries in data entry forms. The AXMEDIS Editor does not detect wrong entries in the fields of data entry forms. *It will be advantageous for users, if the editor can detect and highlight errors in data entry fields and explain how to recover from errors.*

(Error) Messages. Click to “Help → Tool certification” opens the window presented in figure 10. A user may expect to get help about “Tool certification” here. Click to “Guide to AXMEDIS Editor” opens a “Warning” window: “Frame: Guide”. Click to “About AXMEDIS Editor” opens an “About” window: “AXMEDIS Editor, version 0.4, date: Jun 22 2006, http://www.axmedis.org”.



ERROR: The tool has NOT been certified

Frame: Guide

Figure 10: Message windows in the AXMEDIS Editor

Be friendly to the user and provide as title of the dialog window the command which the user has selected, for the above examples “Tool certification”, “Guide to AXMEDIS Editor”, or “About AXMEDIS Editor”. In the window distinguish between “errors” (system errors), “warnings” (minor system problems), and “messages”. For the examples described above, a more appropriate text would be “Message: The tool has not been certified. …. Read help/instructions about Tool Certification …..”, “Message: Guide to AXMEDIS Editor is not implemented yet.”, “AXMEDIS Editor, version 0.4, date: Jun 22 2006, http://www.axmedis.org”.

6.3.3.6 Error Prevention

When all commands and icons can be selected in all system states there is a risk that the selection of a command which cannot be applied to a specific state may cause a crash of the editor. *To prevent errors grey out inactive commands.*

6.3.3.7 Recognition rather than Recall

Context menus are well "hidden" to the user and without guidance and explanation it may be difficult for users to detect these menus.

Data entry forms. Optional data fields are not marked. There is no visual distinction between "choose one" or "choose many" in the data entry forms.

Relationships between the "Start", "Stop", "Pause", "Go back", "Go forward", and "Go to" commands and these icons in the top menu bar and similar icons in the "Resource View" window are not apparent to the user. It may be sufficient to provide these commands/icons only in the "Resource View" window and other windows where they are needed.

6.3.3.8 Flexibility

Shortcuts. Many keyboard shortcuts and function keys are provided. It is questionable whether all these shortcuts are needed. Compare with Microsoft Office software for example, where shortcuts are mainly offered in the "Edit" menu and for some commands in the "File" menu. *Shortcuts should be provided for frequently used commands only.*

Shortcut assignments are not consistent across AXMEDS tools (appendix 3).

6.3.3.9 Aesthetic and Minimalist Design

Icon Design. The design of the icons for "add embedded resource", "add object reference" and "add embedded object" (figure 5) is neither visually nor conceptually distinct.

Data entry forms. Field labels are not always familiar, brief, and descriptive.

Use of color. The SMIL Editor uses spectrally extreme colour s(turquoise) which do not look very appealing. A *softer colour may provide a better look and feel.*

6.3.3.10 Help and Documentation

User guidelines are available and very helpful.

On-line help and instructions which guide the user (e.g. how to complete data entry forms) would be helpful. A "Help" menu is available but not implemented yet.

6.3.4 AXMEDIS players

6.3.4.1 Visibility of system status

It is unclear which element in the object tree is displayed in the other open window. It is difficult to get an overview of the contents of a complex AXMEDIS object because elements are displayed one at a time, and there is no indication which element is displayed at a certain time. *The element which is selected in the object tree should always be the one which is displayed in the other open window.*

It is not visible that some commands and icons cannot be applied in specific system states (especially when no file is open). *Grey out commands and icons in system states where they cannot be applied.*

The commands “Play”, “Pause”, “Stop” in the “Play” menu are needed for playing audio or video files. *It may be advantageous to have these commands only in the window where either audio or video files are played.*

Menus titles do not communicate. The commands “Next Resource”, “Previous Resource”, and “Sliding Show” are located in the “File” menu. *The “View” menu is a better location for these commands.*

“Start Sliding Show” / “Stop Sliding Show”. *A better and shorter command name is “Start/Stop Slideshow*”. A change of the colour of this icon (stronger colour) after start of a slideshow would make the stop function more visible.

A slideshow can either be started by selecting “Sliding show” in the file menu or by click to the respective icon. The command “Sliding show” in the “File” menu is marked when the command is selected in the menu and the mark is removed when it is selected in the menu again. However, the mark in the menu does not disappear when the slide show is stopped by click to the respective icon and vice versa.

Missing label for “Resource index”. There is no indication what the figures between the icons “Previous resource” and “Next resource” in the top menu stand for. Make the label “Resource index“ visible with mouse over (figure 11).

The resource is only updated after click to either “Previous resource” or “Next resource”, but not when a slideshow is started, and not when the user clicks to an element in the AXMEDIS object tree. This may be confusing for users.

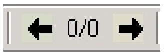


Figure 11: Resource Index

It would be desirable if the marked element in the AXMEDIS object tree changes at the same time as the displayed element in the other window changes (during a slide show, or when using next / previous resource).

Icons without labels. Icons in the “View” window do not have labels.

Delays caused by processing. A *delay indicator icon is needed showing that some processing is going on.* It takes at least 2 seconds to start a slideshow. An inpatient user will click to the slideshow icon again and by doing so stop the slideshow... Delays also occur after click to “Zoom in / Zoom out”.

6.3.4.2 Match between the system and the real world

Menus in the top menu bar contain commands which can be applied to specific windows or to selected elements in one of the windows only. A user may find commands more easily when the menu with the commands is provided in the respective window instead of in the top menu bar. An alternative solution would be to leave the menu in the top menu bar, but to make visible in the menu that the commands can only be applied in a specific window, eg.

6.3.4.3 Consistency and Standards

Inconsistent use of tool name in different locations (table 1, page 49). *Make the name of the tool used in different places consistent for all tools.* The best solution seems to be the one for the AXMEDIS Editor.

Inconsistent command used to exit the tool (table 2, page 55). *Replace "Close" command by "Exit" command".*

Icons are not meaningful. The icons for "Show AXMEDIS structure", "Sliding Show", and some other icons (figure 12), e.g. "Search" and "Home", are neither meaningful nor clearly visible. An icon showing a tree structure would be more meaningful for the "Show AXMEDIS structure" command.

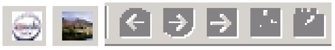


Figure 12: Icons for "Show AXMEDIS structure", "Sliding Show", "Search" and "Home"

Inconsistent icons. The icons "Next Resource" and "Previous Resource" seem to have a similar function as the icons "Go back", "Go forward" in the AXMEDIS Editor (figure 13). Maybe the same icon design can be used in both cases. The black arrows in the AXMEDIS Player look better than the green "Go back", "Go forward" icons.

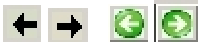


Figure 13: Comparison of "Next/previous resource" icons with "Go back/forward" icons

Inconsistent style of icons. The icons in the "View" window appear in a box, while icons in the top menu bar are not surrounded by a box.

Some icons have names which can be read with mouse over. "Go back", "Go forward", "Go to", "Search" and "Home" icons do not have names.

Either "Start" or "Pause" icon is visible in the "View" window. In other tools, both icons are visible all the time.

6.3.4.4 Flexibility

Shortcuts. It would be desirable to have shortcuts for "Next Resource" (space key) and "Previous Resource" (back key).

Shortcut assignments are not consistent across AXMEDS tools (appendix 3).

6.3.4.5 Aesthetic and Minimalist Design

The design of some icons is neither visually nor conceptually distinct (e.g. "Show AXMEDIS structure" (figure 14), "Sliding Show".

Figure 14: “Show AXMEDIS Structure” icon

The title of the “Play” menu does not communicate. It contains one commands with the same name as the title of the menu “Play”, and the commands “Stop”, “Pause”.

6.3.4.6 Help and Documentation

On-line help and instructions. There is no “Help” menu and no “Help → About” command (table 1, page 49). On-*line help and instructions which guide the user would be helpful.*

User guidelines are available and very helpful.

6.3.5 AXMEDIS Cont Proc Rule Editor

6.3.5.1 Visibility of system status

Some windows do not have titles. “Workspace”, “Output”, “Debug Monitor”, “JavaScript Editing window”, “Selection Editor” and other windows do not have titles. *Window titles add to visibility.*

Window titles appear in different locations. The titles of “Rule view” and “Library view” (figure 15) appear at the bottom of the “Workspace” window as well as “Call Stack”, “Local Variables”, “Watches”, “Breakpoints” in the “Output” window. In the “Selection Editor” (figure16) window titles of different views appear either at the top of the window (“AXMEDIS Query”, “Query Result”) or at the bottom of the window (“AxInfo/DCMI Query”, “PAR Query”). Visibility will be increased when all titles appear at the top of a window.

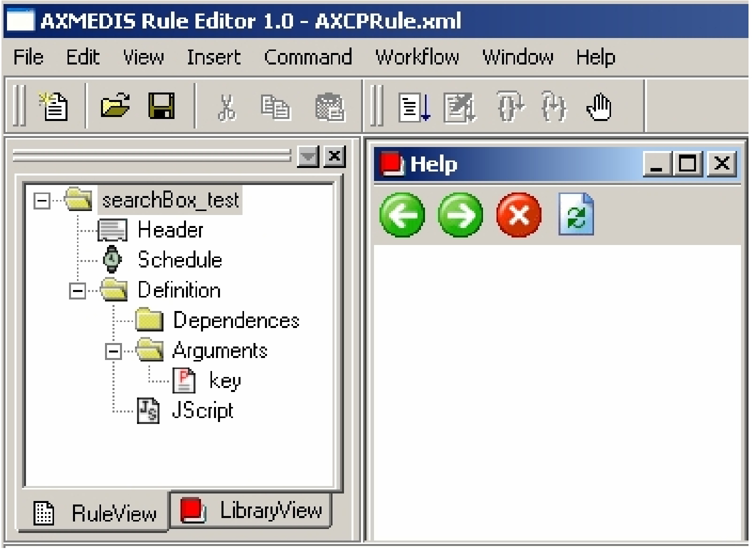
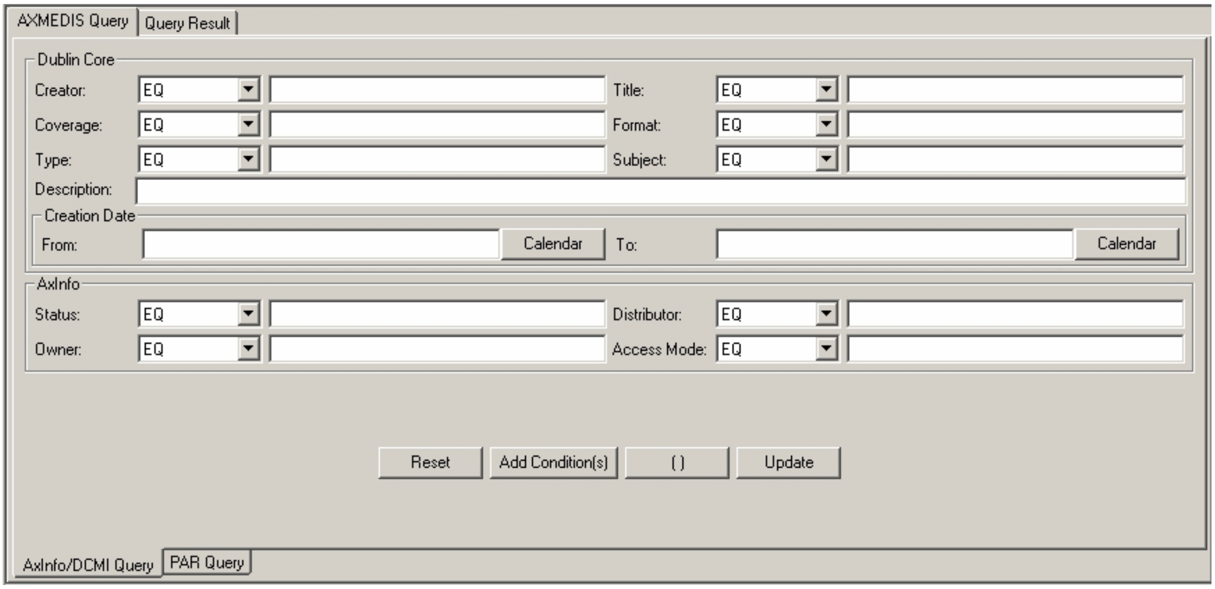


Figure 15: “RuleView”, “LibraryView”, and “Help” windows in the AXMEDIS Rule Editor 1.0



Dublin Core

Creator: EQ

Title: EQ

Coverage: EQ

Format: EQ

Type: EQ

Subject: EQ

Description:

Creation Date

From:

To:

AxInfo

Status: EQ

Distributor: EQ

Owner: EQ

Access Mode: EQ

Figure 16: “Selection Editor” window

Many windows can be open at the same time and not all of them may be visible. For example, double click to several JScript elements in the “Workspace” window one after the other opens one window on top of the others. It is not visible for the user whether the previously opened windows are still open or not. *A better solution would be to open windows in a cascading way.*

Some commands and icons cannot be used in specific system states (especially when no file is open). This is not visible. It is up to the user to know or find out, which commands can be applied in a specific state. *Grey out commands and icons in system states where they cannot be applied so that the user can recognize the commands which can be applied in a state and does not need to recall the commands.*

The “Selection Editor” menu in the top menu bar contains commands which can only be applied in the “Selection Editor” window. *Users may find these commands faster when the menu appears in the “Selection Editor” window instead in the top menu bar.*

Context menus are not visible at all and neither is the relationship of the commands in context menus with commands in the menus of the top menu bar. For example, the selection of “Command → Debug → Start” in the top menu bar seems to start the same operation as click to a JScript element and selection of “Debug” from the context menu. This is confusing. It would be clearer if the command in the context menu also had the name “Start debug”.

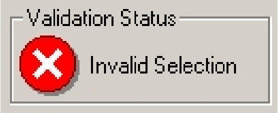
It is not evident how the elements of a new rule can be edited. *It may be helpful if all commands which can be selected from a context menu were also provided in a menu in the top menu bar.*

The relationship between icons and commands in menus is not visible. The menus list commands together with their shortcuts. Visibility will be increased if the menus also show the icons related to the commands. *Figure 2 illustrates an example of how such a menu will look like.*

Unnecessary use of icons and use of spectrally extreme colours for icons and buttons. “Rule View” and “Library view” each have an icon before the title and there is also an icon in the header of the “Help” window (Figure 15). These icons neither add to visibility nor to the meaning of the title. Better remove the icons.

The red colour of the icon before “Library View” and the icon in the header of the “Help” window is inappropriate. The use of green and red colours for the “Back”, “Forward”, “Close” icons in the “Help” window is not appropriate and neither is the green colour of the “Test Query” icon. Avoid the use of spectrally extreme colours except for visual cues to attract attention, alert the user or to stop the user doing something.

A good solution is the use of colour to show the “validation status”: red colour = invalid selection (figure 17), green colour = valid selection.



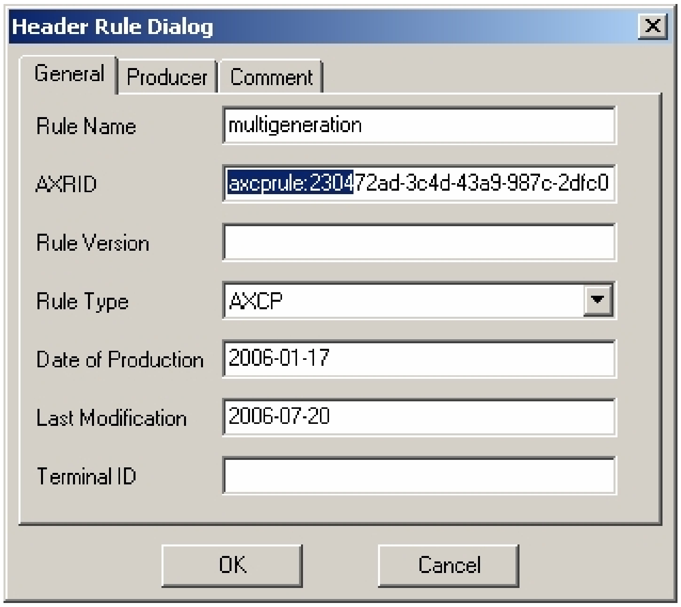
Validation Status

Invalid Selection

Figure 17: Display of Validation Status

The “Selection Editor” is not immediately visible. The command to open the editor “Insert → Selection” is not easy to find. *Visibility may be increased if the command to open the editor were “View → Selection Editor”.*

Missing visual feedback for read-only fields in data entry forms. Some elements in data entry and dialog windows are read-only. For example in the “Header Rule Dialog” window the AXRID element is read-only (figure 18). Grey out the input field with the AXRID element (to be consistent with read-only input fields in the other AXMEDIS tools).



Rule Name: multigeneration

AXRID: axcprule:23072ad-3c4d-43a9-987c-2dfc0

Rule Version:

Rule Type: AXCP

Date of Production: 2006-01-17

Last Modification: 2006-07-20

Terminal ID:

Figure 18: “Header Rule Dialog” window. The AXRID element is read-only.

Processing. Loading and saving of resources and objects can take time and this is not visible. *A delay indicator icon showing that the system is processing is needed.*

Progress of a task is not visible. When the user creates a new rule or edits a rule it is not visible how much of the work has been done and what else needs to be done. *Context maps or procedural maps which show the progress of work may be helpful.*

6.3.5.2 Match between the system and the real world

Some icons are not concrete, not meaningful, and not familiar to the user. “Test query” and “Test selection” icons are not meaningful. It is better to avoid the design of icons for abstract concepts such as “Test query” and “Test selection” (figure 19). *Use a button with the name instead.*

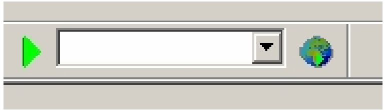


Figure 19: Icon to the left: “Test query” and icon to the right: “Test selection”

Menu names do not communicate. Command names in some of the menus in the top menu bar do not match their corresponding menu titles. E.g. the “Command” menu contains commands as do the other menus. A meaningful menu title is needed such as “Debug” or “Extras.

“Edit” vs. “Insert” menu. Is not insert an editing command? “Edit” vs. “Selection Editor”.

Data entry forms. Field labels in the data entry forms contain much AXMEDIS specific, technical language. *Context specific online help with definitions of terminology may be helpful.*

It is not obvious whether all fields must be completed in a data entry form, which fields are optional or obligatory to complete. Read-only field are not always visible.

“Tab forward” in data entry forms of the “Selection Editor” works, however, for a non technical user in an unexpected way (e.g. creator field, creator operator, title field, title operator, and so on).

Terminology. The tool contains a lot of technical AXMEDIS terms.

6.3.5.3 User Control and Freedom

“Undo” and “Redo” can be applied when editing JScripts. Multiple “undo/redo” is possible but it is unclear how long tracing back and forward is possible.

Arrangement of Windows. “Tile horizontally” and “Tile vertically” seems to work with up to three open windows only.

Set properties. It would be helpful if users could define their own session properties for windows to be open at the start of the tool and for the default arrangement of open windows (e.g. cascading style).

6.3.5.4 Consistency and Standards

Inconsistent use of tool name in different locations (table 1, page 49). Be consistent and use the same name for a tool in different locations. The best solution seems to be the one for the AXMEDIS Editor.

Inconsistent use of “Help → About” in different tools (table 1, page 49). Be consistent and use the same command name for “Help → About” in different tools. A shortcut for “Help → About” is not needed, because this command will not be used frequently. Other tools provide the shortcut F1 for on-line Help.

Inconsistent use of commands and shortcuts for different tools (table 2, page 54). The commands and shortcuts provided for leaving a tool are different for different tools. *Use EXIT as command name in all tools. A shortcut for leaving a tool is not needed, because this command is not used so frequently.*

Inconsistent style of icons. The style of an icon for the same command is not always consistent with other AXMEDIS tools (figure 9, page 55). For example, three different styles are used for the “New” icon (AXMEDIS Cont Proc Editor, v1 (= AXMEDIS Cont Proc Rule Editor 1.0); AXMEDIS DRM Editor and Viewer and AXMEDIS Editor, v2; AXMEDIS Prog and Pub Editor, v3). Four different styles are used for the “Save” icon (AXMEDIS Cont Proc Editor, v1; AXMEDIS DRM Editor and Viewer, v2; AXMEDIS Editor, v3; AXMEDIS Prog and Pub Editor, v4). In AXMEDIS Prog and Pub Editor the name of the icon is always visible below the icon while in all the other tools the icon name is displayed with mouse over.

“Open from Database” and “Upload into Database” look different in the AXMEDIS Cont Proc Editor and in the AXMEDIS Editor (table 3).

Table 3: Inconsistent use of icons

Icons	AXMEDIS ContProc Editor	AXMEDIS Editor
Open from Database		
Upload into Database		

“Back”, “Forward”, and “Close” icons in the “Help” window are larger compared to other icons and are coloured (figure 20). There are two different “Close” icons in the “Help” window (figure 20). The red coloured one is different compared with the other “Close” icon in the same window and in other windows of the AXMEDIS ContProc Editor. *Agree and use a consistent style of icons for all AXMEDIS tools.*

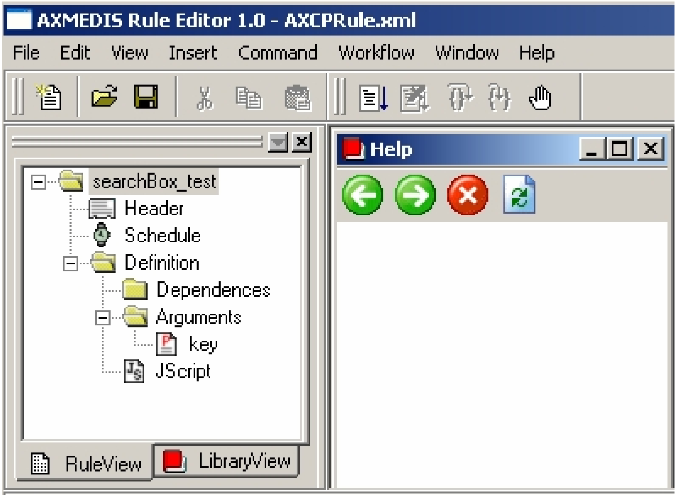


Figure 20: “RuleView”, “LibraryView”, and “Help” windows in the AXMEDIS Rule Editor 1.0

The style of icons for the “Selection Editor” (figure 21) is not always consistent with the other icons and the design of some of these icons is not very clear: “Open Selection” and “Save Selection” icons are consistent with “Open rule” and “Save Rule” icons. “Open from Database” is neither consistent with “Open Rule” nor with “Open Selection”. “Open from Database” and “Upload into Database” icons could be made simpler by using an arrow down / arrow up instead of blue plus sign and red arrow (avoid using colours). The same solution may be adequate for “Import Query” and “Export Query” icons.

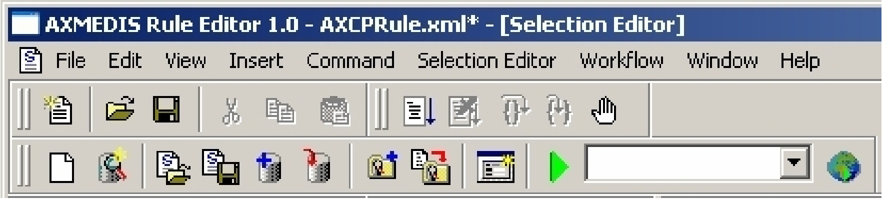
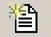
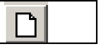

AXMEDIS Rule Editor 1.0 - AXCPRule.xml* - [Selection Editor]
File
Edit
View
Insert
Command
Selection Editor
Workflow
Window
Help

Figure 21: Icons in the AXMEDIS ContProc Editor 1.0

The “Clear Selection” icon is too similar to the “New Rule” icon and inconsistent with icons used in other AXMEDIS Tools (table 4).

Table 4: Inconsistent use of icons

“New Rule”	
“Clear Selection”	
“New” in AXMEDIS Editor and AXMEDIS Player	

The “Customize Query Panel” icon (figure 22) is not meaningful. It seems to be sufficient if this command appears in the menu of the “Selection Editor” only. An icon is not needed for “Customize Query Panel”.

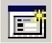


Figure 22: “Customize Query Panel” icon

The yellow star in the “Add new Query” icon is almost invisible and does not contribute to a better understanding of the meaning (table 4).

The correct name for the “Insert Breakpoint” icon is “Insert/Remove Breakpoint”.

A non technical user will not understand the meaning of the icon/button “( )” (nesting level) in the “Selection Editor” window.

Inconsistent location of icons. In both “Output” and “Debug Monitor” windows the “Close” icon is located in the top left corner while in the other windows the “Close” icon is in the top right corner. *Let the “Close” icon always appear in the top right corner.*

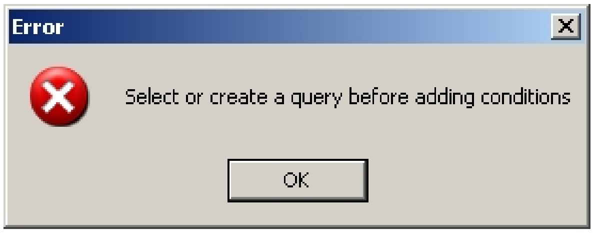
Inconsistent terminology. The selection of “View → Preferences” opens a “Configuration” window. *Be consistent and change the window title to “Configuration” which is more adequate and would be consistent with the solution in other tools.*

Other tools provide a “Configuration” command in the “File” menu. *Be consistent with other tools and move the command to the “File” menu.*

6.3.5.5 Help Users recognize, diagnose, and recover from errors

Wrong entries in data entry forms. The AXMEDIS Cont Proc Editor does not seem to detect wrong entries in data entry fields nor does it highlight or explain them for the user.

(Error) Messages. The response to a wrong selection of a command was (figure 23):



Select or create a query before adding conditions

Figure 23: Message

In cases like this, the term “error” may not be adequate and annoy the user. Inform the user in a more friendly way: “Message: Select or create a query before adding conditions”.

6.3.5.6 Error Prevention

After closing a file the “Selection Editor” menu was still visible in the top menu bar and commands from this menu could be selected. The commands from the “Windows” menu, and “Import JScript”, “Export JScript” from the “Files” menu were also selectable.

6.3.5.7 Recognition rather than Recall

“Selection Editor” is “well hidden”. The user must know that “Insert → Selection” opens the editor. A better solution may be to make the editor available with “View → Selection Editor”.

Data entry forms (e.g. "AxInfo/DCMI Query” and "PAR Query”) have two or more columns with data entry fields. The field label of the right column is very close to the data entry field of the left column. Increase the space between the two columns (figure xx).

Is there a reason why data entry fields are arranged in two or more columns? There seems to be sufficient space to arrange all fields one below each other.

6.3.5.8 Flexibility

Shortcuts. A large number of keyboard shortcuts are provided (appendix 3). It is questionable whether all shortcuts are needed. Shortcuts are used to enable experienced users to speed up their work. However, when there are so many shortcuts it may be difficult for a user to remember all of them. *Provide shortcuts for frequently used commands only.Compare with standard software, where shortcuts are mainly offered for commands in the “Edit” and “File” menus.*

Shortcut assignments are not consistent across AXMEDS tools (appendix 3). *Be consistent with other tools.*

6.3.5.9 Aesthetic and Minimalist Design

Icon design. Some new designed icons are neither visually nor conceptually distinct.

6.3.5.10 Help and Documentation

On-line help. A “Help” menu is available but not implemented yet.

On-line instructions which guide the user are missing (eg. completion instructions for data entry forms and for dialog boxes, simple examples for jscript creation).

Data entry forms. Field labels in data entry forms are not always familiar, brief, and descriptive. *On-line instructions which show how to complete a data entry form may be helpful.*

User guidelines are available and very helpful.

6.3.6 AXMEDIS Cont Proc Rule Scheduler

6.3.6.1 Visibility of system status

Menu title does not communicate. Commands of the “Command” menu do not match their corresponding menu title, because all menus contain commands. Use a more precise name, e.g. “Rule management”.

6.3.6.2 Match between the system and the real world

-

6.3.6.3 User Control and Freedom

-

6.3.6.4 Consistency and Standards

Inconsistent use of tool name in different locations (table 1, page 49). Make the name of the tool used in different places consistent for all tools. The best solution seems to be the one for the AXMEDIS Editor.

Inconsistent shortcut used to exit the tool (table 2, page 54). A shortcut for exiting a tool is not needed, because this command is not used so frequently.

Inconsistent terminology. The selection of “Settings → Preferences” opens a “Properties” window where “Scheduler Settings” and “Grid Settings” are defined. Be consistent with the window title “Configuration” in other tools.

6.3.6.5 Help Users recognize, diagnose, and recover from errors

-

6.3.6.6 Recognition rather than Recall

-

6.3.6.7 Flexibility

-

6.3.6.8 Aesthetic and Minimalist Design

-

6.3.6.9 Help and Documentation

On-line help. “Help” menu is available but help is not implemented yet.

User guidelines are available and very helpful.

6.3.7 AXMEDIS Prog and Pub Editor

6.3.7.1 Visibility of system status

Name of an open file is not visible. It is not immediately understandable and visible that several files can be opened and are shown in the tree hierarchy window. A new user may become aware of this fact only after the use of “New” and when opening the parameters window for a prog&pub file which shows the name in the header of the window. This could be made clearer by changing the root name of the tree hierarchy from “AXMEDIS PnP Programmes” to “AXMEDIS Prog&Pub Files”.

Windows do not have titles. The three windows do not have titles. Provide a title in the top left corner of each window, e.g. “Programme workspace”, “Programme window”, “Output”.

“Expand” / “Collapse” commands may be useful to quickly navigate through a large Prog&Pub tree hierarchy.

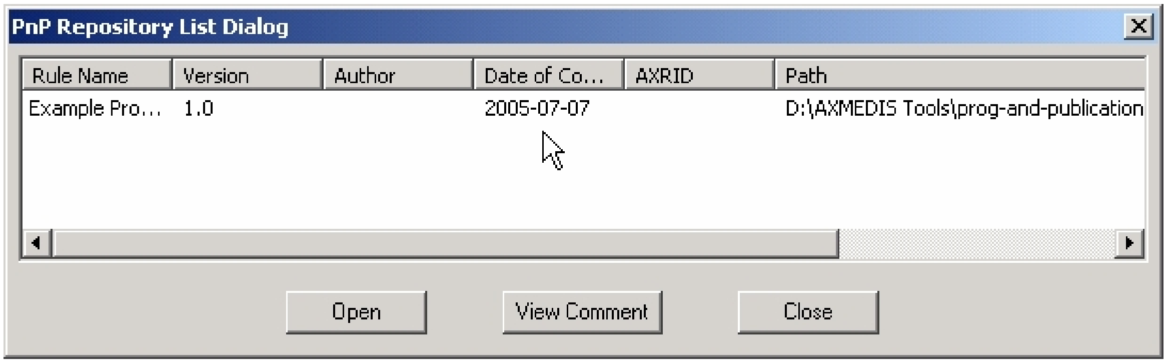
Missing visual feedback for read-only fields in data entry forms. Some elements in data entry and dialog windows are read-only. For example in the “Producer” window the Work item ID” element is read-only. Grey out the input field with the AXRID element (to be consistent with read-only input fields in the other AXMEDIS tools).

Inconsistent location of icons. The “Close” icon is on the left side of the “Output” window. *Move “close” icon to the top right of the window to make it consistent with all other tools.*

“About” command in the context menu is the same command as in the “Help” menu. *In the context menu the “About” command is not needed.*

Several “View” commands which have different functions. Using “View” in the context menu and double click to an element in the tree hierarchy opens a specific window with prog&pub parameters related to the selected element. When using the “View” icon the first of six pages of the parameters window is opened. This is a bit confusing for the user who only sees one “View” command / icon and who expects that the same results will appear. It may be more precise to rename the “View” commands in the context menu to “View header”, “View schedule” and “View distribution” or “View programme”.

Width of windows. The “PnP Repository List Dialog” window (figure 24) is not wide enough. The width of the columns in the dialog window can be changed but the width of the window itself cannot be adapted. A wider window showing the “PnP repository list dialog is desirable so that more content is immediately visible.



Rule Name	Version	Author	Date of Co...	AXRID	Path
Example Pro...	1.0		2005-07-07		D:\AXMEDIS Tools\prog-and-publication

Figure 24: PnP Repository List Dialog

Menu title does not communicate. Commands of the “Command” menu do not match their corresponding menu title, because all menus contain commands. *Use a more precise name, e.g. “Extras”.*

6.3.7.2 Match between the system and the real world

Inconsistent terminology. The element “Distribution” in the Programme and Publication tree is called “Programme” in the data entry form. Be consistent, either use distribution or programme in both places.

“Configuration” window. Non technical users will have difficulties to understand what needs to be inserted in the “Configuration” window. The “User Guide for the Programme and Publication Area” provides some information, but a naïve user may find it difficult to configure the AX-PnP-Engine. E.g. setting the port number. Where do you get the port number?

6.3.7.3 User Control and Freedom

Undo/Redo. Neither “Undo” nor “Redo” commands are available.

User properties. When the tool is started and a file is open there are always all views open. It would be helpful if users could set and store their own session properties.

6.3.7.4 Consistency and Standards

Inconsistent use of tool name in different locations (table 1, page 49). *Make the name of the tool used in different places consistent for all tools*. The best solution seems to be the one for the AXMEDIS Editor.

Inconsistent use of command for leaving the tool (table 2, page 54). *Change the command “Quit” to “Exit”.*

Inconsistent style of icons. The icon design is not always consistent with other AXMEDIS Tools. For example, three different icon designs are used for “New” (AXMEDIS Cont Proc Editor, v1; AXMEDIS DRM Editor and Viewer and AXMEDIS Editor, v2; AXMEDIS Prog and Pub Editor, v3). Four different icons design are used for “Save” (AXMEDIS Cont Proc Editor, v1; AXMEDIS DRM Editor and Viewer, v2;

AXMEDIS Editor, v3; AXMEDIS Prog and Pub Editor, v4). In AXMEDIS Prog and Pub Editor the name of the icon is always visible below the icon while in all the other tools the icon name is displayed with mouse over. "Open from Database" and "Upload into Database" icons as well as Cut", "Copy" and "Paste" icons look different in AXMEDIS Cont Proc Editor and in AXMEDIS Editor.

The label of icons appears below the icon and is always visible. This is not consistent with the solution in all the other tools, where the label appears with mouse-over.

Try to use a consistent style of icons for all AXMEDIS tools.

6.3.7.5 Help Users recognize, diagnose, and recover from errors

-

6.3.7.6 Recognition rather than Recall

-

6.3.7.7 Flexibility

Windows arrangement. A shortcut for "Window → Next" and "Window → Previous" may be desirable.

Shortcut assignments are not consistent across AXMEDS tools (see appendix).

6.3.7.8 Aesthetic and Minimalist Design

Field labels are not always familiar, brief, descriptive. On-line help may be needed.

Menu title does not communicate. All menus contain commands, not just the "Command" menu. Be more precise.

6.3.7.9 Help and Documentation

On-line help and instructions. "Help" menu is available but help is not implemented yet.

User guidelines are available and very helpful.

6.3.8 Workflow tools

Inspection and evaluation of the workflow tools was done on the basis of demonstrations during project meetings, tutorials and video demonstrations.

The Workflow Manager as it is at the moment is a version which can be used to demonstrate that AXMEDIS workflow management can be done. However, the interface is extremely technical and cannot be considered a user interface yet.

6.4 How usable are the AXMEDIS Tools Version 2.5? – Summary results

6.4.1 Barriers to a successful use of the AXMEDIS Tools

Potential barriers to people using the AXMEDIS Tools successfully are:

- People not understanding quickly how to install some of the AXMEDIS Tools (e.g. Workflow Tool)
- Awkwardness around configuration of some of the AXMEDIS Tools
- Due to complexity people may not understanding quickly how the AXMEDIS Tools work
- Some missing visibility makes it difficult to find commands, and needed information
- Some inconsistency of commands, icons, formats within and between the AXMEDIS Tools
- Some confusing terminology (e.g. EXIT, QUIT, CLOSE and different shortcuts for the EXIT command in different tools)
- Page layouts (windows not wide enough to read all text, not always supporting task flows, and lacking simple instructions in place)
- Instability (inevitable during development) and low level error messages (when a tool crashes)

6.4.2 Proposed changes

A number of straightforward changes – feasible within the timescale and design constraints – are proposed to improve the AXMEDIS Tools without major restructuring of the tools:

Visibility

- Add meaningful titles to open windows
- Improve the titles of some menus (e.g. “Command” menu)
- Make delays visible for loading or saving objects and resources, and for other processing which may take time. Provide a delay indicator, e.g. an icon showing that the system is processing.

Consistency

- Make the style of used icons consistent within an AXMEDIS Tool and across the AXMEDIS Tools
- Make data entry forms consistent within an AXMEDIS Tool and across the AXMEDIS Tools
- Reduce the number of shortcuts and make the shortcuts consistent across the AXMEDIS Tools

Terminology

- Clarify the names of some commands (e.g. EXIT)
- Make dialogue box naming consistent with menu items
- Use only one name for an AXMEDIS Tool in different locations
- Make icon naming consistent with the menu items

Layout

- Use red and green colour for icons only, if you want to alert a user.
- Avoid the use of extreme colour

Error Diagnosis

- Improve the wording of (error)messages (in many cases the term “message” or “warning” may be sufficient)

Help

- Provide sufficient on-line help for configuration
- Provide sufficient on-line help for the creation of JScripts including some simple examples

6.4.3 Implemented changes

DRM Editor/Viewer

Expand/Collapse. While in version 1 browsing through a large license tree became tedious, “ExpandAll” and “CollapseAll” icons were implemented in Version 2.3. This is a large improvement.

The consistency of the use of the tool name in different locations has been improved (compare table 1, page with table 2, appendix, page).

The overall stability of the tools has increased (less severe programme crashes).

6.5 Conclusions

The initial usability inspection found that the AXMEDIS Tools are mostly visually appealing. However, usability is affected by some inconsistencies, confusing layout, icon design, some awkward terminology, and a lack of simple instructions.

Most of the usability problems found are not severe problems because the tools could be used without removing these defects. However, this may increase learning effort for potential users, increase the probability of user errors, and by this reduce the effectiveness and efficiency of use.

The usability inspection identified some barriers to use the tool successfully and efficiently, and made specific recommendations for changes that can be implemented in the time available and will make a rapid and big difference in usability. Some of these changes can be made immediately.

Once the tools are in use, we can learn a lot more about their usability, how the Tools are used and what people think of it, and feed the findings rapidly into the next version. This is strongly recommended.

7 Informal Testing of the AXMEDIS Tools by Users: Work performed

The usability partners in the AXMEDIS project (ANSC, ILABS, SEJER, XIM and AFI) have installed and tried out the progressing versions of the AXMEDIS Tools and have continuously fed back comments to the developers in terms of problems they encountered, further requirements detected, their view of usability of the tools. The summary of these comments is provided in the following table.

Question	ANSC (August 2006)	ILABS (Aug-2006)	SEJER	XIM	AFI	ACIT (August 2006)
1) Did you install the AXMEDIS Editor ?	Yes	Yes	YES	Yes	Yes	Yes
2) How many installations	4 (one with the database) installer used v2.5 editor used v4.0 28 July updated by Piero and taken from the repository	Previously 1, currently 3 (2 including the database and 1 without) Tried V2.1 and looked for further versions after the V2.2 but could not find any. We tried the set-up provided in v2-2 and have to say (where it not for the known problems reported by DSI) this would be a good improvement Updated with exe from Bellini	2	2, including running on a Mac via MS Virtual PC, which so far has been successful.	2	1 installation of version 1 on PC (Windows XP) 2 installations each of version 2.2 / 2.3 / 2.4 / 2.5 on PC and Laptop (Windows XP) Files missing (v2.2): MSVCP71.DLL, MSVCR71.DLL missing. Problem solved with help of Kia Ng.
3) how many objects you have produced ?	About 10 simple samples. More interesting ones on musical instruments. 3 already posted in first version: http://www.axmedis.org/documenti/view_documenti.php?doc_id=2232 4 posted in second	A few elementary samples Plain and AXMEDIS objects have been circulated during debug phase and also used to verify if desired functionalities were properly working. Posted on portal 8 sample objects, no	1, in order to test the HTML Document Viewer	5 (mainly test objects so far, in order to test and learn the tools)	We have produced simple object and saved them into our database. However system often fails and objects can't be saved into the AXMEDIS database. All type of audio files can't be listened. Guide can't be opened and system fails while	A few simple objects using own content (audio, video, images, text) and several complex objects using the content of example objects.

	version (solved same AXOID problem but with workaround (ask Piero)) http://www.axmedis.org/documenti/view_documenti.php?doc_id=2243 Others coming soon. Working on Rule editor testing also to produce objects	feedback received from anyone.			trying to following all steps	
4) did you used the hierarchy editor and viewer ?	Yes	Yes	YES	yes	yes	Yes. See comments in the report: Usability Inspection of the AXMEDIS Tools V1.1
5) did you used the visual and behavioral editor and viewer ?	Yes, but it does not work for us (refers to v. 2.3 of the installer)	Yes but having a lot of problems: use is not intuitive, USERGUIDE-Visual-and-Behavior-Editor-and-Viewer.doc is not updated to editor version contained into zip file. We could save smil file and try to view the result of our insertion of scenes and temporization, then we would re-open visual view and adjust shapes, but we could	NO	yes	No it does not work . No idea on how it works.	In version 1 and 2.2. it did not work. Yes in version 2.3

		not manage to go back to that view. We would also like to re-open an already saved smil file but then we can't see visual scenes, while temporal schedule seems to be resetted. On closure editor always crashes. We never could manage to listen to audio. Encountered problems have been always reported and in some cases this turned out either in work-around suggestions or in upgrades in components.				
6) did you used the license editor and viewer ?	Yes.	Yes Encountered problems have been reported in the wiki. Interaction with developing partners turned out either in work-around suggestions or in upgrades in components.	YES but only to test it	yes	We only tried to test it.	Yes. See comments in the report: Usability Inspection of the AXMEDIS Tools V1.1
7) did you used the protection editor and viewer ?	Yes in a very basic way.	Yes the viewer / we tried setting something about protection but while	YES but only to test it	yes	No it does not work for audio files. In addition it is not clear how to use it.	Yes

		finalizing object development the editor crashed so that we are not sure of the actual result Encountered problems have been reported in the wiki. Interaction with developing partners turned out either in work-around suggestions or in upgrades in components.			Guide is not available.	
8) did you used the capability of the axmedis editor to call external procedure for processing content ?	Yes primarily on images.	Yes if if you mean content processing plugins, we used: - audio adaptation success for: - aif to mp3 unsuccess for: - wma to mp3: editor just closes itself without any message - mp3 to mp3: editor just closes itself without any message - wav to mp3: editor just closes itself without any message - image adaptation	NO, where is that?	not yet	Content processing? The mime type that results on the application is UNKNOWN because the mime type is audio/aif. Chancing it from properties to audio/x-aiff 6 plugins suddenly becomes available. But we did’nt succeed in conversion of any file (we tried all types of audio files)	No

		success: - monochrome - contrast - resize - edge - emboss - blur - mirror - rotate - paste				
9) did you installed the AXMEDIS content processing tools, AXCP ?	Yes (the rule editor)	Yes Encountered problems have been reported in the wiki. Interaction with developing partners turned out either in work-around suggestions or in upgrades in components.	YES	yes	Yes (the rule editor)	Yes.
10) did you used the AXCP editor ?	Yes	Yes, we used image processing functions: monochrome, paste, rotate Encountered problems have been reported in the wiki. Interaction with developing partners turned out either in work-around suggestions or in upgrades in components.	YES	yes	NO. Not capable to use it . Too complex as it need technical skill. If it remains as it is it would be a problem for users and content owners.	Yes. See comments in the report: Usability Inspection of the AXMEDIS Tools V1.1
11) did you created	Yes	Yes, we used image	YES	yes	No , It is not in our	Tried, although I

some java script ?		processing functions: monochrome, paste, rotate			capabilities and it is not in capabilities of all potential users providing music, video and other audiovisual content unless they have technological employee	have only little java script knowledge. I also modified the available java script examples and then tried to debug them.
Have you executed those scripts ?	Yes	Yes and we succeded. We found debug functionality very useful.			No. As above	Tried to debug new created java script. Debug functionality is good.
12) why do not have posted them as examples on the portal ?	We will soon together with the produced objects	We did not posted javascript rules as we did not think it was our task. We'll do it if you confirm us we have to. Could you please suggest right place where to put? into axcp folders even if they are not under our responsibility?	Because they are incomplete as some functions are still to be provided by Ivan (managing files and folders)	we want to create better, more interesting examples -- our tests are really of insufficient quality so far.	No. As above	Not able to create a new AND working java script.
13) did you processed objects on the java script ?	We are doing it.	Yes	YES	yes	No As above	No, see above
14) did you connected your database to the AXCP grid ?	No: how to do it? It is not intuitive.	A remote connection has been established since a rather long period. Tests of connection, content search, retrieval and insertion using ILABS provided WEB-Services have	NO	not yet.	We are now trying to configure Focuseek. It is not sample and intuitive and we are having help from Piero Bellini	Not yet

		been performed with good results both in terms of reliability and usage (at least was the result reported to us) with up to 100 seek in parallel…				
What are you waiting for installing the focuseek and connecting your database to the axmedis database via Content Processing ??	We are working on it with our technical staff, finding the best solution for our configuration and framework (also relating to the webservice issue) We will give feedback when there will be relevant figures.	grid installation is planned to next time by our side (after july review) Please note that 1) we provided fully functional description and implementation of a web-service for accessing Tamino, plus a live accessible installation of our accessible in remote to allow Focuseek to design, develop and test the Tamino interfaces. The same infrastructure has been used to test what developed and results reported have been quite good as previously reported 2) lots of efforts have been spent in debugging components not in charge to us and			See above	We do not use an own database which could then be connected to AXMEDIS db.

		helping partners deploy their part (see what has been done for the DB and query support, Editor and AXCP, PMS… as we also had to deliver our demonstrators therefore to meet imposed deadlines we had to schedule and prioritise work. Having a fully and correctly functional version of the editing tool and connected AXCP was far more important than having a fully installed non-functional framework.				
15) did you protected some obejcts ?	No if you refer to the protection viewer. Yes if you refer to DRM.	Yes we tried	Not yet	not yet.	No.	Yes
16) did you installed the AXMEDIS database tools ?	Yes – the process was very low level and much time was needed in the manual configuration and tweaking of parameters such as the individual paths. The process was	Yes we did and the process was not pain-less. We had need for quite intensive support and almost acted as "guinea pigs" furthermore once the editor was put in place it has been	NO	this is still in progress but quite complex. We have tried to install AXDB on Tomcat on the Mac OSX platform rather than a windows server. We do not have a large CMS database	Yes but with many difficulties as it is too complex	Not yet

	equally long and tricky for the parameters to setup un the editor for query support. Some form of step-by-step, possibly with auto-detection of certain parameters would make the process simpler and faster.	quite complex to interface with the already installed DB (the DB was installed prior to the editor)		at XIM so the AXDB was a lower initial priority for us, but of course we still need to fully test it as part of a complete typical factory setup.		
17) did you loaded some objects ?	Yes	Yes samples that were either available or provided	Yes	not yet	No. Not able and while trying the system fails.	Not yet
18) how many objects did you posted into the database ?	Just some for doing query, upload, download, embed etc. now. We also uploaded some templates to use for the instruments objects: downloading from the database and then adding the new resources	Just the samples and some of the very basic results achieved so far Please read carefully previously reported comments.	0	none yet	0	See above
19) did your produced some queries into the database you installed ?	Yes	Yes	NO	not yet	No. We tried to do it but again some technical problems raised and sections of the queries interface and form are not intuitive and user friendly.	See above. I only simulated the creation of queries.
20) did you created queries from the java script in	We are doing it	No since it has not been necessary until now in carrying on	NO	not yet	NO	No. See 11 above.

AXCP ?		our scenarios. We'll manage this aspect as soon as it will be necessary basing on our scenarios and schedulation				
21) what do you think about usability of AXMEDIS Editor ?	Usability is quite good relating to the development level. As already stated, some tools seem too “technical” for an end user. From are testing these sepcific aspects also emerged: The installer is still very basic. More friendly and easy ways of setting up – for example – the params for the query support etc. would be very helpful. Many programs do not run correctly because of missing DLL etc. if installed with the installer (Strongly reccomended): Possibility for multiple selection of objects in the hierarchy view...	Same aspects are not so intuitive (especially as far as object rendering is concerned) but we believe that at present the results of the trials we have performed are not enough to reply, there are still many instability in the system that we are not able to understand if they are caused by the way the system has been installed or if they depend by system bugs. We have requested availability for a meeting to try to solve together all these issues We tested V0.1, V0.2, V0.3, V1, V2.1 and the install procedure of V2.2. except for this last point we have always provided	Too technically oriented... Could benefit of "inductive user interface" approach, i.e. from this view what task am I able to accomplish...	initial impressions were good, familiar GUI, straightforward so far, although we have had some instability issues. This deserves a longer answer and requires more time, but this is the initial impression.	It is completely left to user capability to understand a certain language and to have technical skills. Not suitable to user but to technical people. Many steps to follow are not explained	See the comments provided in the report which has been circulated to all partners: Usability Inspection of AXMEDIS Tools v1.1 and which is also available for download at the portal.

	useful, for example to delete, move objects etc. (Suggested): Show some kind of progress/feedback to the user when loading/saving big objects (that take long)	info to developing partners on issues encountered, I would rather say that in many cases we have even pointed out suggestions and hints acting more as debuggers than as users				
22) which features are missing in the AXMEDIS Editor ?	various, but their lack could just be related to the development status (e.g. more intuitive installation setup which we know is being developed)	Stability above all : editor crashes too often, it's very stressing for content developers. Usability for SMIL Editor. Maybe in tree view it would be nice to visualize also the type of the resource, not only the name: icons are not enough, since all images have the same icon, and similarily all audios have the same icon too. Undo seems not to work. Possibility of "composing" a textual page in a "power-point-way"-like: setting a background, adding	Too soon to tell	difficult to say at this stage, but we will compile a feature list	At this stage the following; □ tools need a more user friendly graphic interface □ Tools functions and language used are too complex and not suitable for content owners and users □ Easy guides for users should be developed □ We could not even open the guides actually available on each tool.	On-line-help for: (1)how to modify the configuration files, (2) how to modify/create java scripts, (3) how to perform realistic task scenarios. Undo and Redo functions (are not yet implemented in all the tools). A possibility to customize the tools for different types of users (first time vs. expert users, users who have to perform simple tasks only vs. professional users who need the whole functionality, etc)

		and formatting text, adding hyperlinks, synchronizing audio. SMIL editor should synchronize image and audio, AXCP can manipulate images... but seems that it is not possible to put the whole result together (we would like to have, for example an image with text and audio in the same page = one audio/visual impact only) Encountered problems have been reported in the action plan and will be posted to the wiki once properly formalised. A list of desiderata will be drawn on the basis of an ongoing analysis of what initially requested, what delivered and testing results				
23) what do you think about usability of AXMEDIS player ?	It is quite usable.	For basic assets is good enough, for the kind of objects we are thinking is difficult to say both because we did not	average (for instance, the zoom is too slow in terms of increments)	initial impressions: good but basic (which is to be expected at this stage of evolution)	Audio files do not work.	Acceptable to me.

		manage to produce them the way we expected and because we did not manage to understand how we should then play them Encountered problems have been reported in the action plan and will be posted to the wiki once properly formalised. A list of desiderata will be drawn on the basis of an ongoing analysis of what initially requested, what delivered and testing results. Some problems have been experienced about "random" crashes – they seemed to be due to memory management – Hard to say if problems have been solved or not since we have upgraded the executable the "random" occurrence of failures seems to have ceased, but given its previous				

		random nature we could not identify a possible set of steps that could bring to such crash even though other crashes still occur (e.g. visual and behavioural editor…).				
24) which features are missing in the AXMEDIS player ?	At this stage hard to say single features. In general at this stage it looks more like an “explorer/viewer” than a real “player”. But probably better testing could be made once the behaviour editor has been implemented and behaviour can be put in the objects to make them more “interactive” UPDATE: - opening audio and vido files which are linked in an HTML page directly in player - decide in the object the initial size of the player	A sequential "slide show" oriented display is not enough, hypertext and navigation oriented approach should be covered too. Encountered problems have been reported in the action plan and will be posted to the wiki once properly formalised. A list of desiderata will be drawn on the basis of an ongoing analysis of what initially requested, what delivered and testing results. Need of a distinction of “what has to be displayed” and what is only “a resource”: if there’s a html page with an image,	Too soon to tell as some features are not implemented yet (annotation, bookmarking, going to a specific time index, etc.)	difficult to say at this stage, but we will compile a feature list	See above	--

	- autofit of images - ”tag” in some way resources that can be visible to player in the structure and ones that must be hidden	the whole page has to be displayed, not the image also.				
25) what do you think about usability of AXMEDIS database ?	It is very low-level	So far so good	N/A	can't say yet	Can’t say yet	Not used yet.
26) which features are missing in the AXMEDIS database ?	More friendly GUI, especially for connection and setup Clearer setup guide	Can't say yet, still under evaluation	N/A	can't say yet, we will compile a feature list		See 25 above.
25) did you installed the Programme and publication ?	Yes	Yes – quite straight and well explained	NO	yes	yes	Yes
26) did you installed the workflow ?	Yes	Yes – very long and hard… doc is sometimes unclear. But after workflow installation, communication with other components has to be established: this is the hardest part. Until now we tried to call from workflow: 1)axeditor ->	NO	yes	yes	Yes, but failed the first time. Another installation with the help of Maulik Sailor and Mauliks tutorial contributed a lot to understanding the complex installation procedure.

		success 2) rule editor -> ko (Directory ‘’ couldn’t be created (error 3: impossibile trovare il percorso specificato) as soon as editor windows acquires focus) 3) pnp -> programme activation succeedes but notification is not sent to workflow – there are also troubles with wprkflow task management Lot of time spent for 3, also some time for 2)… and they are still not working properly				
27) did you used the fingerprint for audio, video, and text ?	No. We can’t manage to get it working: Note for the editor: the editor gives a message “Unable to find plugin C:\....\audioFPplugin.dll”) but the file is actually there	No, these functionalities should be transparent in our opinion even though we should be able to know how they work, which are their results, in which format they handle and store data as such info may have a direct impact on internal data processing	Not yet	not yet	No	Yes. See comments in the report: Usability Inspection of the AXMEDIS Tools v1.1

		steps (storage, search & retrieval, verifications...) Apart from the previously reported notes we would like to stress that such functionalities should be either provided as services that have to be recalled by the AXCP in the ingestion / protection phase, whenever protection verification is needed and also whenever DRM controlled operations are to be performed. This therefore presents a multi-fold scenario: 1) insertion among rules of specific calls to abovementioned services. 2) automatic usage of such services by tools performing operations that are specifically mentioned / recalled in DRM rules and that require verification 3) a				

		specific human-led operation at editing time (in this latter case procedures for using the tools should be by far simplified)				
28) did you used the adaptation algorithm for audio, video, etc.. ?	Yes if you refer to content processing	Yes for images (worked) and audio (see previous comments)	Not yet	not yet	We tried but it always says that it does not recognize the input file (mime type) and it was an aif file	Yes. See comments in the report: Usability Inspection of the AXMEDIS Tools v1.1
29) did you used the descriptors for audio video, and text ??	Yes	No also in this case functionalities should be transparent in our opinion even though we should be able to know how they work, which are their results, in which format they handle and store data as such info may have a direct impact on internal data processing steps (storage, search & retrieval, verifications...) Apart from the previously reported notes we would like to stress that such functionalities should be either provided as services	Not yet	not yet	No	Yes. See comments in the report: Usability Inspection of the AXMEDIS Tools v1.1

		that have to be recalled by the AXCP in the ingestion phase or whenever editing / indexing / retrieval is needed. This therefore presents a twofold scenario: 1) insertion among rules of specific calls to abovementioned services. 2) a specific human-led operation at editing time				
30) which algorithms are missing ?	--	So far we had problems also with some of the available algorithms so we cannot say if something is missing, yet we hope that the conversion tables reported in the requirements have been covered (at least as far as the mandatory ones have been implemented)	Too soon to tell	can't say yet, we will compile a list of desirable algorithms.		--
31) which features are missing the tools ?	In general friendly (tutorial based) help system, wizards and templates for setup and object creation	Installation procedure should be provided with adequate documentation (we	Too soon to tell	initial impressions: the installation procedure needs to be improved. MS runtime DLLs	Installation phase is too complex. We also agree with ILABS.	Non technical persons will need more detailed descriptions of how to install (eg.

		cannot expect the end user to be a programmer nor a system manager); moreover a clearer mapping of tools and related components should be available; some basic example and the procedure followed to achieve them should be provided to enable users to understand how to use the system and its components. From what we could see of V2.2 the development team is moving in the line we where hoping.		were missing from some tools but not others, so we had manually search and copy these to get things working (eg MSVCP71.DLL missing from PaP rule editor but present in axmedis editor, while other MS dlls were present in rule editor but missing in axmedis editor). This would not have been a problem on a development machine with visual c++ present, but we were testing on content production machines.		workflow) and configure tools. On-line-help for: (1)how to modify the configuration files, (2) how to modify/create java scripts, (3) how to perform realistic task scenarios. Undo and Redo functions (are not yet implemented in all the tools). A possibility to customize the tools for different types of users (first time vs. expert users, users who have to perform simple tasks only vs. professional users who need the whole functionality, etc)

8 Bibliography

ANSI/HFES-200-199x - Ergonomic Requirements for Software User Interfaces

Apple Computer, Inc. Macintosh Human Interface Guidelines
http://developer.apple.com/documentation/UserExperience/index.html

Beckwith, L, Burnett, M., & Cook, C. (2002). Reasoning about many-to-many requirement relationships in spreadsheets. In HCC'02, pp. 149-157.

Bevan, N., & MacLeod, M. (1994). Usability measurement in context. Behaviour and Information Technology, 13, 132–145.

Beyer, H., & Holtzblatt, K. (1998). Contextual Design: Defining Customer-Centered Systems. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Bias, R.G., & Mayhew, D.J. (eds.) (1994). Cost-Justifying Usability. London: Academic Press.

Brown, P., & Gould, J. (1987). Experimental study of people creating spreadsheets. ACM Transactions on Office information Systems, 5, pp. 258-272.

Burnett, M., Cao, N., Arredondo-Castro, M., & Atwood, J. (2002). End-user programming of time as an 'ordinary' dimension in grid-oriented visual programming languages. Journal of Visual Languages and Computing, 13(4), pp. 421-447.

Caplan, S. (1990). Using focus groups methodology for ergonomic design. Ergonomics 33(5), 527-533.

Carroll, J. (1995). Scenario-Based Design - Envisioning Work and Technology in System Development. New York: John Wiley & Sons.

Constantine, L.L., & Lockwood, L.A.D. (1999). Software for use. A Practical Guide to the Models and Methods of Usage-Centered Design. ACM Press,NY.

Cooper, A. (1995). About Face. The Essentials of User Interface Design. Programmers Press, IDG Books.

Corritore, C., & Wiedenbeck, S. (2000). Direction and scope of comprehension-related activities by procedural and object-oriented programmers: An empirical study. In Int. Workshop on Program Comprehension, Limerick, Ireland, pp. 139-148.

Denzin, N.K., & Lincoln, Y.S. (2000). Handbook of Qualitative Research. London: SAGE, Thousand Oaks.
Dumas, J., & Parsons, P. (1995). Discovering the way programmers think about new programming environments. CACM, 6(38), pp. 45-56.

Dumas, J. S., & Redish, J. C. (1993). A Practical Guide to Usability Testing. Norwood, NJ: Ablex Publishing.

EC Directive 90/270. European Directive on the minimum health and safety requirements for work with display screen equipment.

Edwards, W.K., Bellotti, V., Dey, A.K., & Newman, M.W. (2003). The challenges of user-centred design and evaluation for infrastructure. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 297-304). ACM Press.

Green, T., Petre, M., & Bellamy, R. (1991). Comprehensibility of visual and textual programs: A test of superlativism against the 'match-mismatch' conjecture. In Empirical Studies of Programmers: 4th Workshop, New Brunswick, NJ, pp. 121-146, Ablex.

Hackos, J.T., & Redish, J.C. (1998). User and Task Analysis for Interface Design. Chichester: John Wiley & Sons.

Hix, D., & Hartson, H. R. (1993). Developing User Interfaces: Ensuring Usability through Product and Process. New York, NY: John Wiley & Sons.

Hoadley, C.M., Linn, M.C., Mann, L.M., & Clancy, M.J. (1996). When, why and how do novice programmers reuse code? In Empirical Studies of Programmers: 4th Workshop, New Brunswick, NJ, Ablex.

IBM Corporation. Object-Oriented Interface Design: IBM Common User Access Guidelines http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/558

Isensee, S., & Rudd, J. (1996). The Art of Rapid Prototyping. London: International Thompson Computer Press. ISO 9126. Information technology - Software product evaluation. Quality characteristics and guidelines for their use.

ISO 9241 Ergonomic requirements for office work with visual display terminals

ISO 13407 Human-centred design processes for interactive systems

ISO23973 Web Usability Standard. http://www.userfocus.co.uk/articles/ISO23973.html

ISO/IEC 9126 parts 2 and 3 contain criteria for the evaluation of user interfaces

ISO 14915 and IEC 61997 contain recommendations for multi-media interfaces

ISO 10741 Dialogue Interaction

ISO/IEC 11581, Icon symbols and functions

ISO/IEC FCD 18021: Information Technology - User Interface for mobile tools (2001)

Jones, S.P., Blackwell, A., & Burnett, M. (2003). A User-Centred Approach to Functions in Excel. ICFP'03, August 25-29, 2003, Uppsala, Sweden, p. 165-176.

Li, Y., Grundy, J., Amor, R., & Hosking, J. (2002). A data mapping specification environment using a concrete business form-based metaphor. In HCC'02, pp. 158-166.

Mayhew, D.J. (1999). The Usability Engineering Lifecycle. A practitioner's handbook for user interface design. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

Mayhew, D.J. (1992). Principles and guidelines in software user interface design. Englewood Cliffs, NJ: Prentice Hall.

Microsoft Corporation. The Windows Interface Guidelines for Software Design: An Application Design Guide
http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/anch_uidesigndev.asp

Mullet, K., & Darrell, S. (1995). Designing Visual Interfaces. Prentice Hall.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. Proc. ACM CHI'92 (373-380).

Nielsen, J. (1993). Usability Engineering. Boston, MA: Academic Press.

Nielsen, J., & Mack, R. L. (eds., 1994). Usability Inspection Methods. New York, NY: John Wiley & Sons.

Nielsen, J. (ed.) (1994). Special Issue: Usability Laboratories. Behaviour and Information Technology, Vol. 15, Number 1 and 2.

Open Software Foundation. OSF/Motif Style Guide. Revision 1.2

Preece, J., Rogers, Y., Sharp, H., Benyon, H., Benyon, D., Holland, S. & Carey, T. (1994) Human-Computer Interaction. Wokingham, UK: Addison-Wesley.

Robertson, S.,& Robertson, J. (2004). Requirements-Led Project Management. Addison-Wesley.

Rubin, J. (1994). Handbook of Usability Testing. New York, Chichester: John Wiley & Sons.

Shneiderman, B. (   ). Golden Rules of Interface Design.

Sun Microsystems, Inc. JavaTM Look and Feel Design Guidelines, second edition
http://java.sun.com/products/jlf/ed2/guidelines.html

Sun Microsystems Inc. Open Look: Graphical User Interface Application Style Guidelines
http://docs.sun.com/app/docs/doc/802-2109?q=OpenStep

Thomas, C., & Bevan, N. (1996). Usability Context Analysis: A Practical Guide. Version 4.04. National Physical Laboratory Teddington, Middlesex, TW11 0LW, UK. This guide and forms can be downloaded from: http://www.usability.serco.com/trump/methods/recommended/context.htm.

VNET5 http://www.vnet5.org

Vredenburg, K., Isensee, S., & Righi, C. (2001). User-Centred Design: An Integrated Approach. Prentice Hall.

Web Content Accessibility Guidelines 1.0
http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/

9 Glossary

Assessment criteria are critical values for relevant measures which are the basis for the assessment of an electronic information service or product.

Context of use describes the technical, physical and social environment in which the application is used.

Critical success factors determine the success of the electronic information application for the organization. Critical success factors can be product oriented (e.g. higher product quality, innovative design), development process oriented (e.g. more efficient and effective development process), standards oriented (e.g. product complies to standards), societal goals (e.g. product can be used by people with special needs).

Customer is a person who decides to purchase a system. The customer has a commercial view.

Design guidelines give advice about how to design a user interface.

Design principles are high-level and universally applicable directing guidelines.

Formative evaluation takes place before implementation (e.g. quality assurance during development) in order to influence the development of the product.

Measurement means a repeatable, objective procedure for generating a measure. The resulting measure(s) are scaled in a known way, and reference values, reliability and validity are known. Detailed information about 'Measurement' 'Construct Validity', 'Reliability', 'Levels of Measurement', 'Survey Research', 'Scaling', 'Qualitative Measures', 'Unobtrusive Measures' can be found at trochim.human.cornell.edu/kb/measure.htm.

Measures are operationalized quality factors. A measure can be subjective or objective, direct or indirect, analytical or empirical.

Method is the formal definition of measures. It describes the means to arrive at reliable and meaningful measures. A validation method is a repeatable, systematic procedure to produce a given result. The specific aspect of validation methods as opposed to a general view of method is that user validation always starts with an objective and validation criteria, i.e. a question such as 'Is design alternative A more efficient to use than B ?', 'Does the application fulfill the minimum requirements?'. A quality general factor such as efficiency is implied by the validation question, and the resulting measure must be shown to be a valid measure for such a quality factor.

Metrics are measures possessing metric properties which express the degree or strength of a quality factor. Metrics are obtained by an objective measurement method. Their scale of measurement is known, they possess scale metric properties, known maxima, minima and reference values, their reliability and validity are known. Metrics are interpreted according to the context in which they were measured. Metrics allow comparisons between applications, as well as between alternative versions of a single developing application, and comparisons with reference values. Conformance to standards and minimal requirements can be tested with metrics. Some metrics tell about the performance of the user applying an application to his work (e.g. efficiency of use, learning effort, errors). Other metrics predict quality of use factors on the basis of a user interface specification, prototype, or fully functional product, and an underlying user model.

Prototype is an experimental design of the whole or part of a system. The purpose of a prototype is to test certain aspects or characteristics of a new system. A prototype can be paper- or computer-based. A vertical prototype contains both high- and low-level functionality but for a restricted part of the system only. A horizontal prototype contains all the high-level functionality of the final system but misses out the lower-level detail. Prototypes may be scale or partial models, may be partly non-functional, or may be full-scale 'trial' objects or processes, and may be designed to be discarded. Prototyping is the simulation of an object or process which is then subjected to systematic testing. Rapid prototyping is a form of simple, rapidly produced prototyping in which the prototype is used to collect information about both the requirements and the adequacy of possible designs; it is not developed into a final product.

Quality is a multidimensional concept consisting of quality factors (also called quality dimensions). Quality dimensions are features by which a product can be assessed such as efficiency of use, task adequateness, cognitive workload, robustness, learning cost, user acceptance. Quality dimensions are the result of the decomposition of the term ‚quality of the application'. They are variables which reflect different independent quality aspects of the application. Validation questions must be formulated in terms of quality factors in order to allow meaningful measurement.

Qualitative data can be categorized in some way but cannot be reduced to numerical measurements.

Quantitative data consists of numerical values.

Quality of use is used synonymously to usability. Quality of use is a concept which consists of multiple, measurable dimensions (e.g. productivity, user preferences). An application's overall quality of use is then determined by specifying the dimensions which are relevant in a certain context of use, by adding priorities to the different quality dimensions (this is optional), and by defining assessment criteria for the dimensions. Quality of use describes the user-centered view of product quality. There are other quality aspects, e.g. technical quality such as portability, maintainability, etc.

Reliability of a measure describes the degree of stability, accuracy and error associated with a measurement procedure. It describes the extent to which a measurement procedure yields the same result if carried out on different occasions, possibly by different people on the same object. A measure is reliable if the application of the measure yields reproducible results. Factors which could reduce the reliability of a measure could be for example number or attributes of subjects involved in experimental tests or subjective decisions made by the evaluator.

Summative evaluation takes place after implementation to test the proper functioning of a product and to investigate user satisfaction and preferences.

Task is described in terms of the goals or a desired end-result of activities a user wants to achieve. More than hone user procedure (a sequence of commands to be executed to reach the goal) may exist to solve the task.

Usability as well as other traditional terms such as user friendliness, usefulness, ease of use, have in common that they are vague and fuzzy terms. They give the impression of just one single dimension. However, users have different needs and requirements and perform different tasks with an application. An application which is usable by one user may be tedious to use by another user. In addition usability is a too narrow concept which does not take into account cost/benefit issues. Hence, the term 'quality of use' is preferred to usability.

Usability Engineering is a well defined process which is performed as part of the application development process. It can be part of the development process of any type of electronic information application. Although each development project is different, the approaches, methods, techniques and activities applied to achieve usability do not vary much.

Usability Specification quantifies target levels of quality of use for a system in terms of: learning requirements, effectiveness, efficiency, robustness, task adequateness.

User is the person who uses a product or service, who has "hands-on experience".

User-centred design emphasizes on early and continuous involvement of users in the design process.

User procedure is a sequence of commands which a user must perform to solve a task or to reach a goal.

User validation describes the gathering of quality of use information about an application which is used within a specified context of use. The user validation process describes a set of ordered activities that contribute to a defined objective of a validation project. A user validation process takes place over time and has precise objectives regarding the results to be achieved. A User Validation Process Model describes the structure and the elements of that validation process in terms of stages and steps, dependencies and data.

Validation emphasizes on early and continuous involvement of users in the development process.

Verification checks if the developing product or service is free of errors and if fulfills the user / customer requirements described in the system specification.

Validity of a measure A measure is valid for a given quality aspect if there exists a correlation of the measurement values and that quality. Validity describes the extent to which a measurement procedure measures what it is intended to measure. It is a psychometric concept, and a number of different ways of testing it have been used: Content validity: Measure of the sampling adequacy of the rules of measurement; Criterion validity: The comparison of the measures obtained against external variables or criteria; Construct validity: This refers the rules of measurement to an underlying theory or model; Ecological validity: The extent to which the conditions simulated in the laboratory reflect real life conditions. Face validity: Does the measure appear to be measuring something sensible?

10 Appendix 1: User validation plan

This User Validation Plan was created with the VNET5 User Validation Planner (www.vnet5.org). The document will be submitted for review and further completion to the user partners. The process owners for user testing and validation of the AXMEDIS framework and tools are the AXMEDIS user partners, partners involved in AXMEDIS take-up-actions, and members of the AXMEDIS user group.

This is a working document expected to be completed continuously as the AXMEDIS project progresses.

10.1 About this user validation plan

This plan is designed to help the AXMEDIS project plan and manage user validation activities. These activities are important in creating products intended for people to use.

The user validation plan is for people who are interested in **-** or responsible for –taking a user-centred approach to product creation.

The user validation plan should be used as a working document, and a source of basic guidance for user validation planning.

The plan was prepared based on information from D2.1.1a/b User Requirements and Use Cases, and other documents.

Involvement from key people who have a stake in the product design, development, marketing and use is essential for the AXMEDIS project to succeed in being user-centred.

User validation cannot take place without good communication. This plan keeps people well informed about what is going on. Share this user validation plan (and updates) with all team members!

10.2 Parameters of the development project

10.2.1 The AXMEDIS framework and tools

The AXEMDIS project aims to research and develop innovative technologies to reduce the costs of digital content production and distribution, including protection models. AXMEDIS is an environment where content producers, aggregators and distributors can gain access to a wide range of digital contents (AXEPTool P2P B2B) and innovative technologies (AXMEDIS framework).

AXMEDIS will organize and realize several demonstrators to deliver different types of digital content via different distribution channels (Internet, mobiles, PDA, PC, I-TV, satellite).

At a later stage, companies and research institutions which are interested in exploiting AXMEDIS technologies can participate in take up actions, funded by AXMEDIS, to develop sustainable real solutions.

Prototype tools, prototypes of tools integrated into the AXMEDIS framework, demonstrators and real solutions will be subject to user validation.

10.2.2 The stakeholders

The main people or groups who have an interest in the results of the AXMEDIS project (e.g. sponsors, designers, technical team, marketing, user groups, etc) are:

- The AXMEDIS user partners (in their role as AXMEDIS tools owners; employees of these user partners in their roles as users):
  - Archives, institutions, schools and content producers (ANSC)
  - Associations of content producers (AFI)
  - Publishers and digital content providers (ILABS, AFI, ANSC, BORDAS and NATHAN of SEJER)
  - Content integration and design, audio and video (XIM, ILABS, SEJER)
  - Content distributors (OD2, SEJER) with content distributed on behalf of UNIVERSAL, SONY, EMI, WEA, The Orchad, Loudeye, AFI, etc.
  - Networks, broadcaster and their technology providers for I-TV-PC (EUTELSAT)
  - Mobile distributor (COMVERSE) for GSM cells or UMTS, etc.
  - Content distributor operators and technicians towards PC on internet (TISCALI, OD2)

- The development team of the AXEMDIS framework and tools
  - Leading European research institutions, who contribute to
    - Cross-media content production (DSI, DIPITA, EPFL, CRS4, UNIVLEEDS, IRC)
    - Distribution and transaction models (DSI, EXITECH, FUPF)
    - DRM and protection (FUPF, FHGIGD, DSI)
    - Simple and complex queries (FHGIGD, CRS4, EXITECH)
    - Usability aspects (ACIT, IRC)
  - Technologies providers and integrators (HP, EUTELSAT, COMVERSE)
  - Content Associations and legal experts on IPR (AFI, ILABS, SEJER, OD2)

- AXMEDIS project management (DSI)

- EC as the AXMEDIS project sponsor

10.2.3 The business objectives and rationale for the project

Why is it being developed? Outline the principal business and organisational objectives for developing the new product or service (if relevant compare with competitive or traditional applications).

Market and end-users are pressing content industry to reduce prices. Reasonable prices are the only solution to set up viable and sustainable business activities with e-content. This is why content providers, aggregators and distributors need more efficient tools.

The main aim of the AXMEDIS project is to develop innovative methods and tools which speed up and optimize content production and distribution through means of automating, accelerating and restructuring the process to make it faster and cheaper.

The objectives in detail are:

- Reducing cost for content production and management by applying Artificial Intelligence techniques to content creation, representation (format) and workflow,
- Reducing distribution and aggregation costs in order to increase accessibility with a Peer-to-Peer (P2P) platform at Business-to-Business (B2B) level, which can integrate content management systems (CMS) and workflows,
- Developing and providing new methods and tools for innovative and flexible Digital Rights Management (DRM),
- Exploitation of MPEG-21 and overcoming its limitations,
- Supporting different business and transaction models.

10.2.4 Requirements and constraints for the development project

What are the working/operating constraints? Outline the principal quality requirements, standards requirements, and legal and technical requirements and constraints.

Technical requirements:

- Integration of Content Management Systems, which are currently in use by the user partners, with the AXMEDIS system (DE2.1.1a page 204)
    - XML CMS based on Extraway engine
    - COMVERSE Content Management System
    - Eutelsat Multicast Toolkit (EMT)
    - LEX - learn eXact(c)
    - HP proprietary CMS
    - Royalty Management System (RMS)
    - XAURA
    - in-house developed CMS written in PHP
    - SEJER proprietary CMS for XML repository, another one for media content

- CMS Platform requirements:
    - Windows XP, 2000 Professional
    - Windows 2000 Server +
    - Win32 based platforms
    - W2003 server with SQLserver
    - Mac OS 9.2, OS X
    - GNU/Linux Mandrake 10 +
    - GNU/Linux Slackware 10
    - Linux Debian
    - Linux SuSE 9
    - Red Hat Linux (7+)
    - SUN Solaris

    - Migration plans
        - XAURA 2

    - Multimedia content formatting tools:
        - Audio: WindowsMedia Encoder (v8, 9), Digital Performer 4 (with various plugins), PEAK, Audio file
        - Image: Photoshop, Adobe Illustrator, Adobe InDesign, Fireworks, Image file
        - HTML layout: Learn eXact Packager; Web animation Flash MX; 3D modelling: Maya, 3D Studio Max, general purpose text editors, Macromedia Dreamweaver MX, Homesite, Microsoft Front Page, Flash MX;
        - Web animation: Flash MX;
        - Encoding: Windows Media Encoder, Helix Producer
        - Video editing: Final Cut Pro;
        - Video compositing and effects: After Effects;
        - 3D modelling and rendering: Maya;
        - Compression: Sorenson
        - Squeeze: Uncompressed: AVI / WAV, High res. Mpeg / mp3
        - Tools included in Comverse CMS
        - Tools included in EMT: Secure Data Broadcast (SDT), Corporate TV (CTV), Private Live Broadcast (PLB).

- Multimedia content automatic production tool:
    - Production: iMovie, iPhoto, iDVD; batch video format conversion: Apple Compressor / Discreet Cleaner, PEAK, DECK, PRO-TOOLS, ITUNESo, Adobe Premiere, Learn eXact Packager, Windows Media Encoder, Niagara;
    - Video post production: FINAL CUT
    - Wave Lab for audio
    - OD2 content production applications
    - TISCALI: mainly internally developed tools devoted to video capture (based on Adobe Premiere) and multi-encoding (based on Windows Media SDK and Helix Producer)
    - Cygwin GNU tools (zip, sed, perl, GNU Make,), MS Tools
    - Comverse CMS and 3rd-party studios supplies content in the required formats

- Multimedia content protection / DRM:
    - Any audio track produced should include the ISRC code which allow to identify main info on that track (country of production, original producer, year of the issue, identification number)
    - We want to find, in the AXMEDIS context, something of more OPEN and STANDARD and NOT OWNERSHIP of a single company
    - Verisign Certificates will be adopted for usage with SDK
    - Learn eXact & Tamino internal tools
    - WindowsMedia DRM
    - Microsoft Windows Media Rights Manager
    - SEJER: Proprietary file/stream format based one gluefish(protection) and zip(compression) libraries. We have developed our own encoder and our own mozilla client decoder (Protocol Handler)

- Specific hardware:
    - Apple MACs
    - Windows 2000 and XP
    - P3/P4 adequately equipped MM-PC
    - Dell servers with Quad Xeon processors
    - Dell Optiplex workstations and Dell Power Edge servers
    - HP EVO notebooks
    - MOTU audio interfaces
    - Firewire DV in-house
    - DV, BetaSP, DVD, DigiBeta, S-VHS, VHS players from main manufacturers (Sony, Panasonic, Philips and others)

- Platform / Operating System:
    - MAC OS X 10.3.5
    - Windows 2000 pro; Windows XP; Mac OS 9.2 (in future MAC OS X)
    - Windows 2000 Server
    - Linux RedHat 7.3 and upper; Linux Fedora Core 2; Windows Media Encoder/Server, Windows Media DRM; Encoder/Licenser
    - Unix platform to use Cygwin

- Tools in-house
    - OD2 Ripper; OD2 Transcoder; OD2 Packager / Packaging on Demand (PoD)

Main access devices:

To be completed later

Legal requirements

- Protecting and controlling the use of digital contents in a scheduled and automated manner
- Privacy of users

Objective quality requirements:

- Work with the AXMEDIS framework and tools must be more efficient compared with the current/traditional work processes.

Subjective quality requirements:

- User satisfaction with AXMEDIS framework and tools should be improved in comparison to the existing tools.

10.2.5 Critical success factors

What matters most for success? Determine the critical success factors of the stakeholders in the project?

Use of the AXMEDIS framework and tools should increase efficiency and thereby reduce cost visibly (to justify the investment to switch to AXMEDIS).

The added value should be seen as positive on medium to long term by the management of the user partners, by the owners of the AXMEDIS tools, as well as by the users.

The AXEMDIS tools should be easy to use, easy to learn, leading to high acceptance in the initial phase of use of each user.

The new functions should add visible value for the users, notably: Access to more content, Web usability.

Existing privacy requirements should be assured.

The process of iterative design-evaluation cycles should help the technical WP4 and WP5 to generate and test new ideas and solutions.

At the end of the project the summary evaluation of the total quality for users and the cost/benefit of the entire AXEMDIS framework and tools for the user partners and external users should prove the added value of AXMEDIS technology.

10.2.6 Who needs and who will use the validation results?

The team of developers involved in WP4 and WP5 need results of usability inspections of components and prototypes with user interfaces: eg. AXMEDIS editors and viewers, work flow manager, query support. Feedback is needed:

- whether the basic concept and the use of the envisioned technology is feasible
- whether high level design deficiencies and user problems in the interface design were detected
- further requirements concerning design features.

Formative evaluation of a prototype for a subset of the AXMEDIS functionality (M18) **-** for use by an increasing circle of users, and a succession of prototypes from month 19 to 48, integrating further and improved technology components. Feedback is needed on design deficiencies and user problems and further desiderata for development within the resources and capacity of the AXMEDIS project.

Summative evaluation at milestones M36, 48 to provide feedback about the cost/benefit and added value of AXMEDIS framework and tools and acceptance by users.

10.2.7 Constraints for user validation

Timing and availability of prototypes:

- AXMEDIS tool components M12
- operational prototype with integrated tools and limited functionality M18
- improved prototypes and demonstrators with additional functionality M24, M36, M48

Access to users: A motivated group of selected users is expected to be available during the development phases. A larger user group will be involved in the testing of the operational prototype and of the demonstrators.

Budget is not specified in detail, but limited to an appropriate share of the development budget for WP4 and WP5.

Available personnel resources for validation:

The main testing will be carried out by members of the development team (with limited experience in usability engineering) in cooperation with the user partners and volunteers from the user group. Help from the usability partner will be available, eg for user preferences analysis, who can perform part of the data collection.

Motivated users are expected to be available flexibly from the user partners and by offering incentives where needed to members of the AXMEDIS user group. Tests are carried out by way of integration into the normal work procedures and environment.

Available skills and experience of personnel:

The competence to carry out evaluation and user validation will be developed (as part of AXMEDIS training) or is available. ACIT will be able to coordinate and guide the user tests where needed.

10.3 When to act, what to evaluate

10.3.1 When is it really possible to do something about being user-centred in the project?

Milestones in the project define when and where user validation can take place:

M6	User need and requirements analysis, use scenarios, UML use cases, test cases
M12	First prototype tools
M18	First version of AXMEDIS framework with integrated tools
M24	Testing of AXMEDIS demonstrators (prototype applications) by larger number of users
M36	Testing of more demonstrators build by Take-up actions
M48	Final test to provide a summative assessment of the tools integrated into the AXMEDIS framework.

Tests will take place at the normal workplaces of the user groups.

What are the opportunities to make changes to design based on validation findings?

The main inspection should take place from M12 and before M18, where meaningful intermediate release dates for feedback to developers remain to be defined. The development interacts closely in the evaluation, and receives feedback immediately. There are immediate opportunities to modify the design and implementation by the development team. Monitoring of progress should take account the need to have sufficient time and resources remaining to make meaningful adaptations to the application.

10.3.2 Focus of the evaluation

What is it that really matters for the product's success?

The focus of evaluation during M12-18 will be on:

- How well the interaction design supports user performance (e.g. task completion, navigation, information finding, error avoiding),
- Evaluation of a language for scripting rules for content processing using the Cognitive Dimensions approach

The focus of evaluation during M19-M48 will shift to:

- User preferences, user satisfaction and acceptance
- Usefulness of metadata for content retrieval
- Assessment of the benefit associated with specific parts of the functionality, user preferences and tradeoffs
- Cost / benefit and added value of the product or service

10.4 Users, tasks and context of use

The rationale for the description of users, tasks and the context of use in the detailed and concrete format proposed here is the need to define precise, concrete tasks and operations, which are a condition for precise, meaningful and interpretable tests.

Tasks are identified in a top-down manner, from the definition of system objectives, use cases or scenarios of use, down to detailed procedures. Sometimes this is called "hierarchical task analysis" - the decomposition of global goals and procedures into increasingly detailed descriptions of partial tasks.

The decomposition stops when the appropriate level of detail is achieved. This depends on the aspects of the system which are under investigation, and may range from high-level scenarios to keystroke-level description. Initially the detail should not be to fine-grade, because this usually assumes detailed technical implementation.

Different high-level scenarios and tasks will contain common subtasks, for example specific data-entry functions and forms. The advantage is that these subtasks need to be described only once, and on the other hand all higher-level requirements can be mapped into the subtasks.

10.4.1 Users

Groups of users are identified with homogeneous characteristics within a group and different characteristics between groups. The characteristics should be relevant for the AXMEDIS framework and tools which are under development.

The main groups of users to be distinguished are:

Managers/Producers

Legal staff: Legal staff has extensive domain/legal knowledge, but limited knowledge and practical experience with IT technology.

These four groups of professional users:

- Authors/Performers
- Editors
- Designers (text, graphics, audio, speech, video)
- Technical experts

have extensive domain knowledge and some knowledge and practical experience with IT technology.

End users / Consumers: This group of users may use multimedia content for education, edutainment, entertainment and other purposes. Some of these users may have medium to low knowledge of IT technology. Some users may have special needs (because they are elderly people, colour-blind, hearing impaired, have speech deficiencies, etc).

10.4.2 Tasks

The main tasks users intend to perform with the AXMEDIS framework and tools are identified here for each user group described in 8.4.1. This should help to prioritise the tasks, so you can focus your development effort on the things that matter. A user/task matrix summarises the answers to the in following questions:

- *Goal of task (what is the user seeking to achieve?)*
- *Task output (what is directly produced or modified; are there side-effects?)*
- *Criticality of the task output to business & user (high / medium / low; reason)*
- *Choice (can users achieve their task goal without using this system? how?)*
- *Task frequency (typical and range; especially the peak frequency)*
- *Task duration (typical time and range; is there a target time?)*
- *Task flexibility (how fixed is the order & number of steps in this task)*
- *Dependencies & linked tasks (what must be in place; is it part of a sequence?)*
- *Risks and safety (consequences of error or failure; health & safety implications)*
- *Factors which make task demanding (e.g. recording information in real time)*
- *How demanding (compared with other tasks)*

Section to be elaborated in year 2.

10.4.3 Context of use

Describe the physical constraints users can encounter with the product or service:

- Location at home, at work, in public places, in the field, other
- Users access the system at work. Possibly from home?
- Portability

- Ambient noise
- Lighting
- Vibration
- Space
- Contaminants

Describe interface device constraints:

- Limitations or constraints in input and display hardware
- Keypad limits
- Screen size and resolution
- monochrome display
- Platform requirements and constraints (e.g. migration, integration): (see D2.1.1a, p205)

Describe legal requirements and constraints:

- Conformance with standards (e.g. ISO 9241)
- Domain-specific laws
- Digital Rights Management

Describe safety requirements:

- Conformance with safety regulations
- Use in critical situations
- Can erroneous user actions lead to critical situations?

Describe privacy requirements:

- General privacy requirements
- End users are concerned about privacy and anonymity. They should have control over the information they would like to access explicitly or implicitly.
- Domain-specific

10.5 Quality dimensions and assessment criteria

10.5.1 Quality dimensions

Determine the quality dimensions that are relevant for your development project. Order the quality dimension according to their priority from highest to lowest priority. In evaluating, you will have to focus on the most important.

1 Productivity and performance: Efficiency and Effectiveness (including usefulness of scripting language)
2 Added Value
3 Usefulness of metadata
4 Robustness **-** Errors and error correction
5 User preferences, satisfaction acceptance
6 Integration with standard user environments, CMSs, Integration into product family

10.5.2 Assessment criteria

For the top three quality dimensions selected in 8.5.1, describe how you intend to measure it, and determine the assessment criteria. More than one measure per quality dimension may be appropriate. For each measure describe the unit of measurement, the critical, required, and optimal value. Start with the most important quality dimension.

The following table shows the assessment criteria for quality dimensions measured in month 12-18. Assessment criteria for M19-48 will be elaborated further.

Quality Dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value
Productivity (How well the interaction design supports user performance)	Inspection whether cognitive dimensions design principles are fulfilled	Number of problems found			0 problems
Productivity (How well the interaction design supports user performance)	Inspection whether user interface design principles are fulfilled	Number of problems found			0 problems

10.6 Select user validation methods

Reconsider the information collected - is it sufficiently complete? Select appropriate methods for user validation, preferably in consultation with someone with experience of applying these methods in your area of work. Ensure you have a clear rationale justifying your choice of method.

For M12-M18

- More detailed context of use analysis
- Usability Inspection on the basis of user interface guidelines and cognitive dimensions principles by members of the development team (possibly together with domain experts / representative users) before tool prototypes are released to users (allowing for urgent improvements before release, in case these are needed).

For M19-48 (to be elaborated further in year 2)

- User preferences assessment and tailor-made questionnaire for the assessment of added value
- Performance measurements to assess efficiency and effectiveness needed for cost/benefit calculation
- Checklists and expert assessment of the degree of integration with standard user environments, CMSs, Integration into product family
- Tailor-made questionnaire for the assessment of the usability of metadata for information access
- User preferences assessment
- User satisfaction measurement (SUMI) plus additional tailor-made questionnaire

10.7 Do the validation

10.7.1 Prepare for the evaluation activities

Ensure that the person who will be leading the evaluation has sufficient skills and experience in the methods used. If necessary, bring in some outside expertise.

- Sound method and conduct are essential. It is all too easy to get misleading results from small or unrepresentative samples, or from evaluator bias, or by over-generalising from single instances.
- Liaise with the design & development and technical teams over timing and status of what is to be evaluated.

- The readiness of designs & prototypes for testing is critical, so be realistic about timing.
- Make sure you understand the technical constraints of the product, and the time and budget constraints for making changes. Discover what changes are most easily possible and what is more difficult.
- If users are to be involved, make arrangements sufficiently in advance, and try to keep to those arrangements.
- It is not easy to find suitable users who have the time available to participate.
- There are PR aspects for the project when involving outside people in testing.

10.7.2 Conduct the evaluation activities

Involve members of the design & development team and other stakeholders in observing the evaluation activities. This is the single most effective way of promoting feedback into design, and getting people to act on results.

10.7.3 Analyse the data

Be impartial, aim for speed with sufficient rigour, focus on the things that matter for the product's success. 'Too-late' results are no use to the project. Same-day analysis is the norm in much commercial evaluation work.

10.7.4 Feedback results to the development team

Discuss the findings with the development team.
Share with them the implications for how the product (and project) will achieve its quality goals.
Do this as early as possible - discuss interim findings and work-in-progress - don't wait until you deliver the formal report! The biggest value of user validation is how it improves design.

10.8 Report the user validation results

Be concise. The executive summary is crucially important - it may be the only part that busy managers read - so give it sufficient attention and complete it last. Ensure that it highlights key findings and recommendations.

Frame the report so that it is meaningful to all relevant stakeholder groups. Remember that readers will have varying levels of technical, business and ergonomic understanding. Acknowledge all contributions to the work.

Most project managers are looking for 'quick wins' (things that give a big improvement for a small change) and 'show-stoppers' (things that simply must be changed). They will ignore over-long lists of issues identified. The report must prioritise, and focus on the things that matter.

10.8.1 Example structure for a user validation report

Executive summary

1. Description of the development project
   1.1 The AXMEDIS framework and tools
   1.2 Objectives, requirements and constraints of the AXMEDIS project
   1.3 Phases and development status in which the AXMEDIS framework and tools were validated

1.4 Objectives of the user validation
1.5 Critical success factors
1.6 Constraints for user validation

2. The quality strategy
2.1 Validation questions agreed with the users of the validation results
2.2 The validation scenario
2.3 Focus of the assessment
2.4 Quality dimensions and assessment criteria

3. Users, tasks, and context of use
3.1 Description of user groups
3.2 Description of the tasks users intend to perform with the application
3.3 Description of the context of use

4. The validation procedure
4.1 Selected methods for user validation
4.2 The user validation plan
4.3 Description of the validation procedure

5. Analysis of the validation results

6. Recommendations and conclusions

Appendix 2:

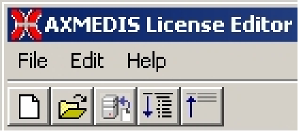
Table 2: Inconsistent use of tool name in different locations – AXMEDIS Tools version 2.3

AXMEDIS Tools menu	**AXMEDIS ContProc Rule Editor**
Message during start	AXCP Rule Editor 1.0
Title bar	AXMEDIS AXCP Rule Editor 1.0
Help –> About … SHIFT+F1	Title bar: Help
AXMEDIS Tools menu	**AXMEDIS DRM Editor**
Message during start	AXMEDIS
Title bar	AXMEDIS License Editor
License Window	AXMEDIS DRM Editor and Viewer
Help –> About AXMEDIS License Editor …	Title bar: AXMEDIS i-icon DRM License Editor. v2.1.9
AXMEDIS Tools menu	**AXMEDIS DRM Viewer**
Message during start	AXMEDIS
Title bar	AXMEDIS License Viewer
License Window	AXMEDIS DRM Editor and Viewer
Help –> About AXMEDIS License Viewer …	Title bar: AXMEDIS i-icon DRM License Viewer. v2.1.9
AXMEDIS Tools menu	**AXMEDIS Editor**
Message during start	AXMEDIS Editor
Title bar	AXMEDIS Editor
Help –> About … SHIFT+F1	Title bar: About i-icon AXMEDIS Editor version: 0.4 date: Jun 22 2006 http://www.axmedis.org
AXMEDIS Tools menu	**AXMEDIS Player**
Message during start	AXMEDIS Player
Title bar	AXMEDIS Player
	No “Help → About” command
AXMEDIS Tools menu	**AXMEDIS Prog&Pub Editor**
Message during start	AXMEDIS Programme Editor v. 1.0
Title bar	AXMEDIS Programme and Publication Editor 1.0
Help –> About … SHIFT+F1	Title bar: About Programmes and Publication i-icon This is about AXMEDIS Programme and Publication Editor

	http://www.axmedis.org
AXMEDIS Tools menu	**AXMEDIS Prog&Pub Engine**
Title bar	AXMEDIS Prog&Pub Engine
AXMEDIS Tools menu	**AXMEDIS Prog&Pub Monitor**
Message during start	AXMEDIS AxPnP Engine v.1.0
Title bar	AXMEDIS Programme and Publication Engine Monitor v.1.0
Help –> About …	Title bar: P&P Engine Monitor i-icon AXMEDIS Programme & Publication Engine Monitor v.1.0 http://www.axmedis.org
AXMEDIS Tools menu	**AXMEDIS Rule Scheduler**
Message during start	AXMEDIS
Title bar	AXMEDIS Rule Scheduler
? –> About F2	Title bar: Error Not implemented yet.

Appendix 3:
Icons, Commands, and Shortcuts used in the AXMEDIS Tools

AXMEDIS DRM Editor



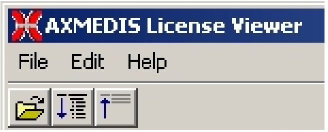
Icon	File	Shortcut
New	New	
Open	Open from file ...	
	Open from server ...	
	Save to file ...	
Save	Save to server ...	
	Close	
	Exit	

Icon	Edit	Shortcut
	Search	

Icon	Help	Shortcut
	About AXMEDIS Licence Editor	

Icon		
ExpandAll		
CollapseAll		
Creates a new grant		
Deletes this grant		
Adds a new condition to the grant		
Deletes a condition of the grant		

AXMEDIS DRM Viewer



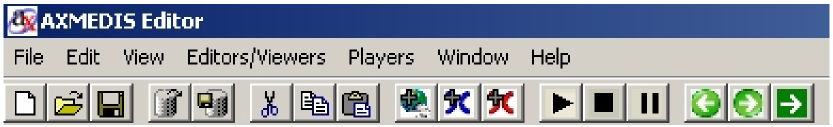
Icon	File	Shortcut
Open	Open from file ...	
	Open from server ...	
	Close	
	Exit	

Icon	Edit	Shortcut
	Search	

Icon	Help	Shortcut
	About AXMEDIS Licence Viewer	

Icon		
ExpandAll		
CollapseAll		

AXMEDIS Editor



Icon	File	Shortcut
New	New	Ctrl + N
Open	Open	Ctrl + O
	Close	Alt + C
Save	Save	Ctrl + S
	Save as	Ctrl + A
	Notify workflow activity completion …	
Open from Database	Open from Database ..	
Upload into Database	Upload into Database ...	
	Configuration ...	
	Plugins	
	Recent files ...	
	Exit	Ctrl + E

Icon	Edit	Shortcut
	Undo	
	Redo	
Cut	Cut	Ctrl + X
Copy	Copy	Ctrl + C
Paste	Paste	Ctrl + V
	Delete	
	Content processing plugins	

Icon	View	Shortcut
	Load	Ctrl + L
	Fit	Alt + F
	Zoom in	Ctrl + I
	Zoom out	Alt + U
	Zoom --> Select	Ctrl + S
	Autofit	Alt + A
	Fullscreen	Ctrl + F
	Start time	
	End time	
	Set background color	
	Show controls --> Show status	Alt + T
	Show controls --> Show current time	
	Show controls --> Show play/stop buttons	Alt + P
	Show controls --> Show slider	Alt + S
	Show controls --> Show zoom buttons	Alt + Z

	Player Viewers --> Video player	
	Player Viewers --> SMIL player	
	Player Viewers --> MPEG4 player	
	Player Viewers --> Audio player	
	Player Viewers --> Image viewer	
	Player Viewers --> Document viewer	
	Player Viewers --> None	

Icon	**Editors/Viewers**	**Shortcut**
	Resource Editor Viewer	Ctrl + R
	Metadata Editor Viewer	Ctrl + M
	Annotation Editor Viewer	Alt + A
	Visual Editor Viewer	
	Behaviour Editor Viewer	
	DRM Editor Viewer	Ctrl + D
	Workflow Editor Viewer	Ctrl + W
	Protection Editor	
	Show all panel	

Icon	**Players**	**Shortcut**
▶ Play	Start	
⏸ Pause	Pause	
■ Stop	Stop	

Icon	**Window**	**Shortcut**
	AXMEDIS Editor	

Icon	**Help**	**Shortcut**
	Tool certification	
	Guide to AXMEDIS Editor …	
	About AXMEDIS Editor …	

← Go Back		
→ Go Forward		
→ Go to		

Icon	**Context Menu (AXMEDIS Root Object)**	**Shortcut**
	Open	
	Open with ... (Behviour View, Visual View etc)	
	Expand Item	
	Properties ...	
	Update from database	
	Add --> Generic Metadata from XML, MPEG21 etc.	
	Add --> Dublin Core	
	Add --> AXMEDIS Info	
✚ Add embedded resource	Add --> Embedded Resource	
	Add --> Referred Resource	

	Add --> New Object	
	Add --> New Object with Resource	
Add embedded object	Add --> Embedded Object	
Add object reference	Add --> Referred Object	
	Request definitive AXOID ...	
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	Delete
	Move up	Ctrl + Up
	Move down	Ctrl + Down
	Recreate tree	

Icon	Context Menu (AXMEDIS non Root Object)	Shortcut
	Open	
	Open with ... (selected view)	
	Expand Item	
	Properties ...	
	Update from database	
	Add --> Generic Metadata from XML ...	
	Add --> Dublin Core	
	Add --> AXMEDIS Info	
Add embedded resource	Add --> Embedded Resource	
	Add --> Referred Resource	
	Add --> New Object	
	Add --> New Object with Resource	
Add embedded object	Add --> Embedded Object	
Add object reference	Add --> Referred Object	
	Insert --> Embedded Resource	
	Insert --> Referred Resource	
	Insert --> New Object	
	Insert --> New Object with Resource	
	Insert --> Embedded Object	
	Insert --> Referred Object	
	Request definitive AXOID ...	
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	Delete
	Move up	Ctrl + Up
	Move down	Ctrl + Down
	Recreate tree	n

Icon	Context Menu (AXMEDIS Info/Dublin Core)	Shortcut
	Open	
	Open with → Metadata View	
	Properties ...	

	Insert --> Generic Metadata from XML ...	
	Insert --> Dublin Core	
	Insert --> AXMEDIS Info	
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	Delete
	Move up	Ctrl + Up
	Move down	Ctrl + Down
	Recreate tree	

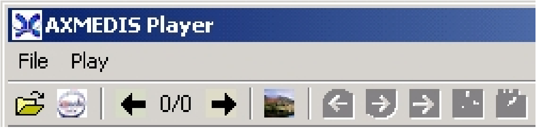
Icon	**Context Menu (Resource)**	**Shortcut**
	Open	
	Open with → Resource View	
	Properties ...	
	Extract Resource	
	Insert --> Embedded Resource	
	Insert --> Referred Resource	
	Insert --> New Object	
	Insert --> New Object with Resource	
	Insert --> Embedded Object	
	Insert --> Referred Object	
	Content processing plugins ...	
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	Delete
	Move up	Ctrl + Up
	Move down	Ctrl + Down
	Recreate tree	

Icon	**Context Menu (MPEG12 Root Object)**	**Shortcut**
	Open	
	Recreate tree	
	Expand tree	
	Properties ...	Ctrl + P
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	
	Move up	
	Move down	
	Add --> Descriptor	SHIFT + D
	Add --> Descriptor Reference	
	Add --> Item	SHIFT + I
	Add --> Item Reference	
	Add --> Component	SHIFT + C
	Add --> Component Reference	
	Add --> Component with resource embedded	
	Add --> Component with resource referred	

Icon	Context Menu (MPEG12 Descriptor)	Shortcut
	Open	
	Recreate tree	
	Expand tree	
	Properties ...	Ctrl + P
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	
	Move up	
	Move down	
	Add --> Descriptor	
	Add --> Descriptor Reference	SHIFT + D
	Add --> Component	
	Add --> Component Reference	SHIFT + C
	Add --> Component Reference	
	Add --> Component with resource embedded	
	Add --> Component with resource referred	
	Add --> Statement	SHIFT + S
	Insert --> Descriptor	ALT + D
	Insert --> Descriptor Reference	

Icon	Context Menu (MPEG12 Component)	Shortcut
	Open	
	Recreate tree	
	Expand item	
	Properties ...	Ctrl + P
	Cut	Ctrl + X
	Copy	Ctrl + C
	Paste	Ctrl + V
	Delete	
	Move up	
	Move down	
	Add --> Descriptor	SHIFT + D
	Add --> Descriptor Reference	
	Add --> Referred Resource	SHIFT + R
	Add --> Embedded Resource	
	Insert --> Descriptor	ALT + D
	Insert --> Descriptor Reference	
	Insert --> Item	ALT + I
	Insert --> Item Reference	
	Insert --> Component	ALT + C
	Insert --> Component Reference	
	Insert --> Component with resource embedded	
	Insert --> Component with resource referred	

AXMEDIS Player



Icon	File	Shortcut
📂 Open file	Open	
Show Axmedis structure	Show AXMEDIS structure	
🖼 shows the image every time seconds	Sliding Show	
→ Next resource	Next resource	
← Previous resource	Previous resource	
	Configuration ...	
	Recent files ...	
	Close	

Icon	Play	Shortcut
	Play	
	Pause	
	Stop	

Icon	View	Shortcut
	Load ..	Ctrl + L
	Fullscreen	Ctrl + F
	Set background color	
	Show controls → Show status	Alt + T
	Player Viewers → Video, SMIL, MPEG4, Audio, etc	

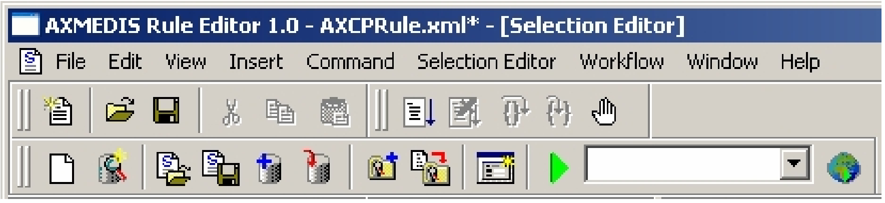
Icon		
←		
→		
Search icon		
Home icon		

Icon	Context menu (left window)	Shortcut
	Open	
	Open with ... (selected Viewers)	
	Expand item	

Icon	Context menu (right window: Image, Movie)	Shortcut
	Load	Ctrl + L
	Fit	Alt + F
	Zoom in	Ctrl + I

	Zoom out	Alt + U
	Zoom --> 25 - 300%	
	Zoom --> Select	Ctrl + S
	Autofit	Alt + A
	Fullscreen	Ctrl + F
	Set background color	
	Return region mode	
	Save region ...	
	Copy region	
	Show status	Alt + T
	Show zoom buttons	Alt + Z
	Video player	
	Audio player	
	Image viewer	
	Document viewer	
	None	

AXMEDIS Cont Proc Rule Editor


AXMEDIS Rule Editor 1.0 - AXCPRule.xml* - [Selection Editor]
File
Edit
View
Insert
Command
Selection Editor
Workflow
Window
Help

Icon	**File**	**Shortcut**
New Rule	New	Ctrl + N
Open Rule	Open	Ctrl + O
	Close	
Save Rule	Save	Ctrl + S
	Save as	
	Import Jscript	Ctrl + I
	Export Jscript	Ctrl + E
	Properties	Ctrl + T
	Page Setup	
	Print Preview	Ctrl + SHIFT + P
	Print	Ctrl + P
	Recent files ...	
	Exit	Alt - X

Icon	**Edit**	**Shortcut**
	Undo	Ctrl + Z
	Redo	Ctrl + SHIFT + Z
Cut	Cut	Ctrl + X
Copy	Copy	Ctrl + C
Paste	Paste	Ctrl + V
	Delete	DEL
	Find	Ctrl + F
	Find Next	F3
	Replace	Ctrl + H
	Replace Again	Ctrl + SHIFT + F4
	Match brace	Ctrl + M
	Goto	Ctrl + G
	Advanced --> Indent increase	TAB
	Advanced --> Indent reduce	BKSP
	Advanced --> Overwrite mode	INS
	Advanced --> Wrap mode	Crtl + U
	Advanced --> Show line endings	
	Advanced --> Show indent guides	
	Advanced --> Show line numbers	
	Advanced --> Show long line marker	
	Advanced --> Show whitespace	
	Select all	Crtl + A

	Select line	Ctrl + L

Icon	**View**	**Shortcut**
	Workspace	Alt + 0
	Output	Alt + 1
	Debug Monitor	Alt + 2
	Preferences	Alt + F7

Icon	**Insert**	**Shortcut**
	Selection	Alt + S
	Parameter	Alt + P
	Jscript	Alt + J
	Dependency → Tool	Alt + T
	Dependency → Script	Alt + I

Icon	**Command**	**Shortcut**
	Activate	Alt + A
	Install	Alt + L
	Find Rule	Alt + N
	Rule List	Alt + R
	Check	Alt + C
Start Debug	Debug --> Go	F5
Stop Debug	Debug --> Stop	
Step Over	Debug --> Step over	
Trace Into	Debug --> Trace Into	
Insert Breakpoint	Breakpoints --> Insert/Remove	F9
	Breakpoints --> View List	Alt + W

Icon	**Selection Editor**	**Shortcut**
Clear Selection	Clear All	
Open Selection	Open ...	
Save Selection	Save ...	
Open from Database	Open from Database ..	
Upload into Database	Upload into Database ...	
	Submit	
Add new Query	Query --> New	
Import Query	Query --> Import ...	
Export Query	Query --> Export ...	
Customize Query Panel	Query --> Add Custom ...	
Test Query	Query → Test	
Test selection		

Icon	Workflow	Shortcut
	Message List	
	Notify Activity Completion	

Icon	Window	Shortcut
	Cascade	
	Tile Horizontally	
	TileVertically	
	Arrange Icons	
	Next	
	Previous	
	Close All	
	Jscript Editor - JSScript	
	XML Editor - selection	
	Selection Editor	

Icon	Help	
	About ...	

Icon	Context Menu (Root)	
	Activate	
	Save	
	Save As	
	About ... (Help)	

Icon	Context Menu (Header/Schedule)	Shortcut
	Open	
	About ... (Help)	

Icon	Context Menu (Definition)	Shortcut
	Insert Script	
	About ... (Help)	

Icon	Context Menu (Dependency)	Shortcut
	Insert Dependency → Tool	
	Insert Dependency → Script	
	Clear	
	About ... (Help)	

Icon	Context Menu (Arguments)	Shortcut
	Insert → Selection	
	Insert → Parameter	
	Clear	
	About ... (Help)	

Icon	Context Menu (key)	Shortcut
	Edit	
	Rename	
	Delete	
	Move Up	

	Move Down	
	About ... (Help)	

Icon	Context Menu (selection)	Shortcut
	Edit	
	View XML	
	Rename	
	Delete	
	Move Up	
	Move Down	
	About ... (Help)	

Icon	Context Menu (jscript)	Shortcut
	Open	
	Debug	
	Check	
	Delete	
	About ... (Help)	

Icon	Context Menu (Selection Editor / root selection)	Shortcut
	Save	
	Clear All	
	New Query	
	Import Query ..	
	Delete	
	Tree View --> Expand Element Items	
	Tree View --> Collapse Element Items	
	Tree View --> Expand Tree	
	Tree View --> Collapse Tree	

Icon	Context Menu (Selection Editor / selection)	Shortcut
	Save	
	Clear All	
	New Query	
	Import Query ..	
	Edit	
	Delete	
	Tree View --> Expand Element Items	
	Tree View --> Collapse Element Items	
	Tree View --> Expand Tree	
	Tree View --> Collapse Tree	

AXMEDIS Cont Proc Rule Scheduler

Icon	Programm	Shortcut
	Add rule	Ctrl - A
	Launch Scheduler	Ctrl - L
	Stop Scheduler	Ctrl - S
	Backup	Alt - B
	Restore	Alt - R
	Minimize	Alt - M
	Exit	Alt - X
	Start Grid Peer functions	Ctrl - F
	test replace parameters	
	test replace schedule	
	test install whole rule as string	
	test replace parameters and schedule	

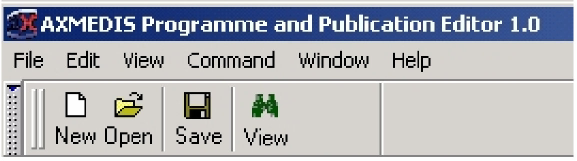
Icon	Settings	Shortcut
	Preferences	Ctrl - N

Icon	View	Shortcut
	Refresh	Alt - E
	Arrange --> Top	Alt - T
	Arrange --> Bottom	Alt - B
	Arrange --> Vertical	Ctrl - V
	Arrange --> Horizontal	Ctrl - H
	Rule Properties	Ctrl - R
	Executor Profile	Ctrl - E
	Logs ...	Alt - L

Icon	Commands	Shortcut
	Enable Rule	Alt - N
	Disable Rule	Alt - D
	Kill Rule	Alt - K
	Pause Rule	Ctrl - P
	Resume Rule	Ctrl - S
	Suspend Rule	Alt - U
	Remove Rule	Alt - M

Icon	?	Shortcut
	Help	F1
	About	F2

AXMEDIS Prog and Pub Editor



Icon	File	Shortcut
New	New	Ctrl + N
Open	Open	Ctrl + O
	Close	
	Close All	
	View Programme	Ctrl + V
Save	Save	Ctrl + S
	Configuration ...	
	Print	Ctrl + P
	Recent files ...	
	PnP Programme Repository ...	
	Exit	Ctrl + X

Icon	Edit	Shortcut
	Insert Selection	
	Delete Selection	

Icon	View	Shortcut
	Toolbar	Alt + 0
	Workspace	Alt + 1
	Output	Alt + 2

Icon	Command	Shortcut
	Activate programme	Alt + A
	Programme test --> Quick trial	F5
	Programme test --> Full trial	SHIFT + F5
	Stop programme	Alt + S

Icon	Window	Shortcut
	Cascade	
	Tile Horizontally	
	TileVertically	
	Arrange Icons	
	Next	
	Previous	

Icon	Help	Shortcut
	About ...	SHIFT + F1

Icon	Content Menu (Axmedis PnP Programmes)	Shortcut
New	New	
Open	Open	
	Close all	

	About	

Icon	Content Menu (selected programme)	Shortcut
View	View	
	Add selection	
	Close	
	About	

Icon	Context Menu (Header/Schedule/Distribution)	Shortcut
View	View	
	About	

Icon	Context Menu (Definition)	Shortcut
View	View	
	Add selection	
	About	

Icon	Context Menu (Selection)	Shortcut
View	View	
	Delete selection	
	About	

AXMEDIS Prog and Pub Monitor

Icon	Action	Shortcut
	Connect	
	Disconnect	
	Save Programme(s)	
	Quit	

Icon	Help	Shortcut
	Index	
	About ...	