

A Review of the Role of Document Engineers in Information Systems Development

José-Antonio Gutiérrez¹, José-Ramón Hilerá¹, Eva García¹, Antonio García¹, Luis de-Marcos¹, and José-Javier Martínez¹

¹ Computer Science Department, University of Alcalá, Alcalá de Henares, Madrid, Spain

Abstract - *The formation of a new type of specialist in document management has been put forward, who is able, among other functions, to participate in Information Systems development projects, with the assignment of carrying out activities related to the planning, design, indexing, retrieval and distribution of the documents used. The term "Document Engineer" is defined as a new type of documentation expert, who will have great impact in the future.*

Keywords: Software Engineering, Document Engineering, Information System, Software Documentation.

1 Introduction

The documentation generated during Information System development is the integrating element of the different activities and participants which are involved, in one way or another, throughout the life-cycle of the system and, however, it is an aspect to which traditionally very little attention has been paid. This fact, as Sommerville [21] affirms, has been precisely what was held up the maintenance of Information Systems throughout the entire history of computing.

The growing concern over this subject has led organizations like ISO and ANSI, to justify those documents management; it must be considered as a process among those which make up the software life-cycle. In the case of ISO, in its norm 12207 [13] the process of documentation forms part of the designated "support processes" group, together with others like those of the configuration management, the quality assurance or the auditing. As for ANSI, in the standard 1074 [1] it is considered to be one of the four "integral processes" (the others are "verification and validation", "software configuration management" and "training") necessary to successfully complete all the activities in a development project, guaranteeing an adequate level of quality. In the norm 9001 about quality systems [12] and in its corresponding application guide in the case of software development [11], ISO also establishes the need to see the documentation control as one of the necessary procedures for software quality assurance.

To carry out all the documentary activities required by these and other standards, we need the participation of

documentation experts who can take charge of implantation, in software development organizations, of Document Management Integrated Systems, which, besides the generation and editing of documents, backs up other functions related to the planning, design, indexing, retrieval and distribution of all the documents involved in the development process. In the following sections, we will establish the professional profile and assignment of those specialists in documentation who must collaborate with software designers to finally obtain Information Systems with an adequate level of quality.

2 The Professional Profile of Document Engineer

The Document Engineer, who forms part of an Information System development project, should be a person with a solid training in the field of Documentation Sciences, as well as possessing knowledge of general documentary needs, which may come up during the project. The subject in matter, which a professional person of this type should master, is related to: computer science, library science, documentary analysis, documentary languages, archival management, information retrieval, planning and evaluation of documentation systems, workflow and hypermedia. The convenience of knowledge of these disciplines and their possible application in the field of Information Systems development is justified as follows.

- *Library Science:* The techniques and procedures used in the field of librarianship can be perfectly applicable in the case of software development organizations, when the size of the handled documents is great, establishing all the elements of the documentation chain and so facilitating the location of documents need to carry out different tasks within the organization.
- *Archival Management:* Knowledge of the basics of this discipline allows the Document Engineer to apply adequate methods and techniques in order to obtain a correct conservation and diffusion of documentation resources in the Information Systems development process.

- *Documentary Analysis:* The objective of Documentary Analysis is to abbreviate the representation of the document's form and contents, so as to facilitate, in the future, their search and retrieval. In the particular case of documentary analysis of the contents, this representation receives the name of *index*, existing thereby standardized methods of selecting from a documentary language, usually of the thesaurus type, the indexing terms that make up the index of a document [10]. In the case of software development, the documentary analysis should be one of the main functions of the Document Engineer, as the efficiency of the searches which are carried out throughout the project, and which can greatly influence its correct development, depends on the correct elaboration of the indexes.
 - *Documentary Languages:* Documentary Languages, especially the thesaurus of descriptors [9], are those that give the terms which can be associated with the documents during the analysis to obtain a precise representation of the contents of the document. There have been a great number of these types of languages which are specific for the contents of universe of discourse to which the documents (medicine, economy, etc.) refer. The Document Engineer should know how to elaborate a documentary language for any field of application, since he should be familiar with the terms adapted to software development needs and so he may be able to carry out the analysis of each one of the generated documents.
 - *Planning and Evaluation of Documentation Systems:* Another subject that should be present in the Document Engineer's curriculum is a reference to the way of planning and evaluation of the impact of the implantation of documentation units and systems in the organization. The professional person who becomes a member of a software development project needs to know these subjects as he must participate in the elaboration of the Project Development Plan, which concerns the planning of all the related aspects, both with the documentation which must be generated during the selfsame project and with that already in existence and which must be used in the said project.
 - *Document Management Systems:* A Document Engineer should be familiar with the use of software tools such as Document Management Systems (or Document-based Information Systems), which allow the automated control of documents throughout its complete life-cycle in an organization, from its initial creation to its final filing, thus, offering for this function mechanisms of edition, identification, storage, follow up, retrieval and presentation of documents. Since most documents used by the information system development organizations are electronic, the convenience of incorporating tools of this type to facilitate their management is evident.
 - *Workflow:* In organizations, especially in software development, people work together within a collective framework of communication, collaboration and coordination. The knowledge, on the part of the Document Engineer, of design methods and techniques of workflow models that take into account these three aspects is fundamental in order to exercise control over the documents that circulate within the organization. This professional person must be aware of recommendations, regarding this task, established by international institutions like Workflow Management Coalition [16], and by national institutions (in Spain, the norm ESTROFA published by the Ministry for Public Administrations [7]). He must also be familiar with and know how to use Document Management Systems which incorporate mechanisms of this type, usually under the epigraph of *groupware* [19].
 - *Hypermedia:* A Document Engineer should have adequate training in multimedia an hypermedia technology, not only regarding the edition of these types of documents, but also the application of modeling techniques which permits the carrying out of a design before the edition, which contemplates characteristics such as synchronization of multimedia contents, the possibilities of hypermedia navigation and the user-interface aspect [20]. This will allow him to apply it to the case of the documentation that must be carried out throughout a software project.
 - *Computer Science:* It is evident that the aforementioned subjects presuppose a training in the field of computer science, since most of those disciplines make use of computer technologies; for example, the use of Document Management Systems requires a certain knowledge of the computing medium upon which operations are carried out (computer, operating system) and, if *groupware* functions are incorporated, a knowledge of computer communications (electronic mail, Internet, Intranet, etc.). In the case of the Document Engineers who join an Information System development project, they should also have rudiments of Software Engineering and they should be familiar with the development methodology used in the organization, including the types of documents required by this.
- Regarding on the curriculum of a professional person in Documentation Sciences, if there is a special prominence in these disciplines, this person will have an adequate profile to be part of a Information System development team in any organization and he will be able to carry out the corresponding functions within software projects, which will be described in the following section.

3 Functions of a Document Engineer in Software Projects

The incorporation of documentation experts in Information System development projects will allow us to improve the fulfillment of the requirements demanded by the norms, standards and methodology on software quality and processes applicable to the project. For this, the professional person will be assigned a group of documentary tasks related to: the planning and design prior to the production of documents; the creation or selection of documentary language that are most adequate for the project (for example thesaurus); the analysis of documents produced, by means of an indexing process, to assign the terms from the documentary language which will enable us to represent their contents; and the selection of search language used in the exploitation of the documents during the software development and subsequent maintenance. The objective of these activities is briefly described as follows:

3.1 Documentation Planning

One of the functions that a Document Engineer should manage is the study of the documentation needs for the software project which will be developed. The result of this study should be gathered in a "Documentation Plan" which establishes, among other things, the type of document to be elaborated, the technology to be used and the sequence of operations for the production and distribution of the documents. Moreover, for the construction of the documentary language, which will be used during the indexing to facilitate the searches, is carried out during the Information System development and maintenance. To elaborate this plan, at least the following task must be carried out within the project:

- To identify the documentation needs, which means the carrying out of an analysis of the different types of documents that must be produced and will normally come with the methodology used in the Information System development?
- To establish the level of appropriate detail for the documents according to the type of project as well as the interrelation of each document to the others.
- To plan the storage of the documents, envisaging its organization in high level structures (folders, document databases, etc.) whose arrangement follows a fixed set criteria.
- To work out a sequence of operations for the production of documents and their distribution for audience approval.

- If a documentary language is found, it is necessary to elaborate it in order to facilitate the searches during the exploitation and maintenance of the documents, such language must undergo a process of construction.
- To describe, according to economic criteria, the most appropriate alternative technology for the implementation of the documentation activities in the project.

To carry out these activities, the Document Engineer must use techniques related to: planning; cost estimation; and specification of the workflow involved in the production and distribution of documents, as they are normally used in the field of *groupware* [14].

3.2 Documentation Design

When designing the characteristics of the documents to be produced within a software project, the Document Engineer must also intervene by means of a description of all the necessary aspects for the document's construction. In this sense, the Document Engineer must undertake the following functions:

- To work out a structure which allows us to identify the parts of the documents and their relationship and simplify their maintenance.
- Define the dynamics or life-cycle of the documents that complement the former static dimension in order to obtain a complete description of software development project documentation.
- As an option, to add hypermedia specifications in the documentation design to show the possibilities of navigation through hyper-links.
- To establish the characteristics of document interface with the audience, bearing in mind its determinant influence over the understanding of its contents.

All these aspects of design must be gathered in different conceptual models elaborated by means of adequate techniques. A focus on object orientation can be adopted by adapting some of the existing modeling methods in the field of Software Engineering (for example UML [5]) to the particular case of document objects. For hypermedia documentation and user interface design we can also apply object oriented techniques like EORM [15] or ADV [26] respectively.

3.3 Documentary Language Elaboration

One of the exclusive functions of the Document Engineer is the elaboration of a language which gathers the terminology used both in the context of the project (conditioned, above all,

by the development methodology that is followed) and in the application domain of the Information System which will be developed. This language must allow us to represent, by means of indexes, the knowledge contained in the generated documents, to facilitate the carrying out of searches, first during the execution of the project, and afterwards during software maintenance.

Because it is a very precise type of documentary language which offers many facilities in the future formulation of search enquiries, the use of thesaurus would be very appropriate [9]. This construction involves the establishment of terms (descriptors, non-descriptors, compound descriptors, groups of descriptors) and semantic relationships among terms extracted from thematic domain, dealt with the documentation software project. The Document Engineer in charge of the elaboration of the thesaurus must collaborate with the Software Engineer, receiving proposals of inclusion of new terms from them, depending on how the execution of the project progresses.

3.4 Indexing the documentation

To facilitate the work of a person involved in an Information System development, who constantly need to consult documents formerly elaborated within the project (for example, a programmer will consult design documents, whereas a designer those pertaining to analysis, etc.), the Document Engineer must carry out a documentary analysis of each documents produced, identifying the indexing terms, which will constitute a reduced model (index); this then will represent its informative contents and will facilitate search of documents in the future. Documentation indexing of a project will simplify its maintenance since whenever changes will be included in any document; it will be possible to work out with others which will be affected and will be able to evaluate the consequences that the changes suppose.

If he possesses the adequate technology, besides assigning terms extracted from documentary language (for example, thesaurus) whose elaboration, as indicated in the former section, is also his task. The Document Engineer must also be able to put into effect an associative indexing that allows hypermedia navigation by means of establishing hyperlinks among the different contents of the documents. To facilitate the maintenance of these links especially with a large amount, he must use some modeling techniques which proceed from the link structure established for the associative indexing, which gathers all the possibilities for navigation through the project documents. There are a few techniques of this type the Document Engineer should know, they are normally based on the theory of graphs, as the designated OCPN (Object Composition Petri Net) [18] or the well-known IDM (Interactive Dynamic Map) [27].

3.5 Document Control

The control of project documents is another of the Document Engineer functions, both the documentation quality control and its final distribution and filing control. In the first case, he must carry out a control of the different factors that directly affect the quality of each of the documents, like the structure determined by its contents, the format and the style of writing. To do this, he must consider the documentation norms and standards previously established in the organization and the recommendation of experts [3], [4], [17].

The distribution and filing of the documents must be controlled in order to assure there is complete fulfillment of those which have been established in the project's "Documentation Plan", regarding the way in which each of the generated documents must circulate, and regarding policy of storage and custody of the original copies and the management of documents elaboration.

As to distribution control, this can be greatly simplified using a workflow system which allows us to design and to carry out models of automated distribution, which can guarantee it, when any project participant will execute a task can automatically receive the documents he needs [14].

4 Automation of Documentation Systems Analysis Requirements

Once considered the effort to be accomplished in software documentation, we are going to consider how recently, at software industry, new scenarios have emerged for automatic documentation on the various phases of product analysis. In this sense, within the requirements analysis, known as the phase where you will discover the processes, customer's needs are specified, negotiated and validated; and despite being billed as "Engineering", there is not much of a documentation management methodology. In other words, the method, which sets out the stages of documentation, is neither clearly established, nor the accurate documentation of the types of requirements. In this sense there are two projects which can help us: the first project is called SDML (*Scenario Description Markup Language*) which proposes not only the use of this language to define scenarios, but also an entire process of scenario development (SMDP), carried out with a modeling environment (SME) based on SDML. The second one, called REM (*Requirement Engineering Methodology*), is strictly a project which manages requirements, comprising by the definition of a new language and an editing tool.

System SDML has a XML language to define the scenarios (ordered set of peer interaction -system and actors-), which is the basis for a process of developing scenario models (SMDP), and it has a number of labels among which highlight:

- <actor>, person or system with a role on the scenario.
- <item>, physical item or information should be available on the scenario.

- <action>, action taken by an actor.
- <interaction>, tuple [<actor>,<action>,<item>] which describes an interaction in the form of action performed by an actor with an item.
- <goal>, expected result in a scenario, after the interactions.
- <trigger>, initial interaction that triggers all subsequent interaction flow.
- <precondition>, prerequisite on the scenario.
- <postcondition>, consequence it must occur as an effect of the implementation stage.

It is probable that in the near future SDML will be nominated as a standard language for describing scenarios, since the possibilities of this language seem limitless. And in the same way, the authors extend the idea to other areas of software life cycle, such as test cases or state diagrams; it will probably end resulting in the generation of extensions to treat the vast majority of areas encompassed in the analysis phase within the software life cycle [2].

The other view is obtained using the tool REM [6]. It is a visual environment to define and handle any requirements of a software project, which allows direct export to XML. The tool allows the manipulation of the requirements of projects with such information and generates XML files, and when combined with XSL sheets it generates HTML pages. That can be considered as the final documentation, based on a WYSIWYG (*What you see is what you get*) philosophy, and can be distributed via the Internet, facilitating group work although its members are geographically dispersed.

The separation of information from its representation through the combined use of XML and XSL allows, externally to the tool, to represent the same information in different ways, facilitating the adaptation of document formats to the needs of different users.

Internally, the architecture of the tool follows the document-view model imposed by the application framework, based on a document which automatically generates application generator MS Visual C++ version 6.0, which was used for the environment implementation.

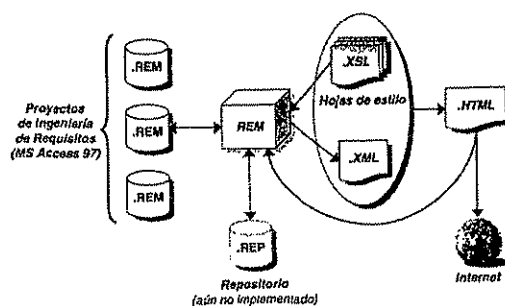


Figure 1: Architecture of REM tool (Durán, 2000 [6])

Inside the REM tool is displayed the XML file with a graphical environment which allows for user editing to work with it comfortably. If the XSD schema defines this as the standard language, portability would be achieved between CASE tools for requirements management, as proposed by SDML.

5 Automatic Documentation Software Design

Moreover, under the heading of software design documentation, scenarios are being considered: MDA, MOF and XMI.

The model-based architecture (MDA - *Model Driven Architecture*) is a proposal by the OMG consortium to separate the business logic and application of the technology platform, which supports (OMG, 2003 [23]). With MDA, it is possible to make an application model or models of behavior and functionality of systems; theoretically on any platform, public or private, including Web Services, NET, CORBA and J2EE. These independent platform models document, the business logic and functionality regardless of the specific technology that implements it.

The aim of MDA is to enable modeling of data and computer-readable application, which will allow greater flexibility in the areas of implementation, integration, maintenance and testing and simulation.

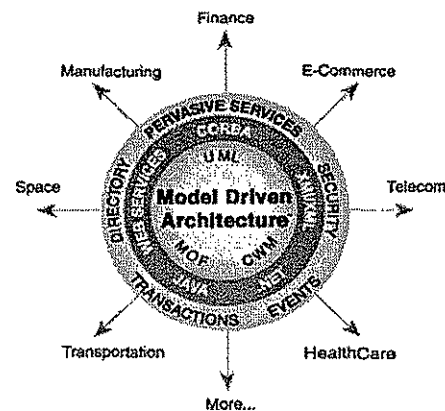


Figure 2: MDA composition.

The consortium of computer industry standards for the specification of OMG has defined a set of specifications covering the process of standardizing the documentation of a software project.

MOF (*Meta-Object Facility*) is an integration framework for defining extensible, handling and integrating data and metadata in an independent way of the platform used. MOF-based standards are used in integrating tools, applications and

data (OMG, 2006). This technology provides a model repository which may be used for specifying and manipulating models, promoting consistency in the handling of models in all phases of use of MDA (OMG, 2002 [22]).

The meaning of metadata is literally data that represents and defines properties for raw data. Metadata was the term originally applied only to data models (e.g. database schemas in which the metadata is the name of the columns and tables). Currently, in the MDA world are included as metadata UML models, workflow models, business process models, business rules, service definitions, configuration and deployment descriptors, and so on.

The MOF standard specifies a language for modeling metadata. A model of a metadata type is called a metamodel, so it could recognize the language as a language of MOF metamodel. The UML language is a subset of the MOF metamodel, so in practice we use UML modeling tools to create metamodels that conform to MOF (Frankel, 2006).

XMI (*XML Metadata Interchange*) is an integration framework for defining, exchanging XML data and objects. XMI-based standards are used for integrating tools, repositories, applications and data warehouses. XMI provides rules by which a schema can be generated for each valid metamodel based on MOF and XMI communicable (OMG, 2006 [25]). XMI provides a mapping of MOF to XML. As long as MOF and XML technologies have been developed, XMI mapping has been updated in order to comply with the latest versions of those specifications (OMG, 2006 [25]).

In practice, XMI is the language that describes portable UML models between design tools; however, it doesn't specify yet how to record graphical information, such as colors, sizes or orientations on the graph, because each tool does it itself.

In systems that are currently developed, the applications are so large that they no longer develop stand-alone systems; applications are developed from existing components and use common services typically used in the implementation. Since the networks, especially the Internet, facilitate interoperability, applications today need to interact with other applications and share information with them.

Through the modeling, architects and designers can better tackle the complexity of software systems. The model allows the separation of design and implementation. Implementation details are not captured, nor the semantics of interoperability is identified, nor the data exchange formats, which limits the model. To provide a comprehensive definition of the application, the architect needs to specify operational semantics, constraints and information exchange formats in addition to the abstract design represented by the model.

The three pillars on which underpin the development of software according to the OMG are: UML, MOF and XMI. MOF meta-metamodel is used to completely describe the software architecture (OMG, 2002 [22]). That is, it describes the modeling languages as UML, a set of metamodels such as CCM (*CORBA Component Model*), EJB (*Enterprise JavaBeans*), and many other metamodels. MOF also contains a set of rules that specify the interoperability and exchange format of its meta. MOF-based architecture enables applications to be described from different levels of abstraction.

6 Conclusions

The documentation generated in the Information Systems Development projects will enable us to carry out the correct maintenance of such systems. The development methodologies defined in the field of Software Engineering and the related standards (ISO 12207, ANSI/IEEE 1074, ISO 9001, etc.) do not ignore this fact, demanding the generation of different types of documents and, in some cases, establishing document management procedures.

When the size of the intended documentation is considerable, it is necessary to do it in a disciplined way following a method of engineering susceptible to automation [8]. The idea is to see the documentary process as a work of (document) engineering which must be carried out by professionals with an extensive training in the field of Documentation Sciences and in the area of Information Technology. This professional person is the Document Engineer. In this article we established the general guidelines for the curriculum of these new professionals who, as well as knowing the fundamentals of formerly created documents, must also be capable of producing abstract models of the static and dynamic structure of documents which must be generated.

This last aspect is not normally dealt with documentation studies, taught at universities (in Spain, the Documentation Degree) which are aimed more towards the training of persons who will practice in libraries or documentation services. Subjects related to design modeling and document construction should be added to these studies in order to be designated the title of Document Engineering, from which Information Systems development companies can feed for the incorporation of Document Engineers, who will work together with Software Engineers, carrying out those activities related to documentation, like planning, design, indexing, control, etc. In the article we describe such activities, which justify the presence of these professional people in the development of Information Systems.

7 References

- [1] ANSI/IEEE Std 1074, Standard for Software Life Cycle Processes. IEEE Computer Society, New York, 1991.

- [2] Arias Monge, La documentación en la Ingeniería del Software, tratamiento metodológico y herramientas de automatización. TFC Universidad de Alcalá.
- [3] Ayer, S., Patrinostro, F.: Documenting the Software Development Process. McGraw-Hill, New York, 1992.
- [4] Bell, P., Browning, C., Carliner, S., Nohr, M., Williams, J.: "Documentation". In Encyclopedia of Software Engineering (J. Marciniak ed.). John Wiley & Sons, New York, 1994, 407-419.
- [5] Booch, G., Rumbaugh, J., Jacobson, I.: Unified Modeling Language UML: User Guide. Addison-Wesley, Menlo Park, CA, 1998.
- [6] Durán, A. (2000), Un entorno metodológico de ingeniería de requisitos para sistemas de información, PhD Thesis, Universidad de Sevilla.
- [7] ESTROFA, Especificaciones para el Tratamiento de Flujos Automatizados. Ministerio para las Administraciones Públicas, Madrid, Spain, 1996.
- [8] Hilera, J.R., González, L.A., Gutiérrez, J.A., Martínez, J.M.: "Software Documentation as an Engineering Process". ACM Software Engineering Notes, Vol. 23, no. 5, 1998, 61-64.
- [9] ISO 2788, Documentation. Guidelines for the establishment and development of monolingual thesauri. International Standards Organization, Geneva, Switzerland, 1986.
- [10] ISO 5963, Documentation. Methods for examining documents. Determining their subjects and selecting indexing terms. International Standards Organization, Geneva, Switzerland, 1985.
- [11] ISO 9000-3, Quality management and quality system elements - Guidelines for development, supply and maintenance of software. International Standards Organization, Geneva, Switzerland, 1991.
- [12] ISO 9001, Quality systems - Model for quality assurance in design, development, production, installation and servicing. International Standards Organization, Geneva, Switzerland, 1994.
- [13] ISO/IEC 12207, Software Life-Cycle Processes, International Standards Organization, Geneva, Switzerland, 1995.
- [14] Khoshafian, S., Buckiewicz, M.: Groupware, Workflow and Workgroup Computing. John Wiley & Sons, New York, 1995.
- [15] Lange, D.: An Object-Oriented Design Approach for Developing Hypermedia Information Systems, Research Report RT0112. IBM Tokyo Research Laboratory, Tokyo, Japan, 1995.
- [16] Lawrence, P.: Workflow Handbook. The Workflow Management Coalition / John Wiley & Sons, New York, 1996.
- [17] Lehner, F.: "Quality control in software documentation: Measurement of text comprehensibility". Information and Management, Vol. 25, 1993, 133-146.
- [18] Little, T.D., Ghafoor, A.: "Synchronization and Storage Models for Multimedia Objects". IEEE Journal on Select Areas in Communications, Vol. 85, no. 3, 1990, 413-427.
- [19] Martínez, J.M., Hilera, J.R.: "Los Sistemas de Gestión Documental en el Ámbito del Trabajo Corporativo". Revista General de Información y Documentación, Vol. 7, no. 2, 1997, 237-255.
- [20] Martínez, J.M., Hilera, J.R.: "Modelado de Documentación Multimedia e Hipermedia". Cuadernos de Documentación Multimedia, no. 6-7, 1998.
- [21] Sommerville, I.: Software Engineering. Addison Wesley, Reading, MA, 1992.
- [22] OMG (2002), Meta Object Facility (MOF) Specification v1.4, <http://www.omg.org/docs/formal/02-04-03.pdf>
- [23] OMG (2003), MDA Guide v1.0.1, disp. On internet <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [24] OMG (2005), Software Process Engineering Metamodel Specification, [Internet], Disponible: <http://www.omg.org/docs/formal/05-01-06.pdf>.
- [25] OMG (2006), Catalog of OMG Modelling and Metadata Specifications, [Internet], Disponible: http://www.omg.org/technology/documents/modeling_spec_catalog.htm [09.06.2006].
- [26] Rossi, G., Schwabe, D., Lucena, C., Cowan, D.: "An Object Oriented Model for Designing the Human-Computer Interface of Hypermedia Applications". Proceedings of the Int. Workshop on Hypermedia Design, Montpellier, France, 1-2 June 1995, Springer-Verlag & British Computer Society, London, 1996, 123-143.
- [27] Zizi, M.: "Providing Maps to Support the Early Stage of Hypermedia Systems". Proceedings of the Int. Workshop on Hypermedia Design, Montpellier, France, 1-2 June 1995, Springer-Verlag & British Computer Society, London, 1996, 93-104.