

UNIVERSITY CARLOSIII OF MADRID

HIGH POLYTECHNIC COLLEGE

COMPUTER ENGINEERING



FINAL DEGREE PROJECT

A Web Site Architecture and GUI for UML Models Search

Project Report

AUTOR: GUILLERMO SUÁREZ DE TANGIL
TUTOR: JUAN LLORENS

12th July 2009

[This page has been left white on purpose]

1 Acknowledgments

I would deeply like to write some short but sincere lines in order to thank all the people who helped and support me during this project.

First of all to my family, my friends and my girlfriend for standing me during dear deadlines.

Next I would like to thank Juan Llorens for enlighten me so many things, especially for all those things that they have nothing to do with the project. I cannot find the commensurate words which reflect the great duty he has performed as Erasmus tutor; welcoming us so many times on his house and with his family. I will never forget it and I will never forget Finland, which is somehow the most important part of this project.

After, I would like to thank, in general, to the entire IE lab for being so warm and friendly and, in particular, to Julián Urbano Merino.

And finally I would like to thank to the SeTI group for letting me finish this project with them.

[This page has been left white on purpose]

2 Index

1	Acknowledgments	3
2	Index	5
3	Tables	8
4	Illustrations.....	11
5	Introduction	14
6	State of the Art.....	16
6.1	Study of the Content Management System (CMSs)	16
6.1.1	Web Content Management System	17
6.1.2	Content Management Software.....	17
6.1.3	Comparison	27
6.1.4	Conclusion	39
6.2	Microsoft Web Technologies	40
6.2.1	Microsoft Web Forms.....	40
6.2.2	ASP.NET MVC Framework	40
6.2.3	Microsoft Silverlight	41
6.2.4	Comparison: ASP.NET MVC Framework vs. Web Forms Applications vs. Silverlight 42	
6.3	Object-Relational Mapping in ADO.NET.....	44
6.3.1	Data Sets.....	45
6.3.2	Linq to SQL.....	45
6.3.3	Entity Framework	46
6.3.4	Comparison: Data Set vs. Entity Framework vs. Linq to SQL.....	46
6.4	EXT Framework	48
6.5	UML.....	49
6.5.1	yUML.....	49
7	Firs Approach: DotNetNuke	51
7.1	Architecture	51
7.2	Conceptual Model.....	51
7.3	GUI.....	53
7.4	Implementation.....	54
7.5	Conclusion	54
8	UmlModels Site	55

8.1	System Context	55
8.1.1	Current Technologies	55
8.2	Use Case	57
8.2.1	Unregistered User	57
8.2.2	Registered User	59
8.3	Requirements.....	67
8.3.1	2-UML-00012: UMLModels Site	67
8.4	Architecture	97
8.4.1	Pagination of the Search Results.....	97
8.4.2	Architectural Layer	98
8.4.3	Deployment Infrastructure.....	102
8.5	GUI.....	104
8.5.1	Level Structure: Elements Definition.....	107
8.5.2	Site Map.....	109
8.5.3	Main Page.....	112
8.5.4	Search Result	113
8.5.5	Study of the Distribution of the Valuable Information of a Model in the Search Result page.....	120
8.5.6	Model's Page	122
8.6	Conceptual Model.....	137
8.6.1	An Artifact for UmlModels	137
8.6.2	Model vs. UML Diagram in UmlModels.....	138
8.6.3	Advanced Information model.....	139
8.6.4	Conceptual Model Diagram.....	142
8.7	Detailed Design	146
8.7.1	Server.....	148
8.7.2	Client.....	166
8.7.3	Data Base.....	170
9	Security Issues	176
9.1	Organic Law 15/1999 on the Protection of Personal Data (LOPD)	176
10	Budget.....	179
10.1	Personnel	179
10.2	Hardware	180
10.3	Software.....	180

10.4	Fungible	180
10.5	Indirect Costs	181
10.6	Summary	181
10.7	Final Budget	181
11	Conclusions	182
11.1	Project Management	183
12	Future Lines	184
13	Reports of Meetings.....	186
14	Appendix A: Free and open source software CMS.....	187
15	Appendix B: UmlModels Requirements by Gonzalo Génova (Spanish).....	191
16	Appendix C: Definitions, acronyms and abbreviations	241
17	References.....	244

3 Tables

Table 1: Use Case diagram for Unregistered User: Login.	57
Table 2: Use Case diagram for Unregistered User: Signup.	58
Table 3: Use Case diagram for Unregistered User: Search.	58
Table 4: Use Case diagram for Unregistered User: View Limited Model.....	58
Table 5: Use Case diagram for Unregistered User: View Limited Search Result.	59
Table 6: Use Case diagram for Registered User:	60
Table 7: Use Case diagram for Registered User:	60
Table 8: Use Case diagram for Registered User: View Model.....	60
Table 9: Use Case diagram for Registered User: Punctuate Resource.....	61
Table 10: Use Case diagram for Registered User: Add Version.	61
Table 11: Use Case diagram for Registered User: Add Title.....	61
Table 12: Use Case diagram for Registered User: Add Description.	62
Table 13: Use Case diagram for Registered User: Add Requirement.	62
Table 14: Use Case diagram for Registered User: Tag Model.	62
Table 15: Use Case diagram for Registered User: Print View.	63
Table 16: Use Case diagram for Registered User: Add Job Offer.....	63
Table 17: Use Case diagram for Registered User: Subscribe Job Offer.....	63
Table 18: Use Case diagram for Registered User: Give Offer to Candidate.....	64
Table 19: Use Case diagram for Registered User: Post Comment Forum.....	64
Table 20: Use Case diagram for Registered User: Add Model To Bookmark.....	65
Table 21: Use Case diagram for Registered User: Download File.....	65
Table 22: Use Case diagram for Registered User: Buy Credit.	65
Table 23: Use Case diagram for Registered User: Edit Personal Info.....	66
Table 24: Requirements. General Requirements.....	67
Table 25: Requirements. General Requirements.....	67
Table 26: Requirements. General Requirements.....	68
Table 27: Requirements. General Requirements.....	68
Table 28: Requirements. General Requirements.....	68
Table 29: Requirements. General Requirements.....	69
Table 30: Requirements. General Requirements.....	69
Table 31: Requirements. General Requirements.....	69
Table 32: Requirements. General Requirements.....	70
Table 33: Requirements. General Requirements.....	70
Table 34: Requirements. General Requirements.....	70
Table 35: Requirements. Roles.....	71
Table 36: Requirements. Roles.....	71
Table 37: Requirements. Roles.....	71
Table 38: Requirements. Roles.....	72
Table 39: Requirements. Roles.....	72
Table 40: Requirements. Roles.....	72
Table 41: Requirements. Roles.....	73
Table 42: Requirements. Roles.....	73
Table 43: Requirements. Roles.....	73
Table 44: Requirements. Free Services.	74

Table 45: Requirements. Free Services	74
Table 46: Requirements. Free Services	74
Table 47: Requirements. Free Services	75
Table 48: Requirements. Free Services	75
Table 49: Requirements. Free Services	75
Table 50: Requirements. Free Services	76
Table 51: Requirements. Free Services	76
Table 52: Requirements. Added Value Area	77
Table 53: Requirements. Added Value Area	77
Table 54: Requirements. Added Value Area	77
Table 55: Requirements. Added Value Area	78
Table 56: Requirements. Added Value Area	78
Table 57: Requirements. Added Value Area	78
Table 58: Requirements. Models Site Design.	79
Table 59: Requirements. Site Sections.....	79
Table 60: Requirements. Human Classification Process.	80
Table 61: Requirements. Economical business model.....	80
Table 62: Requirements. Economical business model.....	81
Table 63: Requirements. Economical business model.....	81
Table 64: Requirements. Economical business model.....	82
Table 65: Requirements. Economical business model.....	82
Table 66: Requirements. Economical business model.....	83
Table 67: Requirements. Economical business model.....	83
Table 68: Requirements. Economical business model.....	84
Table 69: Requirements. Economical business model.....	84
Table 70: Requirements. Economical business model.....	85
Table 71: Requirements. Economical business model.....	85
Table 72: Requirements. UmlModel	86
Table 73: Requirements. UmlModel	86
Table 74: Requirements. UmlModel	87
Table 75: Requirements. UmlModel	87
Table 76: Requirements. UmlModel	88
Table 77: Requirements. UmlModel	88
Table 78: Requirements. UmlModel	89
Table 79: Requirements. UmlModel	89
Table 80: Requirements. UmlModel	90
Table 81: Requirements. UmlModel	90
Table 82: Requirements. UmlModel	91
Table 83: Requirements. UmlModel	91
Table 84: Requirements. UmlModel	92
Table 85: Requirements. UmlModel	92
Table 86: Requirements. UmlModel	92
Table 87: Requirements. UmlModel	93
Table 88: Requirements. UmlModel	93
Table 89: Requirements. UmlModel	93

Table 90: Requirements. UmlModel	94
Table 91: Requirements. UmlModel	94
Table 92: Requirements. UmlModel	95
Table 93: Requirements. UmlModel	95
Table 94: Requirements. UmlModel	95
Table 95: Requirements. UmlModel	96
Table 96: Requirements. UmlModel	96
Table 97: Study of the free width per number of words/line. Width Panel: 1098	121
Table 98: Study of the free width per number of words/line. Width Panel: 635.	121
Table 99: Detailed Design: Presentation Logic Tier: Components from the Search package related with components in the Scripts package. Visitor Role.	152
Table 100: Detailed Design: Presentation Logic Tier: Components from the Search package related with components in the Scripts package. Client Role.	153
Table 101: Detailed Design: Presentation Logic Tier: Components from the Uml package related with components in the Scripts package. Visitor Role.	153
Table 102: Detailed Design: Presentation Logic Tier: Components from the Uml package related with components in the Scripts package. Client Role.	154
Table 103: Detailed Design: Presentation Logic Tier: Components sent to the Presentation GUI Tier.	154
Table 104: Data Base: Performance Test. Comparison Different Approaches	171
Table 105: Budget. Costs per Position.	179
Table 106: Budget. Hours per Person and Position	179
Table 107: Budget. Hardware Costs.....	180
Table 108: Budget. Software Costs.	180
Table 109: Budget. Fungible Costs.	180
Table 110: Budget. Indirect Costs.	181
Table 111: Budget. Summary.	181
Table 112: Budget. Final Budget.	181
Table 113: State of the Art: Free open source software CMS.	190

4 Illustrations

Illustration 1: State of the Art. CMS: Plone.	18
Illustration 2: State of the Art: CMS: Plone object content files and folder	19
Illustration 3: State of the Art. CMS: Drupal.	22
Illustration 4: State of the Art. CMS: Joomla.....	23
Illustration 5: State of the Art. CMS: PHP-Nuke.....	24
Illustration 6: State of the Art. CMS: DotNetNuke.....	24
Illustration 7: State of the Art. CMS: Oracle Application Server Portal.	26
Illustration 8: State of the Art. CMS: WebGUI.	26
Illustration 9: State of the Art. CMS: OpenCMS.	27
Illustration 10: yUml Examples.....	50
Illustration 11: DotNetNuke Architecture.....	51
Illustration 12: Conceptual Model DotNetNuke.	52
Illustration 13: DotNetNuke Deployment Diagram.....	52
Illustration 14: DotNetNuke: Screenshot: Presentation.	53
Illustration 15: DotNetNuke: Screenshot: Grid.	53
Illustration 16: DotNetNuke: Integration UmlModel Module with the Forum Module.	54
Illustration 17: Diagram Components.....	56
Illustration 18: Gene-Sarson Diagram. Activity flow diagram.....	56
Illustration 19: Use Case diagram for Unregistered User.	57
Illustration 20: Use Case diagram for Registered User.	59
Illustration 21: Architecture: Cache of results	97
Illustration 22: Architecture: Layers.....	99
Illustration 23: Architecture: Search Result: Sequence Diagram.	100
Illustration 24: Architecture: Model's Page: Sequence Diagram.....	102
Illustration 25: Architecture: Deployment Structure	102
Illustration 26: Architecture. Deployment diagram.....	103
Illustration 27: GUI: Screenshot Draft Model's Page	105
Illustration 28: Model Information Conceptual Model.....	107
Illustration 29: Collaborative Conceptual Model	108
Illustration 30: Commercial Conceptual Model	108
Illustration 31: Main Page. Screenshot.	112
Illustration 32: Search Result: Format of a search result page	113
Illustration 33: Search Result: Format of the additional information for level of detail 1	114
Illustration 34: Search Result: Format of the additional information for level of detail 2	114
Illustration 35: Search Result: Format of the additional information for level of detail 3.....	115
Illustration 36: Search Result: Format of the additional information for level of detail 4	116
Illustration 37: Search Result: Format of the additional information for level of detail 5	116
Illustration 38: Search Result: Level of detail 1	117
Illustration 39: Search Result: Level of detail 2.....	117
Illustration 40: Search Result: Level of detail 3.....	117
Illustration 41: Search Result: Level of detail 4.....	118
Illustration 42: Search Result: Level of detail 5.....	118
Illustration 43: Search Result: Tooltips.	118
Illustration 44: Search Result. Pagination bar.....	119

Illustration 45: Search Result: Results split in columns.....	119
Illustration 46: Search Result: Screenshot resultant work.....	119
Illustration 47: Example of an uncut short description.....	120
Illustration 48: Example of a cut short description.	120
Illustration 49: Structure of the Model's Page	122
Illustration 50: Header of an unregistered user.....	123
Illustration 51: Header of a registered user.	123
Illustration 52: Model's page: Title of the model. Unregistered user view.	123
Illustration 53: Model's page: Title of the model. Registered user view.	123
Illustration 54: Model's page: Superior index. Unregistered user view.	124
Illustration 55: Model's page: Superior index. Owners view.	124
Illustration 56: Model's page: Central area.....	124
Illustration 57: Model's page: Resizing of the main image. Unregistered user view.....	125
Illustration 58: Model's page: Additional Images. Unregistered user view.	125
Illustration 59: Model's page: Resizing of the main image. Registered user view.....	125
Illustration 60: Model's page: Additional Images. Registered user view.....	126
Illustration 61: Model's page: Comparative area for registered users.	126
Illustration 62: Model's page: Control Panel. Registered user view.	127
Illustration 63: Model's page: Control panel: Profile.	127
Illustration 64: Model's page: Control panel: Upload a Model.....	128
Illustration 65: Models page: Control panel: Add Code.....	128
Illustration 67: Model's page: Control panel: My Resources.....	129
Illustration 66: Models page: Control Panel: Add Documentation.....	129
Illustration 68: Model's page: Control Panel: Bookmarks.....	130
Illustration 69: Model's page: Downloadable files.....	130
Illustration 70: Model's page: Downloading files. Unregistered user.....	131
Illustration 71: Model's page: Downloadable Files: Chargeable.....	131
Illustration 72: Model's page: Downloadable Files: Non-Chargeable.....	131
Illustration 73: Model's page: Requirements.....	132
Illustration 74: Model's page: Descriptions. Unregistered user view.....	132
Illustration 75: Model's page: Descriptions. Registered and Unregistered users view.	132
Illustration 76: Model's page: Forum. First approach.....	133
Illustration 77: Model's page: Tooltips.	133
Illustration 78: Model's Page. Buy Credit.....	133
Illustration 79: Model's Page: PayPal.....	133
Illustration 80: Model's page: Punctuation of a resource.....	134
Illustration 81: Model's Page: Resultant work. Unregistered user.....	135
Illustration 82: Model's Page: Resultant work. Registered user.....	136
Illustration 83: Conceptual Model: Specifications of an Artifact	137
Illustration 84: Conceptual Model: Detailed Specifications of an Artifact.....	138
Illustration 85: Conceptual Model: UmlModels Site Information Model	139
Illustration 86: Diagram Point of View	139
Illustration 87: Assemblies Point of View.....	140
Illustration 88: Software Model Diagram.....	140
Illustration 89: Model to Diagram.....	141

Illustration 90: Model to Images	141
Illustration 91: Conceptual Model: Diagram.....	143
Illustration 92: Detailed Design: Subsystems	147
Illustration 93: Detailed Design: Server Subsystems.....	148
Illustration 94: Detailed Design: Controller Class Diagram	148
Illustration 95: Detailed Design: Presentation Logic Tier	151
Illustration 96: Detailed Design: Presentation Logic Tier: Content Package.....	151
Illustration 97: Detailed Design: Scripts Class diagram.....	155
Illustration 98: Detailed Design: Scripts: Site.Master Package.	155
Illustration 99: Detailed Design: Scripts: Users Package.....	156
Illustration 100: Detailed Design: Scripts: UmlModel Package.....	157
Illustration 101: Detailed Design: Scripts: Resources Package.	157
Illustration 102: Detailed Design: Scripts: Search Package.....	159
Illustration 103: Detailed Design: Scripts: Search Result Package.....	160
Illustration 104: Detailed Design: Scripts: Navigation Package.....	161
Illustration 105: Detailed Design: Scripts: Tags Package.....	161
Illustration 106: Detailed Design: Business: Classes Downloadable Package.....	162
Illustration 107: Detailed Design: Data Access class diagram.....	164
Illustration 108: Detailed Design: Client: Presentation GUI Tier.....	167
Illustration 109: VersionableResource: Image, Title, Description and XML	174
Illustration 110: Design of the Data Base: 'DiseñoBaseDatos3.4.vsd'	175
Illustration 111: Screenshot of Youtube embed into Coveralia	185

5 Introduction

The tools and the way that nowadays exists for documenting doesn't get on well with all the different existing file formats and media files.

There are many ways to document software: with a video, an audio, a word document, a PDF, diagrams in different formats, discussions via email or on a chat, Internet resources and references... And you sometimes need to document another document: i.e. some the meeting reports are a document of another document (the recorded videos) and this dissertation is all the documentation of this project piled up.

UmlModels is trying to gather the main existing ways to document and embed them into an area/place easily accessible everywhere you can connect to the Internet. Many advantages can be taken from here such as no worries storing the versions, the storage, wrong references to renamed or moved documents and many, many others. Here, the door is open to a huge world of new revolutionary ideas.

UmlModels is also a search portal with superior semantic content. UmlModels provides a search service of UML-Models addressed to software community engineers. This provided service does not exist on the Internet and aims to be innovative. Unlike other search engines such as Google, UmlModels provides intrinsic information of the artifacts (see '*An Artifact for UmlModels*') in addition to the context information.

Full machines entirely dedicated to search software artifacts (crawlers) are day and night searching information on the Internet that can be relevant for the software community. The retrieved information is studied, indexed, organized and offered to the users of UmlModels into a representative Model.

The work presented here is the first step of a constant evolution towards a new conception of the software documentation. Nowadays the portal is able to provide almost any kind of software artifact; however only images are automatically searched by the crawlers so only software diagrams are indexed. On the future, as soon as we give intelligence enough to our bots to extract semantic from other elements such as source code, packages, assemblies and many others, will be searched into source code files, zips, textual documents such as docs, power points, PDFs and so on.

Specifically, when an image is retrieved, the semantic (relations, entities...) is extracted from it with a smart image-processor and if the image is an UML diagram, then all this rich information is stored in XMI format. Once the image is in XMI format then textual searches with superior semantic content, generation of source code, requirements and related models can be automatically performed, attached to the Model and offered to the community.

Additionally the model has been endowed with a collaborative area around it in which the users can interact. So members of the community are able to post a comment, punctuate, create a model and add some other artifacts such as requirements, source code packages, descriptions, etc called resources within a Model.

Bearing in mind how cheap can be the extraction of the semantic of the resources if intelligent programs (bots) are involved on this process, a business opportunity is presented if we provide all this semantic to the software community with a low enough cost to be worthy to buy this provided service. For instance, if a software designer finds a diagram that fits with what he is modelling, it will be cheaper for him to pay something like 1 € for downloading the XMI and import it in his case tool than drawing it himself. Many other semantic can be provided such as the source code or even the test code among many others.

Considering also that the users can contribute extremely much to the software community, UmlModels will also allow to the users to add their own resources. The owner of a resource can also take economical advantage from downloads performed by other users of his resources. In these cases half of the price of the download will go to UmlModels and the other to the owner.

UmlModels is also interested on providing a place in UmlModels to the free software community. For this, UmlModels will offer free models as well as the users will have the chance of doing free the resources that they have uploads. Other business areas will be covered such as the inclusion of a Bank Job with offers in which you can do offers added by other users for an economical reward. Finally the last business area that will give profits to UmlModels will be the inclusion of sponsored links into the models page and the search result page. Lastly, many other new business ideas have come up during this project; some of them are commented in the Future Lines.

To conclude, the users are going to be able to perform searches with powerful tools such as specific language domain search, graphical search (diagrammer) and interactive search (Virtual Assistance). All of them will understand the semantic of the queries performed by the user and as result, a set of elements related with the software engineering domain will be presented with useful actions for the reutilization and documentary collaboration.

6 State of the Art

The purpose of this state of the art is to make a review of the main existing technologies for building a Web portal. UmlModels project is focused on *.NET*, so a special attention of the elements based on this platform is paid.

In this state of art has been mainly covered those existing technologies that already provides a structure or a framework to create and develop web pages. Also some programming techniques for automatic converting/mapping relational database types into objects-oriented data types have been studied.

The Web technologies studied are mainly:

- ➔ **Content management system (CMS):** A content management system (*CMS*) is a system used to organize and facilitate collaborative content creation.
- ➔ **Microsoft Web Technologies:**
 - **Microsoft Web Forms:** ASP.NET Webforms is a web application framework developed and marketed by Microsoft, that programmers can use to build dynamic web sites, web applications...
 - **ASP.NET Model View Controller (MVC) Framework:** A free and fully supported Microsoft framework for building web applications that use a model-view-controller pattern.
 - **Microsoft Silverlight:** Microsoft Silverlight is a free framework, which allows you to build new types of applications for the Web regardless of target platform or browse.
- ➔ **EXTJS Framework:** A framework based on JavaScript language for the Client side.

The technologies studied for making easier the mapping of the database to the object-oriented languages called Object-Relation Mapping (ORM) technologies are:

- ➔ **Data Sets**
- ➔ **Linq to SQL**
- ➔ **Entity Framework**

6.1 Study of the Content Management System (CMSs)

In this section is made a review of the web content management systems (WCMS) in order to use one of these technologies to make the Web Portal for UmlModels.

A content management system (*CMS*) is a system used to organize and facilitate collaborative content creation. Recently, the term has been used specifically to refer to programs on WWW servers, but it can also refer to hardware devices that manage documents on a large network.

A CMS gives you an easy way to make the portal structure without wasting time in things that they are already done and focusing in the integration between the providing information engine systems.

Since UmlModels projects are focused on *.NET*, in this section we pay special attention to the web content management system based on this platform. On the other hand, most of the

WCMS are written in PHP, so they are already described due their use has not been dismissed. For the same reason, also they are described some other WCMS that are written in any other language.

This section is made up of four subsections: **Web Content Management System**, **Content Management Software**, **Comparison**, **Conclusion** and **Free and open source software CMS**. In the section first subsection is described what a web content management system is. Next subsection it is described a comparison between the most interesting WCMS for this project. And in the last one it is listed in less detail some of the free and open source CMS software.

6.1.1 Web Content Management System

A web content management system (WCMS) is content management system (CMS) software, usually implemented as a web application, for creating and managing HTML content. It is used to manage and control a large, dynamic collection of web material (HTML documents and their associated images). A CMS facilitates content creation, content control, editing, and many essential web maintenance functions.

Usually the software provides authoring and other tools designed to allow users with little or no knowledge of programming languages or mark-up languages to create and manage content with relative ease of use.

Most systems use a database to store content, metadata, and/or artefacts that might be needed by the system. Content is frequently, but not universally, stored as XML, to facilitate reuse and enable flexible presentation options [1, 2].

A presentation layer displays the content to regular website visitors based on a set of templates. The templates are often XSLT files [3].

Administration is typically done through browser-based interfaces, but some systems require the use of a fat client.

A Content Management System (CMS) differs from website builders like Microsoft FrontPage or Adobe Dreamweaver. A CMS allows non-technical users to make changes to an existing website with little or no training. Web content management systems typically require an experienced coder to set-up and add features, but it is primarily a website maintenance tool for non-technical administrators.

6.1.2 Content Management Software

There are quite a lot of content management systems (see *Appendix A: Free and open source software CMS*). The most popular are:

- **Plone:** Plone is built on the **Zope** application server, which is written in *Python*.
- **Drupal:** Is a free and open source modular framework and content management system written in the programming language PHP.
- **Joomla:** Is a free, open source content management system for publishing content on the World Wide Web and intranets. It is written in the PHP programming language and uses the MySQL database by default.

- **PHP-Nuke:** Is a web based automated news publishing and content management system (a "nuke"[citation needed]) based on PHP and MySQL. The system is fully controlled using a web-based user interface.

There are also other interesting content management systems such as:

- **DotNetNuke:** Is an open source web application framework written in VB.NET for the ASP.NET framework. The application's content management system is extensible and customizable through the use of skins and modules, and it can be used to create, deploy, and manage intranet, extranet, and web sites.
- **Oracle Application Server Portal:** Portal used by the [new web site](#) of the university: www.uc3m.es. Java, PL/SQL.
- **WebGUI:** Is an open source content management system written in Perl and released under the GNU General Public License.
- **OpenCMS:** Is an open source content management system based on Java and XML technology. It is distributed under LGPL license.

6.1.2.1 Plone

Plone is built on the Zope application server, which is written in Python. Plone is made such that all information stored in Plone is stored in Zope's built-in transactional object database (ZODB).

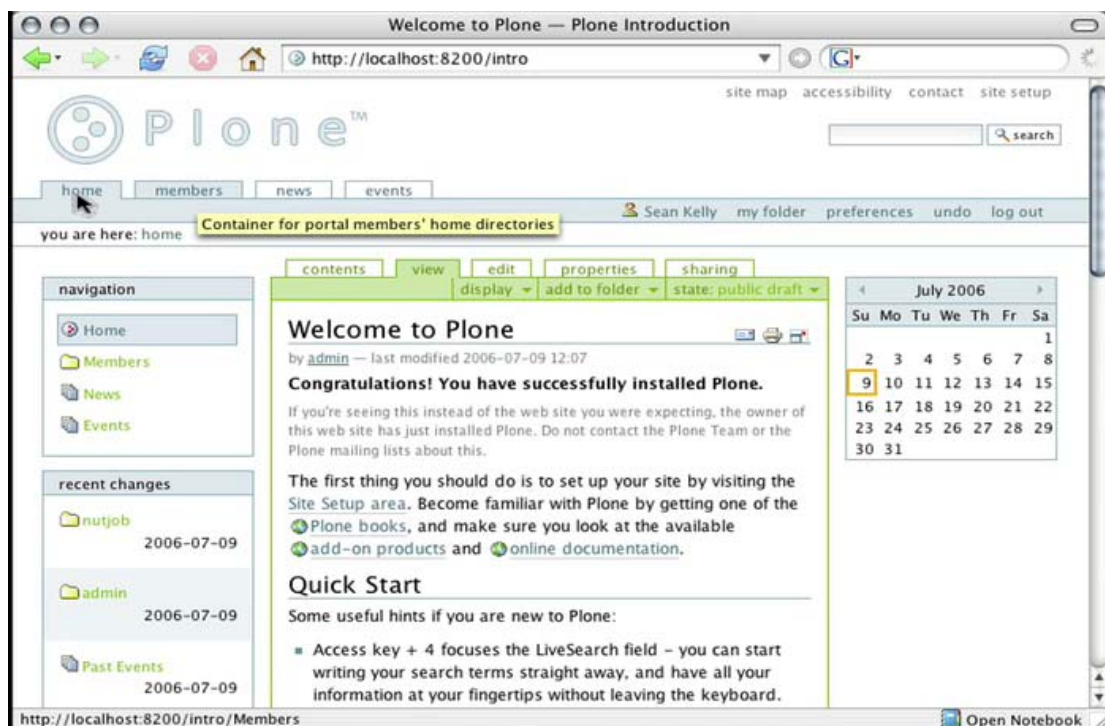


Illustration 1: State of the Art. CMS: Plone.

Plone comes with installers for Windows, Mac OS X, and Linux, along with other operating systems. New updates are released regularly on Plone's website. Plone is available in over 35 languages. Its interface follows the government standard WAI-AAA and U.S. section 508, which allows people with sight disabilities to properly access and use Plone.

A major part of Plone is its use of skins and themes. When working with Plone, templates can be used to customize a website's look. These templates are written with Cascading Style Sheets.

In addition, Plone comes with a user management system called Pluggable Authentication Service. Introduced in Plone 2.5, "PAS" is used to properly sort actions from different users to their respective folders or accounts. PAS is also used to search for users and groups in Plone. Most importantly, PAS covers the security involved for users, requiring authentication in order to login to Plone. This gives users an increase in both security and organization with their content.

A large part of Plone's changes have come from its community. Since Plone is open source, the members of the Plone community regularly make alterations or add-ons to Plone's interface, and make these changes available to the rest of the community via Plone's website.

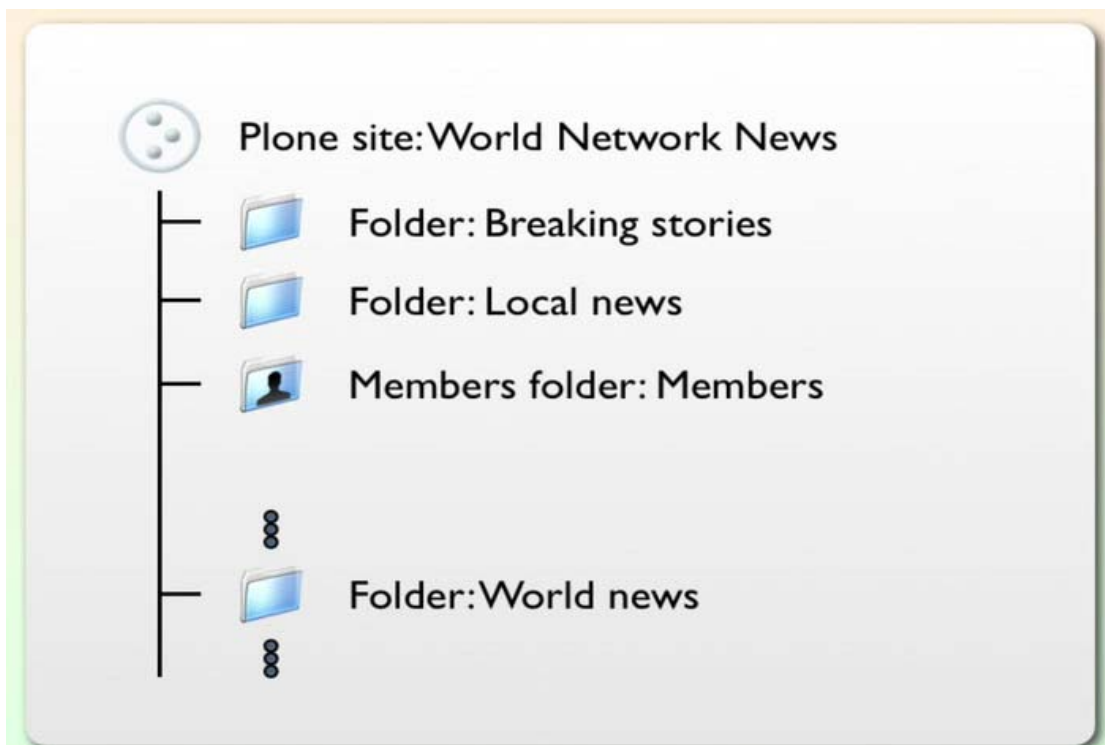


Illustration 2: State of the Art: CMS: Plone object content files and folder

When you work with Plone, you have a content objects which are arrange like files and folders on a computer.

6.1.2.1.1 Strengths/Weaknesses

Plone offers customization options in all its features, and provides many free add-on products on its website. In addition, one of the benefits of Plone is its availability on almost any OS.

Also, Plone's administrative interface works with almost all Internet browsers; many content management systems fail to allow for this. Plone also comes with a fairly simple installation, known to be easier than most other CMS's.

One of Plone's greatest strengths is its accessibility: Plone is built following government standards such that it is user friendly with people with disabilities.

Some have complained about Plone's learning curve, saying that experience in both Zope and Python is necessary to properly use Plone. Also, some have complained that there is not enough documentation available for Plone [4].

6.1.2.1.2 Websites using Plone

- ❖ <https://www.cia.gov>
- ❖ <http://community.ni.com>
- ❖ <http://www.discovermagazine.com>
- ❖ <http://www.fsf.org>
- ❖ <http://www.ilm.com/>
- ❖ <http://www.nasa.gov/>

6.1.2.1.3 Features

These are some of the features available in Plone 3.0:

- Inline editing.
- Working Copy support.
- Link and reference integrity checking.
- Automatic locking and unlocking.
- Collaboration and sharing.
- Versioning, history and reverting content.
- Upgraded visual HTML editor.
- Workflow capabilities.
- Authentication back-end.
- Full-text indexing of Word and PDF documents.
- Collections.
- Presentation mode for content.
- Support for the search engine Sitemap protocol.
- Support for multiple mark-up formats.
- Wiki support.
- Automatic previous/next navigation.
- Rules engine for content.
- Auto-generated tables of contents.
- Portlets engine.
- Support, development, hosting & training.
- LiveSearch.
- Multilingual content management.
- Time-based publishing.
- Human-readable URLs.
- Powerful graphical page editor.
- Navigation and updated site maps.
- Resource compression.
- Caching proxy integration.
- Drag and drop reordering of content.
- XML exports of site configurations.
- Localized workflow configuration.
- Adjustable templates on content.

- Standard content types.
- Content is automatically formatted for printing.
- Standards-compliant XHTML and CSS.
- Accessibility compliant.
- RSS feed support.
- Automatic image scaling and thumbnail generation.
- Free add-on products.
- Cross-platform.
- Comment capabilities on any content.
- Microformat support.
- Installer packages for multiple platforms.
- FTP support.
- In-context editing.
- Backup support.
- Cut/copy/paste operations on content.

6.1.2.2 Drupal

Drupal is a free and open source modular framework and content management system (CMS) written in the programming language PHP.

Drupal, like many modern CMSs, allows the system administrator to create and organize content, customize the presentation, automate administrative tasks, and manage site visitors and contributors. Although there is a sophisticated programming interface, most tasks can be accomplished with little or no programming. Drupal is sometimes described as a "web application framework," as its capabilities extend from content management to enabling a wide range of services and transactions.

Drupal runs in many environments, including Windows, Mac OS X, Linux, FreeBSD, OpenBSD, and any platform that supports either the Apache (version 1.3+), or IIS (version IIS5+) Web server and the PHP language (version 4.3.3+). Drupal requires a database such as MySQL or PostgreSQL to store content and settings.



Illustration 3: State of the Art. CMS: Drupal.

The core Drupal install comes with a blog system, forum, and a menu system to create either a classic "brochureware" website or an interactive community website.

As in WordPress or Joomla!, content is stored independent of a hierarchical menu. The user can choose to provide this "classical" navigation and/or make use of the Taxonomy System which sorts tagged content into categories.

The Drupal website provides more than 2250 free modules written and contributed back by the Drupal community.

6.1.2.2.1 Features

Modules included in Drupal's 5.x core[6] enable users to:

- Post, revise, and categorize content.
- Conduct searches on internal site content.
- Post comments.
- Take part in forums.
- Vote in polls.
- Work on collaborative writing projects.
- Post and view personal profiles.
- Communicate among themselves or with the managers of a site.
- Change the look of a site with off-the-shelf or custom-made themes.
- Build multi-level menus.
- Provide users with an interface in their local language.
- Provide RSS feeds.
- Aggregate content from the RSS feeds of other sites.
- Register and manage user accounts.
- Assign fine-grained user roles, granting users permission to use selected features of a site.

- Use access rules to deny site access to specified usernames, e-mail addresses, and IP addresses.
- Provide statistics and reports for administrators.
- Manage caching and throttling to improve how a site performs in heavy traffic.
- Construct and specify various input filters and content types.
- Generate easy-to-remember URLs that can also be more easily interpreted by search engines (for example, "www.mysite.com/products" rather than "www.mysite.com/?q=node/432") "Clean URLs" (removing the ?q=) are possible by default on most servers, but some may require special configuration.

6.1.2.3 Joomla

Joomla! is a free, open source content management system for publishing content on the World Wide Web and Intranets. The system includes features such as page caching to improve performance, RSS feeds, printable versions of pages, news flashes, blogs, polls, website searching, and language internationalization. Joomla is licensed under the GPL, and is the result of a fork of Mambo.

It is written in the PHP programming language and uses the MySQL database by default.

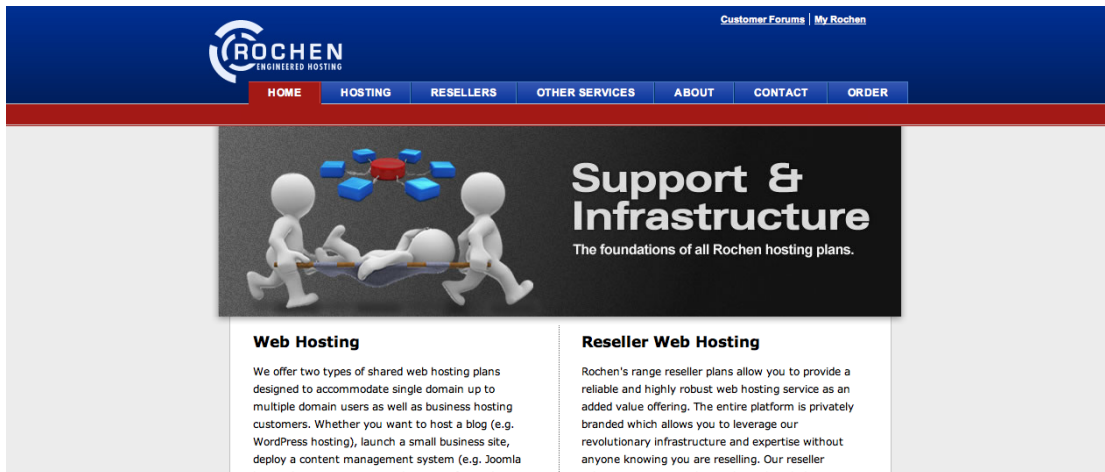


Illustration 4: State of the Art. CMS: Joomla.

6.1.2.3.1 Features

- Dynamic form builders.
- Business or organizational directories.
- Document management.
- Image and multimedia galleries.
- E-commerce and shopping cart engines.
- Forums and chat software.
- Calendars.
- Blogging software.
- Directory services.
- Email newsletters.
- Data collection and reporting tools.
- Banner advertising systems.
- Subscription services.

6.1.2.4 PHP-Nuke

PHP-Nuke is a web based automated news publishing and content management system (a "nuke") based on PHP and MySQL. The system is fully controlled using a web-based user interface.

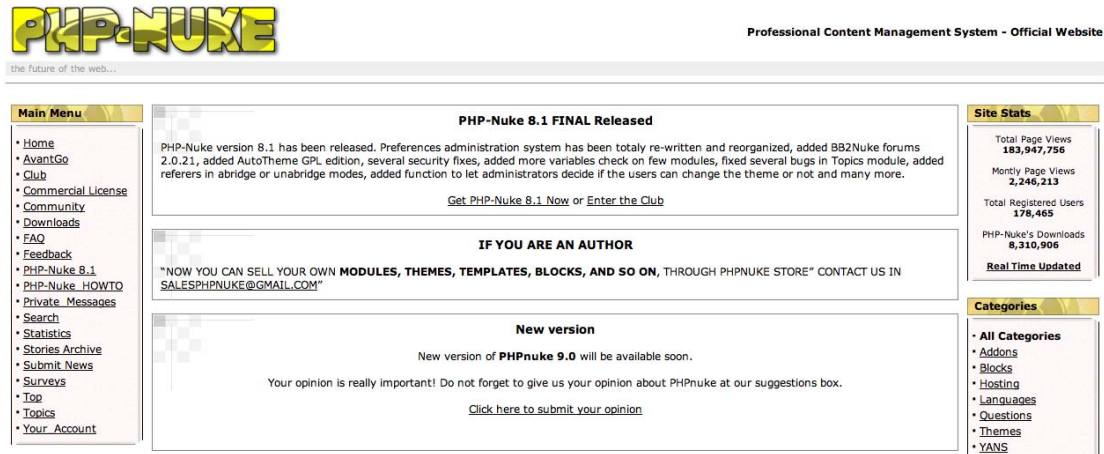


Illustration 5: State of the Art. CMS: PHP-Nuke.

6.1.2.5 DotNetNuke

DotNetNuke is an open source web application framework written in VB.NET for the ASP.NET framework. The application's content management system is extensible and customizable through the use of skins and modules, and it can be used to create, deploy, and manage intranet, extranet, and web sites.



Illustration 6: State of the Art. CMS: DotNetNuke.

About a dozen basic modules are included with the core DotNetNuke distribution, and further modules can be downloaded from the DotNetNuke website, including e-commerce systems, photo galleries, blogs, forums, wiki and mailing list options. Additional third party modules are provided by both the open source community and proprietary commercial developers.

DotNetNuke has a skinning architecture which provides a clear separation between design and content, enabling a web designer to develop skins without requiring any specialist knowledge of development in ASP.NET: only knowledge of HTML and an understanding of how to prepare

and package the skins themselves are required. Skins consist of basic HTML files with placeholders for content, menus and other functionality, along with support files such as images, style sheets and JavaScript, packaged in a ZIP file.

6.1.2.5.1 Websites using DotNetNuke

- ❖ <http://youthroots.faihstreams.com/>
- ❖ <http://www.earsinus.com/>
- ❖ <http://www.wineaustralia.com/australia/>
- ❖ <http://www.bearmountain.ca/>
- ❖ ...

6.1.2.6 Oracle Application Server Portal

OracleAS Portal is a Web-based application for building and deploying portals. It provides a secure, manageable environment for accessing and interacting with enterprise software services and information resources.

OracleAS Portal incorporates a portal-building framework with self-service publishing features that assist you in creating, publishing, and managing information within your portal.

6.1.2.6.1 Features

Key features include:

- **An extensible framework** that integrates Web-based resources such as Web pages, applications, business intelligence reports, and syndicated content feeds, within standardized, reusable information components called portlets. Within a portlet, these resources are personalized and managed as a service of OracleAS Portal. Companies can create their own portlets for their existing Web resources and can select additional portlets from the growing catalog of third-party portlet providers. The portal framework provides additional services including single sign-on, content classification, enterprise search, directory integration, and access control.
- **An easy-to-use**, personalized interface that provides an organized, consistent view of the business information, Web content, and applications needed by each user. Portal administrators use the browser-based control panel to selectively grant access to applications and information by making portlets available only to specific users or user groups. An administrator can instantly deliver new content to thousands of users by simply adding a portlet to the users' view of the portal.
- **Self-service publishing features** allow end users to post and share any kind of document or Web content with other users anywhere in the world. Knowledge workers use intuitive controls for document/file upload, version control, page formatting/display, and access control to publish and manage their content; no technical expertise or knowledge of HTML is required. Users with minimal development experience can build a variety of application components (Web forms, charts, reports, and the like) that display and interact with data managed in an Oracle database.
- **A scalable deployment architecture** that is easily configured for departmental, regional, and enterprise-wide deployment. The deployment model supports a variety of configurations, including single box, multi-tier, and multi-tier with cache, on a broad set of hardware platforms and operating systems. The deployment architecture can be configured to support over 20 languages.



Illustration 7: State of the Art. CMS: Oracle Application Server Portal.

6.1.2.7 WebGUI

WebGUI is a content management platform built to allow average business users to build and maintain complex web sites. It is modular, pluggable, and platform independent. It was designed to allow the people who create the content, to manage it online, rather than content management taking up the time of the busy IT Staff.



Illustration 8: State of the Art. CMS: WebGUI.

6.1.2.7.1 Features

WebGUI is a content application framework, which allows for easy content management, while maintaining the ability to create and install custom applications. With WebGUI, you can:

- publish articles
- participate in forums
- create photo galleries
- conduct surveys and polls
- manage projects
- create interactive event calendars
- create complex data entry forms
- sell and advertise products
- sell and maintain subscription services
- maintain site security through users and groups
- manage individual user interface levels
- much, much more!

6.1.2.8 OpenCMS

OpenCms is an open source content management system based on Java and XML technology. It is distributed under LGPL license.



Illustration 9: State of the Art. CMS: OpenCMS.

6.1.2.8.1 Features

It is a CMS application with features such as Browser-enabled work environment, Asset Management, Integrated user management and permission system, Project based publishing, Workflow and Task Management, WYSIWYG editor, Internationalization support, Versioning of content, JSP and XML Template mechanism, Multi-Language, Online-Help System, Dynamic and static content publishing, Personalization, Caching system, Module mechanism for extensions, Scheduling system, Synchronization mechanism, Import / export of content, Application server integration / EJB support and others...

6.1.2.8.2 Websites using OpenCMS

- ❖ <http://www.alkacon.com/en/>
- ❖ <http://www.debeka.de/>
- ❖ <http://www.destinatortechologies.net/>
- ❖ <http://www.hansenet.de/>
- ❖ <http://www.lanmagazine.nl/>

6.1.3 Comparison

In this section is shown **Requirements, Security, Support, Easy to Use, Performance, Management, Interoperability, Flexibility** and **Applications** comparison between the CMS described previously.

And in Appendix A it's shown a free and open source software CMS list.

6.1.3.1 System Requirements

Product	DotNetNuke 4.5.5	Drupal 6.1	Joomla! 1.5.1	OpenCms 7.0.1	Oracle Portal 10.1.4	PHP Nuke 6	Plone 3.0	WebGUI 7.4
Last Updated	11/29/2007	2/28/2008	2/26/2008	7/26/2007	11/22/2005	2/7/06	9/8/07	1/9/08

System Requirements	DotNet Nuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Application Server	IIS	PHP 4.3.5+	Apache recommended, any server that supports PHP and MySQL	Tomcat, JBoss, Bea Weblogic...	Oracle Application Server 10g	mod_php	Zope	mod_perl
Approximate Cost	Free	Free	\$0	Free	\$10k per processor	Free	Free	Free
Database	MSSQL 2005/2000, MSSQL Express 2005, MSDE	MySQL, Postgres	MySQL	MySQL, PostGreSQL, Oracle, MSSQL	Oracle	MySQL, Postgres, mSQL, Interbase, Sybase	Zope	MySQL
License	BSD (Modified)	GNU GPL	GNU/GPL v2	GNU LGPL	Commercial, per CPU	GNU GPL	GNU GPL	GNU GPL
Operating System	Windows	Any	Any	Any	Windows, Unix, Linux	Any	Any	Any
Programming Language	ASP.NET 2.0, VB.NET, C#	PHP	PHP	Java 1.4. +	Java, PL/SQL	PHP	Python	Perl
Root Access	No	No	No	No	Yes	Yes	No	Yes
Shell Access	No	No	No	No	Yes	Yes	Yes	Yes
Web Server	IIS 5.0 & IIS 6.0 & IIS 7.0	Apache, IIS	Apache	Tomcat, Apache, IIS	Apache	Apache, IIS	Apache, Nginx, IIS, (+anything that can proxy)	Apache

6.1.3.2 Security

Security	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Audit Trail	Limited	Yes	No	Yes	Yes	No	Yes	Yes
Captcha	Yes	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Content Approval	Limited	Yes	Yes	Yes	Yes	No	Yes	Yes
Email Verification	Yes	Yes	Yes	No	No	No	Yes	Yes
Granular Privileges	Yes	Yes	No	Yes	Yes	Limited	Yes	Yes
Kerberos Authentication	Free Add On	No	No	No	Yes	No	Free Add On	No
LDAP Authentication	No	Free Add On	Yes	Costs Extra	Yes	No	Yes	Yes
Login History	Yes	Yes	Yes	Yes	Yes	No	Free Add On	Yes
NIS Authentication	No	No	No	No	No	No	Free Add On	No
NTLM Authentication	Yes	Free Add On	No	No	No	No	Free Add On	No
Pluggable Authentication	Yes	Yes	No	Costs Extra	Yes	No	Yes	Yes
Problem Notification	Yes	No	No	Yes	No	No	Free Add On	Yes
Sandbox	Yes	No	No	Yes	Yes	No	Yes	Yes
Session Management	Yes	Yes	Yes	No	Yes	No	Free Add On	Yes
SMB Authentication	No	No	No	No	No	No	Free Add On	No
SSL Compatible	Yes	Yes	No	Yes	Yes	No	Yes	Yes
SSL Logins	Yes	No	Yes	Yes	Yes	No	Free Add On	Yes
SSL Pages	Yes	No	No	Yes	Yes	No	No	Yes
Versioning	Limited	Yes	No	Yes	Yes	No	Yes	Yes

6.1.3.3 Support

Support	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Certification Program	Yes	No	No	No	Yes	No	No	No
Code Skeletons	Yes	Yes	No	No		No	Yes	Yes
Commercial Manuals	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Commercial Support	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Commercial Training	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Developer Community	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Online Help	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Pluggable API	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Professional Hosting	Yes	Yes	Yes	Limited	Yes	Yes	Yes	Yes
Professional Services	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Public Forum	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Public Mailing List	Yes	Yes	No	Yes	No	No	Yes	Yes
Test Framework	Yes	Free Add On	No	No		No	Yes	Yes
Third-Party Developers	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Users Conference	Yes	Yes	Yes	No	No	No	Yes	Yes

6.1.3.4 Easy to use

Ease of Use	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Drag-N-Drop Content	Yes	Free Add On	No	Limited	No	No	Yes	Yes
Email To Discussion	Costs Extra	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Friendly URLs	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Image Resizing	Free Add On	Free Add On	Yes	Yes	No	No	Yes	Yes
Macro Language	Yes	Free Add On	Yes	No	No	No	Yes	Yes
Mass Upload	Yes	Free Add On	No	Yes	Yes	No	Yes	Yes
Prototyping	Yes	Limited	Yes	No	Yes	No	Yes	Yes
Server Page Language	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Site Setup Wizard	No	Limited	No				No	Yes
Spell Checker	Free Add On	Free Add On	No	Free Add On	No	No	Free Add On	Yes
Style Wizard	Yes	Limited	No	No		No	Free Add On	Yes
Subscriptions	Yes	Free Add On	No	Costs Extra	No	No	Yes	Yes
Template Language	Yes	Limited	Yes	Yes	Yes	No	Yes	Yes
UI Levels	Yes	No	No	Yes	No	No	Yes	Yes
Undo	Limited	Limited	No	Yes	No	No	Yes	Yes
Zip Archives	Yes	No	No	Limited		No	Free Add On	Yes

6.1.3.5 Management

Management	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Advertising Management	Yes	Free Add On	Yes	No	No	Yes	Free Add On	Yes
Asset Management	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Clipboard	Limited	No	No	No	No	No	Yes	Yes
Content Scheduling	Yes	Free Add On	Yes	Yes	Yes	No	Yes	Yes
Content Staging	Yes	Free Add On	No	Limited	No	No	Free Add On	Limited
Inline Administration	Yes	Yes	Yes	Limited	Yes	No	Yes	Yes
Online Administration	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Package Deployment	Yes	No	No	Limited	Limited	No	Yes	Yes
Sub-sites / Roots	Yes	Yes	Yes	Yes	Limited	No	Yes	Yes
Themes / Skins	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Trash	Yes	No	Yes	Yes	Yes	No	Free Add On	Yes
Web Statistics	Yes	Yes	Yes	No	Limited	Yes	Free Add On	Yes
Web-based Style/Template Management	Yes	Yes	Yes	Limited	Yes	Limited	Yes	Yes
Web-based Translation Management	Yes	Yes	Free Add On	No	Yes	No	Yes	Free Add On
Workflow Engine	Free Add On	Limited	No	Costs Extra	Yes	No	Yes	Yes

6.1.3.6 Performance

Performance	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Advanced Caching	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Database Replication	No	Limited	No	Costs Extra	Yes	No	Yes	Yes
Load Balancing	Yes	Yes	No	Costs Extra	Yes	No	Yes	Yes
Page Caching	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Static Content Export	Yes	No	No	Yes	No	No	Free Add On	Yes

6.1.3.7 Interoperability

Interoperability	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Content Syndication (RSS)	Yes	Yes	Yes	Costs Extra	No	No	Yes	Yes
FTP Support	Yes	Limited	Free Add On	No	No	No	Yes	No
iCal	Yes	Free Add On	No				Free Add On	Yes
UTF-8 Support	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
WAI Compliant	Limited	Limited	No	Limited	No	No	Yes	Yes
WebDAV Support	Costs Extra	No	No	Yes	Yes	No	Yes	No
XHTML Compliant	Yes	Yes	No	Yes	No	No	Yes	Yes

6.1.3.8 Flexibility

Flexibility	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
CGI-mode Support	No	Yes	No	No	No	Yes	Yes	No
Content Reuse	Yes	Limited	Yes	Yes	Yes	No	Yes	Yes
Extensible User Profiles	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Interface Localization	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Metadata	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Multi-lingual Content	Free Add On	Yes	Free Add On	Yes	Yes	No	Yes	Yes
Multi-lingual Content Integration	Free Add On	Free Add On	Free Add On	Limited	Yes	No	Yes	No
Multi-Site Deployment	Yes	Yes	Free Add On	Yes	Limited	No	Yes	Yes
URL Rewriting	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes

6.1.3.9 Built-in Applications

Built-in Applications	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Blog	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Chat	Yes	Free Add On	Free Add On	No	No	No	Free Add On	Free Add On
Classifieds	Free Add On	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Contact Management	Yes	Free Add On	Yes	No	Costs Extra	No	Free Add On	Yes
Data Entry	Yes	Free Add On	Free Add On	No	Yes	No	Free Add On	Yes
Database Reports	Yes	No	Free Add On	Costs Extra	Yes	No	Yes	Yes
Discussion / Forum	Yes	Yes	Free Add On	Free Add On	No	Yes	Free Add On	Yes
Document Management	Yes	Limited	Free Add On	No	Yes	No	Yes	Limited
Events Calendar	Yes	Free Add On	Free Add On	Costs Extra	Costs Extra	No	Yes	Yes
Events Management	Yes	Free Add On	Free Add On	No			Yes	Yes
Expense Reports	Costs Extra	No	Free Add On	No	Costs Extra	No	Free Add On	No
FAQ Management	Yes	Yes	Yes	Costs Extra	No	Yes	Free Add On	Yes
File Distribution	Yes	Free Add On	Free Add On	No	Yes	Yes	Yes	Yes
Graphs and Charts	Yes	No	Free Add On	No	Yes	No	Free Add On	Yes

Built-in Applications	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Groupware	Free Add On	Free Add On	Free Add On	No	Costs Extra	No	Free Add On	No
Guest Book	Yes	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Help Desk / Bug Reporting	Costs Extra	Free Add On	Free Add On	No	No	No	Free Add On	Yes
HTTP Proxy	Free Add On	No	No	No	Yes	No	Free Add On	Yes
In/Out Board	Costs Extra	No	No	No	No	No	Free Add On	Yes
Job Postings	Free Add On	Free Add On	Free Add On	Costs Extra	No	No	Free Add On	Yes
Link Management	Yes	Free Add On	Yes	Yes	Yes	Yes	Yes	Yes
Mail Form	Yes	Free Add On	Yes	Yes	No	No	Free Add On	Yes
Matrix	No	No	No	No		No	No	Yes
My Page / Dashboard	Yes	Free Add On	No	No	Yes	No	Yes	Yes
Newsletter	Yes	Free Add On	Free Add On	Costs Extra	No	No	Free Add On	Yes
Photo Gallery	Yes	Free Add On	Free Add On	Yes	No	No	Yes	Yes
Polls	Yes	Yes	Yes	Free Add On	Yes	Yes	Free Add On	Yes
Product Management	Yes	Free Add On	Yes	Costs Extra	No	No	Yes	Yes
Project Tracking	Costs Extra	Free Add On	Free Add On	No	Costs Extra	No	Free Add On	Yes

Built-in Applications	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Search Engine	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Site Map	Yes	Free Add On	Free Add On	Yes	Free Add On	No	Yes	Yes
Stock Quotes	Costs Extra	Free Add On	No	No		No	Free Add On	Yes
Surveys	Yes	Free Add On	Free Add On	No	Yes	No	Free Add On	Yes
Syndicated Content (RSS)	Yes	Yes	Yes	Costs Extra	Yes	Yes	Yes	Yes
Tests / Quizzes	Costs Extra	Free Add On	Free Add On	No	Yes	No	Free Add On	Yes
Time Tracking	Costs Extra	Free Add On	No	No	No	No	Free Add On	Yes
User Contributions	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Weather	Free Add On	Free Add On	No	No		No	Free Add On	Yes
Web Services Front End	Free Add On	Limited	Free Add On	No	Limited	No	No	Yes
Wiki	Yes	Free Add On	Free Add On	No	No	No	Free Add On	Yes

6.1.3.10 Commerce

Commerce	DotNetNuke	Drupal	Joomla!	OpenCms	Oracle Portal	PHP Nuke	Plone	WebGUI
Affiliate Tracking	Yes	Free Add On	Free Add On	No	No	No	No	No
Inventory Management	Costs Extra	Free Add On	Free Add On	No	No	No	Free Add On	No
Pluggable Payments	Yes	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Pluggable Shipping	Costs Extra	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Pluggable Tax	Costs Extra	Free Add On	Free Add On	No	No	No	Free Add On	No
Point of Sale	Costs Extra	No	Free Add On	No	No	No	Free Add On	No
Shopping Cart	Yes	Free Add On	Free Add On	No	Costs Extra	No	Free Add On	Yes
Subscriptions	Yes	Free Add On	Free Add On	No	No	No	Free Add On	Yes
Wish Lists	No	Free Add On	Free Add On	No	No	No	No	No

6.1.4 Conclusion

Comparing the eight different WCMS proposed, **Plone** is the one who cover more properties. In fact, it covers almost all the features studied. As well it is the web content management used by sites such as NASA, CIA, etc. Plone is much extended with such a huge amount of add-ons.

The next who covers more features is **webGUI**, which is written in Perl.

Others like **DotNetNuke** and **OpenCMS** they don't cover many properties such us the ones mentioned above, but they could be useful because they use *.Net* and *Java* environments.

6.2 Microsoft Web Technologies

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. ASP.NET is the successor to Microsoft's Active Server Pages (ASP) technology.

Within ASP.NET technologies, the follow ones are studied and later on compared:

- ➔ Web Forms.
- ➔ ASP.NET MVC Framework.
- ➔ Microsoft Silverlight.

6.2.1 Microsoft Web Forms

ASP.NET is a web application framework developed by Microsoft, which programmers can use to build dynamic web sites, web applications...

Web Forms and ASP.NET were created to get better some of the limitations of ASP. These new strengths include [9]:

- Separation of HTML interface from application logic.
- A rich set of server-side controls that can detect the browser and send out appropriate mark-up language such as HTML.
- Less code to write due to the data binding capabilities of the new server-side .NET controls.
- Event-based programming model that is familiar to Microsoft Visual Basic programmers.
- Compiled code and support for multiple languages, as opposed to ASP which was interpreted as Microsoft Visual Basic Scripting (VBScript) or Microsoft Jscript.
- Allows third parties to create controls that provide additional functionality.

6.2.2 ASP.NET MVC Framework

ASP.NET MVC is a free and fully supported Microsoft framework for building web applications that use a model-view-controller pattern. Like ASP.NET Web Forms, ASP.NET MVC is built on the ASP.NET framework.

As before mentioned, ASP.NET MVC implements Model-View-Controller UI pattern for web application development. It allows you to create applications in a loose coupling manner as the MVC pattern divides the application in three parts: Model, View and Controller. The View is merely html templates that are filled with the application`s data passed by the controller. It is also responsible for rendering the user interface of the application (UI). The Model implements the logic for the data and portrays the business objects of the application that use the View for rendering the interface. Controllers are responds and handles for user input and interactions. Web requests are handled by the Controller, and the Controller will also decide which model objects are to be used and which view objects are to be rendered. The Web Form events are replaced by the MVC model with the Controller actions.

ASP.NET MVC provides the following benefits [10]:

- Provides complete control over your HTML mark-up

- Enables rich AJAX integration
- Intuitive website URLs
- Clear separation of concerns which results in web applications that are easier to maintain and extend over time.
- Testability – including support for test-driven development.

ASP.NET MVC Framework was launched at the end of 2007 and after five previews and when this study was done (the end of 2008) the beta version had been released. Up to date two more beta versions were released and at the moment the first version was released in March of 2009.

This framework allows you to create a Web Portal in a very easy way. The basics of *ASP.NET MVC* are that physical pages doesn't exist on the disc instead of that a view is depending on the path of the URL. This is, without you programming anything *ASP.NET MVC* map all the requests to actions in an existing controller.

For instance, given the follow URL: <http://guillermo.ie.inf.uc3m.es/Home/Index/2> the default behaviour (mapping properties can be easily and flexible configured) will be:

- ➔ The ASP.NET MVC will look into the controller for the *HomeController* class.
- ➔ Within the *HomeController* class the *Index* method will be called.
- ➔ Optionally: The parameter **2** will be given to the *Index* method.

Additionally, the way in which the controller encapsulates the result returned (*ActionResults*) makes the application more flexibility. An action Controller can be implemented, but some defaults actions are provided such as:

1. *ViewResult*: Encapsulate HTML and Markup result.
2. *EmptyResult*: An empty result is provided.
3. *RedirectResult*: Redirection to a new URL.
4. *RedirectToRouteResult*: Redirection to a new Controller Action.
5. *JsonResult*: Java Script Object Notation Result
6. *ContentResult*: Text Result.

To conclude, this framework provides a structured model in which it is clearly separated the parts of an application, makes easier to build unitary tests and support a TDD (Test Driven Development) workflow. Also allow as to maintain a higher control over the published URLs of our application and optionally, a higher control over the HTML that is emitted for them.

6.2.3 Microsoft Silverlight

Microsoft Silverlight is a free runtime that powers rich application experiences and delivers high quality, interactive video across multiple platforms and browsers, using the .NET framework. Microsoft Silverlight enables features such as animation, vector graphics and audio-video playback which characterize rich Internet applications.

Silverlight is consisted of the core presentation framework responsible for UI, basic UI controls, user input and interactivity, animation and graphics, as well as Digital rights management, media playback, and DOM integration. It's composed of the following components [11]:

- Input – handling input from devices like keyboard, mouse, stylus etc.
- UI core – managing rendering of bitmap images (including compressed raster images like JPEG), vector graphics, text and animations.
- Media – playback of MP3, WMA Standard, WMV7, WMV8 and WMV9/VC-1 streams.
- XAML – to allow the UI layout to be created using XAML mark-up language.

To conclude, Silverlight is a powerful framework which supports multimedia, graphics, animations interactivity and playback of WMV, WMA and MP3 media content into a single runtime environment. This might be interesting for the representation of all the multimedia content that can be associated to an UmlModel.

6.2.4 Comparison: ASP.NET MVC Framework vs. Web Forms Applications vs. Silverlight

Microsoft has tried to apply windows form model development for web application development. For simulating windows form, webforms has introduced event-driven, Viewstate and Postback approach. The result is that webforms struggles with the stateless nature of the Internet. Developers don't have the control of the rendering HTML of the web forms. They don't have the Server controls that render html with mixed inline style and deprecated tags that do not follow standards, either.

Another problem with Web Forms is the integration of JavaScript frameworks due to the naming conventions of rendered HTML.

The site life cycle of the Web Form is too elaborate and complex, and has the tightly coupling between all the things in the ASP.net framework. A single class is used for both displaying output and handling user input. Therefore it makes unit testing almost an impossible task. Nowadays in modern software development unit testing is crucially important, especially when following agile methodology and practices. Because of the stateless of the web, Events, Postbacks and Viewstate are not the most efficient way.

ASP.NET MVC simplifies the complex parts of ASP.net Web Forms without compromising the flexibility and power of ASP.NET platform.

Unit testing facility, clear separation of concerns and more control over the HTML and URLs are the main advantages of the MVC models. The MVC model does not enable full control over the application and html rendered by the Views, because it doesn't use Postbacks, Viewstate, Server controls, or server-based forms. Instead of file-name extensions used by the Web Form model, MVC model is using Representational state transfer (REST) based URLs so that it is possible to make search engine optimization (SEO) URLs that are published by the application. At the moment MVC supports solely .Net Framework 3.5.

Comparing the advantages and disadvantages of MVC and Web Forms, we can distinguish [54]:

6.2.4.1 PROs of MVC Model

1. Enable clean separation of concerns (SoC).
2. Enable full control over the rendered HTML.
3. Enable Test Driven Development (TDD) (built with TDD in mind).
4. SEO and REST friendly URL.

5. Easy integration with JavaScript frameworks.
6. Support third-party view engines such as NVelocity, Brail, NHaml.
7. No ViewState andPostBack events.
8. Follows the stateless nature of web.
9. Extensible and Pluggable framework.
10. Ideal platform for Web 2.0 applications.

6.2.4.2 PROs of Web Form Model

1. Provides RAD development.
2. Easy development model for heavy data-driven LOB applications.
3. Provides rich controls.
4. Familiar model for windows form developers.

The question of which of both is better to use has been answered by one of the creators of ASP.NET MVC Framework, [Stephen Walther](#) [12, 13]:

Q.: "When should we use MVC (*ASP.NET MVC Framework*) versus Web Form Applications?"

R.: "In general, if you want to work directly with HTML, JavaScript, and CSS, then MVC is the way to go."

In my opinion, *ASP.NET MVC* is much more productive than Web Forms. The structure follows a natural order and allows you to adopt a REST architecture.

In fact, most of the recent Web frameworks such as *Ruby On Rails* follows this model. Actually, it is clearly the latter in which ASP.NET MVC has inspired.

In the other hand, we can compare the pros of Silverlight vs. ASP.NET Web Forms and with Web Forms [14].

6.2.4.3 PROs Silverlight vs. ASP.NET Web Forms

1. **No ViewState:** The No ViewState also applies to Silverlight. Silverlight brings the "desktop" to the end user and there is no ViewState that is used in Silverlight.
2. **Faster server-side & client-side:** Depending on the point of view, Silverlight is faster on the client/server side as it is compiled in a .NET subsystem of Silverlight. Also, access to multithreading, complex data structures and LINQ is allowed, among many other things. The performance is a lot more efficient compared to AJAX/JavaScript or to ASP.NET because of the client execution as some of the items that normally are handled in a server BLL can be brought down directly to the client.
3. **Simplified model for multiple related views:** The complete separation of the data and the UI is supported by Silverlight. Taking this further by just creating separate views for say another consumer of Silverlight is pretty powerful. The same MVC/MVP pattern can be applied inside Silverlight and achieve this level of abstraction.
4. **Unit Testable:** A Unit Test framework is included as well. It is available for download here: <http://code.msdn.microsoft.com/silverlightut/>

5. **Challenges if you are not running IIS 7:** Silverlight doesn't require running on a specific server: IIS 6, IIS 7 or Apache. Compared to [ASP.NET](#) MVC this is an advantage that Silverlight has.
6. **Client Caching:** ASP.NET Web Forms or MVC the cache is done on the server. In comparison, Silverlight allows caching on the client via Isolated Storage, which can be, if needed, increased to hundreds of Megs. This feature enables applications to perform with fast speed, and without slowing down the hosting server.

6.2.4.4 CONs Silverlight vs. ASP.NET Web Forms

1. **Difficult to convert existing code:** Silverlight differs greatly when it comes to programming platforms of either ASP.NET Web Forms or MVC. Lot of the code will not convert. In majority of the cases of replacing large modules inside the existing ASP.NET site a complete re-architecture is required.
2. **Quality of SEO:** Silverlight is still well behind in search engine optimizing. For example, Google started several months ago to crawl the SWF files; this is not done yet in Silverlight. What it is possible to do Silverlight SEO is the basic tricks of describing the meta data tags properly around the plug in.
3. **Data Access:** Data access is limited to Web Services/WCF/ADO.NET Data Services. Direct calls to a database via ADO.NET or stored procedures are not possible.
4. **Security:** Silverlight runs on the client. Some of the data accesses techniques do not support full Web Service (WS) standard security. Therefore, you have to write it by yourself. The XAML code is quite insecure, as few applications have their Intellectual Property in their UI. For example by using the Silverlight Spy that can be very easily reverse engineered. It is better to encrypt or obfuscate Silverlight assemblies before letting them off in the wild as Silverlight is a little less secure than an ASP.NET MVC application.

In conclusion, even though Silverlight and ASP.NET MVC are two completely different technologies, they have many things in common. The main difference is that the Silverlight does have its shade of meaning with data access and security that developer has to worry about; with the consequent increase of performance but the loose of simplicity.

6.3 Object-Relational Mapping in ADO.NET

Object-Relational Mapping (OR-M) is a programming technique for converting data between incompatible type systems in relational databases and object-oriented programming languages.

We have been studying the use of O-RM (Object-Relational Mapping) in order to reduce costs under ADO.NET [20].

ADO.NET was designed to support the disconnected data architecture. By raising the level of abstraction for data programming, it helps to exclude the impedance discrepancy between data models as well as eliminate the aforementioned between programming languages that

the developers of applications would otherwise have to cope with. Two innovations that make this possible are ADO.NET Entity Framework and Language- Integrated Query [15].

The goal of the ORMs is to extend the level of abstraction for database programming, which completely removes the impedance mismatch between data models and development languages that programmers use to write software applications.

In this study, the follow approaches ones are raised:

- ➔ Data Sets.
- ➔ Linq to SQL.
- ➔ Entity Framework.
- ➔ Comparison.

6.3.1 Data Sets

A data set (or dataset) is a collection of data, usually presented in tabular form. In ADO.NET, *DataSet* objects are a group of classes which describes a simple relational database in memory. These classes' follows a hierarchy in which the *DataSet* object represents a scheme which contains tables (*DataTable*) and relationships (*DataRelation*) between those tables.

The advantage of the dataset is that they present a disconnected model from the database. The programmer doesn't have to take care of the database and the data is cached which makes the shorting and the filtering simple. Additionally, they present a strong way to pass the parameters to the database so SQL injection is reduced.

On the other hand, the main disadvantage is the performance. The datasets are complex. A way to solve this performance issues can be minimize with the use of *typed datasets*. The typed dataset are more efficient due they take from the database just what they need while the normal dataset creates all the objects (*DataTables*, *DataRelation*, *DataColumns*...) so it takes it from the database to type the objects with the subsequent loss of performance.

6.3.2 Linq to SQL

Language Integrated Query (Linq) is a Microsoft .NET Framework component that adds native data querying capabilities to .NET languages.

Microsoft LINQ defines a set of proprietary query operators that can be used to query, filter and project data in arrays, enumerable classes, relational database, XML and third party data sources. Also it defines a Standard Query Operations interface called *IQueryable* and *IEnumerable* which can be implemented over any data source such us LINQ to Objects, LINQ to XML, LINQ to DataSet, LINQ to Entities, LINQ to SQL...

The LINQ to SQL provider gives LINQ the possibility to be used to query SQL Server databases as well as SQL Server Compact databases. Unlike other providers in which the data source is in the same machine, LINQ to SQL provider has to query against a remote SQL Server. For this, the LINQ queries are mapped to an SQL query.

Visual Studio 2008 contains a mapping designer that can be utilized when establishing the mapping between the data schemas in the object as well as the relational domain. It is able to automatically create the compatible classes from the database schema. Visual Studio 2008 also

allows manual editing which enables the creation of a different view by using only a subset of the tables or columns in a table.

So with this the programmer saves loads of time mapping the database to the objects. The amount of code to connect to the database is significantly reduced - to one or two lines making code much more readable and removing coding errors.

The problem with LINQ is that can suffer from low performance, high load applications if you have to operate with large amounts of data before a save.

6.3.3 Entity Framework

ADO.NET Entity Framework (EF) is an improvement of the datasets in which you work straight forward with the data. The EF exists as a new part of the ADO.NET family of technologies which abstracts the relational (logical) schema of the data that is stored in a database and presents its conceptual schema to the application.

The main advantage of using EF is that extends the level of abstraction for database programming, which completely removes the impedance mismatch between data models and development languages that programmers use to write software applications.

EF allows developers to write less data access code, reduces maintenance, and abstracts the structure of the data into a more business-friendly manner. It can also help to reduce the number of compile-time errors since it generates strongly typed classes from the conceptual model.

6.3.4 Comparison: Data Set vs. Entity Framework vs. Linq to SQL

Usually, due to performance reasons, it is recommended the use of Datasets when they are going to exist transactions that require the insertion of many rows [21, 22, 23, and 24].

Between the use of Linq to SQL and Entity framework, at the beginning we decided to use LINQ to SQL because Entity framework doesn't have implemented some small features. But, according to the recommendations of Microsoft described above and bearing in mind that these not implemented features are not functional, we are going to use Entity Framework.

If you are writing an application that requires any of the following features, you should use the ADO.NET Entity Framework [25]:

- The ability to define more flexible mapping to existing relational schema, for example:
 - **Mapping a single class to multiple tables.**
 - Mapping to different types of inheritance.
 - **Directly Modelling Many to Many relationships.**
 - Mapping to an arbitrary query against the store.
- The ability to query relational stores other than the Microsoft SQL Server family of products.
- The ability to share a model across Replication, Reporting Services, BI, Integration Services, etc.
- A full textual query language.
- The ability to query a conceptual model without materializing results as objects

Further on this document (see *Detailed Design* section) is described which approach is taken according to this study.

6.4 EXT Framework

Ext is a JavaScript library for building interactive web applications using techniques such as AJAX, DHTML and DOM scripting.

Ext JS was created by a coordinated team of programmers working together. Their goal was to achieve and be able to supply a consistent core user interface and interaction library, Because of the collaboration of various programmers, the code that is used for different kind of functionalities and widgets is, in comparison, more coherent than in some other JavaScript libraries that are more scattered. [16]

At the time of starting the project, EXT was the 2.x branch. The change from 1.x to 2.x was a major refactoring that included taking full advantage of the newly-created Component model, along with renaming many of the components to provide better organization. These changes have made 1.x code mostly incompatible with 2.x and vice versa. At the time of finishing the project, the 3.x branch was released correcting some of the bugs of the previous versions. Anyway, the 3.x branch is backwards-compatible.

The Ext library started out as an extension to the Yahoo User Interface library, providing an easy to use API (Application Programming Interface), and real world widgets. Even though the Yahoo User Interface tried to focus on the 'User Interface', it didn't contain much that was useful right out of the box.

It was not long before Ext had open-source contributors and developers working in their knowledge to turn the basic YUI extension into a powerful client-side application development library.

Ext provides an easy-to-use, rich user interface, similar than what you would find in a desktop application. This lets the web developers focus on the functionality of web applications instead of the technical issues.

Ext makes web application development simple by:

- Providing easy-to-use cross-browser compatible widgets such as windows, grids, and forms. The widgets are already fine-tuned to handle the intricacies of each web browser on the market, without us needing to change a thing.
- Interacting with the user and browser via the EventManager, responding to the users' keystrokes, mouse clicks, and monitoring events in a browser such as a window resize, or font size changes.
- Communicating with the server in the background without the need to refresh the page. This allows you to request or post data to or from your web server using AJAX and process the feedback in real time.

When we use Ext widgets, the browser compatibility is taken care of by the Ext library, so that each widget looks exactly the same in most of the popular browsers, which are:

- Internet Explorer 6+.
- Firefox 1.5+ (PC, Mac).
- Safari 2+.

- Opera 9+ (PC, Mac).

6.5 UML

Unified Modelling Language is the industry-standard language for the specification, visualization, construction, and documentation of the components of software systems. UML helps to simplify the process of software design, making a model for construction with a number of different views. One of the great merits of UML is the way it helps open up the development process with what are called use cases. These serve to identify principal roles (actors) in the system, boundaries, actions, and so on. Software developers need not write use cases; instead, other stakeholders can provide them. Integrated tools allow for use cases to be incorporated into the development process [17].

6.5.1 yUML

yUML [18] is a web service that provides us the ability to create UML diagrams from a simple by using a short and easy specific language and which can also be included in the URL.

With yUML it is possible to create class and use case diagrams. Activity and State diagrams will come.

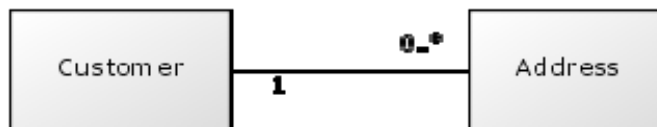
Subsequently they are listed some examples. In this examples are represented the notation, the way they can be called (URL) and the UML diagram that generates:

6.5.1.1 Class

[http://yuml.me/diagram/class/\[User\]](http://yuml.me/diagram/class/[User])

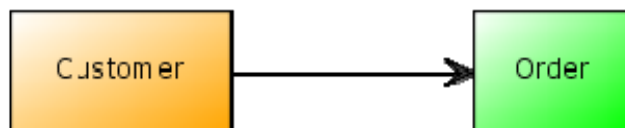


6.5.1.2 Cardinality



[http://yuml.me/diagram/class/\[Customer\]1-0..*\[Address\]](http://yuml.me/diagram/class/[Customer]1-0..*[Address])

6.5.1.3 Splash of Colour



[http://yuml.me/diagram/class/\[Customer{bg:orange}\]->\[Order{bg:green}\]](http://yuml.me/diagram/class/[Customer{bg:orange}]->[Order{bg:green}])

6.5.1.4 Aggregation

[http://yuml.me/diagram/class/\[Company\]<-1>\[Location\],\[Location\]+->\[Point\]](http://yuml.me/diagram/class/[Company]<-1>[Location],[Location]+->[Point])



6.5.1.5 Complex Example

[http://yuml.me/diagram/class/\[note: You can stick notes on diagrams too!{bg:cornsilk}\], \[Customer\]<-1-orders 0..*>\[Order\], \[Order\]++*-*>\[LineItem\], \[Order\]-1>\[DeliveryMethod\], \[Order\]*->\[Product\], \[Category\]<->\[Product\], \[DeliveryMethod\]^ \[National\], \[DeliveryMethod\]^ \[International\]](http://yuml.me/diagram/class/[note: You can stick notes on diagrams too!{bg:cornsilk}], [Customer]<-1-orders 0..*>[Order], [Order]++*-*>[LineItem], [Order]-1>[DeliveryMethod], [Order]*->[Product], [Category]<->[Product], [DeliveryMethod]^ [National], [DeliveryMethod]^ [International])

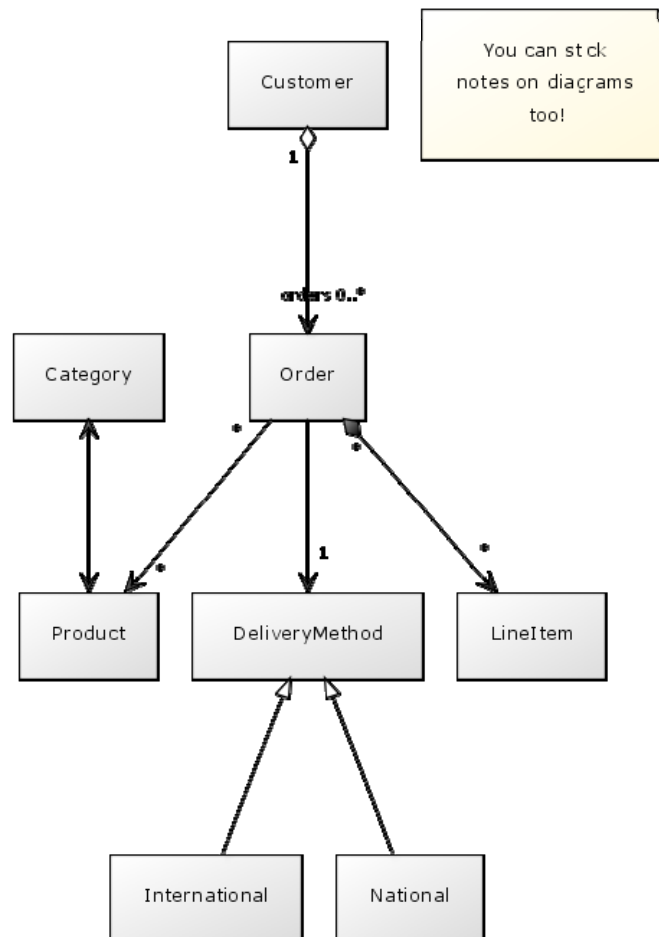


Illustration 10: yUml Examples.

7 Firs Approach: DotNetNuke

For the design of the external part at the beginning we were going to use a content management system. A CMS (See 'Study of the Content Management System (CMSs)') gives you an easy way to make the portal structure without wasting time in things that they are already done and focusing in the integration between the providing information engine systems.

Initially, as it is described next, a proof of concept were done in which an *Architecture*, a representation of the *Conceptual Model* and a basic *GUI* was proposed with its *Implementation* details. Finally a *Conclusion* is presented with the decision why we haven't used any CMS.

7.1 Architecture

The architecture used by the DotNetNuke is three-tier architecture in which a Presentation, a Business and a Data Layer are clearly defined.

The DotNetNuke deployment can have many portals. Each portal can have many pages in which each page have one or more modules.

The Presentation Layer corresponds with the user interface and represents the look of the module. The Business Layer contains the code of the module which communicates with the Data Access Layer through the *SQL* and the *Data Provider*¹ [19]. In the data Access Layer is also defined the tables and the procedures needed for the module.

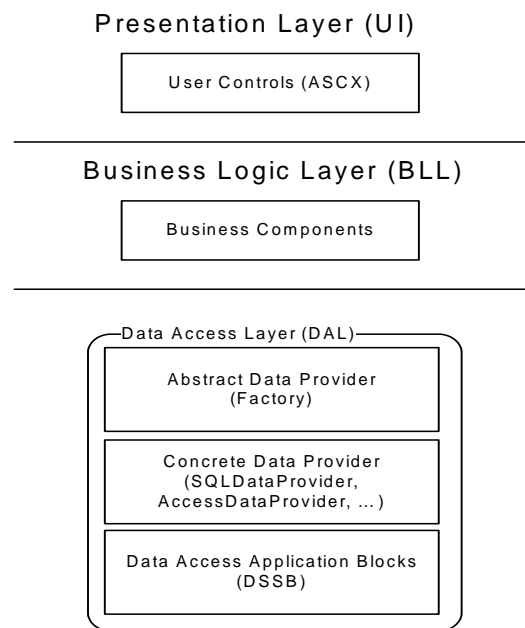


Illustration 11: DotNetNuke Architecture.

7.2 Conceptual Model

In this proof of concept a module for the Models Page (UmlModel) was implemented. This model will be allocated in a page which might contain a Forum module among others (Authentication...).

Several Forum modules where tested and none of them where what the client wanted for UmlModels (a forum similar than the *CodeProject.com* Site).

DotNetNuke 4.8.2 was deployed in a Windows XP with IIS 6.0 and Microsoft SQL Server 2000 (see *Illustration 13: DotNetNuke Deployment Diagram.*). Following is represented the conceptual model diagram which represents the Portal:

¹ Microsoft ASP.NET 2.0 includes a number of services that store state in databases and other storage media. For example, the session state service manages per-user session state by storing it in-process (in memory in the application domain of the host application), in memory in an external process (the "state server process"), or in a Microsoft SQL Server database, whereas the membership service stores user names, passwords, and other membership data in Microsoft SQL Server or Microsoft Active Directory.

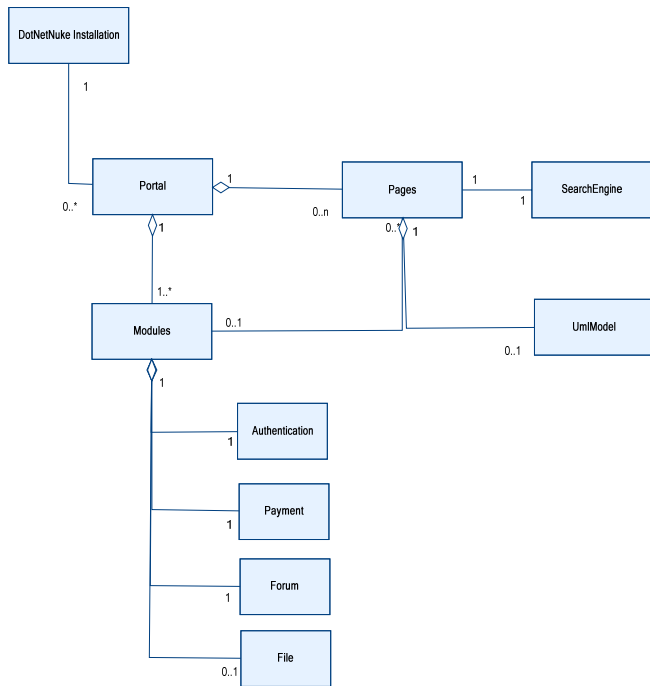


Illustration 12: Conceptual Model DotNetNuke.

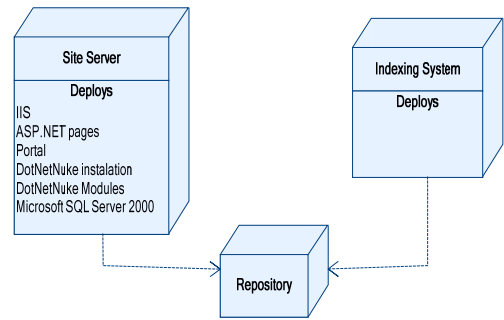


Illustration 13: DotNetNuke Deployment Diagram.

7.3 GUI

Following is presented the User Interface result for this proof of concept. In which a basic structure was presented with the data taken from the Data Access Layer.



Illustration 14: DotNetNuke: Screenshot: Presentation.



Illustration 15: DotNetNuke: Screenshot: Grid.

7.4 Implementation

The business and data layer was implemented for this proof of concept. In the latter, the Business Logic Layer and the Data Access Layer was implemented in detail while the GUI was just presented with a basic structure.

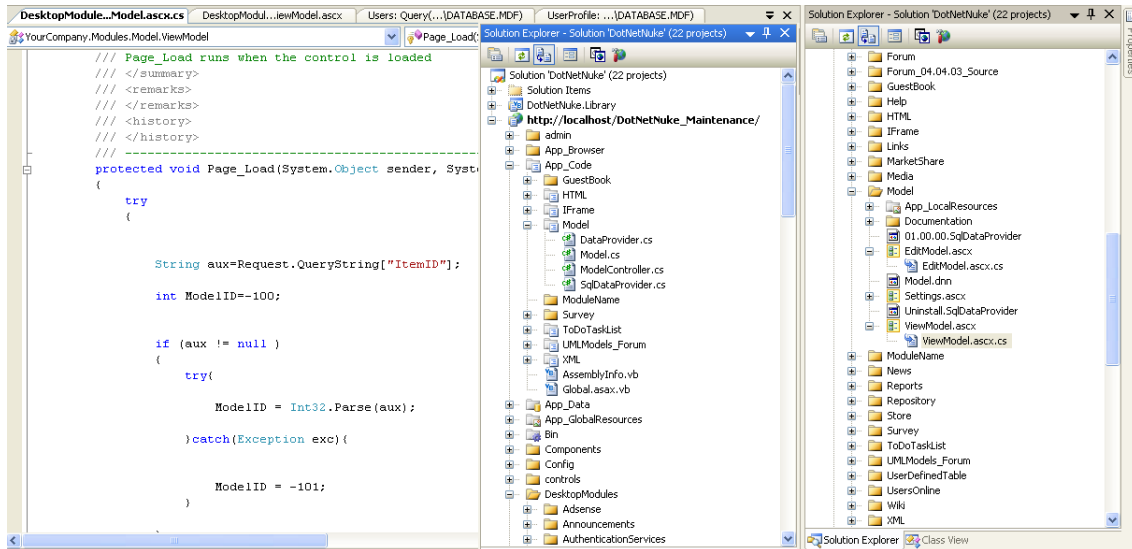


Illustration 16: DotNetNuke: Integration UmlModel Module with the Forum Module.

7.5 Conclusion

To conclude, we realise with this proof of concept that the CMS are not flexible enough for what we need. Besides, the complex of the Site lies on the presentation of the search result page and the models page. There are not going to be many pages to manage and the structure of the page won't change often so the use of the CMS is not the best option.

8 UmlModels Site

The aim of this project is to design the Web Site of UmlModels. This project is part of a complex system, which is specialized on retrieve, analyze and provide the semantic from software artefacts.

In this section is described the analysis and design of the Site as well as the functional description of the components. First are described the existing components within the Site. Second, is detailed the *Requirements* and the use cases of the system. Next, is described the way in which the information is presented to the world (*GUI*) as well as the *Architecture* and the way we represented our model (*Conceptual Model*). Finally, is specified the design (*Detailed Design*) of the site which includes the way in which the information is stored.

8.1 System Context

UmlModels project have been working for many years on different areas for provide information-engineering software. All that effort is been gathered and will be provided to the whole Internet community through this Portal Web.

Existing technologies used by UmlModels are described as follow:

8.1.1 Current Technologies

Before this project was started, UmlModels developed technologies were:

- ➔ Crawler: Searches all around the Internet for artifacts such as images that might be relevant for UmlModels. Once they are selected, these artifacts will be processed to extract all the semantic within this component.
 - Image-processor: At the moment the crawlers are only downloading images which are processed, validated and indexed. The validation checks if the image contains a software diagram with an algorithm that knows how to interpret boxes, text and lines from an image and returns classes, attributes and relationships.
- ➔ Polar Retrieval: Finds similar artifacts to a concrete Query by finding a distance between the Query and the results. The functionality of the polar retrieval is based on graph comparison techniques, which allow as to retrieve similar models in which the semantic similarities are also included with a high performance. This is done by mapping the Query to XMI (or RSHP). There are two different techniques:
 - Similarity retrieve: This type of retrieve works better when the distance big.
 - Inclusion retrieve: Modification of the previous type of retrieve in which the performance it is improved for small distance.
- ➔ Indexer: The indexer knows how to store the information within a diagram in XMI (*XML Metadata Interchange*) or RSHP (*Relationships Information Representation Model*) format.
- ➔ Virtual Assistance: Helps to the user to create a query by interacting with him.

The component described above interacts with the Web Site components as it is represented in the following diagram:

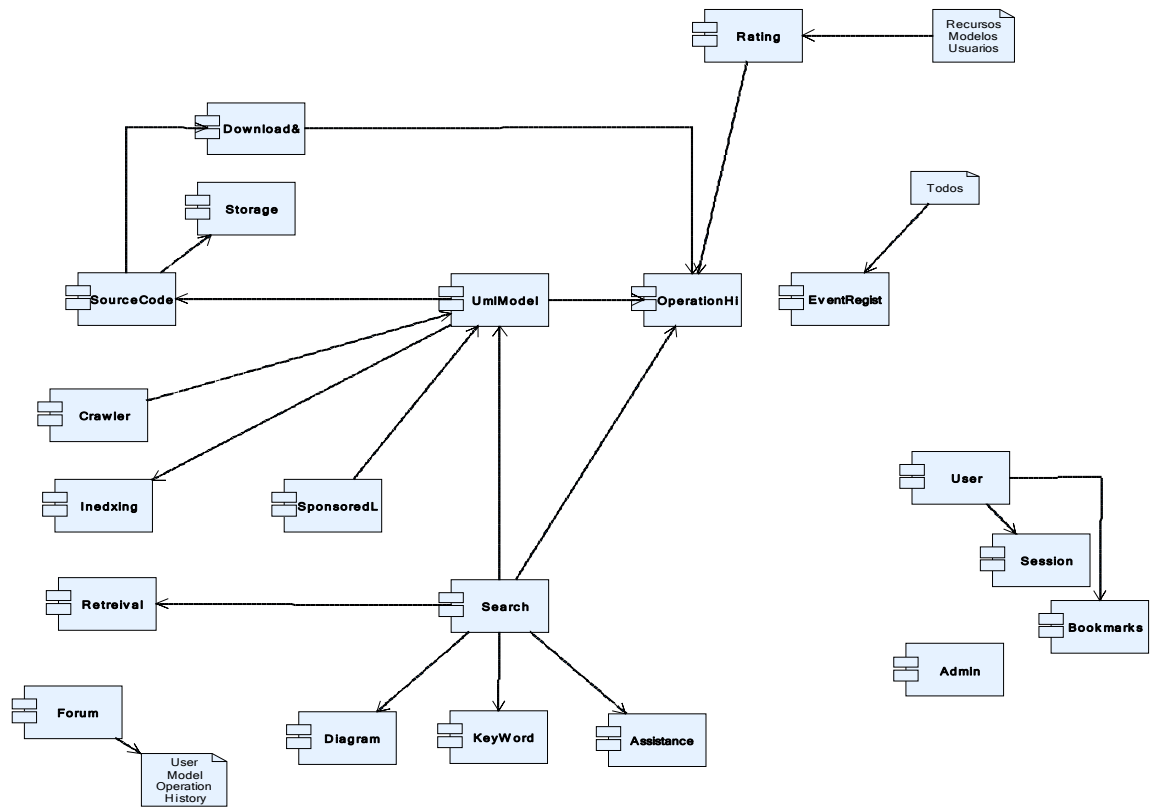


Illustration 17: Diagram Components

Subsequently is described the activity flow represented in the *Illustration 18: Gene-Sarson Diagram. Activity flow diagram*, which will be followed by the users in the next situations: Access, search and indexation of a Model. A user, typically, will access to a Model page either when a user performs a query through UmlModels search engine or through other one like Google (see “Direct access i and ii” in the illustration). Also, the models will be indexed by the crawler (see “Indexing i and ii”) and by the user (see “Indexing a and b”). As well, the users can perform searches through UmlModels search engine (see “Search 1, 2, 3 and 4”).

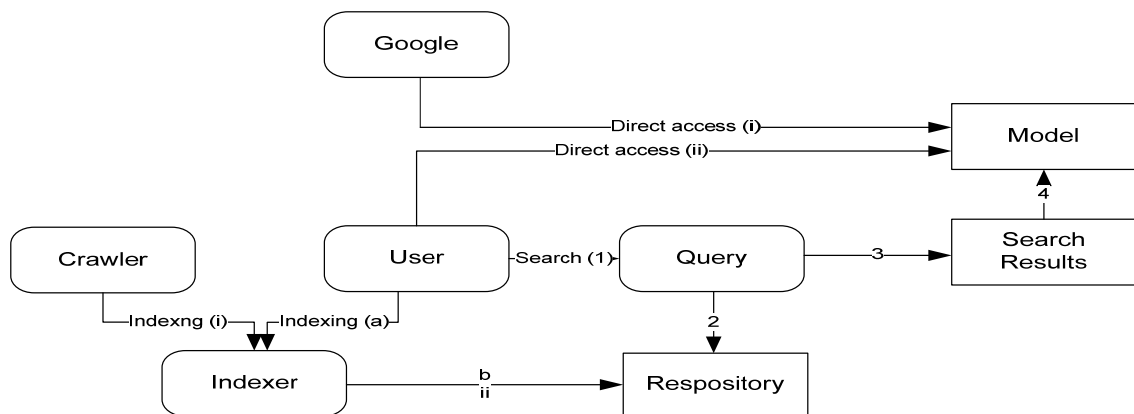


Illustration 18: Gene-Sarson Diagram. Activity flow diagram.

8.2 Use Case

In this section they are described the functionalities of the system in a high level of abstraction. Subsequently they are described the operations that the Registered and Unregistered users can perform.

8.2.1 Unregistered User

In the following diagram is possible to see all the operations that the Unregistered User also called visitor can perform and subsequently are described:

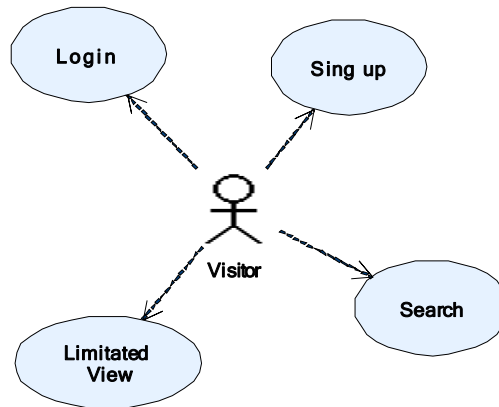


Illustration 19: Use Case diagram for Unregistered User.

UC-01 Login.	
Objective:	Allow unregistered users to log in into the system.
Pre-conditions:	The user must have an account.
Post-conditions:	Registered User view will be loaded.
Basic Scenario:	<ul style="list-style-type: none"> • User enters in any web page of www.umlmodels.com. • Click on Login link. • Enters user name and password.

Table 1: Use Case diagram for Unregistered User: Login.

UC-02 Signup	
Objective:	Allow to unregistered users to register into UmlModels.
Pre-conditions:	None.
Post-conditions:	A confirmation email is sent to the user.
Basic Scenario:	<ul style="list-style-type: none"> • User enters in any web page of www.umlmodels.com. • Click on Signup link. • Enters personal information. • Confirmation of email by clicking in the link sent by email.

Table 2: Use Case diagram for Unregistered User: Signup.

UC-03 Search	
Objective:	Allow to the unregistered users to search for Models.
Pre-conditions:	None.
Post-conditions:	UC-05 View Limited Search Result
Basic Scenario:	<ul style="list-style-type: none"> • User enters in any web page of www.umlmodels.com. • Enters the query. • Press enter or click on the search button.

Table 3: Use Case diagram for Unregistered User: Search.

UC-04 View Limited Model	
Objective:	Allows to the unregistered users to view a limited information of a requested model.
Pre-conditions:	None.
Post-conditions:	None.
Basic Scenario:	<ul style="list-style-type: none"> • User enters in a Models web page.

Table 4: Use Case diagram for Unregistered User: View Limited Model.

UC-05 View Limited Search Result	
Objective:	Allows to the unregistered users to view limited information of the search result page.
Pre-conditions:	UC-03 Search.
Post-conditions:	None.
Basic Scenario:	<ul style="list-style-type: none"> • User enters in any web page of www.umlmodels.com. • Enters the query. • Press enter or click on the search button.

Table 5: Use Case diagram for Unregistered User: View Limited Search Result.

8.2.2 Registered User

In the following diagram is possible to see all the operations that the Registered User can perform and subsequently are described:

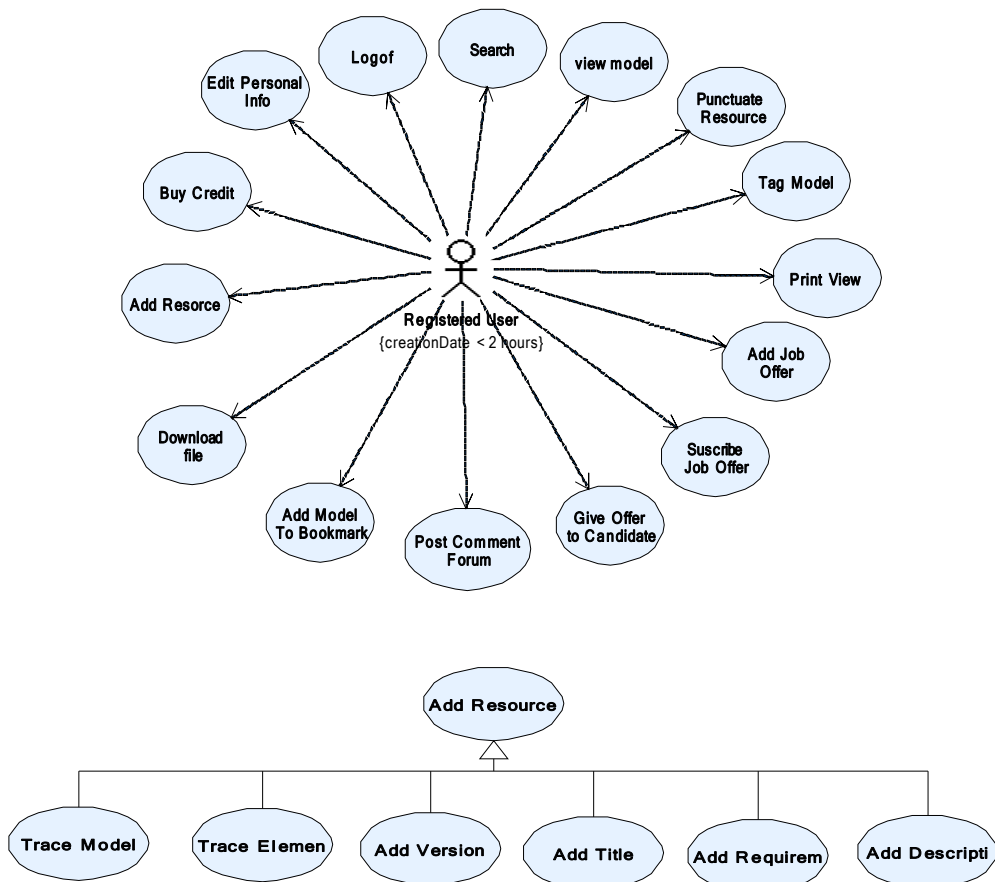


Illustration 20: Use Case diagram for Registered User.

UC-06 Logoff	
Objective:	Allow to the registered users to close the session on the server.
Pre-conditions:	UC-01 Login.
Post-conditions:	The unregistered view will be loaded.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on Logoff link

Table 6: Use Case diagram for Registered User:

UC-07 Search	
Objective:	Allow to the registered users to perform searches.
Pre-conditions:	UC-01 Login.
Post-conditions:	Search results are listed.
Basic Scenario:	<ul style="list-style-type: none"> • User enters in any web page of www.umlmodels.com. • Enters the query. • Press enter or click on the search button.

Table 7: Use Case diagram for Registered User:

UC-08 View Model	
Objective:	Allow to the registered users to view a requested model.
Pre-conditions:	UC-01 Login.
Post-conditions:	None.
Basic Scenario:	<ul style="list-style-type: none"> • User enters in a Models web page.

Table 8: Use Case diagram for Registered User: View Model.

UC-09 Punctuate Resource	
Objective:	Allow to the registered users to punctuate any resource.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	<ul style="list-style-type: none"> • The punctuation will be stored in the data base. • A message of confirmation will be prompted.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the punctuation field and introduces the punctuation for that resource.

Table 9: Use Case diagram for Registered User: Punctuate Resource.

UC-10 Add Version	
Objective:	Allow to the registered users to add a new version of a model.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New version will be added to the model.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the add version button. • Enters required data. • Save pressing the confirmation button.

Table 10: Use Case diagram for Registered User: Add Version.

UC-11 Add Title	
Objective:	Allow to the registered users to add a title resource.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New title will be added to the model.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the add title button. • Enters required data. • Save pressing the confirmation button.

Table 11: Use Case diagram for Registered User: Add Title.

UC-12 Add Description	
Objective:	Allow to the registered users to add a description resource.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New description will be added to the model.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the add description button. • Enters required data. • Save pressing the confirmation button.

Table 12: Use Case diagram for Registered User: Add Description.

UC-13 Add Requirement	
Objective:	Allow to the registered users to add a requirement resource.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New requirement will be added to the model.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the add requirement button. • Enters required data. • Save pressing the confirmation button.

Table 13: Use Case diagram for Registered User: Add Requirement.

UC-14 Tag Model	
Objective:	Allow to the registered users to tag a Model.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New tag will be added to the model.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the add tag button. • Enters required data. • Save pressing the confirmation button.

Table 14: Use Case diagram for Registered User: Tag Model.

UC-15 Print View	
Objective:	Allow to the registered users to view the model in a print view.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	None.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the print view button.

Table 15: Use Case diagram for Registered User: Print View.

UC-16 Add Job Offer	
Objective:	Allow to the registered users to add an offer.
Pre-conditions:	UC-01 Login.
Post-conditions:	New offer will be added to the bank job.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the add job offer button. • Enters required data. • Save pressing the confirmation button.

Table 16: Use Case diagram for Registered User: Add Job Offer.

UC-17 Subscribe Job Offer	
Objective:	Allow to the registered users to subscribe to an offer to perform it.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • Selects an offer.
Post-conditions:	Email will be sent to the owner of the offer.
Basic Scenario:	Users clicks on the subscribe button of an offer.

Table 17: Use Case diagram for Registered User: Subscribe Job Offer.

UC-18 Give Offer to Candidate	
Objective:	Allow to the registered users owner of an offer to give an offer to a subscribed user.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-17 Subscribe Job Offer
Post-conditions:	Email will be sent to the subscribed user.
Basic Scenario:	<ul style="list-style-type: none"> • The owner of the offer confirms a subscribed user.

Table 18: Use Case diagram for Registered User: Give Offer to Candidate.

UC-19 Post Comment Forum	
Objective:	Allow to the registered users to post a comment of a Model.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New comment will be added to the forum of the model.
Basic Scenario:	<ul style="list-style-type: none"> • User clicks on the comment button. • Enters required data. • Save pressing the confirmation button.

Table 19: Use Case diagram for Registered User: Post Comment Forum.

UC-20 Add Model To Bookmark	
Objective:	Allow to the registered users to add a bookmark of a model.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	New Bookmark will be added to the bookmarks of the user.
Basic Scenario:	<ul style="list-style-type: none"> • The user press on the add model to bookmarks button.

Table 20: Use Case diagram for Registered User: Add Model To Bookmark.

UC-21 Download File.	
Objective:	Allow to the registered users to download a file.
Pre-conditions:	<ul style="list-style-type: none"> • UC-01 Login. • UC-08 View Model.
Post-conditions:	Money might be charged.
Basic Scenario:	<ul style="list-style-type: none"> • The user press con the download button. • If the file is not free a confirmation with a warning of the charge for the download will be prompted.

Table 21: Use Case diagram for Registered User: Download File.

UC-22 Buy Credit	
Objective:	Allow to the registered users to buy credit.
Pre-conditions:	UC-01 Login.
Post-conditions:	The balance of the user will be increased.
Basic Scenario:	<ul style="list-style-type: none"> • The user clicks on the buy credit button. • The user is forward to the UmlModels PayPal page.

Table 22: Use Case diagram for Registered User: Buy Credit.

UC-23 Edit Personal Info	
Objective:	Allow to the registered users to edit his personal information.
Pre-conditions:	UC-01 Login.
Post-conditions:	The personal information will be updated on the database.
Basic Scenario:	<ul style="list-style-type: none"> • The user clicks on the edit personal information link. • His personal information will be prompted. • The user updates the information he considers. • Save pressing the confirmation button.

Table 23: Use Case diagram for Registered User: Edit Personal Info.

8.3 Requirements

In this section are described the requirements that this project is following for building the Portal. These requirements were done in the analysis phase of my project in the period in which I was in Finland with Juan Llorens. Subsequently new requirements born, some of them affected to my project --and they were updated according with the requirements included in this section-- and some other will affect to the site in the future. For this reason, in they are included the last version of these requirements in: **Appendix B: UMLModels Requirements by Gonzalo Génova (Spanish)**.

There are two different kinds of requirements:

- Capacity: Functions and operations required by the client to solve a problem or achieve an objective. Describes an operation or sequence of operations.
- Restriction: Restrictions imposed by the client on how the problem is solved or the goal is achieved. Restricts the manner in which software is constructed or operated, without altering or describe the capabilities of the software.

8.3.1 2-UML-00012: UMLModels Site

This package doesn't have any requirements.

8.3.1.1 2.1-GR-00001: General Requirements

2.1-GR-00002: Search Result			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G. Suárez de Tangil
Priority:	1	Kind:	Capacity
Description:	The system should allow the user to make searches, returning as a result Web pages that contain software models.		

Table 24: Requirements. General Requirements.

2.1-GR-00002.1: Search Result			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The search results will be shown as a list of small images (thumbnails), of variable size. Near the image, a small description will be shown.		

Table 25: Requirements. General Requirements.

2.1-GR-00003: Description of the System			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The system will be a Web portal. The site must have a search engine.		

Table 26: Requirements. General Requirements.

2.1-GR-00077: Url			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Restriction
Description:	Each web page must have a static URL even though if the content is generated dynamically.		

Table 27: Requirements. General Requirements.

2.1-GR-00076: Models Page			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Restriction
Description:	Each search must return links to model's web page		

Table 28: Requirements. General Requirements.

2.1-GR-00078: Models Page			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Restriction
Description:	Each model must be in one and only one web page.		

Table 29: Requirements. General Requirements.

2.1-GR-00079: Search Results			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The position of the results of the search results must be sorted according to a rank that each model has.		

Table 30: Requirements. General Requirements.

2.1-GR-00080: Ranking			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The calculation of Rank must be based on the model's number of XMIs, number of source files, and number of executable components, number of trazability elements, number of documentation files and number of post in the forum. As well as if exist or not image and if exist or not comment.		

Table 31: Requirements. General Requirements.

2.1-GR-00090: Technology of the database			
Date:	19/10/2008	State:	G.S.T
Priority:	1	Kind:	Restriction
Description:	The technology of the database will be Microsoft SQL Server 2005 64 bits.		

Table 32: Requirements. General Requirements.

2.1-GR-00091: Technology of the database			
Date:	19/10/2008	State:	G.S.T
Priority:	1	Kind:	Restriction
Description:	The technology of the database will be Microsoft SQL Server 2005 64 bits.		

Table 33: Requirements. General Requirements.

2.1-GR-00092: Law of Personal Data Protection			
Date:	19/04/2008	State:	G.S.T
Priority:	2	Kind:	Restriction
Description:	<p>It will be respected the law 15/1999 of December 13 for the Protection of Personal Data (LOPD) in regard to the information contained in user profiles (and future histories of operations / user searches).</p> <p>(Royal Decree 1720/2007 of December 21, by approving the development regulation of the Law 15/1999, of December 13, the protection of personal data).</p> <p>(Law 34/2002 of July 11, services of information society and electronic commerce).</p>		

Table 34: Requirements. General Requirements.

8.3.1.2 *2.2-RO-00003: Roles*

2.2-RO-00019: Administrator			
Parent:	2.2-RO-00003: Roles	Kind:	2.1-General Requirements
Date:	05/10/2007	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The site must have an Administrator role.		

Table 35: Requirements. Roles

2.2-RO-00020: Unregistered users			
Parent:	2.2-RO-00003: Roles	Kind:	2.1-General Requirements
Date:	05/10/2007	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The site must have an Unregistered User role. This role will also be called Visitor.		

Table 36: Requirements. Roles

2.2-RO-00021: Registered User			
Parent:	2.2-RO-00003: Roles	Kind:	2.1-General Requirements
Date:	05/10/2007	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The site must have a Registered User role. This role will have the following sub-roles: Client, Operator and Administrator		

Table 37: Requirements. Roles

2.2-RO-00047: Unregistered User			
Parent:	2.2-RO-00020: The site must have an Unregistered User role	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The unregistered user must have access to the free services.		

Table 38: Requirements. Roles

2.2-RO-00048: Unregistered User			
Parent:	2.2-RO-00020: The site must have an Unregistered User role	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The unregistered user must have access to the free models		

Table 39: Requirements. Roles

2.2-RO-00049: Registered User			
Parent:	2.2-RO-00021: The site must have a Registered User role	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The registered user must have access to the free services.		

Table 40: Requirements. Roles

2.2-RO-00050: Registered User			
Parent:	2.2-RO-00021: The site must have a Registered User role	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The registered user must have access to the free models.		

Table 41: Requirements. Roles

2.2-RO-00051: Registered User			
Parent:	2.2-RO-00021: The site must have a Registered User role	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The registered user must have access to the added value area.		

Table 42: Requirements. Roles

2.2-RO-00093: Profile			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The user will have a profile with his personal information: Email, name, last name, telephone number, balance, karma, member since and a personal URL.		

Table 43: Requirements. Roles

8.3.1.3 -FREE-00038: Free Services

2.3.1-FREE-00039: Search Model			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Make search of models is a free service.		

Table 44: Requirements. Free Services.

2.3.1-FREE-00040: Images			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View images of the UML diagram from the models searched is a free service.		

Table 45: Requirements. Free Services.

2.3.1-FREE-00042: Number of Files			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the number of files uploaded in each category of the added value area of a model is a free service.		

Table 46: Requirements. Free Services.

2.3.1-FREE-00043: Descriptions			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.3.1-Free Services
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the first characters of a model's commentary is a free service		

Table 47: Requirements. Free Services.

2.3.1-FREE-00044: Posts			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the title of the posts of a model's forum is a free service		

Table 48: Requirements. Free Services.

2.3.1-FREE-00045: Title			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the title of a model is a free service.		

Table 49: Requirements. Free Services.

2.3.1-FREE-00046: Key Words			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	17/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the Key words is a free service.		

Table 50: Requirements. Free Services.

2.3.1-FREE-00053: Hyperlink to the Source			
Parent:	2.3.1-FREE-00038: Free services	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the hyperlink to the source is a free service.		

Table 51: Requirements. Free Services.

8.3.1.4 2.3.2-AVA-00039: Added Value Area

2.3.2-AVA-00052: Added Value Area			
Parent:	2.3.2-AVA-00039: Added Value Area	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the commentaries of a model is an added value service.		

Table 52: Requirements. Added Value Area.

2.3.2-AVA-00054: Added Value Area			
Parent:	2.3.2-AVA-00039: Added Value Area	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	View the post of the forums is an added value service.		

Table 53: Requirements. Added Value Area.

2.3.2-AVA-00055: Added Value Area			
Parent:	2.3.2-AVA-00039: Added Value Area	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A post new comment into the forums is an added value service.		

Table 54: Requirements. Added Value Area.

2.3.2-AVA-00056: Added Value Area			
Parent:	2.3.2-AVA-00039: Added Value Area	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Download files uploaded to the models is an added value service.		

Table 55: Requirements. Added Value Area.

2.3.2-AVA-00057: Added Value Area			
Parent:	2.3.2-AVA-00039: Added Value Area	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Upload models and its content is an added value service.		

Table 56: Requirements. Added Value Area.

2.3.2-AVA-00058: Added Value Area			
Parent:	2.3.2-AVA-00039: Added Value Area	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Download the content of the model in XMI is an added value service.		

Table 57: Requirements. Added Value Area.

8.3.1.5 2.3.3-MSD-00023: Models Site Design

2.3.3-MSD-00074: Models Site Design			
Parent:	2.3.3-MSD-00023: Models Site Design	Kind:	2.3.2-Added Value Area
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The image with the UML diagram must be the most important element to show in the model's web page. All the rest of added value information should be situated in a secondary view.		

Table 58: Requirements. Models Site Design.

8.3.1.6 2.3-SS-00004: Site Sections

2.3-SS-00037: Site Sections			
Parent:	2.3-SS-00004: Site Sections	Kind:	2.1-General Requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The Site must offer a free service available to anyone. And other added value area to those users that have paid.		

Table 59: Requirements. Site Sections.

8.3.1.7 2.4-MCP-00005: Human Classification Process

2.4-MCP-00075: Human classification Process			
Parent:	2.4-MCP-00005: Human Clasification Process	Kind:	2.4-Manual Classification Process
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Models can be indexed by the owner of the model by introducing some key words.		

Table 60: Requirements. Human Classification Process.

8.3.1.8 2.5-MEN-00005: Economical business model

2.5-MEN-00024: Added Value Area			
Parent:	2.5-MEN-00028: Each model can be marked as free model or not free model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Only registered users must have access to the added value area (area of value-added) of the models marked as not free models		

Table 61: Requirements. Economical business model

2.5-MEN-00026: Credit			
Parent:	2.5-MEN-00032: Each user must have credit balance in is count	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	To register, the user must make a payment of an amount of money.		

Table 62: Requirements. Economical business model

2.5-MEN-00027: Advanced Payment			
Parent:	2.5-MEN-00026: To register, the user must make a payment of an amount of money	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	This payment done in the register will be used like advance payment		

Table 63: Requirements. Economical business model

2.5-MEN-00028: Free Models			
Parent:	2.5-MEN-00005: Economic business model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Each model can be marked as free model or not free model.		

Table 64: Requirements. Economical business model

2.5-MEN-00029: Access to the Added Value area			
Parent:	2.5-MEN-00028: Each model can be marked as free model or not free model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	All users must have access to all the added value area of the free models and download all files without paying.		

Table 65: Requirements. Economical business model

2.5-MEN-00030: Downloadable Files			
Parent:	2.5-MEN-00028: Each model can be marked as free model or not free model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Only registered users with positive credit can download files uploaded for each download in the not free models.		

Table 66: Requirements. Economical business model

2.5-MEN-00031: Chargeable Resources			
Parent:	2.5-MEN-00028: Each model can be marked as free model or not free model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Each user owner of a model can decide if his model will be marked as free or not free model.		

Table 67: Requirements. Economical business model

2.5-MEN-00032: Balance			
Parent:	2.5-MEN-00005: Economic business model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Each user must have credit balance in is count.		

Table 68: Requirements. Economical business model

2.5-MEN-00033: Credit			
Parent:	2.5-MEN-00032: Each user must have credit balance in is count	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	When a registered user refills his credit, a bonus credit can be added to his account.		

Table 69: Requirements. Economical business model

2.5-MEN-00036: Payment			
Parent:	2.5-MEN-00005: Economic business model	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The site must have a payment method associated.		

Table 70: Requirements. Economical business model

2.5-MEN-00088: Chargeable Downloads			
Parent:	2.5-MEN-00032: Each user must have credit balance in is count	Kind:	2.5-Economic business model
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Each file downloaded form a not free model makes the credit of the user to be reduced into the cost of a download.		

Table 71: Requirements. Economical business model

8.3.1.9 2.6-MODELS-00006: Models

2.6-MODELS-00060: Models			
Parent:	2.6-MODELS-00006: Models	Kind:	2.6-Models
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A model can have other models.		

Table 72: Requirements. UmlModel

2.6.1-DIAGRAM-00059: Diagram			
Parent:	2.6-MODELS-00006: Models	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A model must have a diagram.		

Table 73: Requirements. UmlModel

2.6.1-DIAGRAM-00061: Image			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram can or cannot have an image.		

Table 74: Requirements. UmlModel

2.6.1-DIAGRAM-00062: Hyperlink To Source			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram must have a hyperlink to the source.		

Table 75: Requirements. UmlModel

2.6.1-DIAGRAM-00063: Hyperlink To Source			
Parent:	2.6.1-DIAGRAM-00062: A diagram must have an hyperlink to the source	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The hyperlink to the source has to be one of those types of links: To the registered user owner of the model (the one who uploaded it) or the source URL in which the model have been found.		

Table 76: Requirements. UmlModel

2.6.1-DIAGRAM-00064: Forum			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram must have a forum associated to it.		

Table 77: Requirements. UmlModel

2.6.1-DIAGRAM-00065: Title			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram can or cannot have a title.		

Table 78: Requirements. UmlModel

2.6.1-DIAGRAM-00066: Description			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram can or cannot have a description.		

Table 79: Requirements. UmlModel

2.6.1-DIAGRAM-00067: Key Words			
Parent:	2.1-GR-00001: General Requirements	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram can or cannot have description key words.		

Table 80: Requirements. UmlModel

2.6.1-DIAGRAM-00068: Trazability Elements			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram must have none, one or more trazability elements.		

Table 81: Requirements. UmlModel

2.6.1-DIAGRAM-00069: Trazability			
Parent:	2.6.1-DIAGRAM-00068: A diagram must have none, one or more trazability elements	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A trazability element has to be one of the next type elements: A link to another related model or a related artifact.		

Table 82: Requirements. UmlModel

2.6.1-DIAGRAM-00070: Artifact			
Parent:	2.6.1-DIAGRAM-00068: A diagram must have none, one or more trazability elements	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	An artifact has to be one of those type elements: Image, XMI, Additional Documentation, Source Code, Description, Title, Post, Tag or Requirement.		

Table 83: Requirements. UmlModel

2.6.1-DIAGRAM-00071: Additional Documentation			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram must have none, one or more documentation files.		

Table 84: Requirements. UmlModel

2.6.1-DIAGRAM-00073: Sponsored Links			
Parent:	2.6.1-DIAGRAM-00059: A model must have a diagram	Kind:	2.6.1-DIAGRAM
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A diagram must have none, one or more sponsored links		

Table 85: Requirements. UmlModel

2.6.1-DIAGRAM-00088: Punctuation			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Restriction
Description:	The owner of a resource won't be able to punctuate his own resources.		

Table 86: Requirements. UmlModel

2.6.1-DIAGRAM-00089: Versions			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Capacity
Description:	When a version is new, this won't inherit the related models neither the rate.		

Table 87: Requirements. UmlModel

2.6.1-DIAGRAM-00094: Chargeable Resources.			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Capacity
Description:	The chargeable elements are: The XMI, the Source Code and the additional documentation. The non-chargeable elements are: the title, the description, the images and the requirements. Thos chargeable elements are free if the owner decides it.		

Table 88: Requirements. UmlModel

2.6.1-DIAGRAM-00095: Punctuation			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Restriction
Description:	A user must download a resource before been able to rate it.		

Table 89: Requirements. UmlModel

2.6.1-DIAGRAM-00096: Report bad content			
Date:	19/04/2008	State:	G.S.T
Priority:	2	Kind:	Capacity
Description:	There will be an option to notify if the content is not proper. This will send an email to the user.		

Table 90: Requirements. UmlModel

2.6.1-DIAGRAM-00088: Storage of the Files.			
Date:	19/04/2008	State:	G.S.T
Priority:	1	Kind:	Capacity
Description:	<p>The files of a resource will be stored on the web server and they will follow hierarchy:</p> <p style="padding-left: 40px;">Resource\ModelId%1000\ResourceId\VersionId</p> <p>For instance:</p> <p style="padding-left: 40px;">Image\ModelId%10000\ModelId\VersionId</p>		

Table 91: Requirements. UmlModel

8.3.1.10 2.7-RR-00081: Reverse Requirements

2.7-RR-00082: Forums			
Parent:	2.7-RR-00081: Reverse Requirements	Kind:	2.7-Reverse requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	An unregistered user cannot write comments in the forums.		

Table 92: Requirements. UmlModel

2.7-RR-00083: Registered User			
Parent:	2.7-RR-00081: Reverse Requirements	Kind:	2.7-Reverse requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A registered user who is not owner of the model cannot index the model		

Table 93: Requirements. UmlModel

2.7-RR-00084: Nicks			
Parent:	2.7-RR-00081: Reverse Requirements	Kind:	2.7-Reverse requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	Cannot be two different users with the same nick.		

Table 94: Requirements. UmlModel

2.7-RR-00085: Nick			
Parent:	2.7-RR-00081: Reverse Requirements	Kind:	2.7-Reverse requirements
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	A user cannot change is nick.		

Table 95: Requirements. UmlModel

8.3.1.11 OT-00086: OTHER

OT-00087: Description			
Parent:	OT-00086: OTHER	Kind:	OTHER
Date:	19/04/2008	Author:	G.S.T
Priority:	1	Kind:	Capacity
Description:	If the registered user is the one of the users that uploaded the model, the description cannot be in blank.		

Table 96: Requirements. UmlModel

8.4 Architecture

As it has been explained, UmlModels will provide a search engine of artefacts for the software engineering community in which the semantic of the model associated to the artifact will be offered.

In each query a complex search is performed in which many elements take part on it: textual descriptions, relations between entities, attributes... The search can be performed in different formats (textual, graphics and many more can appear) and the resultant models that they have to be listed must match having in consideration semantic similarities.

8.4.1 Pagination of the Search Results

As it has been described in the *System Context*, our search engine will be performing complex retrieves against the database. In other words, we will have to deal with performance issues. One part of this complex mechanism is the pagination of the results.

When a user makes a query, all the resultant models that match with the query are ordered based on a ranking (not defined yet). Once they are ordered, just the first ones are returned to the user. And it is just when the user need the next term of results when they are provided.

It represents a challenge to do this process in an efficient way when thousands or even millions of models are stored on the database. Even more, when hundreds of users are connected performing queries, the concurrence can overwhelm our servers.

An efficient pagination is needed. While *MySQL* resolves the pagination problem in an efficient way because the pagination is native to the database, *SqlServer 2005* has to emulate it. For this, ASP.NET provides some mechanisms [26] that we consider they are not good enough since in some cases, depending on the implementation, all the data is retrieved when we only need a subset and in other cases the results are ranked every time the user needs another page of the search result.

In the cases in which the ordination criterion is always the same such as the date, it is easier to implement. But when the criterion depends on many attributes like the number of post, the ranking of the users, money associated to the model, number of visits... that they change every second then, a smart solution may be implemented.

The solution presented for UmlModels, as is represented in Illustration 21, is to add a cache of results per user connected to the server in which we store the IDs of the results of the query retrieved for the concrete query. To the user, we will only present the first N results. And if the user ask for more results for that query, then none access is performed to the database.

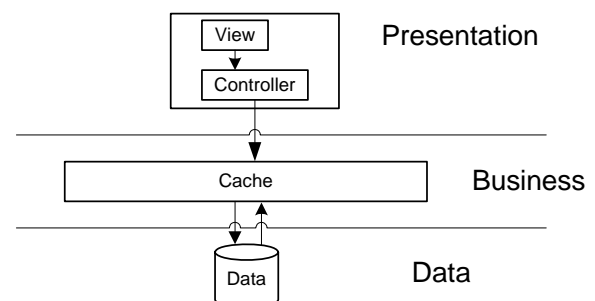


Illustration 21: Architecture: Cache of results

8.4.2 Architectural Layer

It has been deeply discussed the architecture that we are going to follow. It is clear that we have an architecture with a presentation layer in the **client** and a data layer in the **server**, but the controversy came when the pagination of the search results had to be resolved in a efficient way.

Originally, a complex and unnecessary architecture was proposed in which we have a shared huge cache with the last models queried and identified by a hash; solving concurrence problems and with object persistence (similar than *memcached*). So when a new user performs the same query, then the business layer will return it from the cache instead of the acceding to the database. We conclude that it is not necessary to reinvent the wheel when we can simplify our scheme to the point of use an existing data structure to store in the context of the server a list with the id's of the models per query performed. The positive point of this approach is that is the IIS the one that manage all the persistence of the data between connections for each user.

Finally, a new architecture made up of four layers was chosen based on the principle of simplicity with an N-Tier client-server architectural pattern:

- Presentation Tier:
 - Client: GUI Presentation (HTML).
 - Server: Logic Presentation: (ASPX, JavaScript).
- Controller Tier:
 - Web Component:
 - Client: *Web Browser, AJAX*.
 - Server: *Internet Information Server (IIS)*.
 - Business Controller: *ASP.NET MVC Controller*.
- Business Tier:
 - Searcher Component:
 - Processor: JSON to UML.
 - Executor: SQL Code.
 - Indexer Component: Windows Server indexing.
 - Polar Retrieval.
 - Data Access:
 - A list per user-session and query.
 - *ADO.NET Entity Framework*.
- Data Tier: *SQL Server 2005 database*.

The above described architecture is represented as follows and described subsequently:

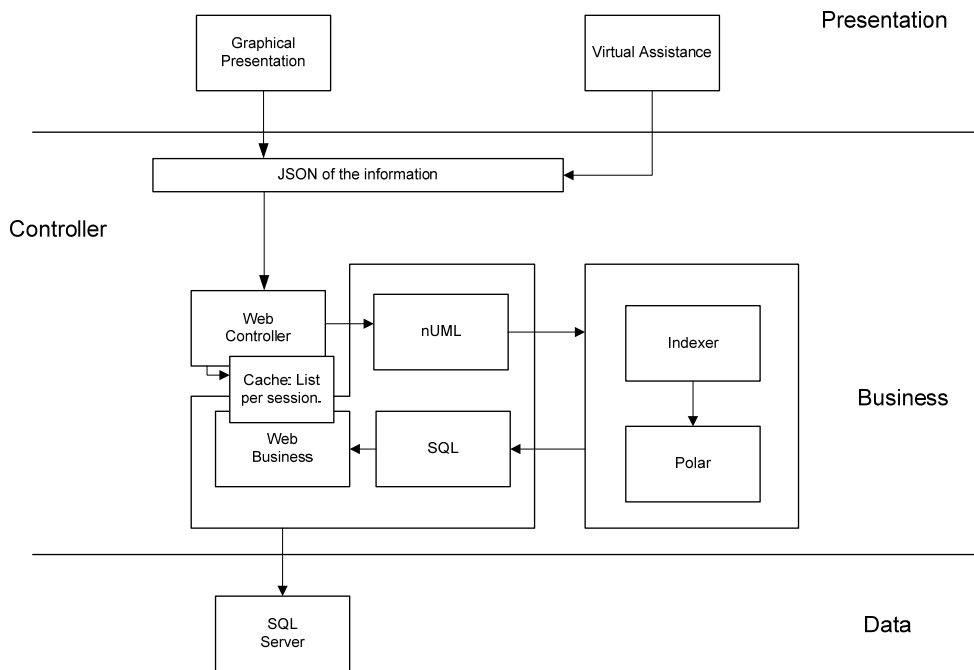


Illustration 22: Architecture: Layers.

The *client* will send to the server the query in JSON for: The **textual search**, the **diagrammer** and the **virtual assistance**. (All three are detailed subsequently).

In the web **server**: *NUML* code is generated (Out of this project see *System Context*), Indexation is performed: A polar retrieval (Out of this project see *System Context*) and SQL code is returned (Textual search is simulated for this project).

Once we have the SQL code, we can either obtain all the models from the SQL Server or obtain only the ID. If we obtain all the models, we could store in the cache all of them but this would cost lots of space; or select the top N models and the rest of the models will be thrown away, what would be a waste of CPU resources. If we obtain only the IDs, then we have to make another select to obtain the first N models what will two selects.

The running's that we did in our test environment shown that the response time for 150 threads performing selects against an indexed attribute of the *UmlModels* table (about 500 MB of size) last less than 1 ms. Considering that the selects are not expensive and since the information of all the results of a search (i.e.: 1000 results) of all the users performing searches at the same time (i.e: 100 users) can have a big enough size to be stored in the session of the server. Additionally, if the number of requests exceeds the maximum capacity of our infrastructure, it is easier to scale the database than the web server.

For all this, we consider that the best solution is to store only the IDs of all the search results in the session and make two selects: One to get first the IDs of the result et and the other to get the first N models to be returned to the user.

Considering the latter approach, following is detailed the interaction between the described tiers in the following scenarios: *Search Result* and *Models Page* request.

8.4.2.1 Search Result

In the search result, the user initiates a search by typing the query in the browser. This query is sent to the server as a HTTP GET or HTTP POST request with a pagination parameter start index. The controller attends the request and if it is a new query, then the controller ask to the business layer to return the results for that query. The controller stores that results and ask to the data layer for some results. The fist results are sent back to the user.

If the user wants to see more results, then a request is sent again to the server with the same query and the pagination index. The controller checks if cached results are stored in the session for that query. If they are, then the controller asks to the data layer for some results. The next results are sent back to the user.

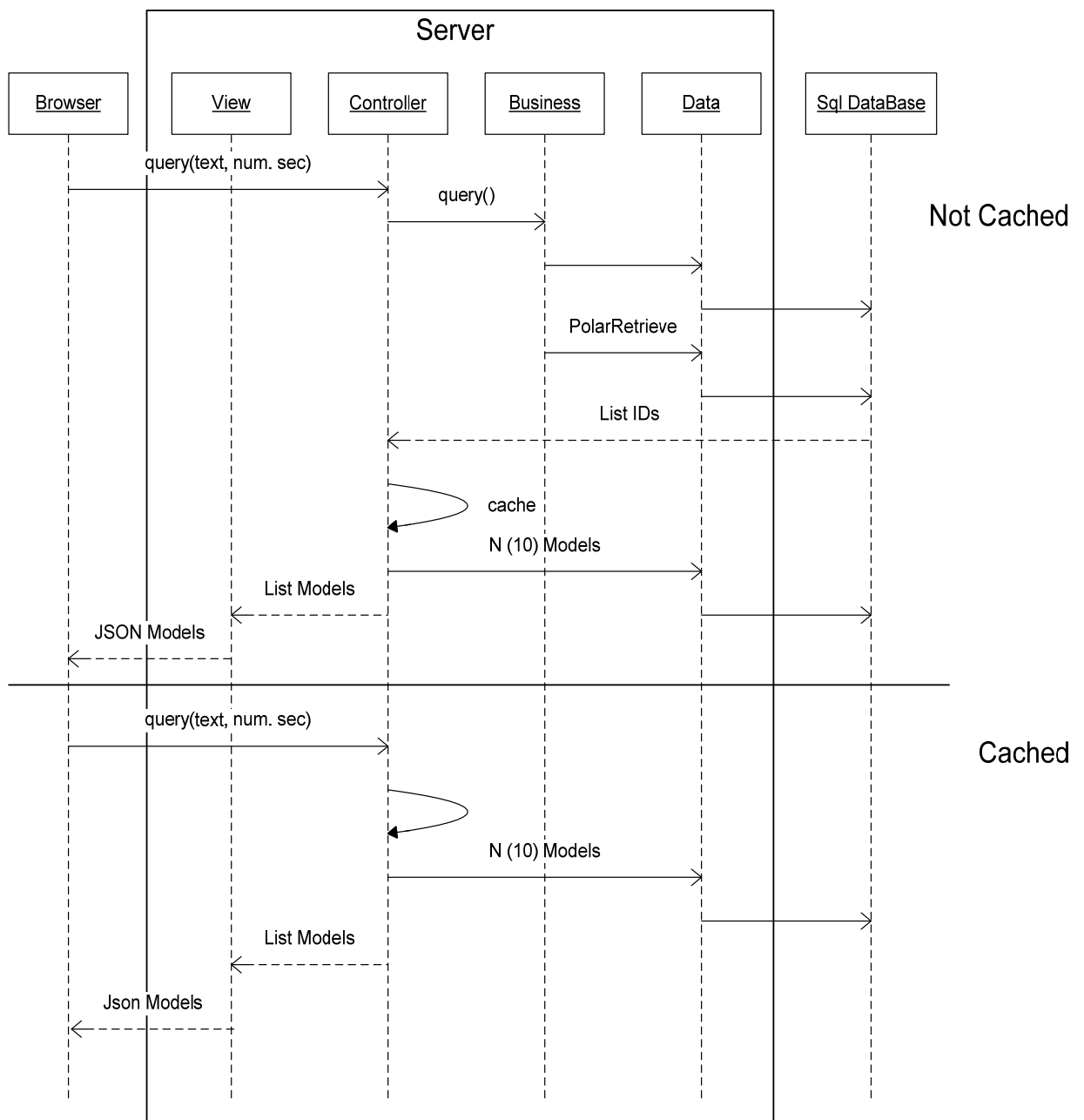


Illustration 23: Architecture: Search Result: Sequence Diagram.

The cache of identifiers is stored in the session and since the action of checking if a model exists or not is very simple, we have decided to delegate this responsibility to the controller (especially since he is the one that knows about the session).

8.4.2.1.1 Textual Search

The new architecture for the server will follow these two requirements:

- Each page of the model will be a different URL, the anchor² won't be used to identify the model and the information won't be requested with AJAX.
- Each query will be a different URL. This part still can be done with AJAX since is not very important to be indexed by Google in the search results.

The user initiates a search by typing the query in the browser. This query is sent to the server as a HTTP GET or HTTP POST request. The controller of the server attends the request and if it is a new query, then the controller ask to the business layer to return the results for that query. The controller stores that results and ask to the data layer for some results. The fist results are sent back to the user.

If the user wants to see more results, then a request is sent again to the server with the same query and the pagination index. The controller of the server checks if cached results are stored in the session for that query. If they are, then the controller asks to the data layer for some results. The next results are sent back to the user.

The cache of identifiers is stored in the session and since the action of checking if exists or not is so simple we have decided to delegate this responsibility to the controller especially since he is the one that knows about the session.

8.4.2.1.2 Diagrammer Search

In the case of the diagrammer, we will have a rich Internet client which will build the query from the diagram. The operations will be the same as in the search result but with different actions. The Controller in the server will have another interface in which a call to the Business is done. The business will perform other operations over this query and the result will also be a list if Ids. The controller will cached them and the first (next) models will be retrieved and returned.

8.4.2.1.3 Virtual Assistance Search

The operations will be similar than the search result. In this case a Controller in client is needed. This Controller is sending the questions to the Server Controller and this to the Business. If the business stores an instance of the NUML that is building, then the rest of the operations are the same.

When constructing the query from the dialog between the virtual assistance and the user, the Server Controller is different. Once the query is built, the user is redirected to a search results page in which the query constructed by the diagrammer will replace a textual query built by a user. The main difference is that in this case the query is already in the server and the model were already indexed son in this moment only a polar retrieved is performed, the Ids are

² Part of the URL behind the '#'. I.e.: www.umlmodels.org/index.aspx#model-id=1

returned to the Server Controller which will cache them and the first (next) models will be retrieved and returned.

8.4.2.2 Model's Page

When a model is retrieved, the operation is much easier than in the search result since only the retrieve of the specified model in the URL is done. In this case, the Controller asks to the Data Access the information of the model and this one is retrieved from the Database.

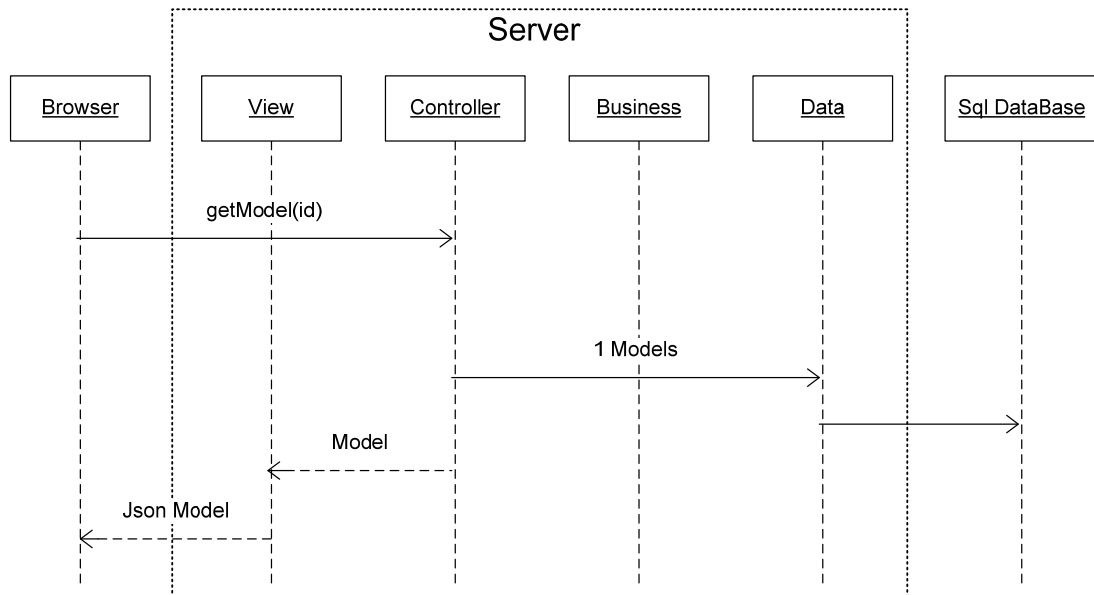


Illustration 24: Architecture: Model's Page: Sequence Diagram.

8.4.3 Deployment Infrastructure

The deployment infrastructure consists of a Client which will access to a Web Server to perform a search, retrieve or insert an UML model. When a query is performed, the indexing of the query is resolved by the Indexer; and once it is done, the server will retrieve the query in a machine with a SQL Server 2005.

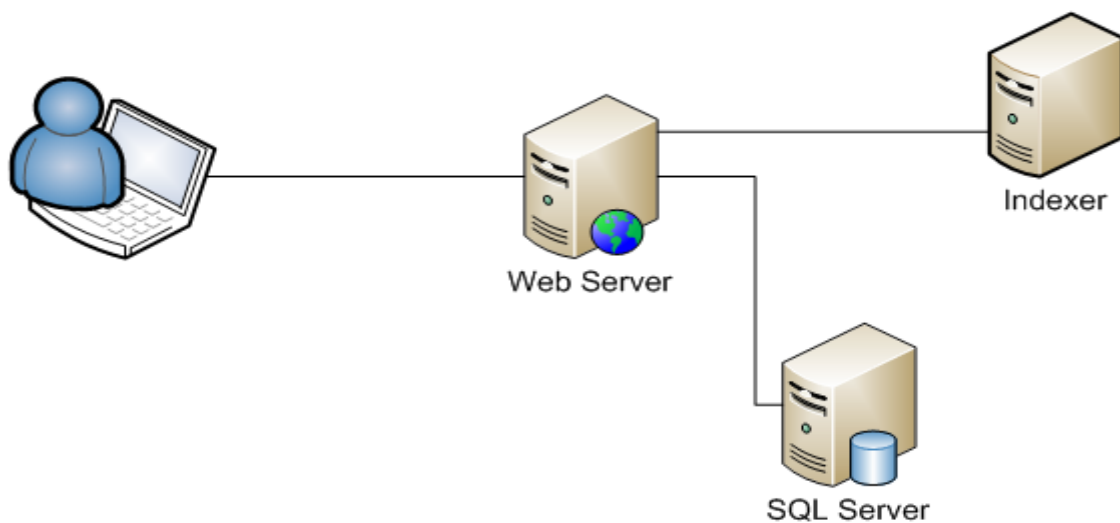


Illustration 25: Architecture: Deployment Structure

The indexer and the web server can be hosted in the same machine. In the deployment diagram this will look like this:

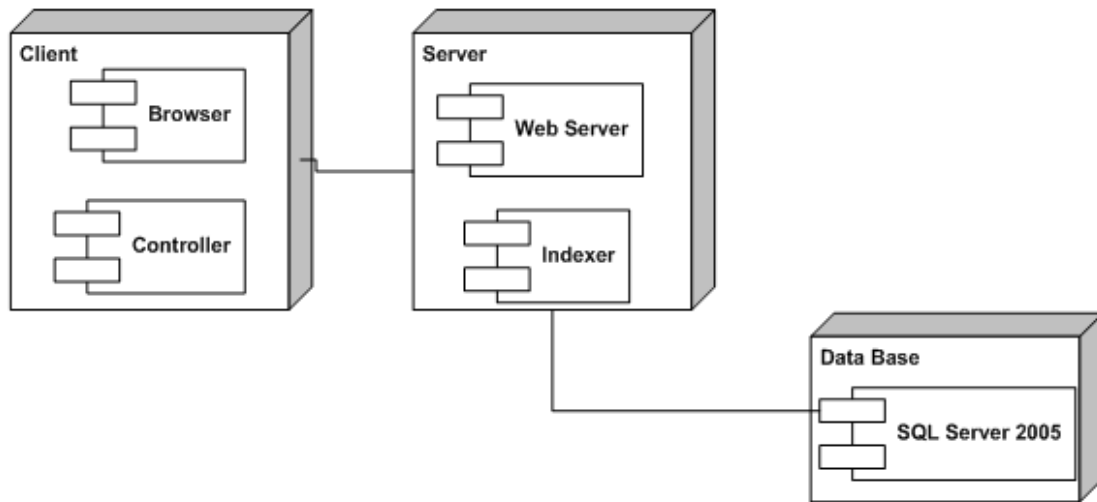


Illustration 26: Architecture. Deployment diagram.

As it is shown in the *Illustration 6*, we have three nodes: Client, Server and the Database. In the client we have two components: the browser and the controller. In the server we have two components: the *web server (controller, business and data tier)* and the indexer (out of this project). And the database will contain an SQL server 2005. These components are described on detail in the *Detailed Design* section.

8.5 GUI

This section is detailed the graphical representation of the presentation layer. It is neither included how the information is retrieved nor the detailed design of the presentation layer (for that refers to further sections in this document). So this section is focused on the visual part of the model pages and the search result page. This is a definition of which elements will be presented and where.

The design of the graphical part of the portal has been guided by the follow specifications:

- The portal has to have a main page with a search engine. This page might be over elaborated or simple (like Google).
- The portal has to have a validation zone. Two areas are clearly defined depending on the role of the user. The content and some functionalities will be restricted (from less to more restrictions):
 - o One for Registered Users:
 - Administrator (*non restriction*).
 - Operator (*not defined in this phase of the project*).
 - Client (*all content and functions but administration ones*).
 - o Another one for unregistered users (*full restriction*).
- Search result page must be designed.
- Model's page must be designed:
 - o One model is composed by N elements. Each element can be different kind of things such as: images, source code, requirements ...
 - o The image must be the cornerstone of the model.
 - o A model can be traced and related against other elements.
 - o A model has a collaboration area in which the users can interact: Add their content and post comments in a forum.
- Human indexing process must be supported.

As it has been described, the site contains collection of models. These models can be queried in the **main page** of the site. Once the query is resolved, the resultant models that match are listed in the **search result page**. Either when a user access to one of these results or when he types specific URL of a model in the browser the **models page** is presented.

Additionally, the Site will have some global elements that they will be common to those three areas presented above. These elements depend on the condition of the user and they will be present in all the pages:

- Common to all the users:
 - o Logo of UmlModels.
 - o Hyperlink to explanation videos in Youtube.
 - o Upload a Model
 - o Search
- Unregistered Users:
 - o Sing In.
 - o Sing Up.
- Registered Users:
 - o Balance & Hyperlink to buy more.
 - o View As: Role switcher.

- Hyperlink to Profile.
- Sing Out.

For UmlModels, the image is the cornerstone upon which all the other information of the model lies (see Illustration 27: GUI: Screenshot Draft Model's Page).



[TITLE OF THE MODEL]

IMAGE:	Sponsored Links
--------	-----------------

[[SOURCE HYPERLINK](#)]

[[XMI](#)] - [[SOURCE CODE](#) [n]] - [[EXECUTABLE COMPONENTS](#) [n]]

[[TRAZABILITY](#) [n]] - [[ADDITIONAL DOCUMENTATION](#) [n]]

DESCRIPTION:

FORUM:

Illustration 27: GUI: Screenshot Draft Model's Page

In the search result page the thumbnails will stand out over the textual information. Furthermore, it will be shown only the most relevant textual information needed to attract the user to the model information page. As well, the position of the results that the site is

providing for a determinate query will be ranked. So the most important results and the most important elements of the results will be shown in a more relevance position.

Even more, in the information model page the description and all the related information will be shown in a secondary manner.

In fact, not all the information will be shown to everybody. Depending on the Role of the user, they will be able to show more or less detail as a gimmick to attract the public.

Once the users have registered into the portal, then many useful downloads will be offered in relation to the model is been shown. Each download will be charged and the money will be divided into the owner of the model and UmlModels.

Finally, when this service will gain access into the developer internet community, many possibilities will show up such as test code, bank job, collaborative and comparative area among many others.

In the follow subsections the *Level Structure* (definition of the elements of the GUI structured in conceptual levels), the *Site Map*, the *Main Page*, the *Search Results Page* and the *Model's Page* are defined.

8.5.1 Level Structure: Elements Definition

In this section a definition of the elements of the GUI is presented and structured into three different conceptual levels:

- ➔ Model of Information.
- ➔ Collaborative Model.
- ➔ Business Model.

8.5.1.1 Level 1: Model Information

In this level is represented the intrinsic information of a model and the meta-information around the owner of the model: A **user** or the **crawler**.

This information will be:

- Owner Title.
- Owner Description.
- Owner Images.
- Hyperlink to Source (Context).
- Hyperlink of the Resource.
- Owner XMI.
- Owner Source Code.
- Owner Additional Documentation.
- Additional Owner Images.

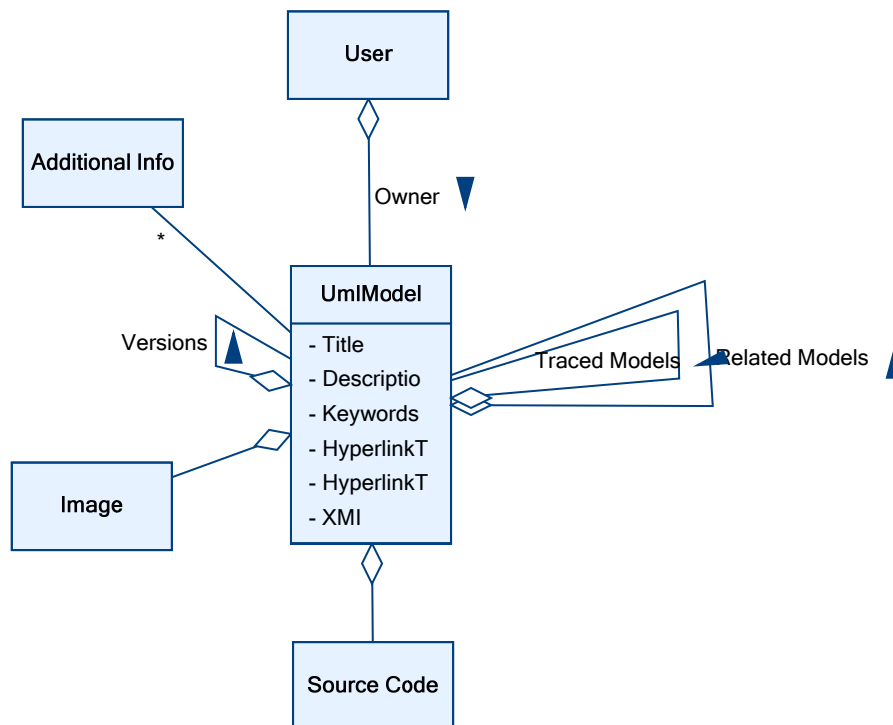


Illustration 28: Model Information Conceptual Model

8.5.1.2 Level 2: Collaborative

In this level is included all the aspects related to the collaboration of other users: UmlModels community. This information will be:

- Punctuations.
- Privacy.
- One Forum.
- Additional Descriptions of other users.
- Additional Images of other users.
- Additional Source Code of other users.
- Additional Documentation of other users.

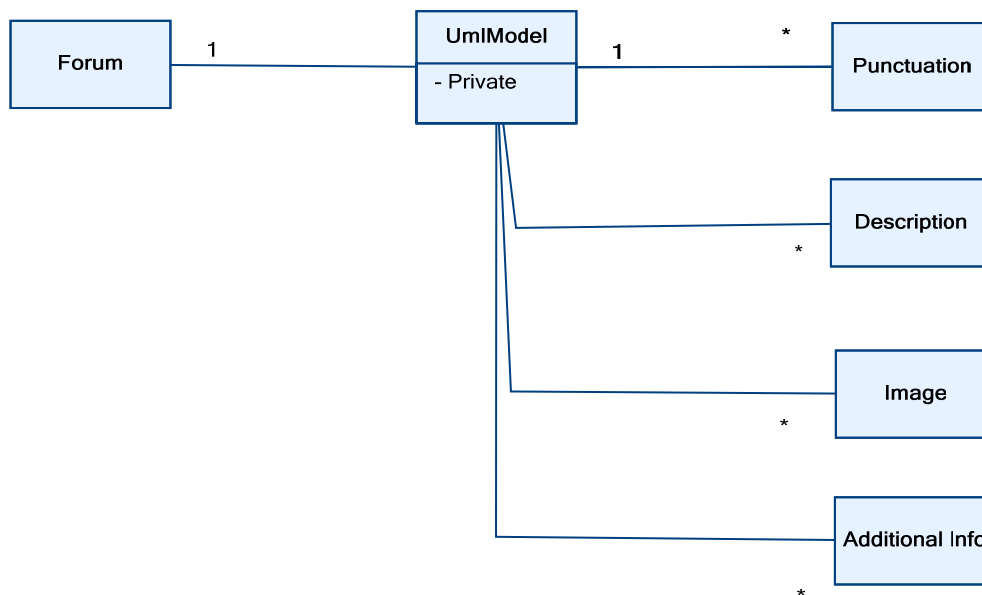


Illustration 29: Collaborative Conceptual Model

8.5.1.3 Level 3: Business Model

In this level is included all the aspects related to the commercial look of the Site. This information will be:

- Sponsored Links.
- Job Bank.

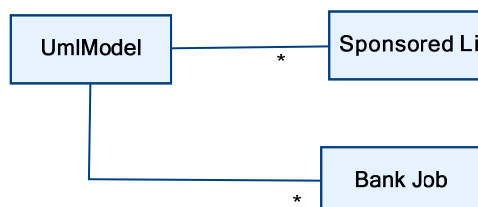


Illustration 30: Commercial Conceptual Model

8.5.2 Site Map

In this section is presented the site map hierarchy as well as the actions the user can perform in each place:

- Home Page: Point of access to UmlModels.
- Search Result:
 - Pagination: Next/Previous results.
 - Level of Detail: From one to five.
 - Link to model's URL.
 - Link to context URL.
- Model's Page.
 - Manage Resources (if registered)
 - Upload Title:
 - Input: String with the title.
 - Input: Boolean indicating if it is main title.
 - Delete Uploaded Title
 - Upload Description:
 - Input: String with the description.
 - Input: Boolean indicating if it is main description.
 - Delete Uploaded Description
 - Upload Requirement:
 - Input: String with the title of the requirement.
 - Input: String with the description of the requirement.
 - Delete Uploaded.
 - Upload Image:
 - Input: String with the title of the image.
 - Input: String with the local path of the file.
 - Input: String with the description of the image.
 - Input: Boolean indicating if it is main resource.
 - Delete Uploaded Image.
 - Upload XMI:
 - Input: String with the title of the image.
 - Input: String with the local path of the file.
 - Input: String with the description of the image.
 - Input: Boolean indicating if the resource is chargeable.
 - Delete Uploaded.
 - Upload Source Code:
 - Input: String with the title of the package.
 - Input: String with the language of the source code.
 - Input: String indicating the type of package: DLL, executable, test, source code...
 - Input: String with the local path of the package.
 - Input: String with the description of the image.
 - Input: Boolean indicating if the resource is chargeable.
 - Delete Uploaded.

- Upload Documentation:
 - Input: String with the title of the documentation.
 - Input: String indicating the type of package: PDF, EXCEL, DOC, AVI...
 - Input: String with the local path of the file.
 - Input: String with the description of the image.
 - Input: Boolean indicating if the resource is chargeable.
 - Delete Uploaded.
 - Add Post in the forum:
 - Edit Post.
- Rate. Resultant action: Log in (if visitor); or rating (if registered user).
 - Model.
 - Images.
 - XMIs: (Error message if the user has not previously downloaded it).
 - Documents: (Error message if the user has not previously downloaded it).
 - Requirements.
 - Source Code: (Error message if the user has not previously downloaded it).
 - Descriptions.
- Buy/Download Resource. Action: Log in (if visitor); Download (if registered user and free), download and reduce balance (if registered user and chargeable) or error message if the user doesn't have enough balance.
 - XMIs.
 - Documents.
 - Requirements.
 - Source Code
- Report bad content:
 - Log in (if visitor).
 - Message to the owner (if registered user)
- Panel de Control (if logged in).
 - Bookmarks: Bookmarks of the user.
 - My Resources: List of all the resources owned by the user with administration actions.
- Profile (if logged in):
 - User name (non editable).
 - Email (non editable).
 - First Name (editable).
 - Last Name (editable).
 - Mobile Telephone (editable).
 - Personal URL (editable).
 - Balance (non editable).
 - Karma (non editable).
 - Member since (non editable).
 - Last log on (non editable).

- Link to PayPal account.
- Link to print view.
- Link to context URL.
- Change the type view (if logged in and with enough permission): Client, Visitor and Operator.
- Log in and register; or log out.
- Sign in Page.
- Sign up page.

8.5.3 Main Page

The home page will have two different views depending on the role of the user:

1. Logged User.
2. Unlogged User.

The home page has to fulfil these requirements:

1. Simple.
2. Flexible.
3. An advertise place for UmlModels: Logo, info, etc.
4. Place for register/login/logoff.

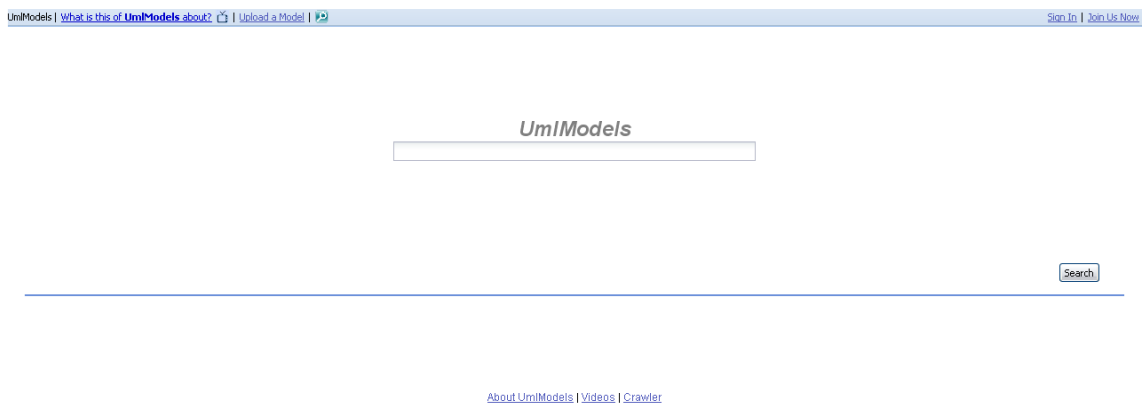


Illustration 31: Main Page. Screenshot.

8.5.4 Search Result

The search result page has to fulfil these requirements:

1. Simple.
2. The results have to be ordered according to a ranking (not defined yet).
3. Thumbnails: The diagrams must be shown in an advantage position.
4. Must have a Hyperlink to the Information model in a primary position.
5. Must have a Hyperlink Link to source in a secondary position.
6. The thumbnail will be accompanied by short descriptions of all the additional information that UmlModels provide about the artifact on the Information Model Page.
7. These short descriptions must be an appeal for the user to visit the information model page instead of the source page.
8. Necessity of seen the information related to the search.

According to the requirements described above, the search result page will follow the next structure:

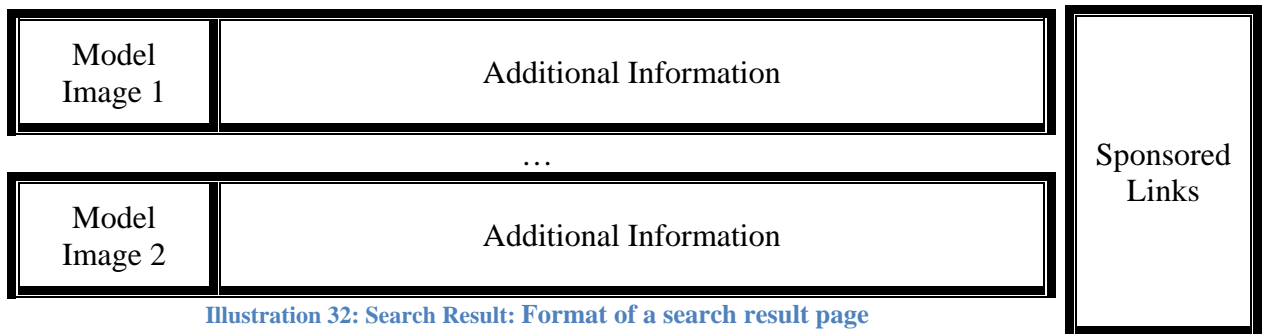


Illustration 32: Search Result: Format of a search result page

The search result page will have different levels of detail. The results will have more information according with the level of detail switched by the user. In all of them the image will be the main element. This will be situated on the left of the description area; and it will blow up as the level of detail is increased.

It has been defined five different levels of detail with the follow elements on it:

- **Level 1**

In this information level of detail, the image will have a size of 80px of high and 60px of width. On the right of the image we will be able to find the next information:

- *SourceTypeModel*: The type where the model come from (Human Image, Crawler Assembly ...).
- *Title*. The title and hyperlink to the page of the model.
- *NumTitAddic*: Number of additional titles added by other users than the owner. .
- *Description*: An extract of the description of the model.
- *Punctuation*: From one to five stars.
- *NumPosts*: Number of posts of the forum and an icon or avatar.

- *JobBank*: Amount of money that the Job Bank can offer in this moment associated to the aforesaid model
- *SourceCode*: Number of available downloadable files and an icon indicating the file source code type.
- *AdditionalDoc*: Number of downloadable files and an icon indicating type of the document.
- *HyperlinkToSource*: Hyperlink to the source page where the model was found.

[<i>SourceTypeModel</i>] Title (NumTitAddic) (<i>One line</i>)
Description <i>...Can have more than one line...</i>
Punctuation – NumPosts – SourceCode – <i>AdditionalDoc</i>
HyperlinkToSource (<i>One line</i>)

Illustration 33: Search Result: Format of the additional information for level of detail 1

- **Level 2**

In this level of information, the image will have double size than in the before level. On the right, on addition of the other information it will be this extra one:

- *DescriptionKeyWords*: Date of the last post in the forum.
- *Versions*: Number of existing versions of the model.
- *Num. Traced Elements*: Number of traced elements over the model.
- *Related Models*: Number of related models.

[<i>SourceTypeModel</i>] Title (NumTitAddic) (<i>One line</i>)
Description <i>...Can have more than one line...</i>
Punctuation – NumPosts – SourceCode – <i>AdditionalDoc</i>
Tags: <i>DescriptionKeyWords</i> (<i>One line</i>)
Versions: NumVersions – Num. Traced Elements: NumTraceElements - Related Models: NumTraceElements (<i>One line</i>)
HyperlinkToSource (<i>One line</i>)

Illustration 34: Search Result: Format of the additional information for level of detail 2

- **Level 3**

In this level of information, the image will have double size than in the before level. On the right, on addition of the other information it will be this extra one:

- *Owner*: The owner of the model.

[<i>SourceTypeModel</i>] Title (NumTitAddic) (<i>One line</i>)
Description <i>...Can have more than one line...</i>
Punctuation – NumPosts – SourceCode – <i>AdditionalDoc</i>
Tags: DescriptionKeyWords (<i>One line</i>)
Versions: NumVersions – Num. Traced Elements: NumTraceElements - Related Models: NumTraceElements (<i>One line</i>)
Owner: UserName (<i>One line</i>)
HyperlinkToSource (<i>One line</i>)

Illustration 35: Search Result: Format of the additional information for level of detail 3

- **Level 4**

In this level of information, the image will have double size than in the before level. On the right, on addition of the other information it will be this extra one:

- *UserPost*: The last user that posted something in the forum.
- *TipeOfMember*: Avatar that represents the rank that the user has (User, Premium ...).
- *DateTime*: Date of the last post in the forum.

[<i>SourceTypeModel</i>] Titulo (NumTitAdic) (<i>One line</i>)
Description <i>...Can have more than one line...</i>
Punctuation – NumPosts – SourceCode – <i>AdditionalDoc</i>
Tags: DescriptionKeyWords (<i>One line</i>)
Versions: NumVersions – Num. Traced Elements: NumTraceElements - Related Models: NumTraceElements (<i>One line</i>)

Owner: UserName - LastPost by User A TipeOfMember DateTime (One line)

HyperlinkToSource (One line)

Illustration 36: Search Result: Format of the additional information for level of detail 4

- **Level 5**

In this level of information, the image will have double size than in the before level. On the right, on addition of the other information it will be this extra one:

- *PreviewLastPost*: A preview of the post that the last user left on the forum associated the model.
- *BankJob*: Short description of the task that the user will have to do if he wants to participate in the job offer and the reward for doing it.

[SourceTypeModel] Titulo (NumTitAdic) (One line)

Description

...Can have more than one line...

Punctuation – NumeroPosts – SourceCode – AdditionalDoc

Tags: DescriptionKeyWords (One line)

Versions: NumVersions – Num. Traced Elements: NumTraceElements - Related Models: NumTraceElements (One line)

LastPost by User A TipeOfMember DateTime (One line)

PreviewLastPost

...Can have more than one line...

HyperlinkToSource (One line)

“This model has the next Bank Jobs”:

- BankJob 1. Amount of Money.

- ...

...Can have more than one line...

Illustration 37: Search Result: Format of the additional information for level of detail 5

8.5.4.1 Actions to be done when any field of the above described doesn't exist

In case any field doesn't exist, it has to be tried to refill the white space by taking some information from the next levels.

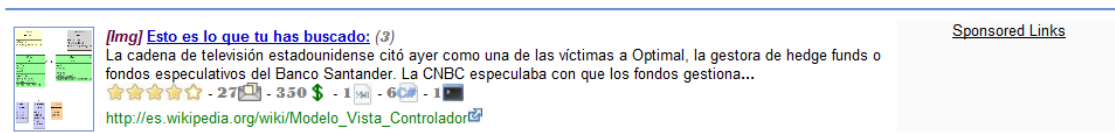
In the case of the title and the description, it will be showed other titles and descriptions added by other users.

Also in each of the levels the GUI will try to fill the white space by expanding the additional information or the preview of the last post until the white space will be reduced.

8.5.4.2 Detailed Structure

8.5.4.2.1 Levels of Detail

- **Level 1:** Screenshot of a search result for this level of information:

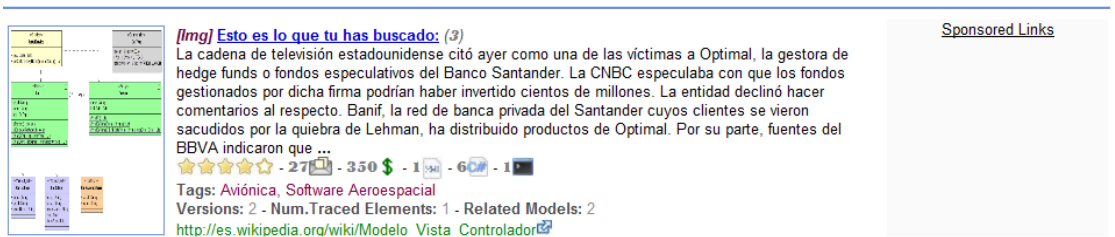


[img] Esto es lo que tu has buscado: (3)
La cadena de televisión estadounidense citó ayer como una de las víctimas a Optimal, la gestora de hedge funds o fondos especulativos del Banco Santander. La CNBC especulaba con que los fondos gestiona...
☆☆☆☆☆ - 27 [img] - 350 \$ - 1 [img] - 6 [img] - 1 [img]
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Sponsored Links

Illustration 38: Search Result: Level of detail 1

- **Level 2:** Screenshot of a search result for this level of information:

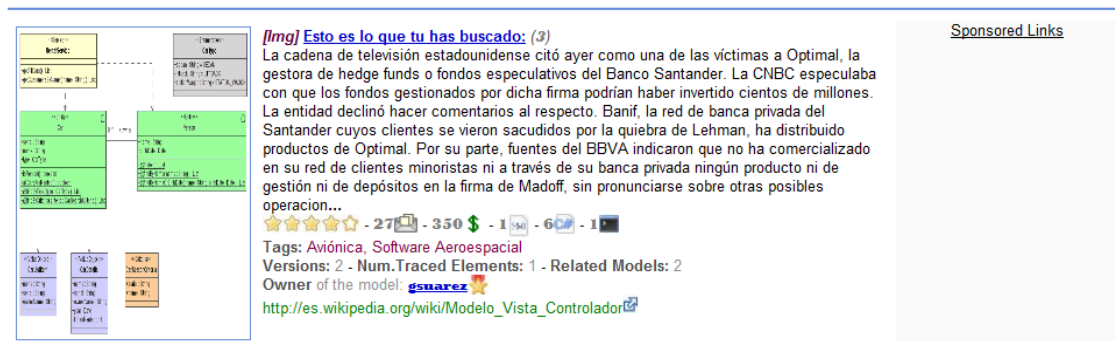


[img] Esto es lo que tu has buscado: (3)
La cadena de televisión estadounidense citó ayer como una de las víctimas a Optimal, la gestora de hedge funds o fondos especulativos del Banco Santander. La CNBC especulaba con que los fondos gestionados por dicha firma podrían haber invertido cientos de millones. La entidad declinó hacer comentarios al respecto. Banif, la red de banca privada del Santander cuyos clientes se vieron sacudidos por la quiebra de Lehman, ha distribuido productos de Optimal. Por su parte, fuentes del BBVA indicaron que ...
☆☆☆☆☆ - 27 [img] - 350 \$ - 1 [img] - 6 [img] - 1 [img]
Tags: **Aviónica**, **Software Aeroespacial**
Versions: 2 - Num.Traced Elements: 1 - Related Models: 2
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Sponsored Links

Illustration 39: Search Result: Level of detail 2

- **Level 3:** Screenshot of a search result for this level of information:



[img] Esto es lo que tu has buscado: (3)
La cadena de televisión estadounidense citó ayer como una de las víctimas a Optimal, la gestora de hedge funds o fondos especulativos del Banco Santander. La CNBC especulaba con que los fondos gestionados por dicha firma podrían haber invertido cientos de millones. La entidad declinó hacer comentarios al respecto. Banif, la red de banca privada del Santander cuyos clientes se vieron sacudidos por la quiebra de Lehman, ha distribuido productos de Optimal. Por su parte, fuentes del BBVA indicaron que no ha comercializado en su red de clientes minoristas ni a través de su banca privada ningún producto ni de gestión ni de depósitos en la firma de Madoff, sin pronunciarse sobre otras posibles operacion...
☆☆☆☆☆ - 27 [img] - 350 \$ - 1 [img] - 6 [img] - 1 [img]
Tags: **Aviónica**, **Software Aeroespacial**
Versions: 2 - Num.Traced Elements: 1 - Related Models: 2
Owner of the model: **gsuarez** [img]
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Sponsored Links

Illustration 40: Search Result: Level of detail 3

- **Level 4:** Screenshot of a search result for this level of information:

[img] Esto es lo que tu has buscado: (3)
 La cadena de televisión estadounidense citó ayer como una de las víctimas a Optimal, la gestora de hedge funds o fondos especulativos del Banco Santander. La CNBC especulaba con que los fondos gestionados por dicha firma podrían haber invertido cientos de millones. La entidad declinó hacer comentarios al respecto. Banif, la red de banca privada del Santander cuyos clientes se vieron sacudidos por la quiebra de Lehman, ha distribuido productos de Optimal. Por su parte, fuentes del BBVA indicaron que no ha comercializado en su red de clientes minoristas ni a través de su banca privada ningún producto ni de gestión ni de depósitos en la firma de Madoff, sin pronunciarse sobre otras posibles operaciones. Fuentes financieras indican que hay más hedge funds comercializados en España con inversiones en la firma estadounidense. Una filial de Madoff figura en los registros de la CNMV, si bien está domiciliada en Londres. También está registrada la firma Fairfield, una de las que distribuía los productos de Madoff, posiblemente también entre inversores españoles. Asociaciones de caridad En Estados Unido...
 ☆☆☆☆☆ - 27 - 350 \$ - 1 - 6 - 1
 Tags: **Aviónica, Software Aeroespacial**
 Versions: 2 - Num.Traced Elements: 1 - Related Models: 2
 Owner of the model: **gsuarez** - Last post in the forum: by **jurhano** 1 hour ago
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Illustration 41: Search Result: Level of detail 4

- **Level 5:** Screenshot of a search result for this level of information:

[img] Esto es lo que tu has buscado: (3)
 La cadena de televisión estadounidense citó ayer como una de las víctimas a Optimal, la gestora de hedge funds o fondos especulativos del Banco Santander. La CNBC especulaba con que los fondos gestionados por dicha firma podrían haber invertido cientos de millones. La entidad declinó hacer comentarios al respecto. Banif, la red de banca privada del Santander cuyos clientes se vieron sacudidos por la quiebra de Lehman, ha distribuido productos de Optimal. Por su parte, fuentes del BBVA indicaron que no ha comercializado en su red de clientes minoristas ni a través de su banca privada ningún producto ni de gestión ni de depósitos en la firma de Madoff, sin pronunciarse sobre otras posibles operacion...
 ☆☆☆☆☆ - 27 - 350 \$ - 1 - 6 - 1
 Tags: **Aviónica, Software Aeroespacial**
 Versions: 2 - Num.Traced Elements: 1 - Related Models: 2
 Owner of the model: **gsuarez** - Last post in the forum: by **jurhano** 1 hour ago
 Yo opino que Microsoft's Internet Explorer contains a number of leaks, the worst of which is an interaction with JScript. When a DOM object is passed to a JavaScript object (such as an event handling function), and when that JavaScript object contains a reference to that DOM object, then a cycl...
 This model has the next Bank Jobs:
 - Modulo tarificación programing. 300 \$
 - Tests Application. 50 \$
http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

Illustration 42: Search Result: Level of detail 5

8.5.4.2.2 Tooltips

With the use of tooltips, all the elements are self explained:

[HumanImage]
 //Create an item You can find all the wiki related classes inside the EucalyptoWiki namespace. You must use the EucalyptoWiki.WikiManager class to insert, read, or delete article data. The main entry point is the WikiManager class. This model have 7 titles added by the owner and 2 Additional Descriptive Titles added by other users
 ☆☆☆☆☆ - 27 - 350 \$ - 1 - 6 - 1
<http://www.univ-psau.tr/LML99/>

Illustration 43: Search Result: Tooltips.

8.5.4.2.3 Sponsored Links

On the right side of the result panel, occupies the 30 % of the wide of the window.

8.5.4.2.4 Pagination

Show the different pages that contain results for the specific query committed. In each page is shown ten results. And it is possible to select five different kinds of views for each page according to the level of detail.



Illustration 44: Search Result. Pagination bar.

Also is possible to split the results into columns:

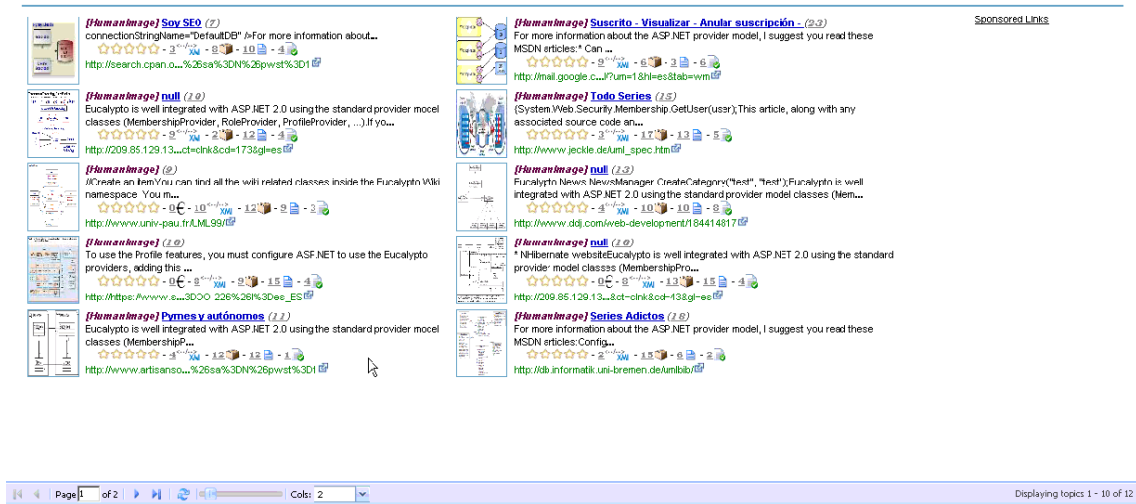


Illustration 45: Search Result: Results split in columns.

8.5.4.3 Resultant Work

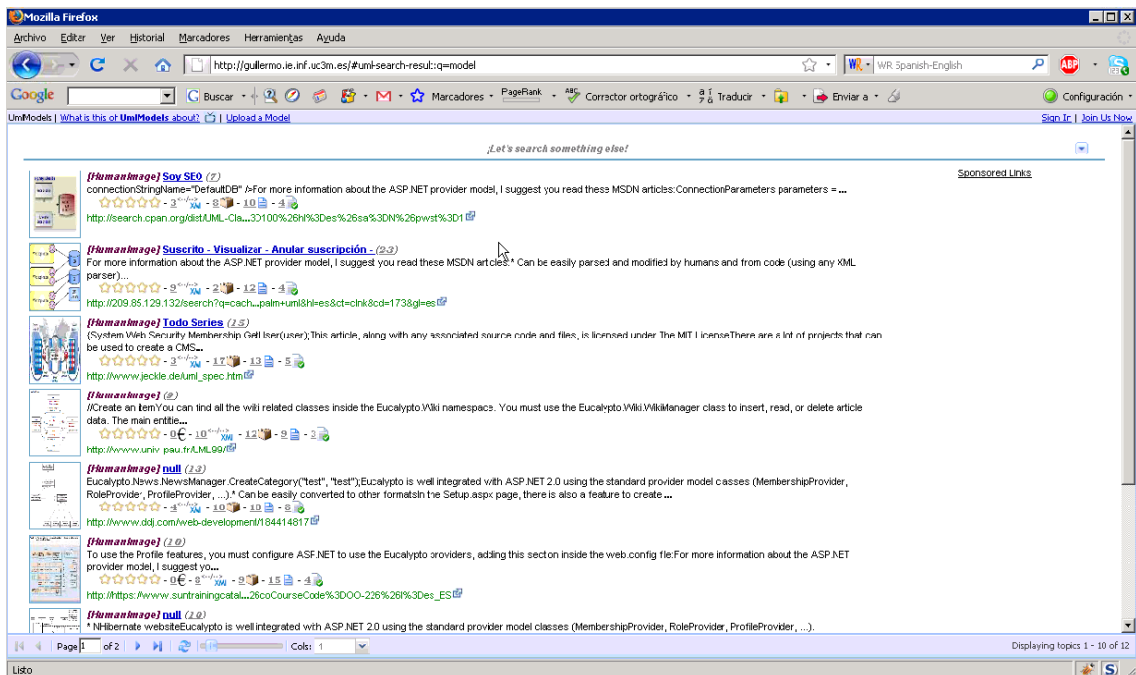


Illustration 46: Search Result: Screenshot resultant work.

8.5.5 Study of the Distribution of the Valuable Information of a Model in the Search Result page.

As it has been described in the previous section, depending on the level of detail we are showing more or less information near the image. The information should fill all the white space on the right side of the image. In other words, the textual information should fit to the available space.

Depending on the size of the image and on the size of the window, in some cases it is necessary to insert more information and in some others less. Since the characters of a sentence have different sizes between each other, it is hard to estimate the number of characters needed fill on sentence.

For this reason, a Study of the distribution of this free space has been performed.

In the following illustration we can see a simple example of what we exactly want to avoid:

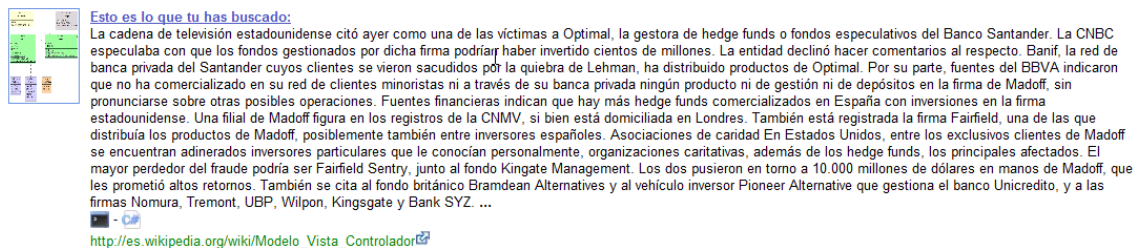


Illustration 47: Example of an uncut short description.

We have a big description, but we would like to just print the necessary information to tighten the text up to the high of the image as it is shown in the following illustration:

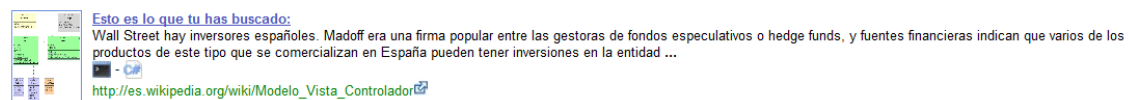


Illustration 48: Example of a cut short description.

The information that can have each of the levels as well as the number of lines that they can take up is described in the previous section. Bearing in mind that information, a statistical study has been done taking a dozen of sentences from different sites on the internet such as news papers making the average of the number of words per line(*) according to the properties of each level of detail. The following table represents the result of this study:

Width Panel						
1098						
Free Space Search Result Panel						
Level Detail	1	2	3	4	5	
Height Img	80	160	240	320	400	
Width Img	60	120	180	240	300	
Free Width	818,4	758,4	698,4	638,4	578,4	
Number Lines	5	10	16	21	27	
Number Words/Line	142,88	133,13	121,50	111,38	103,57	
Ratio Width	5,72808399	5,69690141	5,74814815	5,73198653	5,58455172	>> 5,69793436
Ratio Height	16	16	15	15,2380952	14,8148148	>> 15,410582

Table 97: Study of the free width per number of words/line. Width Panel: 1098

Width Panel						
635						
Free Space Search Result Panel						
Level Detail	1	2	3	4	5	
Height Img	80	160	240	320	400	
Width Img	60	120	180	240	300	
Free Width	448	388	328	268	208	
Number Lines	5	10	16	21	27	
Number Words/Line	75,75	65,38	54,13	42,75	31,43	
Ratio Width	5,91419142	5,93499044	6,06004619	6,26900585	6,61818182	>> 6,15928314
Ratio Height	16	16	15	15,2380952	14,8148148	>> 15,410582

Table 98: Study of the free width per number of words/line. Width Panel: 635.

The average for all the different widths studied give as an average ratio of **5,92**. This ration will be used on the rendering of the search results³ to estimate the number of characters must truncate in each level of detail for a specific with of window. For more details about the calculations see '*Study of the free width per number of words per line.Calculations.xlsx*' in the documentation.

The accuracy of these results might be increased introducing more entropy on the calculations: such as distinguishing between different characters sizes such small ones like 'i', 'l', 'j'... But since the calculations are performed to obtain always an approximation error of half a line and the most important element of precision are the number of lines, this won't feel much difference on the final result.

³ Implementation details: See '*UmlModels.Site/Scripts/UmlModels/SearchResult/ResultRecords.js*'

8.5.6 Model's Page

The UmlModels commercial business model wants to consolidate a huge community of software engineers around UmlModels.

UmlModels users will be able to add, get and discuss around any artefact. This artefact can be a free resource to everybody; which means that everybody can download all the information within the model for free. Or can be a chargeable model, in which users will be charged for getting the 'downloadable elements'.

When a model is chargeable, any download of any resource but the images will cost money to the user. The charged money will be divided into UmlModels and the owner that has uploaded it.

As money is involved in this business model, it is necessary to distinguish between two different groups of users: registered and unregistered users.

In the visual part of the site that offers the models, the UML diagram should be the most important element. Around this element and with secondary importance should be positioned other elements offering information about the model.

The elements that should be positioned around the UML can be different type of elements such as:

- A forum talking about the model.
- Other related models.
- Source code.
- Other related diagrams.
- Manuals.

According to the requirements described above, the model's page will follow the next structure:

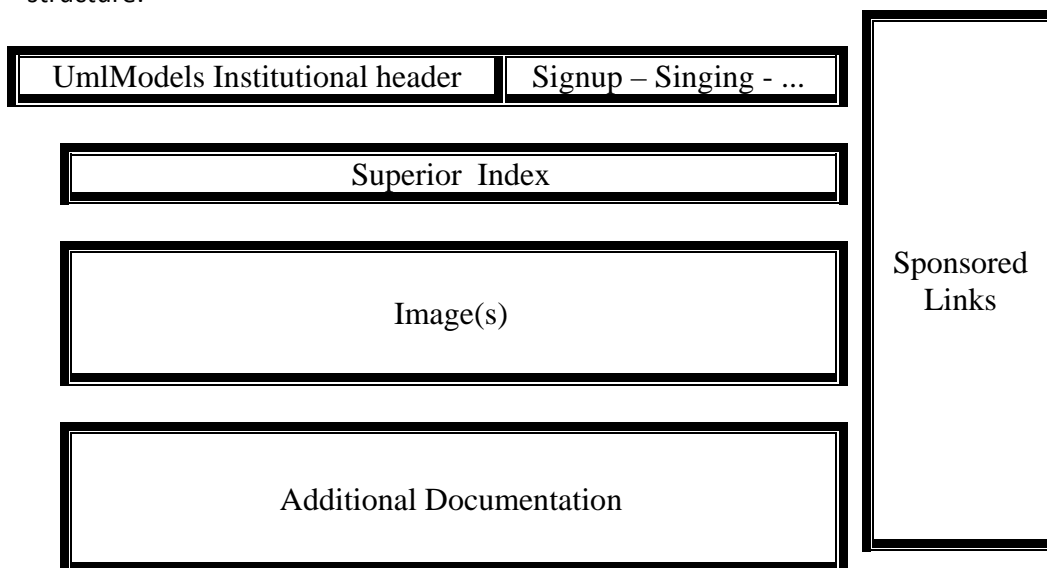


Illustration 49: Structure of the Model's Page

In the Model's page it is possible to distinguish two different views. One for the **unregistered** user and the other for the **registered** ones: Client, Operator and Administrator User.

This section is divided in: *Detailed Structure* and *Resultant Work* subsections and they contain a description of the elements that compound the GUI and screenshots.

8.5.6.1 Detailed Structure

8.5.6.1.1 Header

The top of the page will contain the follow elements:

- Logo: "UmlModels"
- Youtube link: Permanent link to UmlModels Youtube section
- Upload A Model: For unregistered users a sign in/register window will be prompted.
- Unregister: "Join us now" and Sing In.
- Registered: Balance, Buy more balance, Profile and Sing Out

8.5.6.1.1.1 Unregistered Users



Illustration 50: Header of an unregistered user.

8.5.6.1.1.2 Registered Users



Illustration 51: Header of a registered user.

8.5.6.1.2 Title of the Model

8.5.6.1.2.1 Unregistered Users

Tooltips: describing the additional titles that are only visible for registered users. This will act as an appeal for the unregistered users to register into UmlModels.



Illustration 52: Model's page: Title of the model. Unregistered user view.

8.5.6.1.2.2 Registered Users

All the titles of the model: giving more importance to the main title, next to the owner's title and later to the collaborative titles. Additionally they will be ordered according with the punctuation: Titles with more punctuation will be situated on the left.



Illustration 53: Model's page: Title of the model. Registered user view.

8.5.6.1.3 Superior Index

The super index will contain the follow elements:

- Small description to the subsections of this model (if any):
 - o Forum.
 - o Descriptions.
 - o Requirements.
 - o Job Offers.
 - o Downloadable files:
 - Source Code.
 - Additional Documentation.
- Punctuation of the model: Five yellow or white stars.
- Print Version: Link to the print version.
- Owner: Link to the owner profile.

Tooltips: For each of the elements described above.



Illustration 54: Model's page: Superior index. Unregistered user view.

8.5.6.1.3.1 Owner of the model



Illustration 55: Model's page: Superior index. Owners view.

The small descriptions has a short view in this shot screen because I considered that the registered users might know the web site well enough to hide the descriptions of the icons. This has been rejected by the client. Always will be prompted a small descriptions with the icons.

8.5.6.1.4 Central area

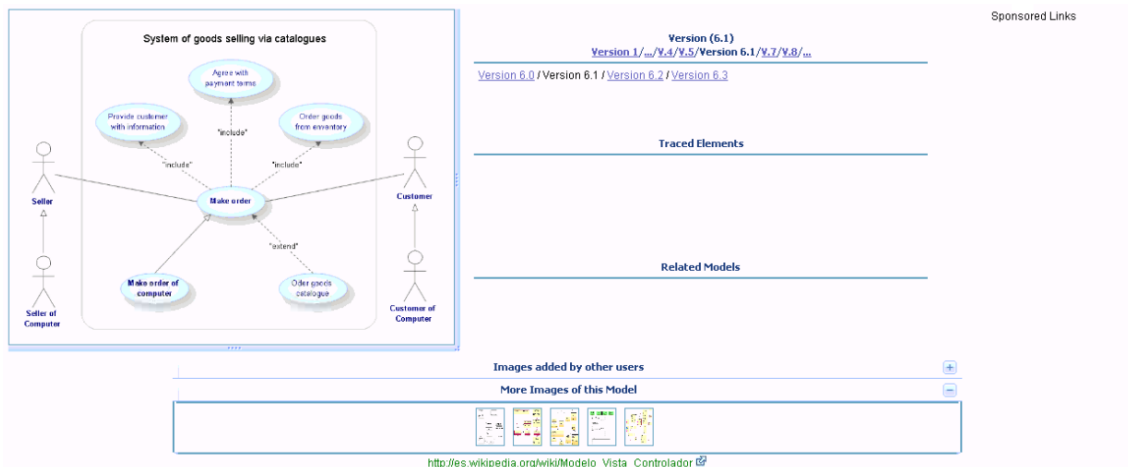


Illustration 56: Model's page: Central area.

8.5.6.1.5 Left side: Images

8.5.6.1.5.1 Unregistered Users

- Resizing: This feature is restricted to registered users. This will act as an appeal for the unregistered users to register into UmlModels.
- Additional Images: This feature is restricted to registered users. This will act as an appeal for the unregistered users to register into UmlModels.

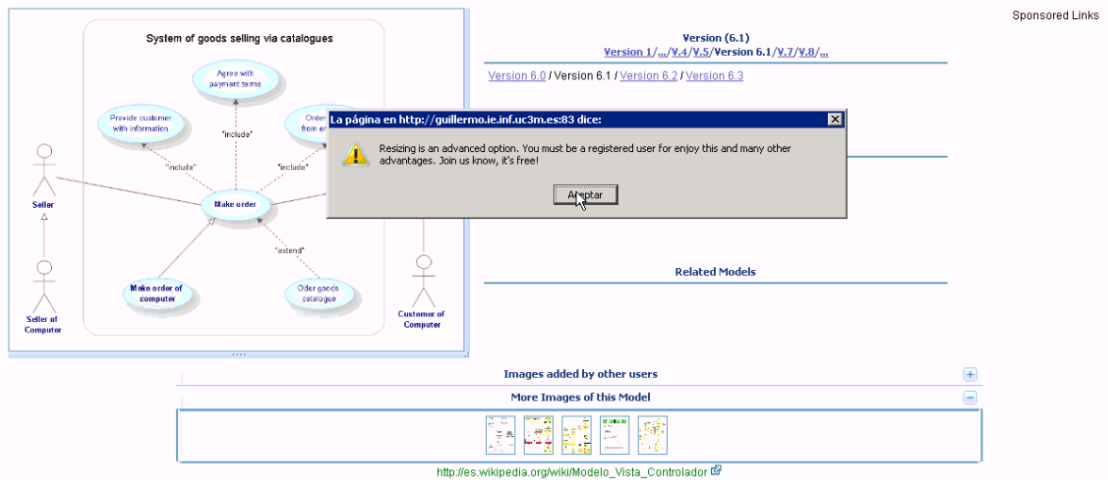


Illustration 57: Model's page: Resizing of the main image. Unregistered user view.



Illustration 58: Model's page: Additional Images. Unregistered user view.

8.5.6.1.5.2 Registered Users

- Resizing: The main image is resizable so the registered users can make it bigger. When the size of the image is wider than the 60% of the screen, then the right side of the central area will be hidden.
- Additional Images: This shows the additional images added by the owner and the collaborative images added by other users (collaborators).

Illustration 59: Model's page: Resizing of the main image. Registered user view.

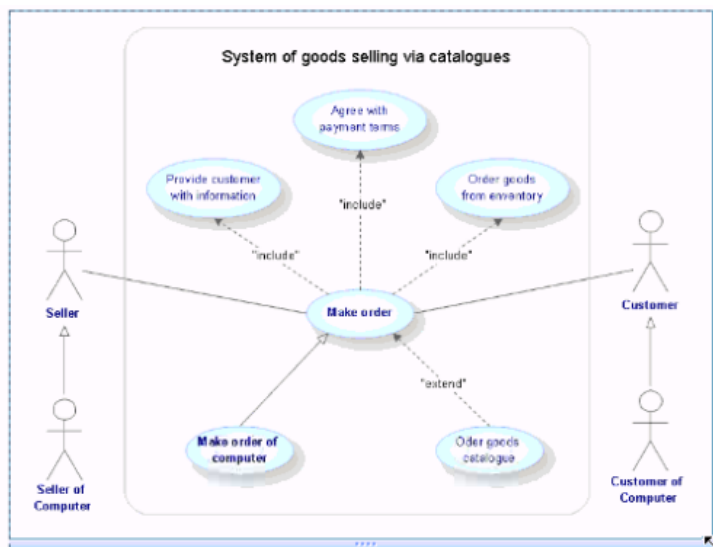




Illustration 60: Model's page: Additional Images. Registered user view.

8.5.6.1.6 Comparative Area

In the right side of the central area is situated another image for comparing it with the main one. The way it works the resize of the main image is as follow: when the size of the left image is bigger than the 60 %, then the second image is automatically hidden.

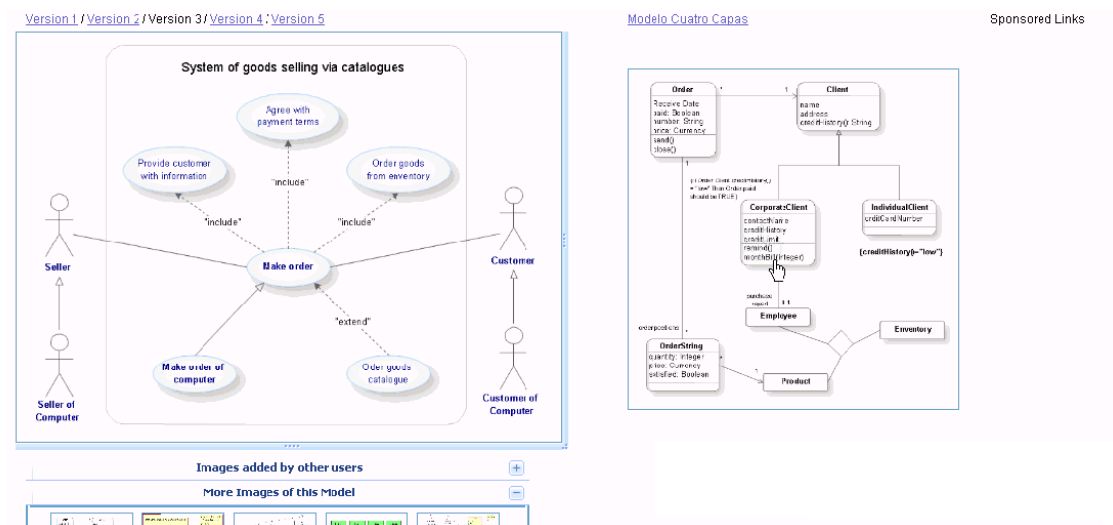


Illustration 61: Model's page: Comparative area for registered users.

When a user clicks on one of the additional images, then this image is situated in comparative area. The comparative mechanism works with java script and the image that it is been compared is added with an anchor parameter in the URL. This allows UmlModels to be "intelligent" and compare in the future any image with the one in this model.

8.5.6.1.7 Sponsored Links

The size of the sponsored links will be between the 20 and the 30% of the width of the screen.

8.5.6.1.8 Control Panel

This section will be placed on the left of the screen and will be only displayed when the users are registered. The content of the control panel will be further designed. The appearance will be as it is shown Illustration 58: Model's page: Control Panel. Registered user view.

The structuring of the control panel is going to be as follows:

- General actions:
 - Profile: View and modify.
 - Upload a Model.
 - Model Privacy.
- Actions over the resource
 - Add Title.
 - Add Description.
 - Add Requirement.
 - Add Image.
 - Add Code.
 - Add Documentation.
- Repository with all the models/resources that the user has decided to bookmark.
- Repository with all the resources in which the user owns.

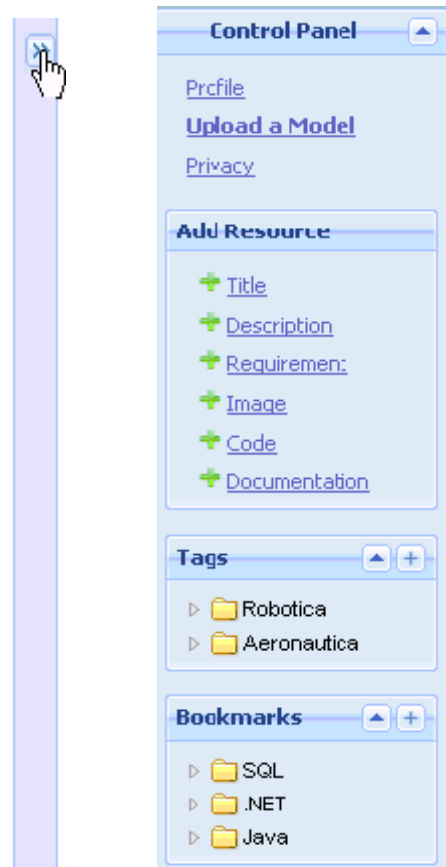


Illustration 62: Model's page: Control Panel. Registered user view.

8.5.6.1.8.2 Profile

The retrieve and the update of the profile are done with AJAX technology. With this, the communication is more efficient.

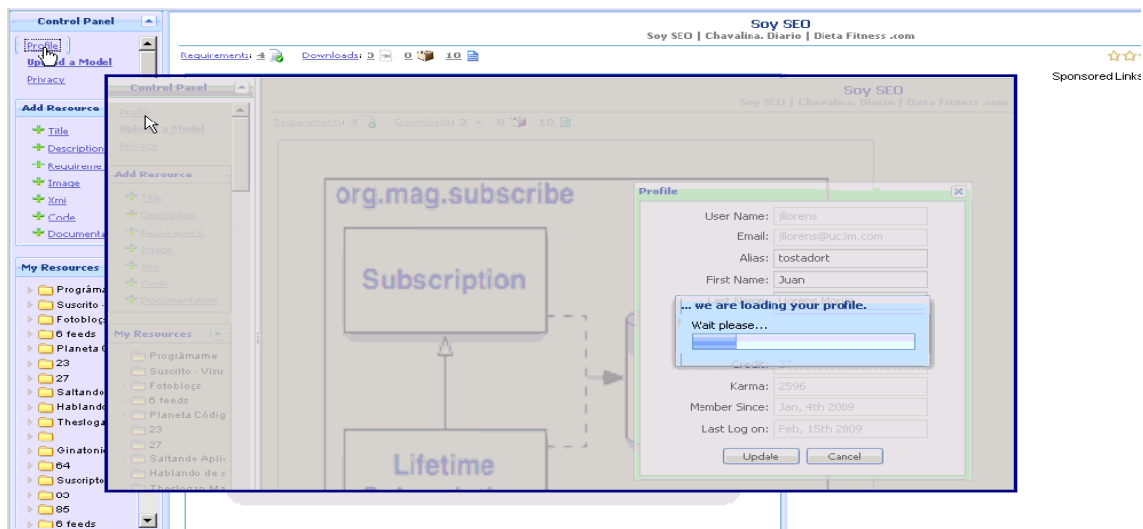


Illustration 63: Model's page: Control panel: Profile.

8.5.6.1.8.3 Upload a Model

Also the new model action is done with AJAX. When a registered user uploads a model, then a form panel (already loaded) will be shown.

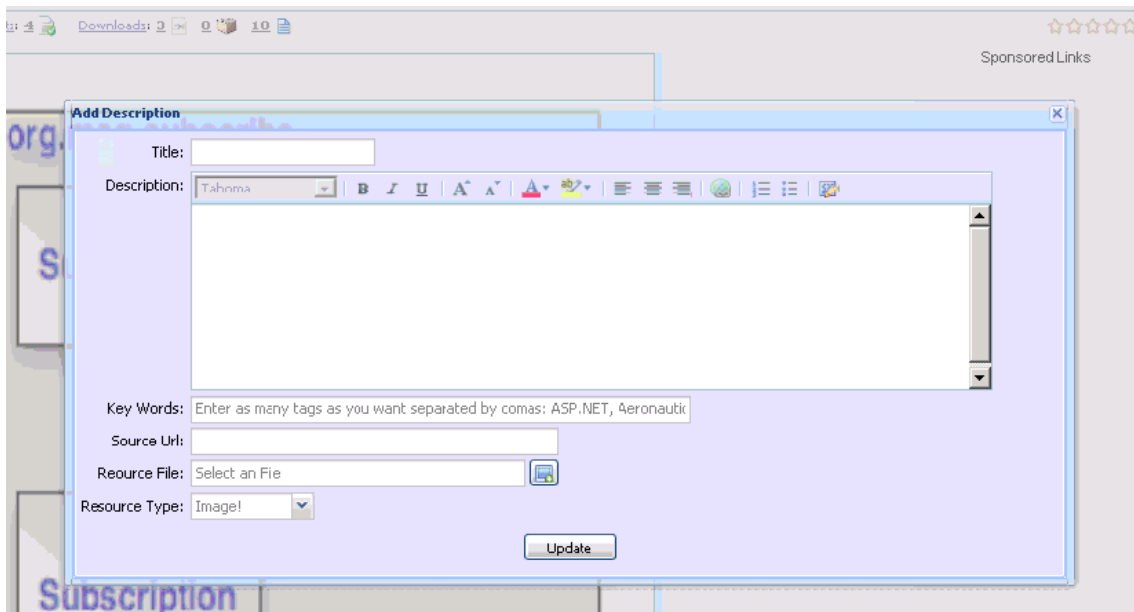


Illustration 64: Model's page: Control panel: Upload a Model.

8.5.6.1.8.4 Add Code

The elements that the user has to fulfil to add a source code are:

- Title
- Language
- Resource Type. Each source type has its own icon.
 - Source Code.
 - Package.
 - Executable.
 - Test.
 - DLL.
- File Path.
- Description.
- Chargeable.

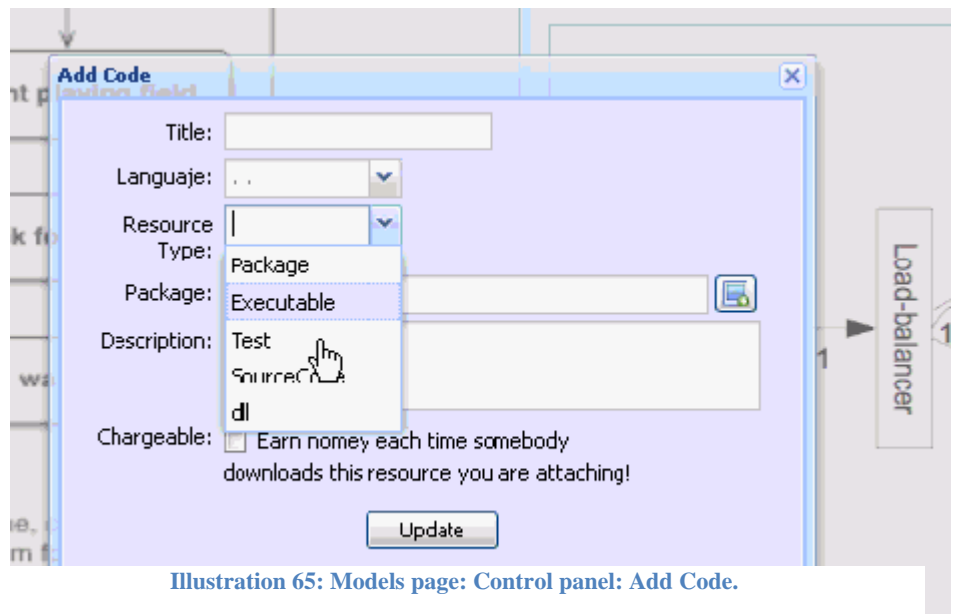


Illustration 65: Models page: Control panel: Add Code.

8.5.6.1.8.5 Add Documentation

The elements that the user has to fulfil to add a document are:

- Title.
- Resource Type: Each source type has its own icon.
 - Pdf.
 - Excel.
 - Doc.
 - Txt.
 - Others
- File Path.
- Description.
- Chargeable.

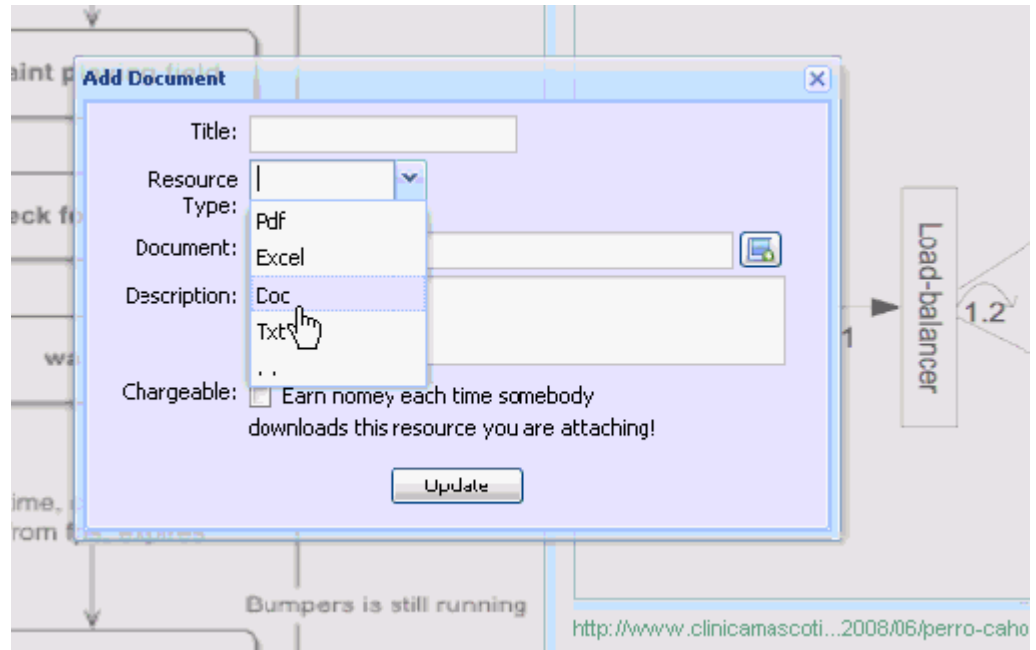


Illustration 66: Models page: Control Panel: Add Documentation.

8.5.6.1.8.6 My Resources

All the resources of a user are placed in a tree panel in ordered by the name of the model. Into the name of the model, a direct link to the UmlModel, and a folder per each type of resource is provided.

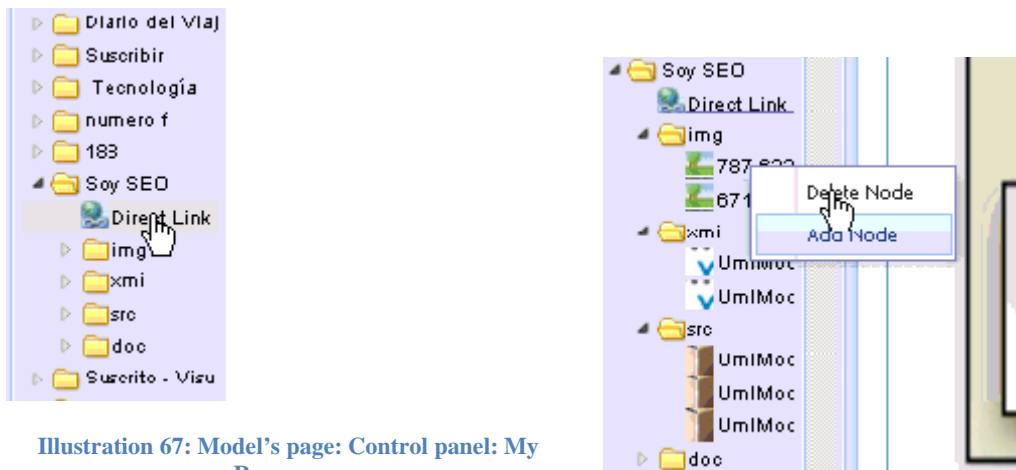


Illustration 67: Model's page: Control panel: My Resources.

In each resource folder is listed all the resources that the user own. In the *Illustration 63* the owner has two images, two XMIs, three source code packages for the resource called "Soy SEO" and with the right button might be possible to manage his resources.

Due all the possible actions that we will be able to do to the resources, it has been proposed the use of the right button menu to manage this actions. This is a powerful tool that can be used additionally to other management mechanism because is not usable and not intuitive.

One alternative is to add all the tools on the top of the tree panel:



Those tools can be:

- Add:
- Delete:
- Configure:
- ...

8.5.6.1.8.7 Bookmarks

Additionally, the users will be able to store in bookmarks the models that they like more:

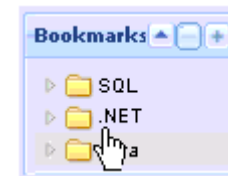


Illustration 68: Model's page: Control Panel: Bookmarks.

8.5.6.1.9 Subsections

8.5.6.1.9.1 Downloadable Files

The downloadable files subsection has, always if there is any element, the follow sections:

- XMI.
- Source Code.
- Documentation.

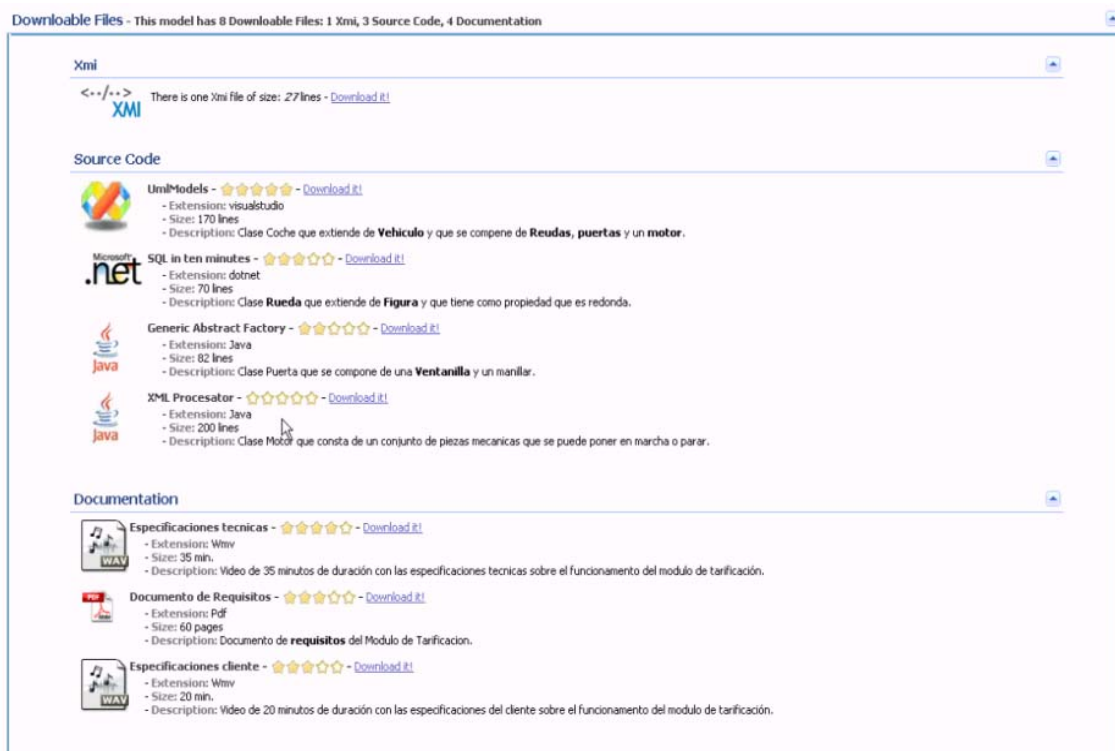


Illustration 69: Model's page: Downloadable files.

All the sections with this icon will be collapsible.

8.5.6.1.9.1.1 Unregistered Users

Downloads are not allow for unregistered users but it will show the information of the files and download link as the registered users. This will act as an appeal for the unregistered users to register into UmlModels.

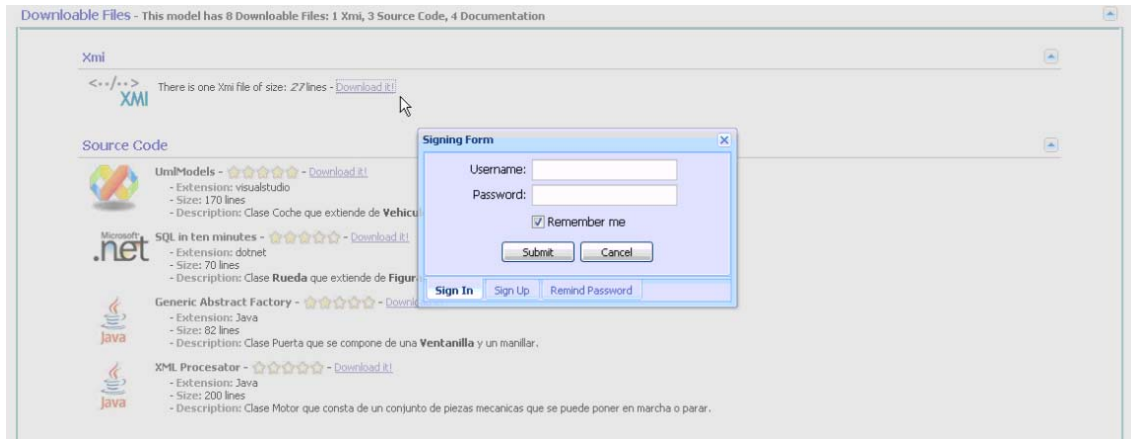


Illustration 70: Model's page: Downloading files. Unregistered user.

8.5.6.1.9.1.2 Registered Users

8.5.6.1.9.1.2.1 Chargeable

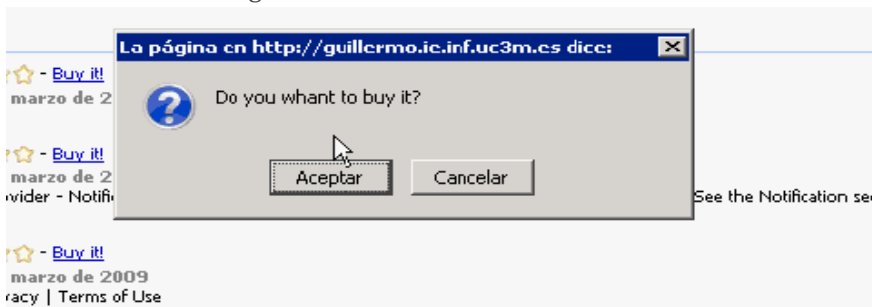


Illustration 71: Model's page: Downloadable Files: Chargeable.

8.5.6.1.9.1.2.2 Non-Chargeable

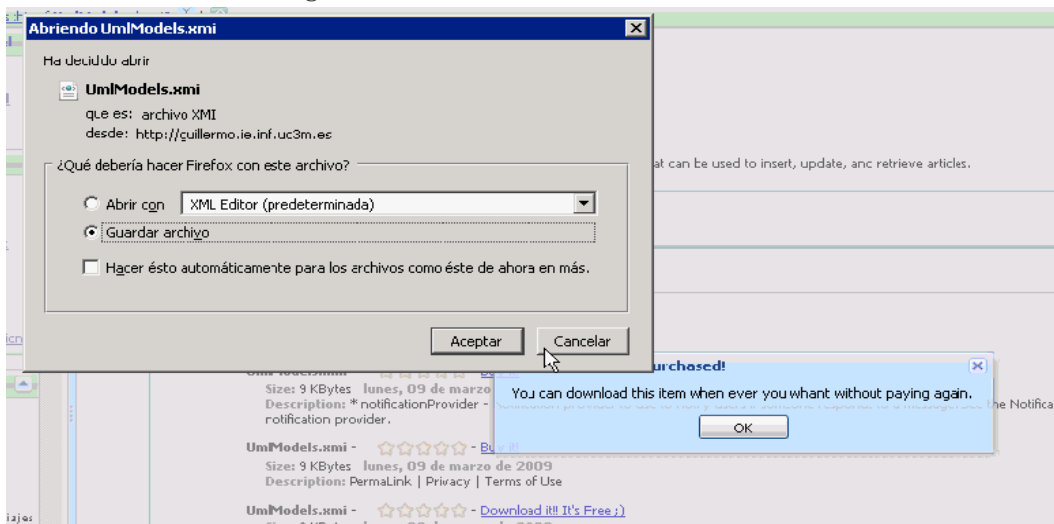


Illustration 72: Model's page: Downloadable Files: Non-Chargeable.

8.5.6.1.10 Requirements

This block is similar than the download files.

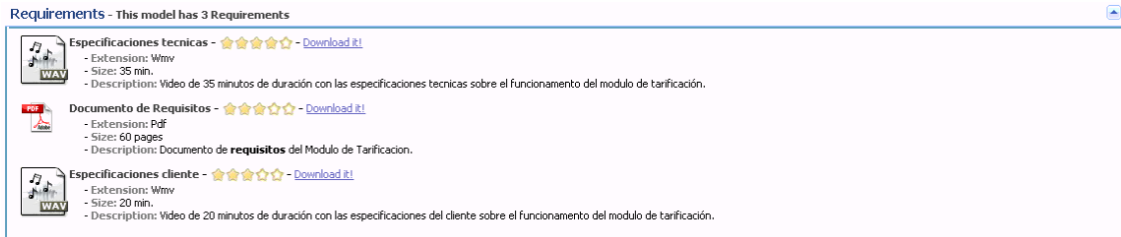


Illustration 73: Model's page: Requirements.

8.5.6.1.11 Descriptions

8.5.6.1.11.1 Unregistered users

The other descriptions are restricted to registered users. This will act as an appeal for the unregistered users to register into UmlModels.



Illustration 74: Model's page: Descriptions. Unregistered user view.

8.5.6.1.11.2 Registered and Unregistered Users

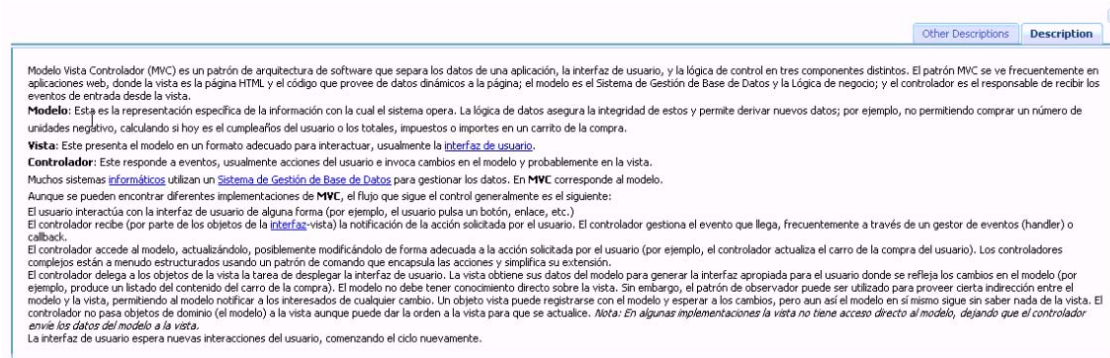


Illustration 75: Model's page: Descriptions. Registered and Unregistered users view.

8.5.6.1.12 Forum

The forum will have to be similar as code project forum. A first approach to a simple forum based on a grid:

Subject	User	Posted
Problema con tarificacion.cs User: Guillermo	Guillermo	09/01/2009
Post: No vamos a ser rehenes de recursos que se están agotando, de gobiernos hostiles, y de una Tierra que se calienta". De esta manera, el presidente de Estados Unidos, Barack Obama, ha sentenciado que durante su Gobierno reducirá la dependencia del petróleo e impulsará el uso de energías de bajo consumo como medio para combatir el cambio climático. Obama ha afirmado que la dependencia del crudo extranjero y el cambio climático representan "amenazas urgentes para la seguridad nacional" estadounidense.		
Mejora propuesta	jlio'ens	09/01/2009
Un ejemplo de lo aburrido que es esto	jurano	09/01/2009
Thanks	jorjo	09/01/2009
Problema con tarificacion.cs	Guillermo	09/01/2009
Mejora propuesta	jlio'ens	09/01/2009
Un ejemplo de lo aburrido que es esto	jurano	09/01/2009
Thanks	jorjo	09/01/2009
Problema con tarificacion.cs	Guillermo	09/01/2009
Mejora propuesta	jlio'ens	09/01/2009
Un ejemplo de lo aburrido que es esto	jurano	09/01/2009
Thanks	jorjo	09/01/2009
Problema con tarificacion.cs	Guillermo	09/01/2009

Illustration 76: Model's page: Forum. First approach.

Reply, new post and edit are not implemented waiting for the search of a forum already implemented. I nothing that fulfils our requirements is found on EXT technology, a new one would have to be implemented on the continuation of my project.

8.5.6.1.13 Tooltips

With the use of tooltips, all the elements are self explained:



Illustration 77: Model's page: Tooltips.

8.5.6.1.14 PayPal

The registered users have the possibility to buy more credit on the Header section.

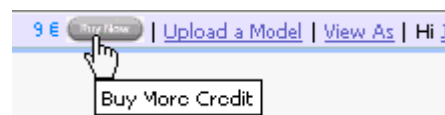


Illustration 78: Model's Page. Buy Credit.

This action will open a new tab with the UmlModels account in PayPal where they can easily buy more credit.

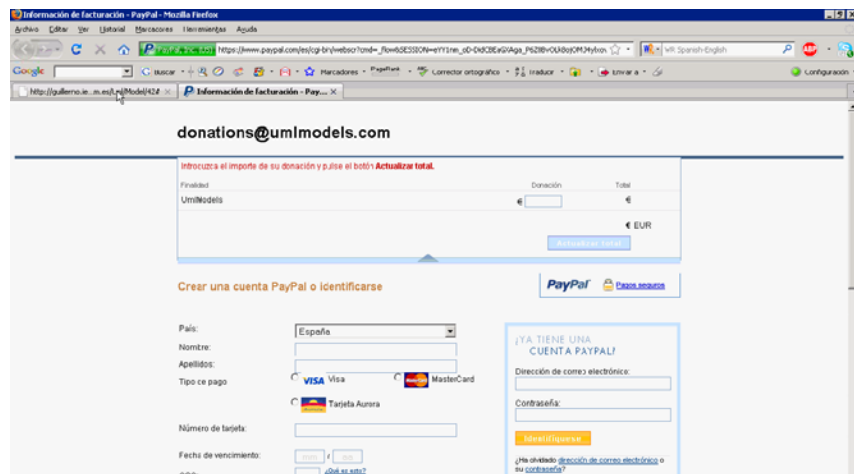


Illustration 79: Model's Page: PayPal.

8.5.6.1.15 HTTPS

A problem found that must be corrected in the further works, is that the EXT Ajax libraries that we are using goes under http and should be under https. And since the log in form is a done with JavaScript, if we use an SSL connection for the password and the user name, the user won't realize it since the main URL is under http.

8.5.6.1.16 Punctuation

A user can punctuate the resources. Each rate can be changed as many times as the user wants; the old rate is replaced for the new one and not added again because we store in the database all the punctuations per user and resource. This is important for the average of the rates and this is informed to the user to prevent having users rating their own models and bothering the database.



Illustration 80: Model's page: Punctuation of a resource.

Once the user has rate a resource, the punctuation that the user sees is the average of all the punctuations. It can be considered to show the punctuation that you have made.

8.5.6.2 Resultant Work

8.5.6.2.1 Unregistered User

Mainly in this view, the unregistered user has to be able to see all the things that an unregistered users can do but without functionality. And in all moment have to be visible a link for being able to join to UmlModels.

An unregistered user won't be able to download any file but must to be able to see the link, the description... Neither will be able to post on the forums, add any resource (model, description, title...). With this we want attract the user to become part of UmlModels.

An unregistered user won't be able to see the additional titles neither the descriptions.

See *Illustration 81: Model's Page: Resultant work. Unregistered user.*

8.5.6.2.2 Registered User

The registered user will have all the functionalities available. Also registered users will have a menu on the left with all the advanced operations that the registered users are able to do.

See *Illustration 82: Model's Page: Resultant work. Registered user.*

Unimodels | What is this of Unimodels about? | Upload a Model | [Sign In](#) | [Join Us Now](#)

Unimodels | [Requirements: 1](#) | [Descriptions: 8](#) | [Downloads: 7](#) | [12](#) | [11](#) | [11](#)

Unimodels | [Sponsored Links](#)

http://ocw.mit.edu/ocw-open-.../pn01_files/image091.jpg | [Report bad content](#) | [☆☆☆☆](#)

[Images added by other users](#)

[More Images of this Model](#)

[http://gdlitz.ing.unipam.edu.ar/downloads/pdfs/introduccion/UML.PDF](#)

Downloadable Files - This model has 30 Downloadable Files: 7 Xmi, 12 Source Code, 11 Documentation

Xmi

- Unimodels.xmi - [☆☆☆☆](#) - [Buy It](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009
Description:
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009
Description: *you can support advanced or customized user interfaces (accessibility, globalization, WAP, XML/RFP, AJAX, ...) or use a different presentation layer but the same business layer
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009
Description: Eucalypto is an open source .NET business/server library to help write ASP.NET content management web sites. Eucalypto currently supports these features:
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009
Description: Eucalypto currently supports these features: library to help write ASP.NET content management web sites.
- Unimodels.xmi - [☆☆☆☆](#) - [Buy It](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009
- Unimodels.xmi - [☆☆☆☆](#) - [Buy It](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
Size: 9 KiBytes | lunes, 09 de marzo de 2009

Documentation

- Doc: Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Doc
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description:
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Audio
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description: * Eucalypto.Profile.EucalyptoProfileProvider - An implementation of System.Web.Profile.ProfileProvider.
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Xp
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Audio
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description: //Create a category
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Document
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description:
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Rfp
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Document
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description: *you can support advanced or customized user interfaces (accessibility, globalization, WAP, XML/RFP, AJAX, ...) or use a different presentation layer but the same business layer
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Xp
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Wmv
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description:
- Doc: Unimodels.xmi - [☆☆☆☆](#) - [Download it! It's free!](#)
- Extension: Doc
- Size: 9 KiBytes
- Upload: lunes, 09 de marzo de 2009
- Description:

Requirements - This model has 1 Requirements

Brief blog - [☆☆☆☆](#)

- Upload: lunes, 09 de marzo de 2009
- Description: Eucalypto.News.NewsManager.CreateCategory("Test", "Test");

[Other Descriptions](#) | [Description](#)

Up! This feature is only available for registered users. Register now, it's free registered users.
- If you are not a registered user, you can become one by [registering](#). It's free!
- If you are a member, please login here.

Forum - New Post

Subject	User	Posted
Problema con tarificación.cs	Guillermo	09/01/2009
Mejora propuesta	Joreno	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Thanks	Jorgo	09/01/2009
Problema con tarificación.cs	Guillermo	09/01/2009
Mejora propuesta	Joreno	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Thanks	Jorgo	09/01/2009

Un vamos a ser reñados de recursos que se están agotando, de gobiernos hostiles, y de una Tierra que se calienta". De esta manera, el presidente de Estados Unidos, Barack Obama, ha sentenciado que durante su Gobierno reducirá la dependencia del petróleo e impulsará el uso de energías de bajo consumo como medio para combatir el cambio climático. Obama ha afirmado que la dependencia del crudo extranjero y el cambio climático representan "amenazas urgentes para la seguridad nacional" estadounidense.

Reply

Problema con tarificación.cs	Guillermo	09/01/2009
Mejora propuesta	Joreno	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Thanks	Jorgo	09/01/2009
Problema con tarificación.cs	Guillermo	09/01/2009
Mejora propuesta	Joreno	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Thanks	Jorgo	09/01/2009
Problema con tarificación.cs	Guillermo	09/01/2009
Mejora propuesta	Joreno	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Thanks	Jorgo	09/01/2009
Problema con tarificación.cs	Guillermo	09/01/2009
Mejora propuesta	Joreno	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Thanks	Jorgo	09/01/2009

Illustration 81: Model's Page: Resultant work. Unregistered user.

6 feeds | Infoque Diferencial | RSS de es.com - Noticias general | 10 feeds | Suscrito - Visualizar - Anular suscripción - Marca.com - Fútbol Liga de Campeones | Pruebasasaf | Model's Page | ConceptDraw Model | User Av | Hi Gullermo Sun, 04

ConceptDraw Developer Site Structure

Images added by other users

Downloadable Files - This model has 27 Downloadable files: 6 Xmi, 11 Source Code, 10 Documentation

Xmi

- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009

Documentation

- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009
- Uml#Model.xmi - 9 Kbytes | Issues, 09 de marzo de 2009

Other Descriptions

Description	User	Posted	Rate
SQL Server 2005 - DriverClassHibernata Driver SqlClientDriver, DetectHibernata Detect MsSql2005Detect, Data Source=localhost; Catalog=Equity;Integrated Security=SSPI;NotificationProvider=UserCenter\validation\validationUser (Everything else Copyright © CodeProject, 1999-2009	Jurbano	09/01/2009	0
Uml#Model.xmi - 9 Kbytes Issues, 09 de marzo de 2009	Jurbano	09/01/2009	1
Uml#Model.xmi - 9 Kbytes Issues, 09 de marzo de 2009	Jurbano	09/01/2009	4

Forum - New Post

Subject	User	Posted
Problema con tarificación.cs	Gullermo	09/01/2009
Mejora propuesta	Jurbano	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Problema con tarificación.cs	Gullermo	09/01/2009
Mejora propuesta	Jurbano	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Problema con tarificación.cs	Gullermo	09/01/2009
Mejora propuesta	Jurbano	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Problema con tarificación.cs	Gullermo	09/01/2009
Mejora propuesta	Jurbano	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Problema con tarificación.cs	Gullermo	09/01/2009
Mejora propuesta	Jurbano	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009
Problema con tarificación.cs	Gullermo	09/01/2009
Mejora propuesta	Jurbano	09/01/2009
Un ejemplo de lo divertido que es esto	Jurbano	09/01/2009

Illustration 82: Model's Page: Resultant work. Registered user.

8.6 Conceptual Model

In this section is described the conceptual model designed.

First is defined what is *An Artifact for UmlModels*; Second is described the difference for UmlModels between a *Model and a UML Diagram*; Next is described the *Advanced information model*, in which is described how a diagram is processed nowadays and how will be in the future, as well as some conceptual specifications for the design GUI and a formal definition of what is a Model for UmlModels; and finally is represented the *Conceptual Model Diagram* as well as some requirements associated the model and some considerations about private and collaborative models, trazability, chargeable and non chargeable money ,...

8.6.1 An Artifact for UmlModels

According to Jacobson in the Unified Modelling Language (UML) and the Unified Process (UP), an artifact is “a tangible piece of information that (1) is created, changed, and used by workers when performing activities, (2) represents an area of responsibility, and (3) is likely to be put under separate version control. An artifact can be a model, a model element, or a document.”

An artifact then will be all those things that can exist in the software engineering (UML diagram, document, equation, requirements...).

UmlModels Site is going to have a repository containing Software Artifacts. Those Artifacts are defined as follows.

It is remarkable to include some interesting reflexive relationships to the Artifact.

The artifacts have a few reflexive relationships:

- An artifact can be traced against others.
- An artifact can have other software artifacts related to it.
- An artifact can have versions.

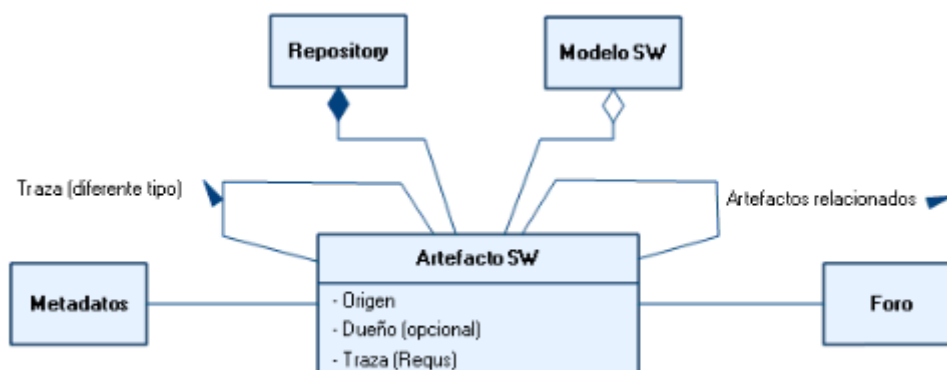


Illustration 83: Conceptual Model: Specifications of an Artifact

Moreover, as it is shown in *Illustration 83: Conceptual Model: Specifications of an Artifact*, the Artifacts are aggregated to a Model. And each Artifact will contain a forum and some metadata such as inclusion date, modification date ...

In addition, an artifact can be from one of those different types:

- Document.
- Equation.
- Diagram.
- Source Code.
- Executable Component.
- Test Code.
- Requirement.

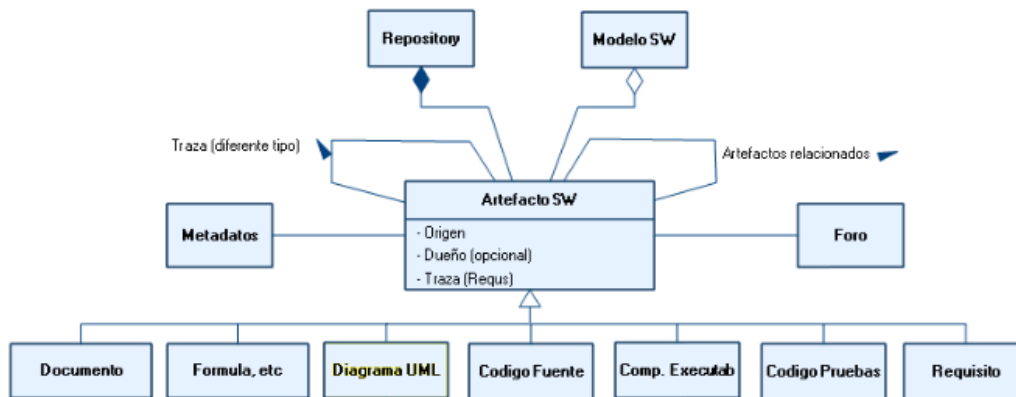


Illustration 84: Conceptual Model: Detailed Specifications of an Artifact

At this moment of the project, the most important element for us will be the UML diagram. In the future they will others come.

8.6.2 Model vs. UML Diagram in UmlModels

At the first phases of development it won't be any conceptual differences when we talk about Model or UML Diagram.

When we get a diagram (from somewhere not defined yet), we are going to transform it into a model by generating what we need.

Being more specifics, an UmlModels diagram can have any of the next elements:

- An Image.
- A XMI.
- An Hyperlink to source.
- An Hyperlink to the owner.
- A RSHP File.
- A tubular vision.
- Many Videos.
- Many related Source Code Packages.

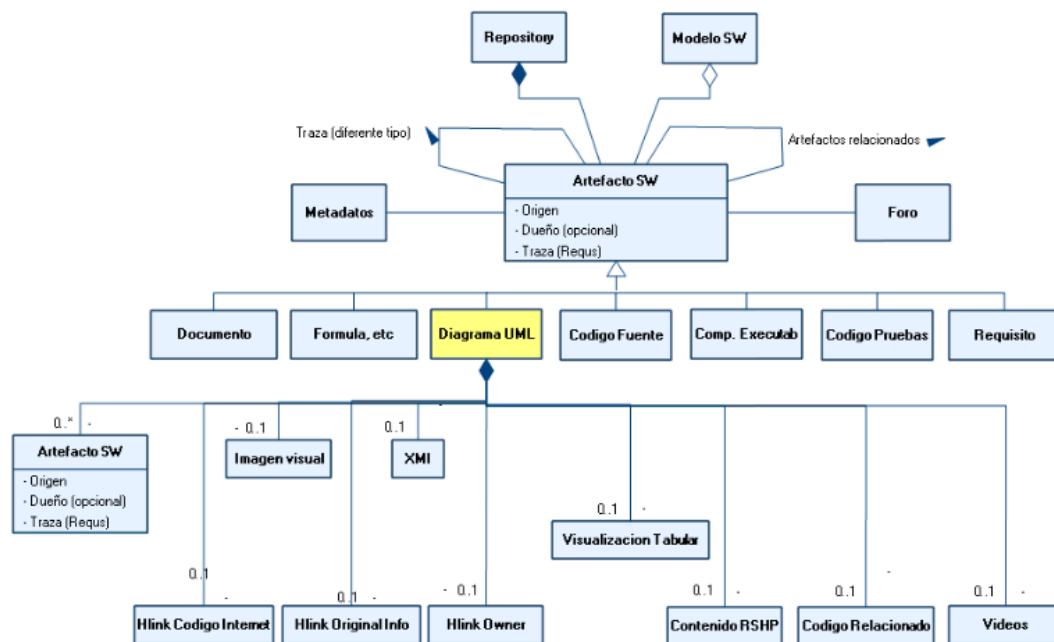


Illustration 85: Conceptual Model: UmlModels Site Information Model

Note that not all the UML Diagrams have an image. This can happen when the diagram is been generated from other elements like assemblies, source code, etc...

To sum up, mainly the Site is going to deal with diagrams. A diagram is a graphical representation of a part of a model while the model is the visual one:

- **Model:** A model has the semantic of the representation and the diagram shows.
- **Diagram:** A Diagram has the visual appearance of what it is been represented.

8.6.3 Advanced Information model

8.6.3.1 Processing a diagram nowadays: Not Human Indexer

Nowadays, when the crawler finds an image through the Internet we will make it a model and it will be treated and indexed as it. This will be done extracting all the information from the diagram with an image processor. Once the information of the model is indexed, the diagram will be assigned to the model and it will be treated as it.

On the other hand, when a crawler finds an assembly and many namespaces are found, the indexer will make a model from each assembly.

Therefore, for the images indexer: *A model is a Diagram* and *A Diagram is a Model*.



Illustration 86: Diagram Point of View

At the same time, for the assemblies' indexer: It is possible to have many namespaces therefore many models but none image will be. Since it is possible to build an image from a model, the relation will be:



Illustration 87: Assemblies Point of View

8.6.3.2 Processing a diagram in the future: Human Indexer

In the future, the site will have a commercial look, and it will be possible to do business on it.

One of the possibilities will be to make something similar as a bank job where the users can upload models (diagrams), a description... and pay for the programming, testing, etc.

According to the idea that a user would be able to create and upload models, it won't fit the idea of associate one diagram to one model. A user can upload more than one diagram to a single model. For this reason, a model will be formed by none, one or many diagrams.

Actually, UML says that a model has views; and these views can be formed by many components from which one of it is a diagram.

This way, with the follow representation, we have a model perfectly represented and compatible with the way we are processing the models and the way we will.

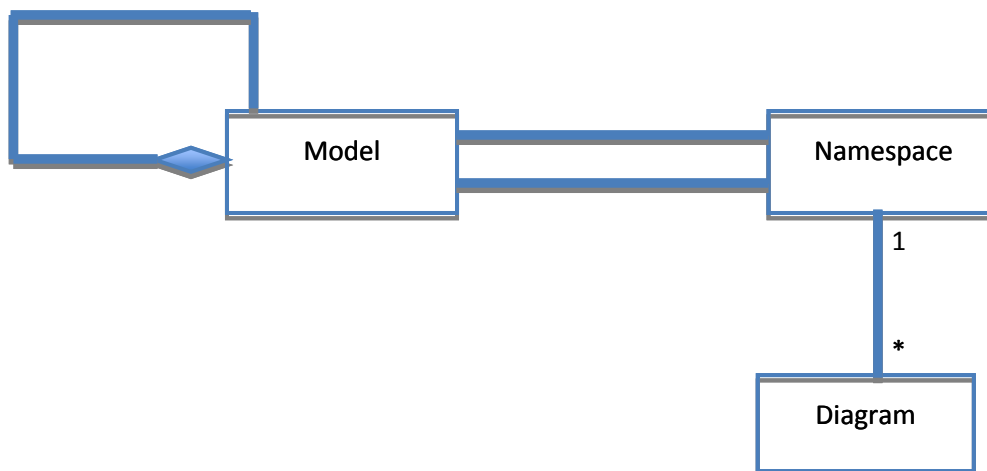


Illustration 88: Software Model Diagram

8.6.3.3 UmlModels GUI

The GUI must support three different kinds of configurations:

1. When one model is associated with a diagram in the way one to one; in which the GUI shows the diagram a part of the model.
2. When one model is made of many diagrams.
3. When one diagram is part of many models.

Number three will be done in the future. At the moment only Model one to one and one to many will be considered. Subsequently the latter is described in more detail:

8.6.3.3.1 Model 1-1

Basically, the main idea is to manage the model as if we had only a diagram, but leaving the door open to the other two possibilities. Considering the possibility in which a diagram have more than one image, then will have the follow situation:



Illustration 89: Model to Diagram

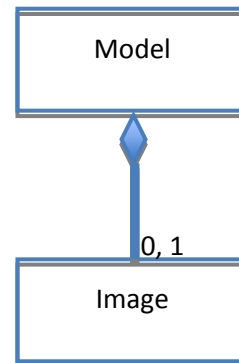


Illustration 90: Model to Images

In the future, the GUI will have the capacity to support one model to many diagrams. And since nowadays there is no diagram generator from an assembly, the GUI will have to support diagrams without an image.

8.6.3.4 What is a Model for UmlModels?

A model for UmlModels is an artifact [see An Artifact for UmlModels] compound of three groups of information:

1. The intrinsic information about the model (image, description ...).
2. The collaborative information about the model:
 - a. Restricted information added by the owner.
 - b. Public information added by the owner.
 - c. Information added by other users.
3. Commercial information about the model.

8.6.4 Conceptual Model Diagram

This diagram (Illustration 91: Conceptual Model) represents conceptually the vision of UmlModels on the first phase of development. However on the first implementation phase the optimum access to the database will prevail.

All the components that nowadays represent a model in UmlModels are represented on the diagram. This is:

- Title.
- Description.
- Image.
- XMI.
- Source Code.
- Additional Documentation.
- Job Offers.
- Requirements.
- Tags.
- Posts.

All the components are going to be treated as **resources**, which can be *punctuables* and *linkables*.

Additionally, a user can be the owner of many resources. At the same time, each model (which will be a linkable and punctuable resource) will contain resources.

Moreover, each of the resources can have a different owner from the owner of the models. And when the owner of the resource is the same as the owner of the model then it will be more significant for the GUI.

We can have two different type of resources: **versionable** and non **versionable**. When a resource is versionable, then different versions of that resource associated to the original resource can exist. The post and the offers are the only non versionable resources.

As well, the resources can be **downloadables** and non **downloadable**. Non downloadable resources will be shown through the browser. And downloadable resources will be returned to the users so they can download them. Downloadable resources are not going to be accessible for non-registered users, and a small amount of money can be charged for the download.

Since a model by itself is going to be a resource, a model can be compound by other models.

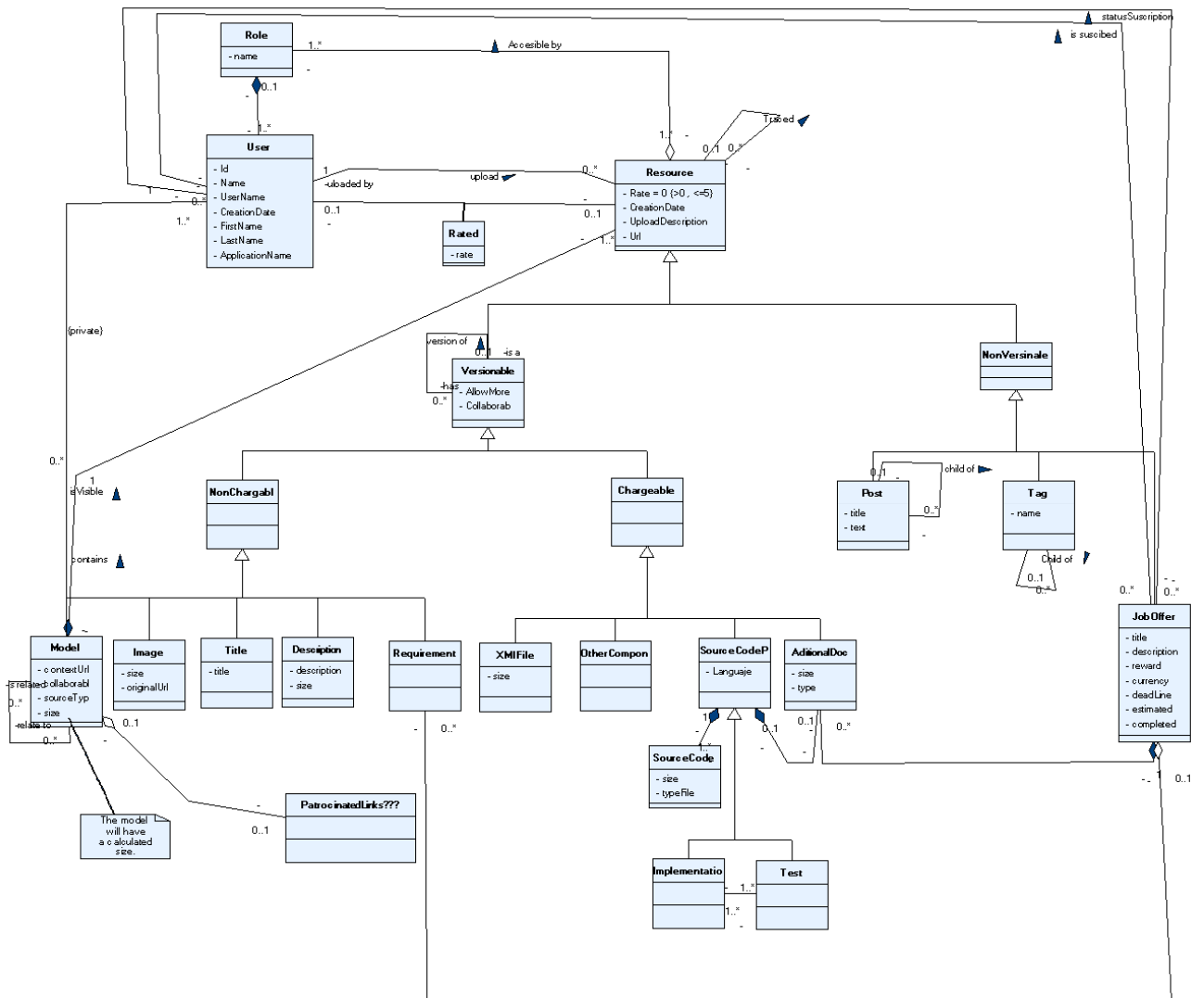


Illustration 91: Conceptual Model: Diagram.

8.6.4.1 Requirements associated to a Model

Even though in the future the artifacts (and not the models) will contain the requirements, in this first phase it is important that the GUI shows that the Model is going to have requirements associated to it; so they will be included as part of the UmlModel.

It has been decided that the requirements won't be chargeable.

8.6.4.2 Private vs. Collaborative Model

A private model is not exposed to public review (it does not enter the public area) while a collaborative model is that one which allows other users add additional content to the model.

The additional content that can be added to the model will be:

- Additional titles.
- Additional Descriptions.
- Additional Images.

A model can allow collaborations and can be private at the same time. In this way, the access to the private models will have to be authorized by the owner of the model. The private model additionally can either be or not collaborative.

8.6.4.3 Trazability

Trazability means the possibility to recognize the semantic evolution of an artifact. This is, to navigate temporary and/or spatially in the evolution of an artifact from its initial phases of a software project to its lasts phases.

For example, the trazability can let you see how a requirement is exploited into a formal specification of the requirement through, for instance, a use case diagram; or how a use case exploits into a collaboration; Etc.

In other words, trazability is the ability of know the evolved path of the software development.

Theoretically, the most classic definition of trazability is the one described above. In practice, thanks to the new software development technologies⁴, the trazability is a direct implication in the software development based on the transformation of models: Given a high level model, the idea is to transform it with meta-transformations until I get something that allows me to generate code.

With this software development point of view, the trazability can not only have to be between a requirement or a diagram; or a diagram and a source code. It is starting to happen other possibilities of trazability on the aim of UmlModels such us the trazability between a model and other implemented model.

In an approach to this conception we are going to support trazability between resources. This way, in the future any other new resource such as risks, software specifications ... will also support trazability.

8.6.4.4 Source Code and Tests

A model can have source code associated to it. The source code will be compound of a package with many files and folders with an extension like: ZIP, RAR, etc.

A model can have many packages and the login unit of download will be the package and no the file.

8.6.4.5 Money: Chargeable and Non Chargeable.

There are two different kinds on money in UmlModels:

- ➔ Changeable: Produced as the result of a job offer.
- ➔ Non changeable: Produced by the models with the chargeable resources.

A little amount of Money will be charged for the download of the following resources:

⁴ MDA: Model-Driven Software Architecture, MDD: Model Driven Development, MDA: Model Driven Engineering, MDSD: Model Driven Software Development, MDSE: Model Driven Software Engineering,

- XMI.
- Source Code.
- Other components without source code (such as a DLL, ...).

This money is known as chargeable while the one that is given as a reward by the job offers is known as changeable. This balance that comes from the offers can be changed for real money.

8.6.4.6 Job Offers in Private Models

It has not been defined yet if a job offer will be on the private area or not. Personally, I think that if the model is public it should be as at the moment. And if it is **private**, then we could find two possible situations:

- The owner sets the **offer as public**. In this case, the user can only see the offer; they won't be able to see the model. The users can subscribe to the offer and only when the owner accepts this user to do the offer, then the user automatically will be able to access to this model. Later on, once the user finishes the offer, the owner can remove the privileges of the user.
- The owner sets the **offer as private** (restricted). Only the users with privileges to see the model will see the offer.

8.7 Detailed Design

An N-Tier Client-Server architectural pattern has been defined and described in the *Architecture* section. Specifically we have defined the follow three packages following detailed:

- Client:
 - o Presentation GUI Tier.
 - o Controller Tier.
- Server:
 - o Presentation Logic Tier.
 - o Controller Tier.
 - o Business Tier.
 - o Data Access Tier.
- Data Base:
 - o Data Tier.

For the implementation it is going to be used the **ASP.NET MVC Framework**. This framework is already defined in the state of the art and provides an easy and natural way to separate the *View*, the *Controller* and the *Model*.

Actually, the *View* will represent the *Presentation Tier* defined in our architecture while the *Model* will be represented by the *Business Tier*.

After the *Firs Approach: DotNetNuke* we show that the CMS were not flexible enough to make what we want. And since the Site is not going to contain loads of sections that requires an easy and fast way to manage but it is necessary to have an environment with a powerful visual tools; then the JavaScript (JS) **EXT Framework** will be used to present the results.

Additionally, we have studied the use of ADO.NET O-RMs (See '*Object-Relational Mapping in ADO.NET*') in order to reduce costs: the amount of code is fewer while the software is more robust with this technique. The three techniques proposed were Data Sets, Linq to SQL and Entity Framework. The idea of using Data Set was rule out because we are not going to have transactions that require the insertion of many rows. Between the use of LINQ to SQL and Entity framework, at the beginning we decided to use LINQ to SQL because Entity framework doesn't have implemented some small features. But, according to the recommendations of Microsoft⁵ and bearing in mind that these not implemented features are not functional, we are going to use **Entity Framework**.

Finally, the database that we are going to use is a **Microsoft SQL Server 2005**.

⁵ We find that our model fulfils many of the recommended features for the use Entity Framework instead Linq to SQL described in '*Comparison: Data Set vs. Entity Framework vs. Linq to SQL*' but we find two especially important for us: First we have many to many relationships and we can map a single class to multiple tables. The other one can be really interesting to gather the information related to a resource in one class such as the punctuation table and resource table.

All the technologies chosen, subsequently is presented where this technologies are used in our architectural design and the way in which they interact:

- Client:
 - o Presentation GUI Tier: UI components from EXT Framework.
 - o Controller Tier: JavaScript.
- Server:
 - o Presentation Logic Tier: EXT Components, View Component from the ASP.NET MVC Framework.
 - o Controller Tier: Controller component from the ASP.NET MVC Framework.
 - o Business Tier: Model component from the ASP.NET MVC Framework.
 - o Data Access Tier: ADO.NET Entity Framework.
- Data Base:
 - o Data Tier: Microsoft SQL Server 2005.

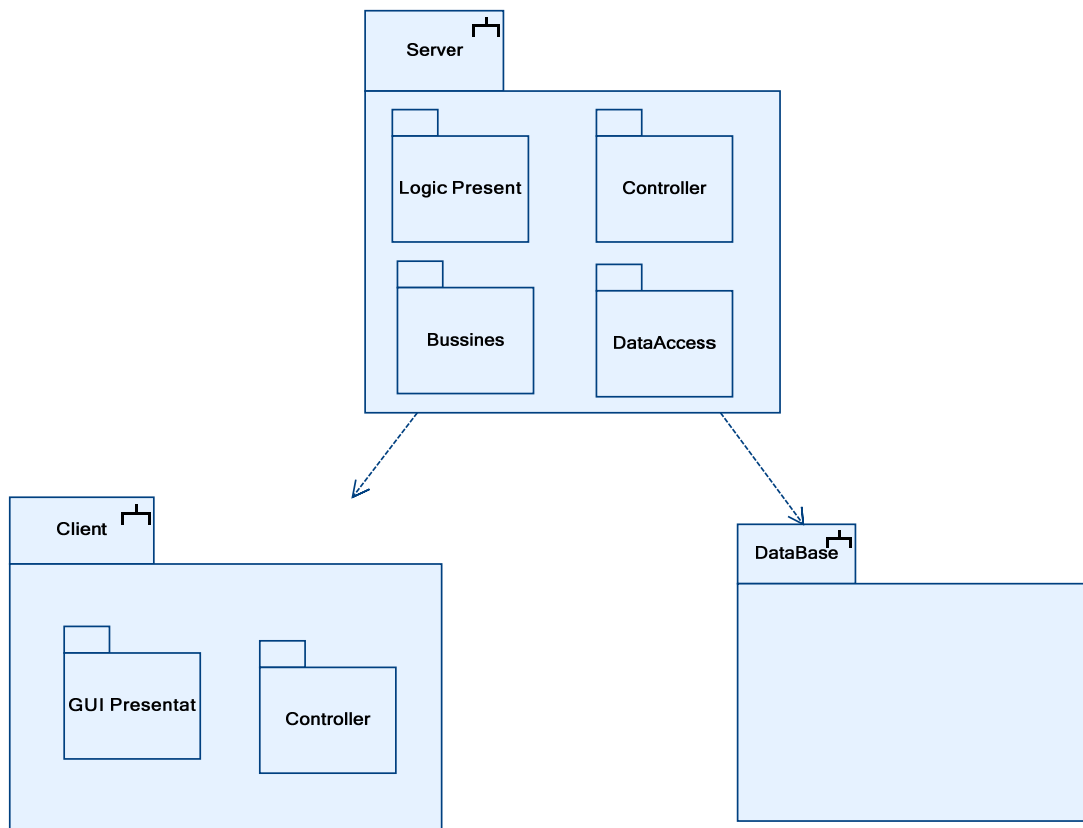


Illustration 92: Detailed Design: Subsystems

Subsequently are described in detail each of the packages:

8.7.1 Server

The server is in charge of attending the requests of the users, retrieve the data and the present it to the client.

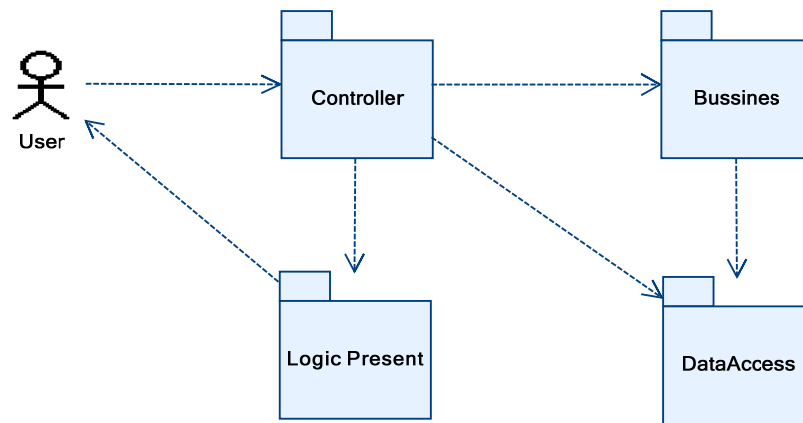


Illustration 93: Detailed Design: Server Subsystems

In this section is described all the packages within the Server: **Controller Tier, Presentation Logic Tier, Business Tier** and **Data Access Tier**.

8.7.1.1 Controller Tier

The controller is in charge of attending the users requests performed from the GUI and take the proper actions by calling to the business and/or the data access packages. The follow controllers are defined:

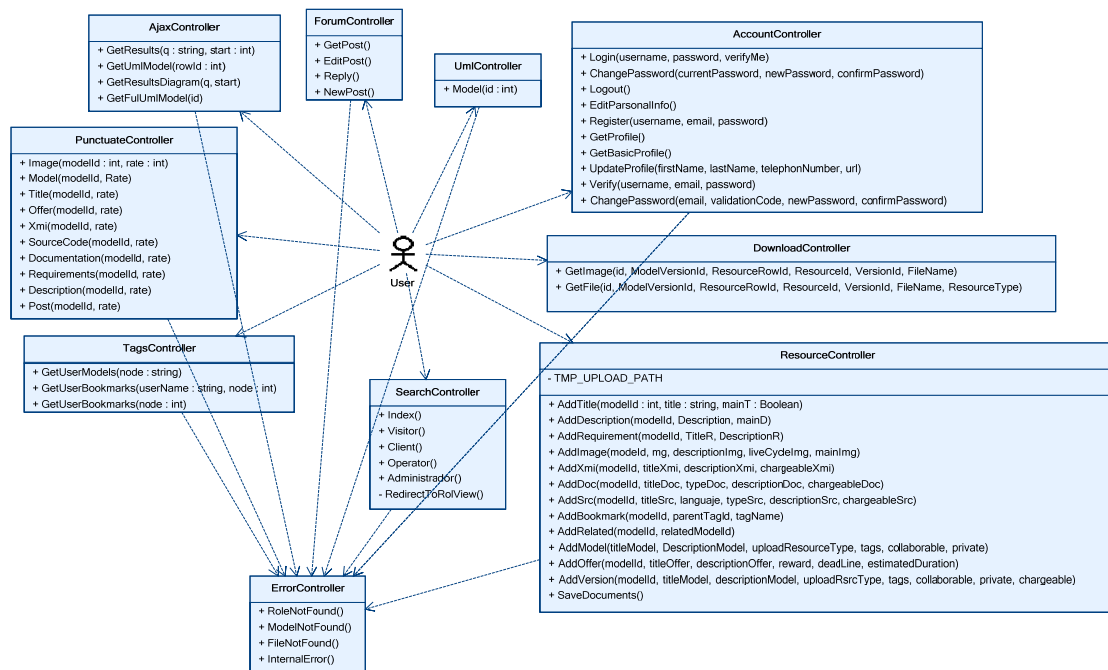


Illustration 94: Detailed Design: Controller Class Diagram

Subsequently is explained each of the class shown in the diagram as well as some implementations details:

- **AjaxController:** This class contains all things needed to attend the AJAX requests for the search results (textual search, diagram search ...) and the model. This class is added to implement the approach *Presentation GUI Tier* section. Since in the future is probably that this can be implemented using the other approach presented in the same section, then all this actions are included in this class to minimize the classes to be changed.
 - o It is also simulated the Polar retrieval here by performing a normal search in the title and description rows.
- **PunctuateController:** This controller gets the requests performed by the client for punctuating the resources.
 - o Checks parameters. A rate must be from 0 to 5. A HTTP 400 status code (Bad Request) is returned with an error message if the parameters are not correct.
 - o Checks that the action comes from a signed-in user. Since the **RoleProvider** is implemented: this is done in a simple step by adding the "**Authorize**" property to the methods. All the methods from this controller must have this property.
 - o A call to the Rate method in the Rate class (**RateUmlModel**, **RateImage** ...) of the data access package is performed with the name of the user (stored in the context), the id of the resource and the rate.
- **AccountController:** Provides all the needed actions to manage the account of a registered user (change password, change profile ...) and the actions required by the unregistered users to become a registered user (sign in and sign up).
 - o Checks parameters received from the net.
 - o When needed, checks that the action comes from a signed-in user (similar than in the *PunctuateController*).
 - o Perform on of the actions defined in the AccountController detailed in the Illustration 71: Detailed Design: Controller Class Diagram by using the **MembershipProvider** defined in the **UmlModels.Site.User** package defined in the business layer.
- **DownloadController:** Provides the needed actions to return a downloadable file.
 - o Checks parameters received from the net.
 - o Checks that the action comes from a signed-in user (similar than in the *PunctuateController*):
 - For the Images: Checks that it is a signed-in user (no matter the role whom belongs). Visitors are allowed to view the images.
 - For the rest of the files: Checks that the user at least has the "Client" role ("*[Authorize(Roles = "Client")]*") property). Visitors are not allowed to download these files.
 - o Depending on the file type of the download (all the valid file types are defined in the **FileSupportUtils** class from the **UmlModels.Site.Storage** package of the business layer):
 - It is call to the CheckAndGo method defined in the UmlModels.Site.DownloadAndChargin package from the business layer.
 - If the download is authorized (role and balance enough), then the path of the file is returned. Otherwise an exception is raised.
 - A DownloadResult (specification of ActionResult) is returned to the user or an error message is printed.
- **ResourceController:** Provides the needed actions to add any resource or version of the resource.

- Checks parameters received from the net.
- Checks that the action comes from a signed-in user (similar than in the *PunctuateController*).
- If needed, uploads to the server the enclosed files (Images, XMIs, additional documentation or source code files):
 - Checks that the files have less than the maximum defined:
 - 5 MB for the XMI files.
 - 10 MB for the images.
 - 10 MB for the additional documentation
 - 30 MB for the source code package.
- A call to the proper *AddResource* method defined in the *UmlModels.Site.Resources* package from the data access layer is performed.
- An information message is returned to the GUI.
- **SearchController:** This class returns a different View according to the role of the user. These Views are defined in the *Presentation Logic Tier* section.
 - If the user is a Client, Operator or Administrator then a SearchResult “**Client**” view is returned.
 - If the user is a not registered, then a SearchResult “**Visitor**” view is returned. This view has les information and functionalities.
- **TagsController:** This class returns either the resources of a user as an asynchronous tree or his bookmarks.
 - Checks parameters received from the net.
 - Checks that the action comes from a signed-in user and at least with “Client” role.
- **UmlController:** This class returns a different View according to the role of the user. These Views are defined in the *Presentation Logic Tier* section.
 - If the user is a Client, Operator or Administrator then a UmlModel “**Client**” view is returned.
 - If the user is a not registered, then a UmlModel “**Visitor**” view is returned. This view has les information and functionalities.
 - If the HTTP request has the **if-modified-since** header, the *LastDateResourceAction* is checked from the database by calling to the proper method in the data access layer. This request is done by the search engines such as Google and is one of the actions requested by Google to be a Google friendly page.
 - If the model was modified, then it is returned.
 - If not, then a HTTP 304 status code (Not Modified) is returned.
 - To fulfil the Google indexing requirement, the information of the model is sent to the View through the **ViewData** structure defined by *ASP.NET MVC Framework*. For more detail see two approaches in *Presentation GUI Tier* and in the meeting reports.
- **ErrorController:** Returns to the user an HTTP Status code error for the following situations:
 - *RoleNotFound*: HTTP 501. When the user is authenticated but is not in any known Role.
 - *ModelNotFound*: HTTP 501. When the model requested does not exist.
 - *FileNotFound*: HTTP 404. When the file requested was not found.
 - *InternalError*: HTTP 500. When an unknown error has occurred.

8.7.1.2 Presentation Logic Tier

The presentation logic tier has two packages using each two different technologies. The first one is the View package in which is embedded the *ASP.NET MVC Framework View*. The other one is the Script package which contains all the JavaScript technology, including the *EXTJS Framework*; both used to present the information to the user.

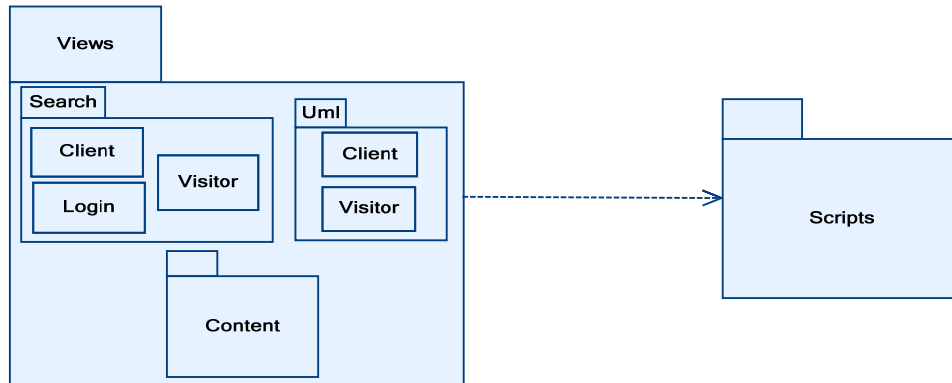


Illustration 95: Detailed Design: Presentation Logic Tier

Subsequently is explained each of the class shown in the diagram as well as some implementations details:

8.7.1.2.1 View

This package contains the two views used to present the GUI showed before in this document. In these views is returned to the client all the logic needed to build the GUI. Especially we have defined two views: one for the *Search Result* page called **Search View** and the other one for the *Model's Page* called **Uml View**.

Additionally this package contains a **Content** package, which contains all the graphical and CSS style of the GUI:

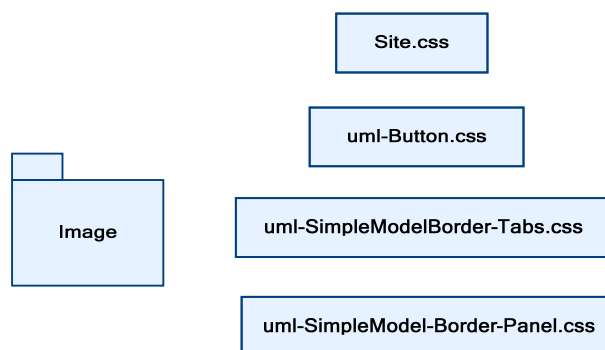


Illustration 96: Detailed Design: Presentation Logic Tier: Content Package

The Controller “knows” which view has to send to the *Presentation GUI Tier* and the View “knows” which content has to include depending on the role of the user: *Client* or *Visitor*.

Been more specifics, following is detailed which components of the Script package are sent to the in each View.

8.7.1.2.1.1 Search

If the user that is requesting the search result page is a Visitor, then the following components from the Scripts package are sent to the *Presentation GUI Tier*:

Component	Package
ActionSearchController	Scripts.UmlModels.Site.Master
Html	Scripts.UmlModels.ui
SponsoredPanel	Scripts.UmlModels.ui
CommonTemplates	Scripts.UmlModels.ui
SearchResultXTemplate	Scripts.UmlModels.ui
VirtualAssistancePanel	Scripts.UmlModels.ui
SearchForm	Scripts.UmlModels.ui
GeneralToolbar	Scripts.UmlModels.ui
GeneralToolbar.Unregistered	Scripts.UmlModels.ui
FormPanel	Scripts.UmlModels.ui
Window	Scripts.UmlModels.ui
SouthPanel	Scripts.UmlModels.SearchResult
ResultRecords	Scripts.UmlModels.SearchResult
SearchResult	Scripts.UmlModels.SearchResult
Panel	Scripts.UmlModels.SearchResult
Panel	Scripts.UmlModels.Search
Searching	Scripts.UmlModels.Search
signUp	Scripts.UmlModels.Users
remind	Scripts.UmlModels.Users
signIn	Scripts.UmlModels.Users
signing	Scripts.UmlModels.Users
Search	Scripts.UmlModels.Site.Master
Defaults	Scripts.UmlModels.Site.Master
Unregistered	Scripts.UmlModels.Site.Master
Site.Master	Scripts.UmlModels.Site.Master

Table 99: Detailed Design: Presentation Logic Tier: Components from the Search package related with components in the Scripts package. Visitor Role.

If the user that is requesting the search result page is a Client, then the following components from the Scripts package are sent to the *Presentation GUI Tier*:

Component	Package
Html	Scripts.UmlModels.ui
SponsoredPanel	Scripts.UmlModels.ui
CommonTemplates	Scripts.UmlModels.ui
SearchResultXTemplate	Scripts.UmlModels.ui
VirtualAssistancePanel	Scripts.UmlModels.ui
SearchForm	Scripts.UmlModels.ui
GeneralToolbar	Scripts.UmlModels.ui
GeneralToolbar.Registered	Scripts.UmlModels.ui
TagPanel	Scripts.UmlModels.ui
FileUploadField	Scripts.UmlModels.ui
FormPanel	Scripts.UmlModels.ui
Window	Scripts.UmlModels.ui
AddModel	Scripts.UmlModels.Resources
SouthPanel	Scripts.UmlModels.SearchResult
ResultRecords	Scripts.UmlModels.SearchResult
SearchResult	Scripts.UmlModels.SearchResult
Panel	Scripts.UmlModels.SearchResult
Panel	Scripts.UmlModels.Search
Searching	Scripts.UmlModels.Search
signOut	Scripts.UmlModels.Users

Component	Package
profile	Scripts.UmlModels.Users
Bookmarks	Scripts.UmlModels.Tags
MyModels	Scripts.UmlModels.Tags
SearchNavPanel	Scripts.UmlModels.Navigation
Search	Scripts.UmlModels.Site.Master
Defaults	Scripts.UmlModels.Site.Master
Registered	Scripts.UmlModels.Site.Master
Site.Master	Scripts.UmlModels.Site.Master

Table 100: Detailed Design: Presentation Logic Tier: Components from the Search package related with components in the Scripts package. Client Role.

8.7.1.2.1.2 Uml

If the user that is requesting the model's page is a Visitor, then the following components from the Scripts package are sent to the *Presentation GUI Tier*:

Component	Package
ActionModelControllerCommon	Scripts.UmlModels.Site.Master
ActionModelControllerUnreg	Scripts.UmlModels.Site.Master
RenderVisitorResources	Scripts.UmlModels.Resources
Html	Scripts.UmlModels.ui
RowExpander	Scripts.UmlModels.ui
PaddingRowExpander	Scripts.UmlModels.ui
SponsoredPanel	Scripts.UmlModels.ui
CommonTemplates	Scripts.UmlModels.ui
SearchResultXTemplate	Scripts.UmlModels.ui
ModelXTemplate	Scripts.UmlModels.ui
VirtualAssistancePanel	Scripts.UmlModels.ui
SearchForm	Scripts.UmlModels.ui
GeneralToolbar	Scripts.UmlModels.ui
GeneralToolbar.Unregistered	Scripts.UmlModels.ui
FormPanel	Scripts.UmlModels.ui
Window	Scripts.UmlModels.ui
ModelTags	Scripts.UmlModels.Resources
ModelXmi	Scripts.UmlModels.Resources
ModelDescriptions	Scripts.UmlModels.Resources
ModelImages	Scripts.UmlModels.Resources
ModelOffers	Scripts.UmlModels.Resources
ModelSourceCode	Scripts.UmlModels.Resources
ModelRequirements	Scripts.UmlModels.Resources
ModelAdditionalDoc	Scripts.UmlModels.Resources
ForumPanel	Scripts.UmlModels.Forum
Connection	Scripts.UmlModels.UmlModel
ModelToolbar	Scripts.UmlModels.UmlModel
ComparationArea	Scripts.UmlModels.UmlModel
RelationModels	Scripts.UmlModels.UmlModel
Panel	Scripts.UmlModels.UmlModel
signUp	Scripts.UmlModels.Users
remind	Scripts.UmlModels.Users
signIn	Scripts.UmlModels.Users
signing	Scripts.UmlModels.Users
Model	Scripts.UmlModels.Site.Master
Defaults	Scripts.UmlModels.Site.Master
Unregistered	Scripts.UmlModels.Site.Master
Site.Master	Scripts.UmlModels.Site.Master

Table 101: Detailed Design: Presentation Logic Tier: Components from the Uml package related with components in the Scripts package. Visitor Role.

If the user that is requesting the model's page is a Client, then the following components from the Scripts package are sent to the *Presentation GUI Tier*:

Component	Package
ActionModelControllerCommon	Scripts.UmlModels.Site.Master
ActionModelControllerReg	Scripts.UmlModels.Site.Master
RenderClientResources	Scripts.UmlModels.Resources
Html	Scripts.UmlModels.ui
RowExpander	Scripts.UmlModels.ui
PaddingRowExpander	Scripts.UmlModels.ui
SponsoredPanel	Scripts.UmlModels.ui
CommonTemplates	Scripts.UmlModels.ui
SearchResultXTemplate	Scripts.UmlModels.ui
ModelXTemplate	Scripts.UmlModels.ui
VirtualAssistancePanel	Scripts.UmlModels.ui
SearchForm	Scripts.UmlModels.ui
GeneralToolbar	Scripts.UmlModels.ui
GeneralToolbar.Registered	Scripts.UmlModels.ui
FileUploadField	Scripts.UmlModels.ui
FormPanel	Scripts.UmlModels.ui
TagPanel	Scripts.UmlModels.ui
Window	Scripts.UmlModels.ui
ModelTags	Scripts.UmlModels.Resources
ModelXmi	Scripts.UmlModels.Resources
ModelDescriptions	Scripts.UmlModels.Resources
ModelImages	Scripts.UmlModels.Resources
ModelOffers	Scripts.UmlModels.Resources
ModelSourceCode	Scripts.UmlModels.Resources
ModelRequirements	Scripts.UmlModels.Resources
ModelAdditionalDoc	Scripts.UmlModels.Resources
AddResources	Scripts.UmlModels.Resources
AddModel	Scripts.UmlModels.Resources
ForumPanel	Scripts.UmlModels.Forum
Connection	Scripts.UmlModels.UmlModel
ModelToolbar	Scripts.UmlModels.UmlModel
ComparationArea	Scripts.UmlModels.UmlModel
RelationModels	Scripts.UmlModels.UmlModel
Panel	Scripts.UmlModels.UmlModel
Bookmarks	Scripts.UmlModels.Tags
MyModels	Scripts.UmlModels.Tags
signOut	Scripts.UmlModels.Users
profile	Scripts.UmlModels.Users
ModelNavPanel	Scripts.UmlModels.Navigation
Model	Scripts.UmlModels.Site.Master
Defaults	Scripts.UmlModels.Site.Master
Registered	Scripts.UmlModels.Site.Master
Site.Master	Scripts.UmlModels.Site.Master

Table 102: Detailed Design: Presentation Logic Tier: Components from the Uml package related with components in the Scripts package. Client Role.

Common to all the views, the following components are sent to the *Presentation GUI Tier*:

Component	Package
ext-base	ext-2.2
ext-all	ext-2.2
en	UmlModels.Lang
common	UmlModels

Table 103: Detailed Design: Presentation Logic Tier: Components sent to the Presentation GUI Tier.

8.7.1.2.2 Scripts

This package contains all the JavaScripts components used to build the pages in the *Presentation GUI Tier*.

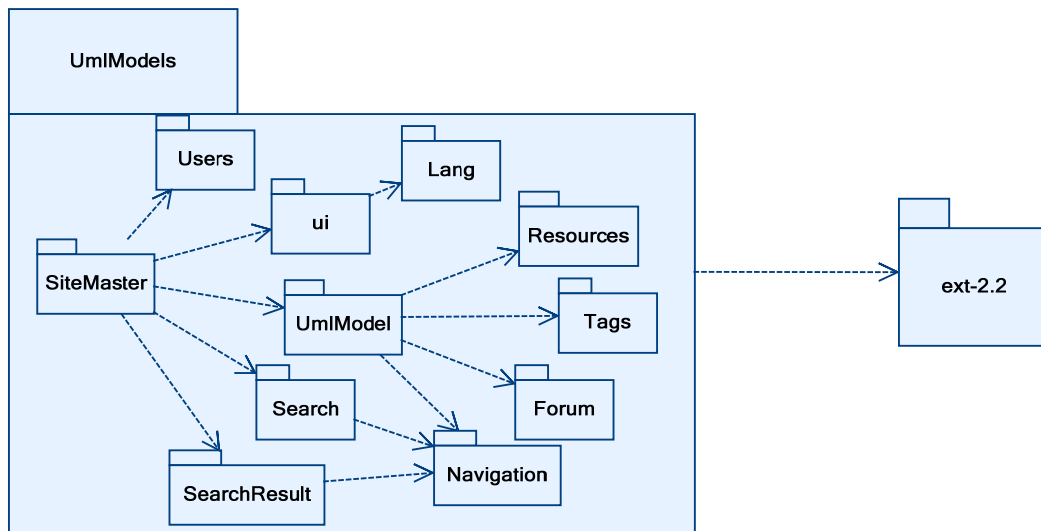


Illustration 97: Detailed Design: Scripts Class diagram.

8.7.1.2.2.1 SiteMaster

The SiteMaster package is the access point from which all the components are started and called. I would be something similar than the main class in C # or Java. All the classes are shown in the next diagram:

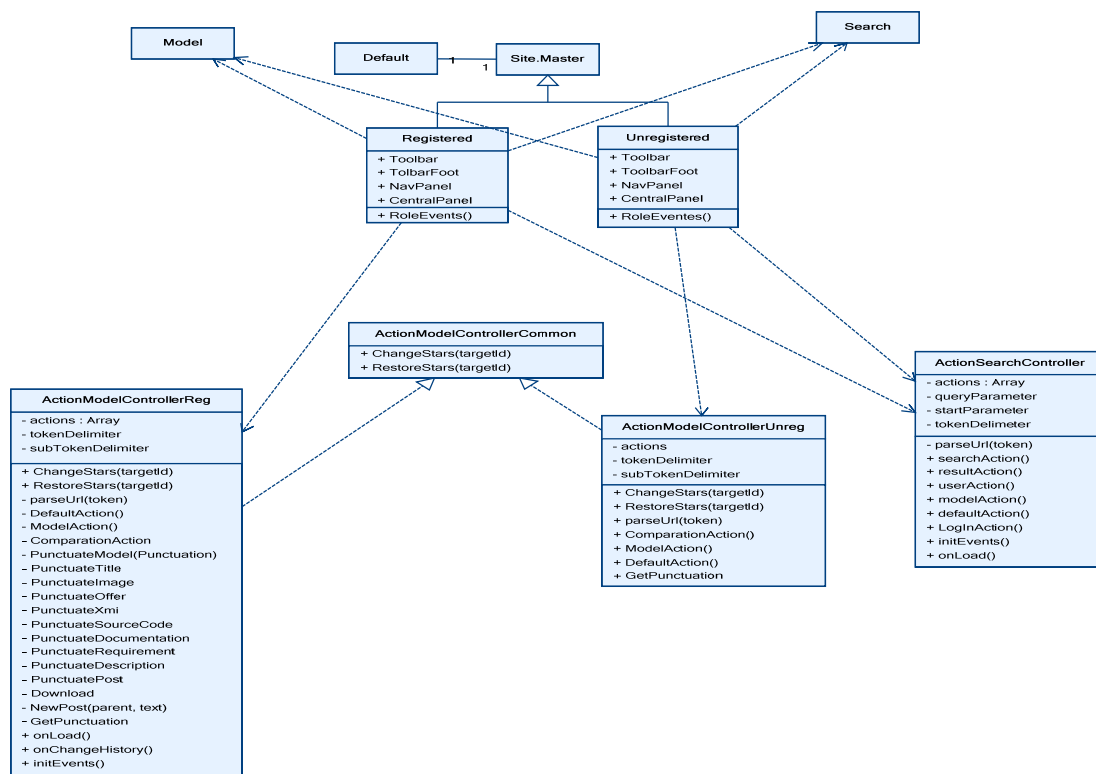


Illustration 98: Detailed Design: Scripts: Site.Master Package.

According with the component loaded by the View (as it has been explained depends on the role of the user and on the page) they will be loaded the following attributes to the `DataGridView`⁶ that manages the layout of the view:

- Toolbar: `Site.Master.items.Toolbar`.
- Foot Toolbar: `Site.Master.items.ToolbarFoot`.
- Navigation Panel: `Site.Master.items.NavPanel`.
- Central Panel: `Site.Master.items.CentralPanel`.

Once this components are loaded, some controllers in the client side will start the events that monitors the user's actions. These controllers are shown in the *Illustration 75: Detailed Design: Scripts: Site.Master Package* and explained in the Client Controller Tier section.

8.7.1.2.2.2 Users

All the classes are shown in the next diagram and following explained:

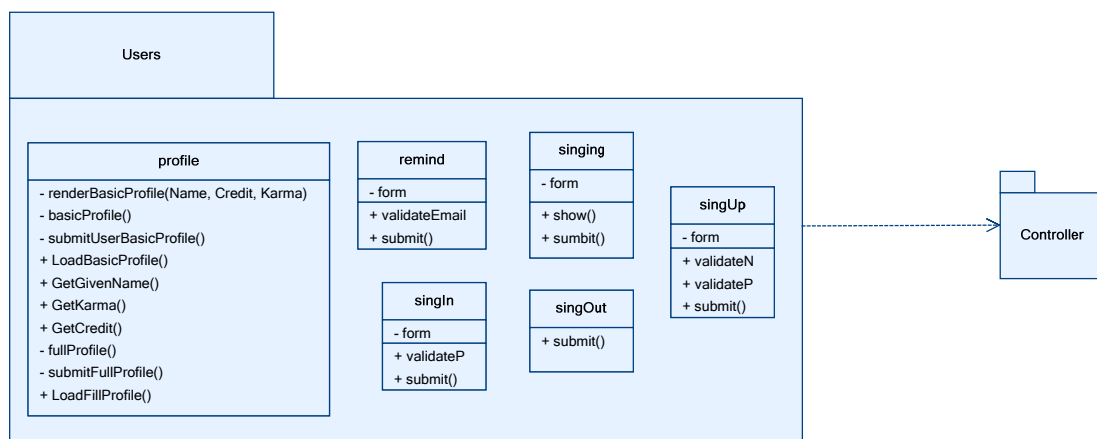


Illustration 99: Detailed Design: Scripts: Users Package.

In this package they are performed actions related to the registered users such as sign in, sign out profile or related became a registered user such as sing up...

8.7.1.2.2.3 UI

See *Presentation GUI Tier*.

8.7.1.2.2.4 UmlModel

This package is in charge of showing the Model pages with its components. Following is presented the classes that this package has with its functionality:

⁶ More information about what a `DataGridView` is and how is used can be found in the *Presentation GUI Tier* section from the *Client* package.

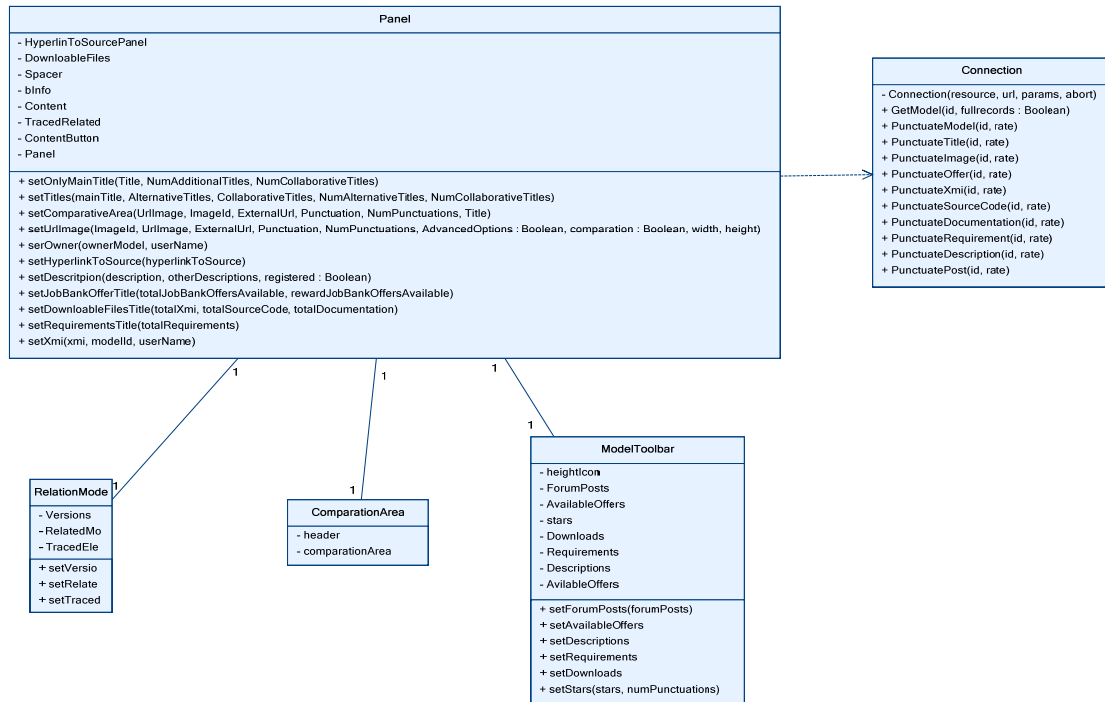


Illustration 100: Detailed Design: Scripts: UmlModel Package.

8.7.1.2.2.5 Resources

This package is in charge of rendering the data received from the server into the correct place according to the role of the user.

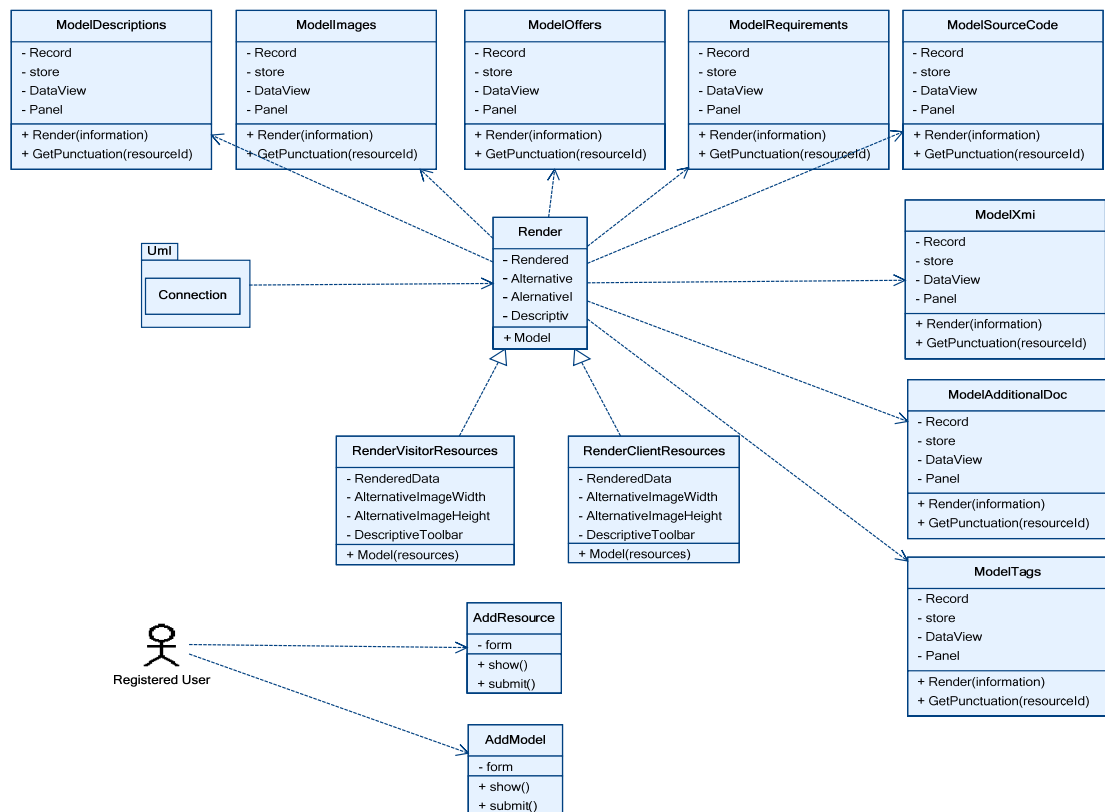


Illustration 101: Detailed Design: Scripts: Resources Package.

8.7.1.2.2.5.1 *RenderVisitorResources*

The `Resources.Render.Model (Resources)` is called when the data in JSON of a model is received from the server by the `Connection` class in the `UmlModel` package. Once is received, the follow actions are taken if there is information in these sections:

- `UmlModel.Panel.SetTitles`
- `UmlModel.ModelToolbar.ForumPosts.SetForumPosts`
- `UmlModel.ModelToolbar.AvalibableOffers.SetAvalibableOffers`
- `UmlModel.ModelToolbar.Requirements.SetRequirements`
- `UmlModel.ModelToolbar.Descriptions.SetDescriptions`
- `UmlModel.ModelToolbar.Downloads.SetDownloads`
- `UmlModel.Panel.SetOwner`
- Find the main image and:
 - o `UmlModel.Panel.SetUrlImage`
- `UmlModel.Panel.SetUrlImage`
- `UmlModel.Panel.SetDescription`
- `UmlModel.Panel.SetHyperlinkToSource`
- `UmlModel.Panel.SetJobBankOfferTitle`
- `Resources.ModelOffers.Render`
- `UmlModel.Panel.SetDownloadableFilesTitle`
 - o Hide Section if no information: `UmlModel.DownloadableFiles.hide()`
- `Resources.ModelXmi.Render`
- `Resources.ModelSourceCode.Render`
- `Resources.ModelAdditionalDoc.Render`
- `UmlModel.Panel.SetRequirementTitle`
- `Resources.Requirements.Render`
- `UmlModel.ModelToolbar.Stars.SetStars`
- Init the events associated with the rendered information
 - o `Site.Master.ActionController.initEvents`

8.7.1.2.2.5.2 *RenderClientResources*

The `Resources.Render.Model(Resources)` is called when the data in JSON of a model is received from the server by the `Connection` class in the `UmlModel` package. Once is received, the follow actions are taken if there is information in these sections:

- `UmlModel.Panel.SetTitles`
- `UmlModel.ModelToolbar.ForumPosts.SetForumPosts`
- `UmlModel.ModelToolbar.AvalibableOffers.SetAvalibableOffers`
- `UmlModel.ModelToolbar.Requirements.SetRequirements`
- `UmlModel.ModelToolbar.Descriptions.SetDescriptions`
- `UmlModel.ModelToolbar.Downloads.SetDownloads`
- `UmlModel.Panel.SetOwner`
- Find the main image and:
 - o `UmlModel.Panel.SetUrlImage`
- `UmlModel.Panel.SetUrlImage`
- `UmlModel.RelationModels.Versions.setVersions`

- UmlModel.RelationModels.TracedElements.setTracedElements
- UmlModel.RelationModels.RelatedModels.setRelatedModels
- UmlModel.Panel.SetDescription
- Resources.ModelDescriptions.Render
- UmlModel.Panel.SetJobBankOfferTitle
- UmlModel.Panel.SetDownloadableFilesTitle
 - o Hide Section if no information: UmlModel.DownloadableFiles.hide()
- Resources.ModelXmi.Render
- Resources.ModelSourceCode.Render
- Resources.ModelAdditionalDoc.Render
- UmlModel.Panel.SetRequirementTitle
- Resources.Requirements.Render
- UmlModel.ModelToolbar.Stars.SetStars
- Init the events associated with the rendered information
 - o Site.Master.ActionController.initEvents

8.7.1.2.2.6 Search

This package is in charge of showing the search page with its components as well as performs the query when a user is searching. Following is presented the classes that this package has with its functionality:

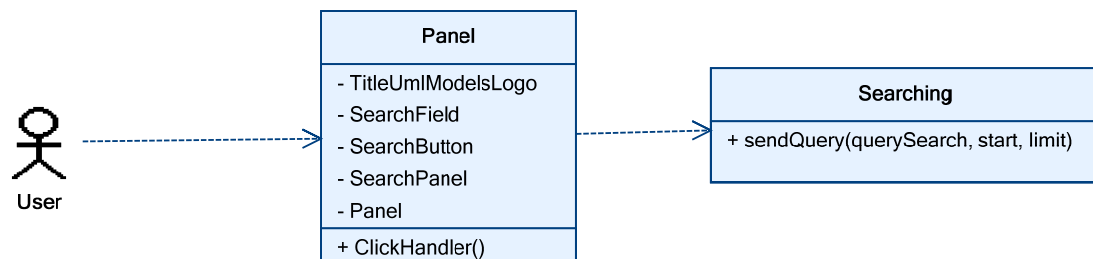


Illustration 102: Detailed Design: Scripts: Search Package.

8.7.1.2.2.7 SearchResult

This package is in charge of storing the search results performed as well as giving support to the rendering process of all the records providing an intelligent functionality for truncating the descriptions according to the width of the window and the level of detail (with and height of the image). Following is presented the classes that this package has with its functionality:

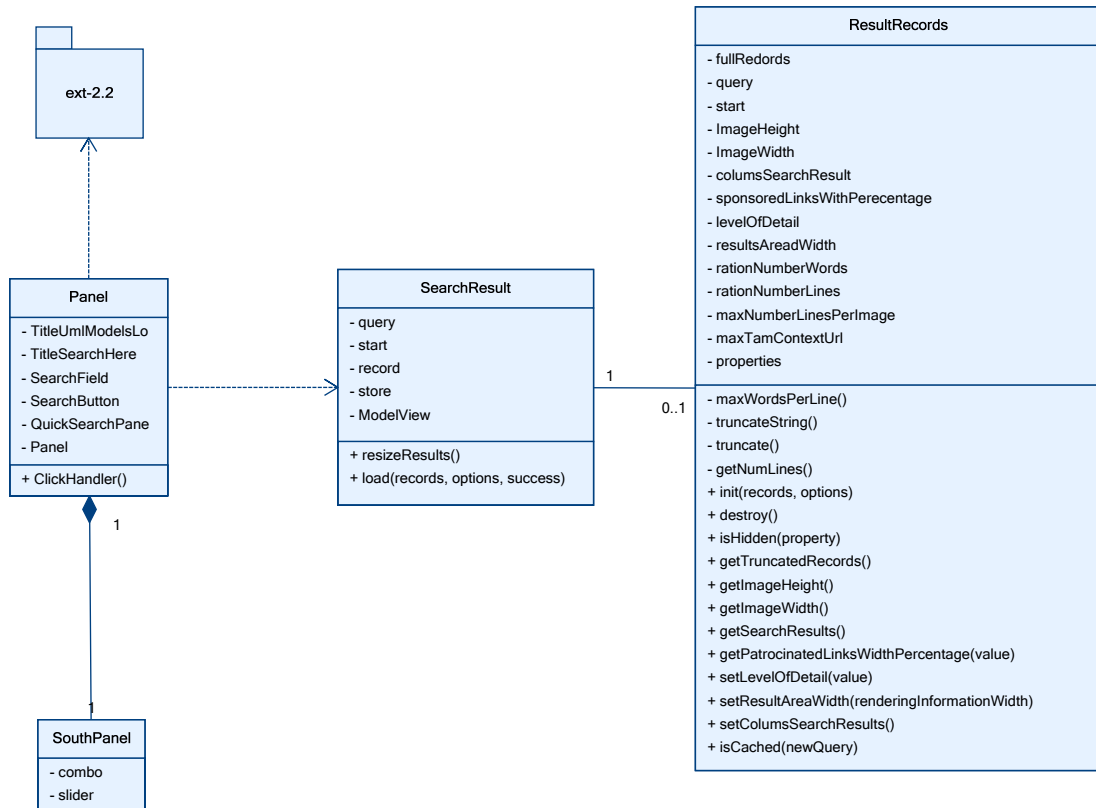


Illustration 103: Detailed Design: Scripts: Search Result Package.

The load method in *SearchResult* is a call-back function when *SearchResult.store.load* finishes the request it is performed this actions:

- `SearchResult.ResultRecords.init(records, options);`
- `SearchResult.resizeResults()`
 - o `SearchResult.ResultRecords.setLevelOfDetail(SearchResult.SouthPanel.slider.value)`
 - o `SearchResult.ModelView.collectData(records)`
 - o `SearchResult.ModelView.refresh()`

8.7.1.2.2.8 *ActionSearchController*

See *ActionSearchController* in the *Client* section.

8.7.1.2.2.9 *ActionModelControllerUnreg*

See *AntionModelControllerUnreg* in the *Client* section.

8.7.1.2.2.10 *ActionModelControllerReg*

See *ActionModelControllerReg* in the *Client* section.

8.7.1.2.2.11 Navigation

This package contains the navigation panel which is shown to the registered users.

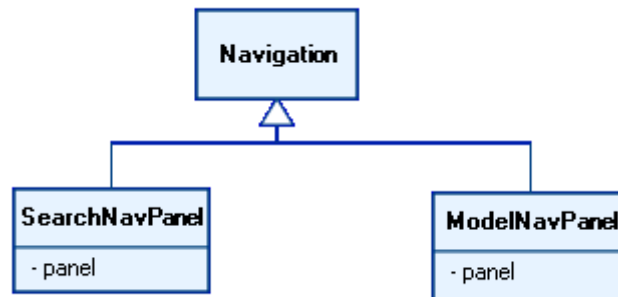


Illustration 104: Detailed Design: Scripts: Navigation Package.

8.7.1.2.2.12 Lang

This package only contains a class called **en.js** which contains the UI contents showed to the user in English language. In the future, if you want to load the GUI in other language, the only requirements needed is to translate the properties stored in this class.

8.7.1.2.2.13 Tags

This package is in charge of connecting the Client to the Server to get the first/next level of the tree of information that is contained in the Bookmarks and the Resources of a user.

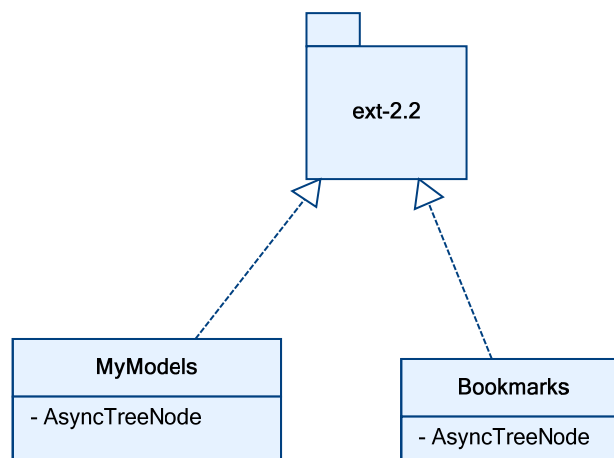


Illustration 105: Detailed Design: Scripts: Tags Package.

8.7.1.3 Business Tier

The business tier associated with the GUI is divided into three packages:

- Storage: Actions related to the physical access to the files of the downloadable resources as well as file support *utils*.
- DownloadAndCharging: Actions related to the checking of the balance and the charge of the download. If it is O.K. it calls use the Storage package to return the file.
- Security: Random number generation for users sign up verification link, email communication...

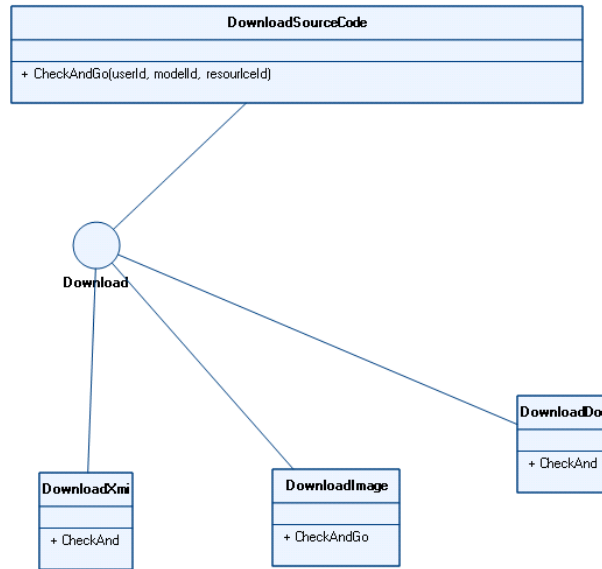
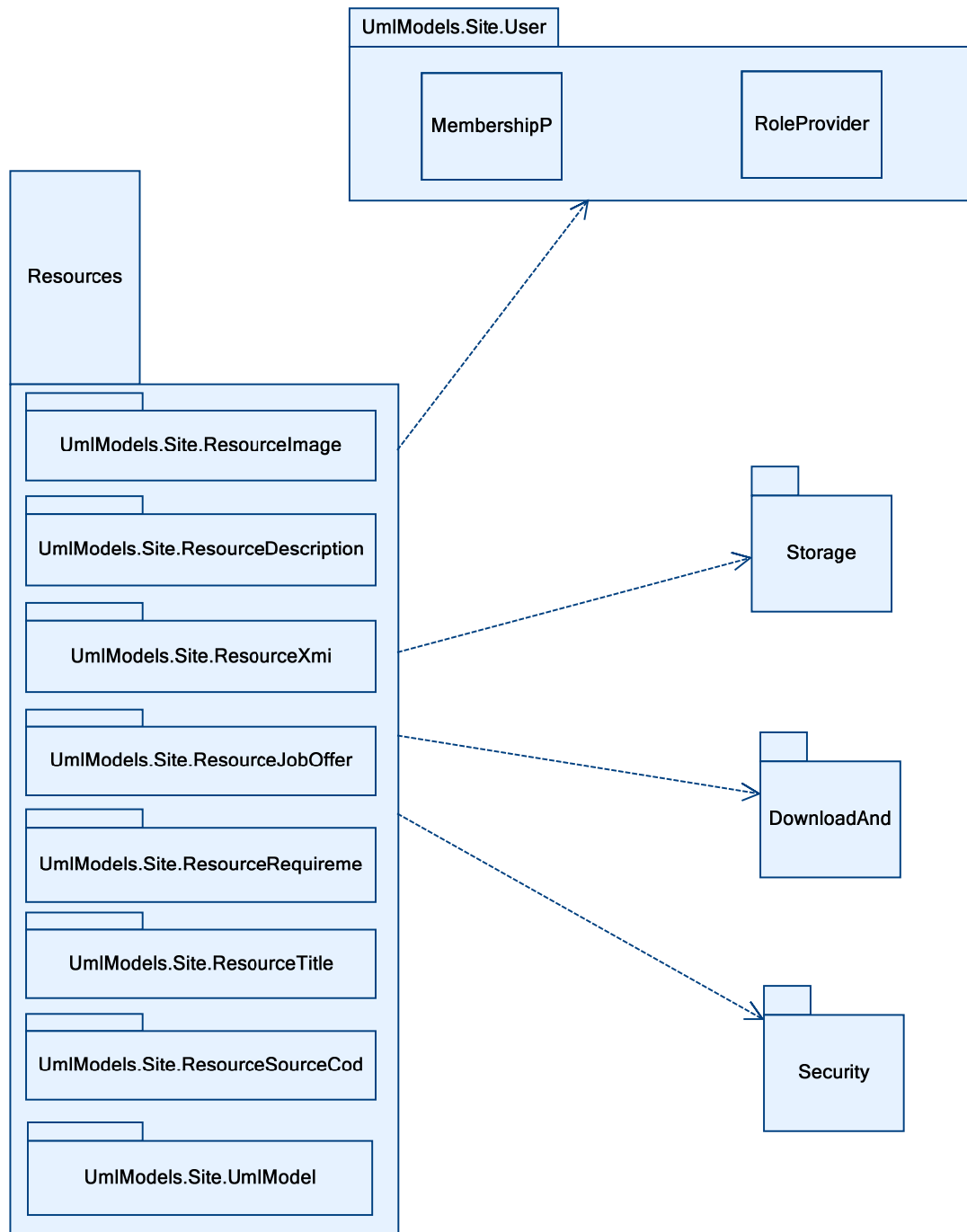


Illustration 106: Detailed Design: Business: Classes Downloadable Package.

Due a better understanding of the implementation, simplicity on the maintenance of the code and a facility to extend the number of resources and its functionalities, it has been implemented each resource in a separate solution (DLL):



We are building a web application in which the users ask for certain information from the database and the behaviour instigated by these actions does not rely on a particular state. The users ask for some information and the server provides it without storing any object instance even though if the users have the browser open for many hours. Because of this, static methods should be used. Actually, the only state that is needed is the session of the user and this is provided by the IIS. For this reason, there are no instances created for the interfaces defined.

8.7.1.4 Data Access Tier

In the Data Access Tier we have a middleware layer that permit as to isolate from the Data Access. This layer is compound by the ADO.NET Entity Framework among some APIs defined in the follow approaches.

In this tier, the follow component interfaces have been defined:

- IResource.
- IVersionable.
- IPunctuable.
- IAuthorizeDownload.

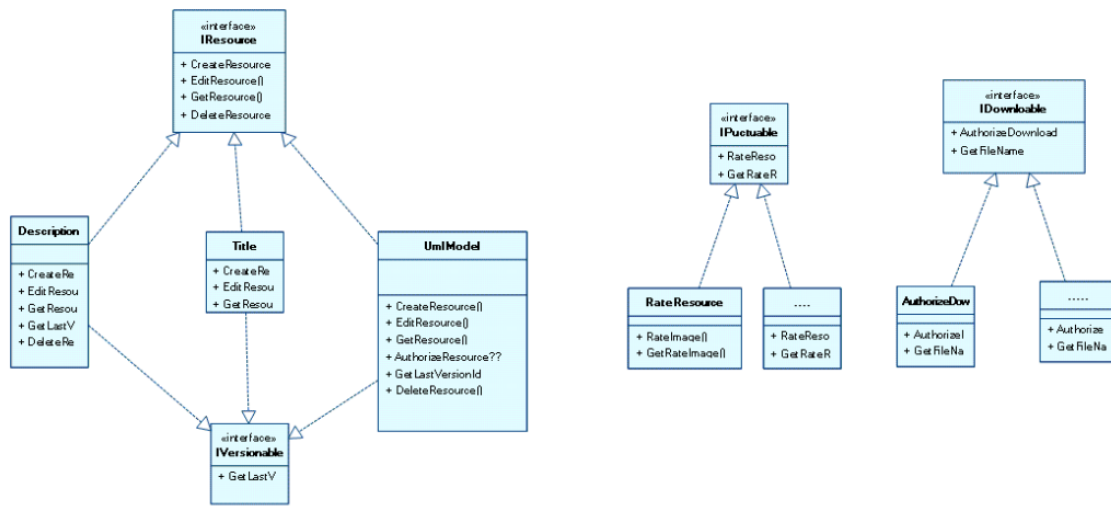


Illustration 107: Detailed Design: Data Access class diagram.

In the diagram is represented the Title resource. Title resource is a versionable resource so implements the IVersionable and IResource interface. Other versionable resources are:

- Description.
- Image.
- XMI.
- Additional Documentation.
- Source Code.
- Requirement.

All the downloadable elements will implement IAuthorizeDownload which controls if a user has authorization to download a resource. The administrators always have authorization unless a model is private. The clients will be authorized to download a resource if they have balance enough and the model is public. The downloadable elements are:

- XMI.
- Additional Documentation.
- Source Code.

In this approach (unlike the other one proposed in *(Minutes of the meeting.2009.02.09.Arquitectura 7 Diseño específico de las 3 capas.wmv.docx see Reports of*

Meetings), we treat each resource as a separate DLL. For that, a new DLL has to be created for the inclusion of a new resource.

With this approach, we will have more classes but the design will be more robust.

8.7.2 Client

In this section is described all the packages within the Client: **Presentation** and **Controller**.

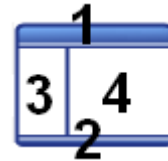
8.7.2.1 Presentation GUI Tier

Two options were presented to the client for rendering the user interface in the client. The first one consists on **generating the page in the server** and giving it (already built) in an HTML to the browser of the client. The other option is to send the JS application (when the user starts the connection for the first time) which contains the structure of the page, get the content of the information to be show with an AJAX request and **built the page in the client**.

Due the efficiency reasons, the second one was selected by the client despite this approach might not fulfil in an efficient way the requirement of been indexed by the searchers engine such as Google⁷.

For this, the structure of the page is sent to the user. The structure of the page is built using the **EXT ViewPort** component which provides a mechanism for rendering itself to the document body, and automatically sizes itself to the size of the browser viewport and manages window resizing. The *ViewPort* is configured with a 'border' layout⁸ and will render the information received into the following sections added to the data view and defined under the *Site.Master package* (In the '*Presentation Logic Tier*' is explained on detail the components that manage each area):

- Toolbar (1): `Site.Master.items.Toolbar`.
- Foot Toolbar (2): `Site.Master.items.ToolbarFoot`.
- Navigation Panel (3): `Site.Master.items.NavPanel`.
- Central Panel (4): `Site.Master.items.CentralPanel`.



Each of these components will render information using one of these components:

- **DataView**: A mechanism for displaying data using custom layout templates and formatting. **DataView** uses an `Ext.XTemplate` as its internal templating mechanism, and is bound to an `Ext.data.Store` so that as the data in the store changes the view is automatically updated to reflect the changes.
- **XTemplates**: Is a template class that supports advanced functionality like autofilling arrays, conditional processing with basic comparison operators, sub-templates, basic math function support, special built-in template variables, inline code execution and more.
- **TreePanel**: The **TreePanel** provides tree-structured UI representation of tree-structured data. **TreeNode**s added to the **TreePanel** may each contain metadata used by your application in their `attributes` property.

⁷ This is done by adding the information of the model in the no-script label and it is sent two times, one there and another one in the AJAX request. For more details see the implementation details in the `UmlController` on the follow section: **Controller Tier**.

⁸ This is a multi-pane, application-oriented UI layout style that supports multiple nested panels, automatic split bars between regions and built-in expanding and collapsing of regions

- PagingToolBar: A specialized toolbar that is bound to a Ext.data.Store and provides automatic paging control. This Component loads blocks of data into the Store passing parameters whose names are specified by the store's paramNames property.
- Slider: This supports vertical or horizontal orientation, keyboard adjustments, configurable snapping, axis clicking and animation. Can be added as an item to any container.
- Panel: Panel is a container that has specific functionality and structural components that make it the perfect building block for application-oriented user interfaces. The Panel contains bottom and top toolbars, along with separate header, footer and body sections. It also provides built-in expandable and collapsible behaviour, along with a variety of prebuilt tool buttons that can be wired up to provide other customized behaviour. Panels can be easily dropped into any Container or layout, and the layout and rendering pipeline is completely managed by the framework.
- Toolbar: Specific panel as a tool bar.
- GridPanel: Specific panel as a grid.
- FormPanel: Specific panel based on a form.
- TextField: Basic text field.
- HtmlEditor: Provides a lightweight HTML Editor component. Some toolbar features are not supported by Safari and will be automatically hidden when needed. These are noted in the config options where appropriate.

Additionally to the EXT components used for present the information to the user, some others were created or just extended from other EXT components. In the next illustration are presented those elements:

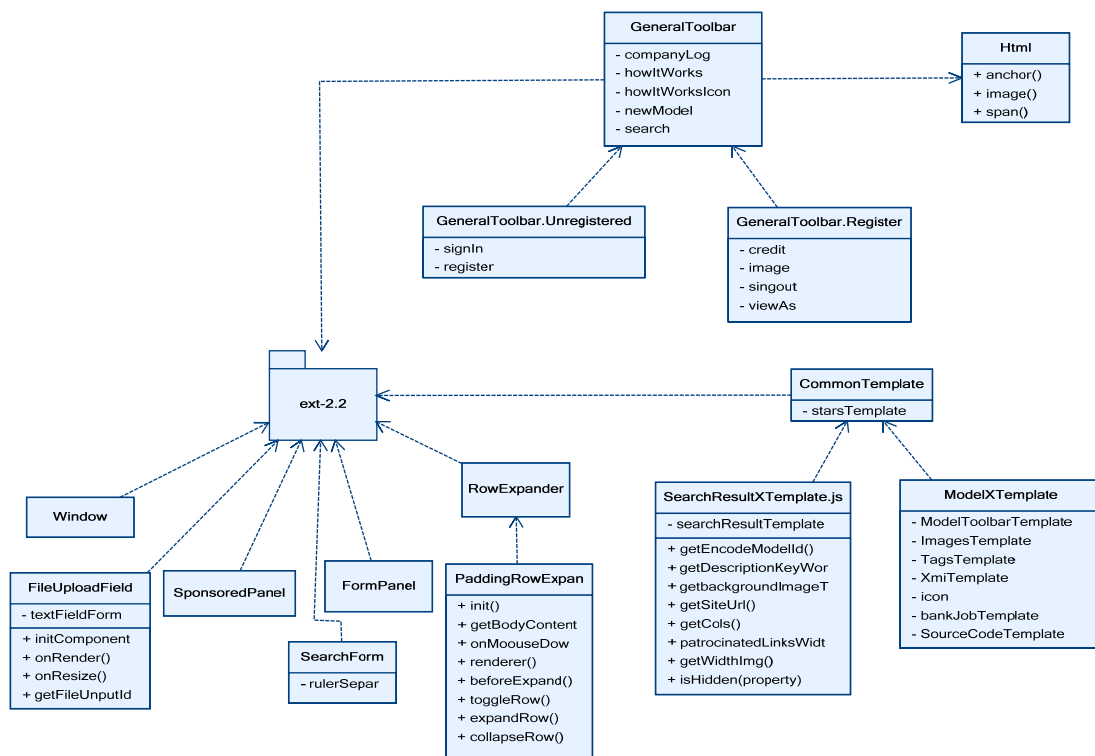


Illustration 108: Detailed Design: Client: Presentation GUI Tier.

8.7.2.2 Controller Tier

As it has been shown in the *Illustration 75: Detailed Design: Scripts: Site.Master Package* and explained in the *Presentation Logic Tier* section, once they are loaded the views some controllers in the client side will start the events that monitor the user's action.

8.7.2.2.1 ActionModelControllerReg

Controls the actions performed by the registered users and map them with the corresponding calls through the net to the server.

On load the model's page, it called to the ActionModelController which performs the following actions:

- **UserAction:** Ajax Request: Users.basicProfile.LoadBasicProfile();
- **parseUrl:** Takes the URL from the document and extracts the parameters needed.
- **ModelAction:** Ajax Request: UmlModel.Connection.GetModel(modelId, true);

Additionally they are initialized the events in charge of triggering the next actions when the user interacts with the UI:

- *ComparisonAction* -> UmlModel.Panel.SetComparativeArea(urlToCompare);
- *PunctuateModel* (Punctuation) -> UmlModel.Connection.PunctuateModel(modelId, Punctuation);
- *PunctuateTitle* (ResourceId, Punctuation) -> UmlModel.Connection.PunctuateTitle (ResourceId, Punctuation);
- *PunctuateImage*, *PunctuateOffer*, *PunctuateXmi*, *PunctuateSourceCode*, *PunctuateDocumentation*, *PunctuateRequirement*, *PunctuateDescription*, *PunctuatePost*.
- *Download* (resourceType, ResourceRowId, ResourceId, VersionId, FileName):
 - o Builds the URL that calls to the Download Controller.
 - o Calls to the URL
 - o Shows a confirmation message if is chargeable.
 - o Updates the balance in the GUI. AJAX call:
 - Users.basicProfile.LoadBasicProfile();
- *NewPost*(parent) -> UmlModel.Connection.NewPost(parent);
- *GetPunctuation* (ResourceType, ResourceId)

8.7.2.2.2 ActionModelControllerUnreg

Controls the actions performed by the unregistered users and map them with the corresponding calls through the net to the server. The functionality is similar than the ones for the registered user, but the ones that they are exclusive for registered user is performed the login action.

8.7.2.2.3 ActionSearchController

Controls the actions performed by the registered users and map them with the corresponding calls through the net to the server.

On load the model's page, it called to the ActionModelController which performs the following actions:

- **UserAction:** Ajax Request: `Users.basicProfile.LoadBasicProfile();`
- **parseUrl:** Takes the URL from the document and extracts the parameters needed.
- Depending on the Action of the URL performs one of the following actions:
 - **SearchAction:**
 - Shows the search panel.
 - Hides the search result panel.
 - Hide the toolbarFoot.
 - Synchronize the View Port.
 - **ResultAction:**
 - Shows the toolbar.
 - Synchronize the View Port.
 - If the query is not cached: *Searching.SendQuery(q, start, limit).*
 - Hide the search panel.
 - Shows the search result panel.
 - **ModelAction:**
 - Send statics of use.
 - Delete the store records.
 - Forward to the model's page.
 - **LogInAction:**
 - Shows the search panel.
 - Hide the search result panel.
 - Shows the sing in panel.
 - **DefaultAction** in non argument:
 - SearchAction.

8.7.3 Data Base

As it has been designed in the conceptual model, one user can have many resources and a model is compound of many resources. These two relations have opened a discussion when designing the implementing model; confronting two mayor principles: performance vs. design. In practice, a few approaches we have to study around these two possible solutions:

- ➔ One resource per table.
- ➔ One table for all the resources.

In theory, the databases are better prepared to support more data lengthwise than widthwise. This mean, that it is possible to get a higher performance in a database with many rows in one table than a few in many tables. When you perform a select, the database makes several actions such as planning of the command, reservation of memory, and so on. Therefore one select against a table with ten times more rows is, generally, faster than ten selects against ten tables getting one row.

The problem of having one table per resource is that you have to access to many tables to obtain a model. On the other hand, the problem of having one table for all the resources is that the resources will have to be able to be null with the consequent risk further explained.

In shorts, when we are modelling (the conceptual model) the concepts are ordered and represented hierarchically; while when we are modelling the database only the needed elements are taken.

Due to performance reasons, at the beginning we thought about having a table per resource and replicate the resource data in a master table called model. In order to obtain the coincidences of a certain search, it would be necessary only to access to one table (master table). And time-to-time a background process will update the modified elements. In the case of the model's page, the idea would be to have in the master table the first N elements of each resource in order to show them at the first attempt to load the model and then when the user wants to see the next ones, the access to the follow elements will go against each resource table (no joins are needed due we already have the id of the model at that very moment).

As one can see, the design of the GUI and the implementation of the database have to be intertwined. Other way of representing the GUI that entails another approach to design the database is splitting each of the three conceptual levels of information into different tabs.

For this reason, we have evaluated the impact of the GUI in the design of the database as well as which semantic controls of the associations of the implementation model are that important that we need to assure. Four different approaches according with the possible designs of the GUI were evaluated: **Unified Tables**, **Separated Tables**, **Intermediate approaches** and **GUI - Data Base Dependence. Tabs & Levels Approach**. Following are presented the results of the comparison realized and subsequently are detailed these four approaches:

Performance Test			
	Ticks	Milliseconds ⁹	Description
Unified Table	24013282	0,2	One big master table.
Intermediate Approach	36119356	0,3	One table per resource and the id on the Model's table: Join
Separated Tables	100287030	1	One table per resource without the id on the Model's table: A group of selects against the resources.
	112849626	1,1	One table per resource without the id on the Model's table: A group of joins against the resources.

Table 104: Data Base: Performance Test. Comparison Different Approaches

The tests were done comparing the performance of the best three approaches. The tests consist of 150 threads performing 10000 each over a database with 125000 models and 6 tables for the resources and one table for the Model.

As we can see, there is a big difference between the unified table and the separated tables approach. It is obvious that either of the separated tables are not a good option. Between the Unified table and the intermediate approach there is only 0.1 milliseconds of difference (which is almost the same) and considering the benefits described of the Intermediate approach over the unified approach, we have decided to use the intermediate approach.

⁹ On tick of the clock is 100 nanoseconds

8.7.3.1 Unified Tables Approach

In the way of having one table for all the resources, it is proposed this approach which will have to be evaluated:

- ➔ One table for Model.
- ➔ And another table for all the Resources.

An advantage that this approach has is that it won't be necessary to create a new table when a new resource type will appear. On the other hand, the main problem that we are going to find is that these tables are going to be extremely huge and very difficult to handle in terms of data integrity. Many relations between resources have to be manually checked: some resources are chargeable, others are versionable.... As well, many resources have different attributes which sometimes can be null and some others not.

Additionally, it can be worthy to gather some the versions and the punctuations.

- ➔ One table for all the Versions.
- ➔ One table for all the Punctuations.

Also will have:

- ➔ One table for Sponsored Links.
- ➔ One table for Job Offers.

The unified approach has been discussed earlier (see the introduction in *Data Base* section). The problem of this approach is the scalability. For instance, on the assumption that we would have 50000 models in our database having the follow rows for each model:

- 1 row for the model.
- 1 row for each image in the model.
- 1 row for each XMI in the model.
- 1 row for each title in the model.
- 1 row for each description in the model.
- ... *and so on* ...

We could have a minimum of 5 rows peer each model; this easily could takes as to have a table with 250000. In the moment in which the collaboration area became to be used, the estimations can be of millions of tables.

Still the client thinks it is worthy to have all this in one table despite of we are going to have a huge table if we can save some joins every time we load a model.

The way in which we are designing the GUI (with the use of AJAX) give as the possibility to show a few resources (typically the main ones) instead of all at the same time. Hiding some of them, if we just replicate the mains ones on the Model table, we won't need to access to any

table but one for loading the model. And if the user wants to bring more resources for this model, no joins will be needed since we already know which model is it (its primary key).

Still if we assume that we are going to have the id of the model that we are trying to obtain, then always a user retrieve a model then no joins are needed either. We just need one select per resource.

Since the moment in which we are not going to overhead the database with joins for retrieving the resources, this approach might be rejected depending on the difference between one or many (between three and five) selects (one per resource table).

8.7.3.2 Separated Tables Approach

This approach defends the use of the database mechanisms to provide the integrity of the relations and of the null elements but with the replication of the main data in the VRUmlModel table.

Despite the performance of the results can be worst than making a single select, the benefits of having the resources in separated tables can be positive for the management of the database.

Within this approach we can find the follow tables:

- ➔ One table for VRUmlModel.
- ➔ One table for User.
- ➔ One table for each versionable resource:
 - VRDescription.
 - VRImage.
 - VRTitle.
 - VRXmi.
 - VRSourceCodePackage.
 - VRRequirement.
 - VRAdditionalDoc.

- ➔ One table for each non versionable resource:
 - NVPost.
 - NVTag.
 - NVJobOffer.

- ➔ One table for all the Versions.
- ➔ One table for each Donwloable/Punctuable resource.
- ➔ One table for Sponsored Links.
- ➔ One table for Job Offers.

See Illustration in (Separated Tables Approach: *Minutes of the meeting.2009.01.19.Modelo de Implementacion II.wmv.docx* see Reports of Meetings section).

8.7.3.3 Intermediate Approach

In this approach, resources without any relation have been unified into the same table:

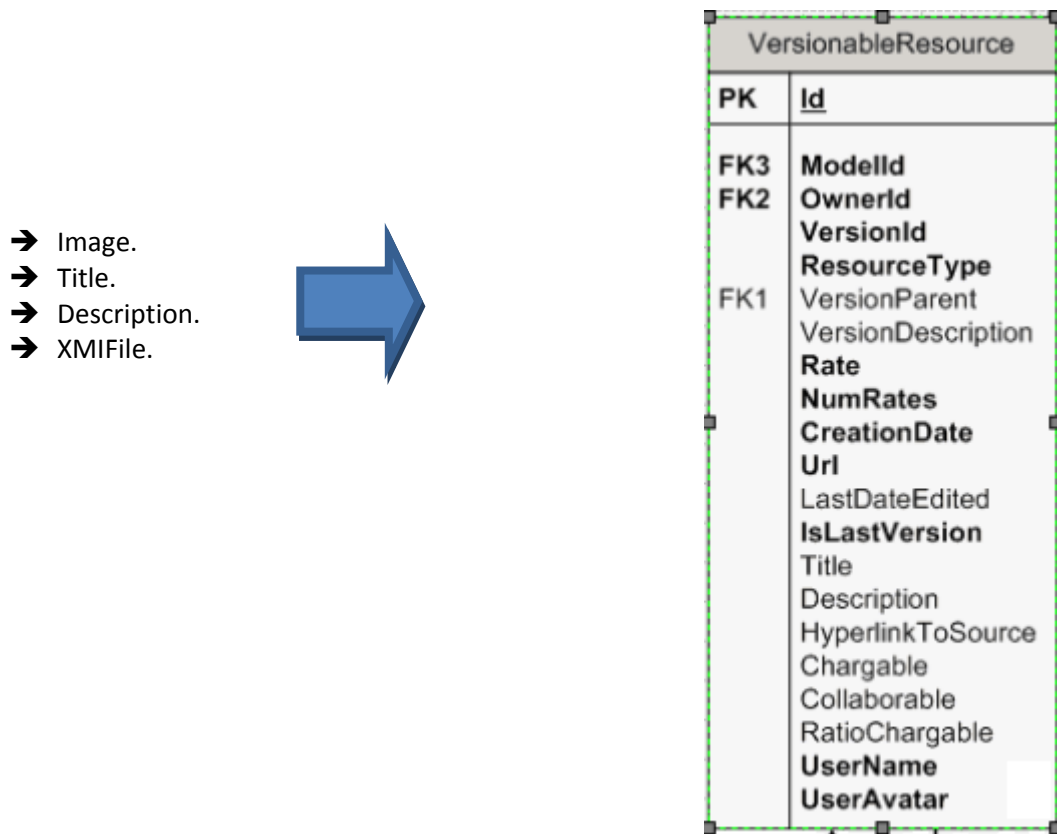


Illustration 109: VersionableResource: Image, Title, Description and XMI

As we can see in (Illustration 109) the table *VersionableResource* has a title, a description and a *HyperlinkToSource*. The last attribute will be used either for the Image and the XMI resource; but it will be null when the row is an Image or a Title resource.

The *Chargeable* and *RatioChargeable* represents the price of a download and the proportion of money that goes for UmlModels in each download of an image and an XMI. This was proposed with the aim to give the possibility to UmlModels to offer promotions and it has been rejected by the client.

The other resources that had relations between other elements of the model have been included into different tables. These resources are:

- VRSourceCodePackage.
- VRAdditionalDoc.
- VRRequirement.

Non versionable resources also have been included in different tables. This are:

- NVRPost.
- NVRTag.
- NVRJobOffer.

The master table which includes the replicated elements is called:

➔ VRUmIModel.

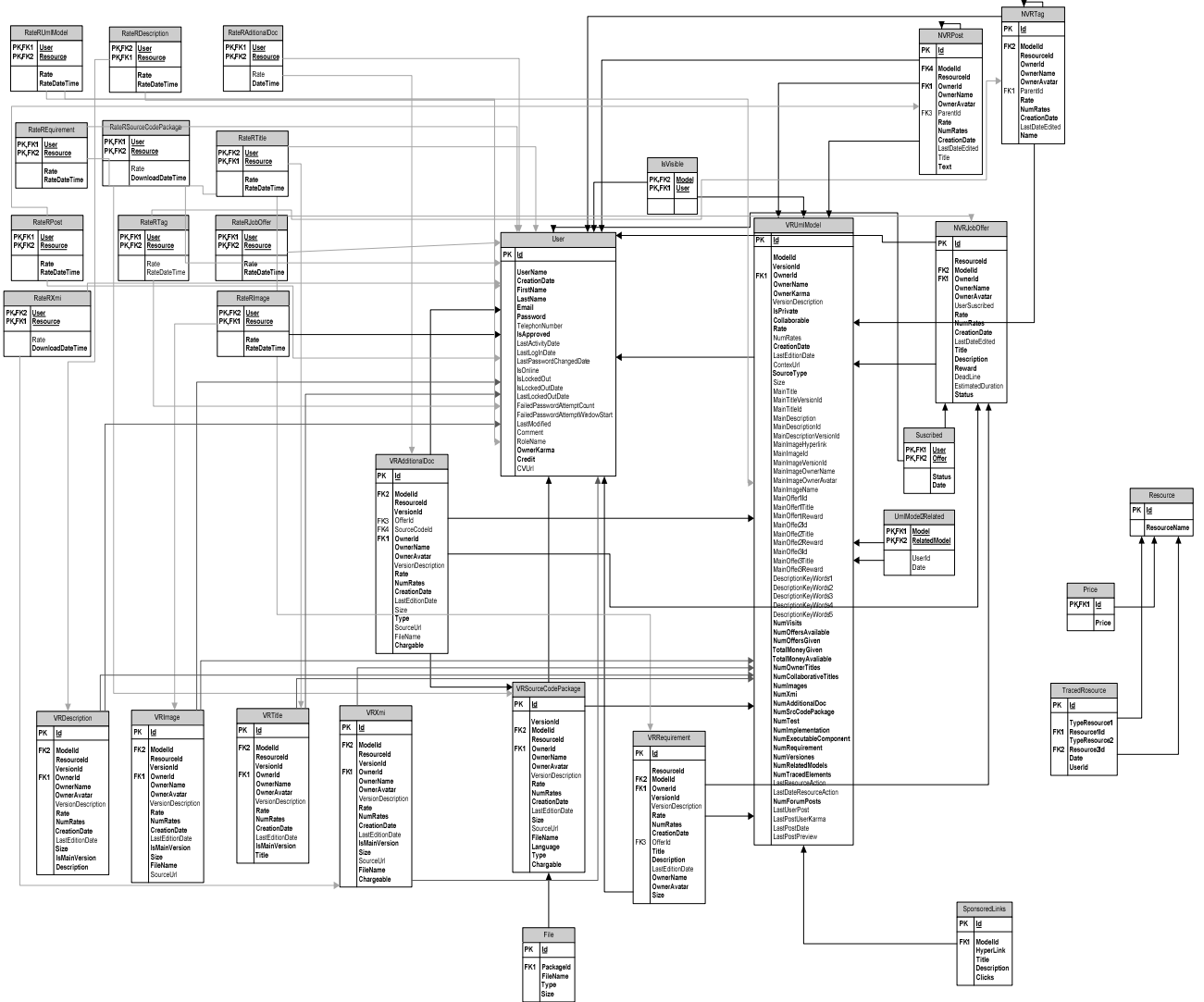


Illustration 110: Design of the Data Base: 'DiseñoBaseDatos3.4.vsd'

8.7.3.4 GUI - Database Dependence. Tabs & Levels Approach.

Another approach is to group into the levels of information defined: Level 1 (Information of the model), Level 2 (Collaborative zone) and Level 3 (Commercial zone).

The main problem with this approach is that we are going to have two different tables for the same information (one for the images of the level 1 and other for the ones in level 2).

For this reason, is not going to be discarded.

9 Security Issues

In this section we focus on cover a study the most important elements of the computer security applied to UmlModels. The objective of computer security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users [56]. These objectives can be sum up in these three general goals of protecting the **confidentiality**, **integrity** and **availability** of the information.

Confidentiality is grantee ensuring that information is accessible only to those authorized to have access. For this, UmlModels is requiring authentication for all the private areas of the Site. This authentication will be done with a password only known by the user; even so, the password is not even stored in the database, instead of it is stored a hash of the password.

Additionally, a site like UmlModels should guarantee the confidentiality during the communication process. For that, always a user is going to either sign in or request private information, this connection is going to be done using SSL.

In the moment that we are storing private information of a user, the Site must be in observance of the **Organic Law 15/1999 on the Protection of Personal Data (LOPD)**. Further on this document is detailed some important issues about this topic.

Since the client specified that the whole page should be done as a JavaScript application, we find the inconvenience that the whole page should be load using SSL, being slightly increased the loading time of the first page.

To prevent unauthorised access to restricted areas, all the methods that they are published to the Internet checks that the user who is requesting the service is authorized by checking if the session is open and the role of the user has enough privileges.

Finally, Integrity during the communication is provided by the transport protocol used by HTTP. In the case in which they are transactions involved in the process such as chargeable downloads, UmlModels guarantees integrity in case of fail performing atomic transactions. For this, if a user pays for a download, the file will be available for this user without paying more even though if the transfer fails.

9.1 Organic Law 15/1999 on the Protection of Personal Data (LOPD)

As it has been agree with the client, this point remains out of this Project; specifically this part was going to be responsibility of Emiliano Fernández Marín from Ciset. Anyway, since it is included in the requirement **2.1-GR-00092: Law of Personal Data Protection**, I decided to include an analysis of the main tasks to do in order to adapt the Site to the law:

- a. Identification of the existing files affected by the LOPD on the Web Site and their specific details.
- b. Registration of the files in the Spanish Data Protection Agency.
- c. Identification and registration of a third party with access to the files.
- d. Privacy Policy.

In order to understand how to proceed, we first have to know what the object of the LOPD is and what is considered Personal Data.

The **LOPD** is a Spanish fundamental law, which aims to guarantee and protect, regarding the treatment of the personal data, the public liberties and the basic rights of the natural persons and specifically their honor, intimacy and privacy both personal and familiar. The main objective is to regulate the treatment of the personal data or files, independently of medium in which they are treated, the right of the citizen over them and the obligations of those who create or treat them.

It is defined as **Personal Data** information any numeric, alphanumeric, graphic, photographic, acoustic or another kind of information capable of being stored, registered, treated or transmitted related to a identified or identifiable physical person.

Following is detailed the steps commented above:

- a. Identification of the existing files affected by the LOPD on the Web Site and their specific details.

The personal data will be store in electronic format. For that, UmlModels will have a dedicated server for the database. The database is a Microsoft SQL Server 2005 called *UmlModels*. It has been identified the following elements susceptible of being Persona Data information in the **dbo.User** table: **User Name, First and Last Name, Email and Mobile Phone.**

The access to the server is restricted to the Administrators, which will be *Julián Urbano Merino*. The access to the database is restricted to the Administrators and to the user of the Web Server.

The purpose and forecast use of this information is to offer a personal service to the users of the Site as well as identify the owners of the models.

The Individuals on which it is sought to obtain this personal data are anyone who connects and register as user to www.umlmodels.com or www.umlmodels.org. This means that the data will come from the users of the Site through a form inside the web site.

Every user should be able to modify or delete their personal information through the web site or sending an email to the administrator: admin@umlmodels.org

- b. Registration of the files in the Spanish Data Protection Agency.

The client must perform the registration on the Spanish Data Protection Agency.

- c. Identification and registration of a third party with access to the files.

It is mandatory to identify and register all those third party companies to whom a part of the files affected by the LOPD such as maintenance enterprise, etc. Ciset and the Reuse Studio Company might have access to the files affected.

d. Privacy Policy.

A Privacy policy must be draw up and showed to the users before the registration.

10 Budget

In this section is detailed the economical cost of the design and develop of the Site. Here is estimated costs of the project including the wages and the fungible and it is also calculated the total amount which the client have to pay.

Costs per Position	
Position	Cost/Hour
Analyst/ Designer	60 €/Hour
Programmer	30 €/Hour

Table 105: Budget. Costs per Position.

Above is detailed the cost per position according to the roles that have been active during this project.

10.1 Personnel

The personnel involved on the development of UmlModels Site are:

- Juan Llorens: Client.
- Julián Urbano Merino: Client.
- Guillermo Suárez de Tangil: Analyst/Designer/Programmer.

In the EXCEL file called 'RecuentoHoras.xls' it is detailed the number of hours per person and per activity done in this project. In the following table is summarized these activities and classified in the roles described in Table 105: Budget. Costs per Position.

Hours per Person and Position		
Position	Name	Total Hours
Client	Juan Llorens	43,3
Client	Julián Urbano	41,25
Analyst	Guillermo Suárez de Tangil	213,5
Designer	Guillermo Suárez de Tangil	184,1
Engineer / Programmer	Guillermo Suárez de Tangil	751,2

Table 106: Budget. Hours per Person and Position

The **total** cost of the personnel associated to this project including the hours of the client as analysts in this project is: **51465 €**.

10.2 Hardware

Following is detailed the costs of the hardware used to develop and deploy the Portal.

Hardware	
Description	Cost
Web Server/Working Machine	670 €
Data Base Server	670 €
TOTAL	1 340 €

Table 107: Budget. Hardware Costs.

10.3 Software

Following is detailed the costs of the software used to develop and deploy the Portal.

Software	
Description	Cost
Pricing for SQL Server 2005 Developer Edition Software	35 €(49 \$)
Pricing for Microsoft Visual Studio 2008 Professional Edition	574 €(799 \$)
Pricing Windows Server 2008	2 x 2 151 €(3 999 \$)
Pricing for Microsoft Office 2007	358 €(499,95 \$)
Pricing for Microsoft Visio 2007	186 €(259.95 \$)
Pricing for Cake Studio	0 €
Pricing for Camtasia Studio 6	214 €(299 \$)
TOTAL	5 669 €

Table 108: Budget. Software Costs.

10.4 Fungible

The costs of the fungible are estimated as follows:

Fungible	
Description	Cost
Office Stationary	35 €
Final Printing and Bookbinding	250 €
TOTAL	285 €

Table 109: Budget. Fungible Costs.

10.5 Indirect Costs

Following are estimated the indirect costs the cost associated to the use of resources like water, electricity... during the development of the project.

In this case, we are going to apply an indirect cost of the 5% of the cost of the project.

Indirect Costs	
Description	Cost
5% Indirect Costs	58 759* 0,05 €
TOTAL	2 937.95 €

Table 110: Budget. Indirect Costs.

10.6 Summary

Following is included the summary of all the costs before including the IVA (taxes).

Summary	
Description	Cost
Personnel	51 465 €
Hardware	1 340 €
Software	5 669 €
Fungible	285 €
Indirect Costs	2 937.95 €
TOTAL	61 696.95 €

Table 111: Budget. Summary.

10.7 Final Budget

Following is presented the final budget of the project elaborated adding the 20% of profit to the costs of the project.

Final Budget	
Description	Cost
Costs	61 696.95 €
20% Profits	12 339.39 €
16% IVA (Taxes)	9 871.512 €
TOTAL	83 907.852 €

Table 112: Budget. Final Budget.

The final budget of the project is **83 907.852** , this is, **EIGHTY-THREE THOUSAND NINE HUNDRED AND SEVEN POINT EIGHT HUNDRED AND FIVETY TOW EUROS.**

11 Conclusions

During this project, a web portal steered towards software community engineers has been built. First, a deeply study of the current technologies needed to build the portal has been fulfilled, selecting the cutting-edge technology in all our milestones: user presentation, architecture deployment and data access.

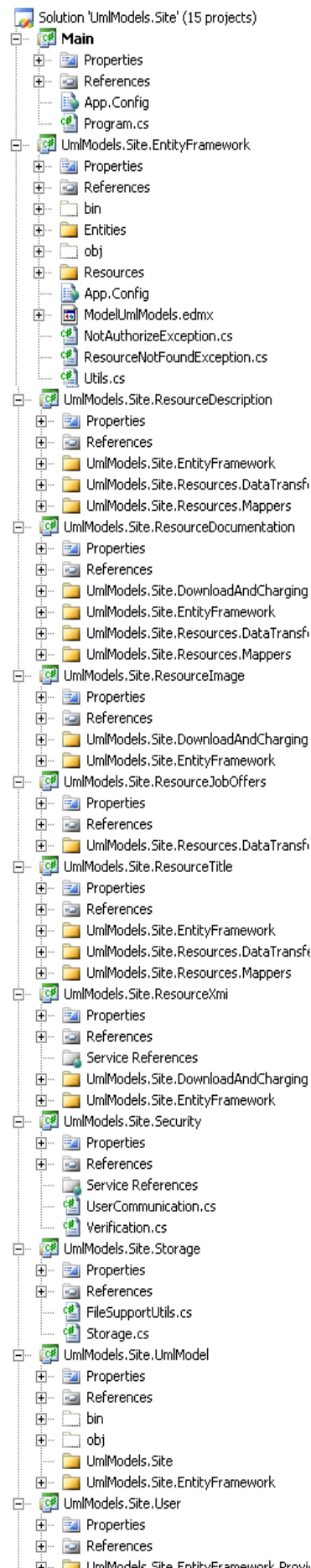
Second, the requirements, in general associated to UmlModels and in particular to the Site, in addition to the economical business model approaches were deeply cogitated.

Next, a graphical user interface has been designed presenting the information in a simple way, showing as much information as possible without overelaborating the graphical result --by introducing different detailed representation levels or using different techniques for handling big structures of information such as collapsible information, pagination and indexes among many other--; as well, the graphical presentation has been designed giving the most active role to the visual elements, being the diagrams the cornerstone upon which all the other information lies as well as innovating with the incorporation of the comparative area.

After, it has been defined and architecture and the business logic needed to support all the necessities of the implemented GUI. While the retrieve of the ranked matching models with semantic similarities for a performed query is simulated.

Finally, the most important UmlModels business rules have been draw from the client; and as result of this process a complex database has been designed and deployed as well as a framework for access and extract the information.

All in all, the work presented here is the first step of a constant evolution towards a new conception of the software documentation in which we have overcome the technology to make such a complex abstraction become real. Subsequently are presented some of this future approaches that UmlModels might tackle.



11.1 Project Management

In this section is assessed some personal evaluations about the management of the project.

A major problem when you are enrolled in a big project as the final thesis degree is that you are sometimes focusing on the end of the project. You are eager to finish the degree. However, as you get deep into a subject you start making it yours and you just want to make it better, even though if you suppose to focus on one part it is inevitable to take more.

On the other hand, to build a portal from scratch is a hard task, especially if exists a lack of experience in this area. Even more when you are trying to use a methodology that takes time with a tight budget and limited resources. In fact, it is even more complicated when you have some specifications that changes when the project is highly advanced and you have to go back and give rise with the consequences. I shorts, it is frustrating when you see that the 80% of your time for a project has pass and still are changing important requirements.

In real projects, problems appear... I feel I have work really hard on this project and I wish I could carry on with it until the final deployment.

12 Future Lines

Nowadays when a user performs a search, the terms described in the query are matched with the textual information lodged in the title and in the description. In the future, UmlModels users are going to be able to perform searches with a specific language of the domain, graphical searches (**diagrammer**) and interactive searches (**virtual assistance**). All of them will understand the **semantic** of the query performed; and the retrieve of this information will be performed with the already commented **Polar Retrieval**.

In addition, the GUI is able to present almost any kind of software artifact; however up to now, only images are automatically retrieved by the crawler thus only software diagrams are indexed. On the future, the bots will have intelligence enough to extract the semantic from other elements such as source code, packages, assemblies and many others; thus other files such us ZIP files, power points, PDFs, ... will be retrieved.

Something else to be mention is all the economical business models that UmlModels will cover. It has already described the *Bank Job Offers* in which the users can create or execute offers exchanging the work done for an economical reward when one offer is fulfilled. Moreover, UmlModels will be a powerful case tool available all around the world (with just an internet connection) and with no additional software. Bearing in mind this, another business area may be the outsourcing of the case tool of an enterprise (a group of users sharing private models). Even more, UmlModels might be also hosted in the servers of an external enterprise and UmlModels can provide technical support. UmlModels will welcome all the private enterprise that they want to use UmlModels into their intranet (their own servers). Also UmlModels will provide privacy mechanism to the users, group of users or enterprise that they only want a specific number of mates collaborating in their work.

On another note, UmlModels is trying to gather the main existing ways to document and embed different file formats and media files into an area/place easily accessible everywhere you can connect to the Internet. Many advantages can be taken from here such us no worries storing the versions, the storage, wrong references to renamed or moved documents and many, many others. Here, many new ideas can turn up with this software documentation philosophy that UmlModels wants to bring. Internet is based on giving information and referencing other information.

According to the latter, a future line would be to plays with the way you can reference things on the Internet. The usual way to make a reference on the Internet is making and hyperlink to the Internet resource you might reference. But sometimes that's not useful if you want the users to stay in your site. In this case, some Sites offers you the possibility to embed the content into your page. An example of this is Youtube.

As it is show in *Illustration 111: Screenshot of Youtube embed into Coveralia*, Youtube offers a complete set of options embed into Coveralia: First you can watch a video; You can also rate it; Either you can pass forward, pause it...; You can even search more videos as well as you can go to Youtube.

Wouldn't it be wonderful to have something similar in the software documentation? The idea is to have in interactive object (like SWF *flash* object) which you can embed it into other web pages and interact (change views...) with the public view of an UML artefact such as a model with his videos, diagrams...

Illustration 111: Screenshot of Youtube embed into Coveralia



Also, a powerful tool that UmlModels can add to the software revision of documentation is an option in which the users can edit a model as if they were the owners. This new version will be stored as version of the model and also will be sent to the owner as a new proposal revision. The owner will have the following options: accept, reject and modify the proposal revision. If the revision is accepted, the version will be added as the main (or current) model. If the owner modifies it, then this new version will be added as main (or current) version and the other one won't be removed; in all the cases all the versions will be stored.

Another tendency that can grow up in the future is all the activity outside the portal that can be generated with the social networks. In the future, some of these activities can be included into UmlModels. Examples of these social networks are pointed out as follows:

1. Help: <http://groups.google.es/group/umlmodels>
2. News (Blog, RSS):
 - a. Facebook: <http://www.facebook.com/group.php?gid=138413735359>
3. Videos: <http://www.youtube.com/UMLModels>

They have been considered a wide range of different future lines but they are many more. Here, to conclude, it has to be said that the door is open to a huge world of new revolutionary ideas:

"The more you can dream, the more you can do" --- Michael Korda.

13 Reports of Meetings

Due the wide extension of the all reports and the fact that almost all its content and the decisions taken in the meetings have been explained during this project report, they won't be enclosed here.

Subsequently is presented all the reports file name. The name of the reports it is self explained; in this, it is possible to find the date of the meeting and the name of the recorded session (if any). These reports can be found in the soft source documentation ant they contain the following minutes of meetings:

- ➔ Minutes of the meeting.2008.07.08.Information Model o Página de resultados, Juan-Julián.wmv.docx
- ➔ Minutes of the meeting.2008.12.10.GUI Parte 1 Juan-Guillermo.wmv.docx
- ➔ Minutes of the meeting.2008.12.12.Advanced information model.docx
- ➔ Minutes of the meeting.2008.12.16.Pagina de Resultados por niveles.wmv.docx
- ➔ Minutes of the meeting.2008.12.19.Pagina del Modelo.wmv.docx
- ➔ Minutes of the meeting.2008.12.23.Era Post Guillermo.wmv.docx
- ➔ Minutes of the meeting.2008.12.28.Modelo Conceptual.wmv.docx
- ➔ Minutes of the meeting.2008.12.31.Modelo Conceptual entero.wmv.docx
- ➔ Minutes of the meeting.2009.01.09.Modelo Conceptual 2 - Juan Julián Guillermo.wmv.docx
- ➔ Minutes of the meeting.2009.01.15.Modelo de Implementacion JGJ.wmv.docx
- ➔ Minutes of the meeting.2009.01.15.Pagina del Modelo 2.wmv.docx
- ➔ Minutes of the meeting.2009.01.19.Modelo de Implementacion II.wmv.docx
- ➔ Minutes of the meeting.2009.01.27.Pagina del Modelo 3.wmv.docx
- ➔ Minutes of the meeting.2009.01.28.Arquitectura 2.wmv.docx
- ➔ Minutes of the meeting.2009.01.28.Arquitectura.wmv.docx
- ➔ Minutes of the meeting.2009.01.30.Modelo Comercial.wmv.docx
- ➔ Minutes of the meeting.2009.02.05.Arquitectura 3 Debate para llegar a Solucion.wmv.docx
- ➔ Minutes of the meeting.2009.02.05.Arquitectura 4 Acordada la Solucion final.wmv.docx
- ➔ Minutes of the meeting.2009.02.09.Arquitectura 5 Paso a diseño.wmv.docx
- ➔ Minutes of the meeting.2009.02.09.Arquitectura 6 Diseño de capas.wmv.docx
- ➔ Minutes of the meeting.2009.02.09.Arquitectura 7 Diseño específico de las 3 capas.wmv.docx
- ➔ Minutes of the meeting.2009.02.26.Pagina del Modelo 4.wmv.docx
- ➔ Minutes of the meeting.2009.03.09.Pagina del Modelo 5.wmv.docx

14 Appendix A: Free and open source software CMS

Name	Platform	Supported databases	Latest stable release	Licenses
AdaptCMS Lite	PHP	MySQL	1.0	GPL
Alfresco	Java	MySQL, Oracle, SQL Server, PostgreSQL, Informix	2.1	GPL
Apache Lenya	Java, XML, Apache Cocoon		2.0	Apache License
b2evolution	PHP	MySQL	1.10.3 "Key West"	GPL
BLOG:CMS	PHP	MySQL		GPL
blosxom	Perl	Flat-file database	2.0	MIT
Bricolage	Perl	PostgreSQL	1.10.3	BSD
CMSimple	PHP	Flat-file database	2.9	Affero
Cyclone3	Perl, XUL, JavaScript, C, Java	MySQL and any Perl DBI	3.0	GPL
Daisy	Java, XML, Apache Cocoon	MySQL	2.0.1	Apache License
Dokuwiki	PHP	Flat-file database	6/11/06	GPL
DotClear	PHP	MySQL Beta version 2.0 supports PostgreSQL	1.2.5	GPL
DotNetNuke	ASP.NET	Microsoft SQL Server (out-of-the-box) or any data storage system (each storage system needs a custom data provider to be installed)	4.6.2	BSD
Drupal	PHP	MySQL or PostgreSQL	6	GPL
DrakeCMS	PHP	Flat-file database / MySQL / PostgreSQL / ADOdb_Lite	0.4.11	GPL
e107	PHP	MySQL	0.7.10	GPL

Name	Platform	Supported databases	Latest stable release	Licenses
eZ Publish	PHP	MySQL/PostgreSQL/Oracle/Microsoft SQL Server	4.0.0	GPL
Exponent	PHP	MySQL/PostgreSQL	0.96.6	GPL
Fedora	Java	MySQL or Oracle	2.2	Educational Community License
Jahia	Java	HyperSonic SQL, PostgreSQL, MySQL, Oracle	5.0.3	JCDDL and JSSL
jAPS - java Agile Portal System	Java, XML on Windows or Linux	HyperSonic SQL, PostgreSQL		GPL
Joomla!	PHP	MySQL	1.5.1	GPL
Liferay Journal	Java	Apache Derby, DB2, HyperSonic SQL, Informix, InterBase, JDataStore, MySQL, Oracle, PostgreSQL, SAP, SQL Server, Sybase	4.4.2	MIT
KnowledgeTree Document Management System	PHP	MySQL	3.5	GPL
Lyceum	PHP	MySQL		GPL
Magnolia	Java	JCR	3.5.4	GPL
Mambo	PHP	MySQL	4.6.2	GPL
MediaWiki	PHP	MySQL, PostgreSQL	1.11.0	GPL
Midgard CMS	PHP (Midgard framework)	MySQL		LGPL
MODx	PHP	MySQL	0.9.6.1	GPL
MoinMoin	Python	Flat-file database	1.6.0	GPL
Movable Type	Perl, mod_perl, FastCGI	MySQL or MS SQL server or Oracle or PostgreSQL or SQLite	4.01	GPL

Name	Platform	Supported databases	Latest stable release	Licenses
Nucleus CMS	PHP	MySQL	3.23	GPL
Nuxeo CPS	Python	ZODB	3.4.3	GPL
Nuxeo EP	Java	PostgreSQL, MySQL, Oracle, SQL Server, Ingres	5.1.1	LGPL
OneCMS	PHP	MySQL	2.5	GPL
OpenACS	TCL AOLserver	PostgreSQL/Oracle	5.1.5	GPL
OpenCms	Java	MySQL, Oracle, PostgreSQL, SQL Server, DB2, HSQL	7.0.3	LGPL
phpCMS	PHP	Flat-file database	1.2.2	GPL
PHP-Fusion	PHP	MySQL	6.01.13	GPL
PHP-Nuke	PHP	MySQL	8.0	GPL
phpWCMS	PHP	MySQL	1.3.3	GPL
phpWebSite	PHP	MySQL or PostgreSQL	1.1.0	LGPL
PhpWiki	PHP	Flat-file database/MySQL/PostgreSQL etc.		GPL
Plone	Python	ZODB, SQLite, PostgreSQL, MySQL, Oracle via Zope	3.0.5	GPL
PmWiki	PHP	Flat-file database	2.1.27	GPL
PostNuke	PHP	MySQL	.764	GPL
PyLucid	Python	MySQL, PostgreSQL, SQLite	0.8.0	GPL
Radiant	Ruby	MySQL, PostgreSQL, SQLite	0.6.4	MIT
Scoop	Perl on mod_perl	MySQL	1.1.8	GPL
Serendipity	PHP + Smarty	SQLite, PostgreSQL, MySQL, MySQLi	1.2	BSD
SilverStripe	PHP	MySQL	2.2.1	BSD
SiteFrame	PHP + Smarty	MySQL	5.0.2	Creative Commons

Name	Platform	Supported databases	Latest stable release	Licenses
Slash	Perl on mod_perl	MySQL		GPL
SPIP	PHP	MySQL	1.9.2	GPL
TangoCMS	PHP	MySQL	1.0.8-Osprey	GNU/GPL 2
Textpattern	PHP	MySQL	4.0.6	GPL
TGS Content Management	PHP	MySQL	0.2.5r3	GPL
TikiWiki CMS/Groupware	PHP	ADODB	1.9.9	LGPL
TWiki	Perl	Perl DBI compatible	4.1.2	GPL
Typo	Ruby on Rails	MySQL, PostgreSQL, SQLite		MIT
TYPO3	PHP	MySQL, PostgreSQL, Oracle	4.1.5	GPL
Quick.Cms.Lite	PHP	Flat-file database	2.0	GPL
WebGUI	Perl on mod_perl	MySQL		GPL
Website Baker	PHP	MySQL		GPL
whCMS	PHP	MySQL	0.102	GPL
WordPress	PHP	MySQL	2.3.3	GPL
Xaraya	PHP with XHTML/XML/XSLT	MySQL, PostgreSQL, SQLite using ADODB and Microsoft SQL Server with Creole	1.1.3	GPL
XOOPS	PHP	MySQL	2.0.18	GPL
XOOPS Cube	PHP	MySQL	2.1.3	BSD
Zena	Ruby	MySQL	alpha	MIT

Table 113: State of the Art: Free open source software CMS.

15 Appendix B: UmlModels Requirements by Gonzalo Génova (Spanish)

As it has been said, UmlModels is a project in constant evolution. During the elaboration of my project many new requirements were starting and many more will. Some of them affected to my project --and they were included in the 8.2 Requirements section-- and some other will affect to the site in the future. For this reason, in this appendix are enclosed the last version of these requirements done by **Gonzalo Génova**:

15.1 Preámbulo de los requisitos.

Actualmente están descritos solamente los requisitos del paquete Sitio Web.

Tipos de requisitos

Sólo hay dos tipos generales de requisitos, siguiendo la nomenclatura del estándar ESA. El tipo de requisito, junto con un número generado automáticamente, forma el identificador de cada requisito:

- Capacidad (CAP): funciones y operaciones requeridas por el cliente para resolver un problema o alcanzar un objetivo; describe una operación, o secuencia de operaciones, que el software debe ser capaz de realizar.
- Restricción (RES): restricciones impuestas por el cliente sobre la manera en que el problema es resuelto o el objetivo es alcanzado; restringe la manera en que el software es construido o funciona, sin alterar o describir las capacidades del software.

Fuente de los requisitos

Remite a la fuente original de donde se ha obtenido la inspiración para el requisito, aunque en su forma definitiva puede tener poco que ver con la fuente:

- Documento previo de requisitos preparado por Juan Llorens, indicando identificador del requisito.
- Reuniones de trabajo: Videoconferencia 18-09-2008.
- Proyectos Fin de Carrera: Jorge Alonso, Johanna Gallo.
- Aplicaciones en uso: Página de descargas de Requirements Studio.
- Personas concretas: Juan Llorens, etc.

Reuniones de validación de requisitos

En las reuniones de validación se modifica a fondo el contenido de los requisitos antes de aceptarlos. No obstante, considerar que estas reuniones son “fuente” para los requisitos y anotarlas como tales en cada requisito sería excesivo, por lo que simplemente se relacionan a continuación dichas reuniones:

- 31-10-2008. Juan Llorens, Gonzalo Génova, George Andreadakis, Julián Urbano, Guillermo Suárez. Revisión de la versión 1 de fecha 30-10-2008, paquete Sitio Web.
- 25-11-2008. Gonzalo Génova, George Andreadakis, Julián Urbano, Guillermo Suárez. Revisión de la versión 2 de fecha 10-11-2008, paquete Sitio Web.

Estado de los requisitos

Cada requisito puede estar en uno de los siguientes estados:

- Propuesto: el ingeniero de requisitos lo ha propuesto.
- Aceptado: el cliente lo ha aceptado/validado.
- Cancelado: el cliente ha cancelado el requisito, ya no se debe implementar.
- Implementado: el equipo de desarrollo lo ha implementado.
- Verificado: el equipo de pruebas ha verificado que se satisface.

Anotaciones a los requisitos

Algunos requisitos contienen aclaraciones u observaciones entre paréntesis al final de la descripción detallada. La aceptación de un requisito se refiere sólo al texto principal del mismo, no a las anotaciones, que deben considerarse como información provisional.

Prioridad de los requisitos

La prioridad de un requisito hace referencia al orden temporal: indica en qué fase de construcción del sistema se incluirá la funcionalidad que realice el requisito. Actualmente están definidas las siguientes prioridades: 1, 2, 3.

Necesidad de los requisitos

La necesidad de un requisito hace referencia al interés de los usuarios/clientes en que la aplicación lo realice, y hasta qué punto estarían dispuestos a pasarse si él. Actualmente están definidos tres valores de necesidad, aunque no son usados: Alta, Media, Baja.

Relaciones entre requisitos

Se asume que los requisitos pertenecientes a un mismo paquete están moderada o fuertemente acoplados, por lo que no es necesario reflejar las relaciones entre ellos. Sólo se expresan, por tanto, las relaciones entre requisitos pertenecientes a distintos paquetes.

Estados de las cuentas de usuario

El siguiente diagrama expresa gráficamente el contenido del paquete Sitio Web / Roles / Estados. Por simplicidad no se han representado los cambios de estado que puede realizar el Administrador (requisito CAP0093-1).

15.2 UMLmodels

El paquete no contiene requisitos.

1. Sitio Web

El paquete no contiene requisitos.

1.1. Roles

Identificador	CAP0001-1	Tipo	Capacidad
Descripción breve	Existencia de roles		
Creado por	Gonzalo Génova Fuster	Creación	23/10/2008 17:05:25
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:54:35
Descripción detallada	<p>Existen cuatro roles de usuario: Visitante (Visitor), Cliente (Client), Operador (Operator), Administrador (Administrator). Los usuarios de los tres últimos roles se denominan también de modo genérico "usuarios registrados".</p> <p>(Podría haber un quinto rol entre Cliente y Visitante denominado Suscriptor. Se registra como los demás, pero es gratuito. Recibe notificaciones sobre actualizaciones del sistema, y otro tipo de noticias generadas por los Administradores. Tal vez acceso a la zona de valor añadido gratuita. Por otra parte, si se asume que un Cliente sin saldo es lo mismo que un Suscriptor, entonces nos ahorramos la creación de un nuevo rol.</p> <p>(Podría haber un sexto rol al mismo nivel que Cliente denominado Proveedor. No está claro si es necesario o sería el mismo rol Cliente.)</p>		
Estado	Aceptado		
Fuente	2.1-RO-00018		
Necesidad			
Prioridad	1		

Identificador	CAP0002-1	Tipo	Capacidad
Descripción breve	Jerarquía de roles		
Creado por	Gonzalo Génova Fuster	Creación	23/10/2008 17:07:05
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:54:38
Descripción detallada	<p>Los roles son jerárquicos: todo lo que puede hacer un Visitante lo puede hacer un Cliente, todo lo que puede hacer un Cliente lo puede hacer un Operador, y todo lo que puede hacer un Operador lo puede hacer un Administrador.</p> <p>(En realidad la jerarquía no es tan pura. Por ejemplo, el aspecto económico de los Clientes no es heredado por Operadores y Administradores.)</p>		
Estado	Aceptado		
Fuente	2.1-RO-00018		
Necesidad			
Prioridad	1		

Identificador	CAP0004-1	Tipo	Capacidad
Descripción breve	Usuarios Administradores por defecto		
Creado por	Gonzalo Génova Fuster	Creación	23/10/2008 17:12:45
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 17:40:35
Descripción detallada	El rol Administrador tiene al menos dos usuarios, creados por defecto al instalar la aplicación. Los usuarios de este rol pueden cambiar, pero siempre deberán ser al menos dos.		
Estado	Propuesto		
Fuente	PFC Jorge Alonso		
Necesidad			
Prioridad	1		

Identificador	CAP0009-1	Tipo	Capacidad
Descripción breve	Usuarios Visitantes		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:13:41
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:55:35
Descripción detallada	Los usuarios Visitantes no son usuarios registrados del sistema, es decir, pueden acceder libremente a las funciones del rol Visitante sin necesidad de registrarse ni de iniciar sesión, y sin límite de sesiones simultáneas.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0003-1	Tipo	Capacidad
Descripción breve	Identificador de usuario		
Creado por	Gonzalo Génova Fuster	Creación	23/10/2008 17:09:16
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:54:41
Descripción detallada	Los usuarios registrados tienen un Identificador auto numérico de usuario que no es visible ni modificable por ellos.		
Estado	Aceptado		
Fuente	PFC Jorge Alonso		
Necesidad			
Prioridad	1		

Identificador	CAP0034-1	Tipo	Capacidad
Descripción breve	Contraseña de usuario		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 10:45:57
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 17:37:35
Descripción detallada	<p>Los usuarios registrados tienen una Contraseña de al menos 4 caracteres alfanuméricos (incluyendo signos de puntuación y otros símbolos) que pueden cambiar siempre que lo deseen. El sistema asesora al usuario sobre la robustez de la contraseña cada vez que quiera cambiarla. Las contraseñas se almacenan y transmiten de manera segura. Las contraseñas nunca se envían por correo electrónico.</p> <p>(Investigar si existe algún requisito legal sobre la robustez y almacenamiento de contraseñas. Puede ser interesante esta dirección: http://www.passwordmeter.com/.)</p>		
Estado	Propuesto		
Fuente	Julián Urbano		
Necesidad			
Prioridad	1		

Identificador	CAP0018-1	Tipo	Capacidad
Descripción breve	Perfil de los usuarios		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:59:54
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 17:35:07
Descripción detallada	<p>Los usuarios registrados tienen en su perfil, además de la Contraseña, un Nombre, un Apellido (en un único campo aunque sean dos), un Apodo (Nickname, opcional), una Institución a la que pertenecen (opcional), un Número máximo de sesiones simultáneas, y una Dirección de correo electrónico. Todos estos datos del perfil los pueden cambiar siempre que lo deseen. La Dirección de correo electrónico es sometida a una comprobación sintáctica antes de admitirla como válida. Los usuarios Administradores tienen además un número de teléfono con el fin de recibir notificaciones vía SMS.</p> <p>(En el caso de los Clientes, el número máximo de sesiones simultáneas influirá en la cuota.)</p>		
Estado	Propuesto		
Fuente	descargas de reqStudio		
Necesidad			
Prioridad	1		

Identificador	CAP0062-1	Tipo	Capacidad
Descripción breve	Fecha de alta y modificación		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 18:11:24
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 17:34:55
Descripción detallada	La fecha de alta de un usuario registrado no es modificable. Se registra además la fecha de última modificación del estado, que es modificada automáticamente al cambiar el estado.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0065-1	Tipo	Capacidad
Descripción breve	Inicio de sesión de los usuarios		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:00:34
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:55:18
Descripción detallada	Los usuarios registrados inician sesión en el sistema mediante su Dirección de correo electrónico y su Contraseña.		
Estado	Aceptado		
Fuente	descargas de reqStudio		
Necesidad			
Prioridad	1		

Identificador	CAP0066-1	Tipo	Capacidad
Descripción breve	Recuerdo local de contraseña		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:13:31
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:55:26
Descripción detallada	Se permite recordar localmente la contraseña de usuario entre sesiones.		
Estado	Aceptado		
Fuente	Julián Urbano		
Necesidad			
Prioridad	1		

1.1.1. Estados

Identificador	CAP0015-1	Tipo	Capacidad
Descripción breve	Estados de las cuentas de usuario		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:49:12
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:19:28
Descripción detallada	<p>Las cuentas de usuarios registrados pueden estar en uno de estos Estados:</p> <ul style="list-style-type: none"> - Inactiva: cuenta recién creada pero aún no verificada, no permite el acceso a ninguna de las funciones propias del rol del usuario. - Activa: permite el acceso a todas las funciones propias del rol del usuario. - Bloqueada: debido a superar el límite de intentos de acceso, no permite el acceso a ninguna de las funciones propias del rol del usuario. - Baja: a petición propia, no permite el acceso a ninguna de las funciones propias del rol del usuario. 		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0008-3	Tipo	Capacidad
Descripción breve	Creación de cuentas de usuario		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:10:36
Modificado por	Admin admin admin	Modificación	21/01/2009 13:31:49
Descripción detallada	Un Visitante puede solicitar la creación de un usuario con el rol Cliente, rellenando un formulario de registro con los datos de su perfil, incluyendo la Contraseña, con mecanismos de bloqueo para usuarios no humanos. La cuenta se crea en el estado Inactiva y se envía al usuario un correo electrónico solicitando la verificación de la cuenta mediante un enlace codificado.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0088-1	Tipo	Capacidad
Descripción breve	Destrucción de cuentas de usuario inactivas		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 17:56:19
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:14:35
Descripción detallada	Si la cuenta Inactiva no es verificada correctamente (por manipulación del enlace de verificación), entonces la cuenta se destruye automáticamente. Así mismo, si la cuenta no ha sido verificada al cabo de X días, entonces la cuenta se destruye automáticamente, por defecto X=10.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0089-1	Tipo	Capacidad
Descripción breve	Activación de cuentas de usuario inactivas		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 18:02:21
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:14:41
Descripción detallada	Si la cuenta Inactiva es verificada correctamente mediante el enlace de verificación, entonces la cuenta pasa al estado Activa.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	RES0019-2	Tipo	Restricción
Descripción breve	Cambio de rol de un usuario activo		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 18:06:32
Modificado por	Admin admin admin	Modificación	21/01/2009 13:32:35
Descripción detallada	<p>Un Administrador puede cambiar el rol de un usuario cuya cuenta esté Activa. El cambio es notificado vía correo electrónico al usuario afectado y a todos los usuarios Administradores. Un Administrador puede cambiar de rol sólo si quedan al menos otros dos Administradores activos.</p> <p>(Es decir, que nuevos Operadores o Administradores deben crearse con el rol Cliente, y luego se les cambia el rol. El motivo de no crearlos directamente en su rol surge de la necesidad de evitar que la contraseña se envíe por correo electrónico, y además así se unifica el proceso de creación de cuentas.)</p>		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0063-1	Tipo	Capacidad
Descripción breve	Bloqueo de cuentas de usuario activas		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 18:35:38
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:42:02
Descripción detallada	La cuenta de un usuario activo pasa automáticamente al estado Bloqueada tras X intentos fallidos de acceso al sistema, por defecto X=10. En este caso se alerta vía correo electrónico al usuario afectado y en el mismo mensaje se proporciona al usuario un enlace codificado de verificación para desbloquear la cuenta.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0091-1	Tipo	Capacidad
Descripción breve	Desbloqueo de cuentas de usuario bloqueadas		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 18:24:16
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:24:36
Descripción detallada	Si la cuenta Bloqueada es verificada correctamente mediante el enlace de verificación, entonces la cuenta pasa al estado Activa.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0076-1	Tipo	Capacidad
Descripción breve	Baja de cuentas de usuarios activas		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 13:11:16
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:40:48
Descripción detallada	Un usuario cuya cuenta esté Activa puede solicitar que su cuenta sea dada de baja, en cuyo caso la cuenta pasa a estado Baja. Un Administrador puede solicitar su propia baja sólo si quedan al menos otros dos Administradores activos.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0092-1	Tipo	Capacidad
Descripción breve	Destrucción de cuentas de usuarios activas		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 18:28:06
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:40:50
Descripción detallada	<p>Un usuario cuya cuenta esté Activa puede solicitar que todos los datos de su cuenta sean destruidos, en cuyo caso se destruye su cuenta, su historial de búsqueda y su historial de operaciones. Un Administrador puede solicitar su propia destrucción sólo si quedan al menos otros dos Administradores activos.</p> <p>(Es necesario revisar la posibilidad legal de destruir sólo los datos estrictamente personales, pero conservando los historiales para nuestro propio beneficio.)</p>		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0093-1	Tipo	Capacidad
Descripción breve	Cambios de estados por un Administrador		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 18:35:59
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:40:40
Descripción detallada	<p>Un Administrador puede realizar los siguientes cambios de estado en todas las cuentas de usuario:</p> <ul style="list-style-type: none"> - Activa -> Bloqueada, siempre que queden al menos dos Administradores activos. - Bloqueada -> Activa. - Activa -> Baja, siempre que queden al menos dos Administradores activos. - Baja -> Activa. - Destrucción de cuentas en estado Activa, siempre que queden al menos dos Administradores activos. - Destrucción de cuentas en estado Baja. - Destrucción de cuentas en estado Bloqueada. 		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0068-1	Tipo	Capacidad
Descripción breve	Destrucción de cuentas de usuarios		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:39:17
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:40
Descripción detallada	<p>Un Administrador puede destruir la cuenta de un usuario registrado, junto con todos sus datos asociados. La modificación es notificada vía correo electrónico al usuario afectado.</p>		
Estado	Aceptado		
Fuente	Juan Llorens		
Necesidad			
Prioridad	2		

1.2. General

Identificador	RES0001-1	Tipo	Restricción
Descripción breve	Idioma de la interfaz		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:30:24
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:57:46
Descripción detallada	Toda la interfaz de usuario se realizará en inglés (American Spelling), pero dejando abierta la posibilidad a otros idiomas.		
Estado	Aceptado		
Fuente	PFC Jorge Alonso		
Necesidad			
Prioridad	1		

Identificador	RES0002-1	Tipo	Restricción
Descripción breve	Tecnología de la interfaz		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:37:22
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:57:47
Descripción detallada	Todas las funciones de todos los roles se realizan mediante interfaces en web. La codificación debe realizarse sin utilizar aspectos propios de ningún navegador, para asegurar su correcta visualización en los navegadores Internet Explorer y Mozilla Firefox.		
Estado	Aceptado		
Fuente	PFC Johanna Gallo		
Necesidad			
Prioridad	1		

Identificador	RES0017-1	Tipo	Restricción
Descripción breve	Tecnología del servidor web		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:22:46
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:57:58
Descripción detallada			
Estado	Propuesto		
Fuente	Juan Llorens		
Necesidad			
Prioridad	1		

Identificador	RES0018-1	Tipo	Restricción
Descripción breve	Tecnología del servidor de base de datos		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:23:15
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:00
Descripción detallada			
Estado	Propuesto		
Fuente	Juan Llorens		
Necesidad			
Prioridad	1		

Identificador	RES0016-1	Tipo	Restricción
Descripción breve	Punto de entrada y datos de contacto		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:16:41
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:08:23
Descripción detallada	El punto de entrada de la aplicación es www.umlmodels.com . Las páginas principales de la aplicación contendrán la siguiente dirección de contacto para resolución de problemas: info@umlmodels.com .		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	RES0009-1	Tipo	Restricción
Descripción breve	Usuario y rol en la interfaz		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 16:17:14
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:02
Descripción detallada	En todo momento serán visibles en un lugar discreto de la interfaz el nombre del usuario y el rol con que ha accedido a la aplicación.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	RES0011-1	Tipo	Restricción
Descripción breve	Expiración de sesión inactiva		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 17:08:49
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:03
Descripción detallada	Una sesión de usuario expira al transcurrir X minutos de inactividad, por defecto X=30.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	RES0004-1	Tipo	Restricción
Descripción breve	Interfaz autoexplicada		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:28:53
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:05
Descripción detallada	Todas las páginas de la aplicación serán autoexplicadas con el nivel de detalle suficiente para que un usuario pueda aprender a usar la aplicación usándola.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	RES0005-1	Tipo	Restricción
Descripción breve	Manuales de usuario		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:30:18
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:06
Descripción detallada	Existe un manual de usuario detallado por cada uno de los roles existentes, en formato HTML y PDF.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	3		

Identificador	RES0006-1	Tipo	Restricción
Descripción breve	Demostraciones		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:31:45
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:08
Descripción detallada	Existen demos en YouTube asociadas a las funciones principales de cada uno de los roles públicos de usuario (Cliente, Visitante).		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	2		

Identificador	RES0007-1	Tipo	Restricción
Descripción breve	Disponibilidad del sistema		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 18:17:58
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:09
Descripción detallada	El sistema debe estar disponible 24 horas al día, 7 días a la semana.		
Estado	Aceptado		
Fuente	PFC Johanna Gallo		
Necesidad			
Prioridad	1		

Identificador	RES0012-1	Tipo	Restricción
Descripción breve	Recuperación del sistema tras fallo de suministro		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 17:44:48
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:10
Descripción detallada	El sistema debe recuperarse automáticamente ante un fallo de suministro eléctrico sin necesidad de intervención humana.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	2		

Identificador	RES0008-1	Tipo	Restricción
Descripción breve	Tiempo de respuesta		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 18:19:21
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:57:26
Descripción detallada	<p>El tiempo máximo de respuesta del sistema (desde la recepción de una petición de carga de una página hasta que ésta es servida) debe ser de X segundos.</p> <p>(Por definir: X; si se tiene en cuenta la tasa de visitas para permitir una respuesta más lenta en caso de sobrecarga; si se tiene en cuenta el volumen de modelos indexados para permitir una respuesta más lenta a medida que el tamaño del repositorio crece; si se establece un tiempo de respuesta más breve para las funciones de administración que para las funciones de búsqueda y recuperación.)</p>		
Estado	Propuesto		
Fuente	PFC Johanna Gallo		
Necesidad			
Prioridad	3		

Identificador	RES0013-1	Tipo	Restricción
Descripción breve	Ley de Protección de Datos		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 18:22:45
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 15:58:13
Descripción detallada	<p>Se respetará la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal (LOPD) en lo que se refiere a la información contenida en los perfiles de usuarios y los historiales de operaciones/búsquedas por usuario.</p> <p>(Real Decreto 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal)</p> <p>(Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico)</p>		
Estado	Aceptado		
Fuente	PFC Johanna Gallo		
Necesidad			
Prioridad	1		

1.3. Zona Administración

Identificador	CAP0059-1	Tipo	Capacidad
Descripción breve	Parámetros de configuración del sistema		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 17:14:29
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:55:18
Descripción detallada	<p>Un Administrador puede configurar los valores de los diferentes parámetros de funcionamiento del sistema:</p> <ul style="list-style-type: none"> - Tiempo de expiración de sesión por inactividad. Este valor se configura directamente en el servidor web. - Programación de copias de respaldo. Este valor se configura directamente en el gestor de base de datos. - Número de días para destruir una cuenta inactiva. - Límite de intentos de acceso para bloquear una cuenta. - Notificación de cambios de estado de usuarios. - Notificación de alarma por sobrecarga del sistema. 		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0020-1	Tipo	Capacidad
Descripción breve	Notificación de cambios de estado de usuarios		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:07:20
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:56:08
Descripción detallada	Todos los usuarios con rol Administrador serán notificados si X vía correo electrónico cuando se produzca una creación, destrucción o cambio de estado de usuario, por defecto X=VERDADERO.		
Estado	Aceptado		
Fuente	descargas de reqStudio		
Necesidad			
Prioridad	2		

Identificador	CAP0014-1	Tipo	Capacidad
Descripción breve	Visualizar y modificar cuentas de usuarios		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:40:06
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:56:29
Descripción detallada	Un Administrador puede visualizar y modificar los datos de las cuentas de los usuarios registrados. Las modificaciones son notificadas vía correo electrónico a los usuarios afectados.		
Estado	Aceptado		
Fuente	PFC Jorge Alonso		
Necesidad			
Prioridad	2		

Identificador	CAP0072-1	Tipo	Capacidad
Descripción breve	Visualizar el registro de operaciones de usuarios		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:50:35
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 18:54:30
Descripción detallada	<p>Un Administrador puede visualizar el historial de operaciones de todos los usuarios registrados. Por cada operación se detalla:</p> <ul style="list-style-type: none"> - Fecha y hora. - Identificador del usuario. - Acción. <p>(Habrá que categorizar las acciones de usuario: inicio de sesión, fin de sesión, cambios de estado, acciones de administración, aportación de saldo por un cliente, etc.)</p>		
Estado	Propuesto		
Fuente	Julián Urbano		
Necesidad			
Prioridad	2		

Identificador	CAP0021-1	Tipo	Capacidad
Descripción breve	Visualizar registro de eventos		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:08:08
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:43
Descripción detallada	Un Administrador puede visualizar el registro de eventos del sistema que incluye errores, alarmas y situaciones anómalas, con vistas a tomar las acciones necesarias para resolverlos.		
Estado	Aceptado		
Fuente	2.1-RO-00019		
Necesidad			
Prioridad	1		

Identificador	CAP0061-1	Tipo	Capacidad
Descripción breve	Visualizar historial de búsquedas		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 17:39:27
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:47
Descripción detallada	<p>Un Administrador puede visualizar el historial de búsquedas de todos los usuarios. Por cada búsqueda se detalla:</p> <ul style="list-style-type: none"> - Fecha y hora. - Tiempo de generación de la respuesta en segundos. - Identificador del usuario. - Nombre del usuario. - Dirección IP desde la que se produce la consulta. - Texto de la consulta. 		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0016-1	Tipo	Capacidad
Descripción breve	Arranque y parada de subsistemas		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:52:19
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:44
Descripción detallada	Un Administrador puede arrancar o parar cada uno de los subsistemas que pueden funcionar independientemente, de acuerdo con la planificación de acciones de mantenimiento del sistema.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0057-1	Tipo	Capacidad
Descripción breve	Programación de copias de respaldo		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 15:36:31
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:45
Descripción detallada	Un Administrador puede programar copias de respaldo (backup) periódicas cada X horas, por defecto X=24.		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	2		

Identificador	CAP0017-1	Tipo	Capacidad
Descripción breve	Importar y exportar información del sistema		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 16:54:51
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:46
Descripción detallada	Un Administrador puede importar y exportar toda la información contenida en el sistema, o parte de ella. (Esto puede ser importante para migrar el sistema a versiones nuevas, etc.)		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	3		

1.3.1. Eventos

Identificador	CAP0019-1	Tipo	Capacidad
Descripción breve	Notificación de eventos		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:03:26
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:00:44
Descripción detallada	Todos los usuarios con rol Administrador son notificados vía correo electrónico y SMS cuando se produzca un error, alarma o situación anómala que requiera su atención.		
Estado	Aceptado		
Fuente	2.1-RO-00019		
Necesidad			
Prioridad	1		

Identificador	CAP0071-1	Tipo	Capacidad
Descripción breve	Alarma por fallo de copia de respaldo		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:47:04
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:04:32
Descripción detallada	Error: fallo en el proceso de copia de respaldo.		
Estado	Aceptado		
Fuente	2.1-RO-00019		
Necesidad			
Prioridad	2		

Identificador	CAP0070-1	Tipo	Capacidad
Descripción breve	Alarma por sobrecarga del sistema		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:42:54
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:04:33
Descripción detallada	Alarma: durante los últimos X minutos ha habido más de Y consultas con un tiempo de generación de la respuesta superior a Z segundos, por defecto X=60, Y=10, Z=60.		
Estado	Aceptado		
Fuente	2.1-RO-00019		
Necesidad			
Prioridad	2		

1.4. Zona Clientes

El paquete no contiene requisitos

1.5. Zona Visitantes

Identificador	CAP0025-1	Tipo	Capacidad
Descripción breve	Búsquedas por contenido		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:34:44
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:06:20
Descripción detallada	Un Visitante puede formular consultas textuales con ayuda de un Asistente para realizar búsquedas por contenido entre los modelos previamente indexados en el sistema.		
Estado	Aceptado		
Fuente	2.1-RO-00020		
Necesidad			
Prioridad	1		

Identificador	CAP0036-1	Tipo	Capacidad
Descripción breve	Búsquedas restringidas a un sitio		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:14:18
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 12:53:09
Descripción detallada	La búsqueda puede estar restringida a un sitio web especificado por el usuario. (Pendiente de estudio de viabilidad en la estructura de la base de datos.)		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0029-1	Tipo	Capacidad
Descripción breve	Resultado de la búsqueda		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 18:01:37
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 19:01:23
Descripción detallada	<p>El resultado de la búsqueda se muestra como una lista de imágenes reducidas, de tamaño variable, de los modelos encontrados, junto con una descripción textual intermedia de cada modelo.</p> <p>(Relacionar con el requisito donde se explique que las descripciones textuales son a tres niveles: resumida, intermedia/tooltip, completa.)</p>		
Estado	Aceptado		
Fuente	Videoconferencia 18-09-2008		
Necesidad			
Prioridad	1		

Identificador	CAP0027-1	Tipo	Capacidad
Descripción breve	Ordenación del resultado de la búsqueda		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:45:11
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:06:23
Descripción detallada	<p>Los resultados de la búsqueda se muestran ordenados por la pertinencia del resultado a la consulta formulada. Para evaluar la pertinencia se consideran en este orden:</p> <ol style="list-style-type: none"> 1. Similitud o inclusión de la consulta en el modelo encontrado. 2. Posicionamiento de la página original de donde proviene el modelo. 3. Recuento del número de veces que ese modelo ha sido visitado desde el sistema. 4. Tamaño del foro asociado a un modelo. 5. Votaciones de los usuarios registrados sobre la utilidad del modelo visitado. <p>(Es necesaria una reunión futura en el plazo de un mes para definir mejor estos criterios, con asistencia de todos los interesados.)</p>		
Estado	Aceptado		
Fuente	Videoconferencia 18-09-2008		
Necesidad			

Identificador	CAP0027-1	Tipo	Capacidad
Prioridad	1		

Identificador	CAP0073-1	Tipo	Capacidad
Descripción breve	Oferta de resultados semejantes		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:54:37
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 12:58:03
Descripción detallada	<p>A continuación del resultado de la búsqueda se ofrecen resultados encontrados/descargados por otros usuarios que buscaban algo parecido.</p> <p>(Analogía con Amazon y otros sistemas semejantes. Tal vez sería interesante que los Clientes pudieran pagar por tener enlaces patrocinados.)</p>		
Estado	Propuesto		
Fuente	Julián Urbano		
Necesidad			
Prioridad	2		

Identificador	CAP0028-1	Tipo	Capacidad
Descripción breve	Navegación hacia la página original de un modelo		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:51:38
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:06:25
Descripción detallada	Se puede navegar hacia la página original del modelo encontrado mediante un enlace proporcionado junto con el resultado de la búsqueda.		
Estado	Aceptado		
Fuente	Videoconferencia 18-09-2008		
Necesidad			
Prioridad	1		

Identificador	CAP0074-1	Tipo	Capacidad
Descripción breve	Navegación hacia la página detallada de un modelo		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 12:59:32
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:06:26
Descripción detallada	<p>Se puede navegar hacia la página detallada del modelo encontrado, que es generada y almacenada por el sistema, mediante un enlace proporcionado junto con el resultado de la búsqueda. En esta página se proporciona la imagen original y la descripción textual completa del modelo, Se vuelve a ofrecer la posibilidad de navegar a la página original del modelo y los resultados encontrados/descargados por otros usuarios que buscaban algo parecido.</p> <p>(Esta página proporciona también otras opciones accesibles sólo para Clientes: descargas XMI, etc.)</p>		
Estado	Aceptado		
Fuente	Juan Llorens		
Necesidad			
Prioridad	1		

Identificador	CAP0060-1	Tipo	Capacidad
Descripción breve	Historial de búsquedas por usuario		
Creado por	Gonzalo Génova Fuster	Creación	30/10/2008 17:27:45
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:06:27
Descripción detallada	<p>Un usuario registrado puede acceder al historial de las búsquedas que ha realizado anteriormente, con el fin de facilitar la repetición o refinamiento de búsquedas. Por cada búsqueda se detalla:</p> <ul style="list-style-type: none"> - Fecha y hora. - Texto de la consulta. 		
Estado	Aceptado		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

1.5.1. Consultas textuales

Identificador	CAP0038-1	Tipo	Capacidad
Descripción breve	Descripción de la consulta		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:21:32
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:30
Descripción detallada	La descripción de la consulta consiste en una secuencia de frases elementales con estructura simple sujeto-verbo-objeto, separadas por '.' o cambio-de-línea, que describen lo que el usuario busca en el modelo.		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0075-1	Tipo	Capacidad
Descripción breve	Estructuración de la consulta		
Creado por	Gonzalo Génova Fuster	Creación	10/11/2008 13:03:49
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 19:04:36
Descripción detallada	<p>La descripción de la consulta puede estar estructurada en niveles anidados, de modo que los niveles superiores constituyen un contexto de espacio de nombres para los niveles inferiores.</p> <p>(La entrada en un nivel inferior se indica mediante el símbolo '>'. La salida a un nivel superior se indica mediante el símbolo '<'. La ausencia de símbolo se interpreta como salida al nivel inmediatamente superior, de modo que para mantenerse en un mismo nivel hay que repetir el símbolo '>' al inicio de cada frase.</p> <p>Ejemplo:</p> <pre>class A has attribute M class B has attribute M, N > attribute M is of type T > attribute N is of type S class C has attribute P</pre> <p>Esto sólo es una posibilidad. La solución concreta queda abierta.)</p>		

Identificador	CAP0075-1	Tipo	Capacidad
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	2		

Identificador	CAP0039-1	Tipo	Capacidad
Descripción breve	Idioma de la consulta		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:30:21
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:32
Descripción detallada	La consulta se formula en lenguaje natural inglés, tanto americano como británico.		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0026-1	Tipo	Capacidad
Descripción breve	Consulta retroalimentada		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 17:39:02
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:32
Descripción detallada	El Asistente proporciona retroalimentación al usuario de modo que éste puede aprender qué palabras clave de su descripción son reconocidas y empleadas para transmitir la consulta al repositorio. Esta retroalimentación se realiza simultáneamente de dos formas distintas: mediante el resaltado en negrita de las palabras clave reconocidas sobre el mismo texto del usuario; mediante la traducción en otra ventana de la descripción del usuario a la descripción entendida por el Asistente.		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0031-1	Tipo	Capacidad
Descripción breve	Consulta con comodines		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 18:12:29
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:33
Descripción detallada	El Asistente permite el uso de comodines (*) para los nombres de los distintos elementos que forman parte de la descripción textual de la fórmula de consulta.		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0094-1	Tipo	Capacidad
Descripción breve	Consulta con variables		
Creado por	Gonzalo Génova Fuster	Creación	26/11/2008 19:04:44
Modificado por	Gonzalo Génova Fuster	Modificación	26/11/2008 19:10:06
Descripción detallada	<p>El Asistente permite el uso de variables (*nombre') para los nombres de los distintos elementos que forman parte de la descripción textual de la fórmula de consulta, y que se repiten en distintos lugares.</p> <p>(Ejemplo: búsqueda de modelos que contienen el patrón Composite.</p> <p>class *leaf inherits from class *component</p> <p>class *composite inherits from class *component</p> <p>class *composite is the composition of class *component</p> <p>Se usa el asterisco como prefijo por analogía con los comodines.)</p>		
Estado	Propuesto		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0032-1	Tipo	Capacidad
Descripción breve	Formas canónicas y variaciones		
Creado por	Gonzalo Génova Fuster	Creación	24/10/2008 18:26:16
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:36
Descripción detallada	El asistente es capaz de reconocer variaciones simples sobre cada una de las frases elementales canónicas. Estas variaciones no se describen en los requisitos, sino en documentos anexos.		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0040-1	Tipo	Capacidad
Descripción breve	Clase		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:40:22
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:38
Descripción detallada	El Asistente reconoce frases del tipo "(abstract) class X", donde la palabra clave "abstract" es opcional en ésta y en el resto de formas canónicas donde aparece la palabra clave "class".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0042-1	Tipo	Capacidad
Descripción breve	Clase-Atributo		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:51:44
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:39
Descripción detallada	El Asistente reconoce frases del tipo "class X has attribute Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0043-1	Tipo	Capacidad
Descripción breve	Atributo-Tipo		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:52:14
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:40
Descripción detallada	El Asistente reconoce frases del tipo "attribute X is of type Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0044-1	Tipo	Capacidad
Descripción breve	Clase-Operación		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:52:30
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:41
Descripción detallada	El Asistente reconoce frases del tipo "class X has operation Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0045-1	Tipo	Capacidad
Descripción breve	Operación-Parámetro		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:54:45
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:41
Descripción detallada	El Asistente reconoce frases del tipo "operation X has parameter Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0046-1	Tipo	Capacidad
Descripción breve	Parámetro-Tipo		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:54:46
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:42
Descripción detallada	El Asistente reconoce frases del tipo "parameter X is of type Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0047-1	Tipo	Capacidad
Descripción breve	Operación-Tipo de Retorno		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:54:48
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:43
Descripción detallada	El Asistente reconoce frases del tipo "operation X has return value of type Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0048-1	Tipo	Capacidad
Descripción breve	Relación genérica		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:56:16
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:44
Descripción detallada	El Asistente reconoce frases del tipo "class X is related with class Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0049-1	Tipo	Capacidad
Descripción breve	Relación de dependencia		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:58:15
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:45
Descripción detallada	El Asistente reconoce frases del tipo "class X depends on class Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0050-1	Tipo	Capacidad
Descripción breve	Relación de herencia		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:58:18
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:46
Descripción detallada	El Asistente reconoce frases del tipo "class X inherits from class Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0051-1	Tipo	Capacidad
Descripción breve	Relación de asociación		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:58:20
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:47
Descripción detallada	El Asistente reconoce frases del tipo "class X is associated with class Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0052-1	Tipo	Capacidad
Descripción breve	Relación de agregación		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:58:21
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:47
Descripción detallada	El Asistente reconoce frases del tipo "class X is the aggregation of class Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0053-1	Tipo	Capacidad
Descripción breve	Relación de composición		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 12:01:43
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:49
Descripción detallada	El Asistente reconoce frases del tipo "class X is the composition of class Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0054-1	Tipo	Capacidad
Descripción breve	Clase-Interfaz proporcionada		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 12:01:46
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:50
Descripción detallada	El Asistente reconoce frases del tipo "class X provides interface Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0055-1	Tipo	Capacidad
Descripción breve	Clase-Interfaz requerida		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 12:06:32
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 16:07:51
Descripción detallada	El Asistente reconoce frases del tipo "class X requires interface Y".		
Estado	Aceptado		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	1		

1.5.2. Consultas gráficas

El paquete no contiene requisitos

1.6. Zona Operadores

Identificador	CAP0035-1	Tipo	Capacidad
Descripción breve	Acceso a la documentación del proyecto		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:04:50
Modificado por	Gonzalo Génova Fuster	Modificación	10/11/2008 13:08:15
Descripción detallada	Un Operador puede visualizar y modificar la documentación asociada con el proyecto.		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	2		

2. Indexador

El paquete no contiene requisitos

2.1. Rastreador

Identificador	CAP0077-1	Tipo	Capacidad
Descripción breve	Subsistema Rastreador		
Creado por	Gonzalo Génova Fuster	Creación	18/11/2008 18:00:43
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 17:43:17
Descripción detallada	El subsistema Rastreador (Crawler) navega automática y autónomamente por la red, buscando archivos que puedan ser modelos UML. (Nótese que en todo este paquete de requisitos no se especifica para nada el tipo de archivo que puede ser un modelo UML. Esto deja abierta la posibilidad a procesar no sólo imágenes, sino archivos XML, código fuente, assemblies, etc.)		
Estado	Propuesto		
Fuente	1.1-CR-00052		
Necesidad			
Prioridad	1		

Identificador	CAP0079-1	Tipo	Capacidad
Descripción breve	Nuevos lugares por visitar		
Creado por	Gonzalo Génova Fuster	Creación	18/11/2008 18:20:34
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 16:05:40
Descripción detallada	<p>El Rastreador se alimenta de nuevos lugares por visitar a partir de: (¿quizás decir Recurso en vez de Lugar?)</p> <ul style="list-style-type: none"> - Sitios sugeridos por los usuarios de la aplicación. Ejemplos: www.acme.com, www.acme.com/projects/. - Hiperenlaces encontrados en los lugares visitados. Ejemplos: www.acme.com/coyote/roadrunner-model.jpg, www.roadrunner.com/roadmodel.htm. <p>(Añadir al rol Visitante la posibilidad de sugerir un sitio de búsqueda. Los demás lo heredan, claro. ¿Habría que limitar la posibilidad de sugerir sitios a los Clientes? En mi opinión no, toda sugerencia de un Visitante puede ser beneficiosa.)</p> <p>(¿Qué ocurre si un usuario sugiere un sitio que ya ha sido visitado? Posibilidades: a) no se hace nada; b) se incrementa de alguna manera su prioridad, para volverlo a visitar cuanto antes; c) se toma la sugerencia como un argumento de búsqueda, proporcionando al usuario los resultados ya indexados.)</p> <p>(No he contemplado lo que Juan dice en 1.1-CR-00052 y 1.1-CR-00054 sobre apoyarse en los buscadores generalistas. Sería algo así como especificar una consulta textual cualquiera para pasarla a Google, u otro buscador, e insertar los X (por ejemplo, 100) primeros enlaces proporcionados por Google en la lista de lugares por visitar. No estoy seguro de la utilidad de esto, ni tampoco de su viabilidad.)</p>		
Estado	Propuesto		
Fuente	1.1-CR-00053		
Necesidad			
Prioridad	1		

Identificador	CAP0078-1	Tipo	Capacidad
Descripción breve	Lugares visitados y por visitar		
Creado por	Gonzalo Génova Fuster	Creación	18/11/2008 18:16:32
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 16:26:30
Descripción detallada	<p>El Rastreador mantiene una lista de lugares visitados previamente, o por visitar en el futuro. Por cada lugar se detalla:</p> <ul style="list-style-type: none"> - URL. (¿es más correcto decir URI?) - Fecha y hora de la inserción del lugar en la lista. - Fecha y hora de la última visita realizada. - Si alguna visita anterior a este lugar ha sido calificada como útil para el sistema. - Rol del usuario que ha sugerido este lugar, en su caso. <p>(Basta con especificar el rol, no el usuario concreto, porque sólo el rol se usa en el algoritmo de cálculo de prioridades.)</p> <p>(Por definir: qué se considera una visita útil. Puede ser una que ha proporcionado un archivo que finalmente ha sido indexado. O una que ha proporcionado un enlace a un archivo finalmente indexado. O un sitio que en algún sub-sitio ha proporcionado archivos finalmente indexados. ¿Otras posibilidades?)</p> <p>(Si algún lugar fue útil, pero en una visita posterior ya no lo fue, ¿debe modificarse la utilidad del lugar?)</p>		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0081-1	Tipo	Capacidad
Descripción breve	Prioridad de los lugares por visitar		
Creado por	Gonzalo Génova Fuster	Creación	18/11/2008 18:29:21
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 17:37:08
Descripción detallada	<p>La prioridad de los lugares por visitar se establece básicamente dando mayor prioridad a los lugares más antiguos, es decir, los que llevan más tiempo pendientes de ser visitados. La prioridad P es un número en punto flotante que se calcula de la siguiente manera:</p> <ul style="list-style-type: none"> - A = antigüedad de inserción para lugares aún no visitados, o bien antigüedad de la última visita útil para lugares ya visitados, ambas medidas en segundos. - U = factor de revisitación de lugares útiles, por defecto $U = 0,01$. - R = factor de rol, para sitios sugeridos por un usuario, por defecto $R = 10$. - J = orden jerárquico del rol (Visitante = 0, Cliente = 1, Operador = 2, Administrador = 3). - $P = A * U * R^J$. <p>Ejemplos:</p> <ul style="list-style-type: none"> a) Un lugar aún no visitado, e insertado hace 1 hora, tiene una prioridad de 3.600. b) Un lugar aún no visitado, e insertado hace 1.000 horas (41,66 días), tiene una prioridad de 3.600.000. c) Un lugar ya visitado hace 1 hora, y calificado como útil, tiene una prioridad de $3.600 * 0,01 = 36$. d) Un lugar ya visitado hace 1.000 horas, y calificado como útil, tiene una prioridad de $3.600.000 * 0,01 = 36.000$. e) Un lugar aún no visitado, insertado hace 1 hora, y sugerido por un Administrador, tiene una prioridad de $3.600 * 1000 = 3.600.000$. f) Un lugar ya visitado hace 1.000 horas, calificado como útil, y sugerido por un Administrador, tiene una prioridad de $3.600.000 * 0,01 * 1000 = 36.000.000$. <p>Lo que resulta en este orden de prioridades: $f > e = b > d > a > c$.</p> <p>(Añadir al Administrador los siguientes parámetros de configuración (CAP0059-1): U, R. En cambio, J no sería configurable. Otra opción sería definir Rv, Rc, Ro y Ra como cuatro factores totalmente independientes).</p> <p>(Alternativa: dar mayor prioridad absolutamente a los lugares aún no visitados pertenecientes a sitios sugeridos por un usuario; los otros lugares sólo serían visitados cuando esta lista estuviera vacía. Me gusta menos, porque corre el peligro de "arrinconar" los otros lugares y no visitarlos nunca.)</p> <p>(El usuario Administrador debe poder visualizar la lista de lugares por visitar con sus prioridades en tiempo real.)</p>		
Estado	Propuesto		
Fuente	Gonzalo Génova		

Identificador	CAP0081-1	Tipo	Capacidad
Necesidad			
Prioridad	1		

Identificador	CAP0080-1	Tipo	Capacidad
Descripción breve	Revisión de lugares visitados		
Creado por	Gonzalo Génova Fuster	Creación	18/11/2008 18:26:46
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 15:42:23
Descripción detallada	<p>El Rastreador revisa frecuentemente los lugares que ya ha visitado previamente, de acuerdo con la prioridad de los lugares por visitar.</p> <p>(¿Cómo saber cuando llegas a un sitio si ha sido modificado desde la última vez? ¿Es posible?)</p> <p>(En realidad este requisito es redundante con el anterior.)</p>		
Estado	Propuesto		
Fuente	1.1-CR-00055		
Necesidad			
Prioridad	1		

Identificador	CAP0082-1	Tipo	Capacidad
Descripción breve	Análisis de un lugar visitado		
Creado por	Gonzalo Génova Fuster	Creación	19/11/2008 16:01:24
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 17:39:48
Descripción detallada	<p>Cuando el Rastreador llega a un lugar realiza las siguientes tareas:</p> <ul style="list-style-type: none"> - Si el lugar es en sí mismo potencialmente reconocible como modelo UML, éste se pasa al Prefiltrador. (¿Preseleccionador, Filtrador?) - Si el lugar contiene hiperenlaces a otros lugares, estos se copian e insertan en la lista de lugares por visitar. <p>En el segundo caso, si el lugar visitado ha sido sugerido por un usuario, entonces todos los hiperenlaces copiados que pertenezcan al mismo sitio (¿servidor, nombre de dominio, anfitrión?) se etiquetan como sugeridos por el mismo rol de usuario.</p> <p>Ejemplos: si www.acme.com/projects/ ha sido sugerido por un Cliente, y contiene enlaces a www.acme.com/coyote/roadrunner-model.jpg y www.roadrunner.com/roadmodel.htm, entonces el primer lugar se etiqueta como sugerido por un Cliente, pero el segundo lugar no. Nótese que sólo se tiene</p>		

Identificador	CAP0082-1	Tipo	Capacidad
	<p>en cuenta la coincidencia en el sitio (¿servidor, nombre de dominio, anfitrión?), no hace falta que también coincida la ruta hasta el archivo.</p> <p>(Es necesario incluir un criterio para descartar en bloque todos los hiperenlaces encontrados en un lugar, es decir, para cortar la búsqueda. En caso contrario, acabaríamos navegando por toda la red.)</p> <p>(Nótese que en la navegación no hay ningún criterio basado en el contenido textual de las páginas para evaluar si son útiles o no.)</p> <p>(El rol Administrador debe poder visualizar la actividad del Rastreador: estadísticas de lugares visitados y por visitar, ritmo de visita y análisis, etc.)</p>		
Estado	Propuesto		
Fuente	1.1-CR-00052		
Necesidad			
Prioridad	1		

2.2. Prefiltrador

Identificador	CAP0084-1	Tipo	Capacidad
Descripción breve	Subsistema Prefiltrador		
Creado por	Gonzalo Génova Fuster	Creación	19/11/2008 17:43:50
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 18:28:45
Descripción detallada	<p>El subsistema Prefiltrador selecciona aquellos archivos proporcionados por el subsistema Rastreador que son considerados buenos candidatos como modelos UML, y decide qué Tipo de procesamiento debe seguirse con ellos: Imagen, Ofimática, XMI, Código, Assembly, etc.</p> <p>(Por el momento es sólo Imagen, pero lo escribo así por generalidad.)</p>		
Estado	Propuesto		
Fuente	1.2-KI-00014		
Necesidad			
Prioridad	1		

Identificador	CAP0085-1	Tipo	Capacidad
Descripción breve	Lista de archivos para procesar		
Creado por	Gonzalo Génova Fuster	Creación	19/11/2008 17:46:47
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 18:30:26
Descripción detallada	<p>El subsistema Prefiltrador mantiene una lista de archivos alimentada por el subsistema Rastreador. Por cada archivo se detalla:</p> <ul style="list-style-type: none"> - URL. (¿es más correcto decir URI?) - Fecha y hora de la inserción del archivo en la lista. - Tipo de procesamiento asignado: Imagen, Ofimática, XMI, Código, Assembly. (¿Otros?) <p>La lista de archivos es procesada dando mayor prioridad a los archivos más antiguos, es decir, los que llevan más tiempo pendientes de ser procesados.</p> <p>(El tipo de procesamiento no lo decide el Rastreador. En el caso de imágenes, lo decide el requisito CAP0083-1, y sería análogamente para el resto de tipos.)</p> <p>(No considero necesario añadir una prioridad más complicada, ya que esta fecha de inserción depende de la prioridad de las visitas del Rastreador.)</p> <p>(El procesamiento Ofimática sería el correspondiente a archivos PDF, Word, PowerPoint, etc.)</p>		
Estado	Propuesto		
Fuente	Gonzalo Génova		
Necesidad			
Prioridad	1		

Identificador	CAP0083-1	Tipo	Capacidad
Descripción breve	Selección de archivos de imagen		
Creado por	Gonzalo Génova Fuster	Creación	19/11/2008 17:41:27
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 18:14:55
Descripción detallada	<p>A los archivos de la lista con formato de imagen se les asigna Tipo de procesamiento = Imagen. Las extensiones reconocidas para este tipo de procesamiento son: BMP, JPG, GIF, TIFF. El resto de archivos no se considera (se deja intacto) para este tipo de procesamiento.</p> <p>(La lista completa de extensiones la proporciona Valentín. La lista está inspirada en xnView, aunque no todos los que esa herramienta contempla están implementados. Podría considerarse un requisito distinto por cada tipo de imagen, si se ve conveniente atomizar más aún el requisito.)</p> <p>(Nótese que el resto de archivos encontrados no se elimina de la lista. Cuando se implementen otros tipos de procesamiento podrá ser una lista utilísima.)</p>		
Estado	Propuesto		
Fuente	1.3-B2V-00036		
Necesidad			
Prioridad	1		

Identificador	CAP0086-1	Tipo	Capacidad
Descripción breve	Filtrado de archivos de imagen		
Creado por	Gonzalo Génova Fuster	Creación	19/11/2008 18:15:06
Modificado por	Gonzalo Génova Fuster	Modificación	19/11/2008 18:30:43
Descripción detallada	<p>Los archivos con Tipo de procesamiento = Imagen se filtran de acuerdo con los siguientes criterios:</p> <ul style="list-style-type: none"> - Tamaño de la imagen en el rango Tmin..Tmax, por defecto Tmin=??? y Tmax=??? - Resolución de la imagen en el rango Rmin..Rmax, por defecto Rmin=??? y Rmax=??? - Número de tonos de gris en el rango Gmin..Gmax, por defecto Gmin=??? y Gmax=??? - Número de colores en el rango Cmin..Cmax, por defecto Cmin=??? y Cmax=??? - Número de líneas verticales y horizontales en el rango Lmin..Lmax, por defecto Lmin=??? y Lmax=???, donde una línea debe tener una longitud en píxeles en el rango PLmin..PLmax para ser considerada como tal, por defecto PLmin=??? y PLmax=??? - Número de cajas en el rango Bmin..Bmax, por defecto Bmin=??? y Bmax=???, donde una caja debe tener una superficie en píxeles en el rango PBmin..PBmax para ser considerada como tal, por defecto 		

Identificador	CAP0086-1	Tipo	Capacidad
	<p>PBmin=??? y PBmax=???.</p> <p>Todos los valores Xmin..Xmax son enteros positivos, $0 \leq Xmin \leq Xmax$. Xmax indefinido equivale a no imponer ningún límite.</p> <p>Los archivos que no cumplan todos y cada uno de estos criterios son marcados en la lista como no útiles, y el lugar de origen es etiquetado como no útil.</p> <p>(Añadir al rol Administrador la capacidad de establecer estos parámetros.)</p>		
Estado	Propuesto		
Fuente	1.2-KI-00046		
Necesidad			
Prioridad	1		

2.3. Vectorizador

El paquete no contiene requisitos

3. Recuperador

Identificador	CAP0041-1	Tipo	Capacidad
Descripción breve	Búsquedas con sinónimos		
Creado por	Gonzalo Génova Fuster	Creación	28/10/2008 11:42:05
Modificado por	Gonzalo Génova Fuster	Modificación	30/10/2008 18:59:22
Descripción detallada	La fórmula de consulta obtenida gracias al asistente es completada con el uso de sinónimos y ontologías aplicados a los nombres de los elementos que forman parte de la consulta para obtener un abanico más amplio de resultados. (Tal vez contemplar también la traducción a otros idiomas.)		
Estado	Propuesto		
Fuente	George Andreadakis, Gonzalo Génova		
Necesidad			
Prioridad	2		

16 Appendix C: Definitions, acronyms and abbreviations

- ❖ **.NET:** Microsoft's Visual Studio .NET. Microsoft's set of software technologies for develop software and connecting information, people, systems and devices.
- ❖ **Add-on:** Add-on, also known as Add-in, is a mini program that runs in combination with the main application in order to extend and improve the functionality of that application
- ❖ **Artifact:** An artifact in the Unified Modelling Language (UML) is the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system.
- ❖ **AJAX:** Asynchronous JavaScript and XML), is a group of interrelated web development techniques used on the client-side to create interactive web applications or rich Internet applications.
- ❖ **Brochureware:** A brochureware website is a business website that has very infrequently updated content, and little of it. Often the site has been developed as a direct translation of existing printed promotional materials, hence the name.
- ❖ **CMS:** A content management system (*CMS*) is a system used to organize and facilitate collaborative content creation.
- ❖ **Crawler:** Component of a search engine that gathers listings by automatically trolling the Web and following links to Web pages (also called a spider or robot or bot). It makes copies of the Web pages found and stores them in the search engine's index.
- ❖ **Diagrammer:** Programs that allows a user to do diagramming. Diagramming software consists of computer programs that are used to produce graphical diagrams.
- ❖ **DOM:** The Document Object Model (DOM) is a cross-platform and language-independent convention for representing and interacting with objects in HTML.
- ❖ **EF:** Entity Framework.
- ❖ **EJB:** Enterprise JavaBeans (EJB) is a managed, server-side component architecture for modular construction of enterprise applications.
- ❖ **ESA:** The European Space Agency.
- ❖ **EXT:** Ext is a JavaScript library for building interactive web applications using techniques such as AJAX, DHTML and DOM scripting.
- ❖ **FTP:** File Transfer Protocol (FTP) is a standard network protocol used to exchange and manipulate files over an Internet Protocol computer network.
- ❖ **GPL:** The GNU General Public License (GNU GPL or simply GPL) is a widely used free software license, originally written by Richard Stallman for the GNU project.
- ❖ **GUI:** Graphical user interface: a user interface based on graphics (icons and pictures and menus) instead of text.
- ❖ **HTML:** Acronym for HyperText Markup Language is the predominant markup language for web pages.
- ❖ **IIS:** Microsoft's Internet Information Services.
- ❖ **IT:** Information technology: the branch of engineering that deals with the use of computers and telecommunications.
- ❖ **JPG/JPEG:** File format commonly used for image compression; Image file with that format.
- ❖ **JS:** Java Script. JavaScript is a scripting language used to enable programmatic access to objects within other applications.
- ❖ **JSON:** JavaScript Object Notation, is a lightweight computer data interchange format.

- ❖ **JSP:** Java Server Pages (JSP) is a Java technology that allows software developers to create dynamically generated web pages, with HTML, XML, or other document types, in response to a Web client request.
- ❖ **LGPL:** The GNU Lesser General Public License (formerly the GNU Library General Public License) or LGPL is a free software license published by the Free Software Foundation.
- ❖ **LINQ:** Language Integrated Query.
- ❖ **LOB:** Line-of-Business.
- ❖ **Model:** A representation of a complex entity or process.
- ❖ **MP3:** Acronym for MPEG layer 3, which is a compressed audio format.
- ❖ **MVC:** Model View Controller pattern.
- ❖ **MySQL:** Open source SQL (Structured Query Language) database.
- ❖ **NGINX:** Is a lightweight, high performance web server/reverse proxy and e-mail (IMAP/POP3) proxy, licensed under a BSD-like license.
- ❖ **ORM:** Object-Relation Mapping in computer software is a programming technique for converting data between incompatible type systems in relational databases and object-oriented programming languages.
- ❖ **OS:** Operative System.
- ❖ **PDF:** Portable Document Format (PDF) is a file format created by Adobe Systems in 1993 for document exchange.
- ❖ **PL/SQL:** Procedural Language/Structured Query Language is Oracle Corporation's proprietary procedural extension to the SQL database language.
- ❖ **Portlet:** Portlets are pluggable user interface software components that are managed and displayed in a web portal.
- ❖ **PHP:** PHP is a scripting language originally designed for producing dynamic web pages.
- ❖ **REST:** Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web.
- ❖ **RSS:** (Really Simple Syndication) is a family of Web feed formats used to publish frequently updated content such as blog entries, news headlines or podcasts.
- ❖ **SEO:** Search engine optimization, the use of various techniques to improve a web site's ranking in the search engines and thus attract more visitors.
- ❖ **SQL:** SQL (Structured Query Language) is a database computer language designed for the retrieval and management of data in relational database.
- ❖ **TDD:** Test Driven Development.
- ❖ **UI:** User Interface.
- ❖ **UML:** Unified Modelling Language (UML) is a standardized general-purpose modelling language in the field of software engineering.
- ❖ **URL:** Uniform Resource Locator (URL). The address of a web page on the World Wide Web.
- ❖ **VC-1:** Informal name of the SMPTE 421M video codec standard initially developed by Microsoft.
- ❖ **Virtual Assistance:** Virtual assistance (or VA) is the professional service of remote administrative office and other specialized support by an independent contractor called a virtual assistant who works with clients in an ongoing, collaborative professional relationship.
- ❖ **WCMS:** Web Content Management System.
- ❖ **WebDAV:** Web-based Distributed Authoring and Versioning, or WebDAV, is a set of extensions to the Hypertext Transfer Protocol (HTTP) that allows users to edit and manage files collaboratively on remote World Wide Web servers.
- ❖ **WMV/WMV7/WMV8:** Windows Media Video (WMV) is a compressed video file format for several proprietary codec's developed by Microsoft.
- ❖ **WS:** Web Service.

- ❖ **WYSIWYG**: What You See Is What You Get, used in computing to describe a system in which content during editing appears very similar to the final product.
- ❖ **XAML**: Extensible Application Markup Language is a declarative XML-based language created by Microsoft.
- ❖ **XMI**: The XML Metadata Interchange (XMI) is an Object Management Group (OMG) standard for exchanging metadata information via Extensible Markup Language (XML).
- ❖ **XML**: Extensible Markup Language is a general-purpose specification for creating custom markup languages.
- ❖ **XSLT**: Extensible Stylesheet Language Transformations (XSLT) is an XML-based language used for the transformation of XML documents into other XML.
- ❖ **ZIP**: The ZIP file format is a data compression and archive format.
- ❖ **ZODB**: Zope's built-in transactional object database.

Google *define* tool has been used for the definitions of the acronyms.

17 References

1. Ethier, K. and Scott, A.: Introduction to Structured Content Management with XML. CMS Watch. Retrieved on (November 2007).
2. Tivy, J.: The XML Content Management System for Document Centric XML. Bluestream Database Software Corporation. Retrieved on (November 2007).
3. Faithfull, W.: Using XSLT to Make Websites. <http://woric.net>. Retrieved on 2008-02-27.
4. Content management with Plone: http://www.456bereastreet.com/archive/200503/content_management_with_plone/. Retrieved on 2008-03-10.
5. Plain Black Corporation: Content Management System Matrix. <http://www.cmsmatrix.org/>. Retrieved on 2008-02-27.
6. ORACLE Technology Network: <http://www.oracle.com/technology/products/ias/portal/documentation.html>. Retrieved on 2008-02-28.
7. The Plone Foundation: http://plone.org/documentation/manual/definitive-guide/definitive_guide_to_plone.pdf. Retrieved on 2008-02-28.
8. NASA: Object Oriented Data Technology. Data grid framework for transparent access to disparate resources. <http://oodt.jpl.nasa.gov/better-web-app.mov>. Retrieved on 2008-02-28.
9. Introduction to ASP.NET and Web Forms: <http://msdn.microsoft.com/en-us/library/ms973868.aspx>. Retrieved on 2008-11-31.
10. What is ASO.NET MVC?: <http://www.asp.net/mvc/whatisaspmvc/>. Retrieved on 2008-11-31.
11. Microsoft Silverlight: http://en.wikipedia.org/wiki/Microsoft_Silverlight. Retrieved on 2008-11-31.
12. Stephen Walther: <http://stephenwalther.com/blog/>. Retrieved on 2008-12-15.
13. Learn ASP.NET MVC: Creating the Controller and the View: <http://www.asp.net/Learn/mvc-videos/video-422.aspx>. Retrieved on 2008-12-15.
14. Silverlight Hack: [http://silverlighthack.com/post/2008/10/23/Silverlight-ASPNET-MVC-vs-Web-Forms-\(Very-High-Level\).aspx](http://silverlighthack.com/post/2008/10/23/Silverlight-ASPNET-MVC-vs-Web-Forms-(Very-High-Level).aspx). Retrieved on 2008-12-15.
15. [http://msdn.microsoft.com/en-us/library/aa697427\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/aa697427(VS.80).aspx). Retrieved on 2008-12-23.
16. Frederick, S., Ramsay, C. and Blades, S.: Learning Ext Js. Build dynamic, desktop-style user interfaces for your data-drive web applications. Packt Publishing (2008).
17. Glossary Network Management, MIBs and MPLS: Principles, Design and Implementation: <http://authors.phptr.com/morris/glossary.html>. Retrieved on 2008-12-25.
18. Create and share simple UML diagrams: <http://yuml.me/>. Retrieved on 2009-06-15.
19. <http://msdn.microsoft.com/en-us/library/aa478948.aspx>. Retrieved on 2008-12-13.
20. Agarwal, V. V., Huddleston, J., Raghuram., R., Fahad Gilani, S. and Reid, J.: Beginning C# 2008 Databases. From Novice to Professional. What every C# programmer needs to know about SQL Server 2005, T-SQL, ADO.NET 3.5, and LINQ.Apress (2008).

21. <http://p2p.wrox.com/ado-net/51201-datasets-advantages-disadvantages.html>
22. Advantages and disadvantages of creating a DataSet
<http://forums.asp.net/p/982658/1259802.aspx>. Retrieved on 2008-12-27.
23. http://www.elguille.info/colabora/NET2005/FernandoLuque_DataSetsII.htm.
Retrieved on 2008-12-27.
24. Typed or not Typed DataSets:
<http://blog.domitienda.com/index.php/2009/06/17/datasets-si-o-no/>. Retrieved on
2008-12-27.
25. When to use ADO.NET EF vs. Linq to SQL
<http://www.davidhayden.com/blog/dave/archive/2007/04/29/ADONETEntityFrameworkWhenToUseItVsLinqToSQL.aspx>. Retrieved on 2008-12-27.
26. Efficient Pagination: <http://geeks.ms/blogs/gperez/archive/2009/03/02/paginando-eficientemente-en-asp-net.aspx>. Retrieved on 2008-12-25.
27. http://en.wikipedia.org/wiki/Language_Integrated_Query
28. <http://es.wikipedia.org/wiki/LINQ>
29. http://cs.rthand.com/blogs/blog_with_righthand/archive/2007/10/06/LINQ-to-SQL-showstoppers.aspx
30. <http://dotnet.org.za/hiltong/archive/2008/02/01/why-use-linq-to-sql-part-1-performance-considerations.aspx>
31. Advantages of LINQ : <http://forums.asp.net/t/1237037.aspx>
32. Linq and Linq to SQL: <http://blog.tercerplaneta.com/2008/02/linq-y-linq-para-sql.html>
33. Price SQL Server 2005: <http://www.microsoft.com/sqlserver/2005/en/us/pricing.aspx>.
Retrieved on 2009-06-28.
34. Price Windows Server 2008:
<http://www.microsoft.com/windowsserver2008/en/us/pricing.aspx>. Retrieved on
2009-06-28.
35. Price Visual Studio: <http://www.microsoft.com/visualstudio/en-us/howtobuy/default.mspx>. Retrieved on 2009-06-28.
36. Price Camtasia: <http://www.techsmith.com/camtasia.asp?cmpn=nf01> . Retrieved on
2009-06-28.
37. N-Tier Architecture: http://www.webopedia.com/quick_ref/app.arch.asp. Retrieved
on 2009-01-20.
38. 50 Useful JavaScript Tools <http://www.smashingmagazine.com/2009/02/08/50-extremely-useful-javascript-tools/>. Retrieved on 2008-11-02.
39. 10 Principles of Effective Web Design:
<http://www.smashingmagazine.com/2008/01/31/10-principles-of-effective-web-design/>. Retrieved on 2009-01-23.
40. 10 Usability Nightmares you should be aware of:
<http://www.smashingmagazine.com/2007/09/27/10-usability-nightmares-you-should-be-aware-of/>. Retrieved on 2009-01-23.
41. 30 Usability Issues to be Aware of:
<http://www.smashingmagazine.com/2007/10/09/30-usability-issues-to-be-aware-of/>.
Retrieved on 2009-01-23.

42. Usability and interface design books:
<http://www.smashingmagazine.com/2008/01/24/usability-and-interface-design-books/>. Retrieved on 2009-01-23.
43. Google Ajax Play Nice: <http://www.softwaredeveloper.com/features/google-ajax-play-nice-061907/>. Retrieved on 2009-01-26.
44. Web Master Principles:
<http://www.google.com/support/webmasters/bin/answer.py?answer=40052&query=cloaking&topic=&type=>. Retrieved on 2009-01-26.
45. Google and dynamic javascripted content:
<http://www.brainhandles.com/2007/03/11/does-google-index-dynamic-javascripted-content/>. Retrieved on 2009-01-26.
46. Privacy: <http://google.dirson.com/post/3879-direccion-ip-dato-personal/>. Retrieved on 2008-12-16.
47. ADO.NET: <http://es.wikipedia.org/wiki/ADO.NET>. Retrieved on 2009-01-08.
48. Data Sets: <http://en.wikipedia.org/wiki/ADO.NET#DataSets>. Retrieved on 2008-12-13.
49. The ADO.NET Entity Framework Overview: [http://msdn.microsoft.com/en-us/library/aa697427\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/aa697427(VS.80).aspx). Retrieved on 2009-01-08.
50. Mitchell, S., Anders, B., Howard, R., Seven, D., Walther, S., Wille, C. and Wolthuis, D.: ASP.NET: Tips, Tutorials, and Code. SAMS, Indiana (2001).
51. Jacobson, I., Booch, G. and Rumbaugh, J.: The Unified Modelling Language User Guide. Addison Wesley (1998).
52. Nagel, C., Evjen, B., Glynn, J., Watson and K., Skinner, M.: Professional C # 2008. Inc. Wiley Publishing (2008).
53. Randolph, N. and Gardner, D.: Professional Visual Studio 2008. Inc. Wiley Publishing (2008).
54. ASP.NET MVC vs. ASP.NET Web Forms:
<http://weblogs.asp.net/shijuvarghese/archive/2008/07/09/asp-net-mvc-vs-asp-net-web-form.aspx>
55. Adecuación de sitios web a la LOPD: <http://www.consultores-ebusiness.com/index.php/lopd/164-adequacion-de-sitios-web-a-la-lopd.html>
56. Computer Security: http://en.wikipedia.org/wiki/Computer_security
57. Ley Orgánica de Protección de Datos de Carácter Personal de España:
http://es.wikipedia.org/wiki/Ley_Org%C3%A1nica_de_Protecci%C3%B3n_de_Datos_de_Car%C3%A1cter_Personal_de_Espa%C3%B1a
58. Aspectos relativos al fichero con datos de carácter personal: <http://www.guia-empresas.net/documento-seguridad-lopd4.php>

[End of the document]