Chapter 4

A DESIGN TOOLKIT FOR HYPERMEDIA APPLICATIONS BASED ON ARIADNE DEVELOPMENT METHOD

Susana Montero, Camino Fernández, Juan M. Dodero, Ignacio Aedo, and Paloma Díaz
Universidad Carlos III de Madrid, Departamento de Informática, Laboratorio DEI, Avda. de la Universidad, 30 – E-28911 Leganés (Spain)
E-mail: {smontero@inf, camino@inf, dodero@inf, aedo@ia, pdp@inf}.uc3m.es
URL: http://www.dei.inf.uc3m.es/~smontero/
Tel: +34 91 624 9419 – Fax: +34 91 624 9129

Abstract The development process of hypermedia applications implies very specific problems mainly related, first, to the use of navigational structures, interactive behaviours and multimedia compositions, and second, to the fact that are used by users with different levels of knowledge and skills, and also different security levels. This paper presents a design environment, AriadneTool, that allows a designer to model a hypermedia application, to validate such a design and to generate dynamically XML + SMIL implementation templates. This environment is based upon the Ariadne method which offers a set of integrated activities to model hypermedia applications in a systematic and iterative way. A practical example is shown to illustrate the use of the tool in the development of a hypermedia application.

Keywords: Development method, Design environment, Hypermedia applications, Hypermedia modelling.

1. INTRODUCTION

The extremely high speed of demand on hypermedia systems, and in particular on web applications, have led to development processes where almost the only existing phase is coding, mainly using tools as NetObjects' Fusion or DreamWeaver with no method behind. But hypermedia development presents very specific problems [6,10,14] that can be summarised in: mechanisms to model sophisticated navigational structures, interactive behaviours and multimedia compositions which have to be usable and harmonic at the same time.

Traditional design methods lack intellectual mechanisms to analyse a design using abstractions and design entities related to the hypermedia domain (e.g., nodes, links, anchors and synchronisms). Some hypermedia methods provide designers with hypermedia specification tools including HDM [10], RMM [12] and OOHDM [14], but the key point of taking into account security issues is not addressed.

The proposal of this paper is the use of a graphical toolkit called AriadneTool, to support the development of hypermedia applications following a development method called Ariadne. The method applies an iterative process based on the evaluation of design solutions with potential users to determine their utility and usability. Compared to similar methods, Ariadne improves three significant aspects: it provides a mechanism to define time- and space-based constraints among contents; it offers a product to model the users structure based on roles and teams that can be used to support personalization and security; and it includes a security model to define the policy that will be applied during the hyperdocument operation.

2. INTRODUCING ARIADNE AND ARIADNETOOL

Ariadne proposes a systematic approach to produce hypermedia applications. Compared to other hypermedia design methods such as HDM, OOHDM or RMM, Ariadne shares a number of similarities concerning the specification of the logical structure, navigation and interface layouts but it also improves or introduces mechanisms to deal in an integrated way with the six complementary views which have to be properly addressed by a development method [6], including navigation, presentation, control, security, processes and data. Several hypermedia methods and theirs CASE tools have been proposed including Autoweb [9], WebML [3] and OO-H [11]. However, these methods have the following weak points:

- Validation and integrity rules to test the correctness, completeness and integrity of the design.
- Contents modelling to organise and harmonise multimedia contents both in their temporal and spatial dimension.
- User modelling to model different types of application users and to apply personalisation as well as security constraints.
- Security modelling to model which contents should be delivered to which users, who can modify or personalise items and which constraints have to be applied.

- Evaluation stage to collect information about the potential usability of a system to improve features and functionality of application interface.

AriadneTool supports the Ariadne development method addressing these features as it is shown in the following sections. In particular, the main contributions of Ariadne are: the introduction of a specific mechanism to establish space and time-based relationships among contents, called alignments and synchronisations respectively; the inclusion of elements to define a users structure that can be used not only for security purposes (e.g., to define adaptive accesses); and, finally, the inclusion of a high-level security model which makes possible to specify a role-based security policy, simpler and easier to maintain than group-based policies [8], using the same design entities than those used to specify other hypermedia features such as the hypertext structure.

This method proposes a development process consisting of three phases: Conceptual Design, Detailed Design and Evaluation. Each of them generates a number of products covering the six views aforementioned Ariadne method is graphically described in Fig. 1. Arrows represent a sequence which is not unique but has been shown very helpful in different developments. Ariadne assumes an iterative process where the evaluation of prototypes is used to gather information to improve the design, whether conceptual or detailed. A detailed explanation of the method can be found in [4] and [6].

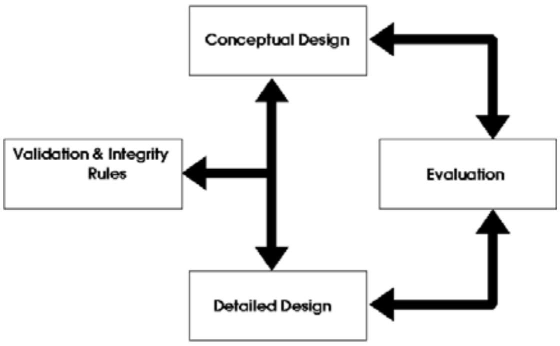
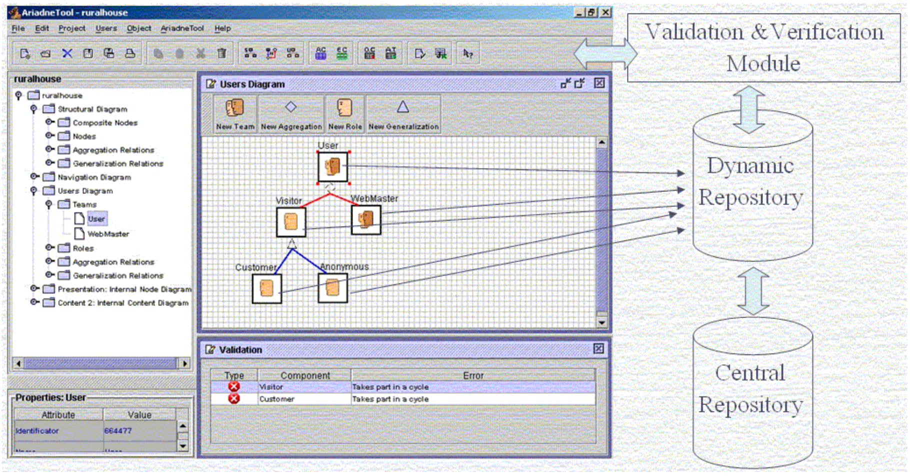


Figure 1. The Ariadne method.

AriadneTool is an environment devoted to the development of hypermedia applications based upon the above design process. The main components to support such mechanisms are (Fig. 2): (1) the `Front-End` that provides a perfect environment for elaborating the Ariadne Method products; (2) the `Validation and Verification Module` that holds the rules to validate and verify the completeness and correctness of the design, notifying any mistake or warning to the designer; (3) the `Dynamic Repository` that holds the components of the front-end in dynamic memory, so that access is faster; and (4) the `Central Repository` that holds the components in a persistent way. The development environment is implemented using SDK1.4, allowing us to obtain an independent operation platform. The Validation and Verification Module is represented by DTD documents in which the rules defining well-formed design products are specified. Elements designed in the different products of the method are stored in the Central Repository by JAXB1.1 which allows an automatic two-way mapping between Java objects stored in the Dynamic Repository and final XML documents. A module to translate synchronisation operations to SMIL specifications has also been included.



Type	Component	Error
✗	Visitor	Takes part in a cycle
✗	Customer	Takes part in a cycle

Attribute	Value
Identificator	664477

Figure 2. The Ariadne architecture.

3. ARIADNE AND ARIADNETOOL WORKING TOGETHER

In this section, a practical example is used to guide the application of the method with the use of the toolkit. The Ariadne Development Method establishes a systematic process composed by three phases: conceptual design, detailed design and evaluation. Each of these phases proposes a number of design products to specify and produce hypermedia and web applications. The conceptual design is focused on identifying abstract types of components, relationships and functions. The detailed design is concerned with specifying in a detailed way the system features, processes and behaviours the application should be generated with. Finally, the evaluation is concerned with using prototypes and specifications to assess the system usability. The method also provides a number of validation and integrity rules, both in the intra and inter phase level to check completeness, consistency and integrity among the various design products. These methodological foundations are the basis for AriadneTool. At this moment, the tool supports the design products of the conceptual design phase and the validation and integrity rules.

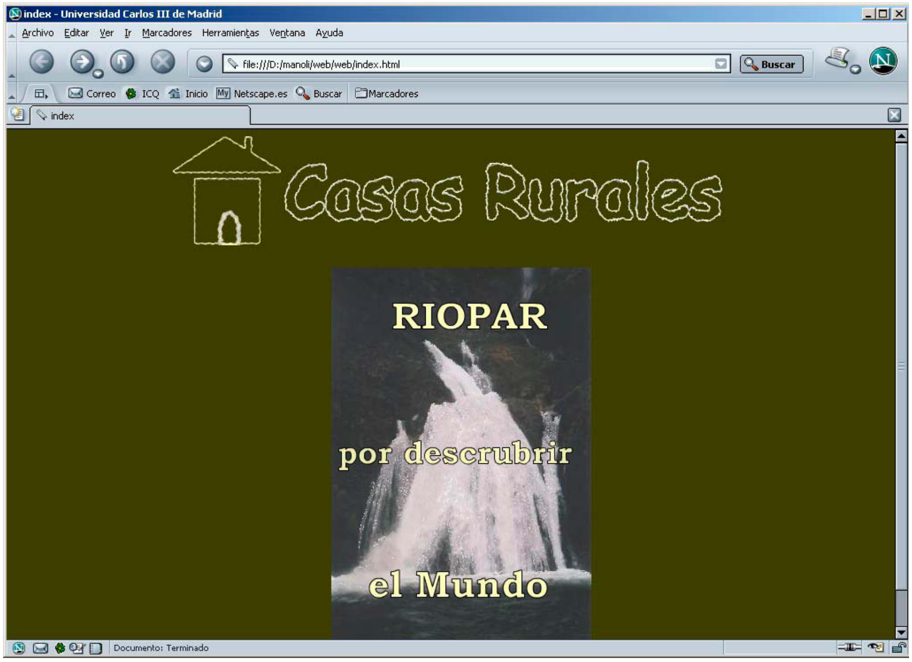

index - Universidad Carlos III de Madrid
Archivo Editar Ver Ir Marcadores Herramientas Ventana Ayuda
file:///D:/manoli/web/web/index.html
Buscar
Correo ICQ Inicio Netscape.es Buscar Marcadores
index
Casas Rurales
RIOPAR
por descrubrir
el Mundo
Documento: Terminado

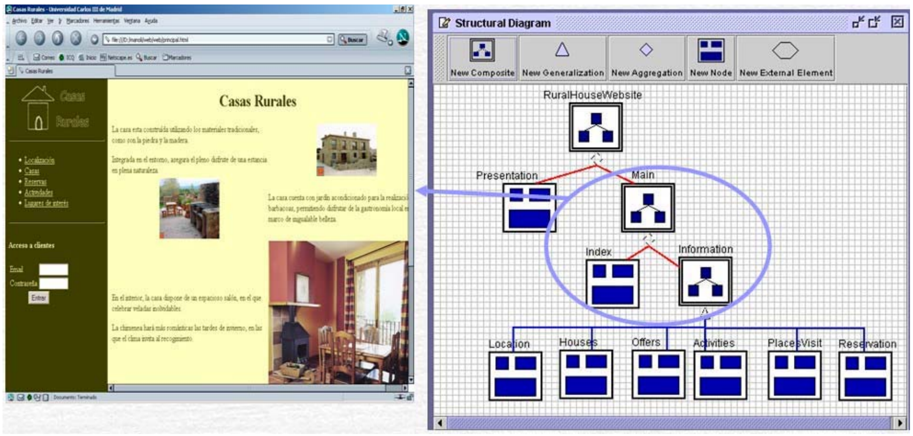
Figure 3. Example: Rural houses website.

The example shown in Fig. 3 illustrates a website about rural houses oriented towards providing information about houses, their equipments, booking and prices as well as places to visit or activities offered. The site will be accessed by different users that can be identified as, for instance, visitors who are anonymous users that browse the web-site, customers who are users that have booked some time a house so they can check their reservations and will be also notified of special offers, and the web-master who is the person in charge of administrating the site. Navigation tools should be offered to access all this information to all these users.

Taking this example site as an starting point, we will get through the phases of Ariadne method showing how to generate the products proposed by the method in the conceptual design phase: the structural diagram, the navigation diagram, the attributes and events catalogue, the internal diagram – including the spatial diagram and the timeline –, the users diagram, the categorisation catalogue and the validation and verification, which are the products included in AriadneTool.

3.1 Structural Diagram

In this diagram, the structural relationships that appear in the application domain are defined by means of composite nodes which are connected to their components (simple or composite) by means of two abstraction mechanisms: aggregation, that refer to a set of nodes as a whole, and generalisation, that represents an inclusion relation involving inheritance mechanisms. The structure of our example (Fig. 4) is defined as an aggregation of a *presentation* node, which is the entry point to the website (Fig. 3), and *main* composite which aggregates the *index* and the *information* nodes. The aim of this structure is to represent the fact that the website is made up of two frames. One frame holds the index node and the other one holds the information node. This last node generalises different kinds of information about what the website offers to visitors, such as how to get there or the features of the houses.



La casa esta construida utilizando los materiales tradicionales, como son la piedra y la madera.

Integrada en el entorno, asegura el pleno disfrute de una estancia en plena naturaleza.

La casa cuenta con jardín acondicionado para la realizació barbacoas, permitiendo disfrutar de la gastronomía local e marco de inigualable belleza.

En el interior, la casa dispone de un espacioso salón, en el que celebrar veladas inolvidables.

La chimenea hará más románticas las tardes de invierno, en las que el clima invita al recogimiento.

Figure 4. Structural diagram.

3.2 Navigation Diagram

The navigation diagram is where the navigation paths and tools the website offers to the users are specified. Navigation paths are settled among nodes using tagged links which can be uni- or bi-directional and n-ary [5]. For example in Fig. 5, from *presentation* node we browse both *index* and *houses* node. Since *information* is a generalisation composite, all its component (such as *houses*, *location*, and so on) will inherit the link *information* and, therefore, the *index* node has turned into a navigation tool, tagged with an NT. Moreover, we can define other navigation paths to external nodes of our application as, for instance, the payment link, used when paying our booking, to access the payment gateway, verify our visa number card and go back again to *reservation* node.

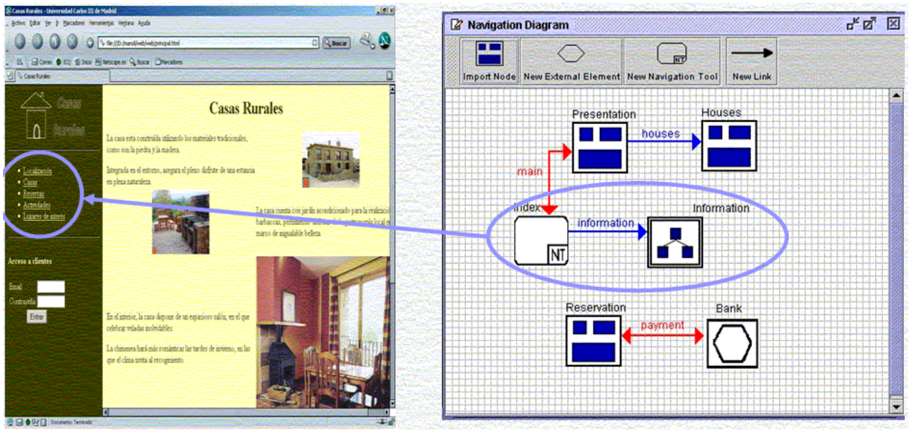
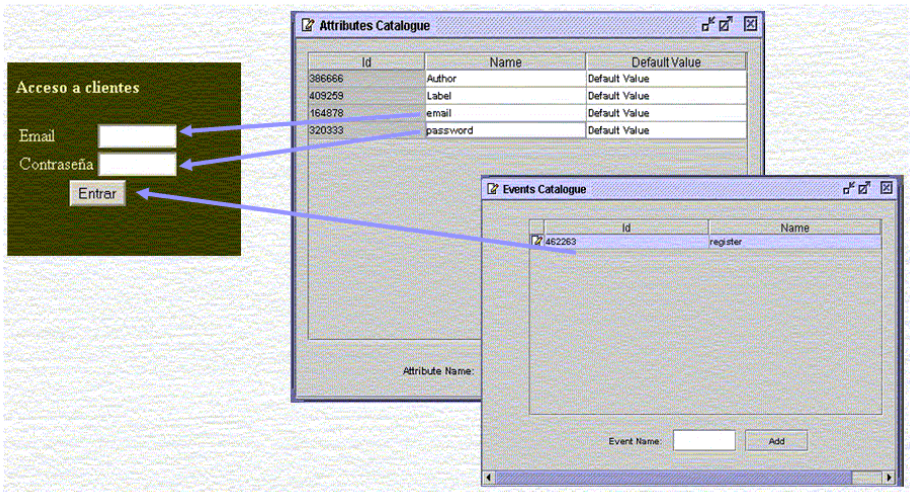


Figure 5. Navigation diagram.

3.3 Attributes and Events Catalogue

The attributes and the events catalogue are repositories where properties and behaviours are held so that they can be reused. The attributes catalogue collects properties which add semantic to modelling. For each attribute there is an entry including its name and its default value, so they can be associated to nodes, contents and links and the value can be overwritten. In the case of the events catalogue, they collect actions that take place when a condition is fulfilled. The same way, there is an entry for each event and they can be associated to nodes or contents.



Id	Name	Default Value
386666	Author	Default Value
409259	Label	Default Value
164878	email	Default Value
320333	password	Default Value

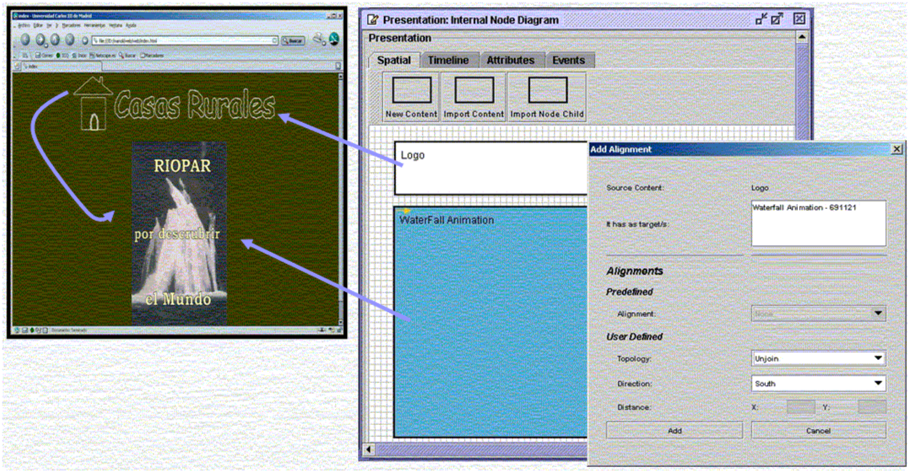
Id	Name
462263	register

Figure 6. Attributes and Events Catalogue.

For example in Fig. 6, we could have two attributes associated to the email and password contents with the same name that holds the values typed by the user. These attributes could be used in an event that checks the values and send them to the server.

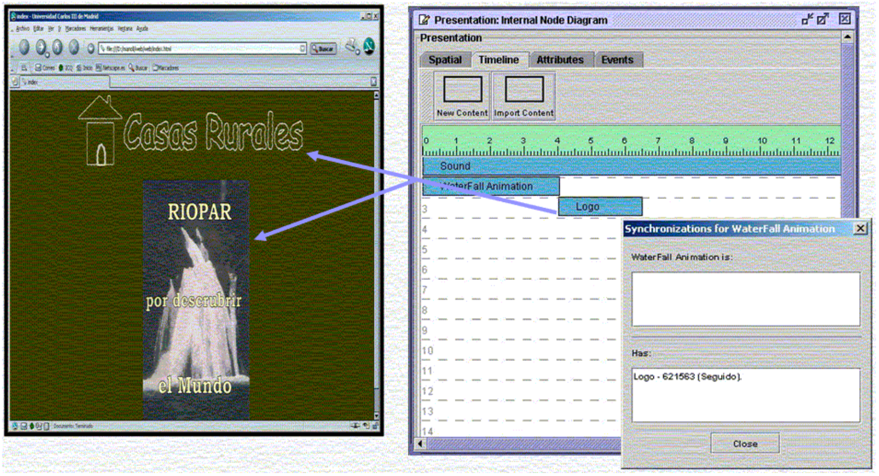
3.4 Internal Diagram - Spatial Diagram and Timeline

In order to provide more information about the nodes defined in the structural and navigation diagrams, an internal diagram is created for each node. This diagram consists of two subdiagrams; the spatial diagram and the timeline, where contents, anchors, attributes and events can be placed. The spatial diagram is a two dimensional space that represents the node visualisation area where contents are placed.



Source Content: Logo
It has as target/s: Waterfall Animation - 691121
Alignments
Predefined
Alignment: None
User Defined
Topology: Unjoin
Direction: South
Distance: X: Y:
Add Cancel

Figure 7. Spatial diagram.



WaterFall Animation is:
Has: Logo - 621563 (Seguido).
Close

Figure 8. Timeline diagram.

In Fig. 7, our presentation node has two contents, the house logo and a waterfall animation. The waterfall animation content is coloured in blue because it acts as source anchor for information and index links. To create more harmonic presentations, space relationships can be set among contents. For example, the waterfall content will be always below the logo content. The other internal diagram is the timeline, which represents how the node evolves throughout a time interval. In Fig. 8, a sound is added which will be played during the time this node is being browsed. Moreover, to create more dynamic presentations, time relationships can be set among contents. In this case we have decided that the waterfall animation content is shown in the first place, and at the end of the animation the logo content appears.

3.5 Users Diagram

In the users diagram, the expected types of users of the application are identified inside roles and teams, with no individual users [2]. A role represents a job function or responsibility such as our visitors or the webmaster. Roles can be specified as hierarchical structures where roles are specialised into more specific roles by means of generalisation relationships. Fig. 9 shows how *visitor* role is specialised into *customers* and *anonymous* visitors. Roles can be grouped into teams that work together by means of aggregation relationships. For instance, all application users are grouped into *user* team. The users diagram makes possible to define user-dependent presentations as well as personalised hyperdocuments. For instance, only users that belong to the customer role will have access to offers node (Fig. 9).

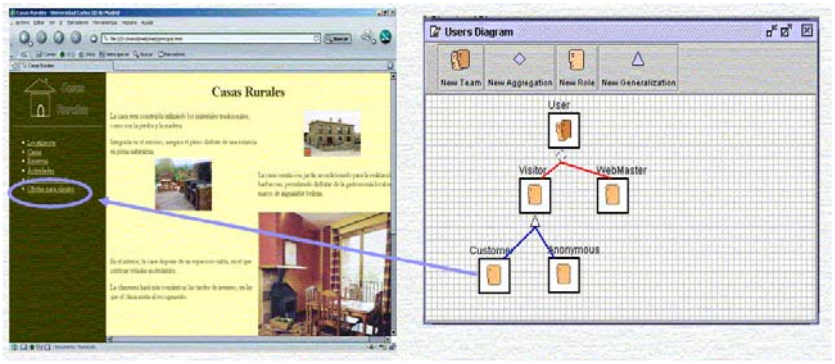
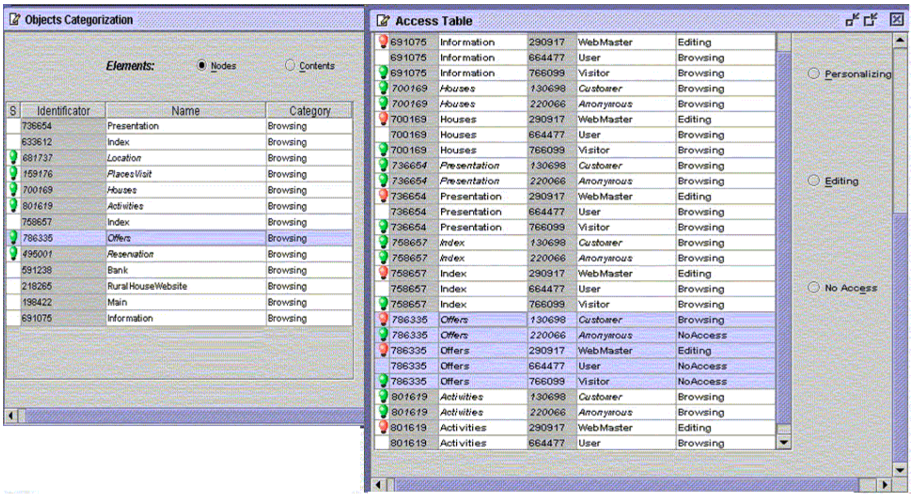


Figure 9. Users diagram.

3.6 Categorisation Catalogue and Access Table

The categorisation catalogue and the access table define who can do what in the system. The categorisation catalogue holds the security category assigned to each node or content defining the most permissive operation to be performed by a user. These categories are: browsing, personalising and editing following the security model described in [2] and [7]. In this case the category for all nodes and contents is browsing, that is all nodes can be read but not modified. The access table allows the definition of the security policies following a RBAC (Role Based Access Control) approach, assigning a manipulation category to each role for each node and content. For example (Fig. 10), the offer node can be only accessed by customers, so the rest of the users will have a no access category, except the webmaster role that needs to edit the page.



Elements: [✓ Nodes] Contents

S	Identificator	Name	Category
	736654	Presentation	Browsing
	633612	Index	Browsing
🟢	681737	Location	Browsing
🟢	159176	PlacesVisit	Browsing
🟢	700169	Houses	Browsing
🟢	801619	Activities	Browsing
	758657	Index	Browsing
🟢	786335	Offers	Browsing
🟢	495001	Reservation	Browsing
	591238	Bank	Browsing
	218265	RuralHouseWebsite	Browsing
	198422	Main	Browsing
	691075	Information	Browsing

🔴	691075	Information	290917	WebMaster	Editing
	691075	Information	664477	User	Browsing
🟢	691075	Information	766099	Visitor	Browsing
🟢	700169	Houses	130698	Customer	Browsing
🟢	700169	Houses	220066	Anonymous	Browsing
🔴	700169	Houses	290917	WebMaster	Editing
	700169	Houses	664477	User	Browsing
🟢	700169	Houses	766099	Visitor	Browsing
🟢	736654	Presentation	130698	Customer	Browsing
🟢	736654	Presentation	220066	Anonymous	Browsing
🔴	736654	Presentation	290917	WebMaster	Editing
	736654	Presentation	664477	User	Browsing
🟢	736654	Presentation	766099	Visitor	Browsing
🟢	758657	Index	130698	Customer	Browsing
🟢	758657	Index	220066	Anonymous	Browsing
🔴	758657	Index	290917	WebMaster	Editing
	758657	Index	664477	User	Browsing
🟢	758657	Index	766099	Visitor	Browsing
🔴	786335	Offers	130698	Customer	Browsing
🟢	786335	Offers	220066	Anonymous	NoAccess
🔴	786335	Offers	290917	WebMaster	Editing
	786335	Offers	664477	User	NoAccess
🟢	786335	Offers	766099	Visitor	NoAccess
🟢	801619	Activities	130698	Customer	Browsing
🟢	801619	Activities	220066	Anonymous	Browsing
🔴	801619	Activities	290917	WebMaster	Editing
	801619	Activities	664477	User	Browsing

Figure 10. Categorisation Catalogue and the Access Table.

3.7 Validation and Verification

The tool also checks by means of a number of inter- and intra validation rules if each entity is fully and correctly described. For example, if visitor is specialised into a customer, at the same time that customer is specialised into a visitor, the inheritance mechanisms would always be applied. Therefore the designer would be notified of this misconception. The validation and verification module is in charge of checking the completeness and consistency of the modelling entities by means of ontologies. The elements stored in the dynamic repository are turned into an instance of an ontology that conceptualise a hypermedia application domain by means of an explicit number of elements, properties and axioms. After that, the instance is introduced into the Jess rule engine and several rules that check possible errors are run over the model and the results presented to the user. For example this rule checks that a team has to be made up of at least one role.

```
(defrule team-whithout-roles
(declare (salience -150))
(PropertyValue http://www.w3.org/1999/02/22-rdf-syntax-ns#type
  ?t http://ariadne/validator/laby#Team)
(not (PropertyValue http://ariadne/validator/laby#Valid
  ?t http://ariadne/validator/laby#Team))
=>
(assert (PropertyValue http://ariadne/validator/laby#Error
  ?t It/has/no/relation)))
```

4. CONCLUSION

AriadneTool is a design toolkit devoted to supporting the Ariadne development method. The main contributions of this method are, on the one hand, the multimedia modelling, offering mechanisms to organise and harmonise multimedia contents in different dimensions such as time and space, and, on the other hand, the user modelling that identifies the expected types of users of the application making it possible to define user-dependent presentations. Moreover, that modelling is used to define access policies in order to specify who can do what into the system.

With regard to the tool, the main contributions are, on the one hand, the validation and verification rules that help the designer to check completeness and consistency of its work, on the other hand, the automatic generation of documentation about the current design project, and, finally, since the conceptual modelling is serialised into XML format, using XML transformation language prototypes can be generated in HTML, SMIL or RDF.

AriadneTool has been evaluated by collecting feedback from the students in a course on Hypermedia Design. This feedback has been very positive, and a large majority believed that the tool supported well the method tasks and made much easier to use the Ariadne Method. The tool has also been used in a research project named ARCE, a web based system envisaged to cope with the lack of synchronism among assistance requests and responses in a multinational environment, whose main goal is to offer an efficient and reliable communication channel among the different agents involved in a disaster mitigation procedure [1].

Future work includes the support of the activities of the detailed design related to the specification of concrete elements from the abstract entities defined in the conceptual design. From this specification, prototypes will be generated in SMIL, XML and HTML. Moreover, a web design patterns repository will be integrated. Those web design patterns have already been formalised using ontologies for the designer to apply the pattern into the modelling phase using a task domain ontology. Finally, the ontology used in the validation and verification module is being used to generate semantic hypermedia applications that are made up of a domain ontology that describes the application domain in RDFS and its web resources using RDF.

ACKNOWLEDGMENTS

We'd like to thank Jose Ángel Cruz and Juan Francisco Arévalo for their cooperation in the development of AriadneTool. This toolkit is part of two projects funded by "Dirección General de Investigación del Ministerio de Ciencia y Tecnología" (TIC2000-0402) and "CAM" 07T/0024/2003.

REFERENCES

[1] Aedo, I., Díaz, P., Fernández, C., and Castro, J., *Supporting Efficient Multinational Disaster Response through a Web-Based System*, in K. Lenk, R. Traunmüller (eds.), Proc. of the 1st Int. Conf. on Electronic Government EGOV'2002 (Aix-en-Provence, 2-5 September 2002), LNCS, Vol. 2456, Springer-Verlag, Berlin, 2002, pp. 215-222.

[2] Aedo, I., Díaz, P., and Montero, S., *A Methodological Approach for Hypermedia Security Modelling*, Information and Software Tech., Vol. 45, No. 5, 2003, pp. 229-239.

[3] Ceri, S., Fraternali, P., and Bongio, A., *Web Modeling Language (WebML): a Modeling Language for Designing Web Sites*, in Proceedings of 9th International World Wide Web Conference WWW9 (Amsterdam, 15-19 May 2000), Computer Networks, Vol. 33, No. 1-6, 2000, pp. 137-157, accessible at http://www9.org/w9cdrom/177/177.html

[4] Díaz, P., Aedo, I., and Montero, S., Ariadne, A Development Method for Hypermedia, in H.C. Mayr, J. Lazanský, G. Quirchmayr, P. Vogel (eds.), Proceedings of 12th International Conference on Database and Expert Systems Applications DEXA'2001 (Munich, 3-5 September 2001), LNCS, Vol. 2113, Springer-Verlag, Berlin, 2001, pp. 764-774.

[5] Díaz, P., Aedo, I., and Panetsos, F., Labyrinth, *An Abstract Model for Hypermedia Applications. Description of its Static Components*, Information Systems, Vol. 22, No. 8, 1997, pp. 447-464.

[6] Díaz, P., Aedo, I., and Panetsos, F., *A Methodological Framework for the Conceptual Design of Hypermedia Systems*, in Proc. of the 5th Conf. on "Hypertexts and Hypermedia: Products, Tools and Methods" H2PTM'99 (Paris, Sept. 1999), 1999, pp. 213-228.

[7] Díaz, P., Aedo, I., and Panetsos, F., *Modelling Security Policies in Hypermedia and Web-Based Applications*, in S. Murugesan, Y. Deshpande (eds.), Proceedings of Conference on Software Engineering and Web Application Development Web Engineering'2001, LNCS, Vol. 2016, Springer-Verlag, Berlin, 2001, pp. 90-104.

[8] Ferraiolo, D.F., Barkley, J.F., and Kuhn, D.R., *A Role-based Access Control Model and Reference Implementation within a Corporate Intranet*, ACM Trans. on Information and Systems Security, Vol. 2, No. 1, 1999, pp. 34-64.

[9] Fraternali, P. and Paolini, P., *Model-Driven Development of Web Applications: The AutoWeb System*, ACM Transactions on Office Information Systems, Vol. 18, No. 4, 2000, pp. 323-282.

[10] Garzotto, F., Paolini, P., and Schwabe, D., *HDM-A Model-Based Approach to Hypertext Application Design*, ACM Trans. on Office Inf. Systems, Vol. 11, No. 1, 1993, pp. 1-26.

[11] Gómez, J., Cachero, C., and Pastor, O., *Conceptual Modelling of Device Independent Web Applications,* IEEE Multimedia, Vol. 8, No. 2, 2001, pp. 26-39.

[12] Isakowitz, T., Stohr, E.A., and Balasubramanian, P., *RMM: A Methodology for Structured Hypermedia Design*, Comm. of the ACM, Vol. 38, No. 8, 1995, pp. 34-44.

[13] Lowe, D. and Hall, W., *Hypermedia and the Web: An Engineering Approach*, John Wiley & Sons, New York, 1999.

[14] Schwabe, D. and Rossi, G., *Developing Hypermedia Applications Using OOHDM*, in Proc. of the Workshop on Hypermedia Development Processes, Methods and Models during the 9th ACM Conf. Hypertext'98 (Pittsburgh, 20-24 June 1998).