

# Fuzzy adaptation for information access to digital libraries

Camino Fernández    Ignacio Aedo    Paloma Díaz  
Departamento de Informática  
Universidad Carlos III de Madrid  
*e-mail: {camino@inf,aedo@ia,pdp@inf}.uc3m.es*

Vicente Matellán  
Departamento de Ciencias Experimentales e Ingeniería  
Universidad Rey Juan Carlos  
*e-mail: vmo@gsync.escet.urjc.es*

## Abstract

Different models have been proposed to build systems that adapt to the user reducing the complexity in the information browsing process, saving user time and offering personalised interfaces. The work described in this paper, named WAY-Z39.50, is an example of adaptive system specially built for accessing digital libraries. A fuzzy decision mechanism is used to implement the adaptation of both the user interface and the information searching process. This paper describes how this system has been designed and implemented, focusing on its fuzzy-based components. Finally, it also briefly discusses the advantages of using fuzzy logic to implement adaptability of the user interface and the searching process.

## 1 Introduction

This paper presents a prototype named WAY-Z39.50 for adaptive access to digital libraries. This system faces two major problems: finding, and accessing information. These are complex problems due both to the inherent difficulty about locating appropriate information servers and the information itself [5]; and to the complexity about presenting the information to the user through an adequate interface [3, 16].

The most promising approach to make easier the access to information is the use of *adaptive* systems, that is, systems that adapt to users automatically, based on their assumptions about them. Adaptiveness can be achieved by different means. For instance, “user models” [4] can provide distinct interfaces to different users by detecting if the user belongs to one group or to a different one. Another way is

to create systems able to modify themselves according to user actions without a preconfigured classification of users.

The last kind of systems need to be continuously monitoring the user actions, and modifying its own parallel actions. The majority of this kind of systems use the paradigm of “agent” to implement these monitoring processes. Most of the projects that provide adaptivity to information access systems have been devoted to web browsing assistants [12] as Letizia [10], FAB[1], or Butterfly [11]; and they are just demonstration systems. Few works have been done in the development of general architectures that help in the design, and implementation of this kind of systems.

In order to provide a general framework for the construction of adaptive information access systems, WAY general model has been proposed. WAY is an architecture model for the construction of adaptive information access systems [8] based on the use of a set of intelligent agents whose main goal is to make easier both the searching and the user interaction processes. In the next section a brief introduction to this model is shown.

WAY-Z39.50 is a particular implementation of this general model for accessing digital libraries. The paper describes how this system has been designed and implemented, focusing in its fuzzy-based components. It is organised as follows: section 2 describes WAY-Z39.50 architecture which is based in WAY model. Section 3 is devoted to the design and implementation of WAY-Z39.50 system focusing in the fuzzy issues, section 4 presents a session with the system and, finally, section 5 outlines the main advantages provided by this system.

## 2 WAY-Z39.50

In order to help in the design of information access systems, WAY [6] proposes a general model of architecture whose main goal is to improve information access systems through the fulfilment of three tasks: adaptation of the interface to the user, help in the searching process, and extraction and maintenance of updated data about the available information servers. These tasks are delegated to agents as has been proposed in other areas as web browsing [12, 10].

One implementation of the WAY model, applied to accessing Digital Libraries, is WAY-Z39.50[7], an application for accessing catalogues that adapts to the user both the searching process and the interface. It is designed to access servers supporting the Z39.50 information retrieval protocol standard [14], that is being installed in most libraries all over the world to provide access to their catalogues.

### 2.1 WAY-Z39.50 architecture

WAY-Z39.50 is based on an agent-decomposition method where each main task of the system is managed by a different agent. Figure 1 shows the five agents that comprise the system, representing by means of dashed ellipses those agents that are instantiated for each user, and by solid ellipses those that are instantiated only once for the whole system.

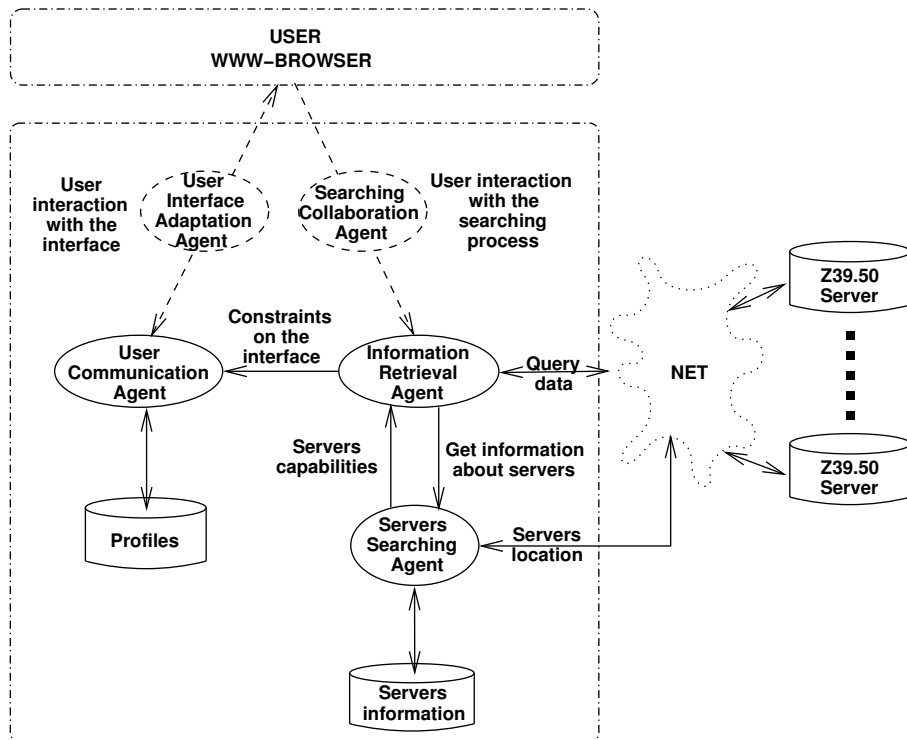


Figure 1: WAY-Z39.50 architecture

The main tasks of the system are carried out by these agents. Three of them use a fuzzy logic [18] based algorithm to implement the adaptive process (the first three ones in the following description), while the other two ones use crisp algorithms:

**User Interface Adaptation Agent:** The task of this agent is to analyze the user actions to decide the best way to adapt the interface to the user. The way the observation can be made would depend on the interface, a web browser has been used in figure 1. The agent takes decisions and transmits its suggestions to the user communication agent in order to perform them (building the interface). The user can always decide not to allow the agent to change anything.

**Searching Collaboration Agent:** The task of this agent is to spy the user searching process in order to help the user find the information she is searching for. When a user makes a query to one information server, this agent looks for the servers she prefers and launches parallel searching processes to the ones considered the most appropriate by the agent. These searches are made through the Information Retrieval Agent, as it is shown in the figure.

**Servers Searching Agent:** The task of this agent is to maintain updated information about the accessible servers. To perform this task, it can use sources such as web pages, mailing lists or USENET news groups. The agent periodically consults these sources and updates the servers' list. The period length is also updated by the agent itself. This information is kept in a relational database represented in figure 1 by the repository named "Servers Information".

The rest of the agents of figure 1 that do not use fuzzy logic to carry out their work are:

**User Communication Agent:** It is in charge of building the user interface according to user preferences, as well as transmitting the user's actions to the user interface agent following the user interface adaptation agent suggestions.

**Information Retrieval Agent:** It communicates with the Z39.50 information servers, translating user inputs into adequate queries and passing the information retrieved to the user communication agent. This is represented in figure 1 by the "Query data" double arrow to/from the Net cloud.

### 3 Fuzzy logic in WAY-Z39.50

In WAY-Z39.50, the mechanism for deciding which modifications in the interface are going to be suggested to the user is based on fuzzy decision systems. Suggestions about user interface comprise both the layout of the components, and their appearance. Besides, fuzzy decision systems are also used to help the user in the information searching process, as well as in the classification of the Z39.50 servers available.

Figure 2 shows the class diagram of the part of the system devoted to the reasoning process, which is a classical fuzzy reasoning system that includes the input and the output variables, the rules, and a mechanism to read those rules. As the rest of the system, this subsystem has been implemented using design patterns [9] in order to ease its modification, enlargement and maintenance. For instance, the points definition used to represent the labels can be easily changed by another representations without affecting the rest of the system.

#### 3.1 Designing the fuzzy relational algorithms

Fuzzy Relational Algorithms [17] (FRAs) are the method used to store the knowledge required to take the decisions (for instance, which server is going to be used). FRAs are composed by a finite set of fuzzy conditional statements (antecedent and consequent). Antecedent of conditional statements are conjunctions of fuzzy variables defined over inputs. For instance:

**IF** *Time\_Server\_1* **is** *Long* **and** *Errors\_Server\_1* **is** *High* **THEN**  
*Accessibility\_Server\_1* **is** *Bad*

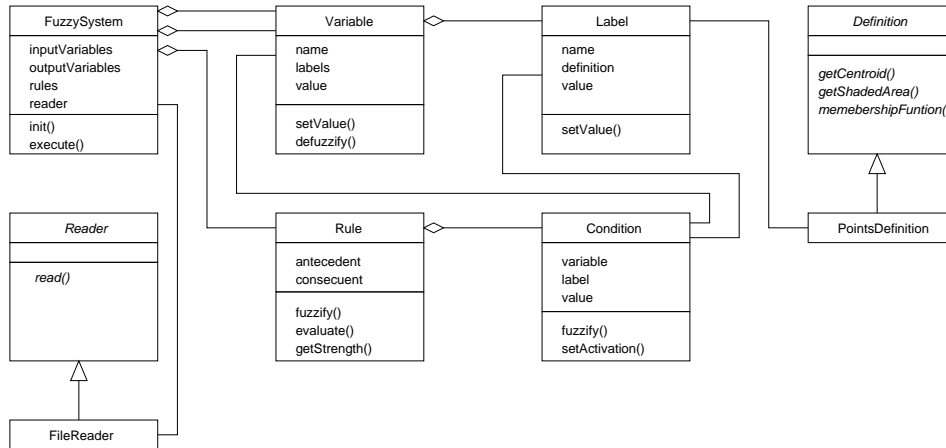


Figure 2: Class diagram of the reasoning process in WAY-Z39.50

The operation named *fuzzification* transforms a non-fuzzy quantity into a fuzzy variable. This operation is made using the definition of labels on these variables, that in WAY-Z39.50 has been done using trapeziums. For instance, a value as  $Time\_server\_1 = 10msec$  will be translated into activation levels on the fuzzy labels 0.8 activation of *Fast*, 0.1 of *Slow*, etc.

An extension of the *modus ponens* [13] is used as the inference mechanism to assign values to the fuzzy labels of the output variables. These values can be translated into non-fuzzy quantities using a *defuzzification* process. The adopted defuzzification operation is a version of the “Centre of Gravity” procedure [19].

Several automatic methods, such as neural networks or genetic algorithms, have been proposed to generate FRA in different domains. The heuristic method has been the one used in the design of WAY-Z39.50. “Heuristic” means that a human designer will use her experience in order to define the fuzzy subsets, trapeziums in the current implementation, and to write the rules.

Three different FRAs have been designed in the current implementation. Each one is used as the processing mechanism of each of the main WAY-Z39.50 agents:

1. The User Interface Adaptation Agent uses a FRA to adapt the user interface taking as inputs the detailed actions of the user during the sessions. The input of this FRA may be any action performed by the user on the interface, such as using one particular searching field, visiting first one kind of records, or visiting only the records written in one language. Its output also affects interface aspects as, for instance, placing the searching fields in a different way, ordering the records in a special manner or showing only the records written in the user mother tongue.
2. The Information Access Agent uses the FRA to decide how many simulta-

neous searches are going to be made to speed up the searching process and which are the servers preferred by each user. The inputs of this FRA would be the actions performed by the user on the results shown, as the extended records visited or the interest shown for previously proposed searches, and the responses obtained from the servers, as how long returning results take. The output of the FRA affects the number of parallel searches launched for every query of the user and the servers to which these searches are sent to.

3. The Servers Searching Agent uses a FRA to keep information about the available Z39.50 servers. As its main task is to maintain information, its inputs have to deal with the variability of the sources of information used to locate servers and the frequency of variations in the servers location and access data. The output of the FRA deals with the period of time of updates of information, as the more variations sources or servers have, the earlier the information is updated.

### 3.2 Implementation issues

One of the requirements for WAY-Z39.50 implementation was that the software being developed should be broadly available. In order to fulfil this requirement, Java programming language was chosen to make WAY-Z39.50 available through a web page. A Java applet implements the interface of the application that is built by the user interface adaptation agent. This agent communicates with the information access agent and the servers searching agent using Java-RMI (Remote Method Invocation) technology to support object distribution.

The software developed is implemented in Java 1.1.5 and it is structured into a hierarchy of packages. The root of this hierarchy is the `way` package, whose main elements are the subpackages `way.interface`, that implements the interface building facilities, `way.z3950`, that implements the client side of the Z39.50 protocol, and `way.agents`, that implements the agents of the system. The most relevant Java subpackage, that concerns fuzzy issues in WAY-Z39.50 code is `way.agents.reasoning` which implements the FRA.

This implementation of FRA comprises the definition of fuzzy input variables, by means of their name and their membership functions (defined as the four points of a trapezium of height 1) defined in their range. The same format is used to describe fuzzy output variables and their labels.

Fuzzy rules are also represented as a separate Java class made up by the `antecedent` and the `consequent`. Each of them is composed by a set of `conditions` (represented by another class). These conditions correspond to membership relations. The `rule` class includes a method to implement the evaluation which invokes the fuzzyfication of the components, then performs the modus ponens, and finally invokes the defuzzyfication process.

Given this implementation, the definition of FRA consists of the enumeration of the inputs, describing the labels using four points which define the trapezium; the enumeration of the outputs in the same way; and the specification of the fuzzy rules which describe how the outputs are modified depending on the inputs.

## 4 Fuzzy process in a session with WAY-Z39.50

Once the architecture of WAY-Z39.50 has been described, a session with the application is going to be detailed.

When a user first enters the system, she is asked to enter a user name and a password. These data do not have a security purpose, since the system may be freely accessed, but they are used to identify which agents are assigned to that user (a user adaptation agent and an information access agent). These agents are in charge of the adaptation process and the storage of the user preferences that are used in the reasoning process. These preferences include the options that the user can explicitly define (as the background colour) and the ones that the agents propose and she accepts. Thus, saving the state of the agents means maintaining the user preferences about the interface and the searching process, as they are a part of the agents. Whenever a user reenters the system, she will be assigned her two personal agents by recovering their previous states and, therefore, her last preferences.

At this point, the user interface adaptation agent builds a particular interface for that user. If this is the first time the user enters the system, a default interface is shown. This first interface depends on the kind of machine the user is connecting from and on the default language selected in the navigator used to access the system. During the session, the user interface adaptation agent is also active and spying user actions.

Figure 3 shows a WAY-Z39.50 customised interface where the user has made a query and is visiting the extended records. In a normal session with WAY-Z39.50 the user enters data in one or more input fields in order to find the bibliographical records that contain the information entered. When the user makes a query to the system, the first result obtained is the number of records satisfying the query. The user can consult a summary of the results as short records (upper right area of figure 3) meaning that she finds acceptable the number of records obtained. If the information shown in these records seems interesting, she can visit their extended version (bottom right area of figure 3).

While the user request is being sent to the Z39.50 server by the system, the information access agent is also working to collect information about the user actions, as records visited or servers selected, and about the behaviour of the servers, as response time or amount of results obtained. With this information, the agent decides and launches new searches on the servers considered more suitable for that user. The searches launched by the agent are suggested to the user indicating the number of records found matching her query in other servers. This way, the user saves the time that she would have employed in making another query in the same server or going to another server and repeating the query, as it should be done in the classical systems that integrate many library servers.

The observation process developed by the user interface adaptation agent generates information that is used as input for its fuzzy reasoning process. For instance, during a session, if the field `title` is being used more frequently than the field `author`, the fuzzy reasoning process performed by the user interface adaptation agent can obtain through the fuzzy process a higher value for the `importance` vari-

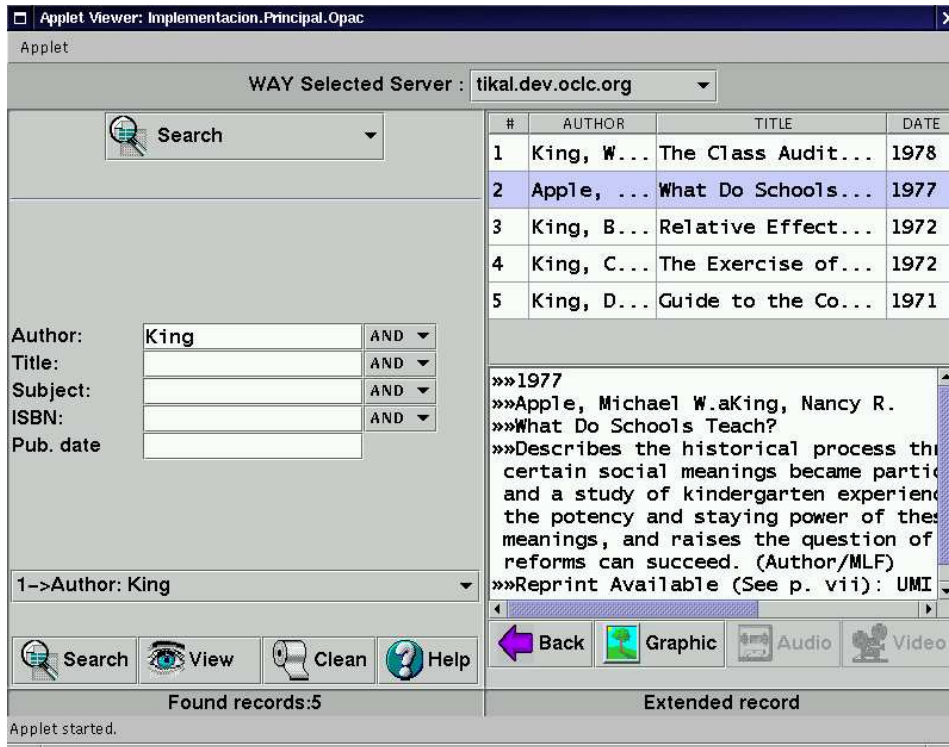


Figure 3: WAY-Z39.50 user interface

able of the field `title` than for the field `author`. If the field `author` is displayed in the first position and the field `title` in the second one, this agent would suggest to interchange their positions. Some of the inputs that affect the `importance` of a field are, for instance, the `use` of the field and its `success_rate`. The `use` of the field is the percentage of times that the field is used to make a query, and the `success_rate` depends on variables such as `searching_success` that again depends on other variables as `response_time`, or `long_records_visited`.

The user can always accept or refuse proposed changes, as some users do not want computers to take decisions that affect their work without their permission. Let's show a couple of examples about possible adaptations. A user may have visited only books records so, as the `books_records_frequency` would have a very high value, the agent would propose the user to order the records beginning with the ones corresponding to books. If the same user consults mainly records in Spanish, the `spanish_records_frequency` would get a higher value than the English or French ones, so the agent would propose to order the records beginning with books in Spanish and following with books in English and so on.

But while the user is querying the system, also the information access agent is



spying the process, which means that it is getting the values of the input variables to its fuzzy reasoning process. The agent has information about the servers that give the user back the best results, where “best results” depends among others, on variables such as `server_accesses`, that is the number of times that the user has accessed a server; `server_success`, the percentage of successful connections to that server; `server_speed`, the speed of the connection; `long_records_visited`, the number of long records looked up in that server, etc. The output of the reasoning process would affect variables as `server_qualification` or `server_usage`, from which the list of preferred servers for every user is built.

Finally, the user can choose to go deeper into one of the proposed searches, which will increase the value of `proposed_searches_visited`, refine the query or choose another server, which will modify its `server_usage`. This input information produces modifications on the number of parallel searches to be launched for every user query, `number_parallel_searches`, as well as on `server_qualification`.

Concerning the adaptive process, the use of fuzzy logic has shown the following advantages:

- The mechanism for specifying the rules to modify the user interface and to participate in the searching process are closer to human language than other reasoning mechanisms, as long as the logic rules are written in almost natural language and the variables, and labels involved in those rules have also got a well-known meaning, which makes them easy to design.
- The rules that guide the adaptation process are easily accessible as they can be stored on a text file or on a data base, so, they can also be easily modified. A debugging process makes sure that the rules are correct and notifies the designer of the system if they are not.
- The reasoning process is really lighter than other mechanisms that require heavy computations.

## 5 Conclusions

One of the main advantages of using WAY-based systems is that their interface adapts itself according to the user actions while she uses the system. In the same way, the system also offers alternative searches in the servers more adequate to the user. As it has been shown in the paper, all these adaptations are decided using a fuzzy reasoning process. So, the first conclusion that can be obtained from WAY-Z39.50 system is that fuzzy powered systems can be used to implement adaptiveness in information access systems.

The use of design patterns make it really easy to modify the fuzzy reasoning motor. For instance, the modification of the fuzzyfication or defuzzyfication mechanisms can be achieved without interfering with any other part of the system. In the same way, the addition of new rules, or the modification of the labels can be easily done.

Preliminary versions of WAY-Z39.50 and in deep information about WAY model are available at <http://way.uc3m.es/~way>. And finally, it has been foreseen

that new WAY-based systems will be built. In those new systems fuzzy reasoning mechanisms will be considered as a major option because of the great performance observed in WAY-Z39.50.

## References

- [1] M. Balabanovic. *An adaptive web page recommendation service*. Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, (California, USA). Feb. 1997.
- [2] D. Benyon, G. Gavies, L. Keller, and Y. Rogers. *A Guide to Usability - Usability Now!* The Open University, 1990.
- [3] D. Benyon and P. Palanque. *Critical Issues in User Interface System Engineering*. Springer-Verlag, 1996.
- [4] David Beyon, Thomas Green and Diana Bental. *Conceptual modelling for user interface development*. Springer, 1999.
- [5] Donald L. Dewitt. *Going digital: strategies for access, preservation, and conversion of collections to a digital format*. Haworth Press, 1998.
- [6] C. Fernández and P. Díaz and I. Aedo. *WAY: An Architecture for User Adapted Access to Z39.50 Servers based on Intelligent Agents*. Proceedings of the European Conference on Digital Libraries pp. 665-666. 1998.
- [7] Camino Fernández, Paloma Díaz, Ignacio Aedo and Vicente Matellán. *WAY-Z39.50: Adaptive access to library catalogues*. Proceedings of the 4th ACM Conference on Digital Libraries, pp. 225-226, Berkeley (California, USA), 1999.
- [8] Camino Fernández. *WAY: Un modelo para el acceso adaptativo a la información*. PhD Thesis, Universidad Politécnica de Madrid. Sept. 2000.
- [9] Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series. Addison Wesley, 1995.
- [10] Henry Lieberman. *Letizia: An Agent That Assists Web Browsing*. Joint Conference on Artificial Intelligence. Montreal (Canada), 1995.
- [11] Henry Lieberman, Neil Van Dyke, and Adriana Vivacqua. *let's browse: a collaborative web browsing agent*. International Conference on Intelligent User Interfaces, Los Angeles (California, USA), Jan. 1999.
- [12] Pattie Maes. *Reflections of Agents that Reduce Work and Information Overload*. Readings in Intelligent User Interfaces. Editors Maybury, and M. T. and Wahlster, W. Morgan Kaufmann Press, 1998.

- [13] E. H. Mamdani. *An analysis of formal logics as inference mechanism in expert systems*. International Journal of Man-Machine Studies. Vol. 21, n. 1, pp. 213-227, 1984.
- [14] J. Michael and M. Hinnebusch. *From A to Z39.50: A Networking Primer*. Mecklermedia, 1995.
- [15] J. Rubin. *Handbook of usability teststing*. Jonh Wiley & Sons Inc. New York, 1994.
- [16] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Publishing Company, 3rd Edition, 1998.
- [17] L. A. Zadeh. *Outline of a New Approach to the Analysis of Complex Systems and Decision-Making Processes*. Transactions on Systems, Man and Cybernetics. Vol. 3, n. 1, pp 28-45.
- [18] L. A. Zadeh. *Fuzzy Logic*. IEEE Computer. Vol. 21, n. 4, pp. 83-93, 1988.
- [19] Hans-Jurgen Zimmermann. *Fuzzy Sets. Theory and its Application*. Kluwer Academic Publishers, 1990.