A Fuzzy approach to Adaptive Hypermedia

Antonina DATTOLO◇, Vincenzo LOIA◇◇

◇ Dipartimento di Matematica ed Applicazioni "R. Caccioppoli"
Universitá degli Studi di Napoli "Federico II" - Italy
dattolo@matna2.dma.unina.it
◇◇ Dipartimento di Matematica e Informatica
Universitá degli Studi di Salerno - Italy
loia@unisa.it

Abstract

This paper presents an open, concurrent model of adaptive hypermedia; all its functionality and data are completely distributed on a web of autonomous actors. The model that enables the system to the interactive adaptivity is a fuzzy stereotype-based user model; its presentation and application are exploited in the cooperative actor-based environment, generated from the hypermedia model. The merge of these two paradigms offers a new concurrent and fuzzy approach in the adaptive hypermedia system domain.

1 Introduction

Adaptive hypermedia systems (AHS's) is a new area of research, but it is attracting considerable attention from the research community, as shown by a growing body of literature [4] and the existence of active research groups [2]. AHS's are *all hypertexts and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user* [4]. In the respect of this definition, three important entities compose an AHS:

- a hypermedia system;
- a user model;
- an interface able to adapt the hypermedia using the user model.

The first AHS's appeared as adaptive graphical interfaces able to support simple but frequent operations such as undo/redo strategies, active help, predefined plans-of-actions schemes. With the evolution of the hypermedia models, and with the enormous diffusion of hypermedia applications in different professional and social classes of users, more efficient adaptive methods are now strongly required. Fortunately, adaptive hypermedia community has beneficiated from the long experience acquired by research on user modelling. Our approach to user modelling is by exploiting effectiveness of fuzzy logic in describing qualitative feature, like interest of the user and his/her concern with information in the hypermedia.

This work has been conceived as part a larger project; it advantages from our experience matured in the hypermedia systems domain [9, 10, 11, 12, 15], in defining a specific adaptive architecture [13, 14] and in treating fuzzy modelling [7, 6, 8].

The paper is organised as follows. Section 2 describes the architecture of our distributed, open hypermedia model. In the following sections, first we introduce (Section 3) our idea of a fuzzy stereotype-based model for hypermedia systems, then in Section 4 we present WebHyDe, the web-based architecture of our model and in more detail the three actor populations composing the general architecture. Section 5 synthesises the adaptive process, emphasising some cooperation activities among the different actor classes. Concluding remarks and future work end the paper.

2 The distributed architecture

In classical hypermedia [19], atomic and composite nodes and links are passive entities. They require external abilities in order to support the operations that are required during the user browsing: a central control unit exploits the execution of these operations, exerting its active role on a set of passive components. In our model this situation is completely turned over; there is no monolithic resource responsible for the global management but an aggregation of autonomous and independent actors [1] own behavioural responsibility and partial perception of the other actor populations. The framework is completely distributed and all the services and the information are sprinkled over a web of autonomous actors, which cooperate to obtain local or global goals. Locally, actors perform hypermedia adaptation, at the nodes; globally they cooperate towards the overall updating of user model involved in the adapting process. Three levels compose our model, as depicted in Figure 1.

The first layer, named Structural level, corresponds to the architectural model provided from

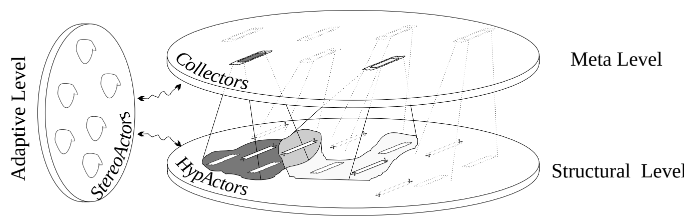

Figure 1: Structural, Meta and Adaptive Levels.

the hypermedia author; it is composed of the populations of HypActors. These actors substitute the concept of classical hypermedia node, offering a dynamic and operative perspective of these entities.

The second layer, named Meta level, provides a more abstract perspective of the hypermedia; the Collectors, constituting this level, improve the modularity and thus reusability of the hypermedia, allowing the visualisation and the manipulation of subsections of the underlying Structural level.

The third layer, named Adaptive level, is composed by the StereoActors. These actors manage a dynamic flow of user information in order to make adaptive the system on the basis of the user browsing.

3 Fuzzy Stereotype based User Modelling

We assume that a user can be approximated via a set of stereotypes. Stereotypes are fixed user typologies. Stereotype differences rely upon different degrees of concern for the information contained in the hypermedia. Let $S = S_1, S_2, \cdots, S_N$ be the set of stereotypes and $D = D_1, D_2, \cdots, D_M$ the set of abstract objects included in the hypermedia. Any D_i represents a text or a video or also part of them. To denote the fact that the stereotype S_j is completely interested in the subjects $D_{i1}, D_{i2}, \cdots, D_{ir}$ we use the writing:

$$[D_{i1}, D_{i2}, \cdots, D_{ir}]^{ci}$$

where ci is the fuzzy number representing the term *completely interested* of the fuzzy variable **Interest** [24]. The complete representation of an user U is a string:

$$U = [D_{11}, \cdots, D_{1g}]^{\alpha}[D_{21}, \cdots, D_{2q}]^{\beta} \cdots [D_{h1}, \cdots, D_{hs}]^{\omega} \tag{1}$$

which reads: user U concern about objects $D_{11}, \cdots, D_{1g}$ is α; user U concern of objects $D_{21}, \cdots, D_{2q}$ is β and so on. In this way, user description is a partition of the set of objects contained in the hypermedia. The set of objects constituting the partition is evaluated by means of terms of a fuzzy linguistic variable. The user description is a fuzzy classification of the hypermedia parts.

¿From a global perspective, we provide the system with a set of fixed description of an user (stereotypes, i.e. fuzzy strings). Hence the user model is constructed by merging some of such stereotypes. The user model, attained in this way, is a new evaluation of the hypermedia components. Such an evaluation affects the adaptation phase, i.e. the building of the pages and the presentation of the nodes to the user.

If we think of the hypermedia as a graph, a fuzzy stereotype representation is a coloured version of the graph of the hypermedia. In the stereotype graph, different paths are coloured differently according to the degree of concern of the stereotype. Some part of the graph will be completely hidden (namely the ones not at all interesting for that kind of user), some other will be depicted very sharply (to mean that the user is very concerned with the information contained in those nodes), generally most of the graph will have a colour in the middle to indicate that the user is interested to those parts but with a fuzzy degree. If we suppose stereotype graph as drawn on transparent sheets, we can think to a real user as a superimposition of some of them. The resulting graph will be a new view of the hypermedia roughly averaging the ones of the stereotypes considered.

To compute formally and numerically such a superimposition we rely on an algebraic framework to aggregate fuzzy partition of a general universe of discourse [7].

Let A and B be two instances of the fuzzy strings we have introduced above:

$$A = a_n^{\alpha_n} a_n^{\alpha_n} \cdots a_1^{\alpha_1} \qquad \text{and} \qquad B = b_m^{\beta_m} b_{m-1}^{\beta_{m-1}} \cdots b_1^{\beta_1}$$

where $a_i (1 \leq i \leq n), b_j (1 \leq j \leq m)$ are subsets universe of discourse D; A and B represent different classification of items in D, and the fuzzy numbers $\alpha_i' s$ and $\beta_j' s$ represent the terms of the linguistic variable used for a qualitative evaluation of items in D; $\alpha_i \leq \alpha_{i+1}$ and $\beta_j \leq \beta_{j-1}$, i.e. mean points of membership functions of fuzzy numbers α_i and β_j respect the order of the semantic of linguistic terms they represent. We define the composition of A and B as:

$$A \diamond B = a_n^{\alpha_n} a_n^{\alpha_n} \cdots a_1^{\alpha_1} \diamond b_m^{\beta_m} b_{m-1}^{\beta_{m-1}} \cdots b_1^{\beta_1} = c_{m+n-1}^{\gamma_{m+n-1}} c_{m+n-1}^{\gamma_{m+n-2}} \cdots c_1^{\gamma_1}$$

$$c_i = \begin{cases} \bigcup_{j=1}^{i} a_{i+1-j} \cap b_j & 1 \leq i \leq m-1 \\ \bigcup_{j=1}^{m} a_{i+1-j} \cap b_j & m \leq i \leq n-1 \\ \bigcup_{j=i+1-n}^{m} a_{i+1-j} \cap b_j & n \leq i \leq n+m-1 \end{cases} \tag{2}$$

$$\gamma_i = \begin{cases} \frac{1}{i} \sum_{j=1}^{i} \frac{1}{2}(\alpha_{i+1-j} + \beta_j & 1 \leq i \leq m-1 \\ \frac{1}{m} \sum_{j=1}^{m} \frac{1}{2}(\alpha_{i+1-j} + \beta_j) & m \leq i \leq n-1 \\ \frac{1}{i-n+j} \sum_{j=i+1-n}^{m} \frac{1}{2}(\alpha_{i+1-j} + \beta_j) & n \leq i \leq n+m-1 \end{cases} \tag{3}$$

the following properties hold [5] : closure, associativity, commutativity, idempotence, existence of a neutral element. Intuitively the composition $A \diamond B$ yields a new string which is a finer classification of D summarising the one represented by strings A and B. Classification is indeed qualitative, since fuzzy numbers yielded by the composition are interpretable as new linguistic terms by means of a linguistic approximation procedure [5]. In terms of partition, we can think of the result $A \diamond B$ as a sort of "common refinement" of the fuzzy partitions A and B.

We rely, also, on the possibility of defining relative relevances of the operands of the composition operation ($\diamond$). A weighting operator defined in [7] will be used. Weighting the string B (in the previous composition) to a degree p_B we obtain a new string B'. We write $B' = B^{p_B}$. The result of $A \diamond B^{p_B}$ will be a new fuzzy evaluation of D, say C', which is biased, with respect to $C = A \diamond B$, towards the evaluation of B.

We define, also, a distance in the space of the fuzzy strings, we have just introduced. Given the two strings: $A = a_n^{\alpha_n} a_n^{\alpha_n} \cdots a_1^{\alpha_1}$ and $B = b_m^{\beta_m} b_{m-1}^{\beta_{m-1}} \cdots b_1^{\beta_1}$

$$D(A,B) = \sum_{x \in D} (A^{-1}(x) - B^{-1}(x))^2 + \sum_{i=1}^{n} Var(a_i, B) + \sum_{i=1}^{m} Var(b_i, A) \tag{4}$$

and for any subset $a \subseteq S$ and any string G :

$$Var(a,G) = \frac{\sum_{x \in a} (G^{-1}(x) - \frac{\sum_{x \in a} G^{-1}(x)}{|a|})^2}{|a|}.$$

The term $Var(a, G)$ is like a variance of the elements contained in a, with respect to their spread in the string G.
In our framework such a distance is involved in the search for the best approximating stereotype which, at the same time, is also compatible with the present profile of the actual user. The chosen stereotype will be the one minimising the distance (4). Some considerations are also to be given about the distance chosen. The distance of two strings representing user concerns like in formula (1) will be proportional to both the difference of the evaluations of the items D_i and also to the difference in the clustering of the hypermedia objects, in the two strings considered. The former condition (difference of the evaluations) warranties the classification of interests: their relative order. The latter condition (grouping of the hypermedia objects) aims at considering the clustering of interests of the users.

Let $D = \{D_1, \cdots, D_M\}$ be the set of hypermedia objects. Suppose to consider N different stereotypes, then there will be N strings $S_1, \cdots, S_N$ defining fuzzy evaluations of the set D. More precisely, each S_j states for any D_i , the degree of concern of the j^{th} stereotype with respect to the i^{th} hypermedia object. At any step t the actual user is represented by a string U_t , also an evaluation of the set D, obtained as the composition of some stereotype strings. When a page P has to be displayed to the user, the system will check in the string U_t the evaluation of the hypermedia objects and will build the actual page accordingly. If for example there is a link to an object with a low evaluation, the link will be hidden or only mentioned as a secondary information; furthermore a text could be represented in the data base D as more than one object, and only those parts which are considered more interesting, according to the string U_t , will be displayed. The string U_t is updated step by step as shown in the following sections.

4 Formal description of the three actor classes

The framework of our AHS is completely distributed. All the information and the services are sprinkled over a web of three actor classes. The computational platform is conceived using an extension of Java towards concurrent and distributed computing, XML and HTML. The main idea at the basis of realisation are introduced in the next subsection 4.1, while the HypActors, Collectors and StereoActors are detailed respectively in subsections 4.2, 4.3 and 4.4.

4.1 WebHyDe

When the project HyDe (Hypermedia Distributed Design) [15] started in the beginning of 90's we were essentially concerned with the study, realisation and experimentation of a set of new tools for designing hypermedia authoring systems. We decided to adopt as basic platform the language HyperCLAS, an extension of CLOS (Common Lisp Object System) [3], towards concurrent computing. HyperCLAS was dutied of efficient features to treat tasks at the basis of hypermedia management (as version control [12] and adaptivity [14]). The "visualization" was performed in X11 environment thanks to designed tools [21] linkable with HyperCLAS.

The availability of the Web today has definitely changed the information scenario and thus we are currently involved with a retargeting of HyDe architecture. This task has started recently and is still ongoing. Here we wish just to give some design choices that we have undertaken and that constitute the pitfalls of the new architecture, namely WebHyDe.

WebHyDe is strongly based on Java. More precisely, the first goal was to implement in Java itself a model of actor computing. To accomplish this goal we implemented an essential entity, the Actor Class, a computation object whose states are shown in Figure 2.

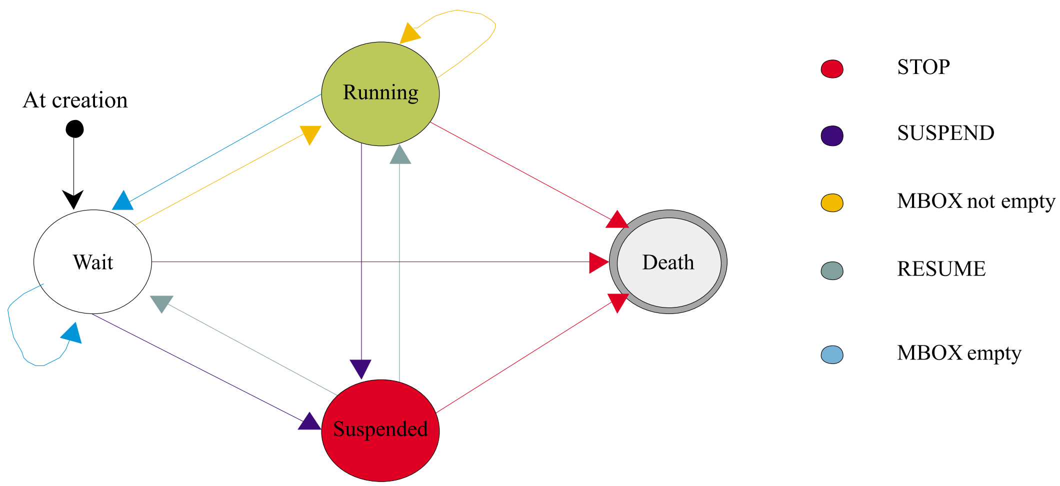

Figure 2: States of the actor entity.

An actor can be in four states: wait, running, suspended and death. At creation each actor is in "wait" state, until it receives a message from an another actor. In this case it changes state. Figure 2 shows the possible changes of state.
The actors are composed by two parts, an active part and a passive part. The active part of the actor has been realised in Java, while the passive part, that is the data part, is composed of two different web-oriented information: XML and HTML. XML [16] is a subset of Standard Generalised Markup Language (SGML) that is optimised for delivery over the Web. It is defined by the World Wide Web Consortium (W3C) [23], ensuring that structured data will be uniform and independent of applications or vendors, so that XML is nowadays a de-facto standard for exchanging data on the Internet. The most important feature of XML is that it maintains the separation of the user interface from structured data, allowing the seamless integration of data from diverse sources. A large percentage of the XML specification deals with various sorts of declarations that defines the semantics of the document to handle. These definitions specify a DTD (Document Type Definitions). In our approach, DTD are used to specify the abstract model of the hypermedia node. Upon the interaction between the user and the hypermedia the contents and the visual properties shall be specified into the XML document, that is an instance of the DTD filled by appropriate data. This corresponds to the Hypactor.

The active link between the data and the actions is set up thanks the use of an XML instruction that creates a link with an external application. The processing instructions (PI) has the form: `<?name pidata?>`. The name, called the PI target, identifies the PI to the application. The application, in our case, corresponds to the Java actors alive in the Web.

4.2 HypActors

The HypActors represent the "atomic" units of hypermedia information, similar to well-known objects, such as frames [17] or slices [22]. An important deviation from more traditional approaches is the dynamic nature of the HypActor, that is able to react to external stimuli activating internal functions (named scripts). A description of the HypActor class follows in Figure 3. In it, we find only the data part of the HypActors, the so-called acquaintances. The frame topic

HypActor		
title	sound	weights
text	to	currUser
images	from	stereotypes

Figure 3: Data part of the HypActor class.

of the HypActor is contained in the slot `title`, while the possible contents of the node are stored in the acquaintances `text`, `sound` and `image`; `to` and `from` are the addresses of the HypActors reachable from the current HypActor or respectively the ones from which it is possible to reach the current actor. The rest of the acquaintances are devoted to the adaptive collaboration with the StereoActors. In particular, the slot `weights` records the relative relevances assigned to the contacted stereotypes with respect to the node managed by the HypActor, the `currUser` maintains the current user model, `stereotypes` contains the addresses of possible stereotypes applicable on the given HypActor.
The operational part of a HypActor is a set of scripts; scripts are like sub-programs representing all the possible actions that an actor can perform.

4.3 Collectors

This class of actors is useful to manage alternative browsing structures and views of partial sections of hypermedia. These actors are introduced to create and handle separate collections of HypActors, by providing a more abstract treatment of browsing techniques. The data part description of the Collector class is given in Figure 4. The Collector inherits many acquaintances

Collector		
title	...	stereotypes
...	...	collection
...	currUser	distances

Figure 4: Data part of the Collector class.

from the HypActor class, but it has a more abstract vision of the same data. For example, the same acquaintance `stereotypes` contains the addresses list of all the existing StereoActors, and the slot `currUser` stores the global, current user model.
Since the Collector manages sections of the hypermedia, the acquaintance `collection` contains the list of HypActors/Collectors that the Collector addresses; in fact, the traditional role of container of more entities is assumed only from a logic standpoint, in the sense that the Collector addresses, and not necessarily encapsulates, the actors in `collection`. The slot `distances` is devoted to store the distances computed by the StereoActors (see next Section 4.4) which the same Collector will use to decide the user model updating.

4.4 StereoActors

The StereoActors are involved in the actual updating phase of the user model. StereoActors receive the present user model from the HypActors and compute possible updating, eventually used by the Collector for changing the actual user model. Figure 5 sketches the definition of a StereoActor. StereoActors store in the acquaintance `fuzzyPartition` the fuzzy evaluation of the hypermedia objects which characterise the stereotype it represents. The slots `weight` and `currUser` have the same function as in the HypActors. StereoActors update `currUser` by componing (see Section 3) it with the value contained in `fuzzyPartition`. The slot `distance` stores the distance between the new content of `currUser` and the value of `fuzzyPartition` according to the formula (4).

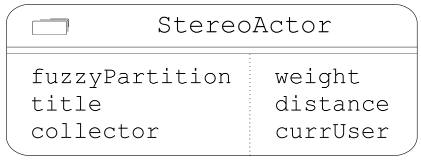

Figure 5: Data part of the HypActor class.

5 Cooperative activities

The adaptive process follows a cycle that involves all the three actor classes in very intensive cooperative activities.

Suppose that at time t the user is navigating on the HypActor H_k. This HypActor received the user model so far computed from the Collector. At time 0 the user model is the trivial fuzzy partition; the whole set of hypermedia objects have the same evaluation (i.e. the most general user model).

Step 1: HypActor $H_k \rightarrow$ *StereoActors* $S_1, S_2, \cdots, S_N$.
The HypActor H_k scans the user model (i.e. the fuzzy evaluations of the objects of its node) and builds the display accordingly. According to the actions that the user performs on the HypActor H_k, it sends the current user model U_k to any StereoActor in `stereotypes` together with a weight w_k representing how much similar is user's behaviour to the one of the stereotype represented by the addressed StereoActor. This action is shown in Figure 6.

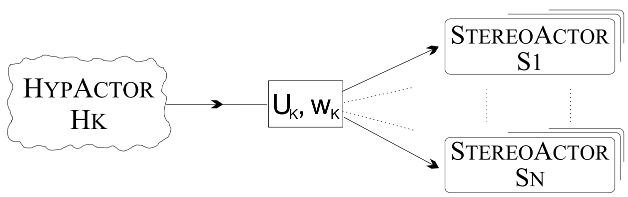

Figure 6: The HypActor H_k updates in multicasting the StereoActors $S = S_1, S_2, \cdots, S_N$.

Step 2: StereoActors $S_1, S_2, \cdots, S_N \rightarrow$ *Collector.*
Let A_k represents the generic fuzzy partition contained in the acquaintance `fuzzyPartition` of the contacted StereoActors; let U_k and w_k be, respectively, the actual user model and the weight, sent to the committed StereoActors by the HypActor.
StereoActors computate $U_k' = U_k \diamond A_k^{w_k}$ and the distance $d(U_k', A_k)$, according to the formulas (2) (3) and (4); these quantities are delivered to the Collector. The update of the current user model is shown in Figure 7.

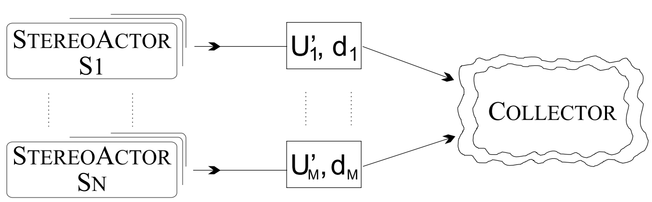

Figure 7: StereoActors $S_1, S_2, \cdots, S_N$ informs Collector about relative user models and distances.

Step 3: Collector $\rightarrow$ *HypActors* $H_1, H_2, \cdots, H_x$.
Collector, in turn, determines the new user model, by chosing among the received ones the $U_{k'}$ such that $d_{k'} = min_k d_k$. The new user model $U_{k'}$ is then sent to the HypActors $H_1, H_2, \cdots, H_x$ in the neighbourhood of the HypActor H_k. This action closes the cycle and is shown in Figure 8. The same HypActor H_k, in gray in Figure 8, is informed about the current updates.

Summarising: the HypActors perform display by exploiting the information in the user model. They commit to the StereoActors the activity of guessing the improvement of the user

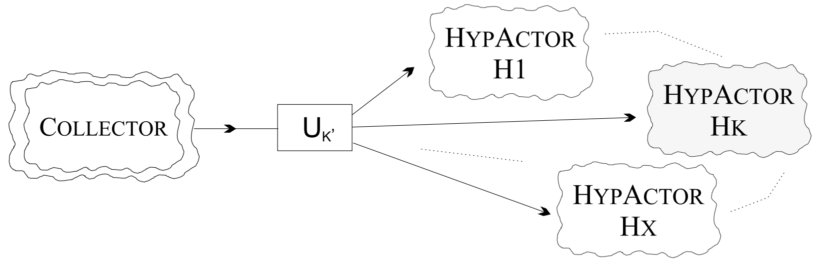

Figure 8: Collector communicates the new user model $U_{k'}$ to the HypActors $H_1, H_2, \cdots, H_x$.

model, by computing new fuzzy approximation of the actual user. Such possible user models are, timely, analysed by the Collector, which decides the best approximation computed and delivers it to a chosen collection of HypActors. In our approach a topological identification of this collection is obtained as the union of the currently active HypActors and their close neighbours, extended by means a recursive process f times (where f is a natural number dependent on the application).

6 Conclusion

We presented an actor framework for a dynamic definition of hypermedia. The actor paradigm was used to define an adaptive user modelling methodology, relying on the concept of fuzzy user stereotype. Fuzzy stereotypes are intended to be fuzzy partition of the hypermedia space.
Hypermedia entities are meant to be dynamic. Stereotypes are also actors cooperating concurrently in devicing the best user description. Thus user model is attained as the result of a parallel activity.
Rationalisation of links and communication duties were to be considered in order to reduce message passing overload.
Thus the model of AHS merges a distributed, actor-based hypermedia architecture with a fuzzy user model. From each of these two features it takes advantages:

1. The use of the actor model entails a concurrent and asynchronous computation; this produces a model completely distributed, in which the same user model is sprinkled on the different actor classes. In this way, thanks to the autonomy and the active behaviour of each actor, the model offers dynamical and run-time adaption to the user.

2. Fuzzy stereotypation allows qualitative treatment of uncertain information. Fuzziness is also used as a means to compress information: qualitative description avoids the necessity of huge set of stereotypes in order to attain fine description. The model here described also avoids problems like coherence and reasoning maintenance, generally entailed by the use of some logical inference calculus.

Addressed research aims at the integration of such a model in a evolutionary system in order to stress self adapting feature of the model. The use of genetic algorithms and evolutionary strategies for the tuning of StereoActors and the dynamic modification of the distributive framework, are topics of ongoing research.

References

[1] Agha, G. (1986). *Actors: A Model of Concurrent Computation in Distributed Systems.* MIT Press, Cambridge, MA.

[2] Adaptive Hypertext & Hypermedia Web home page:. http://www.education.uts.edu.au/projects/ah/.

[3] Bobrow, D. G., DeMichiel, L., Gabriel, R., Kiczales, G., Moon, D., Keene, D. (1988). CLOS specification;X3J13 document 88-002R.. *ACM-SIGPLAN Notices*, **23**.

[4] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modelling and User Adapted Interaction*, **6**(2-3).

[5] Cicalese, F., Gisolfi, A. (1996). Classifying through a fuzzy algebraic structure. *Fuzzy Sets and Systems*, **78** 317-331.

[6] Cicalese, F., Loia, V. (1998). A Fuzzy Evolutionary Approach to the Classification Problem. *Journal of Intelligent and Fuzzy Systems, Special Issue on Evolutionary Computation***6** 117-129.

[7] Gisolfi, A., Loia, V. (1995). A Complete, Flexible Fuzzy-based Approach to the Calssification Problem. *International Jorunal of Approaximate Reasoning***13** 151-1183.

[8] Cicalese, F., Loia, V. (1997) A Distributed, Concurrent Architecture for Fuzzy Evolutionary Models. *Proc. of European Symposium on Intelligent Techniques*, March 20-21, Bari, Italy.

[9] Dattolo, A., Gisolfi, A. (1994). Analytical version control management in a hypertext system. In *Proc. 3th Int. ACM Conf. on Information and Knowledge Management - CIKM'94*, Nov. 29 - Dec. 2, NIST, Gaithersburg, MD, 132-139.

[10] Dattolo, A., Loia, V. (1995). Hypertext Version Management in an Actor-based Framework. *Proc. of the VII Intern. Conf. on Advanced Information Systems Engineering - CAiSE*95*, LNCS(932), Jyvaskyla, Finland, 12-16 June, 112-125.

[11] Dattolo, A., Loia, V. (1996). Agent-based Design of Distributed Hypertext. *Proc. 11th Intern. ACM Symposium of Applied Computing - SAC'96*, February 17-18, Philadelphia, Pennsylvania.

[12] Dattolo, A., Loia, V. (1996). Collaborative Version Control in an Agent-based Hypertext Environment *Information Systems*, **21**(2):127-145.

[13] Dattolo, A., Loia, V. (1997). A Distributed, Self-Adaptive Model of Hypermedia System. *Proc. of the IEEE Hawai'i International Conference on System Sciences - HICSS-30*, January 7-10, Aston Wailea Resort, Wailea, Hawai'i, USA.

[14] Dattolo, A., Loia, V. (1997). Active distributed framework for adaptive hypermedia. *International Journal of Human-Computer Studies*, **26**(5):605-626.

[15] Dattolo, A., Loia, V. (2000). Distributed Information and Control in a Concurrent Hypermedia-oriented System. to appear on *International Journal of Software Engineering and Knowledge Engineering.*

[16] Extensible Markup Language (XML). http://www.w3.org/XML

[17] Garzotto, F., Mainetti, L., Paolini, P. (1995). Hypermedia Design, Analysis, and Evaluation Issues. *Communications of the ACM*, **38**(8):74-87.

[18] Gisolfi, A., Loia, V. (1994). Designing complex systems within distributed architectures. *International Journal of Applied Artificial Intelligence*, **8**(3):393-411.

[19] Grønbæk, L., Trigg, R. H. (1994). Design issues for a Dexter-based hypermedia system. *Communications of the ACM*, **37**(3):40-49.

[20] Halasz, F., Schwartz, M. (1994). The Dexter hypertext reference model. K. Grønbæk and R. H. Trigg eds. *Communications of the ACM*, **37**(3):30-39.

[21] Kimbrough, K., Lamott, O. (1990). *Common Lisp User Interface Environment.* Texas Instruments Inc.

[22] Isakowitz, T., Stohr, E. A., Balasubramanian, P. (1995). RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, **38**(8):34- 44.

[23] World Wide Web Consortium (W3C). `http://www.w3.org/`

[24] Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Infromation Science*, **8** 199-245, 301-357.