The Role of Explanation Systems in MultiAgent Applications

Cecília Kremer Vieira da Cunha and Clarisse Sieckenius de Souza

Departamento de Informática - PUC-Rio
Rua Marquês de São Vicente 225 - 22453-900 - Rio de Janeiro,RJ - Brazil
{ceciliak,clarisse}@inf.puc-rio.br

PUC-RioInf.MCC05/97

ABSTRACT

This paper discusses the role of Explanation Systems in multiagent applications. Our starting point is an Active Design Documentation (ADD) [8] tool. Its typical explanation facility is adapted in order to explore the resources available in multiagent environments. Following a Semiotic Engineering approach [4], explanations in MultiADD (Multiagent ADD) are used as a means of communication between system designer and system user, and provide a common linguistic embedding for user-user communication during negotiation and conflic mitigation.

Keywords

natural language explanations, multiagent communication, semiotic engineering

RESUMO

Este artigo discute o papel de Sistemas de Explicação em aplicações multiagente e multiusuário. Nosso ponto de partida é uma ferramenta de Documento Ativo de Design (*Active Design Document - ADD*) [8]. Sua facilidade de Explicação típica é adaptada de forma a explorar os recursos disponíveis em ambientes multiagente e multiusuário. Seguindo a abordagem da Engenharia Semiótica [4], explicações no MultiADD (ADD Multiagente) são usadas como meio de comunicação entre o designer do sistema e o usuário do sistema, e provêem um recurso lingüístico padrão para comunicação usuário-usuário durante a negociação e resolução de conflito.

Palavras-Chave

explicações em linguagem natural, comunicação entre agentes, engenharia semiótica

INTRODUCTION

Explanation systems are an important resource for knowledge-based applications. They provide a more cooperative environment between users and application, once users can understand system's automatically generated solutions and therefore commit to them. Users are also able to keep track of their own (usually complex) work and rationale. This way, an Explanation facility is responsible for showing the system's model and functioning so that the user can make use of all the benefits provided by the application.

Moore[14] describes criteria for evaluating explanation systems, which are: naturalness, responsiveness, flexibility, sensitivity, fidelity, sufficiency and extensibility (pp.7-8). In the application that will be presented in this paper, we will be concentrating on naturalness (explanations should be structured according to natural discourse strategies); sensitivity (users and tasks models should determine the type of knowledge and information presented in explanations); fidelity (explanations should reflect the system's knowledge and reasoning); and extensibility (explanations facilities should be easily expandible to account for novel demands and needs for information).

Applications' designers apply the concepts of User-Centered System Design (USCD) [15] in order to develop good user interface languages. A user, from a USCD point of view, is communicating directly with a computer system. In a semiotically-motivated approach the system is a means of communication between system designer and the system user [2,4]. The following figure depicts the Semiotic Engineering [4] Framework (already adapted to support our multiagent application and the interesting points that will be presented). USCD proper is a kind of "zoom-in" on the arrows that reach the users.

In this article we will present an Explanation facility for a multiagent system, elaborated according to the semiotic engineering framework. We first disclose MultiADD's Explanation Model and then describe the roles that MultiADD's explanations are supposed to support. We then conclude the paper describing our plans and future work.

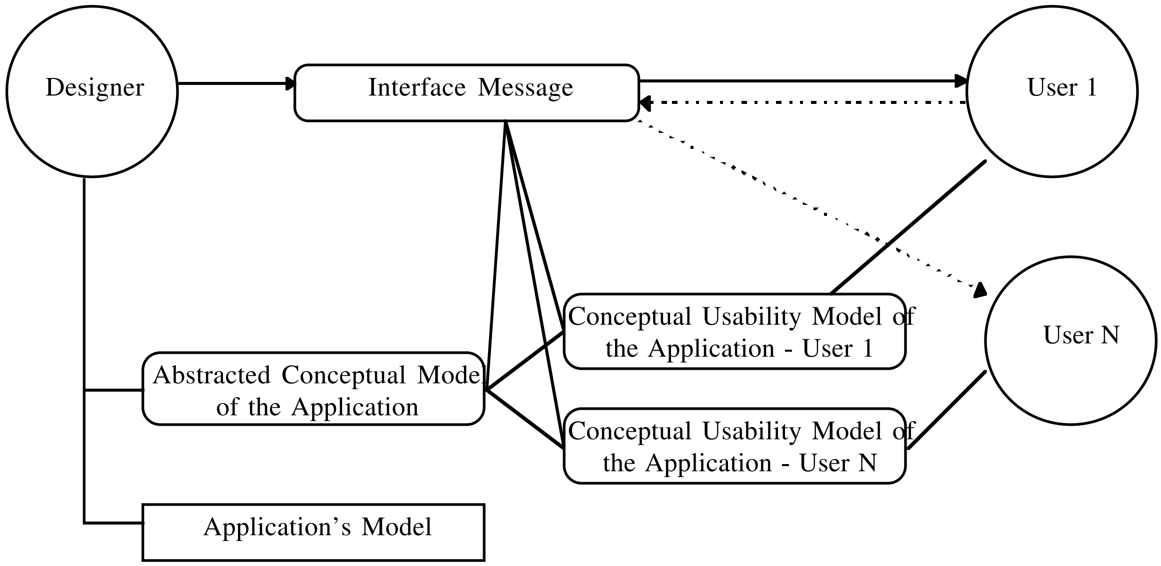

Figure 1: Semiotic Engineering Framework adapted to support MultiADD application

MULTIADD EXPLANATION MODEL

Our research is embedded in an existing multiagent Active Design Documentation tool [8] which consists of 20 collaborating knowledge based systems dedicated to designing offshore oil platform process plants. Each knowledge system is MultiADD's artificial counterpart of an engineer who uses the system to design and document his solutions and rationale for the tasks at hand. MultiADD offers its users a variety of resources such as:

1. a knowledge-based environment for design
2. a knowledge-based explanation environment
3. an annotation database system
4. a knowledge acquisition tool
5. a database query system

Following Moore's requirements for fidelity [14], our Explanation Environment presents MultiADD's knowledge bases and reasoning strategies, along with each subsystem's Design History and User Annotations. A typical explanation in MultiADD is depicted1 in Figure 2.

Users can ask a set of pre-selected questions about design rationale (at this point: What is (the value of this)? Why (this value)? Why Not (some other value)?), which are answered in textual form (natural language - NL). The NL text is meant to meet three major goals: (a) to answer the question; (b) to teach users about the other explanatory resources derived from the system knowledge bases (for example, the Dependency Network graphic); and (c) to support negotiation by helping users understand each other's scenarios in decision-making. Rhetorical Structure Theory (RST) [12] is used to organize content into coherent intertextual representations [9, 10] and to provide varying strategies of communication according to the goal an explanation is striving to meet (see Figure 3). For example: If an Explanation is provided during conflict resolution, the text structure should favor conciliation among the contending parts, whereas if it is provided during individual design the focus of infomation should be centered on constraints and criteria leading to better or worse solutions.

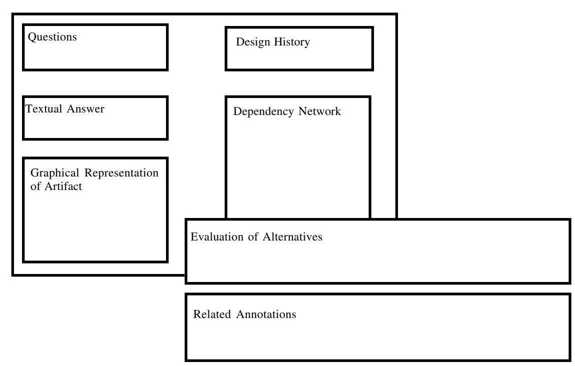

Figure 2: Explanation Resource Interfaces in MultiADD

1All examples in this paper are only layout schemas of the actual screens of the system because of language problems. MultiADD is fully implemented for Brazilian users and all textual elements are in Portuguese.

MULTIADD EXPLANATION SYSTEM AND THE SEMIOTIC ENGINEERING FRAMEWORK

AI technology is critical in terms of maintenance because of its potential for transforming the nature of work in a relatively short term. As soon as work is transformed, new knowledge and requirements emerge, generating a heavy design backlog for maintenance teams. Moreover, the tight connection between typical KB system modules causes a propagation of updates throughout the whole system. For instance, when new kind of knowledge is acquired, a whole new interface for it is required - a knowledge acquisition, an explanation, and a problem-solving interface.

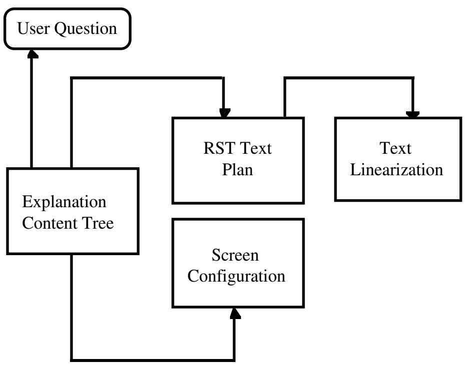

Figure 3: General Schema for Generating an Explanatory Text

User-centered explanation design is typically geared by a thorough analysis of users needs and knowledge. From it, designers derive a set of alternative explanatory patterns that are meant to clarify a predicted set of doubts and misconceptions. Whenever extensions to this set are needed, new explanation strategies as well as new question formulation patterns must be included in the system.

In order to face the maintenance challenges above, a Semiotic Engineering approach shifts the focus of attention from a user-system to a designer-user communication framework. In it, designers are aware that their major goal is to "teach" users two basic things: (1) how and why they think their software meets major user needs; and (2) how to express and interpret messages about the system's model in a given designed language. Goal (1) is typically met by resorting to explanatory text and other graphic representation that discloses the system's model and reasoning (and its potential range of applications). Goal (2) is achieved by a careful selection of cohesive communicative codes and modes that can be understood and wielded by system users.

This change in focus may contribute to postpone (or perhaps eliminate) some of the needs for immediate maintenance, since users may themselves do some sort of "knowledge mining" in search of a way to solve unpredictable (or at least unpredicted) usage situations. Thus, explanations in KB systems become first and foremost an expressive resource for system designers, which users will resort to in a "virtual dialogue" with another human agent, rather than with an artificial agent.

In MultiADD the role of explanations is to tell users how the software works and how it can help a designing team achieve an efficient and effective product with lower costs. For this, MultiADD explanatory resources provide them with the ability to see:

(a) what is the value of the various parameters involved in the design of process plants

(b) when the parameter value was assigned and by whom (human user or expert system)

(c) when and where conflicts have arisen and how have been resolved (if at all)

(d) why a certain parameter has a certain value (logic of decision-making)

(e) how certain decisions compare to certain other alternative decisions about the same parameter

(f) the graphic representation of the process plant for a given sub-system of an oil platform

(g) annotations made by other design team members on specific decisions, alternatives, or deviations from expected values and configurations

Explanatory text is designed to take all the available opportunities to relate the various knowledge sources in order to help users understand the global functioning of MultiADD. Thus, a typical question-answering session might go on as following:

> User: Why not parameter X = val(X)?
>
> MultiADD text: Although MultiADD would also derive val(X) for this parameter, the user accepted and imposed val(Y) instead of val(X) in order to resolve conflict with user of subsystem Z. Search Design History for the moment when this decision was made and know more about its context and impact.

or

> User: Why [is this subsystem design] based on Platform X?
>
> MultiADD text: Because the user accepted and imposed Platform X in order to resolve conflict with user of subsystem Z. However, MultiADD would have chosen alternative Platform Y, as can be seen in the Evaluation Table. MultiADD's decision is based on parameters shown on the Dependency Network.

Explanation Content is decided by a process illustrated in Figure 4. Depending on the context in which a request emerges, MultiADD Explanation System decides on the types of information that an answer must include and fits this context into a Rhetorical Structuring environment of the kind shown in Figure 3 above.

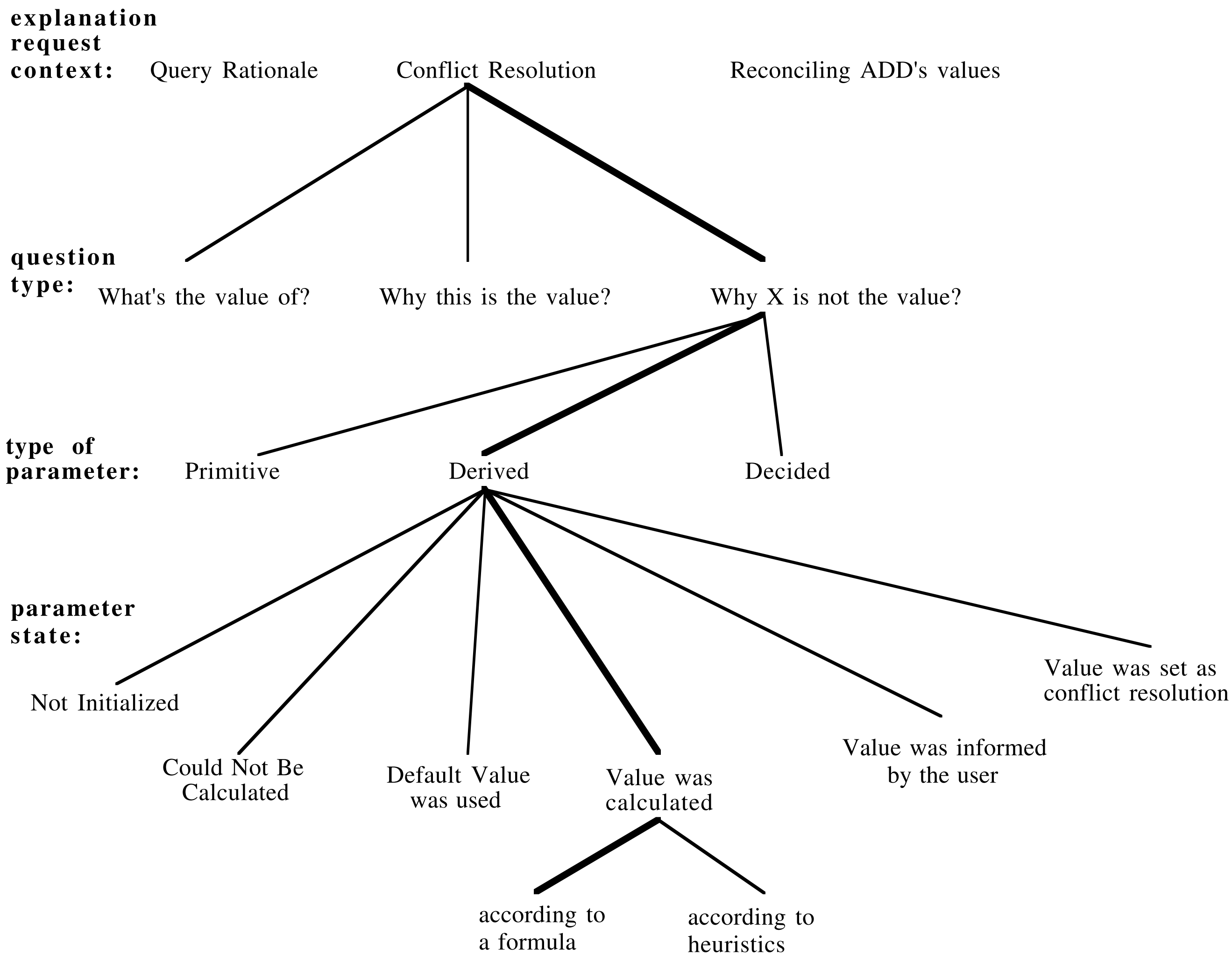

Figure 4: An Example of the Decision Tree for Deciding on Explanation Content

ROLES OF EXPLANATIONS IN MULTIADD

As discussed in the previous section, MultiADD's designers use Explanations to convey to MultiADD's users all the knowledge they judge necessary to warrant extensive usability to the system. Thus, the language they choose to express such knowledge is a crucial resource in the application. We mentioned before, Natural Language is complemented by graphic representations and visual diagrams of various kinds, so that the pertinent abstractions are induced.

When the overall communicative environment is well-designed, abstractions and representations appearing on interface screens evoke correct and useful interpretations on the minds of the users. As a result, productive performance is achieved and a natural "abstraction track" emerges to help users keep focus on a small amount of issues, carefully selected from the context of usage by the software designers themselves.

For instance, although MultiADD offers users a very wide spectrum of knowledge and information about design rationale [8], excessive offer may cause loss of focus and understanding. When trying to evaluate why a certain parameter value is not the one he/she expected, a user may not know where to find the relevant data that directly caused divergence [7]. Browsing the Design History may lead the user off the right track and give rise to "false issues" and misconceptions.

When explanatory text and rhetorical multimode organization of knowledge and information is present, the user's attention is driven to a certain point or region of interest from which he/she may eventually proceed to other knowledge sources. However, the initial focusing strategy determines this interpretation process in that it sets a priority of associations and inferences that was motivated by the MultiADD's designer (for purposes of teaching the user the designer's intellectual model of the software tool).

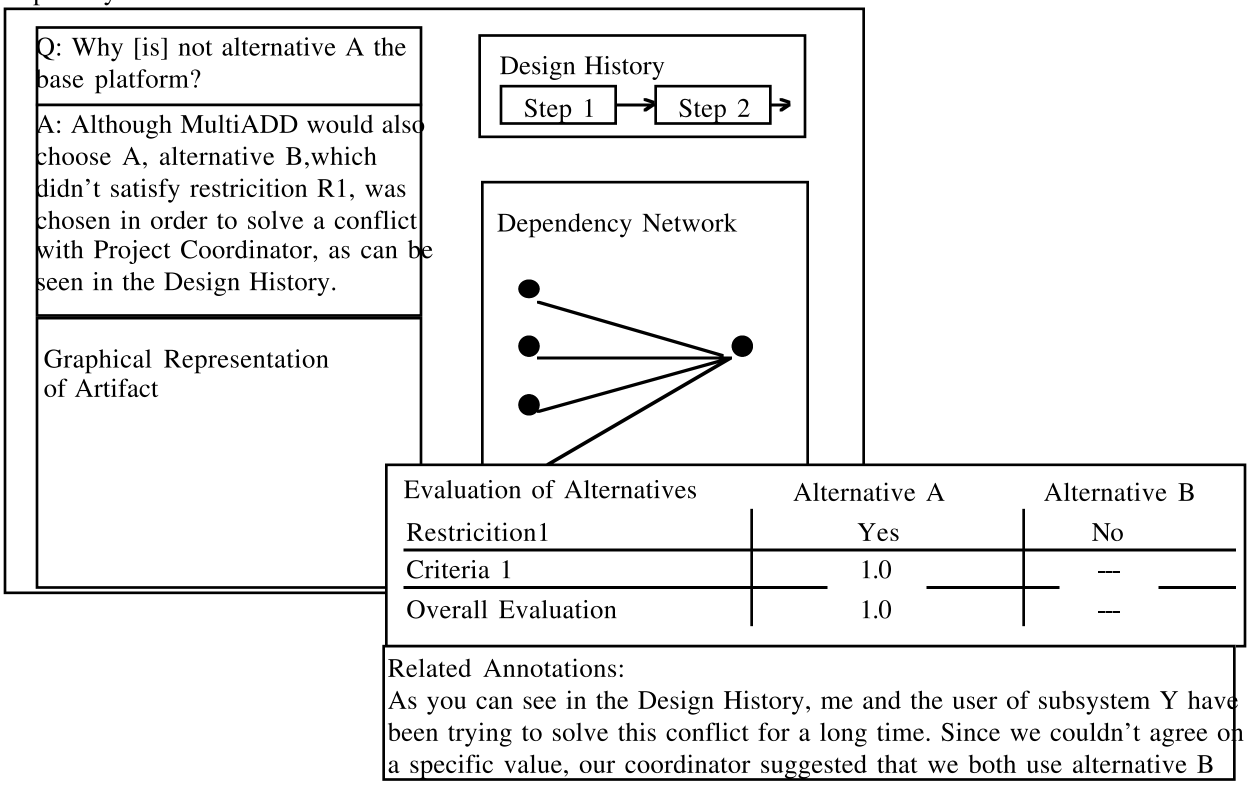

Figure 5: MultiADD's Explanation System, complemented with user annotations

In multiagent environments, focusing is a major issue. Coordinating a team of people around a common goal involves a lot of focused communication and management of divergences from the points being discussed. Although MultiADD does not mitigate conflicts alone (i.e. does not resolve who is right or wrong about decisions made), it does provide an enriched communication platform for conflict mitigation. The central resource for conflict mitigation is precisely its explanation capacity.

Our assumption is that cooperative behavior is brought up if team participants are encouraged to understand each other's rationale and to keep focus on key issues. Explanations provide a means for engineers to understand each other's decisions in that any one participant may ask **why** and **why-not** questions to the artificial representative (actually the Active Design Document generated by the queried user) of his/her colleague without bothering him/her (if not needed or only when needed). It is hoped that this feature will promote mutual understanding and optimize face-to-face communication.

An important design issue in the Explanations System is the language in which explanations are requested and provided. As in every artificial language, arbitrary concept clusterings and abstractions are embedded in its syntax and semantics. Thus, the communicative code itself prevents users from slipping into "undesirable" paths and details. What is "desirable" or not is decided by the explanation language designer in terms of factors that improve cooperation (instead of competition) and focus.

Because MultiADD designers may have made bad choices along the way (i.e. users may actually need to express and communicate ideas that are not accounted for by the provided linguistic environment), an annotation tool is offered as a complement to MultiADD's Explanation facility. Engineers may annotate a series of decisions and rationale they do not see how to express in the limited set of abstractions automatically computable by expert systems. These annotations are incorporated in MultiADD's explanatory texts, as in the following.

The power of explanations, allied to annotation resources, is thus to constitute a means of focused and perspicuous communication between agents. Whenever agents must negotiate about their decisions, thay may send to each other a piece of communication that is - itself - the rationale (explanation) for their stand in the issue. This is a way to help negotiators ponder about each other's motivations and reasons to commit to certain decisions, as well as a means about the impact of requests for changes upon somebody else's project. Most of all, the observation of exchanged communications that include explanations about why stands are taken may reveal important patterns of negotiation in team work.

The present model has been recently implemented in a real application and a project is just starting to be developed in this new technological environment. Thus, our proposed model has no empirical evidence of its impact upon the workflow. Nevertheless, the ideas presented here all derive from theoretical knowledge from the fields of Natural Language Explanations [3, 11, 13, 14] and Computer Semiotics [1, 2, 4, 6]. We firmly believe that a change in perspective may give rise to interesting alternatives in computer-human interaction, especially in applications that involve group work and design.

CONCLUSIONS AND FUTURE WORK

From a Semiotic perspective, Explanation enables users to comprehend how a piece software works and consequently helps them to evaluate the designer's abstracted conceptual model of the application (his/her *interpretant,* in semiotic terms). Validation of the system may be accomplished through explanations, by spotting misconceptions in the designer's application model or a need to choose better means that can express his/her message to users.

Besides, explanations offer a communication *language* for users. The language supports argumentation and conflict mitigation as long as it provides enough information to permit other users to realize the context and rationale that led to the conflicting situation. We believe that this new working environment will enhance productivity (of each individual user and of the group) as well as increase collaboration among the team members.

We are now beginning to test the system in real use situations in order to validate our hypothesis. We will also be evaluating usage of the Annotation System, since it reflects what could not be represented in the Explanation language provided.

We plan to study Design Histories of different projects for we believe that they can show design strategies applied by our users as well as reveal aspects of system potential (for example whether a user feels comfortable with automatic generated values or if he/she always informs values, no matter what the system suggests).

REFERENCES

[1] Andersen, P. B. (1990) "A Theory of Computer Semiotics". Cambridge. Cambridge University Press.

[2] Andersen, P. B. (1993) "Computers as Media". Andersen, Holmqvist and Jensen (Eds.). Cambridge. Cambridge University Press.

[3] Cawsey, A.(1992). “Explanation and Interaction”. Cambridge Mass. MIT Press.

[4] de Souza, C. S. (1993) "The Semiotic Engineering of User Interface Languages". International Journal of Man-Machine Studies. No. 39. pp. 753-773.

[5] de Souza, C. S. (1994) "Testing Predictions of Semiotic Engineering in Human Computer Interaction". Anais do Simpósio Brasileiro de Software”. Curitiba, Pr. Oct’94. pp. 51-62.

[6] de Souza, C. S. (1996) "The Semiotic Engineering of Concreteness and Abstracteness: From User Interface Languages to End User Programming Languages". Dagstuhl Seminar on Informatics and Semiotics. Schloss Dagstuhl.

[7] de Souza, C. S. & Garcia, A. C. B. (1994) "Towards a Rhetoric of Active Design Documents". Proceedings of SBIA'94. Fortaleza, Ce. Oct’94. pp. 523-534.

[8] Garcia, A. C. B. (1992) "Active Design Documents: A New Approach for Supporting Documentation in Preliminary Routine Design" PhD Dissertation. Stanford University. Stanford,Ca.

[9] Garcia, A.C.B. and de Souza, C.S. (1995) “Rhetorical Structures in the Delivery of Design Intent”. Proceedings of. ASCE’95. Atlanta,Ga. Jun’95. pp 1443-1452.

[10] Garcia, A.C.B. and de Souza, C.S. “ADD+: Including Rhetorical Structures in Active Documents”. Forthcoming.

[11] Mellish, C., Dale, R., Zock, M. (1990) “Current Research in Natural Language Generation”. London. Academic Press.

[12] Mann, W. C. and Thompson, S. A. (1987) “Rhetorical Structure Theory: A Theory of Text Organization”. Information Science Institute. University of Southern California.

[13] Maybury, M. T. (1993). “Intelligent Multimedia Interfaces”. Menlo Park, Ca. AAAI Press / MIT Press.

[14] Moore, J. (1995) "Participating in Explanatory Dialogues: Interpreting and Responding to Questions in Context.". Cambridge, Mass. MIT Press.

[15] Norman, D. A.& Draper, S. W. (1986). "User Centered System Design". Hillsdale, NJ. Lawrence Erlbaum.