Rhetorical Structures in the Delivery of Design Intent

Ana Cristina Bicharra Garcia and Clarisse Siesckentus de Souza

Abstract

Active Documents are powerful tools for capturing design intent. However, delivering design rationale is a challenging task. Current implementations of Active Documents are deprived of global rhetorical organization. Thus, users are prone to misinterpreting the relations that hold among pieces of knowledge on the screen, and to subestimating the system's actual potential. In this paper we show how the addition of rhetorically organized explanation plans positively impact the delivery of design intent.

1. Introduction

The design process of an artifact can be roughly divided into two major phases: planning and construction. This process can be carried out by an individual agent or by a team of co-agents who collectively contribute to the final achievement of a design object (architects, structure engineers, etc.). Once the artifact is achieved, its life cycle enters a maintenance process in which any number of individuals whom we will call post-agents evaluate, extend or re-design the artifact to meet novel needs.

Crucial to the artifact's design and maintenance processes is a communication infrastructure — a medium, a common code, and an effective channel. In it, all agents must be able to encode their own ideas and decode those of the others, in an effort to understand, discuss, engender, and possibly change the artifact. Although the artifact's plan is the main theme of messages exchanged by team participants, perceiving design intent (rationale) plays a major role to ensure truly cooperative behavior among co-agents and effective design understanding for post-agents. Sometimes design intent can be inferred from decisions captured in the design plan (design documentation). However, inferences may be drawn by people who are not the original decision-makers, but their interpreters. Consequently,

1Universidade Federal Fluminense, Departamento de Computação, Praça do Valongoinho s/n, Niterói, RJ — Brazil (bicharra@inf.puc-rio.br)
2Pontifícia Universidade Católica, Departamento de Informática, Rua Marquês de São Vicente 225, Rio de Janeiro, RJ — Brazil (clarisse@inf.puc-rio.br)
intent capture is prone to misconceptions caused by either the author's faulty expression of the design plan, or by the reader's faulty interpretation of it.

Powerful design documentation tools have been proposed, among which is the Active Design Document — ADD [Garcia'92]. ADD is a dynamic computational representation of design rationale based on an adjustable underlying model of the device conception process. The system captures and delivers design intent thus providing the necessary design communication infrastructure. The system becomes the medium, the interface language the code, and the computer the channel. The implemented system generates information that can be displayed as an explanation, but it lacks rhetorical organization and structuring.

Delivering intent at the interface level is a complex task, involving strategies that fully explore documentation resources and adequately meet users' cognitive needs. The ADD-HVAC system generates information that can be displayed as an explanation. However, co-agents and post-agents by themselves cannot be expected to (1) explore the full wealth of knowledge resources available in an active document, and (2) diagnose their own misconceptions about the design model and history.

In the present paper we propose the introduction of a rhetorical organization in Active Documents. We suggest that using Rhetorical Structure Theory (RST) in design rationale delivery improves the understanding of designers' intent, and thus the global cooperative behavior among agents.

2. Current Active Document Abilities in Delivering Design Intent

We illustrate ADD's current potential to deliver design intent by means of an example drawn from an application that uses ADD to support design and documentation in an oil company. Off-shore platform design involves a number of different knowledge areas such as Structure, Electric, Mechanic, and Architecture disciplines. The company is in charge of developing the conceptual design for production and exploitation platforms. The company's conceptual design concepts are further used as the basis for the whole bidding process. Production field data are the input for initializing the conceptual design phase.

Field data are not static, however. As the exploration process evolves, new data keeps coming into the design process, making it a totally dynamic activity. Moreover, the highly concurrent sub-tasks cannot be held isolated from each other. Decisions in one, typically affect the other(s). An example of this situation is a need to decrease platform space, experienced by architects, leading to different solutions for the Ventilation and Air Conditioning systems (VAC). Agents involved in both tasks must have a model (even if imprecise) of each other's discipline, in order to manage the impact of their decisions. Additionally, the whole conceptual design process is a time-critical one, which often leads to insufficient model-checking interaction among agents.
Active Documents have been developed to address issues related to the lack of access to design rationale, enabling agents to have better understanding of co-workers activities, as illustrated in the following example.

Figure 1: Co-Agent #2's request for Co-Agent #1's rationale

Co-Agent #1 is ADD's representation of the Mechanical Engineer, responsible for a VAC Project. Co-Agent #2 is the Architect, responsible for the platform's layout. Data-Provider #1 is a representative of the owner's team, who provides the field data, and Post-Agent #1 is one of the Bidders. Except for Co-Agent #1, who is an artificial agent, all other agents are ADD's users, in a variety of interconnected situations.

Figure 2: ADD's response to Co-Agent #2's request

Co-Agent #1, based on layout design, compartment functionality, heating loads, and airflow, has decided on the quantity and location of VAC equipment serving the various compartment groupings, and housed the IT in the available 3 VAC rooms. Co-Agent #2 is worried with the platform's
dimensions and cost and decides to reduce the number of compartments. Considering the total required capacity, she thinks it is possible to eliminate one of the VAC rooms. However, Co-Agent #2 is not sure that Co-Agent #1 can reallocate the needed equipment in the two rooms left. In Figures 1 and 2, respectively, we see Co-Agent #2’s input and ADD’s output.

The output screen shows that because of this change, “Required Number of VAC Rooms” cannot be calculated. Also, restriction “Available Equipment Space ≥ Required Equipment Space” is not satisfied. Browsing ADD’s interface, Co-Agent #2 infers that if she increases one of the VAC rooms area, she may solve the problem.

![Dependency Graph and Design History]

**Figure 3: Redesign impacts after input data changes**

Later, Data Provider #1 sends the latest input data modifications, which decreases the platform crew capacity. Co-Agent #2 receives these modifications and reconfigures the platform layout, by decreasing the number of cabins and WC’s and the cafeteria space. Co-Agent #2 worries about the impacts of this change on “Required number of VAC Rooms”. Co-Agent #1 regroups the compartments, re dimensions and reuses the equipment, engaging in an entire re-design process. ADD’s screen in Figure 3 shows that “Required Number of VAC Rooms” doesn’t change, but that the redesign has led to many other changes, as read on the Design Impacts area.

Some time later, Post-Agent #1 analyses the conceptual VAC model and supposes that the apparent unbalanced dimensions of the equipment are due to bad design solutions. In order to find out what’s happened, he can either probe ADD for an entire decision justification, or concentrate the probing on precisely those aspects that he knows would lead to another decision. During exploration of Co-Agent #1’s design rationale, he discovers
that the unbalance is intentional, and is there to compensate for heavy weight on one side of the platform (see Figure 4).

The above views of the interaction between ADD's and its users serves to show that, although the needed information is there, there is a great deal of inferencing about the connections that hold among the various pieces. Besides, interpretations about the real meaning of ADD's summary phrasing on text fields and labels can be quite widely different from each other. So, in order to reduce the cognitive load and the potential for misinterpretation, ADD's rationale delivery interface has been redesigned and called ADD++.

Figure 4: Post-Agent #1's probing results.

3. Background on Rethorical Structuring

As seen in the previous section, ADD's current interface includes text and graphics screen objects that can be manipulated by users via menu selection, mouse clicks, or window scrolling. Interface events are only loosely connected to each other, which results in potentially uncohesive screen displays that misunderstand users in search of design documentation.

Structural integration of the various modes of information delivery in ADD is needed, in order to turn information into knowledge. Although the relations between knowledge pieces can be treated at display state level only — for example, the Design History display can be automatically scrolled up or down as a parameter is selected in the Dependency Graph — this mechanism is still inadequate to express all of the causal relations that hold among the pieces. Not even the animation of parameter changes along a
temporal line on the Dependency Graph links and nodes could be expected to successfully get the message across to the documentation user. We have then incorporated an additional expressive mode to ADD's interface design—natural language text. In it, perceived user's cognitive needs may be met by means of text planning techniques that maximize user's understanding abilities [Scott & de Souza'90].

Rhetorical Structure Theory [Mann & Thompson'87] has a number of attractive features for text planning. In particular, it is a belief-driven approach in which rhetorical relations between spans of text are specified in terms of communicative goals achievable through their inclusion in a message. This provides a powerful resource for reasoned information selection in text generation systems [Hovy'90, Scott & de Souza'90, Hovy'93, Moore & Paris'93]. By knowing the effect of a chosen relation on the reader's knowledge state, hypothesized sources of misconceptions may be directly addressed by a combination of remedial knowledge transmission in cohesive and coherent textual form.3

![Diagram](image)

Figure 5: ADD interface resources for text generation

The text generation process adopted in ADD [de Souza & Garcia'94] follows a traditional sequential approach [MacKenzie'85] where text is first planned and then realized. Text planning in our application amounts to deciding what to say and when to say it based on (a) the dialogue flow (perceived focus and intention for information retrieval), (b) the system's knowledge base (inferred goals for documentation use), (c) screen layout data (reference to alternative modes of content representation), and (d) a mapping of (a), (b) and (c) contents onto RST schema. Text realization amounts to deciding how to say it by means of local morpho-syntactic rules that are necessary to instantiate a rhetorical schema (enriched by information items) as a natural language text span (see Figure 5).

3For a specification of RST relations, see Figures 6, 7 and 9 in Section 4.
The proposed architecture supports the generation of textual explanations, featuring a number of references to non-textual resources in the interface. In Section 4, we present a detailed example of the effects of rhetorical structuring on the quality of design intent delivery.

4. Rhetorical Structuring in ADD+’s Explanation

In Section 2, we have presented a sequence of interactions for retrieving rationale. We now contrast it with ADD+’s style, where textual explanations are provided. In previous work (de Souza & Garce’94), we have discussed more extensively aspects related to the determination of focus and intentions in user input. Here, we will concentrate on text planning aspects related to ADD+’s output.

Figure 6: Explanation Plan for based on RST-Schemes (first interaction)

Going back to the example in Section 2, we saw that Co-Agent #2 asked what would happen to the required number of VAC rooms if the available rooms amounted to 2 instead of 3 (Figure 1). In addition to ADD’s response (see Figure 2), ADD+’s Interface output includes natural language text generated from the explanation plan presented in Figure 6. The linearized textual form is equivalent to the following span:

(The) Required # (of) VAC rooms has no value if (the) Available # (of) VAC rooms equals 2 instead of 3, because Relation Available Equipment Space ≥ Required Equipment Space doesn’t hold.

Note that the text above makes explicit references to specific parameter changes in the underlying design model (e.g., relations that do not hold). In ADD’s current interface, the information is there, but passive. The user may realize it is important or not. But in ADD+, the text emphasizes critical knowledge to be delivered. This is done, as we said, by a mapping from types of questions onto types of text plans for NL answers.
Further on, we saw that Co-Agent #2 had to redesign the platform's layout because of changes in the field date. The textual complement to the output screen on Figure 3 is generated from the Explanation Plan shown in Figure 7, and could read as follows:

Although there are impacts for the overall design (see Design Impacts Area on the screen) if the platform layout is such, (the) Required # (of) VAC rooms is satisfied.

Continuing the example, we saw that Post-Agent #1 suspected of a mistake in the design process because of an unbalanced distribution of weight on the platform. In Figure 4, we saw the probing interaction for all details related to the Selected Equipment. The possibility of further choosing one among all of the related details has been omitted in the example. However, with a special interactive event, the user could have specified that he was interested in the equipment's weight (focus). In Figure 8, (a) presents the Explanation Plan for
the unfocused answer, and (b) the plan for the focused text. The latter plan’s readout is equivalent to the following:

(The) Selected Equipment is such because of the conditions presented in the Local Evaluation Area on the screen, although the Selected Equipment Weight Value is 20 tons.

Note that the underlined text above is the realization of a pre-defined reference schema to items appearing in the menu-driven part of the interface. The goal of such reference is to make the user aware of further possibilities in exploring the knowledge contained in the model.

5. Expected Improvements on Rationale Delivery

Text spans presented in the previous section capitalize on a number of resources: Screen Layout (e.g., references to valuable information available in specific screen areas), ADD’s Design Knowledge Base (e.g., clarification of critical relations impacting the satisfaction of design model conditions), and RST Schemes (e.g., the rhetorical organization of various information sources into one cohesive text). Dialogue Flow contributions to the ongoing textual interaction basically amount to the impacts of topic continuity (discussed in de Souza & Garcia’94) for the generation of more cooperative explanations in extended user-application sessions.

A comparison between ADD’s and ADD+’s interface resources shows that the system’s power is considerably increased by explicit references to its own resources and reasoning methods. Thus, details that might go unnoticed in ADD’s interface, preventing users from drawing the correct inferences about the designer’s intent, are brought forth in ADD+. Certainly, this style of interaction increases the usability of knowledge resources in Active Documents, since users become aware of the system’s own scale of valuable information.

RST gives us strong evidence as a powerful tool for rhetorical organization both in textual and non-textual output items [Maybury’83]. Besides serving as the basis for the generation of natural language texts, it also guides the automatic reconfiguration of screens according to relevant content relations among the various pieces of information.

At the present time, ADD+’s redesign includes mapping rules from a contextualized typology of questions onto a set of pre-defined Explanation Plan Schemes. The next steps in its implementation include a linearization module for Brazilian Portuguese and the presentation of dynamic links between words and phrases in the explanatory text and their related items or events in the Design History, Dependency Graph, Local Evaluation, and Design Impacts Area on the output Screen.
6. References


