

ECHOES OF SEMIOTICALLY-BASED DESIGN IN THE DEVELOPMENT AND TESTING OF A WORKFLOW SYSTEM

Clarisse Sieckenius de Souza¹, Cecília Kremer Vieira da Cunha¹,

Raquel Oliveira Prates^{1,2}, Simone Diniz Junqueira Barbosa¹

¹Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil

²Departamento de Computação, Instituto de Matemática e Estatística, UERJ, Brazil

ABSTRACT

Workflow systems are information-intensive task-oriented computer applications that typically involve a considerable number of users playing a wide variety of roles. Since communication, coordination and decision-making processes are essential for such systems, representing, interpreting and negotiating collective meanings are a crucial issue for software design and development processes. In this paper, we report and discuss our experience in implementing *Qualitas*, a web-based workflow system. Semiotic theory was extensively used to support design decisions and negotiations with users about technological signs. Taking scenarios as a type-sign exchanged throughout the whole process, we could trace the theoretic underpinnings of our experience and draw some revealing conclusions about the product and the process of technologically reified discourse. Although it is present in all information technology applications, this kind of discourse is seldom analyzed by software designers and developers. Our conjecture is that outside semiotic theory, professionals involved with human-computer interaction and software engineering practices have difficulty to coalesce concepts derived from such different disciplines as psychology, anthropology, linguistics and sociology, to name a few. Semiotics, however, can by itself provide a unifying ontological basis for interdisciplinary knowledge, raising issues and proposing alternatives, that may help professionals gain insights at lower learning costs. Keywords: semiotic engineering, workflow systems, information-intensive task-oriented systems, scenario-based design and development of computer systems, human-computer interaction

INTRODUCTION

Over the past few years workflow technology has been gaining territory in many organizations as a tool to improve coordination and enhance the quality of processes and products. Traditional information systems have been integrated into this technology, allowing users to retrieve information for decision-making, to turn decisions into action, and then to generate more information relative to and derived from these very processes. However, one of the most important challenges associated to workflow technology is to avoid permanent commitments to current organizational practices that can eventually prevent processes from adapting to continuously changing contexts. Research in this field has produced a number of alternative ways to face this challenge [Khoshafian & Buckiewicz (1995), Dourish et al. (1996), Heintz et al. (1999), Malone et al. (1995), Glance et al. (1996)]. As is the case in the field of human-computer interaction (HCI) in general, two major paths to accommodating variety can be identified:

- trying to capture the widest range of work patterns, actual and potential, and affording those to workflow systems users; and
- allowing for customization and extension of implemented workflow systems.

In this paper we will be addressing the first path to designing and implementing workflow systems¹. We will be showing how Semiotics has been used to inform and support the design and development of one particular web-based workflow application — *Qualitas*. The goal of this system was to help maintain the quality of services certified by ISO9001 standards in a large Brazilian company. Our customers expected the system to lead the employees involved in training, consulting, evaluation and coordination activities to abide to a set of certified norms, and to improve the efficiency of the overall process. Their view of this project was that it should suffice to translate certified norms into computerized tools that would generate the same middle and top management indicators they currently used, and they would then reap the benefits of computer-based technology. An instance of such benefits would be the possibility to readily obtain indicators and export them to spreadsheet and database applications, and proceed much faster with decision-making processes.

This view represented an extension and partial integration of information systems network installed in the company, but considerations about the actual work context in which people generated and used information failed to be explicitly formulated and discussed in their initial request for our participation in their project. A consistent analysis of how people would react to their job being exposed to others through the indicators presented in a new technological environment was also not explicitly carried out. Thus, our first and foremost task was to discuss the various dimensions of workflow technology with our customers, so that the application we developed would have a greater chance of achieving their expected goals.

We started by discussing what the new technology would be, compared to the existing information and decision-making systems in the company. Both parties agreed that the technology should be a web-based information-

¹ Our ongoing research about a semiotically-based approach to the design of extensible workflow systems is reported in [Cunha et al. (2000)]

intensive task-oriented system (WITS), a special kind of workflow system. An emphasis on information within the context of organizational tasks was intended to communicate that our aim was explicitly one of *informing* work practices, not of *automating* them.

Subsequent steps involved a lengthy process of negotiation and design, around technologies, organizational goals and practices, and user studies, among others. Throughout this project, Semiotics has guided design and determined the quality of the final product. Three basic concepts in this discipline have been used:

- the peircean sign [Peirce (1931, p. 99)];
- unlimited semiosis [Eco (1979, p. 69)]; and
- abduction [Peirce (1931-5, pp. 151–156)] and pragmatics [Peirce (1931-5, chapter 17)].

In peircean Semiotics, a **sign** is anything that can stand for something else for some interpreter. Although this definition seems to be extremely vague and broad, its very vagueness points to a central tenet of semiotic theory — what makes anything *meaningful* is the fact that there is somebody for whom it *means* something (in fact, anything). Therefore, it is the observer or listener – the interpreter – who establishes the necessary and sufficient conditions for anything to be a *sign* [Eco (1979, p.16)]. For example, in a work situation, everyday before he leaves, a tidy employee may be intentionally arranging his desktop in a certain way, in an effort to *tell* the cleaning staff how he expects to find it in the morning. However, to his disappointment, the cleaning staff doesn't realize there is an intentional arrangement of objects on the desk. But, what is a sign to the employee, isn't one to the cleaning staff, who rearranges everything, everytime. By virtue of the other facet of the Peircean sign, the employee may interpret the different arrangement of his desktop as a sign that the cleaning staff *means* to be irritating, regardless of what is actually the case.

Semiosis is the process of assigning meaning to an observed form or phenomenon. The structure of the Peircean sign involves three elements: the object, the *representamen* (or representation of the object), and the *interpretant* (the ideas, actions, feelings, or other signs triggered by the observation of the *representamen* or of the object). The most relevant feature of semiosis is that it consists of a chain of interpretants, whose elements and length are unpredictable. In the example above, it may simply not be the case that the cleaning staff is trying to make any statement at all about the employee's desktop arrangement. However, the employee may be generating an elaborate and completely subjective interpretation for something that is totally unfounded. As long as interpretations don't depend on objective grounding, semiosis can be out of control for anyone who is generating what is taken to be the *representamen* of the subjective sign.

Finally, **abduction** is a reasoning method that has been previously characterized as *inference to the best explanation* [Josephson & Josephson (1994)]. The gist of this method is to create an explanatory hypothesis for an observed form or phenomenon, and to try and confirm it with reinforcing evidence. As long as there is no counterfactual evidence, the hypothesis holds and every interpretation of the associated phenomena may be taken as a reinforcement of the hypothesis. If at one point observed information challenges the validity of the hypothesis, it is revised and reformulated until another satisfactory hypothesis is found to accommodate the new facts. Carrying our example one step further, suppose the employee has concluded that the cleaning staff is trying to annoy him. This is the meaning he has abducted from observing helplessly his rearranged desk every morning when he arrives at work. However, let us suppose that employee and cleaning staff meet one day, and that they talk to each other. The cleaning staff then tells the employee that they arrange all the desks in the same way, and that they thought they were doing a good job. They are sorry if they have annoyed him, and they will keep his desk as it is from now on. The employee is thus forced to revise his beliefs and conclude that his interpretation (or abduction) of facts was very wrong.

The openness of interpretation in semiotic theory has challenged many semanticists, especially those that take a more platonic view of meaning as is the case in many logic approaches. Nevertheless, Peirce himself was a logician, and the founder of **Pragmatics**, a discipline motivated by the need to explain how and why, despite of unlimited semiosis and the subjective nature of signs, people can achieve mutual understanding and interact productively. At the heart of Pragmatics lies the notion of *negotiating meanings*, given the goals of speakers and interpreters in context. Situations constrain, or if not as much at least motivate, semiosis. Individual and collective goals determine how close or how far from each other speakers and listeners will intentionally allow their interpretants to be. To illustrate a pragmatic principle in action, we can carry our example still one step further and suppose that in the hypothetical dialogue where the cleaning staff is telling the employee why they were rearranging his desk every day, they say that:

"We do this because of Procedure P001C5, which we have been trained to execute in our company."

Unless the employee extensively knows the procedures for cleaning offices, it is very unlikely that he will be able to assign to *Procedure P001C5* the same meaning that the cleaning staff does. But this doesn't preclude them from achieving satisfactory mutual understanding. The employee's vague interpretant of the term (standing for *some norm or procedure that has nothing to do with me personally*) is pragmatically sufficient to help the conversation reach its goal and reestablish a good relationship between these people. Semantically, it can not only be argued, but also probably shown, that these two parties were not talking about the same objective thing [Frege (1892)].

Concepts of the sign, semiosis, and abduction have been extensively used by our design and development team, who chose scenarios as an on-going representation of meanings that should be shared and agreed by customers, users, and developers. It should be clear from this introduction that the *meaning* of technology, of information that flows in it, and even the tasks that can be carried out in computer-supported environments, are not objective for humans, although they are so for symbol-processing systems. In other words, although managers, for instance, may pragmatically generate different interpretants for performance indicators they see on screen at two different points in time, any computer procedure embedded in the system that uses those indicators to produce others will assign to it one specific unchanging (ideal) programmed meaning. And this may affect the workflow in very important ways.

A semiotically-based approach to design and development of computer technology will take these factors in consideration. In our specific case, we will be using different aspects of our own Semiotic Engineering approach [de Souza (1993)] to help designers generate and users interpret the algorithmic meanings [Nake and Gabrowski (to appear)] embedded in WITS. Semiotic Engineering principles that affect the design of WITS can be summarized by the following:

- Interactive software is a one-shot message sent from system designers to system users about the range of interface messages users can exchange with the system in order to achieve a certain set of goals [de Souza (1993)].
- The system is the designers' *deputy* inasmuch as it enacts all and only the conversations that the designers thought they would have with users given their interpretation of who the users are, what they need and want, what they like and prefer [Prates, de Souza and Barbosa (2000)].
- Every interactive system has one unique language that can be used in the communicative exchanges between users and system [de Souza, Barbosa and da Silva (to appear)].
- It is critically important that this uniquely designed interactive language allows users to generate consistent interpretations of interface signs by letting them abduct its fundamental design principles [de Souza, Prates and Carey (2000)].

In the other three sections of this paper we will show that scenarios have been chosen as the basic *representamen* for various semiotic chains that emerged throughout the development of *Qualitas*. We will discuss some aspects of technology-based signs and the contributions of Semiotic Engineering to designing and developing WITS. Our focus will be placed on human-computer interaction, more specifically on the importance of helping users understand technological affordances and designers turn technologies into tools [Adler & Winograd (1992)].

SIGNS AND SEMIOSIS IN TECHNOLOGY

As we have seen in the previous section, technology introduces new signs in the domain. In this section, we will be analyzing the nature of signs and how they are generated, taking into consideration facts about the obsolescence rate of processes, the turnover of employees, and role-switching perspectives on organizational discourse.

WITS signs can be classified into: data, information, and knowledge. This is a traditional categorization, especially in the field of AI [Newell (1982)], but we will highlight some semiotic distinctions between each class.

From the system's perspective, data is the grounding upon which a number of abstractions can be built. It functions as raw material for all information processing inside and outside the system. The representations of data are the focus of important topics in programming languages, namely data structures. Information is precisely what we gain as we abstract upon data. It is dependent on data models that reflect the domain and typically emphasize static relations among entities. Knowledge, however, provides explanations about how information is derived from data. It makes use of an ontology of the domain, from which consistent representations and inference rules are derived. Occasionally, if systems have meta-knowledge, they could also provide explanations about how knowledge itself is used.

In sum, the major semiotic difference between these three classes is that data is an arbitrary symbol whose semantics is outside the system. Information presents an abstraction that can be processed by the system. It is causally dependent on data (which in semiotic terms amounts to an index), but this causal relation cannot be manipulated by the system. At the knowledge level, the system is able to account for the processes that derive information. In other words, at the knowledge level the causal relations between information and data are computational symbols that the system can manipulate in inferential processes. These concepts may be illustrated by the following situation.

Services that do not conform to norm

When training services are provided by employees in this company, an evaluation sheet is passed to the participants who rate the service on a scale from 1 to 5. The certified norms state that the acceptable quality of services is that at least 80% of the ratings fall between 4 and 5. Otherwise, a process of investigation and reporting non-conformity to the standards must be started.

In this case, the ratings are the data, the acceptability of the service is the information, and the remedial action that is required is knowledge about work practices. In *Qualitas*, we have these three levels of signs. Employees can inspect the individual ratings and be informed if there is a problem with the quality of services. Whenever there is, the system itself triggers a series of actions that will lead to a decision relative to the problem that occurred. We can see that the complexity of the signs increases from a single number to a whole sequence of interactions and commitments.

In systems without the knowledge level, users may have the diagnosis of the problem, but they will have to decide what to do and take action themselves. If the system doesn't even have the information level, then users may fail to diagnose the problem altogether.

Another interesting aspect of the above classification is related to symbol grounding. All the data imported or input is computationally unprovable from the importing system's perspective, in the sense that the system takes it as true. On the one hand, although users may know that the data was derived from an elaborate causal chain of facts, as far as the system is concerned, it is an axiom. On the other hand, all the information the system derives from the data is computationally provable and, as long as the system has a knowledge level, it is also explainable. As a consequence, in knowledge systems, information is semiotically an index (a sign that has a contiguous or causal relation to its referent) that can therefore be negotiated, albeit in a limited way, among users and systems.

Obsolescence of signs

One of the best-known problems of knowledge systems is that the cost/benefit ratio in certain domains is not acceptable. This is due primarily to the lack of established knowledge or practices. If knowledge isn't consolidated and is prone to "quantum leaps" as people gain new insights about what they're doing, the maintenance cost of knowledge systems isn't compensated by the benefits of its use. The rate of obsolescence is typically high. The benefits of using knowledge systems are only reaped when there is maturity and tradition in the domain. In this case, the bulk of knowledge is considerably stable, and the evolution is smooth. This allows designers to introduce knowledge acquisition and/or machine learning facilities in the system, which will ensure longer term usability of the system's knowledge base.

From a semiotic perspective, non-consolidated domains present a high rate of sign obsolescence, whereas consolidated domains provide the basis for more long-lasting grounding of signs and meaning-negotiation processes. It may be argued that explanations are relevant for a fixed set of users only during a limited period of time. Once they know why the system generates certain kinds of information, they no longer need clarifications, i.e., the knowledge becomes tacit. Nevertheless, from an organizational perspective (and even as an inherent characteristic of certain domains), the set of users is typically not fixed, due to the turnover of employees. In this situation, explanations are crucially important to support training processes and to help newcomers acquire knowledge not only about the system, but about current practices as well. In semiotic terms, tacit knowledge is typically knowledge that needs not be abducted, which in Artificial Intelligence is often referred to as the common-sense knowledge of a certain community.

Experimenting the *alter semiosis*

One of the greatest problems in human communication is the fact that we often misjudge how we are going to be interpreted by our listeners in conversation. We hope to make ourselves understood, but some facts that determine interpretation are out of our reach. In human conversation, there are a number of mechanisms that help us remedy misconceptions and misunderstandings that may arise. However, in medium and large organizations, the very structure of the company makes it difficult, if not impossible, for employees to have a clear grasp of how their work practices and products are being interpreted by others.

A great advantage of WITS technology is that it contains a model of all the roles and group structures that are involved in organizational processes. As long as these models can be inspected, they allow an employee assigned to a certain role to experiment other roles (in demonstrational or educational simulations). He may realize how the kinds of signs that are generated by him when performing his assigned role are being viewed in a different context. For instance, let us suppose that his productivity indicators are exceptionally good, but that his colleague, whose indicators are not as good, is the one who gets a promotion. Locally, this seems unfair to him.

However, if he could play the role of a top manager, he would perhaps understand that, globally, his colleague's performance has had a greater impact on the company than his own.

We refer to this possibility as experimenting the *alter semiosis*. Outside the technological setting, the success of this experimenting is strongly dependent upon someone's abilities of detached observation, that is, of not letting his own biases play a distorting role in interpreting reality. With technology, we may reduce these distortions by providing enough information to engage users in someone else's role. We hope that the technological protocol that derives from modelled organizational practices provides a large volume of evidence to motivate *semiosis* that is consistent with the underlying models.

FROM DESIGN TO TESTING: SIGNS AND SCENARIOS

Scenarios are stories about people and their activities (Carroll (2000), Imaz & Benyon (1999)). They can be used throughout different stages of system design and development. What they contain, and sometimes lack, reflects a state of our own understanding of what the users do and want. It also keeps the focus in meaning negotiation processes.

Technological proposal scenarios

At the early stages of the project, the goal was to explore and discuss the potential of the technology. After an introductory study of the domain (mainly through interviews and readings of existing documentation), technological proposal scenarios were created. They were a mock-up of our own vision about the future system, simulating a few activities considered relevant by the user community.

This introductory level allowed for the creation of an initial set of shared meanings. These scenarios also supported the definition of the scope of the system, which involved a negotiation of the users' priorities in combination with technological constraints and organizational planning.

At first, we mapped almost directly the existing paper forms onto HTML forms. The idea was to capture and record the data input in the various fields into a global database, from which indicators would be generated for management. However, at this very stage, some technological signs were introduced, as we can see in Figure 1. In the first proposal we had a 1-to-1 mapping from paper to HTML form. The OK and Cancel buttons were technological signs introduced at this stage, and the OK button led users to visualize the complete filled-out form. This form's layout was a *replica* of the original paper form, in order to help users connect their new experience with their familiar practices.

Soon after that, technological constraints played a major role in leading us to a second proposal. The original paper forms contained too much information to fit in an interactively appropriate HTML form. If the first proposal were implemented, users would have to scroll through excessively long pages. Because of this interactive restriction, we split the single HTML page into a number of pages. However, users then needed contextual information from one page to another, and also needed to be informed about the progress of the work. Notice that, in the second proposal, the Name and Client fields from the first HTML page are copied as text to the next. Moreover, a third technological sign, the View button, was explicitly introduced to allow access to partially filled-out versions of the original form.

These scenarios try to capture not only the gist of work practices as they are currently performed and as they should be performed in the new technological *medium*, but also the kinds of additional knowledge users should be prepared to acquire. Specifically, we conveyed in the technological proposal scenarios the idea that training users in web-based systems would be the minimum requirement for the whole project to succeed. This was important because the kinds of technology used by each individual in the community of users were very different.

But because the technological proposal scenarios were typically run by one presenter for an audience of representatives of our customer, the alternation of roles and users (required for a full-fledged computer-supported collaborative system implementing those ideas) was not explicitly included in the mock-up as a designer-generated sign in the technological language under examination. That is, in many occasions the presenter would *tell* the audience, as she moved from one screen to the next, that the person who would be seeing the next screen was a manager, an employee or a company's client.

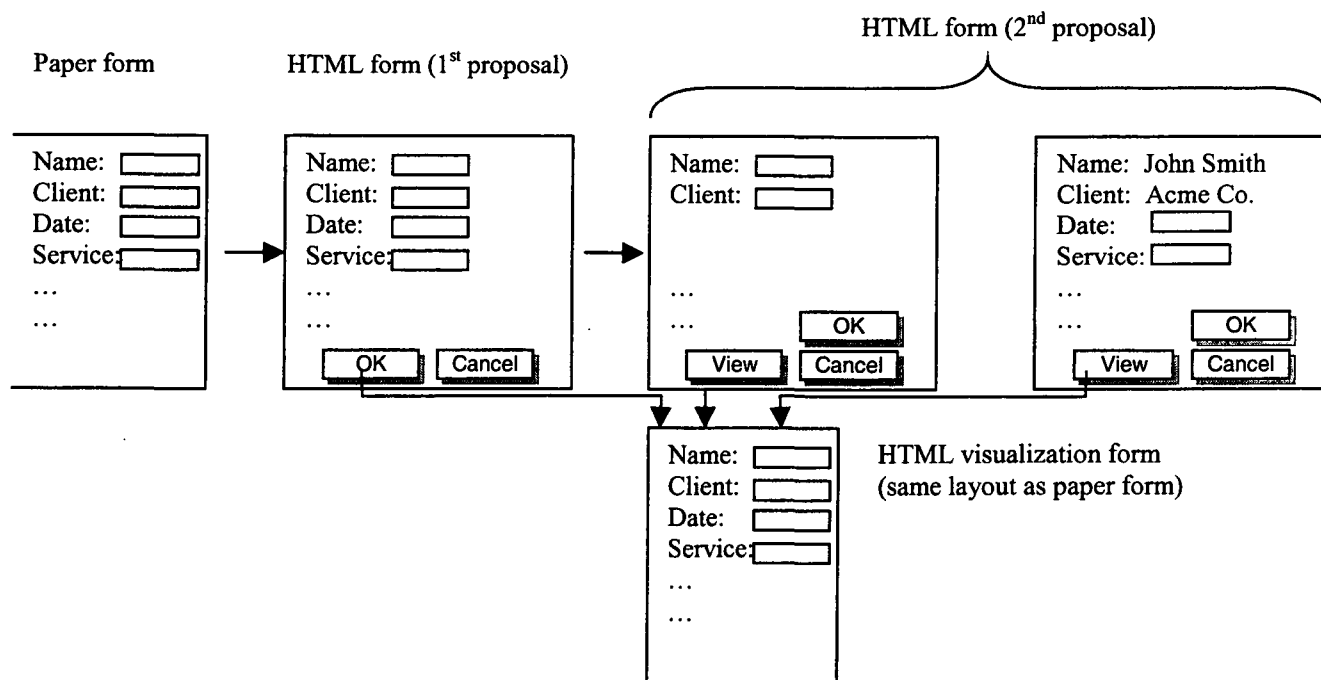


Figure 1 – An instance of evolution in technological scenarios.

Whereas this expedient had the undesired consequence that our mistaken interpretations about the roles of users in different tasks were not evident to our customers till later in the design process, it served to show them the difference between *automating* and *informing* people's work. Although they had verbally agreed with us in previous meetings about what this difference meant, it was not until they were presented the technological sign for it (the mock-up scenario the presenter run) that they realized they did not know about a number of positive aspects of WITS technology.

Other technological proposal scenarios allowed us to show some advantages brought about by the technology, such as the derivation of various indicators that would otherwise be too laborious to be generated. These indicators would also be new signs introduced in the domain, through the use of this technology.

When presenting technological proposal scenarios, users should be made aware that the signs depicted in the mock-ups are temporary representations. Users should be able to have a "feeling" of what the technology is going to be like, although they must be warned that changes are going to happen in the future. In other words, we are talking about sign-types, although we are using a sign-token as a placeholder. For instance, the OK button had its label replaced by the name of the next stage in the process.

User Studies Scenarios

These scenarios extensively described the work processes. At first, we gathered information from studying existing documentation, which was mainly composed of the application of ISO9001 regulations to this particular company and the corresponding paper forms already in use. The interesting point in these studies was the awareness that norms prescribe ideal processes, whose initial input is typically abstracted and omitted. The consequence for work practices is that people create their own habits for non-regulated processes, and this is precisely the reason why it is so difficult to introduce technology that will not only implement regulated processes, but also has to accommodate personal variations in carrying out jobs.

What matters to the norm are the protocols through which information is fed into the process. Technology designers have a choice to implement strictly the norm, or to support the surrounding processes that generate information that is protocolled to the norm. The risks involved in the first choice is the potential impedance between the quality of the regulated process and the quality of the information required for this process. This information may be produced under circumstances that cannot warrant an adequate degree of quality.

In *Qualitas*, we made the second choice. For instance, we observed that an occasionally lengthy preparation stage was being carried out informally by employees. The paper forms were used as a draft to gear the final documented version. Drafts were used as a basis for negotiating contracts with clients, and only when a final agreement was reached was the final form produced and filed. We chose to include this early stage in *Qualitas*, in order to provide better support for the negotiation processes with the client, and avoid some amount of rework involved in transposing information from drafts to the protocolled form. Figure 2 shows schematically the

difference between the flow of processes before and after the introduction of the technology.

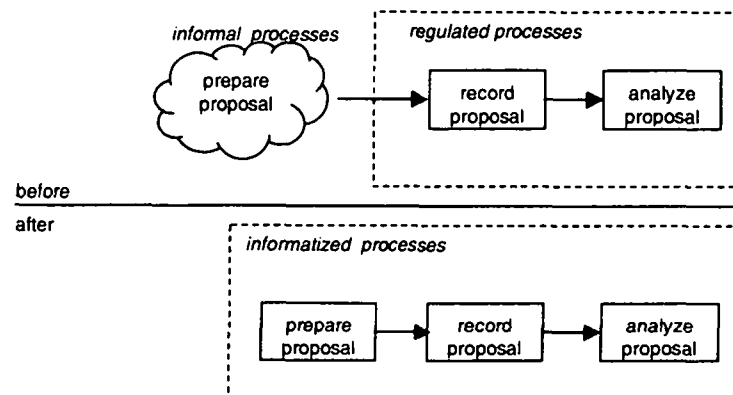


Figure 2 – Users' work processes before and after the introduction of the technology.

Another problem with the norms was that they did not prescribe remedial actions for all kinds of exceptions that occurred in work situations. As a result, employees took relatively arbitrary decisions about how to proceed, and this again jeopardized the overall quality of the regulated process. Since a major requirement of any software development process is to handle exceptions in an adequate way, the very introduction of technology triggered a knowledge elicitation process in the company. In it, the whole community of users was prompted to describe known exceptions, the way they handled them, as well as to envision other plausible exceptions and remedial actions for them.

For instance, a portion of the norm stated that:

"... the conditions of the contract should be formally communicated to the customer within five business days after the initial request of service ..."

However, it didn't explain how to proceed if, for some reason, the customer was not notified within the deadline. It was implicit that, in this case, a non-conformity report should be generated and the remedial process undertaken. But the very fact that the text didn't explicitly say so allowed for different interpretations of what a non-conformity *really* was. In order to explore the extent of latent practices, one of our scenarios explicitly introduced a technological contingency that would make a whole set of norms become completely inapplicable. We took the norms that referred to the procedures involved in preparing materials for a conference and used it to show them that the text was only applicable to very traditional conference settings. There were references to the quality of "overhead projectors", "slides", "xerox copies", and the like. We simply showed that in case of videoconferences, for instance, virtually 80% of the checklist imposed by the norm would be useless. What would they do then?

Reaction to these scenarios helped designers sense the latency of interpretants, which plays a major role whenever exceptions contexts arise.

Progressive Implementation Scenarios

Throughout the design and development process, many iterations of technologically-grounded scenarios were discussed with customers and users. They invariably led the participating teams to revise their beliefs and adjust interpretants about the company's goal, the employees' work practices, and the technological requirements. However, the most interesting aspect of these scenarios was that they helped us identify the emergence of new types of signs in the technological language. In the following, we will discuss two instances of this situation.

The generation of a contract's history

Because WITS must have internal representations for both information and tasks, it can generate logs of users' activities within the scope of the system. As a consequence, it was easy to generate a brief **summary** of how users carried out their jobs providing services involved in contracts they signed with their customers.

An example of a summary can be seen in Figure 3. Although the text is in Portuguese and the narrative is totally fictional, it shows the reader that even a bar chart is included in the document, depicting quantitative indicators relative to the service provided by the user.

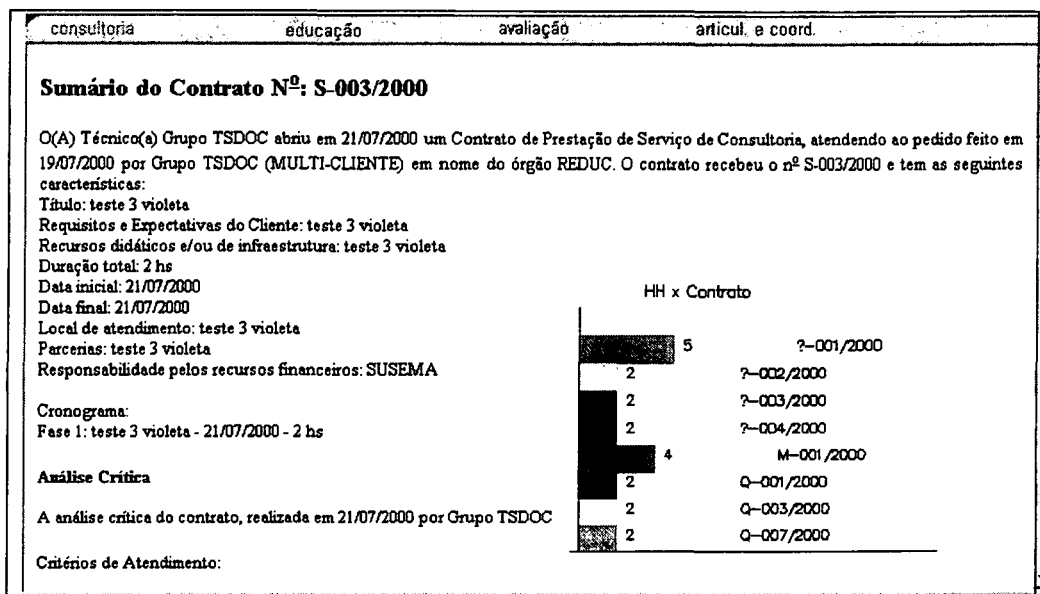


Figure 3 – An automatically generated summary of work processes.

In this scenario we see that the availability of internal system representations used by developers to check for consistency and keep track of turn-taking in complex collaborative goals, in a way motivated the interpretant of a process history. Designers saw internal representations as a sign of system behavior throughout the various WITS activities. Thus, they easily concluded that this could also be of interest to the users, themselves. The emerging sign was readily accepted and adopted by users, who recognized the contribution of technology at this point.

Performing Technological Signs

As soon as *Qualitas* was introduced in the company, even in its early Alpha and Beta versions, it became clear to customers and users that it was a performing version of the norms and procedures certified by ISO9001. Prior to *Qualitas*, the only way for an employee whose role was to provide a consulting service to understand what it was like to be his coordinator was to read about the coordinator's role (in the documentation) or to talk to the coordinator about it and to wonder. However, after *Qualitas* was running, a demonstration version of it (which later was used for educational purposes) allowed users to *play the role* of whoever they wished, and to gain a much wider understanding of what it meant to be someone else as far as work practices in that company were concerned.

This is another example of the way the designers of *Qualitas* used internal representations to communicate to users some interesting information they, designers, could "read" in the system. They had a global view of how roles were interconnected and how the overall process in the company was carried out by actors playing these roles. Thus, they saw in this an opportunity to explicitly represent this emerging sign at the interface, for users to explore in a demonstration version. This may be considered a technological sign at a meta-level if compared to the summary sign, because it assumes that users will switch between application modes and contexts.

Test case scenarios

Test case scenarios have been, along with training and technological evolution scenarios, the most revealing ones in the context of WITS. Because *Qualitas* was a kind of computer-supported collaborative system, test cases had to cover coordination and communication aspects of as large a portion of its operations as possible. The scenarios we generated for this stage were cast as a sort of role playing game (RPG). A group of approximately ten users was selected and assigned roles and tasks they should be engaged in acting out for nearly two weeks. The period of time was decided on the basis of plausible deadlines dictated by the enforced norms and procedures of the company.

As these were the last batch of scenarios prior to the system's full installation in the company, they made use of all the range of signs emerging in the company's context — not only in the system's context. They referred to contracts, clients, divisions and managers who were known facts, people, places, occasions, and settings for that community. Although the tasks people were performing in their RPG-like activity had no further consequence

beyond the scope of a simulation, the realistic grounding of the game contributed to engage the users. The value of these scenarios, in addition to the familiar and expected indication of occasional *bugs* and opportunities for improvement in the software itself, was to serve as a testimony of the company's maturity acquired by virtue of new technological settings. A comparison between test case scenarios and the early versions of the certified norms and procedures, the technological proposal scenarios, and the user studies scenarios, showed how much knowledge had been gained in two years of intensive research and development work. Both, our customers and us, had learned much about what we do and what the others do.

Training Scenarios

Like test case scenarios, training scenarios have given the design team and the customers alike a unique opportunity to reflect upon work practices. The role played by signs in this context has been paramount. The most interesting of all instances is reported below.

The Flow of Processes

In the first training session for managers who were going to use *Qualitas* from then on, the instructor presented them a schematic overview of the whole system, where simple graphic conventions were meant to show the audience the various processes covered by the system, the different professionals involved at each stage, and the dependencies between action taken at one point and the others. The graphic was prepared by the head of the design team, who was trained not only in HCI and Computer Science, but also in Semiotics and Rhetoric. This led her to depict what she thought were the important **abstractions and invariants** of work practices across different areas of the company, the connections between these and the interactions afforded through the system's interface, and the projection of roles people played onto processes they were involved with in the overall organizational dynamics. Her view was determined not only by the whole history of the design process she coordinated, but also by her aim to teach the audience how she thought the system reflected the essence of their work practices and introduced new ones that she thought would be beneficial for the organization.

A picture of the flow of processes she used for didactic reasons is seen in *Figure 4*:

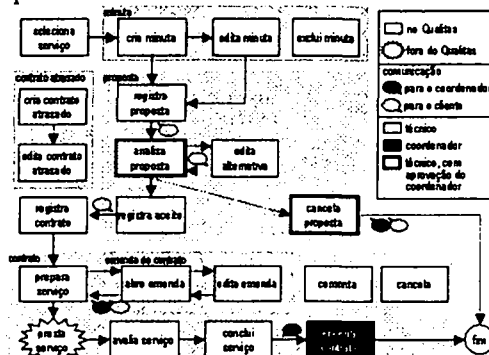


Figure 4 – The flow of processes carried out inside and outside Qualitas

The surprising effect of the diagrammatic sign appearing in one of the instructor's slide was that it caused an immediate reaction from the audience. The trainees related to the flow of processes with an intuitive understanding that helped them have an instant gestaltic perception of the technology. This was evidenced by the follow up questions they asked and the scenarios they built themselves to query the functioning of the system. They immediately asked for a copy of the diagram.

DISCUSSION

It should be concluded from previous sections that scenarios have played a central role as *representamina* that were used to induce a variety of interpretations throughout the development of *Qualitas*. In this discussion we will call the reader's attention to the semiotic qualities we have explored in scenarios (for a comparison between the scenario-based software development process of *Qualitas* and that of similar applications, see Monteiro, Barbosa and de Souza (2000)).

Information technology has been often mistaken by both its producers and consumers in terms of the effects it can bring about in organizational environments. Whereas it is undeniably true that there are many benefits in

providing a systematic treatment of information, that can be thus integrated into larger systems and processes, some of the consequences of computer-mediated information and work flows have not always been adequately analyzed [Suchman (1995), Dourish et al. (1996)].

Our choice of technological proposal scenarios in the form of a mock-up web-based *Qualitas* environment was a (performing) sign of how certain tasks would be changed by the introduction of WITS technology. As we said, our focus was not on interface widgets and screen layouts, which were intentionally left very sparse, but rather on the types of conversations that we expected to occur in the system. These conversations involved new languages and new signs which we specifically brought forth in order to trigger our customers' interpretants in unpredictable semiotic chains. The scenarios were *told* in these languages.

Since in our approach an achieved system is a one-shot message from designers to users, the ultimate purpose of our technological proposal scenarios was to foster an ongoing focused dialogue about what this one-shot message should eventually be. The approach helped us negotiate meanings in constantly evolving sign systems. Misconceptions that went unnoticed in one sign system, often emerged and were remedied in a more elaborate one.

The major contribution of semiotic theory dealing with technological proposal scenarios was to give the designers awareness of an emerging unique technological language (or sign system). The semantics of signs in this language is algorithmically embedded in the system. The words, phrases and conversational structures that will occur during interaction must engage users in semiotic processes that will pragmatically settle around interpretants that are consistent with programmed meanings and with culturally grounded work practices. Another important contribution of semiotic theory is a **conceptual framework** or, depending on the designers' expertise in the field, an **ontology** of communicative phenomena that can be put to use in order to produce a fair characterization of the designer's *deputy*.

The technological proposal scenarios function as a reification of a mutually agreed interpretant of what the technology should be. Once the meaning of the proposal is settled, a thorough analysis of future users in their current situation must be carried out as part of requirements engineering and user modeling processes. User studies scenarios in *Qualitas* were intentionally generated by the design team (and not by users), after numerous interviews, observation sessions, documentation analysis, and negotiation with customers. Our aim was to produce another reification of a mutually agreed interpretant, but this time it should be thoroughly focused on the users' work practices, independently of considerations about technological change.

User studies scenarios were told in a language that was totally familiar to users, unlike the case of previous scenarios, where the language was an innovative technological sign system. Cases were described in natural language narratives, where fictional characters performed non-fictional tasks in the company. The tasks included situations that had actually occurred in the analyzed history of contracts, as well as some plausible variations on those. Paper forms and standardized procedures were examined in these the scenarios, with a special emphasis on arbitrary interpretations the design team was aware it had generated due to incomplete information provided in the resources they had used. Another emphasized topic in user studies scenarios, as seen in the previous section, was that of exceptional situations, which helped designers explore latent interpretants in the community of users.

The contributions of semiotic theory for elaborating and interpreting user studies scenarios were twofold. On the one hand, the theory highlighted the abundance of arbitrary abductive processes followed by the employees (our future users), which not always settled around a consistent set of interpretants. This was a warning for our team, in that we should be able to let our users know as exactly as possible which of all registered interpretants the implemented system would be prepared to cope with. And although all interpretants were socially efficient prior to the introduction of *Qualitas*, allowing the company to operate pragmatically in line with acceptable standards, the technological protocols embedded in *Qualitas* could bring about major changes in that scene.

On the other hand, our team was also faced with the situation that normative texts are a genre where argumentation is an inadequate rhetoric to use. Therefore, the exploration about the reasons for norms, which would immensely help employees decide what to do in case of exceptions, was typically left out of documents generated from or about work practices. Notwithstanding this aspect of norms, information systems are exactly the kind of technology devoted to helping users decide what to do when they find themselves in a situation they are unsure of their next step. Therefore, it would not be appropriate to merely transpose the signs that occurred in the organizational discourse genre prior to the introduction of *Qualitas* into the new system. A thorough interpretation and evaluation of signs in the new genre was called for, so that the users could the benefits of WITS beyond and instead of fearing that their job would be automated and they would thus lose their status in the company.

Moving on to the progressive implementation scenarios, the encouraging contribution of semiotic theory in cases such as the one illustrated in the previous section was the basis it provided for multi-coded information presentation. The information content of the message conveyed by text and graphics presented in Figure 3 was the same as that conveyed by a set of filled-out forms (identical to the paper forms used prior to the introduction of *Qualitas*) offered by the system in one of many alternative visualizations of task-related information. Once the

system's algorithmic interpretant of tasks is adequately characterized, it becomes the semantic grounding of a number of alternative *representamina* — visual, textual and even procedural. A remarkable consequence of the emergence of the service summary sign in one of the progressive implementation scenarios was that the company revised its own paper forms in order to include explicitly a field where the employees were required to write a brief narrative of their contract's history.

The second instance of progressive implementation scenarios, in the previous section, leads us to yet another valuable semiotic contribution to software design practices. Systems are performing messages that can be told in many different codes and many different genres. The unique nature of computer systems is that they allow users to experience *virtualities* [Winograd (1996)] and to gain perspectives that cannot, by any other means, be systematically communicated. An exploration of the relations among co-referential signs across codes and genres shows the designers a wealth of possibilities for intentional *intertextuality* [Kristeva (1980)] that can eventually invest technology with meanings that go untapped. This feature is clearly related to experimenting the *alter semiosis*, where roles played by different people in a community can be better reified in signs that can be explored from an action perspective.

The history of changing scenarios, documentation, practices and systems is equivalent to a semiotic process. Along the historical chain in *Qualitas* scenarios, we could see the reifications of interpretants resulting from a continuing abductive process that had led us to the final system implementation. Quite importantly, semiotic theory warned us against the stability of semiosis and the definitiveness of meanings for human beings.

This caused a major impact on our development team's conception and on the architecture of WITS we eventually committed ourselves to using. We realized that the success of *Qualitas* or any similar system can only be achieved through mechanisms that can allow users (or developers in narrow collaboration with users) to customize, adapt and extend WITS [Cunha et al. (2000)]. It also called out attention to the importance of introspection in software sub-systems, which Dourish has referred as the system's *accountability* [Dourish (1997)]. If software can *account for its behavior*, for instance by generating narratives of its inner workings at an adequate level of abstraction for users to understand, it can productively reveal its algorithmic interpretant. Thus, even if users cannot make the system understand their innovative interpretants and use it appropriately, they can always examine the system's unchanging interpretations and try to compensate this inflexibility temporarily with novel ways to use information and do their work.

As far as training scenarios are concerned, Semiotics pointed to the importance of alternating codes in communication in order to convey complementary views on the same phenomenon. As shown in the examples provided earlier, the team saw that the flow diagram used primarily as pedagogical support should be added to the system's interface as a sign through which users would not only be able to have a rapid access to the various modules of the system, but also and more importantly be able to assess the temporal and causal relations between different stages in the processes covered by that technology. Additionally, the diagram could easily be turned into a dynamic visual sign, that would be able to indicate how far or close the users are from task completion, whom their job affects and by whom it is affected, where the process bottlenecks are, and other contextual information. All of the latter are important resources that support opportunistic problem-solving practices that are most encouraged to ensure an organization's competitiveness in the market today.

The noticeable point about this is that the emergence of the sign, in static form, and the opportunity to alter its nature to take advantage of underlying algorithmic interpretants by turning it into a dynamic sign, came directly from the ontological underpinnings of design choices made by design team participants.

Yet another kind of scenarios could be identified in the development process, although the haven't been fully explored to-date, the technological evolution scenarios. However, there are clear indications of what those would be. As was mentioned previously, one of the services provided by our customers is that of coordination. This amounts to *organizing meetings and larger events, and acting as a leader in contingent group activities of various sorts*. Although there are some general norms and procedures relative to these tasks, they are in fact the least understood processes of all, because so much of it is dependent on unique situations that require different skills and styles from coordinators.

Our customers decided to postpone the specification of this part of *Qualitas* to a future date when they hope there will be more established knowledge about work practices and information use. However, our design team decided to take a more proactive attitude and to propose technological evolution that basically embraces WITS and sees coordination services as something will not be *decided, planned and reported* in *Qualitas*, but will be actually *carried out* in the system. In other words, whereas educational activities, for instance, are analyzed and planned in *Qualitas*, carried out outside the system (in some external physical setting), and then *reported and evaluated* back in the system again, coordination activities would be quite extensively achieved in *Qualitas* through an elaborate set of communication and coordination tools.

This proposal derives from all the work done in the field of *groupware* and *collaboration environments*, but there is an interesting semiotic addendum to it. In all other modules of *Qualitas* an interested examiner (an auditor or a new employee, for example) can take *Qualitas* as an active document [Stefik (1995, pp.143–145)] that *enacts* (or performs) the *discourse* that is told about working practices. The structure of such discourse is partly determined

by certified norms and procedures, partly by technological protocols; and partly by social protocols present in this community. The difficulty to produce discourse about coordination services is that it is fairly unstructured and deeply dependent on every different situation the employee encounters while doing his job.

Existing norms state that meetings, for instance, must be called by the coordinator, who is also responsible to propose the best date and venue for it, to prepare the meeting's agenda, and to make sure that it is achieved by the participants on the occasion the meeting takes place. Whereas this is all true, the norm doesn't say much that somebody doesn't already know about coordinating meetings and, more crucially, it doesn't capture the coordination patterns adopted by the employees. These patterns are important indications of how efficiently the process is being carried out.

The technological evolution scenario we are currently working with is one of adding tools that will, by necessity, implement algorithmic interpretants of the processes involved in such activities. By allowing users to obtain an account of these interpretants and to customize a considerable portion of the offered tools, we intend to generate a new technological sign — partial immersion. By this we mean the ability that users will have to progressively eliminate the distinctions between discourse and action. In particular, as teleconferencing facilities are added to WITS architectures, once work is done in *Qualitas* a series of sign variations about work (history, cases, patterns, tendencies, and many other *indices*), both static and dynamic, can be automatically generated. If group decisions can affect the way work is carried out, so that the most efficient practices are always known and considered for a first choice, the whole purpose of norms is superseded by more agile information conveyed by a combination of social and technological organizational discourse.

CONCLUSION

In this paper, we reported and discussed our experience in implementing *Qualitas*, a web-based workflow system. Semiotic theory was extensively used to support design decisions and negotiations with users about technological signs. Taking scenarios as a type-sign exchanged throughout the whole process, we could trace the theoretic underpinnings of our experience and draw some revealing conclusions about the product and the process of technologically reified discourse. Although it is present in all information technology applications, this kind of discourse is seldom analyzed by software designers and developers. Our conjecture is that outside semiotic theory, professionals involved with human-computer interaction and software engineering practices have difficulty to coalesce concepts derived from such different disciplines as psychology, anthropology, linguistics and sociology, just to name a few. Semiotics, however, can by itself provide a unifying ontological basis for interdisciplinary knowledge, raising issues and proposing alternatives, that may help professionals gain insights at lower learning costs.

We have seen how such semiotic concepts as *interpretant*, *semiosis* and *abduction* were repeatedly invoked to provide the knowledge scaffolding needed to build explanations for observed phenomena and to generate solution to identified problems. The fact that systems can only produce algorithmic interpretations of signs helped us appreciate the extent to which this fixed semiotic process is equivalent to a reification of interpretants. The systematic repetition of interpretations by the system creates an imprint of itself that in time gains a concrete character, that is very different from natural human interpretations (that are seldom the same, even under nearly identical circumstances). This ontological contrast must be faced and dealt with by designers, not reduced to the idealization that if we study users well and long enough we will eventually be able to embed the *correct* processes and meanings in the technology we are developing.

Once the reifications are designed and implemented, a productive process of sign variations on a theme can be carried out to explore the inherent communicative qualities of different codes such as the textual, visual and procedural. We saw that in *Qualitas*, demonstrational walkthroughs, natural language narratives and descriptions, filled-out forms, as well as histograms, pie charts and X-Y graphs, all amounted to different renditions of a fixed set of meanings programmed in the system. This undeniably helped not only *Qualitas* users, but also the development team and our clients, gain a wide perspective on what we were all doing and/or saying we were doing.

In sum, the most important insights supported by the semiotic approach whose echoes we have tried to convey in this paper have to do with collaborative sense-making. This was the key underlying process taking place between us and our clients, between our clients and the employees who were going to be affected by their decision to introduce new technology in the company, between us and these employees, and finally between them and our system. The success of this multi-faceted collaborative sense-making process can be assessed by evaluation experiments of various sorts. The results of evaluations can be contrasted to every semiotically-motivated design choice and tell much about how well the application is communicating shared knowledge by means of the signs present in its interface.

ACKNOWLEDGMENTS

The authors would like to thank CNPq and Tecgraf for the financial support that allowed this work to be developed. They are very grateful to all participants of the development team who have always engaged themselves in discussing relevant research issues associated to this project and contributed with valuable insights.

REFERENCES

- Adler, P.S. & Winograd, T.A. (1992) **Usability: Turning technologies into tools**. Oxford University Press. New York, NY.
- Brown, G. & Yule, G. (1983) **Discourse analysis**. Cambridge University Press. Cambridge, UK.
- Carroll, J.M. (2000) **Making Use: Scenario-Based Design of Human-Computer Interactions**. The MIT Press. Cambridge, MA.
- Cunha, C.K., de Souza, C.S., Quental, V. & Schwabe, D. (2000) "A Model for Extensible Web-Based Information-Intensive Task-Oriented Systems". In **Proceedings of HCI'2000**. Sunderland, UK.
- de Souza, C.S. (1993) "The Semiotic Engineering of User Interface Languages". **International Journal of Man-Machine Studies**. No. 39. pp. 753-773.
- de Souza, C.S., Barbosa, S.D.J. & da Silva, S.R.P. (to appear) "Semiotic Engineering Principles for Evaluating End-User Environments". To appear in **Interacting with Computers**.
- de Souza, C.S., Prates, R.O. & Carey, T. (2000) "Missing and declining affordances: Are these appropriate concepts?". In **Proceedings of the III Workshop on Human Factors in Computer Systems, IHC'2000**. Gramado, RS, Brazil.
- Dourish, P., Holmes, J., MacLean, A., Marquardsen, P., Zbyslaw, A. "Freeflow: Mediating Between Representation and Action in Workflow Systems". In **Proceedings of Computer Supported Cooperative Work '96**. Cambridge MA USA. pp.190-198.
- Dourish, P. (1997) "Accounting for System Behavior: Representation, Reflection, and Resourceful Action". In Kyng, M. & Mathiassen, L. (eds.) **Computers and Design in Context**. The MIT Press. Cambridge, MA. pp.145-170.
- Eco, U. (1979) **A theory of semiotics**. Indiana University Press. Bloomington, IN.
- Frege, G. (1892) "Über Sinn und Bedeutung" **Zeitschrift für Philosophie und philosophische Kritik** 100, 25-50.
- Glance, N. S. and Pagani, D. S. and Pareschi, R. (1996) "Generalized Process Structure Grammars (GPSG) for Flexible Representations of Work". In **Proceedings of Computer Supported Cooperative Work'96 (CSCW'96)**; pp. 180-189.
- Heinl, P., Horn, S., Jablonski, S., Neeb, J., Stein, K. and Teschke, M. (1999) "A Comprehensive Approach to Flexibility in Workflow Management Systems" In **Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration - WACC'99**. pp. 79-88.
- Imaz, M. & Benyon, D. (1999) "How Stories Capture Interactions". In **Proceedings of Human-Computer Interaction - INTERACT'99**. IFIP.
- Josephson, J.R. and Josephson, S.G. (eds., 1994) **Abductive Inference: Computation, Philosophy, Technology**. Cambridge University Press, Cambridge.
- Khoshafian, S. & Buckiewicz, M. (1995) **Introduction to Groupware, Workflow, and Workgroup Computing**. John Wiley & Sons. New York, NY.
- Kristeva, J. (1980) "Word, Dialogue, and Novel" **Desire and Language**. Rudiez, L. (ed). New York. Columbia University Press. Pp.64-91.
- Malone, T. W. and Lai, K. and Fry, C. (1995) "Experiments with Oval: A Radically Tailorable Tool for Cooperative Work". In **ACM Transactions on Information Systems**; Vol. 13; No. 2; April 1995; pp. 177-205
- Monteiro, C. de C., Barbosa, S.D.J. & de Souza, C.S. (2000) "The Role of Designer-Generated Scenarios in Developing Web Applications: A Case Study". In **Proceedings of the III Workshop on Human Factors in Computer Systems, IHC'2000**. Gramado, RS, Brazil.
- Nake, F. & Gabrowski, S. (to appear) "Interaction is pseudo-communication".
- Newell, A. (1982) "The knowledge level". **Artificial Intelligence**, Vol. 18, 1. pp.87-127.
- Peirce, C.S. (1931-5) **Philosophical Writings of Peirce**. Selected and edited by Buchler, J. Dover Publications, 1955. New York, NY.
- Prates, R.O., de Souza, C.S. & Barbosa, S.D.J. (2000) "A Method for Evaluating the Communicability of User Interfaces" **ACM interactions**. Jan-Feb 2000. pp.31-38.

Stefik, M. (1995) **Introduction to Knowledge Systems**. Morgan Kaufmann. San Francisco, CA.

Suchman, L. (1995) "Representations of Work". Guest Editor in **Communications of the ACM**. Vol. 38, No. 9. pp. 33–35.

Winograd, T. (1996) **Bringing Design to Software**. ACM Press. New York, NY.