Bringing Interface Design and Software Development Processes Together: How Organizational Issues Impact Product's Usability

M. Cecília C. Baranauskas[1] **Juliana Salles**[2] **and Roberto S. Bigonha**[2]

[1]Institute of Computing- IC
State University of Campinas – Unicamp
Cidade Universitária Zeferino Vaz
13083-970 Campinas – SP – Brazil

[2]Computer Science Department-ICEX
Federal University of Minas Gerais
Av. Antônio Carlos 6627
31270-010 Belo Horizonte - MG – Brazil

cecilia@ic.unicamp.br ,{jusalles, bigonha}@dcc.ufmg.br

Abstract. *While usability issues have been widely stressed in literature as a goal of the software design methodologies, quality in the design and development process has not received the same attention. This work aims to investigate software production processes by addressing the communication among work groups in a software design organization. Our focus is on understanding the communication process that takes place between groups in a software design organisation, the nature of the changed messages in relation to their meaning for the audience, and how they impact and are propagated in the whole process. We based our understanding of communication in Semiotic foundations, to propose a meta-model for analysing the whole process of system design and development. A case study with the design process of a large software organisation illustrates our proposal.*

1. Introduction

Organizational issues involved in software development and use have long been the focus of the Information Systems (IS) field. Nevertheless, some approaches in the Human-Computer Interaction (HCI) field have also concerns about organizational aspects influencing the design and use of computational systems. HCI and information systems developments have evolved from different backgrounds and focus to a common end: to achieve high quality in software usage [7]. While quality in the product has been widely stressed in literature as a goal of the software design methodologies, quality in the design and development process has not received the same attention. This work aims to investigate software production processes by addressing the communication among work groups involved in a software production organization.

A group can be defined as a collection of persons arranged or classified together because of some specificity, common interest, shared ideals, etc. A work group adds to this definition a shared goal and articulated actions to achieve it. This means that the interaction and relationship among the group members is an important differential. In the same way, work organisations can be thought of as "*groups of groups*" [10], as the dynamic processes defining relationships among the groups are important factors in the promotion of the established goals.

Even though the organisation as a whole may have a single clear set of goals and interests, individuals and groups within the organisation may not share these goals and interests in the same way. For instance, in a software development organisation, members of the Usability Engineering team may differ substantially from members of the Development team in terms of their background and work practice. To most software engineers, the system is thought of in terms of its structure, functions, and components. To a usability engineer, the system includes the human user and all his potential actions interacting with the software. In addition, there are different performance parameters for different groups inside the organisation. While the software developers have their concern on development schedules and correction of code, the usability engineers have the preoccupation of insuring usability without causing delay in the process. A kind of tension between groups is not rare. In addition, within organisations, work groups can be influenced by the different cultures revealed through their assumptions, beliefs, self-image, feelings and fears, and the language they use to talk about their daily work [3], [10].

A comprehensive and integrated understanding of the connections among design and development activities is necessary at all levels of the process. Understanding and integrating the whole development process requires an understanding of how different groups in the organisation acquire and communicate this overview. As pointed out by Hix and Hartson [11] this seems to be true in any large system (e.g. a new automobile design). Participatory [15] and Contextual Design [3] propose several techniques as communication mechanisms in order to bring a design team to a shared understanding of the customer.

While we recognise the efforts of some methodologies for a shared understanding of the subject of design, we argue that a global view of the communication among the different groups in the organisation is important to ensure quality in the process and in the product, as a consequence. As a responsibility of the whole project, not of individual initiatives, new communication requirements emerge in the design and development process.

Semiotics allows us to understand information and communication aspects involved in organisations. Liu [13 p.7] points to "*a set of methods that can be used by researchers and business users in their understanding, development, management and use of information systems*". Our focus in this work is on understanding the communication process that takes place between groups in a software design organisation, the nature of the changed messages in relation to their meaning for the audience, and how they impact and are propagated in the whole process. We based our understanding of communication on Semiotic foundations, to propose a meta-model for analysing the whole process of system design and development. A case study with the process of a large software organisation illustrated our proposal.

The paper is organised as follows: Section 2 presents the theoretical rationale for our understanding of communication in the design process. Section 3 presents a case study to highlight some results of representing communication involved in a product development cycle and walking-through the proposed meta-model. Section 4 concludes pointing out to further work.

2. Understanding Communication in the Software Design Process

In a software development organisation, all the work groups need to know what is going on, and the impact that the interaction and communication processes taking place among all these people has in the product.

> "There are the rest of the engineers on the project who have to believe in the system enough to code it. There are the project teams working on systems that have to interface to yours. There is your manager, his manager (…) There is the marketing and the product-planning department who tend to be sceptical of ideas coming out of engineering. There is the sales force of a commercial product, which needs to understand what makes the new product worth selling. And there are the customers, who need to be convinced that the new system will improve their lives" [3, p. 199].

Beyer and Holtzblatt [3] argue that it is not feasible to include all the people who care about the result in the same design team. On the other hand, they maintain that a cross-functional team, despite being important to bring the perspectives of the different groups to design, does not guarantee communication back to the groups. The authors propose multiple strategies and techniques to enable communication among the groups, through the Contextual Design.

Contextual Design presupposes that any system embodies a way of working. As so, it is a method that define a series of actions that lead a team to agreement on what their customers need and how to design a system for them. The main contribution of this approach is a series of techniques that forces people to interact with data, to share a space for analysing customer data, to engage with work models, and mainly to tailor conversation to the concerns and work style of each involved group.

Semiotic approaches perceive the software interface as a communication act between designers and users, using the computer as medium [16], [6], [2]. The designers establish the limits of this communication and create a set of signs that users can activate [2], which means that designers are the senders of this communication model. Semiotic Engineering [6] considers the message itself a meta-communication artefact, since the interface exchanges messages with users.

The focus of this work is the communication involved in the design and development of computational artefacts. One aspect of this communication is the dialogue between designers and users. As Adler and Winograd point out [1], this kind of communication is embedded in every kind of artefact. Through their structure and appearance, designed objects express more or less effectively what they are, how they are used, and how they are integrated with the embedded context. According to semiotic approaches in HCI, this structure and appearance - i.e., the interface- can be understood as a message sent from designers to users using the computer as channel.

We argue that in order to design this message, all the involved groups need to develop a common understanding about what is going to be done. On the one hand, there are the users' needs and expectations; on the other hand, there are contextual, technological, budget and schedule constraints that limit the possible design solutions. Not only designers and users, but also the remaining agents must engage in a process of negotiation, which we understand as a conversational process. In this conversation, many communication acts occur, many messages are exchanged using different channels.

In this work, we argue that to understand the dimensions involved in the construction of the interface as message, it is important to develop a better understanding about the dialogue that occurs among the many parties involved in the design activity.

3. Design as a Matter of Communication

Design has been understood as a technical term referring to a software production step in which the lines of code are put together according to some software development methodology. In this work, as used by some influential authors in the area of HCI [27],[19],[24], the English language usage is adopted for the term, meaning the invention and organization of any structure.

In order to design the message (i.e. the interface), designers and users must engage in a dialogue using different channels. As Gould and Lewis point out [8], "*getting useful design information from prospective users is not just a matter of asking*". Designers need to ask questions, represent the information they get and the solutions they provide in a meaningful way to check it, perceive users' questions and reactions and so on [18], [20]. The designer-user dialogue has been carried out through usability tests in the successive phases of a design-implement-evaluate loop that characterizes the design process, which is, by its very nature, iterative.

More recently, the importance of the work context in the design process was widely recognised. Besides the evaluation of the users' performances when they interact in isolation with an artefact, another goal came onto the scene: designers realised that it was also important to analyse how an interface can reflect users' work context and support users' social interactions to execute their tasks. New approaches, like participatory [15] and contextual design [3], were developed and introduced additional channels in user-designer communication.

Besides designer-user communication, the other groups are also engaged in some type of communication through different channels: designers talk to marketing people, customer support mediates between developers and users, external consultants help both users' and developers' organizations, etc. The use of different channels is related to the needs of the communication acts: the nature of information desired, the projects' contexts, the audience's language, etc.

Communication Models and the computer as media

To understand communication in the design process, we must first form a coherent understanding of what communication itself involves. Several models for communication have been presented and discussed by thinkers from diverse philosophical schools. We will examine some of them in order to reach the essential aspects of the concept.

For a long time, the dominant model in the theory of communication was Shannon and Weaver's Mathematical Theory of Communication [25]. Grounded in the Theory of Information as a system of mathematical basis to study the problems of transmitting messages through physical channels, it is therefore a technique of communication engineering in which they identify source, encoder, channel, noise, decoder and receiver of information, as illustrated by Figure 1.

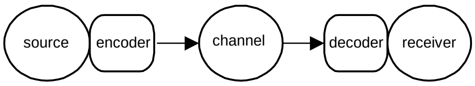

Figure 1. Shannon and Weaver's communication model

To understand the linguistic processes of verbal communication, Jakobson [12] proposed a model in which he identifies 6 factors that constitute every verbal communication act. The addresser sends a message to the addressee. To be effective, the message requires a context to which it refers, apprehensible by the addressee, a code total or partially common to both (addresser and addressee), and a contact, a physical channel between the addresser and the addressee. As in Jakobson's proposal, the semiotically improved model adds the role of the repertory of signs of both the sender and the receiver and requires the overlapping of the two repertories as a necessary condition for communication [5].

Westley and MacLehan [26] proposed a model derived from mass communication that reflects the various interactions among the entities (sender, receiver and channel), involved in the communication, including the receiver-sender, receiver-channel and channel-sender directions. In this model, the receiver plays an active role. He or she asks for information and it is this question that determines the content and form of the message created by the sender. There are feedback points that seem to better represent the conversational process; through them, the senders can add content to a message or change its form to facilitate the receiver's comprehension.

The direct transposition of the models derived from the Information Theory to understand human communication has many drawbacks, however. Parameters such as entropy or redundancy, while pertinent to "messages" are not the most important and do not reflect the specific nature of human communication. Another difficulty pointed out in the literature [5] is the unidirectional movement associated to the communication act, from a source (or addresser) to a receiver (or addressee). Receivers and senders actually engage in dialogues involving a process of meaning negotiation towards a common understanding. What receivers have understood from what a sender has "said" is frequently revealed in what receivers say themselves when they next take a turn at the communication act [4]. Moreover, these models do not explore alternative channels through which the communication between senders and receivers also take place. In our view the diversity of channels used for communication is what makes it inherently complex and powerful. On the other hand, models derived from verbal communication do not bring into focus the channel particularities and the ways they can shape the message being conveyed.

In designing the interface (i.e., the message), designers need that users express their needs and expectations, describe their tasks, work process, and so on. They exchange ideas with designers, present documents describing internal process to designers, show the designers the codes they use to represent their data, and so on. In doing so, users are communicating with designers using channels other than the computer artifact alone, so the designers can understand what needs to be designed. On the other hand, designers must check what they understood against what users have "said"; they must represent it in a meaningful way so they can "talk back" to users about it. They are also using channels to exchange these other messages; frequently, they are even creating new codes to carry out these communication acts.

As pointed out by Beyer and Holtzblatt [3], all artifacts even the most informal, have structure and content (information that the artifact carries). In designing the artifact, the designer establishes a conversation with the design medium, in the sense explained by Schon [24]. This level of communication is also part of the process and thus should be considered.

A fractal communication model, proposed in [21], [22] captures the nature of the communication process involved in software design, while stresses the fact that, in order to design the primary message (the interface), other fractionated messages must carefully be designed and appropriate channels must be chosen to convey them. Figure 2 illustrates the main concepts of the fractal model of communication.

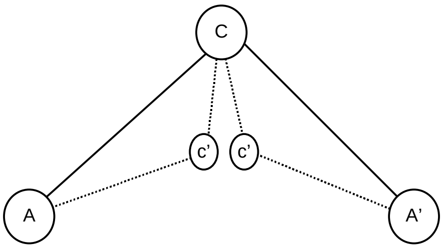

Figure 2: The fractal model of communication

In this diagram, nodes represent the communication agents (A and A') and channels (C and C'). The arrows are bi-directional, which means that an agent sends and receives messages. Nodes C' represent the fractal nature of communication. Different foci of the design process can be highlighted: the designer-user communication (A-A') using the interface as message, in a first level, as the interface is the unity- message conveyed by the computer (which is the first channel). The designer-artefact communication (A-C), and the user-computer communication (C-A') are represented in a second level of the fractal, having C' as special channels.

An inner level could be opened for analysis, showing a third level of the fractal structure, as illustrated by Figure 3. Usability engineers, for example, communicate with users using a usability test as channel. In designing the test, usability engineers communicate with the emergent artefact (the test) through a checklist as channel, for example.

In this model, the interface is understood as a unity-message that reflects what was grasped through the fractionated messages. So, the interface as a unity-message is directly affected by the choice of channels and messages used to compose the fractionated messages during the design process. This means that, in designing the interface, or the unity message, many fractionated messages are being exchanged. Each one of those messages is also being designed and should be carefully designed to make the designer-user communication through the unity-message smoother. As both senders and receivers in this communication process, designers and users change turns and cooperate on the development of the interface.

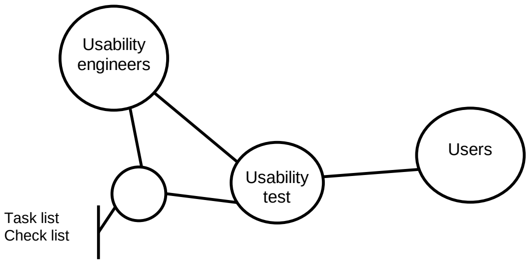

Figure 3. An instance of the fractal inner level

This model reflects the authors' understandings about design. This structure can better reflect the potential contributions of each part in the whole process. Also, in designing the interface, designers should also design other messages and choose appropriate channels to send them. The design of the fractionated messages is highly influenced by the development context [9] and the particular characteristics of each project. The conversational process (first level) and the designers and users communication with the emerging artefact (second level) are made explicit in the proposed model as we consider them fundamental aspects to improve the quality of the communication in the first level.

4. Meta-Modelling a Product Development Cycle: A Case Study

The meta-model was applied in a major software design company, given the pseudonym here "ORG", which employs leading professionals in all areas involved with the creation of commercial applications also called off-the-shelf software. The main groups identified as agents of the communication process were marketing people, program managers, designers, usability engineers, developers, testers, and support. Customers, clients, beta-users and users are four types of external agents called "users": clients are people who have already bought the product and receive the service of the support team for problems with the product. Customers represent the market: people who are identified by the marketing as a potential client. Users are the subjects recruited by the usability team to participate in the usability tests. Actually, these three categories of persons are potential end-users for the product.

Figure 3 illustrates the meta-model applied to the product development cycle of this particular organisation. An overview of the communication among the groups is showed in a simplified way, with two levels of the fractal structure.

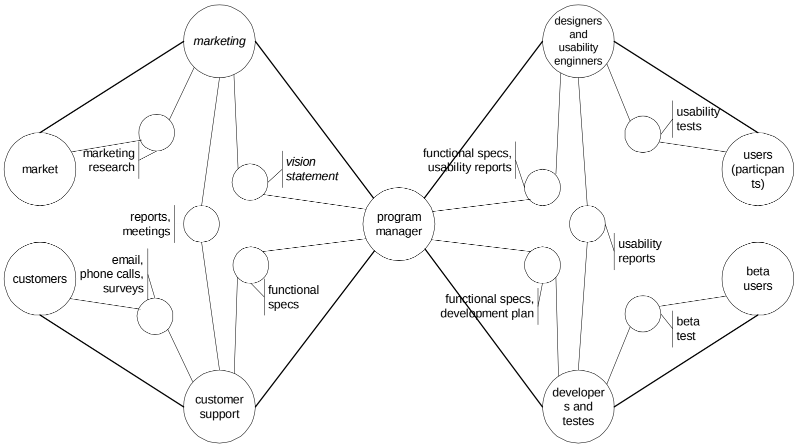

Figure 3. Meta-model for the product development cycle at ORG.

Marketers are responsible for ensuring a product meets a need for which people will pay money. Besides dealing with issues related to the market, they are engaged in product planning activities, through the product manager. They focus in demographics rather than on work practice, in identifying what kind of customer makes up the market. Marketers deal with feature lists, lists of customer needs, requirement lists, wish lists, etc. Customers are people who make the market and are targeted by the marketing group.

Program managers are the people in charge of the project, whose primary responsibility is to ensure the promised features are being completed on time and with acceptable quality.

Developers are programming engineers and have their focus on technology and on what make a clean program code. Traditionally they get directions from the marketing, through the program manager, who specifies what the overall system must do. They focus on clean implementation of the system design. Testers are people who work very close to the developers, testing code and reporting bugs.

Designers and usability engineers are the people who would have to be in charge of the product design, assuring usability for it. The usability group is a kind of quality control group, as they have the direct observation of the problems experienced by a user interacting with the system. Actually, at ORG designers and usability people are in charge of the product interface design and test.

Support are people in charge of responding to problems encountered by clients in the use of the system.

Typically the life cycle of the ORG products has three phases: planning, development and stabilisation phases. In the planning phase the functional specification and the project schedule is produced. In the development phase, internal liberation of a functionality subset is produced. In the phase of stabilisation, the product is extensively tested, in house and externally (beta tests).

During the planning phase, communication between the product manager (marketing) and the programming manager is established to produce the vision document. This artefact defines a set of goals that drive the product development. These agents also produce a high level specification for the product with a preliminary list of features to be present in the final product. The product life cycle at ORG is oriented by this vision document and by the high level specification.

During the development phase, communication between program managers and developers take place and the functional specification evolves as a result of this conversation. Developers have an important role in the choice of the features to be implemented. Subsets of this functionality are defined and milestones are stated for their liberation. Each set of features liberation involves intensive communication between developers and testers. While developers write code based on functional specification, the later test it for bugs. It is also in this phase that the interface is designed and tested for usability problems. After the last liberation, modifications in the main interface components (menus, dialogue boxes, etc.) are not allowed anymore.

During the stabilisation phase, the software product is extensively tested at ORG and outside, with Beta-version clients. Testers and developers establish communication with Beta-version clients during this phase. When high severity bugs are not found, the product is liberated to manufacturing.

The different agents also establish conversation with the emergent artefact, using channels related to their function in the organisation. As examples, there is a conversation between usability engineers and the emergent interface, using inspection methods as channels; developers establish a conversation with the code they are creating, through the tools they use (compilers, debugging tools, etc.).

Walking through the meta-model

An overview analysis of the meta-model for the design process point to aspects of communication among the agents, which deserve further discussion:

- The analysis on the meta-model tells us that the process is marketing-oriented, which is consistent with the economical and cultural contexts of ORG. This is reflected in the development driven by the functionality. The program manager acts as a channel between the marketing and all the other groups (developers, testers, designers and usability engineers). He propagates information to developers and testers using the functional specification, communicating the intended functions for the product and the priorities to implement them. From the developer perspective, this is an efficient message. However, designers and usability engineers need more elements to understand how the functions should be arranged and structured in the product as a whole, determining the user experience. On the other hand, designers and usability engineers' vision should be considered especially in the planning phase of the cycle. However, as the meta-model shows, their participation is noticed only in development activities (concentrated in the right part of the diagram), when the product specifications are already defined.
- The understanding about the user has different meanings to different agents of the process. The meta-model shows us different "users" communicating to different agents: some users participate in the usability tests, communicating to the usability

engineers; customers are the people the marketing is targeted to; clients are users in communication with the support group; beta-test participants are users communicating to testers. Nevertheless, the marketing group is the only agent involved in the identification of (**who** are) the prospective users. Other agents in the process demand understanding about the users: designers should know for whom they are designing. Marketers use specific channels (market research) to get a demographic description of the prospective users. However, the data interesting to the marketing group do not necessarily interests to the designers group. New channels and specific messages could be introduced between the agents in communication with those users and the designers group to convey a more complete view of the user and to accommodate specific necessities of designers and usability engineers.

- Support to the clients and beta-tests would be powerful additional channels to identify (**what** are) the usability problems, as they have direct access to problems pointed by users. However, in the first case, the data collected by the support group do not propagate to the usability engineers. In the beta-test case, the collected data are not adequate to answer for usability questions. Despite having the channel for communication, new messages should be designed in the beta-test to address also usability issues. Using data from beta-tests and from the support records would mean the utilisation of already existent opportunities. Data collected through these channels could be used as input in the design of usability tests to answer how the problems occur. Thus, the whole set of messages would be complemented more effectively.
- Certain channels, despite present, do not establish a psychological connection between two groups. In analysing a channel, it is important to adequate the messages to the audience. One group needs ways to communicate that are tailored to the concerns and work style of the other group. The program manager acts as a channel between the marketing and designers/usability engineers using the functional specification document. While developers think in functions as units of implementation (consistent with marketing objectives), designers need a much more wide view that is not communicated by the specification document. Also, the communication between usability engineers and developers through the usability report and email does not seem adequate to the developers who are much more technically focused. As a consequence this message does not reach the developer.

Results of a first analysis on the meta-model point to the necessity of new channels and messages to enhance communication among key elements of the process. The dynamic enabled by new communication channels and the message propagation could lead to a more integrated design and development process and potentially a better product.

5. Conclusion

In this work we presented our understanding of design as a communication process and proposed a meta-model that accommodates all the activities of the design/development process. While being a general model for communication, it captures the necessary contributions that should occur among the different agents of communication, as senders and receivers in this communication process. Also, it stresses the fact that, in order to design the final interface, designers must carefully design other messages and choose appropriate channels to convey them, always considering the projects' resources and limitations.

In summary, the fractal communication model organises an analysis space which unifies some current independent practices of design while brings new issues which deserve more investigation to light. Several questions can be addressed with support of this meta-model, as for example: Are there enough channels and messages to answer to usability questions? (who are the users, what are the usability problems, how and why they occur)? Do the messages relative to each question propagate efficiently to the project team? Is there a type of communication act best suited for the design of a specific artefact? What are the consequences of a "broken" propagation resulting from lack of appropriate channel or message?

A Communicative Walkthrough - an inspection method based on the fractal model of communication is being proposed [23] for systematic analysis of the effectiveness of communicating usability issues among the groups involved with the product design and development. Further work is being done through case studies to evaluate how different design approaches fit into the proposed meta-model. The possibility of making explicit the communication levels stressed by the application of a particular technique or methodology brings new possibilities of analysing all the elements involved in the design of computer artefacts.

The context for developing off-the shelf products, as is the case at ORG, constrains, up to some extent, the interaction with users. With the design of new channels and messages to collect and to propagate data about what the usability problems are, usability engineers could make the communication with users much more effective. This view of the organisational context of software development allows to search for continuous improvement in the process and potentially to develop more usable and useful software.

Acknowlegdements

This work was partially supported by the CNPq and FAPESP.

6. REFERENCES

1. Adler, P S, and Winograd, T A (1992) (eds.) Usability: turning technologies into tools, Oxford University Press, 1992.
2. Andersen, P B, (1997) *A Theory of Computer Semiotics*, Cambridge University Press, Cambridge, UK.
3. Beyer, H B and Holtzblatt, K, (1998) *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann Publishers.
4. Brown, G, (1995) *Speakers, listeners and Communication*, Cambridge University Press, Cambridge, UK..
5. Coelho Neto, J T,(1996) *Semiótica, informação e comunicação*, 4ª ed., Editora Perspectiva, SP Brazil.
6. de Souza, C S, (1993) The semiotic engineering of user interface languages, *International Journal of Man-Machine Studies*, v. 39, p. 753-773.
7. Ehn, P., Lowgren, J. (1997) Design for Quality-in-use: Human-Computer Interaction meets Information Systems Development, in: Helander et al (eds.) Handbook of Human-Computer Interaction, Elsevier Science.

8. Gould, J, and Lewis, C, (1985) Design for Usability: Key Principles and What Designers Think. *CACM*, pp. 300-311, v. 28, n. 3.
9. Grudin, J, (1991) Interactive systems: bridging the gaps between developers and users. *IEEE Computer*, April.
10. Grudin, J., Markus, M.L. (1997) Organizational Issues in Development and Implementation of Interactive Systems, in: Helander et al (eds.) Handbook of Human-Computer Interaction. Elsevier Science.
11. Hix, D., Hartson, H.R.(1993) Developing user interfaces: ensuring usability through product and process. John Wiley & Sons.
12. Jakobson, R. (1960) Closing Statement: Linguistics and Poetics, in: Sebeok, A, ed., *Style in Language* Cambridge, Mass.: The MIT Press.
13. Liu, K. (2000) *Semiotics in Information Systems Engineering*, Cambridge University Press
14. Mcluhan, M, (1964) *Os meios de comunicação como extensões do homem*, Cultrix, SP Brazil.
15. Muller, M, (1997) Participatory Practices in the Software Lifecycle, in: Helander et al (eds.) Handbook of Human-Computer Interaction. Elsevier Science.
16. Nadin, M.(1988) Interface Design and Evaluation – Semiotic Implications, in: Hartson, H. R and Hix, D, (eds.), Advances in Human-Computer Interaction. v. 2. Ablex Publishing Corporation, 1988.
17. Norman, D A, (1988) The Psychology of Everyday Things, Basic Books.
18. Norman, D A, (1986) Cognitive Engineering, in: Norman, D and Draper, S W, ed., User centered system design: new perspectives on human-computer interaction Lawrence Erlbaum Associates.
19. Norman, D.A. (1998) *The Invisible Computer*, The MIT Press, Cambridge, Massachusetts
20. Rheinfrank, J, Hartman, W R. and Wasserman, A, Design for Usability: Crafting a Strategy for the Design of a New Generation of Xerox Copiers, in: Adler, P S and Winograd, T A, (eds.), Usability: turning technologies into tools, Oxford University Press.
21. Salles, J.P. (2001) O Modelo Fractal de Comunicação: Criando um Espaço de análise para Inspeção do Processo de Design de Software. PhD Thesis, May, 2001.
22. Salles, J.P., Baranauskas, M.C.C., Bigonha, R.S. (2000) A Communication Model for the Interface Design Process. Workshop in Semiotics, CHI2000
23. Salles, J.P., Baranauskas, M.C.C., Bigonha, R.S. (2001) Towards a Communication Model Applied to the Interface Design Process (forthcoming)
24. Schön, D and Bennet, J, (1996) Reflective Conversation with Materials, in: Winograd, T, (ed.), Bringing Design to Software ACM Press.
25. Shannon, C L and Weaver, W, (1949) The mathematical Theory of Communication, Urbana, Univ. of Illinois Press.
26. Westley B H and MacLehan, M S, (1972) Un modelo conceptual para la investigacíon en comunicaciones, in Smith G, (ed.), Comunicacíon y cultura Nueva Visión, Buenos Aires.
27. Winograd, T. (1996) *Bringing Design to Software* ACM Press.