4th International Conference on Enterprise Information Systems, *Ciudad Real – Spain, 3-6, April 2002*

ANALYSING COMMUNICATION IN THE CONTEXT OF A SOFTWARE PRODUCTION ORGANISATION

M. Cecilia C. Baranauskas
Institute of Computing, State University of Campinas, Unicamp, Cx Postal 6176 13083970 Campinas, SP, Brazil
Visiting Fellow at the Department of Computer Science, University of Reading, UK
Email: cecilia@ic.unicamp.br

Juliana P. Salles
Computer Science Department,Federal University of Minas Gerais, UFMG, Belo Horizonte, MG, Brazil
Email: jusalles@dcc.ufmg.br

Kecheng Liu
Department of Computer Science, University of Reading, Whiteknights, Reading RG6 6AY, UK
Email: k.liu@reading.ac.uk

Keywords: Communication, Organisational Semiotics, Design Process, Design Quality, Human-Computer Interaction

Abstract: While quality has been widely stressed in literature as a goal of the software design methodologies, quality as a result of the interaction among the actors involved in the design and development processes has not received the same attention. This work aims to investigate the software production process by addressing the communication among work groups in the organisation. Our focus is on understanding the communication process that takes place among the groups, considering that the computational artefact emerges as a result of the communicational acts issued between people with different roles in the process. We base our understanding of communication in semiotic foundations, to propose a framework for analysing communication in the whole process of system design and development. The design process of a real organisation that produces commercial software illustrates our main ideas.

1. INTRODUCTION

Organisational 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 also have concerns about organisational aspects influencing the design and use of computational systems. HCI and information systems development has evolved from different backgrounds and focus to a common end: to achieve high quality in software usage [3]. Quality has been widely stressed in literature as a goal of the software design methodologies, and it has been grounded in the Software Engineering tradition, focusing on optimising technical quality in the product and in the process. For the HCI community, quality has been understood as how well applications fit the needs of individual users. We argue that both perspectives should be considered in order to achieve a global view of quality in the whole design and development process. This work aims to investigate the soft-ware production process by addressing quality in the communication that takes place among the different work groups involved in a software production organisation, including several categories of users.

A comprehensive and integrated understanding of the connections among design, development, and evaluation activities are necessary at all levels of the software de-sign process. Understanding and integrating the whole process requires an under-standing of how different groups in the organisation acquire and communicate their views. As pointed out by some authors [5], [6] this seems to be true in any large sys-tem (e.g. a new automobile design). Participatory [9] and Contextual Design [2] pro-pose 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 inspect quality in the process and in the product, as a consequence.

The study of the signs used in communication and the rules operating upon them and upon their user forms the core of the study of communication. Thus, there is no communication without a system of signs. Semiotics as a discipline concerns the analysis of signs or the study of the functioning of sign systems. Organisational Semiotics (OS) is one of the branches of Semiotics particularly related to business and organisations [8]. As Liu [13, p.19] points out, the study in OS is based on the fundamental observation that all organised behaviour is effected through the communication and interpretation of signs by people, individually or in groups. The aim of OS studies is to find new and insightful ways of analysing, describing and explaining structure and behaviour of organisations, including their inner workings, and the inter-actions with the environment and with one another.

Our focus in this work is on understanding the communication process that takes place among groups in a software design organisation, in order to inform about the main production process. A meta-model, proposed in previous work [12], organises an analysis space that is extended based on OS foundations towards a framework to inspect quality in communication.

The paper is organised as follows: Section 2 presents the theoretical foundations for the work: the semiotic approach to modelling an organisation, and the concept of communication. Section 3 discusses the software design process as a matter of communication and presents the fractal model of communication. Section 4 outlines a framework for inspecting quality of communication based on the OS foundations and on the fractal model of communication, and discusses results of applying it to a real organisation. Section 5 concludes.

2. THE THEORETICAL FOUNDATIONS

2.1 The Semiotic Approach to Modelling an Organisation

According to the OS approach, an organisation can be seen as an information system where agents employ signs to perform purposeful and co-ordinated actions. An organisation can be characterised by a structure of three layers: the informal, the formal and the technical, which Stamper [15] names “the organisational onion”. The whole organisation is regarded as an informal information system in which its culture – values, beliefs, habits and pattern of behaviour of each individual member plays an important role. It is in this layer that meanings are established, intentions are under-stood, beliefs are formed, commitments are made and responsibilities are negotiated through discussion and physical actions. Inside the informal layer there is the formal information system layer, where literate culture dominates through rules that specify how the work should be done and how the tasks should be performed. In this layer, form and rule replace meaning and intention. As far as the tasks are mechanistic and repetitive the rules and procedures can help to achieve efficiency. However, the vision of people being used only to transmit and process sign-tokens is certainly problematic. As people lose understanding of the meanings of the signs they process they lose the global context and cannot anymore evaluate their actions or relate it to the other agents actions to know whether they are doing what was supposed to be done. The third layer concerns the technical system that is placed inside the formal layer to automate part of the formal system. It presupposes well-defined work processes, clearly understood human responsibility for the jobs and explicitly specified rules for operations. However, as far as the introduction of a technical system in an organisation is intended not just to automate work but to support work practices, a clear under-standing of the whole process, as well as flexibility to change as the organisation changes is needed. In summary, the three layers have a strong connection and a representation suitable to adapt and adjust to the global view of the organisation is needed for modelling it. To go deeper in the understanding of this whole structure, the communication among the groups involved in the organisation should be carried out.

A traditional method for organisational analysis is the hierarchical diagram showing the different positions and reporting relationships. This type of representation does not help us to understand neither what an organisation does nor how it does it. The organisational semiotics approach helps us to focus on the organisational functions rather than just the structure, providing morphology of the tasks and functions of an organisation. Three areas of organisational functions and its consequent types of tasks and of associated norms are defined: the substantive, the message passing, and the control areas. Functions in the substantive area are concerned with and directly contribute to the organisation main objective. Functions in the message passing area are related to the information people must have about the relevant facts about the work, to co-ordinate the temporal and spatial use of resources from the substantive area. Functions in the control area aim to reinforce the substantive and message passing areas through tasks that include monitoring and evaluation of substantive and message passing actions. Each area can be further sub-divided in the same way until the level of detail is sufficient to the analysis required. A balance of the three areas should be observed in a healthy organisation. As Liu [8] points out, an “unhealthy” or badly designed organisation will consume a great part of its energy with message-passing and control subsystems.

According to the semiotic approach we are considering, a sound modelling must cover the issues in semantic, pragmatic and social levels. To clarify the semantic problems and represent a social organisation clearly is the most critical and difficult task [8]. It is important to know how people are involved and co-ordinate their tasks and duties. This means more then knowing how they implement a plan or conduct their particular jobs, we are interested in knowing the communicational actions that take place among people in each position of the organisation. For this purpose we need to focus attention of analysis on several levels of the communication among the groups involved in the production organisation. This means identifying communicative agents, messages, channels, and their substantive, communicative and controls actions. An analysis of the communication, taken in a broad sense, among the agents could inform about the “health” of the organisation.

To understand communication in the organisational 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. In the next section we will examine some of them in order to highlight the essential aspects of the concept.

2.2 Communication Models

There are two main schools in the study of communication, with different standpoints and associated models. The first one, which Fiske [4] names the “process school”, sees communication as the transmission of messages. It is concerned with how senders and receivers encode and decode and how the transmitters use channels and media of communication, with efficiency and accuracy. This school sees communication as a process by which one person affects the behaviour or state of mind of another. If the effect on the receiver is different from the intended by the sender, it is considered a failure in communication and the causes of it are searched in the stages of the process. This school has its foundation on the Shannon and Weaver’s Mathematical Theory of Communication [14]. 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.

This mathematical theory first arose in telegraphy and telephony, being developed for the purpose of measuring the information content of telecommunication signals. It concerned only the signals themselves, as transmitted along wires or broadcasted, and is abstracted from all questions of "meaning". As a theory, it lies at the syntactic level of sign theory and is abstracted from semantic and pragmatic levels.

Other models developed with other concerns were derived from this very influent model. Westley and MacLehan [16] proposed a model for 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. To understand the linguistic process of verbal communication, Jakobson [7] proposed a model in which he identifies 6 factors that constitute every verbal communication act: addresser, addressee, context, message, contact, and code.

The second school sees communication as the production and exchange of meaning [4]. The main method of study in this school is Semiotics. This school is concerned with how messages, or texts, interact with people in order to produce meanings. It does not consider misunderstanding to be necessarily evidence of communication failure, as they may result from the cultural differences between the parts involved in communication. Searle, as described in [8], studied the roles of language usage in communication and defined an illocutionary act as a basic, meaningful unit of human communication. The fundamental aspect of this model is that, during conversation, the hearer receives that communication unity and interprets the meaning based on his own perspective, which can be different from the meaning intended by the sender. It is interesting to notice that the sender himself is a hearer while utters an expression and this cause also on him a perlocutionary effect. This model illustrates how the agents involved in communication alternate roles as sender and receiver of messages.

Our understanding of communication in this work is in line with the school of Semiotics. To understand the computer as media, and the way it can shape communication, we need to go deeper in situating the concept in the process of software construction.

3. . THE SOFTWARE DESIGN PROCESS AS A MATTER OF COMMUNICATION

Design has been understood as a technical term referring to a particular step in the software production 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 [17],[11],[13], the English language usage is adopted for the term, meaning the invention and organisation of any structure. Therefore, we adopt here a broad understanding of design referring to the software design as a process embracing all the intertwined activities related to the creation and development of a computational artifact: the design of the look and feel, the functionality definition, the codification etc.

According to semiotic approaches in HCI, the software interface can be understood as a message sent from designers to users using the computer as channel [1], [10]. 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. 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' organisations, etc. The use of different channels is related to the needs of the communicational acts: the nature of information desired, the projects' contexts, the audience language, etc. The focus of this work is in the communication among the several groups involved in the design and development of computational artefacts. The dialogue between designers and users is just one aspect of this communication.

3.1 The Fractal Model of Communication

We argue that in order to design the interface as a message, not only designers and users, but also the remaining actors 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. Furthermore, in designing the artefact, the designer establishes a conversation with the design medium, in the sense explained by Schon [13]. This level of communication is also part of the process and thus should be considered.

A fractal model of communication proposed in our previous work [12] captures the structure of the communication process involved in the whole process of software design. It 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 1 illustrates the main concepts of the fractal model of communication.

In this diagram, nodes represent the agents in communication (A and A') and the channels (C and C') used by them. The links are bi-directional, which means that the agents share (both send and receive) messages. Nodes C' represent the fractal nature of communication. Different foci of the design process can be highlighted: the de-signer-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, in the sense discussed by Schon (A-C), and the user-computer communication (C-A') are represented in a second level of the fractal, having C' as channels.

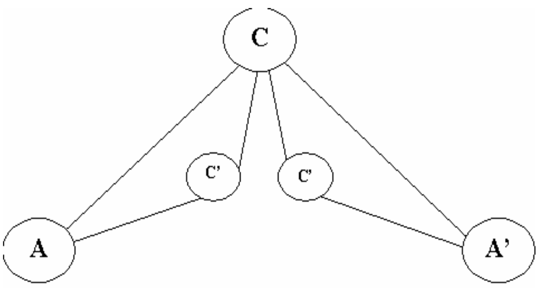

Figure 1: The fractal model of Communication

An inner level could be opened for analysis, showing a third level of the fractal structure, as illustrated by Figure 2. 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.

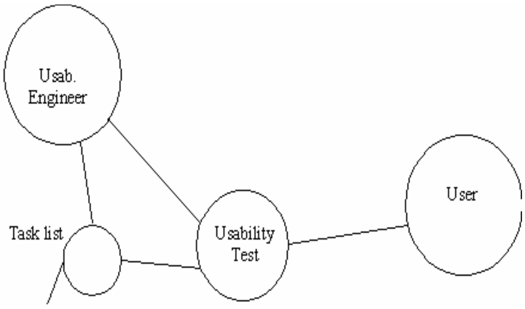

Figure 2: An instance of the fractal inner level

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 software interface, or the unity message, many fractionated messages are being ex-changed. 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 co-operate on the development of the interface.

4. TOWARDS A FRAMEWORK TO INSPECT QUALITY IN ORGANISATIONAL COMMUNICATION

As we have already seen in the previous sections, from the OS standpoint, an "organisation" is seen as an information system where agents employ signs for communication towards the performance of purposeful and co-ordinated actions. We need now to focus attention of analysis in the several interconnected levels of communication among the groups involved in the work. Different levels of the Fractal Model of Communication can be represented to grasp communication in the different layers (informal, formal and technical) and to show the connections between the nested layers. Furthermore the morphology of functions can also be identified in order to evaluate the balance among substantive, message passing and control functions of the agents.

Understanding the relationships among the several groups involved in a production organisation, and how they articulate actions towards the main objective involves understanding how they communicate with each other in the organisational structure. The quality of communication as a whole depends on the quality of communication between agents in each level of the fractal structure, which should consider the substantive, message passing and control actions.

4.1 Modelling Communication in a Product Development Cycle: an Example

To exemplify the main ideas proposed for the framework, a major software design company, here given the pseudonym "ORG" will be considered. This organisation 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 are marketing people, program managers, designers, usability engineers, developers, testers, and support people. Customers, market, beta-users and users are four types of external agents called "users": customers are people who have already bought the product and receive the service of the support team for problems with the product. Market represents the 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. Beta-users are people already using the beta version of the product. Actually, these four categories of people are potential end-users for the product.

Typically the life cycle of the ORG products has three phases: planning, development and stabilisation phases. During the planning phase, communication between the marketing and the program 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 takes 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 release. 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

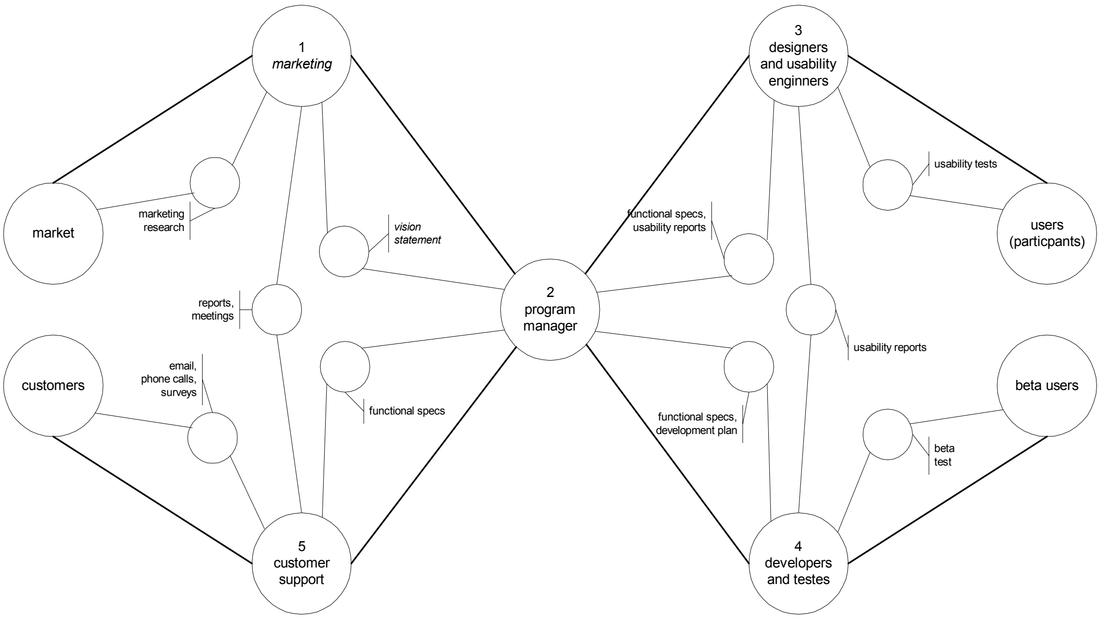

Figure 3: Meta-model for the product development cycle at ORG.

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. For example, 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.). Figure 3 illustrates the Fractal Model of Communication applied to the product development cycle of this particular organisation. An overview of the communication among the groups is shown in a simplified way, with two levels of the fractal structure.

Marketers have their tasks and responsibilities in the informal layer, as they are in charge of ensuring a product meets the need for which people will buy it. They have a message passing function as they act as channel between people who would buy the product and the program manager. Their substantive functions are concerned with marketing research. Their substantive actions consist of identifying what kind of customer makes up the market. Their substantive objects of action are questionnaires, feature list, list of customer needs, requirement list, wish list, and so on. Besides dealing with issues related to the market, they are engaged in product planning activities, with the product manager. Despite their substantive actions being concerned to the informal layer, they focus on demographics rather than on users' work practice. No control functions are associated to them.

Program managers are people who make a bridge from the informal layer to the formal layer. Their primary function is control, as they are in charge of the whole project, and have as primary responsibility to ensure the promised features are being finished on time and with acceptable quality. The object of action in the control area is the project schedule. Their substantive functions are to specify what the overall product (for example a computational application) must offer, and to manage the project. Their substantive actions involve defining the vision document (sharing it with marketers), and defining the functional specifications (sharing it with developers). Their substantive objects of action include the vision documents and the functional specification. They have a function of message passing as they act as channel between marketers and developers. Furthermore, they act as channel between all the other agents, excluding the four categories of users.

Developers (programming engineers) have their tasks and responsibilities in the technical layer with focus on technology and on what makes a clean program code. They are receivers of marketers message through message passing mechanisms of the program manager, as they get directions from the marketing, through the program manager, who specifies what the overall system must offer. Their primary function is substantive, related to the implementation of system functions. Their substantive action is codifying and their substantive objects of action are the functional specification and the program code. Their control functions are carried by as a consequence of their communication with testers, who test code and report bugs.

Usability engineers are the people whose main actions are in the informal layer as they have the direct observation of the problems experienced by users interacting with the application. Their primary function is control, as they function as quality control agents for product design to assure usability to it. Their substantive function is quality control of the usability of the product. Their substantive actions are related to designing and applying usability tests. Their substantive objects of actions are basically usability tests and usability reports. At ORG designers are people in charge of the product interface design. They should lay a bridge between the informal and formal layers. Their function is primarily substantive and their substantive actions involve designing the interface. Thus their substantive object of action is the product inter-face. Their control functions are conducted as a consequence of communication with the usability engineers. Together, designers and usability engineers have message passing functions as they act as channel between users and developers, and between users and the program manager.

Customer supporters are people who make the bridge from the technical to the in-formal layer. Their substantive function is concerned to responding to problems en-countered by clients in using the application. Their substantive actions involve emailing, phoning or sending a survey to the clients. Substantive objects are questions from and answers to the clients.

The four categories of agents representing users are in the social layer and their functions in the organisation are mostly concerned with the control area as they feed message passing mechanisms of usability engineers, testers and customer support groups.

4.2 Discussion

The instantiation of the framework presented tells us that the process is marketing-oriented, which is consistent with the economic 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). His object of conversation with developers and testers is the functional specification (substantive area), communicating the intended functions for the product and the priorities to implement them (control area). From the developer perspective, this is an efficient message as both program managers and developers share the specification document as one of their substantive objects. 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. There is a lack in the message passing function between the program manager and the designers/usability group concerning the vision document, which should be part of the substantive actions of the latter. Designers and usability engineers' vision should be considered especially in the planning phase of the cycle.

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 who marketing is targeted to; clients are users in communication with the support group; beta-test participants are users communicating to testers. Nevertheless, the identification of the "user" is only in the substantive actions of the marketing group as it is the only agent involved in knowing (who are) the prospective users. Other agents in the process demand understanding about the users: designers, for example, should know for whom they are designing. Marketers use a specific substantive function (market research) to get a demographic description of the prospective users. However, the substantive objects of action of the marketing group do not necessarily fit interests of the designers group. New substantive actions could be introduced for the agents in communication with users to convey a more complete view of the user and to accommodate specific necessities of substantive functions of designers and usability engineers.

The substantive object of action of the customer support group and of the testers group would be powerful additional objects to identify (what are) the usability problems, as these agents have direct access to problems pointed out by users. However, in the first case, an expansion in the substantive area of the usability group would be necessary to include new substantive objects of action. In the second case, new messages should be designed in the beta-test to address also usability issues. Using data from beta-tests and from the support group records would mean the utilisation of al-ready existent substantive objects of action. 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 being 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 they are tailored to the concerns and work style of the other group. This means that they must share substantive objects of action. The program manager acts as a channel between the marketing and designers/usability engineers using the functional specification document. While developers and marketers share substantive objects of action - developers think in functions as units of implementation consistently with marketing objectives, designers need a much broader view that is not communicated by the specification document-designers and marketers do not share the same substantive objects of action. Also, the communication between usability engineers and developers through the usability re-port and email does not seem adequate to the developers whose substantive objects of action are much more technically focused. As a consequence the message from us-ability group does not reach the developer.

Results of a first analysis on the meta-model using the proposed framework point to the necessity of new channels and messages, as well as more shared substantive areas among some agents to enhance communication among key elements of the process. The dynamic enabled by improving quality in communication 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 framework based on Organisational Semiotics to model communication in a software design organisation. The framework captures the sharing of substantive areas that should occur among the different agents of communication, as senders and receivers in this communication process, and the aspects of control and message passing among the agents. It also stresses the fact that, in order to design the final product, several messages should be carefully designed and appropriate channels must be chosen to convey them.

Summarising, the Fractal Communication Model organises an analysis space in which foundations from Organisational Semiotics can be used for modelling communication among agents in the organisation. 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: We gratefully acknowledge the Brazilian agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for partial financial support.

REFERENCES

Andersen, P. B.,1997. *A Theory of Computer Semiotics*. Cambridge University Press, Cambridge.

Beyer, H. B., Holtzblatt, K., 1998. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers.

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.

Fisk, J., 1990. *Introduction to Communication Studies*. Routledge, London.

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.

Hix, D., Hartson, H.R., 1993. *Developing user interfaces: ensuring usability through product and process*. John Wiley and Sons, New York .

Jakobson, R., 1960. Closing Statement: Linguistics and Poetics. In: Sebeok, A. (ed.): *Style in Language*. The MIT Press, Cambridge, Mass.

Liu, K., 2000. *Semiotics in Information Systems Engineering*, Cambridge University Press.

Muller, M., 1997. Participatory Practices in the Software Lifecycle. In: Helander et al (eds.) *Handbook of Human-Computer Interaction*. Elsevier Science, New York.

Nadin, M., 1998. Interface Design and Evaluation – Semiotic Implications. In: Hartson, H. R and Hix, D, (eds.): *Advances in Human-Computer Interaction*. v. 2. Ablex Publishing Corporation.

Norman, D.A., 1998. The Invisible Computer, The MIT Press, Cambridge, Massachusetts.

Salles, J.P., Baranauskas, M.C.C., Bigonha, R.S., 2000. A Communication Model for the Interface Design Process. *Workshop on Semiotic Approaches to User Interface Design*, CHI2000.

Schön, D., Bennet, J., 1996. Reflective Conversation with Materials. In: Winograd, T, (ed.), *Bringing Design to Software* ACM Press.

Shannon, C. L., Weaver, W., 1949.*The Mathematical Theory of Communication*, Urbana, Univ. of Illinois Press.

Stamper, R. K., 1973. *Information in Business and Administrative Systems*. John Wiley and Sons, New York.

Westley B. H., MacLehan, M. S., 1972. Un modelo conceptual para la investigacíon en comunicaciones. In Smith G, (ed.): *Comunicacíon y cultura* Nueva Visión, Buenos Aires (1972)

Winograd, T. (ed.), 1996. *Bringing Design to Software*, ACM Press.