BRINGING SOCIAL CONSTRUCTS TO THE INFORMATION SYSTEM DEVELOPMENT PROCESS:
Contributions of Organisational Semiotics

Carlos Alberto Cocozza Simoni, M. Cecília C. Baranaukas
Institute of Computing, UNICAMP – State University of Campinas, Av. Albert Einstein 1251, Campinas, Brazil

Rodrigo Bonacin
CESET, UNICAMP - State University of Campinas, Rua Paschoal Marmo, 1888, Limeira, São Paulo, Brazil

Keywords: Software Engineering, Organisational Semiotics, Unified Process.

Abstract: Literature has shown the influence of the social, cultural and organisational aspects involved in the process of developing information systems. The Unified Process (UP) has been widely used in the software industry, but literature has shown its drawbacks when applied to the modelling of human actions in the social and organisational contexts. Our research investigates the use of Organisational Semiotics (OS) methods combined with the UP to compose a complete cycle of system development, aiming at bringing social constructs to the development process of information systems.

1 INTRODUCTION

As discussed by Ehn and Lowgren (1997), the first approaches to IS development can be characterized by a strong belief in systematic design methods based on mathematical and logical theories. Research interest in accuracy and technical control has guided these approaches. The main assumptions behind them, as suggested in some methods from Software Engineering (SE) (Jacobson *et al*, 1999, Sommerville, 2001; Pressman, 2001), seem to be that the users (end-user, client, customer, stakeholder or problem owner) are supposed to give complete and explicit descriptions of their demands in terms of the system to be developed.

UP literature has also pointed out that: "At present, formalism during the analysis phase should be restricted to the syntax and semantics of the static structure of the system. We do not know of any sound, practical and strictly formal technique for satisfactory specification of the system's dynamic behaviour at this critical phase. A more practical, descriptive technique is therefore preferable to a mathematical, formal method that is not yet fully mature. A formal technique is better used later on, especially during implementation. As the more formal techniques mature, they will probably be preferred" (Jacobson et al., 1996, p.14).

With the popularisation of the Object Oriented (OO) approach and the UML (Unified Modelling Language), the OO community has proposed new processes to specify how to work with UML models during software development. Nowadays, the Unified Process (UP) and its commercial versions are widely used in the software industry.

We agree with Dourish (1995) and other authors in arguing that our interactions with technology are embedded within social and organisational situations and, as so, their influences must be taken into account in the analysis and design of systems. Literature in Organisational Semiotics (OS) (Liu et al, 1994; Liu, 2000; Heusden and Jorna, 2000 and Stamper et al, 2000) has shown that the social, cultural and organisational aspects involved in the problem must have a more decisive role in the process of developing the information system, while traditional methods have emphasised the technological solution itself.

Although the UP deals with very important issues related to software development, research in Organisational Semiotics (OS) have shown some weaknesses of the OO paradigm when applied to the modelling of human functions in the organisational

Alberto Cocozza Simoni C., Cecília C. Baranaukas M. and Bonacin R. (2005).
BRINGING SOCIAL CONSTRUCTS TO THE INFORMATION SYSTEM DEVELOPMENT PROCESS: Contributions of Organisational Semiotics.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 112-119
DOI: 10.5220/0002523201120119

context. Xie and Liu (2003) have shown some aspects of the RUP that could be improved by the Organisational Semiotics methods.

Thus, on one hand we have the theory and methods of OS, which allow us to deal with the social constructs of organisational contexts in which the software system will be embedded; on the other hand we have methods for software design and implementation accepted by the software industry. In the whole picture, the best of the two worlds seems to be necessary to a broader understanding of the problem of developing information systems that make sense to their users in their organisational contexts. Our work investigates the use of the OS methods in a combined way with the UP, to compose a complete cycle of information system development. We have been practicing OS and UP techniques together (Bonacin et al, 2004), as well as OS within a traditional system development cycle (Simoni and Baranauskas, 2004a). The first outcomes have shown that this practice has allowed the analysts, together with the problem owners and stakeholders, to have a deeper understanding of the problem and its context, leading to potentially more meaningful solutions.

In this paper we present an approach that include a new and valuable vision of the organisational context based on OS methods. The paper is organised as follows: Section 2 presents the Unified Process and some key concepts of the OS methods that could have a role in the UP. Section 3 presents the proposed approach, and discusses benefits and drawbacks of the proposed approach, and Section 4 concludes the paper.

2 THEORETICAL BACKGROUND

The Unified Process - UP "has emerged as a popular software development process for building object-oriented systems" (Larman, 2002), and the software development process is understood as an approach to building, deploying, and maintaining software. The Rational Unified Process – RUP has been widely adopted and we can find some commercial extensions of RUP, as EUP (Enterprise Unified Process), which cover the concept of maintenance and support in the development cycle (EUP, 2004).

The UP is based on 6 considered best practices in the software development industry (Leland et al., 2002): iterative software development, management of requirements, component-based architectures, visual models, quality control and configuration.

Iterative development stands above the other practices (Larman, 2002), and the "development is organized into a series of short, fixed-length mini-projects called iterations". Each iteration includes its own development cycle (analysis, design, implementation and test), and the result of each iteration is a tested, integrated and executable system. The iterations are organized across four major phases: Inception, Elaboration, Construction, and Transition.

UP describes work activities, in the development cycle, as 'disciplines' (or workflows), which is a set of activities in one subject area: Business Modelling, Requirements, Design, Implementation, Test, Deployment, Configuration and Change Management, Project Management, Environment and Operations & Support.

Figure 1 shows the life cycle representation, in which the process is structured along the dimensions: phases, disciplines and iterations.

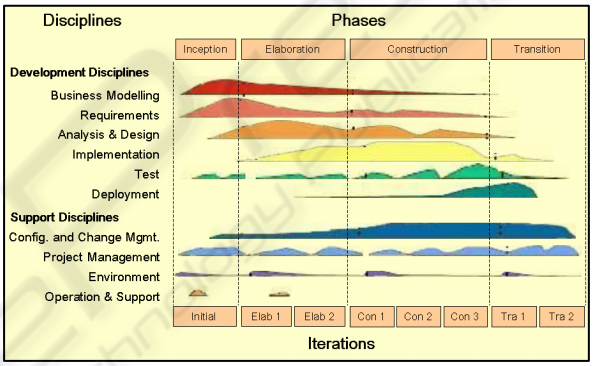

Figure 1: The Life Circle for Unified Process.

The requirement analysis in the UP uses the Use Case Model to explore and record the functional requirements. This model play "the heart-and-center overarching requirements approach, replacing other requirements documents as the central element; use cases suffuse and drive the requirements work…" (Larman, 2002, pg. 44), and is a mechanism to capture goals and system requirements, help in keeping them simple and understandable for all stakeholders. It is done by writing stories of using a system to help fulfil the stakeholders' goals. "A use case is a collection of related success and failure scenarios that describe actors using a system to support a goal' (Larman, 2002, p. 47).Other requirements can be recorded in the Supplementary Specification artefact. The main information captured in a Use Case are: primary actors, stakeholders and interests, preconditions and success guarantees, main success scenario (basic flow) and steps, alternate flows, special requirements, and technology and data variations list.

Organisational Semiotics (OS) provide us with methods to construct a meaningful understanding of the organisational context, which will embed the Information System. Therefore the OS methods

could be useful in extending UP to deal with the influence of the organisational aspects of social nature in the definition of the system requirements.

In OS, an organisation can be seen as an information system in which interdependent links between the organisation, the business process and the IT system occur (Liu, 2000). At an informal level there is a sub-culture where meanings are established, intentions are understood, beliefs are formed and commitments with responsibilities are made, altered and discharged. At a formal level, form and rule replace meaning and intention. At a technical level part of the formal system is automated by a computer-based system. The informal level embodies the formal that, by its turn, embodies the technical level, meaning that changes in some level have impact in the other levels. The information system is impacted by and reacts to the environment, as Figure 2 illustrates. In a semiotic perspective, different layers of meaning must be considered in the information system analysis and software design (Stamper, 1973).

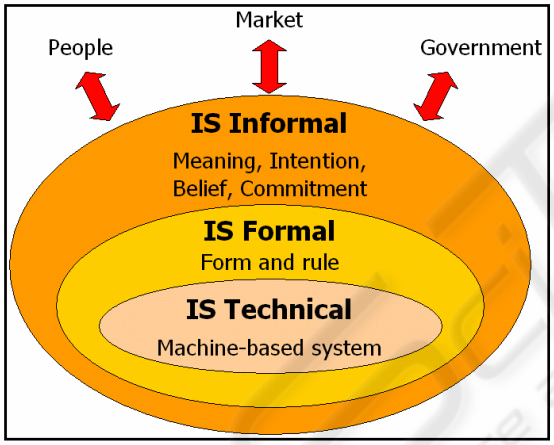

Figure 2: The Organisational Onion, adapted from Liu (2000, p.109).

It is agreed that OS methods can provide a better understanding for the interested parts of a focal problem, their requirements and intention, as well as the restrictions not only regarding the information system, but the software system as well.

Our approach considered some of the MEASUR (Methods for Eliciting, Analysing and Specifying Users' Requirements), which resulted from a Stamper research work in the late 70´s (Stamper, 1973 and 1993). Stamper proposed a set of methods to deal with all aspects of information system design, related with the use of signs, their function in communicating meanings and intentions, and their social consequences. From MEASUR we are working with the following methods:

PAM – Problem Articulation Methods: consist of a set of methods to be applied in the initial phase of a project, when the problem definition is still vague and complex. The analyst is helped in defining system units that will be validated by stakeholders using Stamper's Semiotics Framework (Liu, 2000). PAM is composed by the following methods:

Stakeholder Analysis: allows to investigate all the interested parts (the stakeholders), that directly or indirectly have influences or interest in the information system in analysis.

Evaluation Framing: allows to identify, for each stakeholder, their interests, questions and problems, in order to discuss possible solutions.

Semiotic Diagnosis: traditional system development methodologies emphasize technical issues (physical world, empirics and syntactic) and the analyst misses the opportunity of analysing other levels of relationship (semantic, pragmatic and social), which direct or indirectly affect aspects of the system design. The use of the Semiotic Framework allows us to examine the organisation as a social system that is constructed through the use of information.

Morphologic Analysis: allows the investigation of the norms that govern people's behaviour within the systems. Three main components of the analysis are the substantive, which focus on aspects that contribute directly to the organisational objectives, the communicational and the control aspects.

Collateral Analysis: allows the analysis of relationships between unitary systems that compose the complex system. It locates the effective limits of the system in the environment, the focal system and its infrastructure. The collateral systems provide maintenance, backup and recovery, inputs and outputs, etc.

SAM – Semantic Analysis Method: assists analysts and users or problem owners in eliciting and representing their requirements in a formal and precise model. With the analyst in the role of a facilitator, the required system functions are specified in the Ontology Model, which describes a view of responsible agents in the focal business domain and their actions or patterns of behaviour called "affordances". It is a process of conceptualisation of a business organization, in which the organisational behaviour is analysed and captured in the Ontology Model. In Semantic Analysis the ontological relationship is considered as the most fundamental relationship to be modelled. The result of the Semantic Analysis is complemented with the dynamic aspects (constrains, rules, etc.), obtained with the Norm Analysis.

NAM – Norm Analysis Method: focuses on social, cultural and organisational norms that govern the actions of agents in the business domain. A norm, in a formal or informal sense, defines a responsibility of an agent engaged in a task, or condition under which certain actions may (must, must not, etc.) be performed by the agent. Each specified norm is associated with an action pattern described in the Ontology Model.

3 AN APPROACH INTEGRATING OS AND UP

Jacobson *et al* (1999, p. 342-343) show that before starting the Inception Phase it is necessary to define the problem context ("you had some knowledge of what you are going to do"), and its boundaries, to get estimative about cost, schedule and return-of investment (ROI). They mention that this kind of knowledge can be gained from:

- Studies from marketing or management people;
- A department or general management, sometimes supported by business engineers that "feels" the need for a software system, and provides "a description of what they have in mind";
- Request for proposal that "often contains considerable details of requirements".

Also the same authors consider that the "major challenge is that the customer, who we assume to be primarily a non computer specialist, must be able to read and understand the results of requirements capture".

We argue that this initial work on the problem clarification should be part of the information system development, considering information system in a broader sense. We propose the use of MEASUR methods, PAM, SAM and NAM to explore the problem and its context. Previous studies conduced with business organizations (Simoni and Baranauskas, 2003, 2004a, 2004b) showed that these methods were valuable to capture the core problem and its context, and provide a common language between non-technical and technical people involved in the process.

Figure 3 presents the rationale underlying our approach. PAM is used to understand the forces involved (needs, intentions, existing conflicts, etc) among the stakeholders, allowing a big picture of the problem context and the main requirements. SAM and NAM are both used to model this context, capturing informal and formal aspects related to it. Both the static (SAM – terms, concepts, etc) and dynamic aspects (NAM - constrains, rules, etc.) are modelled, and the outcomes are inputs to the UP for software development.

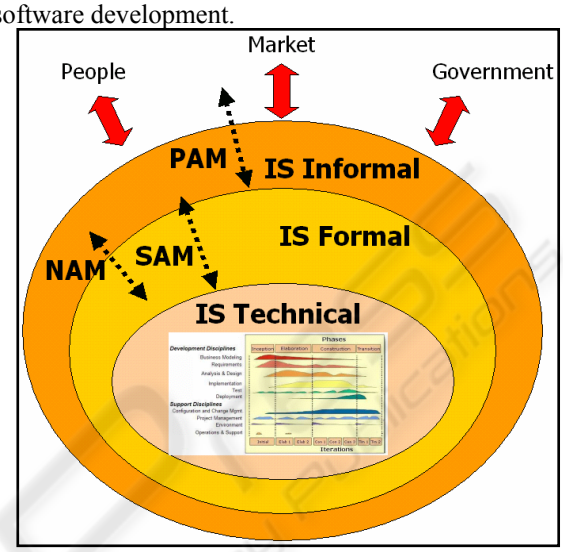

Figure 3: UP and OS integrated in an Organisational IS.

Thus, the proposed approach integrates OS and UP defining a new Discipline we named "Conception", applied prior to the Inception Phase, involving the MEASUR methods PAM, SAM and NAM, as illustrated by Figure 4. The relationships between the outcomes of these methods and features of the UP are analysed in this section. In addition, UP Disciplines are extended to encompass the Conception Phase:

Configuration and Change Management to control the versions of OS artifacts.

Project management, because we consider that, in fact, the project starts with the problem articulation.

Environment to allow the use of the OS artifacts.

In the next sections we analyse: (1) the software engineering practices of the UP and the consequence of the using OS within these practices, (2) the main UP phases and how the OS methods could contribute to complete each of them, and (3) how the OS methods could be inserted as new disciplines in the IS life cycle.

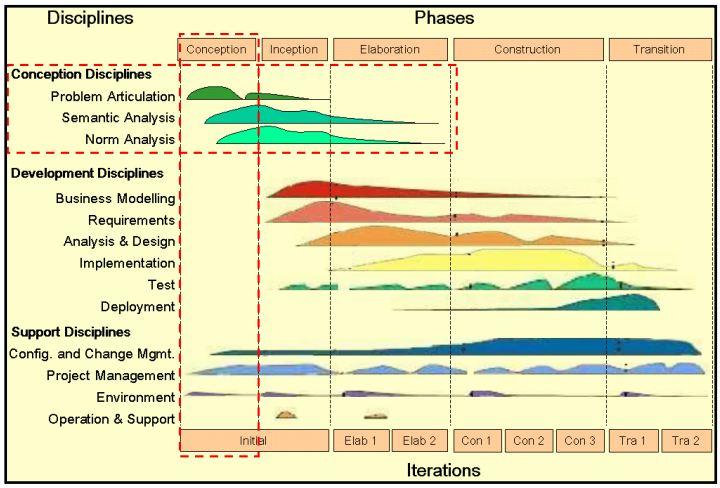

Figure 4: OS Methods integrated with UP.

3.1 The Impact of PAM, SAM and NAM in the UP Practices

As previously mentioned the Unified Process is based on six "best" software engineering (SE) practices:

Iterative Software Development: by including OS models in the UP iterative cycle we can anticipate the effects of each prototype in the organisational context. As the UP has not specific methods to model the social aspects of the organisational model, these effects would be perceived in the UP only in the later cycles. Our previous practices (Bonacin et al, 2004; Simoni and Baranauskas, 2004a) have shown that OS is compatible with iterative development and OS models facilitate the revision of the concepts in the organisational context, at the beginning of each iteration.

Management of Requirements: according to Kruchten (1999, p.8) "The challenge of managing the requirements of a software-intensive system is that they are dynamic: you must expect them to change during the life of a software project". The use of OS methods allows a better understanding of the problem focused, the stakeholders and their interests (through PAM) identifying the aspects of the organisation that are less likely to change (through the SAM) and the aspects more likely to change (by using the NAM). Therefore we can focus in norms of the organisation, manage their changes and visualise effects in the system requirements, tracking the requirements changes.

Component-Based Architectures: PAM, SAM and NAM are valuable to the "Component Based Architecture" because the social context must be considered in the definition of a system architecture. Different choices about the architecture of the software system have impact on the organisational context.

Visual Modelling for Software: some methods of PAM, like Stakeholder, Morphology and Collateral Analysis are visual models, and the SAM produces the Ontology Chart. The OO model does not have a visual representation for existential relations between elements of the organisational context. The Ontology Charts represent aspects of the context that usually are addressed in an informal way. The semantic analysis goes one step further in the direction of using a visual model to capture the semantics of the context.

Quality Control: the OS methods allow a better description of the work context; therefore we have parameters to evaluate the desired behaviour of the system. Reliability, functionality and performance are dependent on the context that we are analysing. For example: some seconds of response delay could be acceptable for one context but could not be acceptable for others; norms could specify which are the acceptable delays in a certain task and consequently the desired system performance.

Configuration: it is essential to control the changes produced during the system development. Changes in the OS models are captured and can be controlled.

3.2 The Impact of PAM, SAM and NAM in the UP Phases

As described before, the UP have four major phases: Inception, Elaboration, Construction and Transition. The OS methods could contribute for each phase:

Inception: is the phase in which the OS methods are more visible; this phase ends with the Life-Cycle objective milestone. Regarding the evaluation criteria for the inception phase:

Stakeholder concurrence on scope definition. The PAM allows the identification of the stakeholders, their interest, questions and problems. The PAM also discusses possible solutions and locates the effective limits of the system in the environment.

Requirements understanding. PAM, SAM and NAM allow a wider view of the requirements elicitation; they analyse the semantic, pragmatic and social levels of the organisation to understand and to model the system requirements. The SAM and NAM can be used as a vehicle in the interaction of the participants discussing the requirements of the system.

Credibility of the cost and schedule estimates, priorities, risks, and development process. We can only achieve higher levels of credibility in the priorities and risks if we have a good description of the organisational context. The priorities are defined in the organisational context and many risks are associated to contextual factors. The PAM, SAM and NAM focus on the organisational aspects, not usually captured.

Elaboration: Regarding potential contributions to evaluation criteria for the elaboration phase:

Is the vision of the product stable? To have a stable vision of the product we need a stable vision of the norms that govern the environment and the use of the system. We cannot expect to have clear vision of what the product should be if we do not know well where it will be applied. NAM can be applied in this phase to make explicit the norms that govern the system environment;

Do all stakeholders agree that the current vision can be achieved if the current plan is executed to develop the complete system, in the context of the current architecture? SAM and NAM can make explicit the aspects of the work context by having models that are shared by all stakeholders. Therefore these models can be used as a tool to support the interaction/communication among the stakeholders, during discussion of the current plan, in order to achieve the agreement about it.

Construction: This phase ends with the initial operational capability milestone. Regarding evaluation criteria for the Elaboration phase:

Are all stakeholders ready for the transition into the user community? The PAM, SAM and NAM produce models of the organisational contexts that could be used by the stakeholders to evaluate the transition.

Transition: Regarding evaluation criteria for the elaboration phase:

Is the user satisfied? The answer to this question includes the evaluation of human and social factors that can be better understood within a subjectivist analysis of the organisational context (supported by PAM, SAM and NAM).

3.3 The Impact of PAM, SAM and NAM in the UP Workflows

As mentioned previously, the Unified Process has six "core process workflows" and four "core supporting workflows", which are revisited again and again throughout the life cycle of systems development. In this section we analyse how SAM, PAM and NAM could introduce artefacts, to support the core process workflows.

OS and the Project Management Workflow

The Business-Process Analyst have seven main activities during the Project Management Workflow: Identify Risks, Develop Project Plan, Staff Project, Develop Iteration Plan, Execute Iteration, Evaluate Iteration and Revisit Risk List.

PAM provides information to organize and maintain the management plans. The outcomes from Stakeholder Analysis together with the Evaluation Framing allow us to identify potential risks or problems to be managed during the project time. From Stakeholder Analysis and Collateral Analysis we identify the project staff, users priorities that influence the iterations, project activities not directly related to the software development, as well as the needs for changing management, training, advertisement etc.

OS and the Business Modelling Workflow

The Business-Process Analyst has several main activities in the Business Modelling Workflow:

Regarding to the capture of a Common Vocabulary. We agree with Xie and Liu (2003) who argue that we should not assume the existence of a common vocabulary in the domain; we have to seek for the signs used in the domain. Using the Semiotic Framework (PAM), we begin to elicit and understand the vocabulary and meanings of the context and we can construct, together with the users during the system iterations, the Ontology Charts that make explicit the affordances and relations between elements of the domain (Bonacin, 2004; Simoni and Baranauskas, 2003, 2004a, 2004b). A more appropriated name for this activity would be to Construct a Common Understanding of the Domain..

The Business Designer has four main activities in the Business Modelling Workflow: describe a Business Use Case, find Business Workers and Entities, describe a Business Worker and describe a Business Entity. During the business modelling OS methods allow to describe the norms that act as a force field, delineating the agents' behaviour during the execution of the use cases. The proposed activities for the Business Designer involve norm

analysis. The norms are associated to the affordances of the ontology charts. In addition we could link these norms to the use cases.

SAM describes the things that exist in the business domain. Previous work shows how to construct class diagrams directly from ontology charts (Bonacin et al., 2004) without the necessity of constructing object models of the business domain.

The Business Model Reviewer has two main activities into the Business Modelling Workflow: to Review the Business Use-Case Model and to Review the Business Object Model. Similarly this Worker could review the Ontology Charts and the Norms Specification.

OS and the Requirements Workflow

The System Analyst has six main activities into the Requirements Workflow: Develop the Vision, Elicit Stakeholder Needs, Capture a Common Vocabulary, Find Actors and Use Cases, and Structure the Use-case Model. We have to understand the organization and its needs in order to develop a vision, aligned with the organisational requirements. The process could be started from results of the Evaluation Framing (PAM), and the Ontology Charts can be used to develop the vision document based on the affordances of the agents in a organisational context.

The system analyst together with the users can specify the high-level features of the system using the OS models, which facilitates the communication, in order to fulfil the stakeholder needs. The Use-Case Specifier has one main activity into the Requirements Workflow: to detail a Use Case. An Specifier could not only specify the uses cases but also associate the use cases with the norms specified during the norm analysis. The User-Interface Designer has two main activities into the Requirements Workflow: User-Interface Modelling and User-Interface Prototyping. Prototyping can be done from mapping the Ontology Chart elements to the corresponding screen elements, as we can see in previous work (Simoni and Baranauskas, 2004b). We also verified that the prototype contributed to the validation of Semantic and Norm Analysis, as it represents the "materialization" of the model and refinements represented through it.

The Architect's activity into the Requirements Workflow is to Prioritise Use Cases. The architect could also ensure the integrity not only of the significant use cases, but also the norms that could affect or be affect by architectural decisions.

OS and the Analysis and Design Workflow

The Architects, Designers and Database Designers have to translate the requirements into a specification that describes how to implement the system. There is an approach discussed in previous work to construct the design models for the system and database, using the OS models as a starting point (Bonacin et al, 2004; Liu, 2000).

3.4 Discussion

The work reported here showed that by using OS concepts, through MEASUR methods, we could bring up for discussion a social vision of an organization potentially enhancing the UP. The proposed approach articulates MEASUR methods (PAM, SAM and NAM) in a same process of information system development. The introduction of these new methods does not necessarily leads to a replacement of the disciplines already consolidated in the UP. Some of the main ideas proposed in this paper were applied in the development of Pokayoke (Bonacin, 2004), a CSCW system designed for the context of problem solving in a manufacturing organization. During the development of Pokayoke, class diagrams and behaviour diagrams where constructed from the results of Semantic and Norm Analysis. The Pokayoke system was developed in five prototyping cycles. Ontology charts and norm descriptions were used as a bridge between the system design and organisational issues discussed during meetings with the users. The last version of Pokayoke has 215 classes produced from the semantic diagrams. In general, the application of specific procedures (Bonacin et al, 2004) in the ontology charts produced class diagrams in line with the social context. These class diagrams use the same signs captured in the semantic analysis to specify classes, attributes and operation names. Furthermore the associations, hierarchy and aggregations are derived from social constructions captured by using semantic and norm analysis.

Someone could imagine that by introducing more artefacts, consequently more work would have to be done in the software development, but our experience in practical work, and in results reported in literature as we pointed out before, showed that there is a need for problem conceptualisation. We also observed that Business and System Analysts, more and more, have been involved in this process and need methods and artefacts to deal with these activities. With this initial effort, supported by MEASUR methods, we could minimize risks and reworks, by having a problem solution well suited for the user's context.

4 CONCLUSION

Studies in information system development need to address how people understand the world and how to represent this understanding; social, cultural and

organisational aspects involved in the problem are decisive in this process.

Our motivation integrating OS and UP methods have been to approximate the technical and social orientations present in problems of information system and software design. The OS methods allow a deeper consideration of the semantic, pragmatic and social levels, and an understanding of the social practices for which the system would have to support.

The final results encourage to further work towards a formalization of the approach. The support of case tools with graphical representation, mainly for PAM, is points to be addressed, to increase the integration and control of the use of the tools. Finally, practical work in companies is being continued, allowing us to verify the influence of the approach in the quality not only of the software application, but of the business process as well.

REFERENCES

Bonacin, R.(2004), "A Design Model to Support Co-operation Based on Participatory Design and Organisational Semiotics." Ph.D. Thesis, State University of Campinas, Campinas, Brazil.

Bonacin, R., Baranauskas, M. C. C., and Liu, K. (2004) "From Ontology Charts to Class Diagrams: semantic analysis aiding systems design", 6th ICEIS, Portugal.

Dourish, P. (1995), Developing a Reflective Model of Collaborative Systems. ACM Transactions on HCI, Vol. 2, No. 1, March 1995.

Ehn, P., Lowgren, J., 1997, "Design for Quality-in-use: Human-Computer Interaction Meets Information Systems Development", in M. Helander, T.K. Landauer, P. Prabhu (eds.), Handbook of Human-Computer Interaction, 2nd ed., Elsevier Science.

EUP, 2002, "Enterprise Unified Process" in the web http://www.enterpriseunifiedprocess.info/, accessed in 3/8/2004.

Heusden, B. van e Jorna, R. J., 2000, "Reconsidering the Standard: A semiotic Model of Organization(s)", Groningen University, Groningen, New Zealand.

Jacobson, I. et al, 1996, Object-Oriented Software Engineering: A Use Case Driven Approach. ACM Press, Addison-Wesley Ed.

Jacobson, I., Booch, G. B. e Rumbaugh, J., 1999, "The Unified Software Development Process". Addison-Wesley.

Kruchten, P. (1999) The Rational Unified Process: an Introduction, Addison Wesley.

Larman, C., 2002, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process". Prentice Hall, Inc.

Leland, N. et al., 2002, "Rational Unified Process" in the http://www.cs.tufts.edu/g/180/fall02/public_html/team1/index.html, accessed in 3/8/2004.

Liu, K., Ades, Y e Stamper R. K., 1994, "Simplicity, Uniformity and Quality – the role of Semantic Analysis in system development", Software Quality Management, volume 2, p. 219-235, Computational Mechanics Publications. ISBN: 1-85312-353-6.

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

OSW (1995), "The circulation document". Organizational Semiotics Workshop, Enschede apud Liu, K. (2000), Semiotics in Information Systems Engineering, Cambridge University Press, Cambridge.

Pressman, R. S. (2001) Software Engineering: A Practitioner's Approach, McGraw-Hill, 5th Edition.

Simoni, C. A. C. and Baranauskas, M. C. C. (2003) Launching Organizational Semiotics in the Real World: How to Prepare for it? The 6th International Workshop on Organizational Semiotics. Reading, UK.

Simoni, C. A. C. and Baranauskas, M. C. C. (2004a) Organizational Semiotics Embedded in a System Development Cycle: A Case Study in a Business Organization. 6th ICEIS - International Conference on Enterprise Information Systems. Porto, Portugal.

Simoni, C. A. C. and Baranauskas, M. C. C. (2004b) The Practice of Software Development and the Organizational Semiotics Approach: A Case Study in Business Organizations The 7th International Workshop on Organizational Semiotics. Portugal.

Sommerville, I., 2001, "Software Engineering", International computer science series, Addison-Wesley Publishers Limited - USA, 6th Ed.

Stamper, R, K. (1973), Information in Business and Administrative Systems. John Wiley and Sons, New York apud Liu, K. (2000), Semiotics in Information Systems Engineering, Cambridge University Press.

Stamper, R.K. (1993), Social Norms in requirements analysis – an outline of MEASUR, in Jirotka, M., Goguen, J. and Bickerton, M. (eds.), Requirements Engineering, Technical and Social Aspects. Academic Press, New York.

Stamper, R., Liu K., Hafkamp M. and Ades Y., 2000, "Understanding the roles of signs and norms in organizations – a semiotic approach to information system design", Behaviour & Information Technology, v.19/1, 15-27