USER AUTONOMY IN REQUIREMENTS CHANGING SUPPORTED BY ORGANIZATIONAL SEMIOTICS AND TAILORING

Carlos Alberto Cocozza Simoni, Maria Cecilia Calani Baranauskas
Institute of Computing, State University of Campinas, Av. Albert Einstein 1251, Campinas, Brazil

Rodrigo Bonacin
CenPRA - Rodovia Dom Pedro I, km 143,6, Campinas, Brazil

Keywords: Organizational Semiotics, Tailoring, Software Engineering.

Abstract: Nowadays, organizations face the impact of changes from several sources. Literature points out that we still have a gap between the dynamic of the system maintenance and changes in the organizational processes. To cover this gap we consider the use of practices coming from Organizational Semiotics (OS) and Tailoring. OS allows a deep understanding of the organizational context and the technical system embedded in it, and Tailoring proposes to provide autonomy to users in dealing with changes in computer systems. This paper presents a case study developed in our University to explore and extend an existing approach to provide more autonomy to computer system users, according to the evolution of their business requirements.

1 INTRODUCTION

Although more investment has been done in IT the evolution of the existing computational systems don’t follow the evolution of the organizational requirements (Liu *et al*, 2002). Flexible architecture in computational systems (Truex *et al*. 1999), evolutionary Information Systems (IS) and co-evolution between software and business process (Liu *et al*. 2002) are efforts to cover this gap.

Organizational Semiotics (OS) allows us to understand the role played by the IT in the organizational contexts, and clarifies how an organization interacts with the environment. Tailoring (Morch 1995) proposes autonomy to users, modifying the software according to the continuous changes in business requirements.

In this paper we present a case study conducted in our University to provide more autonomy to end users in changing their applications according to the evolution of their business requirements. The paper is organized as follows: Section 2 presents some key concepts of the OS methods and Tailoring, Section 3 presents the proposed approach, the case study and discusses benefits and drawbacks of the proposed approach, and Section 4 concludes the paper.

2 THEORETICAL BACKGROUND

In this session we present the main concepts involving Organizational Semiotics and Tailoring applied to our work.

Organizational Semiotics (OS)

Social, cultural and organizational aspects involved in the problem must have a more decisive role in the process of developing the information system (Liu 2000). The MEASUR - Methods for Eliciting, Analyzing and Specifying User’s Requirements (Stamper, 1993) is a set of methods, based on OS, to deal with the use of signs, their function in communicating meanings (semantic), intentions (pragmatic), and their social consequences (social). In this work we use the Semantic and Norm Analysis methods.

The Semantic Analysis Method (SAM) assists analysts and users in eliciting and representing their

Alberto Cocozza Simoni C., Cecilia Calani Baranauskas M. and Bonacin R. (2006).
USER AUTONOMY IN REQUIREMENTS CHANGING SUPPORTED BY ORGANIZATIONAL SEMIOTICS AND TAILORING.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 493-496
DOI: 10.5220/0002492004930496

requirements in a formal and precise model. It describes a view of agents with responsibilities in the domain and their behavior patterns called affordances.

The Norm Analysis Method (NAM) focuses on social, cultural and organizational norms that govern the actions of agents. A Norm defines a responsibility of an agent engaged in a task, or condition under which certain actions may (deontic operators) be performed by the agent.

Tailoring

Morch and Mehandjiev (2000) consider that end users of software should be responsible for the process of changing a computer-based tool in its usage context.

Morch (1995) classifies Tailoring in three categories: Customization, Integration, and Extension. The focus of our work is in Extension where the functionality of an application is enhanced by adding new code. This author considers that Extension is a not common feature of commercial applications, and the main question pointed out by him concerns the implementation language to be used in this scenario, which should be accessible to end users.

He also considers that Tailoring techniques could be extended to support organizational changes. We argue that OS may provide methods suitable for doing Tailoring, since it puts equal emphasis on both technical and organizational issues.

3 AN APPROACH PROVIDING AUTONOMY

In this session we present an approach to provide software autonomy to specialists and end users in changing their applications according to their needs.

Based on previous works (Bonacin and Baranauskas 2005; Simoni *et al*. 2005 a/b) we use OS to provide a contextual understanding of the problem, to model the static characteristics of the context (SAM), and its dynamic aspects (NAM).

Based on Tailoring and OS concepts, a norm-based framework called NBIC – Norm-Based Interface Configurator was structured (Bonacin and Baranauskas 2005) to support the work of interface specialists. In the following sessions we present this framework and the case study carried out in our University.

3.1 NBIC – Norm-Based Interface Configurator

By using NBIC architecture, the system maintenance is understood in tree levels (Figure 1):

1. End users (or domain specialists) use tailorable interfaces to change the system behavior;
2. An interface engineer interacts with three tools: (a) Norm Manager to see the results of the tailoring activity at level 1, (b) Action Manager to link norms actions to interface objects and actions, and (c) Tailoring Manager to specify the tailoring interface to be used by the end-users.
3. The code of the system static part could be maintained.

The users' participation is not restricted to a specific level. They should cooperate with the interface engineer in the second level and with the developers in the third level.

The end-user applications interact with the NBIC by two mechanisms: perception and action. The "perception" mechanism is the part of the system that transmits context information to the inference machine using a specific protocol. The "action" mechanism reads the parameters resulted from the inference machine that evaluate the specified norms and triggers the appropriated action.

3.2 Extending NBIC with Norm Modeler: a Case Study

We have structured a course (Special Topics in Information Systems) in our University allowing the students to have contact with the theme of the evolving computational systems. The main goal was to provide an overview about ways to deal with the existing dependencies between the software and organizational changes. The focus of our case study was the first level of NBIC showed in Figure 1, where the end-user can directly change the application.

The class was divided in 9 teams with 4 students in each one. The students' groups worked with the conception and the construction of two prototypes: (1) a business application to be tailored and (2) a norm modeler.

The method allowed the students to model the problem context in terms of static (SAM) and dynamic elements (NAM). All teams succeeded in modeling the context through agents, their behavior and norms that guide them.

Our interest was in investigating what kind of software flexibility we could get and the end-user skill that would be necessary to use this scenario.

Each team chose its own technical platform, allowing us to verify the independence between the business solution and the Norm Modeler.

The norm modeler was used as the tailoring interface box represented in Figure 1. The tailoring interface is a framework that can be configured by the tailoring manager tool, in order to provide end-users ways of specifying norms using direct manipulation. The Norm Modeler was not restricted to this framework; the students could explore any technique to provide an interface where end-users could specify norms. The main focus was on the construction of applications that are easy to maintain by using the tailoring and the OS theoretical referential.

To allow the interaction between the business application and the Norm Modeler the students realize the necessity of a common dialog between them. They proposed a set of interface elements that should be provided and managed, such as: agents, affordances, attributes, rule names, actions, interface objects. To allow non-specialist users in computing to interact in this scenario, the students considered that, for each element, it would be necessary to allow users to assign names, descriptions and graphical representation, registering their own expression of each element.

3.3 The Norm Modeler

The main interface proposed by Team 1, which had achieved the most complete solution for the Norm Modeler, is shown in Figure 2. The features developed are marked with capital letters:

A. To chose a norm name.

B. To specify the conditions to activate the norm.

C. To specify the deontic effect of the norm, permitting/obligating/prohibiting agent actions.

D. To specify the agents involved in the action.

E. To specify the agent actions that should be performed, according to the deontic effect.

F. To show to the end-user the result of the norm specification.

Each feature described above is activated and controlled by through the software that supports the interface. In Figure 2 we can also see an interface to allow the end-user to describe each element involved in the norm specification, to associate some graphical representation, and to write expressions.

3.4 Validation and Discussion

To evaluate the solutions proposed, each team that designed a Norm Modeler tested their solution with one of the norm-based applications designed to support the chosen business problem. We could verify that the business interfaces and the norm modelers designed had a good independence between them, in terms of technology employed and design process.

The majority of the students considered that a norm-based modeling, as provided by Semantic and Norm Analysis from OS, was an important factor to provide this independence and autonomy.

The students with some experience in working with business organizations, had considered that the software designed could be used by specialists in the business domain, with some experience with process, notations etc.

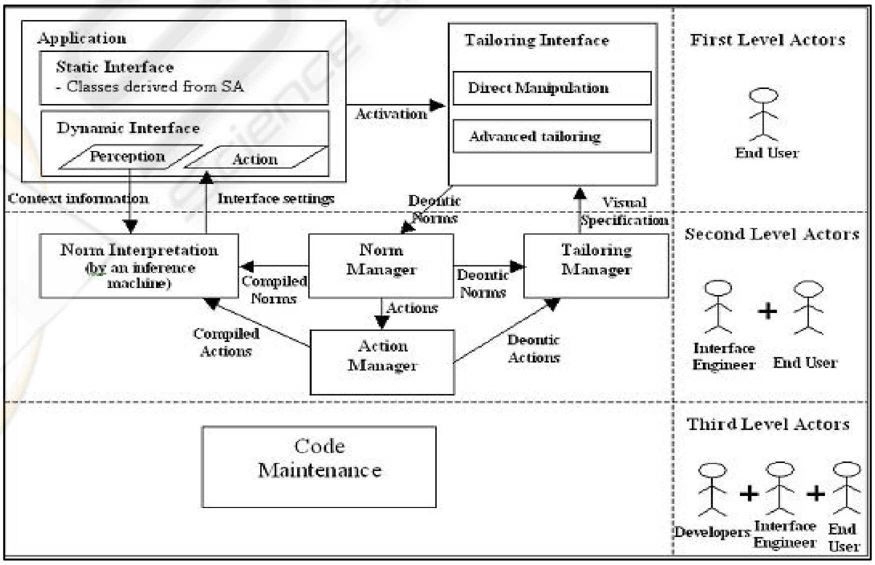
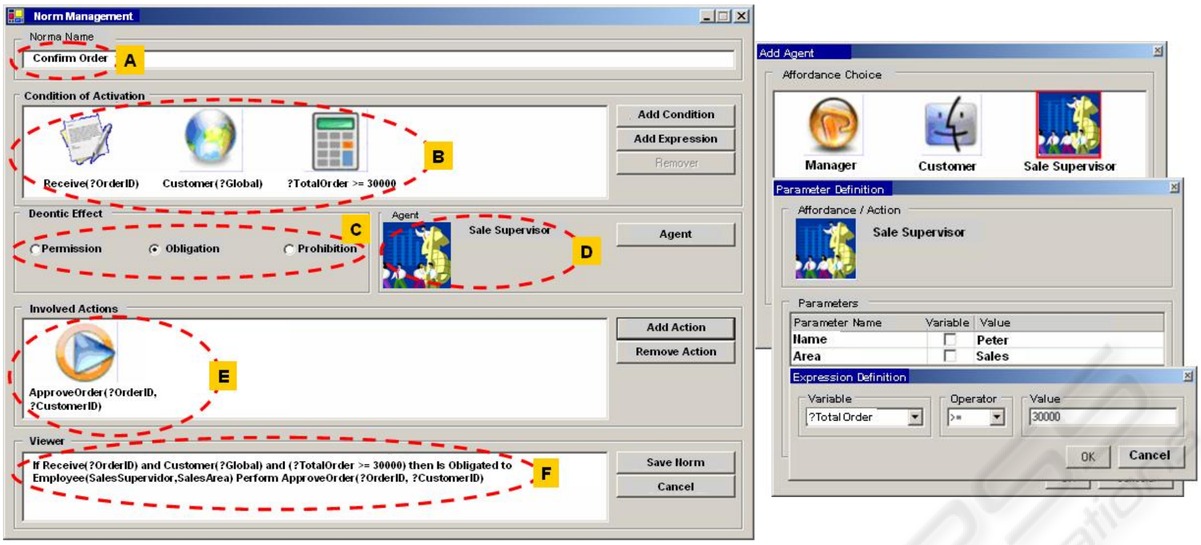


Figure 1: NBIC Architecture Overview.



Norma Name: Confirm Order

Condition of Activation: Receive(?OrderID) Customer(?Global) ?TotalOrder >= 30000

Deontic Effect: ○ Permission ● Obligation ○ Prohibition

Agent: Sale Supervisor

Involved Actions: ApproveOrder(?OrderID, ?CustomerID)

Viewer: If Receive(?OrderID) and Customer(?Global) and (?TotalOrder >= 30000) then Is Obligated to Employee(SalesSupervidor,SalesArea) Perform ApproveOrder(?OrderID, ?CustomerID)

Affordance / Action: Sale Supervisor

Parameter Name	Variable	Value
Name	□	Peter
Area	□	Sales

Variable: ?Total Order Operator: >= Value: 30000

Figure 2: The main Norm Modeler interface from Team 1, and an example of interface to facilitate element description.

4 CONCLUSION

With this work we have verified the possibility to support users in dealing, by themselves, with changes in software resulted from modifications in the business process and their requirements.

Integrating OS and Tailoring concepts and techniques allowed us to think about a development environment where business and technical rules, constraints, and actions could be modeled and described in terms of the semantic employed by the agents involved in the organizational context.

The final results encourage further work towards exploring this approach in practical work, using actual business scenarios, allowing us to verify the desired user autonomy and the influence of the approach in the quality of the software application, and in the business process as well.

REFERENCES

Bonacin, R. and Baranauskas, M. C. C., 2005. An Organizational Semiotics Approach Towards Tailorable Interfaces. *Proceedings of 11th International Conference on Human-Computer Interaction*, 2005, Las Vegas.

Morch, A., 1995. Three Levels of End-User Tailoring: Customization, Integration, and Extension. *Third Decennial Aarthus Conference*, August 14-18 Aarthus, Denmark, p.41-51.

Morch A. and Mehandjiev N., 2000. Tailoring as Collaboration: the Mediating Role of Multiple Representations and Application Units. Kahler, H, Morch, A., Stiemerling, O., Wulf, V. (Eds), *Special Issue on Tailorable Systems and Cooperative Work, CSCW*, Kluwer, v. 9, Issue 1.

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

Liu, K., Sun, L. and Bennett, K., 2002. Co-Design of Business and IT Systems – Introduction by Editors. *Information Systems Frontiers*, 4(3), p. 251-256.

Simoni, C.A.C., Bonacin, R. and Baranauskas, M.C.C., 2005a. Bringing Social Constructs to the Information System Development Process: Contributions of Organizational Semiotics. In *7th ICEIS - International Conference on Enterprise Information Systems*. 25-28 May. Miami, EUA, p.112-119.

Simoni, C.A.C., Melo A. M. and Baranauskas, M.C.C., 2005b. Towards a Social-based Process for Information System Development: A Case Study. In *The 8th International Workshop on Organizational Semiotics*. 23-24 June. Tolouse, França, p.135-149.

Samper, 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*. New York: Academic Press, apud LIU, K., 2000. *Semiotics in Information Systems Engineering*, Cambridge: Cambridge University Press.

Truex, D.P., Baskerville, R. e Klein, H., 1999. Growing Systems in Emergent Organizations: As the new economic realities pressure your organization to change from stable to emergent, new practices for IT support are required. *Communications of the ACM*, August 1999, 42 (8), p.117-123.