

# INFORMATION SYSTEM CUSTOMIZATION

## *Toward Participatory Design and Development of the Interaction Process*

Daniela Fogli

*Dipartimento di Elettronica per l'Automazione, Università di Brescia, Via Branze 38, Brescia, Italy*

Loredana Parasiliti Provenza

*Dipartimento di Informatica e Comunicazione, Università di Milano, Via Comelico 38-40, Milano, Italy*

**Keywords:** Customization, Communication gap, Multi-facet specification, Participatory design, Participatory development.

**Abstract:** This paper proposes the adoption of human-computer interaction methods to address some of the problems related to the customization of information systems, and particularly of enterprise resource planning systems. The paper specifically describes a multi-facet approach to participatory design and development of information systems to build the dialogue between the information system and its users. It encompasses i) a specification framework for representing and translating the different perspectives of the members of the design team, including the end users' perspective, ii) a methodology for collaborative design of the interaction process, and iii) a set of guidelines to carry out the development activities.

## 1 INTRODUCTION

Large scale-software design and implementation projects, such as those involving information systems (IS) and, particularly, enterprise resource planning (ERP) systems, represent a challenge for many companies all over the world.

These systems are aimed at managing and integrating business processes within a complex organization, by adopting existing industry best practices. However, experience studies document several project failures, in which cultural misfits between developers, vendors and consultants, on the one side, and company managers and employees, on the other side, are some of the main reasons for failure (Molla and Loukis, 2005; Soh et al. 2000).

ERP systems are software packages, whose high complexity is due to their cross-module integration, data standardization, adoption of underlying business models and the involvement of several stakeholders. ERP systems are generally composed by a core part and a set of modules that can be customized according to the company's characteristics, needs and target market. However, this customization is often a critical problem because the philosophy adopted is usually 'shaping' the

company and its business processes according to the predefined ERP model, rather than vice versa. Consequently, users are forced by the new system to work and reason in some way different from what they would naturally do, possibly far from their usual ways of working and reasoning.

ERP system customization may include either code modification or ERP parameter configuration, with the aim of adapting for instance workflows, reports, queries, data formats. An important aspect of ERP customization that, in our opinion, is not yet faced adequately by vendors and consultants is concerned with designing and developing a proper dialogue between the ERP system and its users. A suitable dialogue is however fundamental to make the interaction effective and efficient. Whenever the dialogue is not well designed, problems arise: for example, it often happens that users are not aware of the presence of some useful functionalities, because they are not able to find them or even to recognize them among the available options. These problems are often due to a presentation and interaction language alien to users' experience, which makes the system not understandable and the interaction process complicated.

Such a situation leads frequently to failures manifested by implementation delays, cost

increasing, lack of system use. Such failures are normally ascribed to users, whose resistance to, or even rejection of, the software being introduced are rarely well understood by developers and consultants (Wagner and Piccoli, 2007).

In this paper, we advocate the adoption of human-computer interaction (HCI) methods to carry out the customization of the dialogue between users and an ERP system. Among HCI approaches and practices, participatory design appears as the most promising for reducing the failures described.

Participatory design is a design paradigm that invokes direct participation of users as active members of the design team (Schuler and Namioka, 1993) in order to produce interactive systems that are usable and meet users' needs.

We have been directly involved in participatory design experiences of interactive systems that support collaborative work in specific domains – e.g. medical diagnosis and mechanical engineering (Costabile et al., 2007). On the basis of those experiences, we propose here a participatory approach to designing and developing information systems. Particularly, the approach can be adopted in the context of ERP customization to develop, in a participatory way, the dialogue between users and the system, by taking into account users' culture, language and system of signs. To this end, we suggest the creation of participatory design teams constituted by at least three categories of stakeholders, as owners of different expertises: 1) software professionals, including developers, vendors and consultants; 2) domain experts, including representative users, i.e. employees and company managers; and, last but not least, 3) HCI experts, namely professionals in human-computer interaction who should play the role of mediators between software professionals and domain experts, in order to satisfy usability requirements.

However, in such a multidisciplinary team, communication gaps may arise (Folmer et al., 2005), due to the different skills, cultures and languages of team members, and this may seriously compromise the collaboration. Another problem emerges when the participatory approach remains at the high level of system design, without explicitly addressing the problem of system development (Wagner and Piccoli, 2007).

To overcome these problems, our participatory approach is characterized as follows: 1) it is multi-facet, in that it recognizes the existence of the different stakeholders' perspectives and permits their expression according to the stakeholders' culture and language; 2) it supports an effective communication among stakeholders, by permitting

the translation of each stakeholder's perspective into a language comprehensible to the others and by sustaining the iterative design of the interaction process; 3) it advocates a true engagement of users in carrying out the IS customization, by giving them the possibility to participate in creating the interaction experience they would like to live with the system.

The paper is organized as follows: Section 2 presents the motivation underlying this work and discusses related literature; Section 3 describes the proposed participatory approach; Section 4 concludes the paper by delineating a promising research line.

## 2 MOTIVATION AND RELATED WORK

Daneva (2003) describes various degrees of success and failure in ERP requirements engineering. With reference to a well-known commercial product, she analyses the main causes of success and failure. Some of the reasons for failures are interesting to be mentioned here, such as insufficient validation efforts, clashes between process stakeholders, ad-hoc ways of working, underestimation of the importance of certain practices. As one can notice, they are in some way due to inadequate comprehension of user requirements and, generally, to insufficient user involvement.

Soh et al. (2000) wonder, from an Asian perspective, if an ERP can be considered a universal solution. Through a study performed in seven public hospitals in Singapore they arrive at identifying different types of misfits – data misfits, functional misfits, output misfits –, namely gaps between what is offered by the ERP package and organizational requirements. Then, the authors observe that most solutions to these problems “require the users to work around with the alternatives offered by the package” (Soh et al., 2000, p. 50). Other solutions consist in working without validation checks, thus compromising control, or in going back and forth through different screens to look for necessary information, thus compromising productivity. Soh and colleagues argue that the strategy to be followed to cope with these misfits is asking users for assimilating the package functionality in some depth. They say that users “must now consciously ‘get into the ERP software’ to evaluate the appropriateness of the new configured system or the alternatives adopted” (Soh et al., 2000, p. 51). This conclusion assumes that users should become expert

also in information technologies and in the specific package characteristics, besides being expert in their own application domain. Actually, this is a usual attitude of software developers and vendors, who consider users as some people to be 'educated'. A similar 'recipe' is suggested in (Simosko, 2008), who presents tips "on how to prepare and enable your users", by considering user education as a key business process. While, in principle, we can agree on most of these tips, what we question here is the undervaluation of users' competencies, experience and knowledge.

As Rittel (1984) highlights in his "symmetry of ignorance" principle, each person has different knowledge and experience, but none is more important than the others. Therefore, all stakeholders must recognize that each one complements the ignorance of the others, and that it is necessary to reach a mutual understanding by means of a peer collaboration. Thus, Rittel suggests that the knowledge owned by every stakeholder be shared and integrated with other stakeholders' knowledge.

This perspective is in line with the ideas presented in (Molla and Loukis, 2005), where an ERP system is not considered as a mere technical product, but as a socio-technical system including several stakeholders, each one holding a certain cultural assumption towards implementing and using the ERP system. The analysis they performed on real cases suggests that the congruency between the *system culture* – the views of ERP developers, vendors and consultants – and the *host culture* – the views of the organization's project team, managers and employees – may contribute to the ERP success.

As advocated in HCI by participatory design scholars, methods, techniques, and possibly software tools should be defined and developed to bring together different and controversial perspectives, with the aim of avoiding misunderstandings and communication barriers that may compromise the collaboration among the different stakeholders. In (Bodker and Iverson, 2002), the authors discuss how to concretely involve users in participatory design by means of scenarios and prototypes. Scenarios represent an informal way of investigating current practice and triggering ideas for future system usages. Prototypes, besides permitting evaluation of hypotheses, also encourage exploration of design alternatives.

However, even if the adoption of participatory design promises to offer solutions to users' resistance (or even rejection) and system's requirements change, in practice this is far from an automatic occurrence. In the study described in (Wagner and Piccoli, 2007), it was observed that

"users tend not to fully engage until the system's impact on their working life is apparent – generally when the system 'goes live'" (p. 52). According to the authors, to make participation more powerful, users should be involved on topics really salient for them. This suggests to consider participatory design and participatory development as two interrelated activities (Pekkola et al., 2006). Pekkola and colleagues propose a multi-methodological approach that bridges the gap between participatory design and information systems development (ISD) methodologies, where methods and techniques for system design and system development are used in an interleaved way. Even though this represents an interesting proposal, a lot remains to be done for developing "a formalized end-user-oriented ISD method that is useful not only for the system designers but also for every party involved in the ISD process in each of its phases" (p. 28). The present paper proposes an approach that goes in this direction.

### 3 THE PROPOSED PARTICIPATORY APPROACH

We present here the multi-facet participatory approach to system design and development. The approach derives from our direct experiences in participatory design of domain-specific interactive systems (Costabile et al., 2007; Costabile et al., 2008). It aims at giving value to the stakeholders' different perspectives and, at the same time, at supporting their convergence toward a shared understanding of the interaction with an information system. Through proper procedures, guidelines and techniques, the approach has the goal of fostering true engagement of team members and supporting communication among them during all system lifecycle phases.

In the context of IS customization, the participatory design activity has the purpose of allowing different stakeholders to achieve a common specification of user-system dialogue in terms of the functionalities to be made available to users, the interaction metaphor on which the system should be based, and the interaction style it should support. Whilst, the participatory development activity aims at building such a dialogue in order to obtain an information system that is understandable, usable, reliable and accepted by its users. A key issue in both system design and development is to permit each stakeholder to describe the interaction process between user and system from her/his point of view and to support an efficient and effective communication among the different stakeholders.

Therefore, the approach foresees (Figure 1):

- *A specification framework*, which provides languages to support the expression of one's own view of the interaction experience with the system and the communication among the team members in all the stages of the design and development processes;
- *A design methodology*, which establishes how the different stakeholders in the multidisciplinary team should work together and collaborate in designing the interaction process between user and system, by specifying it according to their own perspectives;
- *Development guidelines*, i.e. a set of recommended practices to be followed when developing, in a participatory way, the designed interaction process.

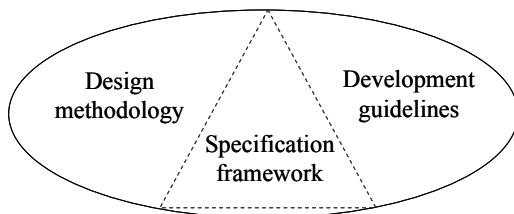


Figure 1: Our participatory approach.

For sake of simplicity, but without losing generality, in the following we restrict our reasoning to the description of the collaboration among three categories of stakeholders: software professionals, domain experts, and HCI experts. It is worth noticing that, especially in the case of ISs, one can identify different stakeholders among software professionals, as well as among domain experts. The approach is scalable to situations where the various stakeholders are identified more precisely. Another important consideration about the team composition is the presence of HCI experts, who play the role of mediators between domain experts and software professionals. In our participatory design experiences, as HCI experts, we empirically adopted a framework of languages and translation procedures in order to bridge the communication gap between domain experts and software engineers. We actually operated translations from the domain experts' view to the software engineering experts' view and vice versa. The approach presented in the following aims at formalizing and generalizing these experiences, which can be effectively applied to the case of IS customization.

### 3.1 Specification Framework

In our participatory design experiences (Costabile et

al., 2007; Costabile et al., 2008), we have observed that each stakeholder looks at and describes the interaction experience with a software system from her/his own perspective and according to her/his goals. Based on these observations, we have formalized a specification framework constituted by different *interaction languages* – one for each different stakeholder – whose sentences provide an explicit description of the interaction process between the user and the system according to the different perspectives (Fogli et al., 2007). By restricting to the case of three kinds of stakeholders – domain experts, HCI experts, and software professionals –, we have introduced three interaction languages to describe the different perspectives. The first language, the *interaction trace language* (ITL) describes the interaction experience from users' point of view. It is expressed as finite sequences of images on the screen coupled with the names of operations users perform on them. The second language, the *direct manipulation language* (DML), describes the interaction experience from the HCI point of view, by adding state-based descriptions to the elements of the previous language. Finally, the third language, the *finite state machine-interaction language* (FSM-IL), describes the interaction experience from the software professionals' point of view, by enriching the previous language with computational-oriented information. ITL, DML and FSM-IL share some core elements, which can be used to define translation procedures for mapping each language into the others. They are called *animated pictorial elements* (APEs): the pictorial parts of the system each stakeholder can see on the screen and reasons about during interaction, together with the names of operations that can be performed on the system. Figure 2 gives an overview of the specification framework.

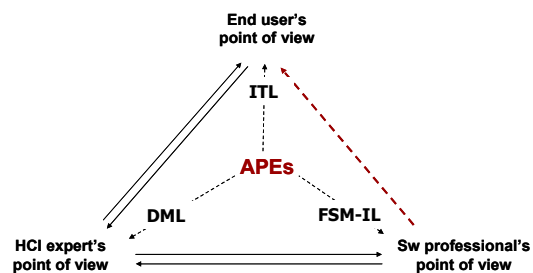


Figure 2: The specification framework.

### 3.2 Design Methodology

The design methodology is a procedure that establishes how the members of a multidisciplinary team work together to design the interaction with the information system, by providing different but

related specifications. The methodology is multi-facet since it aims at building, through the framework, three different descriptions of the interaction process. The procedure consists of the following steps to be performed several times in an order dictated by the results of each step:

1. Domain and HCI experts work together by discussing scenarios that describe current work practices and possible future system usages with the purpose of providing a description of the interaction process from users' point of view. Specifically, they firstly identify the alphabet of APEs, on which the interaction process will be based, according to users notation and system of signs. Then, they collaborate each other in defining the ITL.

2. HCI experts analyse the ITL obtained in the previous step from the usability point of view, by taking into account intention formulation, action planning and user perception and interpretation of system feedback; then, they translate the user's view – ITL – into a state-based specification of the system – DML – to be submitted to software professionals.

3. Software professionals analyse such state-based specification and produce a computational-oriented specification of the system – FSM-IL. In this phase, software professionals could notice problems in the state-based specification that may ask for a revision; this requires a new collaboration phase with HCI experts, and thus to go back to step 2. Whenever this revision affects the user's description of the interaction process, HCI experts need to collaborate again with domain experts to decide the necessary modifications, thus going back to step 1.

4. The computational-oriented specification is then used to generate a prototype, which is given to domain experts in order to be tested.

5. Domain experts can thus analyse the prototype physical appearance and behaviour, possibly noticing interaction problems due to misunderstandings, incompleteness, lacking in consistency, etc.

6. To solve the above problems, domain experts should ask for collaborating with HCI experts to revise together the ITL. This situation will also occur at use time whenever users, as well as their work environments, organization procedures and adopted technologies, evolve. In both cases, it is necessary to go back to step 1.

Note that, as suggested by step 6, the design process is active throughout the IS lifecycle.

### 3.3 Development Guidelines

During participatory design activities, it may happen that the specific users' needs are not clearly understood and that attention is not paid to users' ideas. Very often, users are not even able to express their point of view because design problems are presented them by using terms alien to their experience and culture. Consequently, users are not able to understand soon the impact of the new system in their life and work, and tend to perceive participatory design just as an overhead to their work activities (Wagner and Piccoli, 2007).

To support an effective user engagement during IS customization, our participatory approach includes also the following development guidelines:

- each member of the team should be able to analyse the interaction experience according to her/his own reasoning strategies using her/his own notations and system of signs;
- each team member should be able to develop the interaction experience from her/his own perspective;
- all team members should be able to communicate about and collaborate in designing and developing the interaction experience with the system.

These guidelines suggest that each member of the team should be provided with a development environment permitting the description of one's own perspective on the interaction with the system under customization. Such environment should facilitate its users in defining the corresponding interaction language formalized in the framework. The different perspectives should be exchanged among the stakeholders and properly interpreted by the development environment used by each stakeholder.

In the past, we have proposed the software shaping workshop (SSW) methodology to support participatory design and development of domain-specific interactive systems. Particularly, the methodology has been applied in the mechanical engineering and medical diagnosis domains (Costabile et al., 2007). SSWs are software environments conceived according to the development guidelines delineated in this paper: they are customized to stakeholder's culture and knowledge, they provide all and only the tools to be used to perform the stakeholder's activities, and are organized in a network to support communication among stakeholders. We argue that similar tools might be developed to make concrete our participatory approach to IS customization.

## 4 CONCLUSIONS

This paper asks for a greater awareness by IS developers and consultants of the Rittel's symmetry of ignorance principle. They should regard users as the 'owners' of the problem, thus recognizing that users' domain knowledge complements their ignorance (and not only vice versa, as software professionals are used to think).

The multi-facet approach here proposed recognizes this symmetry by allowing a multidisciplinary team to take into account all the different perspectives in designing and developing the dialogue between an information system and its users. More specifically, the approach provides a methodology for participatory design and a set of development guidelines, which recognize and give value to the different expertises of the team members while fostering their collaboration. The approach also provides a specification framework that supports team members in both design and development by allowing them to represent and translate their different perspectives.

The languages in the framework are formal tools that may facilitate system prototyping, since they can be used for creating proper software environments that support system specification and prototype generation from the specification. The generated prototypes can then be directly tested by the users, who can thus ask immediately for system refinement.

As a future work, we plan to apply the approach to a real ERP context, in order to evaluate its effectiveness in the case of information system customization. In general, our objective is to persuade IS vendors and developers of the importance of providing their clients and consultants with software environments for participatory design and development of the dialogue between an information system and its users.

## ACKNOWLEDGEMENTS

The authors wish to thank Piero Mussio for his contribution to the approach presented in this paper. This work capitalizes on participatory design experiences carried out in collaboration also with Maria Francesca Costabile and Antonio Piccinno, who are herewith acknowledged. The authors are also grateful to Sonia Di Labio for the interesting discussions about the content of the paper.

## REFERENCES

- Bødker, S., Iversen, O. S., Staging a Professional Participatory Design Practice – Moving PD beyond the Initial Fascination of User Involvement, *Proc. NordCHI 2002*, October 2002, Aarhus, Denmark, 11-18.
- Costabile, M. F., Fogli, D., Lanzilotti, R., Mussio, P., Parasiliti Provenza, L., Piccinno, A. Advancing End User Development Through Meta-Design, in S. Clarke (Ed.), *End User Computing Challenges and Technologies: Emerging Tools and Applications*, Hershey, PA, IGI Global, 2008, 143-167.
- Costabile, M. F., Fogli, D., Mussio, P., Piccinno, A., Visual Interactive Systems for End-User Development: a Model-based Design Methodology, *IEEE Trans. on Systems Man and Cybernetics* 37(6), 2007, 1029–1046.
- Daneva, M., Six Degrees of Success or Failure in ERP Requirements Engineering: Experiences with the ASAP Process. *Int. Workshop on COTS and Product Software: Why Requirements are so Important* (11th IEEE Int. Conf. Requirements), Monterey Bay, 2003.
- Fogli, D., Marcante, A., Mussio, P., Parasiliti Provenza, L., Piccinno, A., Multi-facet Design of Interactive Systems through Visual Languages. In F. Ferri (Ed.), *Visual Languages for Interactive Computing: Definitions and Formalizations*, Hershey, PA, IGI Global, 2007, 174-204.
- Folmer, E., van Welie, M., Bosch, J. Bridging patterns: An approach to bridge gaps between SE and HCI, *Journal of Information and Software Technology*, 48(2), 2005, 69–89.
- Molla, A., Loukis, I., Success and Failure of ERP Technology Transfer: A Framework for Analysing Congruence of Host and System Cultures, *Working Paper Series*, University of Manchester, 2005.
- Pekkola, S., Kaarilahti, N., Pohjola, P., Towards Formalised End-User Participation in Information Systems Development Process: Bridging the Gap between Participatory Design and ISD Methodologies, *Proc. PDC 2006*, 2006, Trento, Italy, 21-30.
- Rittel, H., Second-Generation Design Methods. In Cross, N. (Ed.), *Developments in Design Methodology*, John Wiley and Sons, New York, NY, 1984, pp. 317-327.
- Schuler, D., Namioka, A. (eds.), *Participatory Design - Principles and Practices*, Hillsday, N.J: Lawrence Erlbaum Associates, 1993.
- Simosko, N., The IT Transformation Process – 5 Tips on How to Prepare and Enable Your Users, *SAP Insider*, Apr-May-Jun 2008, www.SAPInsideronline.com.
- Soh, C., Kien, S. S., Tay-Yap, J., Cultural Fits and Misfits: Is ERP A Universal Solution?, *CACM* 43(4), 2000, 47-51.
- Wagner, E. L., Piccoli, G., Moving Beyond User Participation to Achieve Successful IS Design, *CACM* 50(12), 2007, 51-55.