10-2008

Socio-instrumental Design Patterns

Jonas Sjöström
Uppsala University, Sweden, jonas.sjostrom@dis.uu.se

Pär J. Ågerfalk,
Uppsala University, Sweden, par.agerfalk@dis.uu.se

Socio-instrumental Design Patterns

Jonas Sjöström, Uppsala University, Sweden (jonas.sjostrom@dis.uu.se)

Pär J. Ågerfalk, Uppsala University, Sweden (par.agerfalk@dis.uu.se)

Abstract: This paper draws on pragmatic philosophy and design theory to propose the concept of the socio-instrumental design pattern. These patterns are design theories that acknowledge that solutions are to be found in the source social setting (where design takes place) and the target social setting (where the IT artefact will be embedded and put into action). Solutions may be discussed in terms of what to do and what to focus in the design process, not merely in terms of properties of the IT artefact. Besides introducing the concept, the paper proposes a way to structure socio-instrumental design patterns, and an example pattern is included to illustrate the approach. The example is grounded in empirical data from an ongoing action research project with an emphasis on socio-instrumental qualities of IT artefacts in their social context.

Introduction

Design patterns, first introduced by Alexander et al. (1977), were brought to computer science by Gamma et al. (1995). Their seminal work has been heavily cited and a multitude of design patterns have emerged since its publication. These patterns typically show generic object-oriented solutions to common software design problems. A pattern is a generic solution to a problem class. By applying a design pattern to a problem, a solution adapted to the problem may be rapidly developed. A pattern is thus a way of re-using knowledge of how to solve a certain problem type. Each pattern is named, and typically consists of a problem description, a generic solution, and advice as to when the pattern could be applied, along with potential drawbacks. The success of design patterns within computer science has led to an interest in the concept within other related disciplines. Human-computer interaction (HCI) scholars, for example, have adopted the concept and adapted it into interaction design patterns, i.e. patterns for user interface design rather than internal system design. Several scholars within human-computer interaction have put forward ideas on how to structure such patterns (e.g. Tidwell, 2005; Borchers, 2001; Graham, 2003; van Welie & van der Veer, 2003), and several patterns have been presented in HCI research (e.g. van Welie, 2008; Tidwell, 2005).

Design patterns addresses the need for flexibility in design processes in that they represent not complete process guidance but support for addressing specific generic problems. The idea is that they can be used and combined so as to provide method support if and when needed. In that sense, they are similar to other method modularization constructs, such as method components (Karlsson & Wistrand, 2006), method fragments (Brinkkemper, 1996) and method chunks (Rolland & Prakash, 1996). However, it is important to understand that design patterns are not meant to replace such constructs but to complement them with specific guidelines related to specific design situations.

Essentially, design patterns are design theories. March & Smith (2003, p. 254) state that "rather than posing theories, design scientists strive to create models, methods, and implementations that are innovative and valuable". Interaction design patterns may consist of models (relations between problem classes and solutions), methods (how-to-instructions to create specific solutions based on a generic pattern), and instantiations (e.g. user interface exemplars that represent solutions to problem classes). Design patterns as well as design theory are coherent with the philosophy of American pragmatism, in the sense that they are based on the principle that an application of concept – rather than its origin – is the criterion of its value (James, 1907; Goles & Hirschheim, 2000).

Interestingly, while computer science design patterns and interaction design patterns are useful for designing internal and external properties of IT artefacts, they tend not to consider the social

environment of the artefact. First, the solutions are biased towards the IT artefact as a design product, rather than the social process of designing the artefact. Hevner et al (2004) recognize the concept of design both as a noun (the IT artefact as a product of design) and a verb (the design process). We conceive of this as the source social setting which brings the IT artefact into existence. Second, there is a target social setting: The one in which the IT artefact is embedded and which it is expected to support. This is problematic, since it is reasonable to assume that the final qualities of the IT artefact are affected by, and affect, these social settings. Several IS scholars (e.g. Orlikowski & Iacono, 2001; Goldkuhl & Lyytinen, 1982; Ågerfalk, 2003) propose a view of IT artefacts as embedded in a social setting. Checkland's (1999) soft systems methodology, for example, thoroughly points out the difference between design of technical (hard) systems and the induction of change in human (soft) systems. The idea of problem solving as a social process is also in harmony with a predominant goal within interaction design research: Building IT artefacts of high usability or quality-in-use (Bevan et al., 1991; Bevan, 2001; ISO 9241-11, Preece et al., 2003). This is typically achieved through a user-centered design process and through the application of techniques such as mock-ups, scenarios and prototypes to promote learning and communication between users, designers, and other stake-holders (Preece et al., 2003). Research on participatory design (e.g. Ehn, 1995) and socio-technical design (Mumford, 1995; Bansler, 1989) emphasize the need for an understanding of change work in the social setting accompanying the development of IT. All in all, much IS research support the view of information system design as a social process, and the view that the IT artefact needs to be understood as embedded in a social context.

Moving back to the pragmatic characteristics of design theory, Dewey (1938) views inquiry as an endeavour to change an indeterminate situation into a determinate one. This may be translated into a problem-solving context. Inquiry is about understanding the situation-in-view (i.e. problems), and finding solutions, i.e. changing the situation to accomplish some goal or meet some ideal. Dewey's (ibid.) definition of a situation is that it is a contextual whole consisting of objects and events. The IT artefact may be considered an object, and the use situation is characterized by a number of events (or actions) where the IT artefact is an instrument for action. Interaction design patterns are partially coherent with the pragmatic idea that a concept should be put into action and evaluated based on its usefulness in action. However, the notion of a contextual whole consisting of objects and events, which is an important aspect of Dewey's pragmatism, is typically not incorporated in solutions in design patterns: Such patterns focus objects isolated from events.

This paper draws on pragmatic philosophy and design theoretical insights in the IS field, and proposes the concept of the socio-instrumental design pattern. Such patterns are design theories, based on the practically oriented notion of design patterns, but also an extension of these, as they acknowledge that solutions are to be found in the source social setting (where design takes place); and the target social setting (where the IT artefact will be embedded and put into action). Solutions may be discussed in terms of what to do – and what to focus – in the design process, not merely in terms of properties of the IT artefact. This may be clarified through an example: On the one hand, Nielsen's (1993) consistency principle prescribes that terms and graphics should be consistent in their meaning between different parts of the user interface, which is clearly a prescription regarding the design product. On the other hand, when it comes to Nielsen's (ibid.) task match heuristic, the design process needs to include an activity to learn about – and sometimes critically assess and re-design – the users' tasks. Furthermore, the solution to a problem class may reside in a change in organizational activity, rather than in the characteristics of the artefact as such.

Besides introducing the concept, the paper proposes a way to structure socio-instrumental design patterns, and an example pattern is included to illustrate the approach. The example is grounded in empirical data from an ongoing action research project with an emphasis on socio-instrumental qualities of IT artefacts in their social context. The main message of the paper is then summarized through a discussion about the validity of socio-instrumental design patterns as design theories, along with implications for research and practice.

A socio-instrumental view of the IT artefact

In order to understand the concept of the socio-instrumental design pattern, we must first establish an understanding of the underlying socio-instrumental perspective on IT artefacts. Figure 1 shows relations between important concepts connected to design and evaluation of IT artefacts. Although the terminology in the figure is focused on design, one may easily adopt the model to an evaluation context. Or, more generically, we could use the concept of inquiry in a Deweyan sense, which encompasses both design and evaluation. The figure is represents the merging two existing models: Orlikowski's (1992) structural model of technology and Ågerfalk's (2003) A3 model (explaining the relations between action, actor and artefact). On top of synthesizing these two models, the concepts of design practice, design ideal, and use quality have been added.

The A3 model, which is part of Information Systems Actability Theory (e.g. Ågerfalk, 2003; Goldkuhl & Ågerfalk, 2005; Sjöström & Goldkuhl, 2004), states that an actor performs social action using an IT artefact, which thus is an instrument in the performance of action. As an elaboration on the original model, the relations between action, actor, and artefact have been named, the social setting has been related to them, and the encapsulation of these four concepts has been named "IT artefact in context" in Figure 1 .

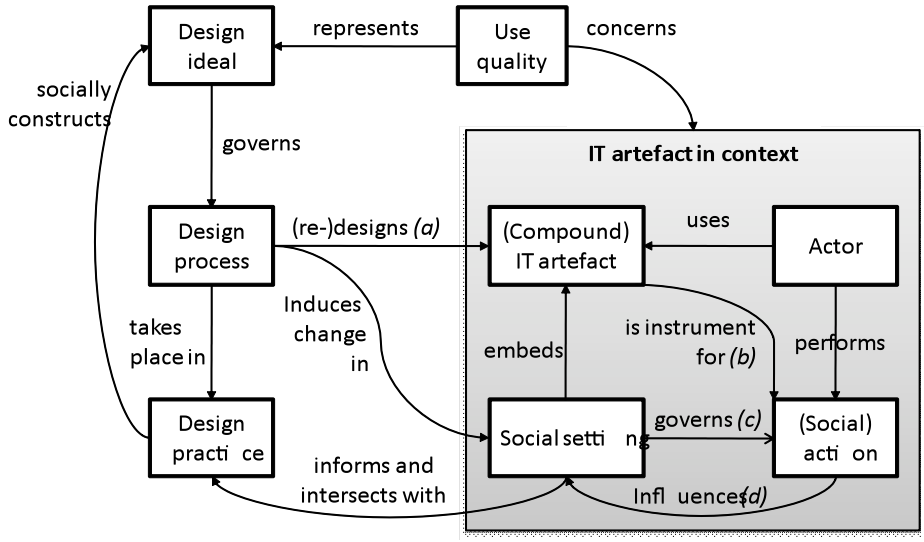

Figure 1 A socio-instrumental contextualization of IT design (Sjöström, 2008)

The integration of Orlikowski's (1992) structural model of technology is transparent in the model through the letters (a) through (d). Orlikowski states that we need to understand technology as a product of human action (a), which is shown above as the IT artefact as a result of a design process, taking place in a design practice. In Orlikowski's (1992, p. 410) words, "Technology is an outcome of such human action as design, development, appropriation and modification." Further, Orlikowski's (ibid) view is that technology is a medium for human action (b): "Technology facilitates and constrains human action through the provision of interpretive schemes, facilities, and norms." IT artefacts are designed in a certain way and their properties are institutionalizations constraining and enabling certain actions in the particular social setting. Orlikowski (ibid., p. 410) further explain the institutional conditions for interaction with technology (c): "Institutional properties influence humans in their interaction with technology, for example, intentions, professional norms, state of the art in materials and knowledge, design standards, and available resources (time, money, skills)." Finally, Orlikowski (ibid., p. 410) explains the institutional consequences of interaction with technology (d), still based on Gidden's reciprocal relationship between agency and structure, in the following way: "Interaction with technology influences the institutional properties of an organization, through reinforcing or transforming structures of signification, domination, and legitimation." Orlikowski's model is based on Giddens' (1984) theory of structuration, which treats these relationships between

action and structure on a more abstract level, where action in general is influenced by – and reinforces or transforms – social structures. One example of the validity of Giddens' more general view is that the users' perception of an IT artefact may very well be cultivated during the coffee breaks, not only in situations where actors use the IT artefact as an instrument to perform action.

Figure 0-1 also pictures the IT artefact as embedded in a social setting. The notion of a design practice, which informs a design process and intersects with it in a user-centered design manner, has been integrated into the model. A design process is governed by design ideals, which we conceive of as socially constructed by the actors in the design practice. Those ideals may be sprung from various sources, such as experts/consultants proposing a certain quality framework or best practice approach, or representatives from the workpractice who proposes business goals or business values as a form of design ideals. Finally, a use quality may be understood as a representation of a design ideal. Design ideals concern the IT artefact in context, as depicted in Figure 1. While conceiving use qualities as representations of such ideals, these too focus the IT artefact in context. A consequence of adopting this perspective is that one should be careful not to adopt a narrow view of use qualities in a design or evaluation situation – the ideals that govern our actions need to acknowledge the full context of the IT artefact in use, in action, and within a social setting.

A design process induces change in the social setting in which the IT artefact will be put into use. Sometimes, designers have limited opportunity to induce such change in the social setting (e.g. when designing off-the-shelf products, or when the development effort is an "IT project", i.e. the development of technology without mandate to make changes to the social or organizational realm). However, in some situations, the designers also have access to proposing changes to business processes or attempt to influence the norms and roles in the social setting where the IT artefact will be embedded. People's experience of the quality of the IT artefact will be formed as a consequence of both the properties of the IT artefact and by other experiences. For instance, a user-centered process will affect both the artefact as such, but also the users' willingness to accept the changes at the workplace. Furthermore, apart from the introduction of a new IT artefact, there may be other organizational changes accompanying the introduction of a new artefact, such as new roles, workgroups and assignments. We need to conceive of the use qualities of an IT artefact in the light of such changes to the organization. If an individual states that an IT artefact is useless, it may indicate that the artefact as such should be re-designed, but it may also indicate that the particular individual (or the group which he/she represents through a role) has not been adequately introduced to the artefact. It may also indicate that the person needs more knowledge of the social context.

A Template for Socio-Instrumental Design Patterns

This section proposes a template for describing design patterns, comprising of a number of description categories. The template has been derived from various theoretical sources on design patterns and design theory, as discussed below.

The *name* (i) of a design pattern is needed in order to refer to the pattern, and efficiently communicate its meaning and content to others.

A pattern's *problem class* (ii) informs the reader of the type of problem the pattern provides a solution to. It is thus a form of contextualization of the pattern, and a delineation and definition of its usefulness. Problems may be conceived of as deviations from an ideal situation. Thus, in order to properly understand and represent a problem we may need to be explicit about this ideal.

The perspective on the IT artefact in relation to source and target social settings (as discussed above) is the basis for the concept of socio-instrumental design patterns: They differ from other types of design patterns in that they make a clear distinction between three different *solution domains* (iii): The characteristics of the IT artefact as such, the source social setting (i.e. the design process), and the target social setting (i.e. the social context in which the IT artefact will be embedded and put into action).

Gamma et al. (1995) suggest that a design pattern description should include a discussion of *forces*

(iv) and *counter-forces* (v). This is to inform the reader about conditions for successful application of the pattern and obstacles that may mitigate against its application, in some situation. Forces and counter-forces may be understood as social phenomena, as will be shown in the instantiation below.

Design patterns do not exist in a theoretical void, although their presentation often seems to suggest this. Therefore, design theory concepts, such as Gregor & Jones' (2007) *kernel theories* (vi) or micro theories are important in the theoretical grounding of patterns.

Finally, a pragmatic ideal in design theory (e.g. March & Smith, 1995) is to provide instantiation(s) (vii) of the theory in order to provide evidence to the theory, but also better to communicate to others how the theory may be applied. This is related to design patterns, which typically provide exemplar solutions as part of a pattern description. In design patterns, such exemplars may be represented by an object-oriented design model (Gamma et al., 1995). An interaction design pattern typically includes design elements that constitute a solution to the problem class (e.g. Tidwell, 2005; Borchers, 2001). Socio-instrumental design patterns aim at providing solutions related to the IT artefact and the source and target social settings. Therefore, instantiations need to be represented in a less formal way, using some appropriate notation to illustrate a specific solution to the problem. Given the potential complexity of socio-instrumental solution, it is not possible to generalize how such a representation should be structured or described, but it could consist of scenarios, user interface design, and process models, to mention but a few means of representation. The point being that these should simplify the use of the pattern by providing an example.

To summarize, seven categories are used to describe socio-instrumental design patterns: *Name*, *problem class*, *solution domains*, *forces*, *counterforces*, *kernel theories*, and *instantiations*. The solution domain consists of the three sub-categories *target social setting*, *instrumental*, and *source social setting*.

The Social Transparency for Accountability (SoTA) Pattern

This section provides an example pattern that is empirically grounded through two instantiations: One based on the action research project presented in section X, and one based on statements in media and government inquiry reports. The second instantiation is relevant in order to provide evidence of the proposed concepts as a recurrent pattern. Below follows the pattern description, based on the categories shown in the following section.

Name: Social Transparency for Accountability (SoTA)

Problem class: Commonly, there is a demand for accountability in business processes. One contemporary norm is to sign papers or to validate content using some e-id technology. Such mechanisms may create bottlenecks in the workflow, i.e. the process halts while awaiting validation of content from a number of participants.

Solution (target social setting): Signatures, whether technology-supported or not, are a means to achieve accountability. By adopting e-id technology, one further institutionalizes old norms by incorporating those into the technological infrastructure. Norms, such as the requirement of signatures, need to be critically assessed in order to come up with new ways of working which harnesses the potential of IT. We propose the combination of the concepts of social transparency and the opportunity and responsibility to object to erroneous representations as a complementary way to achieve accountability in business processes (a more thorough discussion about signatures v. social transparency is presented by Sjöström & Goldkuhl, in press). Removing the requirement for signatures is a shift in perspective: We assume that representations are correct, and we object to errors, rather than assuming that representations are incorrect, and need to be audited and confirmed through a signature.

Solution (instrumental): IT still plays an important role when the requirement of signatures is abandoned – however, its primary purpose is to clearly share representations between actors, and allow the different parties to object to erroneous representations. The IT artefact should represent the current state of affairs, and actions performed in the social setting; allowing the user to understand the

situation (a Deweyan contextual whole, consisting of representations of objects, and the events which lead to them). The IT artefact should also afford the possibility to comment upon and pose objections to representations of previous actions. The detailed properties of the IT artefact need to be conceived of as a result of the source social setting (see above). It should be noted that the IT artefact should be conceived in a broad sense: It might be a web based IT system, but it might also be printed forms and handwritten comments and objections.

Solution (source social setting): There is a need to investigate business processes, and centre the design of IT around the users to determine the meaning of "social transparency" in the current situation. The design process is also vital to identify stakeholders in the target social setting, which are important in the process of assuring accountability. The social transparency and the actions of commenting and objecting to the "state of affairs" should be open for all such stakeholders.

Forces: In order to come up with innovative solutions, unleashing the potential of technology to do things in a new and "better" way, there is a need to question established norms. Thus, innovation and is an important driving force for this type of change.

Counterforces: This design pattern requires changes in norms, which is typically a complex endeavor. It requires soft systems thinking to induce such changes in an organization (Checkland, 1999) and a broad awareness of the problems among stakeholders (Kotter, 1996).

Kernel theories: The concept of social transparency in the IT artefact has been thoroughly investigated in research on information systems actability (e.g. Ågerfalk, 2003), which includes a number of design ideals which are relevant to achieving social transparency. Apart from a representation of objects in the world, there is a need to represent social relationships that are formed when action is performed, e.g. that I am obliged to do something or permitted to do something. Further, the events that lead to representation of material objects and social relationships need to be properly represented. This can be supported in many ways in the IT artefact design, by providing an action memory. Previous and anticipated future actions may be represented. Human action is complex, and may be represented in many ways, e.g. its intentionality (Is this statement only a comment, or is it an objection to some representation of the world?), visible actors (who said this, and who am I talking to if I press this button?), and so forth. Going back to Dewey's (1938) definition of a situation, there is a need to understand a contextual whole as a number of related objects and events. An orderly understanding of objects and the events leading to the state of those objects is needed for an actor to make sense of the situation. From this perspective, an important characteristic of an IT artefact is to represent the "state of affairs" by properly representing the objects and events in order to allow for the actor to understand the situation as a "contextual whole".

Instantiation 1: In Sweden, the *Act concerning Support and Service for Persons with Certain Functional Impairments (LSS)* regulates services provided by the municipality to the individuals. The intention of the law is to enable persons with functional impairments full participation in everyday life. Among other things, the legislation regulates the services of personal assistants to the functionally impaired. Over time, the use and administration of such assistance have increased drastically, and the administration costs have become very problematic for the municipalities. A thorough analysis of the administrative processes was conducted as part of an action research project during 2007 and 2008. One of the most significant problems was related to the financing of personal assistance: It is partially paid by the Swedish Social Insurance Agency, and partially paid by the municipality where the citizen in need of service lives. The Social Insurance Agency requires accountable time reports from the municipalities to transfer funds. The typical means to achieve accountability in the Swedish public sector is to appoint someone accountable for the information, who is authorized to sign documents. In the LSS case, the citizen in need is the formal addressee of financial support from the government, and is thus the one responsible for auditing and signing "bills". Sjöström & Goldkuhl (in press) report a number of problems related to signatures in the LSS context, e.g. that the citizens or their representatives aren't always able to assess the truthfulness of the assistants' time reports (e.g. signing to get their funds, without really understanding the reports and the obligations involved), and that an immense amount of paperwork and administrative processes have been established to deal with the signatures. The project rendered a number of suggestions for alternative ways of achieving

accountability. There is an agreement among stakeholders that a socially transparent process, where all actors can see what others have said, and comment and complaint about others' statements, is a fully adequate substitute to the signatures. An IT artefact, in this context, may facilitate such transparent communication between the stakeholders, and also support the different actors' activities through its design, thus increase accountability. An example from a design prototype is shown in figure 2, where a municipal work manager is asked to confirm a time report that deviates from the plan. In order to do this, related previous actions connected to the issue at hand are shown, and the work manager is provided with the opportunity to further investigate the situation by contacting all actors involved in the process.

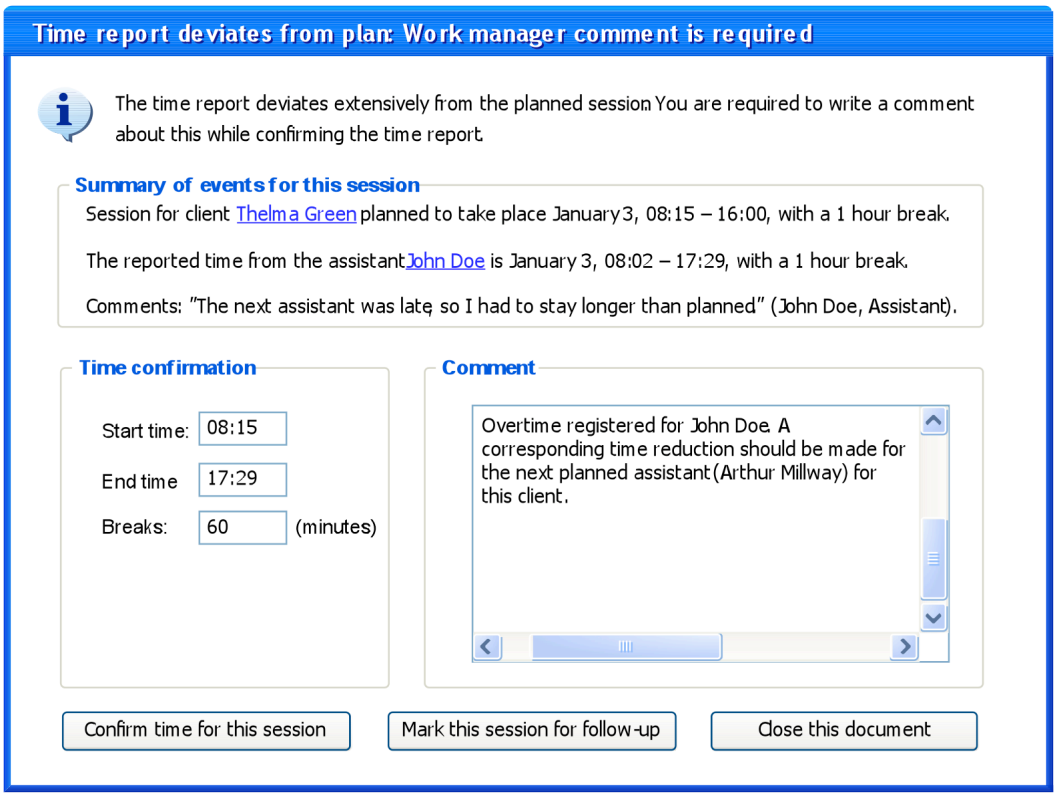

Figure 2 – Sample screen shot from the LSS design prototype

The signature norm is strongly rooted in the Swedish public sector, and so far the Social Insurance Agency has not agreed the alternative approach to accountability. Thus, there is a strong counter-force to implementing this pattern, although there are apparent advantages. A change in norms is an example of innovation that unleashes the potential of IT in the administration work.

Instantiation 2: In order to show that this is indeed a recurrent pattern, we provide a brief example from the Swedish tax authority. Each year, they send a tax return form to all citizens, specifying the prepaid taxes from the citizen and the anticipated final amount that should be paid. Based on this, the citizen should either pay additional taxes or get a tax return. The citizen needs to audit, possibly revise, and sign the form. A large number of citizens do not revise their forms; they just sign it as is. The form is thus often based on correct information that the tax authority imported from banks, employers, stockbrokers, other government agencies *et cetera*. The act of "signing" the form may be performed in several ways: (i) Using electronic identification (e-ID), (ii) using a pre-printed code available on the tax return form, (iii) using an SMS text message, (iv) using a regular phone, and (v) by signing the printed tax form and handing it in to the tax authority. In 2007, 56% of the citizens signed the printed form (Skatteverket, 2008). The tax authority has repeatedly advocated a revised legislation during the last 20 years (HD, 2005) – if no revisions are made, there should be no need for the citizen to sign the form. This is referred to as silent approval: Acceptance through non-action. It would require less effort

from the citizen, reduce the administrative costs for the tax authority, and have a positive environmental impact due to the reduced transportation of papers. The tax authorities in countries such as Denmark, Finland and Ireland already use the concept of silent approval, and Swedish authorities are currently making an inquiry into the possibility for a revised legislation to allow it (ibid.). Thus, from an efficiency point of view, we may conclude that the design of various technological solutions to improve how things are done still leaves room for organizational improvement. The design of new instruments to do things the old way is not entirely satisfactory – there is a need to induce a change of norms (in this case even legislation) in order to improve processes (Stamper et al., 2000), still ensuring that the process is accountable by allowing the citizen to review important information and potentially object to it if it is considered erroneous. Thus, the problem at hand lies within the problem class defined in the social transparency pattern, and the desired solution is in line with the solution proposed by the pattern.

Discussion

When it comes to the social aspects of design, one must reflect upon the situation that many social phenomena do not lend themselves well to formalization. The approach of socio-instrumental design patterns should not be conceived as an attempt to formalize social aspects of development. It is rather to be seen as a way of abstracting lessons learned and packaging them as a useful representation, avoiding to present fragments of design belonging to one of the two sides of the coin: the social and the instrumental (cf. Ågerfalk & Eriksson, 2006). The presented design pattern concerns a specific class of problems; still there is a need for a human creativity to determine its applicability in a specific context. A socio-instrumental design pattern emphasizes the relation between properties of technology and social issues. The LSS/LASS example proposes that we may receive accountability without signatures, however there is a condition that needs to be met: The IT support needs to clearly reveal important representations of the "state of affairs", but also the performed social actions which lead to these representations. In order to assess the accountability of our representations, we need representations of the events that resulted in the representations of objects. The difficulty of formalizing social phenomena leads to rather abstract technological prescriptions, and some prescriptions that are aimed at the design process rather than the characteristics of the design product. In the LSS/LASS case, a stake-holder-centric design process was a necessary design strategy in order to determine suitable characteristics for the IT artefact, and the need for change of norms in the organization. For the designers, knowledge of the social transparency pattern would surely have been valuable food for thought and guidance for action in the design process. Our experiences from the Swedish public sector indicate that the problem in the LSS/LASS case is an instance of a problem class. The formulation of a design pattern may thus aid designers identify other instances of the problem class in other situations, and reason about solutions in a situation. The brief example of the Swedish tax authority shows one additional instance of the problem class.

The concept of design patterns may be conceived of as a way to package "light-weight" design theories. Such patterns are typically focused on either the instrumental or the social. In this paper we have proposed an extended conception of how to reason about solutions in design patterns to acknowledge both social and instrumental aspects. The provided examples show a problem class and propose a way of solving problems that are instances of that class. This way of presenting the solution makes it accessible without a sophisticated preconception of language philosophy and social action theories, which has, for example, proven problematic in previous research on actability theory (Ågerfalk, 2003). The coherent representation of the design pattern concept facilitates a means to communicate such theories between research communities, and to practitioners. Further investigations into how such patterns are interpreted and put into action in practice should be conducted to validate the hypothesized usefulness of the approach.

The concept of the socio-instrumental design pattern reinforces the need for a flexible view of method support in IS development. Specifically it embraces the view that the design process should be influenced by and tailored to the social context of the IT artefact being developed. Most method engineering approaches and related modularization constructs, such as those mentioned in the

introduction, rely on some way to characterize the development situation in order to tailor situation specific methods (e.g. van Slooten & Hodes, 1996; Mirbel & Ralyté, 2006). Socio-instrumental design patterns, as discussed in this paper, can be seen as a specific way to characterize design situations based on an articulated theoretical perspective on the use of information technology in social settings. While exploring this in detail is beyond the purpose of this paper, we would argue that they may serve an important role in future method engineering work that aims to provide descriptions of development situations sensitive to the social context of the system to be developed.

To sum up, this paper makes two primary contributions: (i) At a high level, the concept of socio-instrumental design patterns has been proposed, which is targeted towards scholars and reflective practitioners who seek to find a way to abstract and re-use their experiences. (ii) At a lower level, the Social Transparency Pattern has been presented. The pattern is useful in a practical design context, as an instrument to find a solution to a specific class of accountability problems in organizational change work. Ideally, other scholars will adopt the idea of socio-instrumental design patterns, suggest refinements of how to construct them, and provide examples of other patterns based on this concept.

Acknowledgements

The action research project mentioned in this paper is funded by the Swedish Governmental Agency for Innovation Systems (VINNOVA) project E-*services for cooperative use* (e-tjänster för Sambruk).

References

Ågerfalk, P. (2003) "Information Systems Actability: Understanding Information Technology as a Tool for Business Action and Communication". Doctoral dissertation, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Ågerfalk, P. J. and Eriksson, O. (2006). Socio-Instrumental Usability: IT Is All About Social Action, Journal of Information Technology, 21(1), 24–39.

Alexander, C., Ishikawa, S. and Silverstein, M. (1977): "A Pattern Language". Oxford University Press Bansler (1989)

Bevan N, Kirakowski J, Maissel J (1991) "What is usability?" In Proceedings of the 4th International Conference on HCI, Stuttgart, September 1991

Bevan N (2001) "Quality in use for all". In User interfaces for all (Ed, Stephanidis C) Lawrence Erlbaum.

Borchers, J. O. (2001): "A Pattern Approach to Interaction Design". John Wiley and Sons Dewey, J. (1938) "Logic: The theory of Inquiry"

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. Information and Software Technology, 38(4), 275–280.

Checkland, P. (1999). Systems thinking, systems practice, John Wiley & Sons, Chichester, UK.

Ehn, P. (1995) "Informatics: Design for Usability". In The Infological Equation: Essays in Honor of Börje Langefors, (B. Dahlbom, Ed.) Gothenburg, Sweden: Gothenburg studies in information systems, Gothenburg University, pp. 159-174

Gamma, E. , Helm, R., Johnson, R. and Vlissides, J. (1995) "Design Patterns: Elements of reusable object-oriented software". Addison-Wesley Publishing

Giddens, Anthony (1984) "The Constitution of Society. Outline of the Theory of Structuration". Polity. Cambridge.

Graham, Ian (2003): "A Pattern Language for Web Usability". Pearson Education

Goldkuhl, G., Lyytinen, K. (1982) "A Language Action View of Information Systems". In Ginzberg, Ross (eds.), Proceedings of the 3rd International Conference on Information Systems. Ann Arbor, December 13-15, 1982.

Goles, T. and Hirschheim, R (2000). The paradigm is dead, the paradigm is dead... long live the paradigm: the legacy of Burrell and Morgan, Omega, 28 (3), 249–268.

HD (2005) http://hd.se/ekonomi/2005/12/01/signering_deklarationen_vaeg_bort/ , accessed on August 15, 2008 (Helsingborgs Dagblad is a Swedish newspaper).

Karlsson, F. and Wistrand, K. (2006). Combining method engineering with activity theory: theoretical

grounding of the method component concept. European Journal of Information Systems, 15, 82-90.
Goldkuhl, G., Ågerfalk, P. J. (2005) "IT Artefacts as Socio-Pragmatic Instruments: Reconciling the Pragmatic, Social, Semiotic and Technical". International Journal of Technology and Human Interaction, Volume 1, issue 3.
Hevner et al, 2004 ISO 9241 (1994) "ISO 9241-11 DIS Ergonomic requirements for office work with visual display terminals (VDTs):– Part 11: Guidance on usability"
James, W. (1907) Pragmatism. A New Name for Some Old Ways of Thinking. Longmans, Green & Co., New York Kotter (1996)
March, S. T. and Smith, G.F. (2003) "Design and natural science research on information technology". Decision Support Systems 15(1995) pp. 251-266
Mirbel, I. and Ralyté, J. (2006). Situational Method Engineering: Combining Assembly-Based and Roadmap-Driven Approaches, Requirements Engineering, 11(1), pp. 58–78.
Nielsen, J. (1993) "Usability engineering". Academic Press Inc., Boston.
Mumford, E. (1995) "Effective Systems Design and Requirements Analysis: the ETHICS Method". Macmillan
Orlikowski (1992) Orlikowski, W. and Iaccono, S. (2001) "Research Commentary: Desperately Seeking the 'IT' in IT Research—a Call to Theorizing the IT Artefact," Information Systems Research, 12(2), pp. 121–134.
Preece, J. Rogers, Y., Sharp, H. (2003) "Interaction design: beyond human-computer interaction". John Wiley & Sons. ISBN 0-471-49278-7.
Rolland, C. and Prakash, N. (1996). A proposal for context-specific method engineering. In S. Brinkkemper, K. Lyytinen & R. Welke (Eds.), Method Engineering: Principles of method construction and tool support (Vol. 191–208): Chapman & Hall.
Sjöström, J. and Goldkuhl, G. (2004) "The semiotics of user interfaces : a socio-pragmatic perspective", in Virtual, Distributed and Flexible Organisations - Studies in Organisational Semiotics, Liu, K. (Ed.), Kluwer Academic Publishers. ISBN 1-4020-2161-5.
Sjöström (2008) "Making Sense of the IT Artefact – a Socio-Pragmatic Inquiry into IS Use Qualities". Licentiate Thesis, Linköping University. FIF-thesis No. 95 . ISBN: 978-91-7393-796-2 .
Sjöström & Goldkuhl (in press) "Socio-Instrumental Pragmatism in Action". In Handbook of Research on Socio-Technical Design and Social Networking Systems. (Whitworth, B. and de Moor, A., Eds.). IGI.
Skatteverket (2008) www.skatteverket.se, accessed on August 15, 2008
van Slooten, K. and Hodes, B. (1996). Characterizing IS Development Projects, In Method Engineering: Principles of method construction and tool support, (Ed, Welke R), Chapman & Hall, pp. 29–44. SOU (2006:89)
Stamper, R. K., Liu, K., Hafkamp, M. and Ades, Y. (2000). Understanding the Roles of Signs and Norms in Organizations: A Semiotic Approach to Information Systems Design, Behaviour & Information Technology, 19(1), 15–27.
Tidwell, J. (2005) "Designing Interfaces: Patterns for Effective Interaction Design". O'Reilly and Associates
van Welie, M. and van der Weer, G. C. (2003) "Pattern Languages in Interaction Design: Structure and Organization". In Proceedings of Interact '03, 1-5 September, Zürich, Switzerland, Eds: Rauterberg, Menozzi, Wesson, p527-534, ISBN 1-58603-363-8, IOS Press, Amsterdam, The Netherlands
van Welie, M. (2008) www.welie.org, accessed on June 19, 2008