In proceedings of the
Tenth International Conference on Information Systems Development (ISD2001)
London, UK, 5–7 September, 2001

DEMYSTIFYING THE INTERNET-BASED SOFTWARE ARTEFACT

Fredrik Karlsson, Pär J. Ågerfalk and Anders Hjalmarsson*

ABSTRACT

There is a growing interest in information systems implemented as Internet-based software artefacts. Little attention has been paid to a comprehensive picture of such artefacts and their difference in relationship to traditional software artefacts. This paper presents an analysis of the Internet-based software artefact in order to expose differences from and similarities to traditional information systems. The analysis is based on Organizational Semiotics and the concept of actability focuses on user requirements as a basis for information systems development.

1. INTRODUCTION

During the last decade, we have seen a shift in focus of information systems development (ISD) from 'traditional' software artefacts (TSAs) to distributed applications delivered via the Internet. This has led to a shift in the technology used to provide software solutions. Even the Internet as a platform for applications has changed. In the early days of the Web, the platform was mainly used for e-mail exchange and trivial websites. Although these services still constitute a great deal of the Internet, more sophisticated applications are increasing in number. They are sophisticated both in content and in the technology on which they are based. Mobile Internet is the latest in a trend that is making information systems available to users. We refer to this type of IT-system as an Internet-based Software Artefact (IBSA). Consequently, we do not restrict our analysis to Web-based systems, as does for example Isakowitz et al. (1998).

These more complex software artefacts impact on organizations to a greater extent than before. The expansion in scope of these artefacts presents new opportunities for doing business. Many hypotheses have been made about how these artefacts will change

* Fredrik Karlsson, Research Group VITS, Dept. of Informatics (ESA), Örebro University, SE-701 82 Örebro Sweden. Pär J. Ågerfalk, Research Group VITS, Dept. of Informatics (ESA), Örebro University, SE-701 82 Örebro Sweden. Anders Hjalmarsson, Research Group VITS and the Research Platform Trading, School of Business and Informatics (IDA), University Collage of Borås, SE-501 90 Borås, Sweden.

the entire structure of industries (cf. Hoffman *et al*. 1995; Tenenbaum 1998). Nevertheless, there is a lack of understanding of the development of this type of artefact. In our role as researchers and consultants, we have been working on projects involving the development of IBSAs (see Karlsson *et al*., 2001). In so doing, we have worried about how this type of artefact and its development differs from traditional software artefacts. Practitioners emphasize the newness of technologies, design, development environment and development process. For example, one of the projects we participated in aimed at developing an Internet-based online ordering service. The project team had to manage a number of challenges, including new technology (Windows DNA), a new development process (Rational Unified Process), and a new type of deployment (online). In their eyes, the IBSA was new and mysterious.

In the literature, there seems to be a focus on Web applications, or more specific websites, as an end product. For example, the usability aspect of websites has been discussed frequently (cf. Nielsen, 1999). Other authors address the modelling aspect of Web-based information systems (cf. Conallen, 2000; Takahashi and Liang, 1997) and the technical aspects (cf. Spreitzer and Janssen, 2000; Urien, 2000; Juric *et al*., 2000). Overmyer (2000), focusing on website requirements engineering, has applied a somewhat broader perspective, and, complex websites excluded, found three major differences: different focus, different disciplinary emphases, and shorter life-cycles. Still, the literature gives only a fragmented description of the IBSA. With only a fragmented picture of the artefact, it is difficult to see the effects of changes in different parts of an IBSA project or its conditions, and predicting these effects is even harder.

With the increasing size and complexity of IBSA-projects it is important to gain a comprehensive picture of the IBSA and its characteristics. This paper focuses on remedying the fragmented picture of the IBSA by presenting an empirically justified description of its characteristics. The aim, in contrast to, for example, Isakowitz *et al*. (1998), is to demystify the IBSA and its development by pointing at similarities with rather than differences from traditional ISD, with special focus on user requirements. Today we can find games, e-commerce systems, corporate intranets, *etc.*, based on the Internet platform. With such a wide variety of artefacts available, we limit our attention to software artefacts for business action and communication.

In this paper we adopt a research approach, including an analytic framework based on Organizational Semiotics and the concept of actability (Ågerfalk *et al.*, 2001) used for the analysis of the IBSA and structuring of the results. We conclude by reflecting on the IBSA and how the results will influence our future work.

2. RESEARCH APPROACH

A qualitative approach has been used for analysis of the IBSA. The Grounded Theory of Strauss and Corbin (1998) inspired our approach. Our analysis was performed in a reflective and iterative fashion (Alvesson and Sköldberg, 2000; Walsham, 1995). In this approach, the evolving theory plays an active part in the collection and interpretation of data (Walsham *ibid.*), and the strengths of both induction and deduction are utilized. In addition, an explicit pre-categorization in the form of an analytic framework has been used to direct attention during analysis.

The analytic framework was based on the concept of actability (Goldkuhl and Ågerfalk, 2000) and the semiotic framework (Stamper, 1994). Actability is defined as ‘an information system's ability to perform actions, and to permit, promote and facilitate the performance of actions by users, both through the system and based on information from the system, in some business context' (Goldkuhl and Ågerfalk, *ibid.*). From this definition we have derived the four main categories of our analytic framework: the actor, the action, the artefact, and the business context (see Figure 1). Actors are the people and organizations performing and interpreting actions through and by an artefact, i.e., the users of the artefact (Ågerfalk, 2001). The ternary relationship between these three categories is shown in Figure 1. Three additional binary relations (dashed lines) are also depicted in Figure 1. These show that there might be generic properties that are important factors in relation to artefacts irrespective of the (type of) action performed, such as general human-computer interaction heuristics (Nielsen, 1993). Every action is performed and interpreted by actors within a specific business context; that is, the persons involved are supposed to be familiar with the business they are doing. Action can in this context be understood as speech acts (Searle, 1969) or communicative actions (Habermas, 1984).

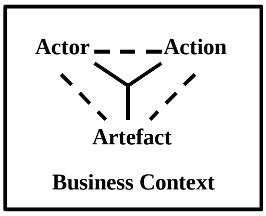

Figure 1. The four main categories of the analytic framework (Ågerfalk *et al.* 2001)

More detailed analysis of the action through the artefact requires examination of the main categories of the artefact. To this end, the semiotic framework (Stamper 1994) was used as a guide to support the actability categories and to keep the analysis focused on demarcated aspects of the IBSA. The semiotic framework is a six-layer model acknowledging both social and technical aspects of information systems. In our framework, artefacts can be studied at five different semiotic levels, referred to as: physical world, empiric, syntactic, semantic, and pragmatic. As in the semiotic framework, the idea is that one level can, in principle, be studied in isolation from the others. Even so, every level relies on a coherent set of features of the underlying level.

At the physical world level we are primarily interested in the systems' hardware. At the empiric level the focus is on measurable properties of the hardware, such as transmission rates. At the syntactic level the focus is on the complex structures that constitute the system's software. Within actability, we sometimes talk about an information system's *message processing* as the automated parts of the system (Ågerfalk, 1999). This is obviously related to these three lower semiotic levels, which Stamper (*ibid.*) refers to as the *IT platform*. Another part of the IT platform, and a category at the syntactic level, is the actability concept of the *screen document*. A screen document is the carrier of communicative action and is used as an interactive arena for the language game played via the system (Ågerfalk, *ibid.*).

On a functioning IT platform, which can be thought of as the medium for communication, various *human information functions* can be performed. This is also Stamper's (*ibid.*) term for the aggregate of the uppermost semiotic levels. Within these, the semantic level concerns the 'meaning' of the concepts used in the communication through the artefact. This corresponds to the *propositional content* (Searle, 1969; Goldkuhl and Ågerfalk, 2000) of the speech acts performed. Pragmatics, on the other hand, concerns the intentions of communicating certain concepts and the potential social consequences for communicators and interpreters. In this sense, it is possible to talk about an *action mode* associated with a particular propositional content (Goldkuhl and Ågerfalk, *ibid*.; Ågerfalk, 1999). Stamper (*ibid*.) adds a sixth level on top of the pragmatic level, referred to as 'social world'. This is an unfortunate choice of name, however, partly because pragmatics is also highly social; pragmatics must be understood in relation to existing subjective as well as inter-subjective knowledge (Goldkuhl and Ågerfalk, *ibid*.; Ågerfalk *et al*., 2001). We refer to this knowledge, which is both a prerequisite for and a result of action, as a *cognitive base*. The cognitive base can be divided into a subjective part and an inter-subjective part—a personal cognitive base and a shared cognitive base (*ibid.*). An important aspect of information systems from an actability point-of-view is their facility to remember actions performed and important prerequisites for action (Ågerfalk, 1999). This *action memory* is an important part of the cognitive base.

Validity is another important concept for actability. The successful performance of actions relies fundamentally on an interpreter's acceptance of the validity of the actions performed. This validity concept draws on Habermas' (1984) universal validity claims, presupposed by communicators to be accepted by interpreters in order for communication to be successful. As we se it, validity, and Habermas' validity claims, concern syntactics (the claim of *comprehensibility*), semantics (the claim of *truthfulness*), pragmatics (the claim of *sincerity*), as well as the cognitive base (the claim of *rightness*).

Altogether, these categories at their different semiotic levels contribute to the understanding, and the realization, of the *action potential* of the artefact. That is, they contribute to the perceived and actual action repertoire that the artefact delivers to its users.

The empirical scene for this work was framed by several action research projects in different Swedish software development organizations. Additionally, experiences from the authors' work as independent consultants, developing IBSAs on a commercial basis, have served as an important knowledge source as well as a source for further reflection.

3. THE INTERNET-BASED SOFTWARE ARTEFACT

This section follows the structure of our analytic framework: we start with actors, and follow on with the artefact. Throughout the section, action is considered in relation to these two categories, and the business context is referred to when appropriate.

3.1. The Actors

There are two main categories of actors that use IBSAs. First, we have the host organization that owns the artefact. Second, we have the users of the artefact. These constitute the host organization's target group and can be further categorized into three classes of generic target groups (see Table 1): internals, partners, and public. Users classified as internals are members of the host organization and can thus be assumed to have a thorough understanding of the business context. Partners are actors in a strong mutual, usually contractually based, relationship with the host organization. Those users outside the host organization and with no formal connection to it are the general public.

Table 1. Generic target groups and methods of connecting to an IBSA

	Internals	Partners	The Public
Controlled environment	Intranet site	Extranet site	Internet site
Open environment	Internet site	Internet site	Internet site

We can add a vertical dimension illustrating how the users connect to an IBSA. At least two principle types of connection are possible. A controlled environment requires authentication before or during use. The opposite is an open and insecure environment. These two dimensions are utilized in Table 1 to explain the more commonly used terms intranet-site, extranet-site and Internet-site, which we have found inappropriate for characterizing the IBSA.

Different target groups have a different understanding of the propositional content and action modes in IBSAs. These actors have different opportunities to use the system and different reasons for using it. Thus, the purpose of the artefact should be the guide for deciding its target groups. A comparison with traditional software artefacts reveals a usage context as well. The main difference is the target groups. If we transform the concepts described above into traditional user categories, internals are 'in-house' development and the public is 'off-the-shelf' development. The partner category receives greater attention in IBSA development because these artefacts have opened up opportunities for deeper co-operation with other businesses. The combination of user categories using the same artefact can be quite complex in IBSAs. A specific IBSA can, for example, be partly an intranet-site and partly an Internet-site.

Application of this actor categorization in a project helped us to distinguish between knowledge about, and power over, the users. This distinction is essential for an understanding of the developer's options to elicit user requirements on the IBSA. The situational characteristics of each IBSA-project imply that the developer has various options by which to gain an understanding of the users. Lack of knowledge about internals, a situation valid in large multinational corporations, may not be so critical because of the degree of power over internal users. When power is utilized by the host organization, the possibility of pursuing the development successfully is at its highest. Either knowledge is attained through power, or power is used to implement the system without gaining sufficient knowledge, irrespective of the users. In Figure 2, the first case involves altering the state from high power/low knowledge to emulate a situation with low power/high knowledge (arrow 1). The second case involves acquiring high power/high knowledge (arrow 2).

With IBSAs aimed at the public, developers must act cautiously. Under these circumstances developers often face situations where their knowledge of and power over the users are both low. In this case, developers can only estimate the character of the actors that will use the artefact; at the very most they can find guidance as to the design of the artefact through the definition and characterization of a designated target group of actors.

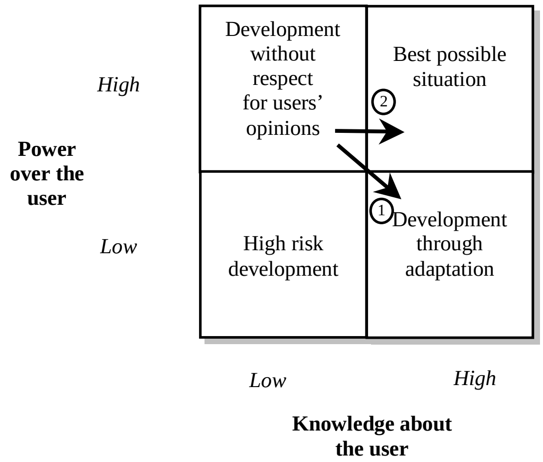

Figure 2. Power and knowledge as prerequisites for understanding the user

If power over the users is low, but knowledge about them is high, the developers can adapt their development effort in accordance with the users' needs. This is the case when developing artefacts for partners, where the level of control may still be fairly low. Nonetheless it is possible to define an exact target group in this context and therefore to gain knowledge about the users.

TSA development faces much of the same challenge with regard to requirements engineering. The difference is that IBSAs are easier to deploy and might therefore reach a more heterogeneous and remote target group. However, the threshold for reaching a competitor is lower for IBSAs than for TSAs. Therefore, this situation is more sensitive.

The time to deliver is related to 'the need for speed' generally characterizing the development of IBSAs. Internet-related development processes are often surrounded with the nimbus that the IBSA should be delivered even more rapidly than 'traditional' artefacts just because it is an IBSA (Overmyer, 2000). The requirement of aggressive release dates can be justified if the artefact's purpose is business critical. With other and more non-competitor-related purposes, the time to deliver should not necessarily be stressed more than in the development of TSAs.

Another property of the IBSA is that the artefact may impose a need for integration with legacy systems. An IBSA is often an interface-shell around a legacy system. This integration may call for advanced mapping activities between different artefacts. In a large information system context, this requires an interaction between many actors and interests during the development process. This may increase the complexity and risk in the development project by opening up existing systems to new user categories (partners and public).

The newness of the artefact could also introduce complexity and risk into the development. In our view, IBSA development is to a large extent similar to traditional

ISD. However, with IBSA development, the construction of the Graphical User Interface (GUI) is often confused with the complete process of developing the IBSA. Graphic design is an important aspect of IBSA design, but it is only part of the entire development process. This confusion can be due to the fact that contracts in the projects we have studied, have often been based on dummy GUI prototypes. The critical issue is to combine traditional ISD practices with Internet technology and artistic work, that is, to create a greater bond between artists and engineers. This reality can make execution as well as management of IBSA projects even more complex than for TSA projects.

Other aspects of newness are linked to the techniques used to realize the IBSA. If these tools have low maturity (e.g., are insufficiently tested, lacking documentation, have uncertainties concerning support, *etc.*), complexity and risk in the project might increase as well. Another facet of the problem is related to the development teams' experience with these new tools, techniques and the Internet as a foundation for development as a whole. If the development team is lacking in competence, a seemingly non-complex project could be permeated with obstacles and uncertainty. Inexperienced teams, requiring training in these development techniques before and during the development of the projects, raise the demands for thorough preparation and support. These teams are also likely to need more time to complete projects than more experienced development teams. Inexperience can increase costs and delay completion of the project. This contradicts the need for IBSAs to be delivered quickly, but inexperience as a cost driver is not unique to IBSAs.

3.2. The Artefact

3.2.1. Human Information Functions

An important property of information systems is their possibility to remember what has been said and done via their action memory. Actors both inside and outside of the host organization can affect the action memory. Actors outside the organization, partners, and the general public, have direct access to the action memory without going through internals. Thus, IBSAs allow for a greater impact on the host organization than TSAs. Webshops are one example where almost all supplier actions could be automatic actions. In this case, with the general public as the target group, the design of the action memory should suit an 'unknown' user.

An action performed by an actor consists of a propositional content embedded into an appropriate action mode (Goldkuhl and Ågerfalk, 2000). Both aspects are visualized through the IBSA and can be made more or less easy to grasp. Users from different target groups differ in their knowledge of the business context. Therefore, their understanding of business actions supported by the artefacts differ. Given the target group, it is still necessary to find a common denominator, in order to deliver a correctly perceived action. Otherwise, for example, actors using a webshop might not be aware at what stage they have actually committed a financial transaction. The obligations of such a commitment must be obvious to the user. Thus, the visual representation and the propositional content of the IBSA becomes more crucial than it is in TSAs, even though less may be known about the users.

IBSAs are easily deployed to a wide range of users. They may be distributed worldwide at a low cost. Then, differences in cultural background and belonging must be considered with respect to action mode and propositional content. Furthermore, there are differences in conceptual models (Norman, 1988) regarding the business context and the consequences of business commitments. Therefore, each IBSA could involve versions adapted for local business contexts. In these cases, actors need to be categorized further within each target group presented in Table 1. This elaboration must be based on the characteristics of each specific project.

Due to the distributed nature of IBSAs, there may be a great physical distance between users and host organization. The artefact may be the only communication channel to the host organization; that is why the concept of trust (e.g., Ågerfalk *et al*., 2000; Friedman *et al*., 2000) becomes important. As we see it, trust is an important part of the cognitive base. Depending on the design of the artefact it is possible to emphasize this characteristic (Ågerfalk *et al*., 2000). For example, a shift from an open to a controlled environment potentially yields a higher degree of trust. Trust can be related to information security. In turn, information security relates to the actability concept of validity (especially truthfulness, sincerity, and rightness), and thus both relate to the cognitive base and the pragmatics and semantics of actions. Of course, the importance of trust is reduced when users have a more formal connection to the host organization. If use of the artefact is purely internal to the host organization, trust is built through additional channels. Depending on the target group, trust is emphasized more in IBSAs than in TSAs.

3.2.2. The IT Platform

In our actability framework we distinguish between the screen documents and the message-processing part of the artefact. Screen documents are the GUI of the IBSA, while the message-processing part is the software and hardware needed to distribute and execute the business actions through the artefact. Since focus in a distributed system is on deployment and processing, we choose to talk about message processing as the deployment architecture of the IBSA. The same overall structure can be found in non-IBSAs.

The screen document should visualize the interactive action potential of the information system. Since the IBSA may be the only channel for business actions between actors, who might be novice users, it needs to be intuitive. Design of an ‘intuitive’ user interface is often based on one or more metaphors, relating to a concept already known to the users. A study of IBSAs reveals that the page-metaphor is the main metaphor for IBSAs deployed (at least partly) as websites. Within the choice of ‘base-metaphor’, different business-specific metaphors may be found, such as the ‘shopping cart’ used in many webshops. The combination of metaphors should be based on the artefact’s target group. With a more heterogeneous actor group we can relate to the discussion about the human information functions in finding a common denominator in a metaphor. The use of different metaphors depends on possible techniques available in the deployment architecture.

The design of the IBSA is restricted by the host organization’s GUI standards. Since in many cases the IBSA is a publicly available artefact, it has to promote the host organization’s identity. The identity comes from the host organization’s perceived image in other business arenas. Via the design, this characteristic of the IBSA indirectly restricts the mood that screen documents communicate to users. The mood of the artefact is the interacting user’s first impression. Hence, it is important in an environment where alternatives might be ‘one click away’. This characteristic is most important with a public target group.

Each IBSA has a certain action potential, which is available through its screen documents. The action potential can be related to the artefact’s degree of interaction. The degree of interaction has two dimensions, illustrated in Figure 3. First, in the vertical dimension we find the terms ‘dynamic‘ and ‘static’, which refer to whether the presented information is assembled in run-time or stored ‘as-is’ (e.g., static HTML). A dynamic presentation could, for example, be a WAP-site depending on ASP-technology. Second, in the horizontal dimension we can classify the artefact as non-affectable or affectable. ‘Affect’ is used in this characteristic to indicate that use of the IBSA can affect the action memory. IBSAs could have a purpose seldom found in TSAs—information publishing. This case is found in the bottom-left corner in Figure 3. This type of IBSA, sometimes referred to as brochureware, has the lowest possible degree of interaction. At the other extreme is an IBSA based on run-time assembly of information that is affectable, such as a typical Web shop.

	Non-affectable	**Affectable**
Dynamic	Information based on action memory content *E.g., viewing info. from other (legacy) system.*	Action memory updated by user *E.g., a webshop or e-business application*
Static	No connection with action memory *E.g., traditional non-interactive Web site*	No connection with action memory *E.g., simple form-based Web site capable of sending e-mails*

Figure 3. Different degrees of interaction provided by an IBSA.

Principally, use of the Internet implies three parts constituting the deployment architecture. The first part is the server containing the IBSA, or software for run-time assembly of an IBSA, before it is deployed. In order to distribute the IBSA we need a network and finally a client capable of using the IBSA. Since each of these components consists of both hardware (physics) and software (syntactics), we refer to them as the server, the client, and the network configuration (with corresponding server- and client performance, and network throughput at the empirical level). The profile of each component, and hence the deployment architecture, will depend on the type of IBSA, whether it is a business system in Java or an application for mobile Internet, *et cetera*.

Within each configuration we find a wide range of hardware and software. In the server configuration we find servers, databases and middleware. PCs, mobile devices, browsers and communication software are parts usually found in the client configuration. Main parts of the network configuration are network hardware, communication protocols and software. However, choices from the range of technologies cannot be made independently of each other. It all boils down to mixing these technologies with the demands from the required human information functions and the design of screen documents to create desired action potential. Compared with TSAs, we have less control over the client configuration, and have a more complex mix of technologies to balance.

The choices in each configuration affect the technical security and performance. Technical security is a brick in building trust at the cognitive base. Using a secure protocol for transferring data used by the IBSA is one way to show that the host organization is trustworthy. Considering performance, each part in the deployment architecture has its own performance. Together they give the total performance of the artefact. Performance depends highly on where different parts of the IBSA are executed. It is, for example, common to talk about thin clients, thick clients and distributed components (Conallen, 2000).

Each part of the IBSA, screen documents, middleware, *et cetera*, has to be realized with one or more programming languages. Significant for IBSAs is the need to integrate different kinds of programming languages. The situation depends on possible needs to integrate with legacy systems, the server platform, the power of the clients, and the strategy for deployment. For example, a WAP-service cannot today rely on client-side execution.

4. CONCLUSION

In this paper we have used an analytic framework based on actability and organizational semiotics to reach a more comprehensive picture of the IBSA. There are differences in emphasis between IBSAs and TSAs, but the characteristics themselves are not unique. TSAs have a usage context just as do IBSAs, but with different target groups. Internals can be thought of as traditional 'in-house' development users, and the general public as 'off-the-shelf' development users.

In order to deliver 'correct' action potential it is important to understand who is going to use the artefact. Thus, just as in traditional projects, understanding the users' requirements is important, but it differs in prerequisite and in focus. The target group could, for example, make it difficult to manage the combination of low knowledge of and power over the users, the need to design for trust, the right mood and intuitive GUI, *et cetera*. These characteristics must be managed through an accurate deployment architecture, which can vary greatly in configuration and performance between target groups. This is different from TSAs where you can prescribe a certain configuration in order to guarantee a certain performance.

The degree of interaction affects the need for requirements engineering. A static website presenting information could mean less focus on requirement work than usual. However, a dynamic and affectable IBSA is as complex an artefact as a TSA. In such cases, requirements work cannot be reduced. Since IBSAs often involve legacy system integration, IBSA requirements engineering could involve a great deal of reverse engineering, but so can TSA development; they also inherit restrictions, as well as potential benefits, from legacy systems.

Aggressive release dates have been proposed as one of the more significant characteristics of the IBSA. These can be justified in projects involving business critical actions, but such projects exist in traditional ISD as well. In the case of an IBSA, however, the artefact could itself constitute the whole business. Even though IBSAs are 'hot' and built with new technologies, we must remember that so were TSAs 10 years ago.

In this paper we have shed some light on the IBSA phenomenon from a user requirement perspective. The deeper understanding of this type of artefact, its characteristics and specifics will, for example, be used in our work with configuration of methods to support the development of IBSAs. The aim is to integrate this work into a rigorous framework for method configuration (cf. Karlsson *et al*. 2001). The results will also serve as input to our research on evaluation of actability, as well as our research on how to characterize systems development practices on the basis of creative ability and systematic work.

5. REFERENCES

Ågerfalk, P. J., 1999, Pragmatization of Information Systems – A Theoretical and Methodological Outline, Licentiate thesis, IDA, Linköping University, Sweden.

Ågerfalk, P., J. 2001, Who's the User in User Centred Design? In poster proceedings of HCI International 2001, August 5–10, New Orleans, LA, USA.

Ågerfalk, P. J., Karlsson, F., Hjalmarsson, A. 2001, Exploring the Explanatory Power of Actability: The Case of Internet-based Software Artefacts. Accepted for IFIP WG8.1 Working Conference: Organizational Semiotics: evolving a science of information systems, 23-25 July, Montreal, Canada

Alvesson, M., Sköldberg, K., 2000, Reflexive methodology: new vistas for qualitative research. Thousand Oaks, Calif. SAGE, London.

Conallen, J., 2000, Building Web Applications with UML. Addison-Wesley, Reading, MA, USA.

Goldkuhl, G., Ågerfalk, P. J., 2000, Actability: A Way to Understand Information Systems Pragmatics. CMTO Research Papers No. 2000:13, Linköping University. Presented at the 3rd International Workshop on Organisational Semiotics, 4 July 2000, Stafford, UK.

Habermas, J., 1984, The theory of communicative action 1. Reason and the rationalization of society. Beacon Press, Boston.

Hoffman, D. L., Novak, T. P., Chatterjee, P., 1997, Commercial Scenarios for the Web. In Kalakota R, Whinston A B (eds.) Readings in Electronic Commerce, Addison Wesley, Reading MA (USA), pp. 29-54.

Isakowitz, T., Bieber, M., Vitali, F., 1998, Web Information Systems. Communications of the ACM, Vol. 41, No. 7, pp. 78–80.

Juric, M. B., Rozman, I., Hericko, M., 2000, Performance comparison of CORBA and RMI. Information and Software Technology, Vol. 42, Issue. 13 (Sept., 2000), pp. 915-933.

Karlsson, F., Ågerfalk, P. J., Hjalmarsson, A., 2001, Process Configuration with Development Tracks and Generic Project Types. Accepted for the 6th CAiSE/IFIP8.1 International Workshop on Evaluation of Modelling Methods in Systems Analysis and Design (EMMSAD'01), Interlaken, Switzerland, 4-5 June 2001.

Nielsen, J., 1993, Usability Engineering. Academic Press, San Diego, CA.

Nielsen, J., 1999, Designing Web Usability. New Riders Publishing, Indianapolis, Indiana, USA.

Overmyer, S. P., 2000, What's Different about Requirements Engineering for Web Sites? Requirements Engineering, Vol. 5, pp. 62-65.

Searle, J. R., 1969, Speech Acts. An essay in the philosophy of language. Cambridge University Press, London, UK.

Spreitzer, M., Janssen, B., 2000, HTTP 'Next Generation'. Computer Networks, Vol. 33, Issue 1-6 (June 2000), pp. 593-607.

Stamper, R. K., 1994, Signs, Information, Norms and Systems. In Holmqvist B et al. (eds.), Signs at Work, De Gruyter, Berlin, pp. 349–397.

Strauss, A., Corbin, J., 1998, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. 2nd ed. Thousand Oaks, California: SAGE Publishing, Inc.

Takahashi, K., Lang, E., 1997, Analysis and Design of Web-based Information Systems. Computer Networks and ISDN Systems, 29 (1997) 1167-1180.

Tenenbaum, J. M., 1998, WIS and Electronic Commerce. Communications of the ACM, Vol. 41, No. 7, pp. 89–90.

Urien, P., 2000, Internet card, a smart card a true Internet node. Computer Communications, Vol. 23, Issue 17

Walsham, G., 1993, Interpreting Information Systems in Organisations. Wiley, Chichester.