

# INFORMATION SYSTEMS DEVELOPMENT IN EMERGENT ORGANIZATIONS

## Empirical findings

Toni Alatalo, Harri Oinas-Kukkonen, Virpi Kurkela, Mikko Siponen

\*

## 1. INTRODUCTION

Traditional ISD methods have been criticized for placing too much emphasis, at least implicitly, on stability of IS development activities (Truex and Baskerville, 1998). Advocates of so-called lightweight methods (Beck, 2000; Cockburn, 2000) have presented similar thoughts. Discussions on IS development in emergent organizations (Truex et al., 1999) provide recent profound critique of ISD methods (Truex et al., 2000). In fact, advocates of IS development in emergent organizations have set new goals for developing ISs (Truex et al., 1999).

Emergent organizations are organizations that constantly try to adapt to their changing environments, but they never achieve the stability they are striving for. The idea of emergent organizations as it bears on IS development stems from (Truex et al., 1999; Truex and Baskerville, 1998). They see that emergent organizations have a constantly changing environment in which they ongoingly try to adapt to meet the evolving requirements. In fact, "ideal" emergent organizations are always in change — they will never achieve any stability (Truex et al., 1999).

As can be seen from above, emergent organizations are the very opposite to stable organizations. Due to fundamentally different assumptions on the nature of reality and IS development, the process for designing stable and emergent ISs are seen different (Truex et al., 1999). In fact, the process in these two cases follow

---

\* HYTEC Research Lab, University of Oulu, Department of Information Processing Science.  
<http://hytec.oulu.fi/>

goals contradictory to each other: Ideal ISD in *stable* organizations follow five goals, namely: 1) economic advantages of lengthy analysis, 2) user satisfaction, 3) abstract requirements, 4) complete and unambiguous specifications and 5) new system projects as achievements (Truex et al., 1999). As to IS development in *emergent* organizations, Truex et al. propose the following four goals instead: 1) always analysis, 2) dynamic requirements negotiations, 3) incomplete, usefully ambiguous specifications and 4) continuous redevelopment (Truex et al., 1999).

The existing work on ISD methods in the context of emergent organizations is mostly conceptual-theoretical research (Truex et al., 1999; Truex and Baskerville, 1998; Truex et al., 2000). There is work where emergent technologies have been studied empirically in field studies of intranet and web development (Bansler et al., 2000; Balasubramanian and Bashian, 1998), but they do not address question of emergent organizations, and call for further empirical studies, reaching over longer time periods (Bansler et al., 2000). As a move towards addressing this gap in literature, we seek to evaluate the goals of (Truex et al., 1999) empirically. With the specific organization in this case study, the still ongoing series of development projects, where we have participated as researchers and designers, began already in 1998. Here the focus is on a subproject that took place during year 2001.

This paper is organized as follows: Section 2 describes the research setting. Section 3 describes the research results. Section 4 discusses the results in light of the research problems and related research. The work is concluded and future research questions addressed in section 5.

## 2. RESEARCH SETTING

To investigate how IS development occurs in an emergent organization, we have studied a particular development process, with the aforementioned proposed goals in mind. The organization in question is POEM, the Northern Film and Media Centre, which is the first regional film and media resource centre in Finland, situated in the city of Oulu. It's main goals include increasing skills, knowledge and production resources and establishing an infrastructure for the regional film and media industry in Northern Ostrobothnia. POEM is a member of the European network of regional and film and media centres, and promotes the distribution of short and documentary film exploring the sources of digital film production and distribution. To advance towards these goals, POEM decided to develop an information system to be used by the different parties over the Internet, mainly using Web browsers — hence we have conceptualized it as a Web IS (WIS) (Isakowitz et al., 1998).

Early ideas about the POEM WIS were discussed already in 1998, when the organization itself did not even exist yet but was in formation. As cooperation with the local university, the first author of this article was involved in this

early planning. The ideas included a location database, with information about e.g. skillful people and suitable locations for film and media production. Also the possibilities of digital media distribution and new forms of content were discussed from early on. The organization was officially launched in autumn 1999, and during spring 2000 two of us worked on requirements analysis and design specification for the IS as a part of the research at the university. Already at this early phase the emergent nature of POEM became apparent. Things changed at a fast pace and the tools that were used to model the WIS were not flexible enough to support change making process. After the preliminary design phase POEM started to search for financing and contractors for the actual implementation of the WIS. During spring 2001 the WIS development project was launched officially. By then the financing and contractors were set.

There were four main participators in the project during year 2001: POEM, a software house, an advertising agency and our research project OWLA. POEM produced all the content for the WIS and connected the database / business logic and the user interface to each other by server page programming. The software house was responsible for database design and business object programming, whereas the advertising agency planned and built the visual look and the user interface. The OWLA project performed usability evaluations utilizing questionnaires, heuristic evaluation and usability testing, and supported the other participants when needed.

The project was divided into four iteration cycles. This paper tackles the first cycle, that was completed before the end of year 2001. Currently, in 2002, the systems exists and several new applications are developed to provide services around it.

To be able to answer specific questions about the ISD process, two of us interviewed at the end of October 2001 the eight key people involved in the development. They were from POEM and the two contractors. The interviews were semi-structured and non-leading, i.e. basically only concepts that had already been introduced by the interviewee during that session were brought up by us. However, we had specified research questions and related interview themes beforehand. None of these questions were shown to the interviewees nor directly asked, but they did speak about them in detail in their own terms. One of the goals during the interviews was to find out how the development process had proceeded from the perspectives of the different parties. Based on these results, we aimed at finding out how the process that had occurred reflected the aforementioned goals of ISD in emergent organizations. The other two themes where: 1. the role of conceptual modelling (Wand et al., 1995; Mylopoulous, 1998) in collaboration between the participating organizations 2. the usefulness of usability evaluations. These other themes will be discussed here aside the main question, as they are also related to the goals: firstly, (Truex et al., 1999) suggest that conceptual models (e.g. diagrams) help to redesign changes to the systems,

and secondly, usability evaluations can be seen as a form of analysis, related to the goal “Always analysis”.

POEM as an emergent organization is summarized as follows: POEM is a young organization where things change fast all the time. This builds up pressure and stress in the minds of the personnel. The publicly (mostly EU) funded organization is one of its kind in Northern Finland, with a lot of expectations and requirements from the outside. However, POEM has a strong vision of its mission. This vision guides the WIS development process and helps to keep the ultimate goal clear in the minds of developers and other people who are involved with the project. Also the potential users (stakeholders) expect a lot from the WIS. POEM tries to satisfy these needs. In the regular meetings with domestic and international (potential) collaborators new requirements constantly emerge. POEM also collaborates closely with the vibrant IT industry in the area.

### 3. RESULTS

In this section we describe our findings related to the four goals of ISD in emergent organizations as stated in (Truex et al., 1999).

#### 3.1. Always Analysis

Plans, models and other work done were analyzed all along the whole project. This was carried out, for example, during initial background research, requirements analysis, design specification, when specifications and propositions were given to POEM, when needs for changes arose, when server page programming was done, during inspections / evaluations and when risks and time use was analyzed.

The whole project was built on continuous interaction between creating and commenting on proposals. The latter took place either immediately after the change proposals had risen and been presented, for example in a meeting, or afterwards through e.g. email or phone.

At POEM, the technical person was the one spending time reading the models the software house had made (UML Class Diagrams). He described them as “handy”. The person that is in charge of the content creation in POEM said that the models were “easy and quite logical to interpret” and that it was easy to detect the relationships between different entities and groups. During this interpretation process, needs for changes arose.

Analyzing was also done, when POEM’s technical person connected the user interface and the database together with so-called server page programming. Indeed, he found some points from plans and the code that needed to be fixed or implemented in a different way. And again, there was the question about addressing the issue during the current iteration cycle or later. Moreover, as a by-product, some new ideas rose during everyday working and in meetings.

In addition, the OWLA project evaluated WIS usability. The techniques used included usability testing, heuristic evaluation and questionnaires. With usability evaluations we aimed at detecting problems, e.g. functions that were hard to use, learn and/or remember. POEM manager told us that, in his opinion, usability tests created a sense of security that the project is going to the right direction. Another interviewee told us that performed tests made her think about new things.

### 3.2. Dynamic Requirements Negotiations

Requirements specifications are often used in contracting between the user organization and software supplier(s) to define the work to be done beforehand. The notion of dynamic requirements negotiations obviously makes this difficult. In this case there was a sort of an open contract between POEM and the software house, where the cost and roughly the amount of work was predefined, but the actual requirements were constructed and prioritized dynamically as described. Before this was possible, mutual trust had to be established — that was a subtle process in itself.

During the actual working period, the software house and the advertising agency implemented change propositions if they were seen as reasonable and technically possible. Sometimes change propositions were moved to next iteration cycle or further, if they came up too late during the particular iteration cycle, if the change propositions were too complicated to implement or if they were still too abstract.

Most of the change propositions came from the person who is in charge of content creation at POEM and from POEM's manager. POEM's technical person discussed with the software house representatives how difficult these changes would be to implement and when it would be best done. In addition, the person who is in charge of content creation in POEM discussed these same things with the advertising agency regarding the user interface. Change propositions that were presented by the software house and the advertising agency were again reviewed by POEM. Also the actual changes done to the database / business logic and user interface were commented on by POEM.

### 3.3. Incomplete, Usefully Ambiguous Specifications

In this case, design specifications were mostly in-house tools at the software house. They were, however, also means of communication from the software house to the people working on the project at POEM, which supported constant analysis as described earlier.

The database and business logic specifications were UML diagrams. When needed, associations, objects, attributes and components were easy to change, delete or add. This is in line with the statement of Truex et al. (1999, 122): “easily maintainable specifications, like object-oriented designs, make it easier

and cheaper to re-specify IT systems when change is needed". However, the changes were easy to do into the database / business logic specifications also due to the fact that they were mostly quite small. As more major changes to the system are on the way, this point can be re-evaluated later.

The specifications were incomplete indeed, i.e. not everything was prescribed in them. Related to this, one major misunderstanding occurred, namely the work on the server page programming. To put it simply, everybody thought that somebody else would do it. Finally, POEM's technical person did the programming. This mix-up occurred probably because there are so many participants in this project. Therefore, the possibility of human error is high. But undeniably it was an ill-defined issue — something to bear in mind while evaluating the usefulness of ambiguous specifications.

### 3.4. Continuous Redevelopment

The project was divided into four iteration cycles. Requirements were divided between the different iteration cycles based on how critical each requirement was. The first cycle naturally included most of the basic functionality. The software house made a suggestion about the division and then it was discussed and agreed upon in one of the steering group meetings. This division changed somewhat during the project.

Change needs occurred all the time and most of them were carried out: new features were added and old features were improved. Changes were almost entirely small ones, which however caused hours of work. There was only one major change that had to be carried out, about colors and photographs. Generally, change needs have been about changing, deleting and adding attributes and associations. Based on the interviews, one could expect substantial change needs to rise in the near future and during the WIS' whole life cycle. They can have a huge effect on the WIS. For example, there will be a need for new databases, services and features (currently user registration and personalization are on the way) plus different user interfaces (e.g. mobile use, possibly digital television).

The user interface is much harder to change than the database and the business logic because of the way the user interface has been implemented. For example, every link in the link list consists of two pictures. Therefore, even small changes to the links would require a lot of work. Indeed, the updater has to be good at graphics processing, hypertext mark-up and client page programming in order to be able to update the user interface when needed. However, WIS actual content is easy to update.

Prototyping was an important aspect of the development process. The schedule for it was tight, but the work was finished in time. At the interview, one of the participants said it was a good idea to construct a demo with a certain deadline. This brought structure to the process and required everyone to be in contact with other participants. In addition, the demo was part of the first usability evaluation. Through this, POEM had a chance to find out early in

the development process how the potential users felt about the WIS, and some problems were discovered that had not been noticed before. Truex et al. (1999, 122) emphasize the importance of prototyping. The experiences received during the development of POEM WIS support this statement.

### 3.5. An Overview of the Findings

Overall the first iteration cycle of the POEM WIS development proceeded along the goals presented in (Truex et al., 1999). This may not be surprising, as the process described resembles the immature ad-hoc way software is often made. Interestingly, however, it became clear during the investigations that the development process of POEM WIS differed from the software houses' other projects with respect to process' formality. Usually the software house performs very formal inspection sessions a few times during the project. Between these sessions, the software house concentrates on programming. In this case, the software house did conduct some formal in-house inspections, but also quite frequently went through the specifications and checked for change needs quite informally with POEM. However, despite this informality, the software house tried to follow its own process model as much as possible, but, possibly because of POEM's emergent nature, realized that they could not perform the project as formally as they were accustomed to. The project manager at the software house thought the reason was that POEM's employees are artists, and not used to software business (unlike many other customers of the software house, such as hardware manufacturers and heavy industry).

Also, when the advertising agency starts a project, they try to plan completely in detail how the project will proceed. Therefore, the plan is not too flexible when change needs rise. Then again, the software house was paid for making the changes, whereas for the advertising agency it was not so. This explains partly why the software house was much more flexible with regard to change making than the advertising agency was.

The questions a) whether the process was productive and b) how it was and could be supported, remain. Of the ways of supporting discussed in (Truex et al. 1999, p. 122), we see here that easily maintainable specifications and prototyping were essential in the process. Whether the architecture resulted as an open one that can be easily utilized in different settings is put to test in near future. So far, there has not been the kind of end-user development nor back-channel communications suggested in (Truex et al., 1999), but these might indeed be good improvements. In this case there is not an internal IT department but a myriad of changing partnerships. Finally, the importance of a proper rewards system, that values adaptation, was demonstrated by a) how it was crucial for POEM and the software house to gain mutual trust to form an open enough relationship that allowed emergence b) the understandable reluctance of the advertising agency to make all the changes to the user interface that had not been properly agreed upon. Of the productivity we only know that the shared

understanding at the time of the interviews was that all the major objectives had been reached on schedule — that the project was considered a success so far.

#### 4. DISCUSSION

The major limitation of this paper is the poor generalization of the results. As here only one organization has been studied — furthermore quite a special one — this may not tell much about other organizations. However, it may be that the issues related to emergence are quite common. In the words of the software house’s responsible manager in this project:

“People have realized that there will always be changes that need to be made into systems. Experienced designers take this into consideration in almost every case, and they try to design the systems so that they can be changed whenever needed.”

Shortly after this the the interviewee referred to the ideology of eXtreme Programming, that he was interested in — albeit in this case it was not actually in use, but an attempt at a more traditional iterative software engineering process took place instead.

Of the interactions between the people in the project and the relationships of their organizations it is difficult to pin-point the most significant events and their root causes. Also the decision to look at the process so strongly in the light of (Truex et al., 1999) has coloured the interpretation. With this modest account we, however, wish to contribute to advancing the understanding of these sometimes overwhelming issues.

#### 5. CONCLUSIONS

Goals of information systems development (ISD) in emergent organizations were presented from the literature, and analyzed based on a case study with such an organization. The process under study did proceed according to the principles for ISD in emergent organizations, even though those principles are in contradiction with the traditional ISD goals, and the company doing the actual software development for the organization is a rigorous one (i.e. tries to manage projects along the lines of the software engineering doctrine). The reasons for this were threefold: 1. software professionals have learned to embrace change in any case 2. the emergent organization’s nature and actions encouraged the software company to adopt practices stated in the goals, such as continuous redevelopment and dynamic requirements negotiations, and 3. the participating academics were studying the phenomena of emergent organizations and therefore intervened in the process in ways that did not enforce a rigid process model. As

for continuing research, the ways of supporting emergence need to be studied more closely and developed further.

## 6. ACKNOWLEDGEMENTS

We wish to thank POEM and the interviewees — without them this study would not have been possible. We are also grateful for the anonymous reviewers who helped us to see what this work is actually about.

## References

- Balasubramanian, V. and A. Bashian: 1998, 'Document Management and Web Technologies: Alice Marries the Mad Hatter.'. *Communications of the ACM* **41**(7).
- Bansler, J., J. Damsgaard, E. Havn, J. Thommesen, and R. Scheepers: 2000, 'Corporate Intranet Implementation: Managing emergent technologies and organizational practices'. *Journal of the AIS* **1**, 1–39.
- Beck, K.: 2000, 'Emergent Control in Extreme Programming'. *Cutter IT Journal* **13**(11), 22–24.
- Cockburn, A.: 2000, 'Balancing Lightness with Sufficiency'. *Cutter IT Journal* **13**(11), 26–33.
- Isakowitz, T., M. Bieber, and F. Vitali: 1998, 'Web information systems'. *Communications of the ACM* **41**(7), 78–80.
- Mylopoulos, J.: 1998, 'Information Modeling in the Time of the Revolution'. *Information Systems* **23**(3/4), 127–155.
- Truex, D., R. Baskerville, and J. Travis: 2000, 'Amethodical systems development: the deferred meaning of systems development methods'. *Accounting, management and information technologies* **10**, 53–79.
- Truex, D. P. and R. Baskerville: 1998, 'Deep Structure or Emergence Theory: Contrasting Theoretical Foundations for Information Systems Development.'. *Information Systems Journal*.
- Truex, D. P., R. Baskerville, and H. Klein: 1999, 'Growing systems in emergent organizations'. *Communications of the ACM* **42**(8), 117–123.
- Wand, Y., D. Monarchi, and J. Parsons: 1995, 'Theoretical Foundations for Conceptual Modelling in Information Systems Development'. *Decision Support Systems* **15**(4), 285–304.

