Evolutionary Business Information Systems -

Perspectives and challenges of an emerging class of information systems[1]

Gustaf Neumann, Stefan Sobernig, Michael Aram
Institute for Information Systems and New Media,
Vienna University of Economics and Business,
Welthandelsplatz 1, 1020, Vienna, Austria

This article reflects on existing and emerging future challenges arising in the area of "evolutionary business information systems," a class of systems that demand an evolutionary software development process and support secondary design on various conceptual layers. We place both existing contributions and future research opportunities in context by referring to an idealized, preliminary system architecture. Finally, we emphasize our pluralistic perspective on the research object and the resulting need for methodological flexibility in the sense of interdisciplinary configurations of research methods.

Keywords: Evolutionary business information systems, Secondary design, Pluralistic information systems research

1 Introduction

This article describes the research area "evolutionary business information systems" that is developed within an understanding of information systems research that sees this field as part of social sciences with the goal of improving business performance. In this context, a "business information system" is understood as a socio-technical system containing human beings and machines which use and produce information to support and enable the processes and operations of an enterprise (Hansen and Neumann 2009). This socio-technical view is also referred to as "ensemble view of technology" (Orlikowski and Iacono 2001).

Information technology can reduce transaction and coordination costs drastically, often leading to significant changes in the way companies run their business. Therefore, Malone et al. (1999) call information systems "tools for inventing organizations." Since technological artifacts (March and Smith 1995) shape the design space of a business information system, it is important to analyze them not only in isolation but to study situated artifacts and the impact of their underlying design decisions within the context of a real-world information system (see Figure 1).

---

[1] The final version of this manuscript was published in English and German via http://www.springerlink.com und http://www.bise-journal.org: Neumann G, Sobernig S, Aram M (2014) Evolutionary Business Information Systems. Perspectives and Challenges of an Emerging Class of Information Systems. Bus Inf Syst Eng. doi: 10.1007/s12599-013-0305-1.
See http://link.springer.com/article /10.1007/s11576-013-0397-8

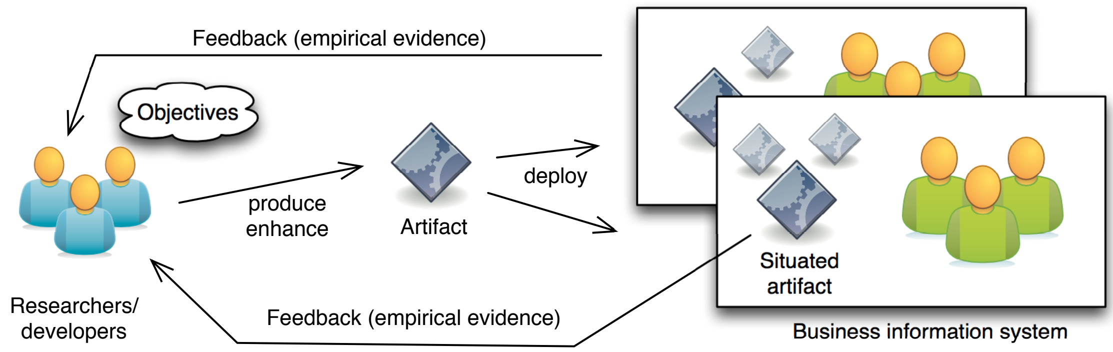


Fig. 1 Situated instantiated artifacts provide empirical evidence.

2 Evolutionary Business Information Systems

Information systems of large organizations are continuously evolving to cope with changing business demands. Therefore these systems face a permanent development lag. Already in the 1990s, Allen and Boynton (1991, p. 435) stated that IS efforts "generally automate the status quo, freezing the organization into patterns of behavior and operations that resolutely resist change." Business information systems are therefore never "finished" (Bjerknes et al. 1991) in the sense of fulfilling all business requirements.

Many approaches have been developed in a range of research areas to reduce this development lag, ranging from participatory design (Muller and Kuhn 1993) and agile methods (Abrahamsson et al. 2010) to model-driven development (Stahl et al. 2006) and software product lines (Clements and Northrop 2007). Although these efforts have led to significant improvements in their fields, a unified view for information systems is missing and, in practice, the characteristic development lag persists.

The challenge of evolutionary business information systems is to provide a socio-technical information system infrastructure that is capable of meeting changing business requirements incrementally, where (unanticipated) changes can be incorporated incrementally (without service interruptions) directly by the stakeholders. These are not necessarily software engineers. Note that the term "evolutionary" refers to the whole information system but not necessarily to single applications or processes. Evolutionary business information systems can support self-coordination by non-hierarchical communication (Kieser and Kubicek 1992). Self-coordination can question existing structures permanently to adapt an organization to changing requirements. Thus, organization development as a permanent activity has to be supported by the information system.

An important concept of evolutionary business information systems is secondary design (Germonprez et al. 2011), which refers to a setting where users of a tailorable information system become the primary actors of its continuous redesign. The users modify the system in the context of their use, often without being aware of the primary design. This design perspective recognizes that people's behaviors and business contexts change over time and that information systems are inhabited and engaged by people who tailor the system for the work they are accountable for. The primary artifact designer gives up central control over the design and allows for user-driven innovation (von Hippel 2009). To this end, business information systems need to be designed as highly tailorable technology (Germonprez et al. 2007) to support a person's reflections, actions, and tailoring of the system. Such a system cannot allow as much freedom to all users as a traditional wiki system, where every user can make arbitrary changes. Instead, it has to provide means for governance like an enterprise wiki system, where domains of responsibilities can be defined, letting the stakeholder modify only those parts for which they are responsible.

Desirable properties of evolutionary business information systems include:

- Seamless integration of incremental (ad-hoc) changes
- Secondary design of the content and behavior through a multitude of contributors
- System introspection to provide feedback about the current system state, behavior and previous actions
- Managing multitudes of versions and variants of instances and schemata
- A balance between system/organizational requirements (governance, stability) and individual demands (flexibility, tailorability)

The development cycle of primary design, as illustrated in Figure 1, is certainly still required, but it is no longer the only mechanism to extend and adapt the system.

Several of these desired properties are available in today's systems. For example, traditional wiki systems already support secondary design, but primarily on the content layer. The research in ad-hoc workflows (Georgakopoulos et al. 1995) typically focuses on small teams of professionals and aims at supporting unanticipated activities that require a rapid workflow execution. Such ad-hoc changes usually address workflow instances rather than workflow schemas. Enterprise mashup (EM) systems are designed for end-user programming, thus enabling users to create personalized, situational applications that address their immediate business needs (Pahlke et al. 2010).

A goal of evolutionary business information systems is to support not only ad-hoc changes on the instance level, but to let domain experts modify and reuse schema definitions directly, e.g. by defining a new class of business processes. This requires them to have sufficient operational knowledge to anticipate the consequences of their design actions. The more practitioners are able to modify the system behavior not only for their personal use, the more important it is to raise the transparency of the system behavior (Breu et al. 2011). Hence, artifacts must be highly introspectable and design activities must be traceable for users to understand the consequences of changes. Evolutionary business information systems permit observations by the stakeholders to establish empirical evidence about business behavior and to analyze and further improve the systems.

When many domain experts are able to create and co-develop the system behavior, a system must be able to deal with a potentially high number of versions and variants in a scalable fashion (see e.g. the case study in Section 3). Finally, when ad-hoc changes affect multiple applications and not only a single instance (ad-hoc workflows) or a single user (EM systems), it is important to provide support for governance to limit the changeability of certain properties to ensure reliability and predictability.

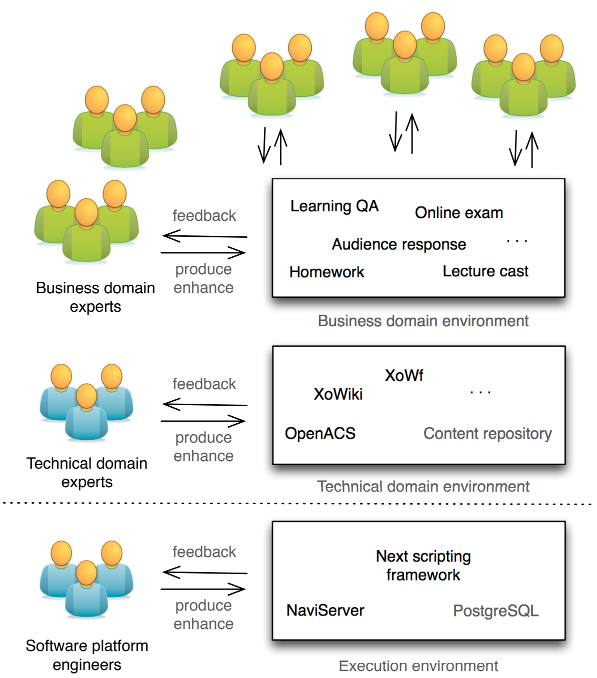
Tab. 1 System archetypes and their evolutionary properties

	Wiki	EM Systems	Ad-Hoc Workflow Systems	Evolutionary Business IS
Ad-Hoc Changes	+	+	+	+
Secondary Design	+	+	-	+
Content Development	+	+	-	+
Instance Development	+	+	+	+
Schema Development	-	-	-	+
Variability Management	+	-	+	+
Control Flow Adaption	-	-	+	+
Feedback Channel	+	-	-	+
Governance Support	-	-	-	+

The following section describes a large situated instantiation in the domain of e-learning which we aim to develop into an evolutionary business information system.

3 A Case Study

A proven path for research in business information systems is to develop artifacts in situ, where the researchers can evaluate the effects of their measures at first hand. We use the Learn@WU system (Alberer et al. 2003) here as an example of a system showing many of the desired properties. It is one of the largest university e-learning systems in terms of use (over 160,000 learning resources, up to 4 million page impressions per day, up to 2,500 concurrent users).



Open source software artifacts in Learn@WU:

- Workflow system supporting ad-hoc changes of schemas and instances (XoWf)
- Enterprise wiki system (XoWiki)
- Community framework, supporting decentralized development by non-technical stakeholders (OpenACS)
- Dynamic scripting language, language support for design patterns, software evolution (Next Scripting Framework, Extended Object Tcl)
- Scalable scripted Web server supporting incremental software evolution, extensive monitoring (NaviServer)

Fig. 2 Sample artifacts and stakeholders in the multi-layered architecture of Learn@WU

The stakeholders of the Learn@WU system are technical domain experts and various groups of business domain experts (e.g. teachers, e-learning assistants, program directors, or the learning quality assurance team) that are able to shape the interactions with and among their students and to develop learning contents and applications. Given the nature of the used software components, all application-specific aspects of the system can be incrementally enhanced without service interruptions. The system supports decentralized development by providing management rights for the domain experts, who are equipped with high-level tools that can be configured and/or extended via scripting. One such adaptation is an audience response tool (Andergassen et al. 2012) based on a wiki-based workflow system (Neumann and Erol 2008) for obtaining student feedback.

The more stakeholders actually modify the system, the more the variability increases. We could, for example, analyze the users' secondary design on the content layer within the wikis in Learn@WU in an approach similar to Germonprez et al. (2011). Beyond that, however, it is also interesting to analyze the variety of the workflow definitions and instances, i.e. the secondary design of the behavior. Currently, the system uses 636 different workflow definitions (defined and modified by 59 contributors) with 1,417 revisions. There are over 500,000 workflow instances with over 2.5 million backtrack points. More than 20,000 participants have used these instances. These figures emphasize the need for a scalable variability management when supporting user-driven development in the large.

In our experience, the provision of wiki-based workflow definitions has led to a higher productivity of the developers and a higher variability of the components. Although the technical support team of Learn@WU consists of only six people, there have so far been about ten times as many contributors who would not have been able to define workflows without the provision of these definitions. We are confident that the number of contributors can still be significantly increased.

Domain-specific transaction monitoring already enables us to better understand the learning activities of our students (Mödritscher et al. 2013).

4 Industry Applications

In general, the field of evolutionary business information systems builds on ideas derived from end-user participation and is extended with concepts from evolvable systems and secondary design. The industry has already begun to adopt these concepts. Enterprise wiki systems (such as Confluence) are used in thousands of companies to improve collaboration and knowledge sharing.End-user participation beyond the content (wiki) layer is less widespread but has found its way into the software portfolios of major players in the IT landscape. For example, SAP has investigated the EM paradigm by prototyping the SAP Research Rooftop Marketplace (Hoyer et al. 2009). Oracle offers EM functionality as part of the *WebCenter* suite. IBM and Software AG have productized their EM platforms as *IBM Mashup Center* and *ARIS MashZone*. The concept of domain-specific languages (Fowler 2010) was introduced to improve the communication between domain experts and developers in practical applications, but its adaption is often limited due to a lack of reliable domain knowledge available to DSL developers (Mernik et al 2005).

The software components of Learn@WU have already attracted the interest of industry and government organizations. For example, Daimler AG employs the components of the Learn@WU system for knowledge management in supply chain management (company and suppliers). LMS.at uses these to serve more than 2,600 schools in the Austrian school sector.

5 Research Directions

The overall research goal is to systematically improve the adaptability of business information systems by stakeholders while still preserving certain governance structures and system stability. The main research directions are:

1. How can we systematically identify business potentials in the growing design spaces?
2. How can a running business information system be developed incrementally?
3. How can we increase the degree of participation of non-technical stakeholders in the (secondary) design process of the business information system?

For each research direction, different research methods have to be applied. For the first direction, empirical evidence stemming from the situated artifacts provides a primary source. The systematic analysis of transaction data is quite established in the area of business analytics, but recently the focus has shifted toward the analysis of the behavior. An important source is coordination science (Malone and Crowston 1994), in particular when transaction data are combined with e.g. external (sensor) data to determine potentials for in-situ improvements. Examples are process mining (van der Aalst 2011), data-driven decision making (Brynjolfsson et al. 2011), and learning analytics (Siemens and Long 2011) in the e-learning domain.

The second research direction addresses the need to increase the flexibility of information systems and aims at lightweight development cycles, both from the organizational and the technical point of view: an operative, enterprise-level system requiring recompilation and restarts after each change would not be able to handle hundreds of updates on a production installation per day. The goal is to work towards evolutionary systems that support self-organization and that can adapt their behavior on the fly. Improving the state of the art in this dimension requires research in self-organizing social systems (Wulf 1999) and the integration of organization and technology development (Wulf and Rohde 1995) as well as research in software flexibility, in particular in the areas of multi-layered software development (Ousterhout 1998), dynamic software evolution (Rank 2002), dynamic languages (Callaù et al. 2011), and software product lines (Clements and Northrop 2007).

The challenge of the third research direction is to increase the ongoing collective participation of domain experts in such a way that they can modify the system directly without violating its integral properties. At least the following areas require further research:

- Methods suitable for specification of modifiable behavior by stakeholders, e.g. engineering of domain-specific languages (Strembeck and Zdun 2009)
- Design principles for constructing and combining compositional units for reuse in the problem domain, e.g. via feature-oriented programming (Apel and Kästner 2009)
- Methods for the systematic provision of appropriate feedback channels for all stakeholders, which also address security and privacy aspects; methods for data/process mining/monitoring suitable for end users
- Methods for scalable variant and version management, schema selection, migration, change frequency, analytics and monitoring; development of decision support and recommender systems based on situation analysis and experiences

Ideally, the research directions should not be addressed in isolation. Instead, research contributions that improve the state of the art in all dimensions in concert should be developed.

6 Interdisciplinary, Domain-Specific Research Configurations

The focus on the ensemble view demands a unified approach based on behavioral sciences and design sciences (Hevner et al. 2004). Recent efforts try to extend established construction-oriented research perspectives (Peffers et al. 2007; Vaishnavi and Kuechler 2007) to better fit the ensemble view (Sein et al. 2011), and to explicitly include users as "reflective and active participants in an ongoing design process" (Germonprez et al. 2011, p. 677).

Putting emphasis on the business domain level requires a focus on domain-specific research to be able to provide appropriate abstractions in the information system (for example, in the e-learning domain). Therefore research in evolutionary business information systems in general demands a pluralistic conception of research (Frank 2006). In our experience, even different instantiations in the same domain require different configurations of methods from various research areas. Figure 3 sketches relevant research areas for investigating an information system as presented in the case study.

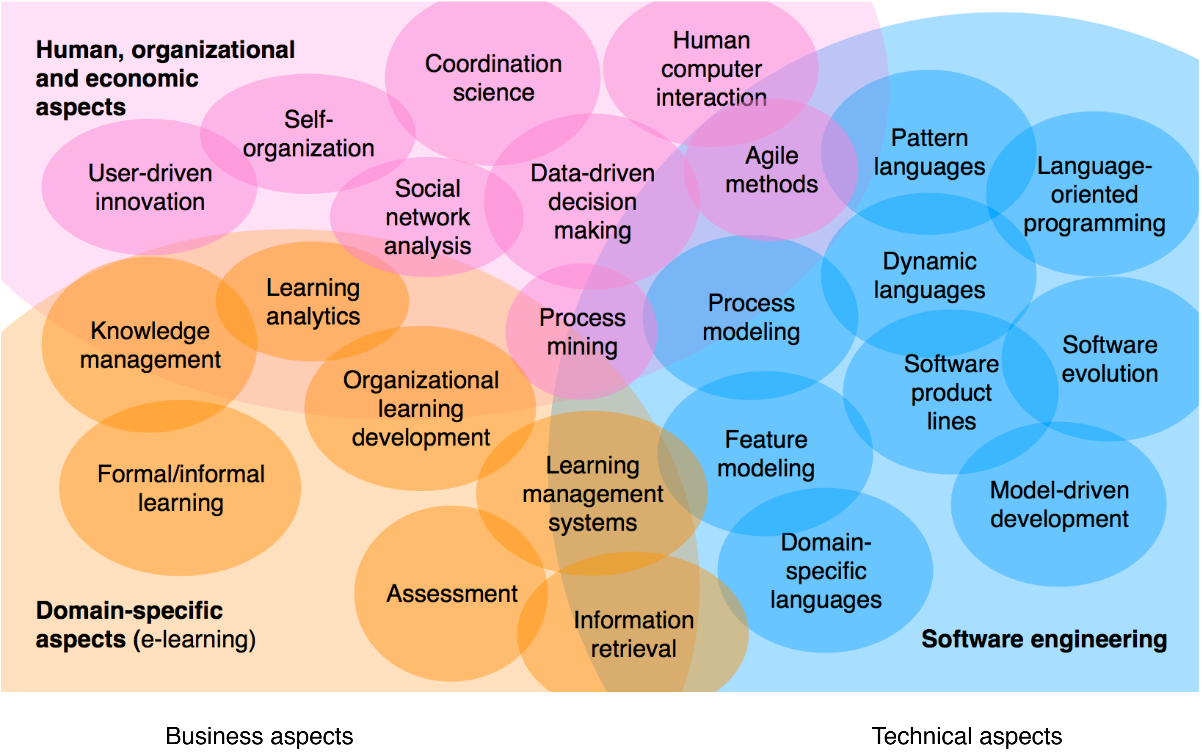


Fig. 3 Reference disciplines for interdisciplinary research in evolutionary business information systems in the area of e-learning

This pluralistic view integrates behavioral research activities (necessary for understanding, explaining and predicting phenomena within existing situated systems) with construction-oriented research activities (necessary for improving the status quo through the creation of innovative artifacts). The research field does not demand that every contributing researcher has an in-situ instantiation at hand. Design-oriented research can follow a "consortium research" approach, which frames the cooperation between researchers and practitioners (Österle and Otto 2010). Furthermore the field can benefit from research outcomes from multiple research fields, ranging from computer science to social sciences.

7 Summary

In this paper, we introduced the concept of evolutionary business information systems, an emerging class of information systems that support secondary design on various conceptual layers. These systems are subject to continuous change, driven by stakeholders with greatly varying degrees of domain knowledge and technical expertise. Software artifacts that are to be included in such a system have to be designed to support continuous (secondary) design and continuous evaluation. We argue that studying evolutionary business information systems demands a pluralistic research perspective as the research object is inherently interdisciplinary. The information systems research community can contribute to this emerging field through innovative artifacts. Working software acts as an important vehicle for this kind of research as it embodies research outputs and allows for the investigation of their behavior and appropriateness within real-world systems. In addition to traditional dissemination channels, open source software provides additional visibility and – to a certain degree – reproducibility of the research conducted.

References

Abrahamsson P, Oza N, Siponen M (2010) Agile software development methods: A comparative review. In: Dingsøyr,T., Dybå T, Moe NB (eds) Agile Software Development. Springer, pp 31–59

Alberer G, Alberer P, Enzi T, Ernst G, Mayrhofer K, Neumann G, Rieder R, Simon B (2003) The Learn@WU Learning Environment. In: Proc Wirtschaftsinformatik 2003, Dresden

Apel S, Kästner C (2009) An overview of feature-oriented software development. Journal of Object Technology, 8(5):49–84

Allen BR, Boynton AC (1991) Information architecture: In search of efficient flexibility. MIS quarterly, 15(4):435–445

Andergassen M, Guerra V, Ledermüller K, Neumann G (2012) Browser-based mobile clickers: Implementation and challenges. In: Proc IADIS International Conference - Mobile Learning 2012, Berlin

Bjerknes G, Bratteteig T, Espeseth T (1991) Evolution of finished computer systems. Scandinavian Journal of Information Systems, 3:25–45

Breu R, Agreiter B, Farwick M, Felderer M, Hafner M, Innerhofer-Oberperfler F (2011) Living models – Ten principles for change-driven software engineering. Int. J. Software and Informatics, 5(12):267–290

Brynjolfsson E, Hitt LM, Kim HH (2011) Strength in Numbers: How Does Data-Driven Decision-making Affect Firm Performance? SSRN working paper, Social Science Research Network (SSRN)

Callaù O, Robbes R, Tanter E, Röthlisberger D (2011) How (and why) developers use the dynamic features of programming languages: the case of Smalltalk. Empirical Software Engineering, pp 1–39

Clements P, Northrop L (2007) Software Product Lines: Practices and Patterns. Addison-Wesley

Fowler M (2010) Domain-specific languages. Pearson Education

Frank U (2006) Towards a pluralistic conception of research methods in information systems research. ICB Research Reports 7, University Duisburg-Essen, Institute for Computer Science and Business Information Systems

Georgakopoulos D, Hornick M, Sheth A (1995) An overview of workflow management: from process modeling to workflow automation infrastructure. Distributed and parallel Databases, 3(2):119–153

Germonprez M, Hovorka D, Collopy F (2007) A theory of tailorable technology design. J. AIS, 8(6):351–367

Germonprez M, Hovorka D, Gal U (2011) Secondary design: A case of behavioral design science research. J. AIS 12(10):662-683

Hansen HR, Neumann G (2009) Wirtschaftsinformatik 1, Grundlagen und Anwendungen, Band 1, 10. Auflage. Lucius & Lucius, UTB 2669, ISBN 978-3-8282-0479-9

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS quarterly, 28(1):75–105

Hoyer V, Gilles F, Janner T, Stanoevska-Slabeva K (2009) SAP research RoofTop marketplace: Putting a face on service-oriented architectures. In: 2009 World Conference on Services - I, pp 107–114

Kieser A, Kubicek H (1992) Organisation, 3. Auflage, de Gruyter, Berlin

Malone TW, Crowston K (1994) The interdisciplinary study of coordination. ACM Comput. Surv. 26(1):87–119

Malone TW, Crowston K, Lee J, Pentland B, Dellarocas C, Wyner G, Quimby J, Osborn CS, Bernstein A, Herman G, Klein M, O'Donnell E (1999) Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. Management Science, 45(3):425–443

March ST, Smith GF (1995) Design and natural science research on information technology. Decision Support Systems, 15(4):251–266

Mernik M, Heering J, Sloane AM (2005) When and how to develop domain-specific languages. ACM Comput. Surv. 37(4):316-344

Mödritscher F, Andergassen M, Neumann G (2013) Dependencies between e-learning usage patterns and learning results. In: Proc i-KNOW 2013, 13th International Conference on Knowledge Management and Knowledge Technologies, Graz

Muller MJ, Kuhn S (1993) Participatory design. Commun. ACM, 36(6):24–28

Neumann G, Erol S (2008) From a social wiki to a social workflow system. In: Ardegna D, Mecella M, Yang J (eds) Business Process Management Workshops 2008 LNBIP 17, Milan, pp 698–708

Orlikowski WJ, Iacono CS (2001) Research commentary: Desperately seeking the "IT" in IT Research – A call to theorizing the IT artifact. Information Systems Research, 12(2):121–134

Österle H, Otto B (2010) Consortium Research: A Method for Researcher-Practitioner Collaboration in Design-Oriented IS Research. Business & Information Systems Engineering, 2(5):283–293

Ousterhout JK (1998) Scripting: Higher level programming for the 21st century. IEEE Computer, 31(3):23–30

Pahlke I, Beck R, Wolf M (2010) Enterprise mashup systems as platform for situational applications. Business & Information Systems Engineering, 2(5):305–315

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. Journal of Management Information Systems, 24:45–77

Rank S (2002) A Reflective Architecture to Support Dynamic Software Evolution. PhD thesis, University of Durham, UK

Sein MK, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. MIS Quarterly, 35(1):37–56

Siemens G, Long P (2011) Penetrating the Fog: Analytics in Learning and Education. EDUCAUSE Review, 46(5):30–32

Stahl T, Voelter M, Czarnecki K (2006) Model-Driven Software Development: Technology, Engineering, Management. John Wiley & Sons

Strembeck M, Zdun U (2009) An Approach for the Systematic Development of Domain-Specific Languages. Softw., Pract. Exper., 39(15):1253–1292

Vaishnavi VK, Kuechler W (2007) Design Science Research Methods and Patterns: Innovating Information and Communication Technology. Auerbach Publications

Van der Aalst WMP (2011) Process Mining – Discovery, Conformance and Enhancement of Business Processes. Springer

Von Hippel E (2009) Democratizing innovation: The evolving phenomenon of user innovation. International Journal of Innovation Science, 1(1):29–40

Wulf V, Rohde M (1995) Towards an integrated organization and technology development. In Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques, ACM, pp 55-64

Wulf V (1999) Evolving cooperation when introducing groupware: a self-organization perspective. Cybernetics & Human Knowing, 6(2):55-74