IS Design Principles for Empowering Domain Experts in Innovation: Findings From Three Case Studies

Research-in-Progress

Sven-Volker Rehm
WHU – Otto Beisheim School of Management
Burgplatz 2, 56179 Vallendar, Germany
sven-volker.rehm@whu.edu

Thomas Reschenhofer
Institute for Informatics,
Technische Universität München
Boltzmannstraße 3, D-85748 Garching, Germany
thomas.reschenhofer@tum.de

Klym Shumaiev
Institute for Informatics,
Technische Universität München
Boltzmannstraße 3, D-85748 Garching, Germany
klym.shumaiev@tum.de

Abstract

Today a significant part of innovation activities in firms is carried out within innovation networks of cooperating enterprises. In such networks, one key challenge is to provide software that enables to systematically share and adaptively integrate knowledge between the partners' domains of expertise. One potential answer is to apply application software that allows for end-user or domain expert configuration. We provide preliminary empiric evidence from a field test of an expert-configurable collaborative information system in three innovation networks. In a three-year qualitative study, we have identified challenges to software support originating from knowledge, methodical and relational diversity in the networks. We formulate design challenges and design principles relevant for developing and applying domain expert-configurable software. We provide insights into the significance of related user roles in cooperative innovation projects, and offer the role of 'facilitators' as mediating agents in application configuration.

Keywords: Domain expert configuration, design science, innovation networks, domain knowledge, design principles, case study

Introduction

A significant part of the R&D activities of small and medium enterprises (SMEs) today is being carried out within inter-organizational partnerships (Davenport, Leibold, and Voelpel, 2006). Such partnerships are often established within the firms' business ecosystems, through innovation networks, which can be considered as temporary constellations of partnering firms (Moore, 1993; Cowan, Jonard, and Zimmermann, 2007; Adner and Kapoor, 2010). Such networks engage in cooperative New Product Development (NPD) projects that intend to develop distinct new products, processes and services. Finding appropriate arrangements and integrating between partners in such networks have been emphasized by previous literature as crucial for cooperation success (Schilling and Phelps, 2007; Word, 2009; Borgh, Cloodt, and Romme, 2012).

Information technology (IT) plays a key role in these networks as the various IT systems, tools and software applications involved provide the underlying infrastructure of cooperative activities and act as integrators and facilitators (Nambisan, 2003; 2013). At the same time, managing this infrastructure opens up a paramount challenge to the planning, implementation, uptake, and management of IT-based networked relationships, and can be considered as a comprehensive *IT project* in itself, aiming to build the 'IT base of innovation networks'.

In this context, Information Systems (IS) literature suggests various *types* of IS to support NPD, e.g. project management systems, resource management systems, knowledge management systems, cooperative work systems, collaboration software for product design and development, or Product Lifecycle Management (PLM) systems (cmp. Pavlou and El Sawy, 2006; Banker, Bardhan, and Asdemir, 2006; Nambisan, 2013). However, there is a gap in literature that relates to the question how firms can adequately align, integrate or cooperatively develop their cooperative activities' IT base within an innovation network.

One particular aspect in the management of the IT base of innovation networks is the provision of software applications and services that operatively support cooperative work throughout the innovation process. In innovation networks usually various IT systems, tools and software applications are used in parallel, because involved firms bring into the project their particular IT-related requirements and challenges for exchanging and processing information and knowledge, which they largely try to cover by deploying firm-owned tools. When innovation networks target radical innovations however, a high level of exchange and integration of information and knowledge between cooperating firms is required. In such contexts, often NPD project team members from different technical and business backgrounds need to cooperate (Enkel and Gassmann, 2010). Their diverse domains of expert knowledge need to be integrated through systematic innovation procedures (Scozzi, Garavelli, and Crowston, 2005; Hidalgo and Albors, 2008). This is a challenge to software applications that enable handling of expert knowledge, as well as systematically sharing and integrating it within cooperative work environments (Ram and Ram, 1996). As a result, software engineering encounters the problem that the specific needs of involved experts cannot be anticipated, and consequentially, adequate software functionalities cannot be held available.

In this context, one approach currently used in software engineering is to develop application software that allows for *domain expert configuration* (Mørch et al., 2004). This method of software design and deployment implies that the original "users" – as domain experts – or dedicated "facilitators" adopting a consulting role, take over the task to configure the application software for their specific purposes and business or technical contexts (Boren and Ramey, 2000; Følstad, 2007). This approach holds a promise for application in innovation networks, because it might allow to answer the peculiar requirements of involved specialists and at the same time enables cooperating domain experts to integrate their diverse backgrounds into one software application "on-the-fly". However, studies on the success of this approach in connection to preparing the IT base in innovation networks to date are lacking.

We have conducted case studies of three innovation networks which we followed over a three years period. The involved innovation networks, consisting of three to seven SMEs each, used a shared, wiki-based collaborative information system providing various functionalities that enabled domain expert configuration. As an intense field test we have observed involved domain experts adopting several roles in utilizing the system for conceptualizing, configuring and using applications in context of their cooperative NPD projects. Our research intends to inform system design literature on critical factors (or features) that are relevant to the design of application software that uses domain expert configuration. We provide evidence from operative use of the developed applications within several NPD projects. Our analysis explores *design principles* which are relevant for developing and providing information systems and services that allow to successfully apply domain expert configurable software in cooperative NPD project settings.

Following a design science approach (March and Smith, 1995; Hevner et al., 2004), in this paper, we first give an outline of the overall problem scope we have identified in relation to supporting NPD teams through domain expert-configured collaborative software applications. We then describe the main characteristics of the collaborative IS, in order to link the identified challenges for software support with our solution strategies and features of the system as design artefact. We provide evidence from qualitative case studies observing how the NPD teams made use of the system and available functionalities for configuring own applications.

From our study we can identify several essential design characteristics, and we propose an early draft of a comprehensive framework containing *design principles for domain expert configuration*. We can also formulate some inferences regarding the process of adopting domain-expert configurable software on firm and network levels. We identify the role of facilitators as central to software design, application development, and adoption processes. We can infer that facilitators provide principal services in scope of generation, provision and use of domain expert configured software applications, which opens up new opportunities for creating business models for IS development.

Domain Expert Configuration

In software engineering, the idea of "domain expert configuration" can be positioned within the larger disciplinary endeavor to construct IT systems that provide the flexibility to be continuously redesigned and adapted to changing contexts. Such systems rely on component-based technologies, which are easier and more rapidly adaptable to frequently changing requirements and can be used over a longer time period than conventional ones (Vitharana, 2003; Mørch et al., 2004). This creates an economically interesting effect because their overall costs, i.e., accumulated over the whole life-cycle, are lower than the costs of comparable systems that are not adaptable. Also the effort connected to development, implementation, deployment and maintenance of such systems is comparatively lower. Reduced efforts and costs coincide with increased effectiveness due to a better fit of the functionality required by the use case setting and the functionality provided by the system (Spahn, Dörner, and Wulf, 2008). Further benefits have been identified related to easier adoption of re-designable systems by involved domain experts as users, because the users become an essential part of the development process and attain vital influence on tailoring the system to their needs (Wulf and Jarke, 2004).

Thus, regarding the development of re-designable systems, integrating domain experts into the design and development process has early been identified in literature as a core issue (Wulf and Rohde, 1995). The software development process then needs to be conceived as an iterative cycle, which allows for evolutionary and participative improvement of software systems. However, high complexity of adaptation mechanisms might prevent domain experts from configuring the software systems on their own (Spahn et al., 2008). Analyzing the complexity of related adaptation mechanisms on technical level is thus a prerequisite to identify the ability, or proneness, of a software system to be tailored by domain experts (MacLean et al., 1990). In this respect, Costabile et al. (2003) identify two main motivations for domain expert development: Firstly, *user creativity* involves that the users may devise novel ways to exploit the system in order to satisfy some needs not considered in the specification and design phase; secondly, *user acquired habits*, which relates to the aspect that users may follow some interaction strategy or routines to which they are (or become) accustomed. Both motivations serve as starting points to create synergies through domain expert configuration approaches, and as well cause design strategies that need to be facilitated by adequate design characteristics (Matthes, Neubert, and Steinhoff, 2011).

Such design characteristics embrace two basic classes of *domain expert activities* (Costabile et al., 2003): Class 1 includes activities that allow users, by setting some parameters, to choose among alternative behaviors (or presentations or interaction mechanisms) already available in the application; such activities usually include parameterization, customization, personalization, or annotation. Class 2 includes activities that imply programming, including any programming paradigm such as programming by demonstration, programming with examples, visual programming, macro generation etc., thus creating or modifying a software artefact. The artefact we are testing through our case studies offers to use both classes of activities, which are however intended to serve needs of different types of users.

Design Principles in Domain Expert Configuration

In system design literature, *principles* are understood as high-level concepts describing characteristics of the system and its design. Gould and Lewis (1985) describe principles which focus on the usability aspect of systems, while Bahill and Botta (2008) propose principles for good system design more generally without focusing on a certain aspect. For a specific case, Gulliksen et al. (2003) name and describe principles for the user-centered design of information systems (UCSD). Their UCSD principles are principally related to the design process, i.e. to the behavior of "designers". The design principles we outline in this article refer to the designed system and its features as IS artifact, because our focus is on challenges that arise from the innovation process, and that are answered through specific features of the applied system. This way we are able to describe design challenges and principles largely independent from the behavior, perception and understanding of the "designers" of the system.

The idea of domain expert configuration can be seen as embedded in the larger context of End-User Development (EUD). Soriano et al. (2007) derive principles for what they term 'mashup-enabled enterprise 2.0 collaboration architecture'. Mashups are understood as applications which are composited of various data sources, gadgets and services by domain experts in an agile development process. The resulting architectures are hence conceived to support the end-user driven and collaborative development of highly flexible and adaptable enterprise applications ("2.0") by combining pre-defined data-oriented components to a custom enterprise mashup. Their principles however are mainly addressing cultural aspects of enterprises such as fostering user participation and community-based collaboration. Lizcano et al. (2011) outline EUD principles as factors in composite web development environments, e.g., end-user empowerment. While they already state the relevance of enhancing traditional end-user interaction and name a couple of related principles, their work lacks further illustration of these principles as well as a description of which concrete challenges these principles actually address.

Our work contrasts extant literature with respect to its focus on studying and presenting principles for information systems supporting DEC in innovation networks. Thereby, these principles address concrete software engineering-related challenges, whereas the context of an innovation network including the collaboration of multiple firms leads to additional challenges (e.g., integration of innovation processes).

Research Question and Method

The research team has conducted an intense field test of a shared, wiki-based collaborative information system (CIS) enabling domain expert configuration, in three innovation networks. We want to know which requirements the cooperative innovation processes impose on knowledge exchanges, and to what extent these requirements have been answered through the applied CIS. Our study intends (a) to identify design challenges for domain expert configuration (DEC) that arise in scope of cooperative NPD project settings, and (b) to explore design principles (features) of DEC-enabled application software that answer to the observed challenges. In particular we want to classify observed challenges in order to make them better manageable, and to formulate insights for improving the functional scope of application software which is intended for DEC in the long term.

The three case studies are part of the relevance cycle in design science research (Hevner et al., 2004; Hevner and Chatterjee, 2010:20). During three years, we have observed the utilization of the system's features and the use of domain expert configured applications vis-à-vis the progressing innovation processes, assuming a qualitative and interpretative approach (Yin, 2003; Stake, 2005). Our data collection includes more than 50 episodic interviews on the progress of activities (Denzin and Lincoln, 2005; Myers and Newman, 2007), participation to more than 50 work meetings, feedback from domain expert developers in their different roles and from the software provider about the developers' inquiries; and observations on the use of IT systems in operative work situations.

Each of the networks pursued a radical innovation project, comprising the necessity to newly develop or re-engineer involved materials, associated production processes, machinery and related services. The networks were all comparatively diverse in terms of partner composition and involved industrial sectors. Due to the high heterogeneity of the expertise domains of NPD team members, the cases provided a good choice for our study. By paralleling the behavior of involved experts, and the resulting applications, we could recognize differences that originated from individual or firm contexts, or from the specific technical contents the teams were dealing with.

During the field test, the research team closely monitored CIS use – and partly participated in – the application configuration processes. Data analysis started with an identification and characterization of the network teams' activities related to CIS use. We tried to understand how the teams adopted and used the CIS, e.g. as repository or for knowledge exchanges or to support concurrent engineering processes. We also tried to get an impression of the quantity and quality of pages and other artifacts created in relation to the progressing innovation process, and of their specific features, e.g. the type of created wiki pages. We classified these activities with respect to the challenges they imposed on CIS use and functionalities. This involves reflecting on why the team used certain functionalities (requirements from the innovation process), and how successful the implemented solutions were (requirements for DEC and CIS). The latter aspect included also the awareness created towards the use of shared CIS and DEC in general. From a method perspective, these classifications serve as measures to evaluate CIS use and DEC processes (Hevner et al., 2004).

Data collection with regard to applications and other artifacts that have been created in the system is still ongoing. We also continue data analysis with regard to knowledge exchange requirements and system use. From our analysis so far we arrive at three characteristic facets cooperative NPD in innovation networks imposes on CIS design. The data collected from three networks allows for a juxtaposition of specific implementations and our interpretation of the observed processes. We perceive several arising design challenges as specific to our study context. They bring about exceptional requirements for CIS, and we consequently identify related design principles. Overall goal of our study is to formulate these design principles, and hence, to advance system design by further improving on these design principles and emphasizing them to domain experts and users.

Case Studies of Three Innovation Networks

The networks conducted NPD projects targeting several new products and services: Network 1, a motorbike helmet, incorporating a novel dampening material, including partners from automotive, chemical, engineering and consulting sectors; Network 2, a cardiovascular stentgraft, as a medical device incorporating new functional materials, featuring partners from healthcare, technical textiles, textile processing, textile machinery, physics, engineering, consulting sectors; Network 3, new textile coating processes and services (chemical, textile, home textiles, furniture, consulting, engineering sectors).

The partners had not worked together in these networks before, and faced strong challenges to align each other in terms of partnerships, their value contributions, and innovation processes. In this scope, they made extensive use of collaborative IT systems. The CIS providing the core of the IT base, was supplemented by further tools for business process modeling and communication (telephony, chatting). The partners in all three cases utilized further dedicated IT systems and software applications for their specific tasks.

The critical role of the CIS was, first, to provide wiki-based editing and basic project management functionalities, and second, to enable the configuration of customized information structures – including models, data types and role definitions – meant to build the basis for distinct *innovation procedures 'apps'* allowing to coordinate cooperative business processes. Innovation procedures, e.g. Quality Function Deployment (QFD), typically ensure stringent integration of diverse contributions, i.e. from diverse expertise domains, within processes producing intermediate results for defined single steps of an innovation process. QFD for instance allows to translate customer requirements into product or process design characteristics (Akao, 1990). Software products supporting execution of such procedures only scarcely exists; in our field test thus the networks attempted to implement several procedures by configuring corresponding apps that were tailored to their needs.

Design Characteristics of the Collaborative Information System as Tested Artefact

In order to address this purpose, the networks used a shared collaborative information system (CIS). CIS is a wiki-based information system on Java basis. CIS design involves an overarching *information model* comprising *information objects* which for instance can be represented by wiki pages (see Figure 1). However, while common wiki pages contain only unstructured content, CIS implements a "hybrid" wiki concept enabling the collaborative and iterative enrichment of wiki pages with semantic structures. Thereby, users can define schemes, or meta models, by specifying types (Task, Project etc.) which consist of attributes and relations (assigned to, start date etc.). The types can be specifically connected to wiki pages so that they reproduce a particular scheme.

The associated CIS Expression Language (CxL) enables domain expert-driven definition and configuration of apps based on CIS's meta model, information model and the existing content. For example, if the meta model defines a type *Task* that contains an attribute *Status*, a domain expert developer can specify an app for displaying all tasks, in which color-coding highlights each task's status. To do so, the developer defines a CxL expression referring to the tuple *Task/Status* which generates an HTML-based visualization based on the respective information model. For defining expressions, the meta model is used, while expression execution is based on the information model and existing content. In this sense and as described previously, CIS refers to domain-expert activities of Class 2 (Costabile et al., 2003) since it enables the development of data-oriented apps by programming.

Moreover, in CIS multiple *user roles* can be defined, comprising individual access rights to information objects (wiki pages), meta model elements, or apps.

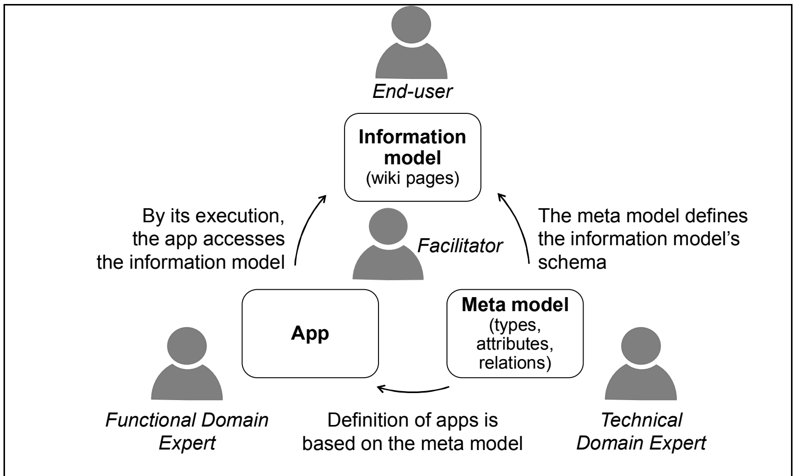

Figure 1. Conceptual Model of CIS and User Roles.

Analysis: Domain Expert Configuration (DEC) in Three NPD Projects

In our case studies, various roles facilitated deployment of CIS. More than 100 expert professionals involved into cooperative NPD tasks of the innovation process e.g. engineers and researchers, were *End-users* of CIS and the created apps. Some of them, e.g. project managers or consultants, served as *Functional Domain Experts* supporting conceptualizing innovation procedures and configuring them as apps. They provided the specialist background knowledge to adapt generic innovation methods to the networks' contexts and conditions. They were supported by *Technical Domain Experts* who configured the meta model. App configuration followed an agile approach, with Functional Domain Experts being in close contact or on-site with end-users, while receiving monthly feedback from sprint meetings with Technical Domain Experts. While the end-users could continuously use the CIS system, app development phases comprised daily to weekly, and meta model changes monthly cycles. As a fourth role, *Facilitators* regularly evaluated the created apps from business, functional and technical as well as user interaction perspectives, and proposed further tailoring steps. Facilitator roles would be assumed by proficient IT users such as researchers or skilled expert professionals.

Characteristic Facets of Cooperative NPD in Innovation Networks

Studying the three networks, we observed that the context of cooperative NPD in innovation networks shows three principal, characteristic facets, each posing particular challenges to IT support of the networking partners:

(A) *Knowledge diversity*: The diversity of expertise and knowledge meeting in such cooperations is high compared to other work environments. The members of the respective NPD project teams need to express their contributions, generally by writing in a shared work space, and need to find appropriate ways to align their individual views to the overall problem scope. With relation to CIS use, a team member in one of the networks formulates:

> *"The principle problem which arises in this kind of knowledge repository is how to structure it. Which is not simple and it will be simple if only one person had to look at it. But different persons especially with different knowledge backgrounds, with different mental approaches, and so on, will have or could have, some problems in finding the right information on the site." (Process Engineer at Innovation Services Provider, Network 1)*

A process resulted that establishes the generation of shared views and knowledge structures which need to be represented with help of IT, as they are the basis to gather data and information, and need to be stored and accessible for future tasks in the innovation process. In this respect, team members as domain experts need to discuss about existing contents in order to share meanings and different views, which contributes to the creation of new knowledge (structures):

> *"I went in the system, I found [the diagram] and we discussed a little bit about possible issues, advantages, how to read the diagram and so on. Just being over there, both on the same page. For these kind of activities I think [the CIS] is really useful." (Process Engineer at Innovation Services Provider, Network 1)*

The CIS was configured by the team of Network 1 in a way that it formed a shared point of reference. Their structuring oriented at the challenges they faced, e.g. product design, an explanation of a critical prototype testing procedure, and a collection of information and articles about their target markets. This way, the team was able to integrate their diverse perspectives and create separate "building blocks" for future tasks.

(B) *Methodical diversity*: Due to the disruptive nature of the targeted innovations in our cases, the NPD teams had to cope with different "logics", e.g. when combining chemical processes to production process and product design parameters; they consequently applied diverse systematic approaches to arrive at viable solutions for their tasks. The data and knowledge created through such approaches provide the input for further developments and thus need to be available for future activities, too. From a software engineering perspective, this means that on the one hand, newly created structures need to be re-usable, but also need to be merged with each other for different purposes ("logics") on the other hand. On one occasion, a network entered into a new NPD project, but was still able to rely on the previously created workspace:

> *"…, we can totally use all information that we have now. […]. Yes, we will use it and already we started to work with another customer to develop some tests to see where we can go…" (Textile Engineer at Interior Textiles Producer, Network 3)*

In the following, the CIS feature to prepare blueprints for future processes, e.g. to start a new project, was well adopted in the respective network.

(C) *Relational diversity*: The diverse backgrounds of people involved to the teams caused varying needs with respect to documenting, retrieving and representing information, interacting with each other, and utilizing the software. An involved researcher speaks about encountered problems in this respect:

> *"Usually we encounter problems related to misunderstandings concerning basic definitions, for example, restoring force or radial force… So, in [the CIS] we should see all basic aspects, definitions, key parameters…" (Material Researcher at Research Services Provider, Network 2)*

The case study users brought in different skills regarding software use, and different roles could be assumed during the app configuration processes. On firm level, each partner firm had access to the shared CIS for their respective project spaces, and was supported by facilitators regularly discussing requests, wishes, and experiences.

Design Challenges and Principles

Within these characteristic facets of the innovation network context, in our analysis we have identified a number of challenges to software support. We have measured the extent these challenges were answered through design features of CIS. Going back and forth between analyzing for challenges and design features we formulated a number of design principles arising in our study context. Table 1 provides an overview and a short description of challenges and principles.

The role of facilitators emerged as a significant finding from our study. In particular for context alignment (facet C), and the related generation of templates, facilitators provided active stimulus to the teams. On one occasion, a systematic innovation procedure is initiated together with an NPD team. After meeting in person, and preparing an initial questionnaire on the procedure, the structures are captured as templates in CIS.

> *"So he suggested, '...to try the Product Potential Analysis method'. And then... he sent a questionnaire to the two companies, I don't know, let's say a week before we had a meeting. And then we had a meeting, with him, with the three (network) partners. And then we went over the filled out questionnaires and he explained a bit to the companies, and they explained a bit their answers, so that was really a useful day." (Textile Engineer, Textile Engineering Service Provider, Network 3)*

This incidence showed that – with reference to the two basic classes of domain expert activities – facilitators were able to mediate between Class 1 and Class 2 activities. In this respect they were able to close the gap between "principally" existing design features of CIS, and the actually required (but not necessarily exercised) options available for technical domain experts (and end-users).

Design Challenge (exemplary)	**Design Principle (Facet)**
During the innovation process, knowledge structures are evolving; emerging from unstructured to structured contents.	**Iterative Modeling (A)**: Semantic structures link structured and unstructured content in the "hybrid" wiki approach.
Content generated might change in context e.g. due to deployment in different projects, or be considered from different perspectives.	**Content-Context-Relation (A)**: The information model (IM) and meta models (MM) can be adaptively defined, and included to apps.
Data and information generated during specific activities provide input to future tasks.	**Information Re-use (B)**: MM allows integrating Information objects (IO) in different apps.
Innovation procedures that are already modelled might serve as blueprints for future tasks.	**Function Re-use (B)**: App 'logics' defined through MM and IM can be reused as templates (Generalization) and altered (Customization).
Existing knowledge structures at times need to be combined, e.g. extended, or united. The innovation procedures need to adapt respectively.	**Merging/Splitting (B)**: The meta model allows adaption of apps to integrate new Information Models, or to separate MM into sub-parts.*
The alignment of innovation procedures and the concurrent development of supporting software demands integration of various sorts of expertise.	**Context Alignment (C)**: Facilitators mediate in the alignment process. They are supported by context-dependent evaluation techniques (e.g. field error visualization).
For coping with diverse user needs, best practices are required for GUI design that support facilitators.	**Configuration templates (C)**: Schemes and template processes for app configuration.*
Ownership of knowledge is critical in innovation projects. Contributions need to be retraced, and availability of data ensured after project end.	**Ownership (C/A)**: Definition of access rights and roles are part of the meta model.

* (not provided by CIS in the use cases)

Table 1. Design Challenges and Principles Arising for DEC in Innovation Networks.

Discussion and Future Research

We base our categorization of design challenges and principles on three facets arising from cooperative NPD projects in innovation networks and their related innovation processes. These three facets, knowledge, methodical and relational diversity, originate from our observations and interpretations. Further work is needed to theoretically frame them in order to better understand to what extent design (and design research) can answer to requirements stemming from innovation processes. We believe that in depth analyses of the related knowledge exchange processes, e.g. with a focus on the creation of transactive memory (Wegner, 1987) can contribute to this.

The CIS that was applied in our case study represents a particular kind of inter-organizational information system (IOS). The question how IOS can be designed – by drawing on theory-based artifact design – has only occasionally been discussed in IS literature. Recent findings on adoption of IOS, their impact on governance of inter-organizational relationships, and their consequences on organizing however promise further interesting avenues for improvement of cooperative innovation activities (see e.g. Robey, Im, and Wareham, 2008). These need to be further discussed in view of the availability of DEC-enabled application software.

Our preliminary findings are only a first step to better understanding the management of the IT base of innovation networks. In our case setting, various apps were created that solved specific problems occurring in the innovation process. From our observations it seems plausible to orient the IT management for innovation networks increasingly at the knowledge exchange processes of domain experts. Answering the experts' knowledge requirements that originate from differing backgrounds and interdependencies between tasks can be put to a considerable extent in the experts' own hands. In this respect, in our case study we have seen how systematic innovation procedures translated into apps through DEC can resolve critical issues in an efficient and effective manner, without needing to rely on highly specific software which has to be purchased separately. Considering this particular significance of knowledge more closely might offer new approaches to IT management in the cooperative innovation context.

Findings and Conclusion

In scope of a design science research approach, we have carried out an intense field test of a software artefact, CIS, in the context of three innovation networks as case studies. Our goal was to identify and explore design principles for domain expert configurable software which are relevant in scope of cooperative NPD project settings. Starting from characteristic facets of innovation, we have formulated challenges and principles for design of the domain expert configurable CIS.

The design principles are specific to the domain expert configuration (DEC) context, because only domain experts involved to the innovation process are able to identify the needs for changes in the software support during the progressing, evolving project – that in turn address the identified design principles. As we have observed in our case studies, these principles allow building on the factors of user creativity and developing routines, which have been identified by previous literature as motivators of using DEC.

We provide empiric evidence from an intense field test of a software artefact (CIS) in the context of innovation networks. This is significant because it opens up a new avenue for IS research on supporting innovation processes. Our interpretation suggest that DEC is a viable way to allow for high-level support of expert professionals involved into innovation and at the same time provide sustainable IT system (as in our case the CIS), which are sufficiently flexible to adapt to the dynamic contexts and processes of innovation. The design principles we have identified are a first step towards delimiting the architecture and functionalities of such systems.

The empiric findings gathered during our field test could provide insights to the design science research process. Testing artefacts – in our case a marketed software product – within specific settings might hold surprising insights that stimulate improvements beyond the artefact's pre-assumed scope.

In the role of *facilitators* we have experienced their particular contribution as mediating agents between different requirements, expectations, and methodological 'logical' necessities arising in the innovation context. In our case studies, these facilitators were key actors supporting the adoption of CIS by the NPD team members. As a matter of fact, consequently to the field test, the software provider of CIS has altered its business model; application development is switched from IT experts in favor of providing support to facilitators. We offer the "facilitator" role to literature on end-user development and domain expert configuration – but also to literature on IS support for innovation – as mediating agents balancing, and matching complementary aspects, between knowledge, methodical and relational diversity. To what extent this role can be fulfilled by human agents or by software is still an open question to be tackled by future research.

Acknowledgements

This work has been partly funded by the European Commission through the Project SmartNets: The Transformation from Collaborative Knowledge Exploration Networks into Cross Sectoral and Service Oriented Integrated Value Systems (Grant Agreement No. 262806). The authors wish to acknowledge the Commission and all the SmartNets Project partners for their support.

References

Adner, R., and Kapoor, R. 2010. “Value creation in innovation ecosystems: how the structure of technological interdependence affects firm performance in new technology generations,” *Strategic Management Journal* (31:3), pp. 306–333.

Akao, Y. 1990. *Quality Function Deployment: Integrating Customer Requirements into Product Design*, Cambridge, Mass: Productivity Press.

Bahill, A.T., and Botta, R. 2008 "Fundamental principles of good system design," *Engineering Management Journal. Citeseer.*

Banker, R. D., Bardhan, I., and Asdemir, O. 2006. “Understanding the Impact of Collaboration Software on Product Design and Development,” *Information Systems Research* (17:4), pp. 352–373.

Boren, M. Ted, and Ramey, J. 2000. “Thinking Aloud: Reconciling Theory and Practice,” *IEEE Transactions on Professional Communication* (43:3), p. 261.

Borgh, M., Cloodt, M., and Romme, A. Georges L. 2012. “Value creation by knowledge-based ecosystems: evidence from a field study,” *R&D Management* (42:2), pp. 150–169.

Costabile, M.F., Fogli, D., Letondal, C., Mussio, P., Piccinno, A. 2003. “Domain-Expert Users and their Needs of Software Development”, *Proceedings of the UAHCI Conference*, Crete, June 22-27 2003, pp. 232-236.

Cowan, R., Jonard, N., and Zimmermann, J.-B. 2007. “Bilateral collaboration and the emergence of innovation networks: Bilateral collaboration and the emergence of innovation networks,” *Management Science* (53:7), pp. 1051-1067.

Davenport, T. H., Leibold, M., and Voelpel, S. 2006. *Strategic management in the innovation economy: Strategy approaches and tools for dynamic innovation capabilities*, Erlangen: Publicis.

Denzin, N. K., and Lincoln, Y. S. (eds.) 2005. *The SAGE handbook of qualitative research*, Thousand Oaks: Sage Publications.

Enkel, E., and Gassmann, O. 2010. “Creative imitation: exploring the case of cross-industry innovation,” *R&D Management* (40:3), pp. 256–270.

Følstad, A. 2007. “Work-Domain Experts as Evaluators: Usability Inspection of Domain-Specific Work-Support Systems,” *International Journal of Human-Computer Interaction* (22:3), pp. 217–245.

Gould, J. D., and Lewis, L. 1985 "Designing for usability: key principles and what designers think." *Communications of the ACM*, pp. 300-311.

Gulliksen, J.; Göransson, B., Boivie, I., Blomkvist, S., Persson, J., and Cajander, Å. 2003. “Key Principles for User-Centred Systems Design,” *Behaviour and Information Technology* (6), pp. 397–409.

Hevner, A. R., and Chatterjee, S. 2010. *Design research in information systems: Theory and practice*, New York, London: Springer.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. “Design Science in Information Systems Research,” *MIS Quarterly* (28:1), pp. 75–105.

Hidalgo, A., and Albors, J. 2008. “Innovation management techniques and tools: a review from theory and practice,” *R&D Management* (38:2), pp. 113–127.

Lizcano, D., Alonso, F., Soriano, J., and Lopez, G. 2011. “End-User Development Success Factors and their Application to Composite Web Development Environments” *Proceedings of the International Conference on Systems*.

MacLean, A., Carter, K., Lövstrand, L. and Moran, T. 1990. “User-tailorable systems: pressing the issues with buttons,” *Proceedings of the CHI '90*, ACM Press, pp. 175 - 182.

March, S. T., and Smith, G. F. 1995. “Design and natural science research on information technology,” *Decision Support Systems* (15:4), pp. 251–266.

Matthes, F., Neubert, C., and Steinhoff, A. 2011. “Hybrid Wikis: Empowering Users to Collaboratively Structure Information,” *Proceedings of the 6th International Conference on Software and Data Technologies (ICSOFT)*, Seville, pp. 250-259.

Moore, J. F. 1993. “Predators and Prey: A New Ecology of Competition,” *Harvard Business Review* (71:3), pp. 75–86.

Mørch, A. I., Stevens, G., Won, M., Klann, M., Dittrich, Y., and Wulf, V. 2004. “Component-based technologies for end-user development,” *Communications of the ACM* (47:9), p. 59.

Myers, M. D., and Newman, M. 2007. “The qualitative interview in IS research: Examining the craft,” *Information and Organization* (17:1), pp. 2–26.

Nambisan, S. 2003. “Information systems as a reference discipline for new product development,” *MIS Quarterly* (27:1), pp. 1–18.

Nambisan, S. 2013. "Information Technology and Product/Service Innovation: A Brief Assessment and Some Suggestions for Future Research," *Journal of the Association for Information Systems* (14:Special Issue April 2013), pp. 215–226.

Pavlou, P. A., and El Sawy, O. A. 2006. "From IT Leveraging Competence to Competitive Advantage in Turbulent Environments: The Case of New Product Development," *Information Systems Research* (17:3), pp. 198–227.

Ram, S., and Ram, S. 1996. "Validation of Expert Systems for Innovation Management: Issues, Methodology, and Empirical Assessment," *Journal of Product Innovation Management* (13:1), pp. 53-68.

Robey, D., Im, G., and Wareham, J. D. 2008. "Theoretical Foundations of Empirical Research on Interorganizational Systems: Assessing Past Contributions and Guiding Future Directions," *Journal of the Association for Information Systems* (9:9), pp. 497–518.

Schilling, M. A., and Phelps, C. C. 2007. "Interfirm collaboration networks: The impact of large-scale network structure on firm innovation," *Management Science* (53:7), pp. 1113-1126.

Scozzi, B., Garavelli, C., and Crowston, K. 2005. "Methods for modeling and supporting innovation processes in SMEs," *European Journal of Innovation Management* (8:1), pp. 120–137.

Soriano, J., Lizcano, D., Cañas, M. A., Reyes, M., and Hierro, J. J. 2007. "Fostering Innovation in a Mashup-Oriented Enterprise 2.0 Collaboration Environment" *UK, sai: sisn* (24), pp. 62–68.

Spahn, M., Dörner, C., Wulf, V., 2008. "End user development: approaches towards a flexible software design," *Proceedings of the 16th European Conference on Information Systems (ECIS)*, Galway, Ireland, June 9 – 11 2008, pp. 303-314.

Stake, R. E. 2005. "Qualitative Case Studies," in *The SAGE handbook of qualitative research*, N. K. Denzin, and Y. S. Lincoln (eds.), Thousand Oaks: Sage Publications, pp. 443–466.

Vitharana, P. 2003. "Risks and challenges of component-based software development," *Communications of the ACM* (46:8), pp. 67–72.

Wegner, D. M. 1987. "Transactive memory: a contemporary analysis of the group mind," in *Theories of Group Behavior*. B. Mullen & G. R. Goethals (eds.), New York: Springer-Verlag, pp. 185–208.

Word, J. (ed.) 2009. *Business network transformation: Strategies to reconfigure your business relationships for competitive advantage*, San Francisco, CA: Jossey-Bass.

Wulf, V., and Rohde, M. 1995. "Towards an integrated organization and technology development," *Proceedings of the DIS '95*, ACM Press, pp. 55 - 64.

Wulf, V., and Jarke, M. 2004. "The economics of end-user development," *Communications of the ACM* (47:9), pp. 41-42.

Yin, R. K. 2003. *Case study research: Design and methods*, Thousand Oaks, Calif: Sage Publications.