International Journal of
Information Systems and Project Management
ISSN (print):2182-7796, ISSN (online):2182-7788, ISSN (cd-rom):2182-780X
Available online at www.sciencesphere.org/ijispm

Use Cases and Collaboration Scenarios: how employees use socially-enabled Enterprise Collaboration Systems (ECS)

Petra Schubert
University of Koblenz-Landau
Universitätsstr. 1, 56070 Koblenz
Germany
www.shortbio.net/schubert@uni-koblenz.de

Johannes H. Glitsch
University of Koblenz-Landau
Universitätsstr. 1, 56070 Koblenz
Germany
www.shortbio.net/jglitsch@uni-koblenz.de

Abstract:
In recent years we have seen the emergence of a new type of collaboration software, the so-called "Enterprise Social Software". The "social features" of this software type have stimulated a renewed interest in Enterprise Collaboration Systems (ECS). In this article we present findings from a longitudinal research project on the introduction and use of ECS in companies. We argue that ERP Systems and ECS are inherently different and that the process-paradigm that is common to ERP cannot be applied identically to ECS. To address this issue, we suggest the two concepts *use case* and *collaboration scenario* for the analysis and description of collaboration activity in companies. From the literature and 26 case studies we identified typical use cases and collaboration scenarios that can serve as blueprints for ECS introduction projects. The longitudinal objective of our research is to assist companies with their ECS initiatives and to provide them with a catalog of existing use cases and collaboration scenarios from various industry settings.

Keywords:
Enterprise Collaboration Systems (ECS); Enterprise Social Software (ESS); CSCW; Use Cases; Collaboration Scenarios.

DOI: 10.12821/ijispm040203

Manuscript received: 17 February 2016
Manuscript accepted: 2 May 2016

1. Introduction

Enterprise Collaboration Systems (ECS) are software systems that support the collaborative work of employees. ECS comprise all areas of collaboration such as information and content sharing, communication, cooperation and coordination as described in the 8C Model for Enterprise Information Management [1]. The first forms of ECS evolved under the name "groupware" [2] around the year 1984. Since then, research on ECS has been conducted in the field of *Computer-Supported Cooperative Work (CSCW)* [3]. Enterprise Collaboration Systems (ECS) support employees in all areas of their joint work and are an important enabler of the modern digital workplace. They have recently gained renewed attention through the emergence of a new form of *socially-enabled* collaboration software. Since around 2005, Social Media platforms have become very popular for *private use* and it was only a question of time before their "social features" (e.g. social profiles, microblogs, chat, activity streams) were implemented into business software, bringing forward a new software type that is now discussed under terms such as *"Enterprise Social Networks (ESN)"* [4] or *"Enterprise Social Software (ESS)"* [5] in the academic literature.

Our research showed that Enterprise Social Systems (ESS) will soon become a necessary component of the basic IT infrastructure especially in innovative and service-oriented companies. Heinz and Kumar call it "backbone" in their talk on the introduction of a large ESN at Robert Bosch, a large German manufacturer of home appliances and automotive parts. At a business conference in February 2016 they stated that "The #ESN will be the backbone of future organizations – and thus a prerequisite for business operations" (Heinz and Kumar, IBM Connect, Orlando, Feb 1, 2016). ESS are changing the way that employees work together [6] just as the introduction of E-Mail changed communication between the hierarchical structures in companies more than 20 years ago [2]. Even though early adopters of ESS are confident that this software will enable their companies to become more agile and to collaborate more effectively [7], there are still many open questions regarding the opportunities for use that this new type of software brings about.

In this article we are looking at collaboration software for businesses in general, old and new systems alike. We are using the term *Enterprise Collaboration System (ECS)* for software applications that support collaboration in companies [6]. In our understanding ECS are socio-technical systems that include hardware and software as well as people, processes and organizational aspects.

Some of the open questions regarding the proper use of ECS have their roots in the characteristic traits of this kind of software. The use of an ECS cannot be prescribed and it is hard to develop manuals or guidelines for its use. Following theory on social construction of technology (SCOT) the affordances of the software are open to an interpretative flexibility [8] meaning that the capabilities of the software are dependent on the experiences and skills of the person using it. Its features are thus partly defined through the actual use. To give an example, the software product IBM Connections provides users with the possibility to create templates for activities with lists of single tasks that can be assigned to group members. The process of creating such a template is straightforward but the *areas of use* are endless. During our research we found evidence for multiple purposes of such task lists such as project and event management, checklists for the repair of machines or the onboarding of new employees. These areas for use (which we call *use cases*) are identified and implemented by a specific company and require a certain degree of creativity on the part of the user.

There are *fundamental differences* between software that supports collaboration between people (ECS) and the more process-oriented ERP systems. The first important difference concerns their *application area and the structure of their content*. ERP systems are based on a process-oriented view [9] with the aim of supporting clearly defined and repeatable business functions following built-in business rules. ERP systems are critical to businesses because they support the core order fulfilment process. ECS, on the other hand, are designed to support *joint work* among people in the workplace. They are supportive in nature and their continuous availability is usually less business critical than in the case of ERP systems [7]. Whilst ERP data comprises highly structured master data and transactional data reflecting the company's resources and business activities, ECS contain, for the most part, unstructured content such as documents, blogs or posts. Another difference lies in the *implementation process*. It is accepted that the selection and implementation of ERP systems must follow a well-defined project plan [10], [11] whilst ECS are often reported to

follow a "bottom up" [12] and rather experimental [13] introduction approach. They also differ in *purpose and use*. ERP systems give little room for creativity and they impose their structure and their implemented order of events onto the user. The use of ERP systems is mandatory for activities in the order fulfilment process. ECS, on the other hand, are tools for ad-hoc use which offer choice and thus entail uncertainty [14]. Both system types require skills for their use, however, ERP skills are much more routine. ECS require the user to understand the suitability of a tool for a current task at hand and to make appropriate selections. ECS use is often voluntary so that the user has to acknowledge the benefits of using the tool. This is why *"user acceptance"* has traditionally played an important role in research on collaboration systems [15].

In our article we argue that the use cases and collaboration scenarios supported by the new generation of socially-enabled Enterprise Collaboration Systems (ECS) are not yet well understood and that we need ways of classifying and describing the dimensions of collaboration scenarios.

We believe that the paradigm of the *business process* that describes a defined sequence of tasks and events does not work for areas that involve a high degree of collaboration. We argue that we need a new paradigm when we define the recommended use for collaboration systems because the sequence of activities that is supported by the ECS is mostly ad hoc and thus in large parts difficult or impossible to prescribe or automate. In a collaboration activity, the user is continuously making choices about which tool to use to support the task at hand. In the early stages of adoption of an ECS making this choice requires an intellectual effort for the person performing the task. It is only over time that users appropriate [15] collaboration technology and (may) become able to use them in an effortless manner and without too much thinking about it.

As a consequence, we argue that whilst *ERP implementation projects* are about understanding business processes and finding ways to ideally support them, *ECS implementation projects* are about identifying use cases and collaboration scenarios that best suit a specific company and the people working in it. By understanding the potential of the ECS, companies can create a better and more efficient digital workplace for their employees. Our final research objective is to develop a *catalog* (database) of *use cases* and *collaboration scenarios* that provides a structured overview of current practices and stimulates ideas for future use.

2. Use Cases and Scenarios in the Literature

The term use case was first used in 1987 by Jacobson [16]. Jacobson defines the term use case as a "special sequence of transactions, performed by a user and a system in a dialogue" ([17] cited in [18]). The concept of the use case can be found in both, the academic literature as well as in publications by practitioners [19] and has, since its first occurrence, become a very popular way of describing software requirements [16]. The field of computer sciences has described the use case as a formal concept in UML (Unified Modeling Language) [18], [20]. The Object Management Group's (OMG) specification of UML considers use cases to be "means for specifying required usage of a system" [21, p. 597]. In the OMG's definition, use cases are specific to *one* organization and describe a situation at a high level with little specific detail. This is underlined by the example in the UML specification describing a telephone catalog at a very general level [21, p. 585]. Generally, use cases contain the description of actors and how these actors interact with a (computer) system to achieve a defined business goal. Jacobson et al. [22] emphasize that the descriptions of use cases, which often occur in the form of stories, should also include the value that a system provides to its users.

In practice, the concept of a use case is not always applied according to its above definition and use cases may seem ambiguous in some respect. Irwin and Turk [23] mention that some elements in particular, such as "actor" and "association between actors and use case" are not used in a consistent way. A selective search for the term "use case" in the CSCW literature confirms this ambiguity. Osimo et al. [24], for example, identified a number of use cases such as "internal management process", "knowledge creation and sharing (internal)" and "expertise location". Along the same lines, a Gartner report [25] lists common use cases, for example "internal communications", "project team coordination" and "knowledge management". Whilst all of these examples seem to be valid use cases, the level of abstraction that they contain varies. A use case named "management process" seems to be on a much higher abstraction

level than the very specific-sounding use case of "expertise location". While it appears that there is an agreement in the literature that use cases describe *what* happens, their level of detail and their exact use is not consistent across the literature even though a large percentage of articles refer to the UML definition. The same applies to the literature in the field of CSCW, which is also lacking a uniform use of the term.

Based on our literature review and following the general concepts provided by UML and the initial ideas of Jacobson [17] we define the term "*use case*" as follows:

> A use case describes a high level business activity with a focus on the interactions of a user and a (computer) system to support the tasks that are required to complete the activity (i.e. to achieve a business goal). Use cases can describe activities that are applicable to many companies (e.g. project management) or they can be specific to a particular organization (i.e. supporting an activity only found in this company). The use case is characterized by a high level of abstraction and is technology agnostic. It can be further detailed with the help of collaboration scenarios (see below).

As shown in the previous section, use cases are defined at a high level of abstraction, which calls for a more detailed concept that brings us closer to the level of the actual software features. We propose the term *collaboration scenario* to further specify the steps of the interaction in a use case. The term "scenario" is widely used in the literature. During our literature search we found thousands of mentions of the term. The term is also broadly used in everyday language, where a scenario is often understood as an outline or description of a scene (e.g. Merriam-Webster and Oxford Dictionary). Bolloju and Sun [26] note that scenarios have been used in many ways in the literature, not only in terms of what they describe, but also how they are described. The possibilities seem to range from any text-based representation of activities to structured diagrams. They use a graphical representation as a basis themselves, which is put in the context of requirements engineering. The term is inconsistently used in areas where collaboration takes place [27]. From our research we could find several examples where scenarios are used to help in the description of requirements with some collaborative aspects (e.g. [28], [29]). There are a number of articles that reflect a meaning and intention of the term that supports our purposes, for example in publications about "Anwendungsszenarien" [30] (German: application scenarios") and "Anwendungsbereiche" [31] (German: "application areas") or simply scenarios that are textually described [32]. Examples of scenarios from this previous work include "information sharing", "discussion" or "internal marketing". Alternative uses of the term scenario include the work by Niemeier [33], who uses the word "application scenario" to describe actual fields of application such as "innovation" or blueprints such as "training on the job". Other authors in the field of CSCW are using the term "cooperation scenarios" [34] or simply "scenario" [35].

Based on our literature review and our previous research [36] we define the term *"collaboration scenario"* as:

> A composition of activities that are carried out by one or more people (actors) to achieve a common task (collaboratively). Collaboration scenarios describe the specific steps of the interaction among human actors and/or social documents involved in the joint work. Collaboration scenarios are generic components that can occur in different use cases. Collaboration scenarios include references to concrete software features and can be used to identify the necessary software. They can thus be used in the evaluation process as a link between use cases and actual collaboration software.

As previously stated [36] the nature of collaboration scenarios is different from that of a business process as the collaborative interactions depicted in it are more detailed. The sequence of activities does not describe a pre-defined order of tasks but rather a flexible set of tasks and checkpoints that may or may not be put in order due to preconditions imposed by other factors like the creation of documents. The way to reach a checkpoint may depend on multiple factors, one of which is the artifacts that are involved. Social documents [37], [38] such as files, blogs or wiki pages enriched by tags, hyperlinks or likes are examples of such artifacts and may impose certain conditions on the way to complete a collaboration scenario and can be central to the outcome. A more conceptual view of use cases and collaboration scenarios is presented in section 4 of this article.

3. Research Design

The following section describes the interpretive, qualitative approach taken in our research. The research was mostly conducted in the years 2015/2016 and was organized in three phases (cf. Fig. 1):

1. Framework development: categories, terms and definitions;
2. Coding, framework enrichment and revision;
3. Completion of framework and population of the catalog.

Phase 1 (cf. section 3.1) was aimed at understanding and developing the basic terminology framework surrounding use cases and collaboration scenarios. For this purpose, a structured literature review was conducted and 14 existing cases (descriptions of ECS implementation projects) were analyzed. In *phase 2*, the initial framework was used to guide the structured coding of 12 additional cases by two independent researchers. Again, the researchers analyzed and interpreted existing descriptions of ECS implementation projects looking for use cases and collaboration scenarios, which they documented in the form of "codes". The findings were discussed and full agreement on the codes was established.

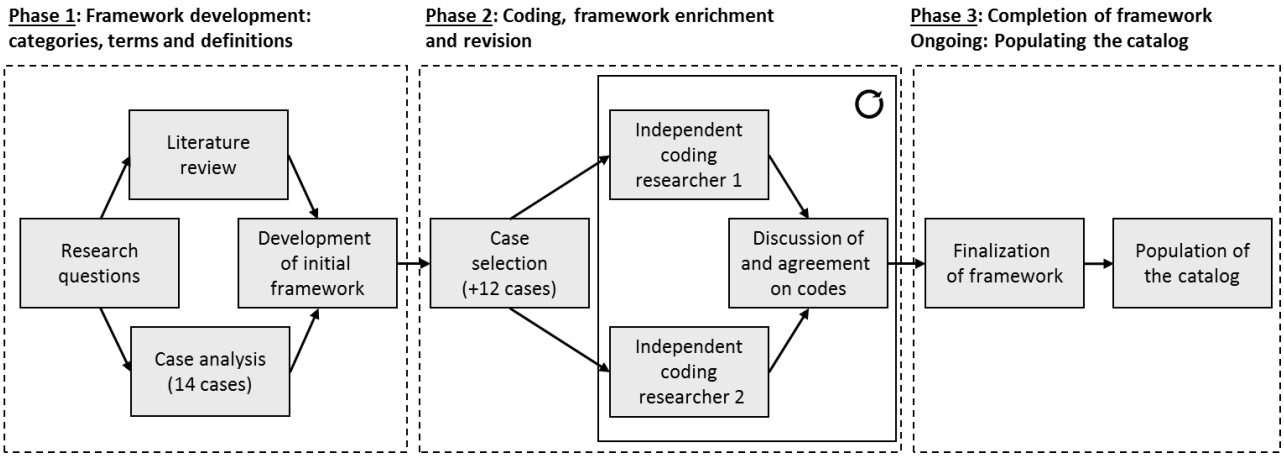

Fig. 1. Research steps

In *phase 3* the initial framework was revised and its dimensions were used to create a database of use cases and scenarios (which we call "catalog"). The catalog was populated with the codes identified in phase 1 and 2.

The first findings of phase 1 were presented in a previous publication [36]. In this article, we focus on phases 2 and 3. The activities of all phases will be further described in the following sections.

3.1 Phase 1: Framework development: categories, terms and definitions

The first phase of our research involved an examination of existing literature and a preliminary analysis of cases on ECS implementation projects. The findings from these two sources helped us to develop our definitions and our understanding of the dimensions of use cases and scenarios.

In order to gain a better understanding of the relevant terms and definitions a *structured literature review* following Webster and Watson [39] was conducted. The search was carried out using the EBSCOhost search engine, which allows the search across several scientific publication databases including Business Source Complete, EconLit and SocIndex. First, a broad search was performed on (peer-reviewed) academic journals using the keywords "use case" as well as "scenario" in order to gain a general feeling for the number of occurrences of these terms in the literature. The search resulted in 1,414 and 78,437 hits respectively. In the next step, the search parameters were narrowed down. We used

either the keyword “use case” or the keyword “scenario” in combination with other search terms to improve the relevance of the results. Combinations with keywords such as “collaboration”, “cscw”, “computer supported cooperative work”, “groupware”, “origins”, “history” and others were used. This procedure proved to be more helpful, however, only few results could be found that matched the specific context of our investigation (i.e. Enterprise Collaboration Systems). This subset of articles was then examined for references to further literature that seemed relevant for our topic area (snow-ball technique). Some important findings of our literature review have already been presented in section 2 of this article.

In parallel to the literature search, a *case analysis* was carried out that yielded codes for our initial framework and served as an important input for the next research phase. Fourteen industry cases were examined. The findings have been documented in [36]. The initial framework included 13 use cases and 13 collaboration scenarios listed in Table 1 and Table 2.

Table 1. **Use Cases** sorted by occurrence identified in the first 14 cases [36].

#	Use Case	Grounded
1	Knowledge sharing	11
2	Enterprise communication	8
3	Project organization	7
4	Sales opportunity handling	2
5	Collaborative quote compilation	1
6	Accounting organization	1
7	Human resources organization	1
8	Idea and innovation organization	1
9	Internal marketing	1
10	Software development organization	1
11	Team organization	1
12	Workshop organization	0
13	Trade show organization	0

Table 2. **Collaboration scenarios** sorted by occurrence identified in first 14 cases [36].

#	Collaboration Scenario	Grounded
1	Information and knowledge handling	12
2	Information exchange (“push/subscription”)	11
3	Knowledge collection (e.g. handbook) (“pull/on-demand”)	9
4	Expert search	8
5	Discussion	7
6	Document lifecycle handling	5
7	Meeting minutes and tasks	4
8	Conference	3
9	Joint authoring (synchronous/asynchronous)	3
10	Problem solving	3
11	Organization of meetings	2
12	Reporting	1
13	File sharing	0

3.2 Phase 2: Coding, framework enrichment and revision

In phase 2, another 12 industry cases were analyzed and coded in order to extend and, if necessary, revise the initial framework. The industry cases were selected from the E2.0 Cases database (www.e20cases.org). This open access database contains industry cases on software implementation projects in the domain of collaboration. The cases that were selected for our analysis are categorized as “orange” and “gold”. “Orange” cases (cases 1-14) follow the eXperience method, a structured approach for writing cases [40]. “Gold” cases (cases 15-26) are also rich cases that are systematically written, but do not follow the well-defined eXperience structure. Table 3 gives an overview of all industry cases that were analyzed in phase 1 and 2 with information on company size, industry sector, project objectives and the software used. The case IDs are later used for the documentation of the sources of our codes in the results tables (cf. section 5).

Table 3. Cases used in phase 1 and 2 to develop the framework.

ID	Case	No. of Employees	Industry Sector	E2.0 Project Objective	Software
1	ABB	120,000	Energy and Automation Technology	Blog and wiki in enterprise communication	Windows SharePoint Services 3.0
2	ADTELLIGENCE	10	Advertising	Organizing all information with social software	Several Web 2.0 tools
3	Börse Berlin	26	Securities trading, B2B	Communication between exchange and private investors	Invision Powerboard
4	Capgemini	100,000	B2B services and solutions	Expert identification and discussion	Yammer
5	Communardo	180+	Information and Communication	Microblogging	Microblogging bespoke software
6	DocHouse	11	Consulting, IT, software	Collaboration CRM	IBM Lotus Quickr
7	ESG	700	B2B development, integration and operations	Knowledge management	Atlassian Confluence
8	Fritz & Macziol	700	B2B and B2A consulting and system house	Knowledge gathering, transfer and expert search	IBM Lotus Connections
9	Pentos	35	Consulting, IT, software	Employee blogging	IBM Lotus Notes
10	Rheinmetall	20,000	B2B and B2A development and production	Team room, discussions and yellow pages	IBM Lotus Collaboration Technology
11	SFS Services	4,246	IT services	Wiki for knowledge transfer	MediaWiki
12	Siemens	405,000	B2B consulting, development and production	Global knowledge management and expert search	Liferay
13	Siemens Building Technologies	40,000	Software, systems, services	Knowledge transfer and communication	Collaboration platform Reference+
14	T-Systems Multimedia Solutions	1,000	Software, consulting	Collaborative team work	Atlassian Confluence Enterprise Wiki
15	Siemens	475,000	B2B consulting, development and production	Weblog for knowledge management	Twoday.net-based Weblog
16	Sun Microsystems	35,000	IT services	Wikis and weblogs	Atlassian Confluence
17	Saia-Burgess Controls AG	340	Electronic automation and controls	Information and knowledge management	Google Apps (for Business)
18	Teufelberger	750	Manufacturing (steel ropes, composites, …)	Information and knowledge management	Microsoft SharePoint Server 2010
19	Factline Webservices	11	IT services (information management and communication)	Task management with tags	Task management software (custom implementation)
20	Greentube	160	Full service provider in	Knowledge management	MediaWiki

ID	Case	No. of Employees	Industry Sector	E2.0 Project Objective	Software
			online gaming	with wiki	
21	IQ mobile	27	Full service provider for mobile media	Knowledge distribution with weblog	WordPress
22	Schuldnerberatung Wien	35	Debt counseling	Knowledge management with wiki	Wiki (custom implementation)
23	WINTERHELLER	150	Software development; IT consulting	Software documentation with wiki	WINTERHELLER Competence Center
24	Valyue Consulting	No data	IT consulting	Enterprise communication and knowledge management	Jive
25	Swiss Re Ourspace	10400	Reinsurance	Project management with Jive	Jive
26	Swiss Re	10400	Reinsurance	Enterprise collaboration with ECS	Jive

We took a multi-level coding approach [41] for the analysis of the selected cases (cf. Fig. 2). The codes from the initial framework (cf. Table 1) were applied by both researchers independently, who, at the same time, were challenging these codes and scanning for new codes (i.e. new use cases and collaboration scenarios). This *structural coding process* was followed by a discussion of the codes with the objective to establish an agreement of the identified use cases and scenarios. The first coding round yielded 34 codes, therefore enlarging the original code set of 26 codes by additional 8 codes.

In a second round of coding the codes were checked against *all* 26 industry cases again including the newly defined and previously undiscovered codes. Upon completion of the second round of coding the results were discussed again. The final set of codes consisted of 14 use cases (cf. Table 4) and 18 collaboration scenarios (cf. Table 5).

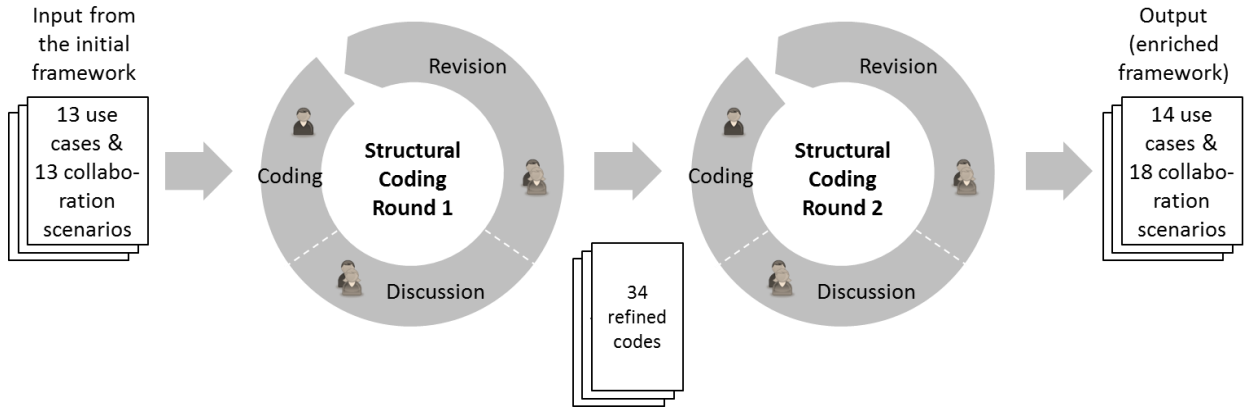

Fig. 2. Two rounds of coding

3.3 Phase 3: Completion of framework and ongoing population of the catalog

In phase 3 the framework was finalized: For each of the codes a short description was written based on the literature review and the analysis of the industry cases. Additionally, collaboration scenarios were mapped to use cases and

features were mapped to collaboration scenarios. The results were used to populate the catalog of use cases and collaboration scenarios. The codes and the content of the catalog will be further described in section 5.

4. Analytical Framework: Structuring Enterprise Collaboration

In this section we will further explain the theoretical framework that guides our analysis. Research findings show that the introduction phase of an ECS is critical for the adoption of this technology [42]. Often, ECS fail to be accepted by staff in the early implementation phase and it is difficult to turn user perception around once a negative opinion has been formed regarding the new system.

In order to support management in the decision process for an ECS we were searching for ways of structuring the problem domain in order to facilitate the ECS evaluation process. With this objective in mind we used the IRESS framework previously described in [36]. The IRESS framework provides a contextual view at the social software requirements of a company. The acronym "IRESS" stands for "Identification of Requirements for Enterprise Social Software (ESS)". The framework is composed of four levels (cf. Fig. 6 in appendix A) containing conceptual elements that can be used to model the collaboration requirements of a specific company. The *top level* suggests the identification *of business processes and use cases* that need to be supported by collaboration technology. The *second level* is dedicated to the *collaboration scenarios*, which are, in accordance with our above definition, modular components that support business processes and use cases. The *third level* gives an overview of the *software components*, which are necessary to support collaboration scenarios. The *bottom level* contains the actual *"collaborative features"* and is structured using the dimensions of the 8C Model for Enterprise Information Management by Williams [1]. The *top two levels*, business process/use cases as well as their supporting collaboration scenarios represent the *"organizational view"* in evaluation projects whereas the focus of the *two lower levels* is on the *actual software support*.

The IRESS framework implies a *task-oriented approach* and provides a systematic view to bring order to the rather unstructured field of collaboration. Comparable to other models for business analysis (such as ARIS) the IRESS framework requires companies to analyze their *business processes* and *use cases* first, to establish an overview of their sequence of activities (*process map*) and their organizational units (*organizational chart*). Most companies will not be able to model all their business activity in processes because not all business activity is strictly sequential. Processes are based on the idea that the sequence of tasks is more or less predictable and stable (structured) but there is also *project-oriented work* going on in companies which cannot be described in a strict sequence and which requires a higher degree of flexibility in the order of events. We propose to describe these "other" forms of business activities in use cases, e.g. the *organization of a trade show* or classical projects such as *product development* or *research*.

Business processes are characterized by activities that have a structure and that can be modelled as a pre-defined sequence of tasks. We use the term *use case* to describe other forms of business activity for which the sequence of events is unpredictable. Both concepts, processes as well as use cases, can be supported by *collaboration scenarios* as defined above.

The process map and the overview of use cases on the top level of the IRESS framework serve as the basis for identifying candidate areas for collaboration that contain a high concentration of C^4-activities (communication, cooperation, content, coordination). The identified business processes and use cases are analyzed and their *collaboration scenarios* are identified. Typical (generic) collaboration scenarios are, for example, creating meeting minutes and tasks or file sharing.

Collaboration scenarios can then be mapped to *feature bundles*, which we call *collaborative software components* that support one or several C^4 activities. The final aim of our research is to provide a mapping between collaboration scenarios and collaborative software components in a *Collaboration Scenarios Catalog (CSC)*. The catalog has been designed to contain a range of (generic) collaboration scenarios that frequently occur in companies.

Fig. 3 shows a taxonomy for collaboration activities that helps clarify the level of discussion. *Use Cases* form the top of the taxonomy. They are general descriptions of a business activity and can occur in multiple companies. Examples are

"Event Management" or "Project Management". The actual instance of a use case on a detailed level is company-specific. As explained above, we use *collaboration scenarios* to describe the detailed view of activities. These are rather general in nature and applicable to multiple companies. However, variations from the generic collaboration scenario during actual instantiation are possible. On the lowest level, these collaboration scenarios are supported by a composition of (atomic) software features (e.g. a blog post or a text message).

The *use case* is meant to demonstrate the *business value* that the users can derive from the application of collaboration software. The collaboration scenario shows the *actual actors, tasks and their interaction* and how they can be supported by technology.

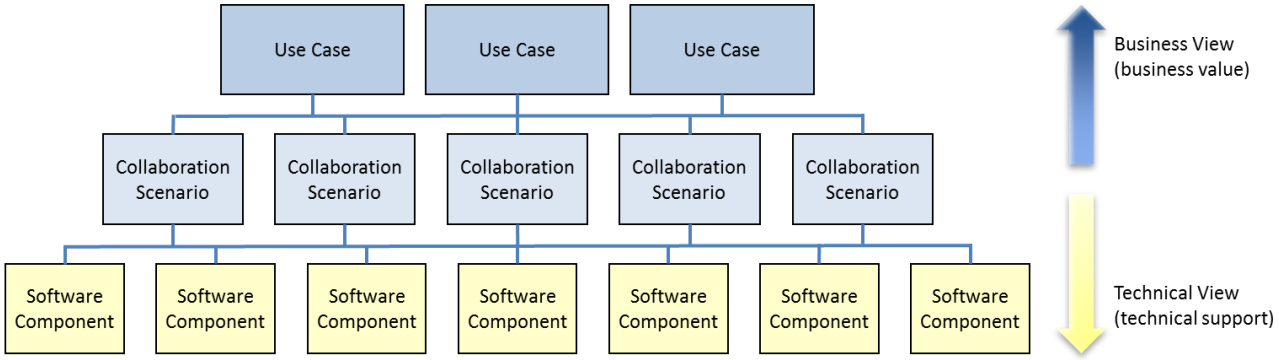

Fig. 3. Use cases consist of collaboration scenarios which are supported by software components [36, p. 165]

Our initial research showed that the distinction between use cases and collaboration scenarios is useful in the context of Enterprise Collaboration Systems [36]. However, from the cases that we analyzed so far, it became apparent that a strict two-level distinction is not enough. Our coding showed that it was possible to identify independent use cases that are composed of different collaboration scenarios. Collaboration scenarios, however, are sometimes composed of other collaboration scenarios. Some scenarios appear as subcomponents in other scenarios, which calls for a nested concept. These *nested collaboration scenarios* result in a two-way relationship between collaboration scenarios and their possible compositions (cf. Fig. 4). On the one hand, a collaboration scenario may (but does not need to) be composed of other collaboration scenarios. On the other hand, a collaboration scenario may (but does not have to) be a component of another collaboration scenarios.

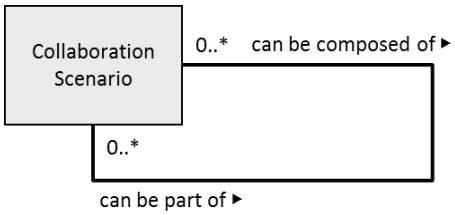

Fig. 4. Concept of nested collaboration scenarios (using UML)

To give an example, the use case "project organization" could be made up of the four collaboration scenarios *"expert search"*, *"discussion"*, *"meeting minutes and tasks"* and *"file sharing"*. At the start of the project the team needs to be staffed with the right people (expert search). The team needs a platform for the exchange of ideas (discussion) and a joint library for files (file sharing). During the meetings notes need to be taken and tasks need to be assigned to the team members (meeting minutes and tasks). While these collaboration scenarios are all part of the same use case, file sharing may occur in a discussion or in the context of meeting minutes and a task as well. Posts in a forum (discussion) may contain shared files. The same applies to minutes. Therefore, the collaboration scenario "file sharing" can either be used separately or as a subcomponent in the other two collaboration scenarios as illustrated in Fig. 5.

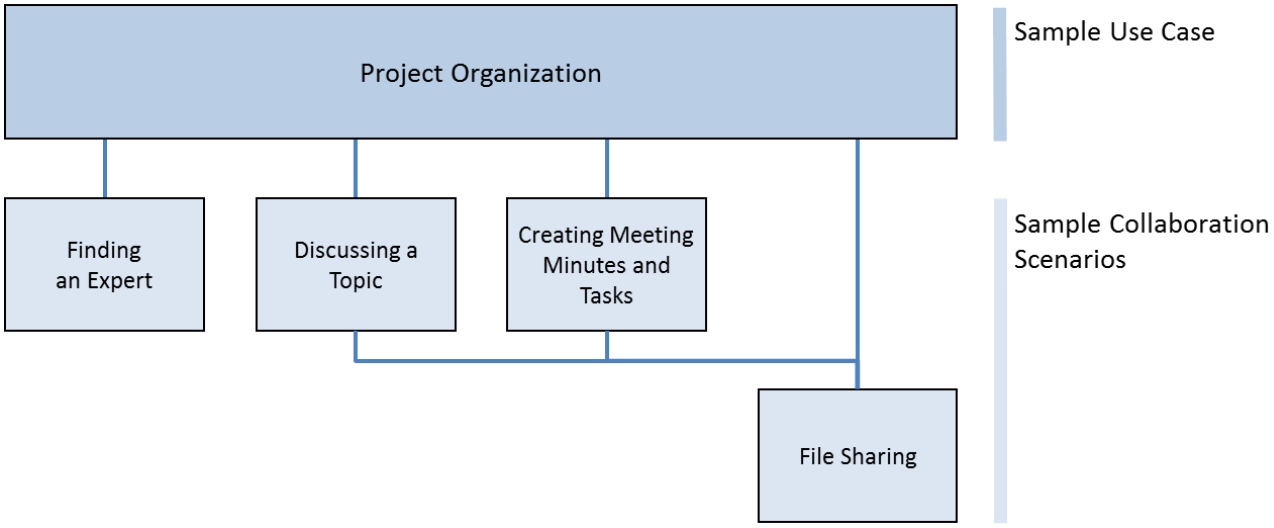

Fig. 5. Example for nested collaboration scenarios in the use case project organization

In the next section, we will describe the codes that could be identified in the analysis of the cases with the help of our initial framework.

5. Findings: Use Cases and Collaboration Scenarios

In the *first phase* of our research we developed an understanding of the special nature and the distinctive characteristics of *use cases* and *collaboration scenarios* and we were able to successfully identify a series of cases and scenarios that match our definition [36]. In the *second phase* we deepened our understanding and extended the code base in two additional coding rounds with the help of further industry cases. Table 4 lists the codes for *use cases* from this second phase and provides a description for each use case. The table also shows the number of times a code occurred in the cases (groundedness) and the sources in which this code was found. The column "sources" contains the IDs of the cases shown in Table 3 above. The last column contains examples of related scenarios in order to illustrate the actual activities in this case.

In the second round of coding we revised the naming of use cases and scenarios. We are now using *nouns* for *use cases* and make use of *verbs* for *collaboration scenarios* to facilitate the differentiation. We also added the prefixes UC (use case) and CS (collaboration scenario) to make the description unambiguous. To illustrate the new naming concept, the use case originally just called *"knowledge sharing"* has now been renamed to *"UC: Knowledge management"*. "Software development organization" is now called *"UC: Software development"*. The new naming scheme was a result of our refined understanding of use cases in the ECS context.

Table 4 also shows examples of collaboration scenarios that illustrate the use cases. For example, *UC: Knowledge management* usually requires that information is available. One way of putting such information into the system can be done by *CS: Documenting information*. Also, to make it easier to find the information later, some form of document enrichment might be necessary (e.g. tagging). This can be described with the collaboration scenario *CS: Managing information*. The frequently mentioned use case *UC: Project organization* commonly includes meetings; example collaboration scenarios applicable include *CS: Organizing a meeting* as well as *CS: Conducting a meeting*.

The coding of the cases had some limitations common to the analysis of *secondary literature* that was written for a different purpose. We believe that some of our developed codes are not necessarily describing "ideal" cases and scenarios. The codes are a representation of what was reported in the selected cases using the level of detail that was provided by the authors. A more detailed analysis of the actual activities in Enterprise Collaboration Systems will be necessary to develop a richer representation of collaboration activities that can serve as an orientation for best practice.

Table 4. Use cases identified (sorted by column "grounded").

No.	Use Case (UC)	Short Description	Grounded	Sources	Related Scenarios (examples)
1	UC: Knowledge management	Activities involving the documentation of experiences and expertise of employees making this knowledge available for others.	21	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18, 20, 21, 22, 23, 24, 26	Documenting information, enriching information
2	UC: Enterprise communication	General support of communication within the enterprise, comprising synchronous and asynchronous forms of information exchange between employees.	11	2, 3, 4, 5, 6, 7, 8, 9, 18, 24, 26	Discussing topics, conducting a meeting
3	UC: Project organization	All activities necessary to organize a project, including typical work such as joint task management and meeting preparation and documentation.	10	1, 2, 3, 4, 6, 7, 11, 18, 25, 26	Organizing a meeting, conducting a meeting
4	UC: Sales opportunity handling	Management of collective information available to decrease the time for a customer response and the quality of the information provided.	3	11, 14, 24	Finding an expert, retrieving information
5	UC: Software development	Collaborative support for software development teams, typically involving task management and documentation.	3	4, 16, 23	Documenting information, conducting a meeting
6	UC: Customer communication	Collaborative activities with a focus on the customer, typically supporting CRM activities such as marketing material, newsletters, etc.	2	12, 24	Discussing topics, posting news
7	UC: Idea and innovation management	Supporting creative processes in the company e.g. by means of ideation management.	2	6, 18	Discussing topics, documenting information
8	UC: Management accounting	Support of collaborative tasks of post calculation of projects.	1	11	Documenting information, retrieving information
9	UC: Human resource management	Support of collaborative tasks of members of the HR department.	1	4	Documenting information, finding an expert

No.	Use Case (UC)	Short Description	Grounded	Sources	Related Scenarios (examples)
10	UC: Internal communications	Support of collaborative tasks of members of the internal communications department (e.g. monthly newsletter to employees).	1	8	Posting news, alerting to a news
11	UC: Quote compilation	Access to information necessary to compile a quotation, e.g. existing company knowledge or finding the right expert in the company.	1	11	Retrieving information, discussing topics
12	UC: Team organization	Long-term management of an organizational unit (e.g. a division, department or group) including typical work such as joint task management, meeting support and documentation; community without a fixed end date.	1	9	Organizing a meeting, conducting a meeting
13	UC: Event management	Support of activities for unique or recurring events such as a trade show.	0	n/a	Organizing a meeting, documenting information
14	UC: Workshop organization	Support of activities for workshops.	0	n/a	Organizing a meeting, documenting information

Looking at occurrence (groundedness), it is interesting to see that there are three use cases that are mentioned in many cases. *UC: Knowledge management* is the dominant use case with 21 unique mentions in 26 cases. It is followed by *UC: Enterprise communication* (11) and *UC: Project organization* (10) which both occur in more than one third of the cases. All other use cases could only be identified in between one and three cases. This supports our belief that use cases are rather company-specific.

Our identification of use cases and their importance is in accordance with previous findings in the literature. Even though the authors of related literature did not explicitly look at use cases they mention similar concepts, e.g. *the drivers* for the investment in Enterprise Social Software. Miles [43] lists the sharing of knowledge (*UC: Knowledge management*) as one of the biggest drivers for Enterprise 2.0. Other authors implicitly refer to the three top use cases when looking at the *achieved or unachieved contributions* generated by collaboration software (e.g. [44], [45]). The use case *UC: Project organization* is often described in publications about the collaborative nature of interactions in ECS (e.g. in [43]–[47]). While, again, the level of detail in the description of drivers and contributions varies, the general idea of beneficial use of ECS for the use case *UC: project organization* is supported by this literature. Other examples that are similar to our use cases could be identified as well. These include *UC: Customer communication* [44] and *UC: Idea and innovation management* [45]. The use of similar concepts for drivers, benefits and use cases makes it apparent that such a high-level view (that of use cases) alone is not enough thus calling for the more detailed view of *collaboration scenarios*.

Table 5 shows the *collaboration scenarios* that could be identified in the cases. The table has the same structure as the previous one, showing a description of the scenario, the groundedness, the sources in which this code was found and some exemplary features that would be used for this scenario.

Table 5. Collaboration scenarios (sorted by column "grounded").

No.	Collaboration Scenario (CS)	Short Description	Grounded	Sources	Related Features (examples)
1	CS: Documenting information	Making information available for future use	23	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23, 26	Blog posts, Wiki pages, markup of changes, tagging
2	CS: Retrieving information	Actively searching information in the ECS, targeted search and assembling of existing information	21	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 17, 18, 20, 21, 22, 23, 24, 26	Visualization of tag usage, search
3	CS: Discussing topics	Synchronous and asynchronous conversations between people	15	1, 2, 4, 6, 7, 8, 9, 11, 12, 13, 15, 17, 18, 25, 26	Chat, discussion forums, comments
4	CS: Sharing information	Active distribution of information to receivers with or without previous subscription ("push/subscription")	14	1, 2, 4, 5, 6, 8, 9, 11, 13, 17, 18, 20, 25, 26	Microblog posts, Blog posts, comments, content subscription
5	CS: Enriching information	Enriching or improving information such as adding meta data and annotations	12	2, 4, 6, 7, 8, 11, 14, 15, 16, 18, 20, 25	Ratings, pointers or references to content, tagging
6	CS: Finding an expert	Identification of matter experts in the collaborative network	9	3, 5, 6, 7, 8, 9, 11, 13, 18	User profiles, search, tagging
7	CS: Posting news	Writing a news message	7	2, 3, 7, 8, 12, 18, 26	Posts, message boards, tagging
8	CS: Conducting a meeting	Meeting with others in an online meeting environment	6	2, 4, 9, 10, 12, 19	Video conferencing, unified communication, screen sharing
9	CS: Alerting to news	Sending out alerts on news	6	2, 7, 8, 11, 18, 26	message boards, shared workspaces, workspace awareness, like
10	CS: Joint authoring	Synchronous and asynchronous collaborative authoring of documents, articles, etc.	6	2, 3, 7, 23, 24, 25	Shared authoring, shared workspaces, document and version control
11	CS: Problem solving	Solution of individual or common problems using collaborative capabilities	5	8, 9, 13, 15, 26	Discussion forums, comments, workspace awareness
12	CS: Creating meeting minutes and tasks	Writing of meeting minutes and creation of corresponding tasks	5	1, 3, 17, 22, 24	Posts, comments, tagging
13	CS: Organizing a meeting	Organizational steps towards conducting a meeting such as finding a date, booking rooms, writing minutes	4	2, 3, 14, 24	Discussion forums, chat, shared workspace

No.	Collaboration Scenario (CS)	Short Description	Grounded	Sources	Related Features (examples)
14	CS: Administering documents	Maintenance of documents such as archiving or activities to enrich documents with meta data	3	6, 7, 11	Ratings, pointers or references to content, tagging
15	CS: File sharing	Sharing of files with co-workers in directory-like structures	3	11, 17, 26	Shared workspace, document management, document and version control
16	CS: Conducting a poll	Asking for feedback or opinions on one or a few questions for quick results	1	26	Microblogging, polls and voting, ratings, rankings
17	CS: Conducting a survey	Asking for feedback or opinions on a matter with an online questionnaire for more comprehensive results	1	17	Posts, microblogging, polls and voting
18	CS: Rating information	Giving feedback on the perceived quality or usefulness of certain information	1	18	Posts, comments, ratings (e.g. stars)

18 collaboration scenarios were identified in the selected industry cases. The dominant collaboration scenario (mentioned in 23 of 26 industry cases) is *CS: Documenting information. CS: Retrieving information* is in second place. This is in accordance with the findings for the use cases because the first two collaboration scenarios are components of the number one use case.

6. Conclusions and Outlook

In our article, we present findings from an analysis of industry cases describing the use of ECS in companies. We suggest using the terms *use case* and *collaboration scenario* as a lens for the analysis of collaboration activities. The analysis of the literature showed that these two terms are not clearly defined. We are proposing a framework for the description of use cases and collaboration scenarios with the intention of providing a means to examine and develop requirements for Enterprise Collaboration Systems. With the help of 26 case studies on ECS introduction projects we were able to identify a set of concrete use cases and corresponding collaboration scenarios. These can be used for ideation and identification of possible uses in future ECS implementation projects.

Our findings are limited by the small scope of cases as well as the limited level of detail on collaboration activity that was provided by the case authors. As a consequence, we believe that the list of cases and scenarios presented in this article is by no means complete and more work must be done to develop them to successfully guide companies in their design of Enterprise Collaboration Systems. We were, however, able to demonstrate that our framework provides a suitable tool for the identification of cases and scenarios. We will continue our longitudinal work by investigating companies that have ECS in place and we are confident that the data collected in the field will help us to further populate the catalog of use cases and collaboration scenarios.

Acknowledgments

We would like to thank the members of the initiative IndustryConnect for sharing their real-world experiences on the actual use of their Enterprise Collaboration System. IndustryConnect is a collaboration project between early adopters of a leading integrated collaboration system. The initiative addresses current problems and issues in the field of collaboration in the digital workplace. The project is about the exchange of experiences between user companies and allows the researchers to gain valuable insights into current problems and best practice of ECS use in the field.

References

[1] S. P. Williams, "Enterprise 2.0 and Collaborative Technologies," Working Report of the Research Group Business Software, May 2010, University of Koblenz-Landau, Koblenz, Germany, 2010.

[2] W. J. Orlikowski, "Learning from Notes: Organizational Issues in Groupware Implementation," in *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'92)*, 1992, pp. 1–9.

[3] J. Grudin, "Computer-Supported Cooperative Work: History and Focus," *IEEE Comput.*, vol. 27, pp. 19–26, 1994.

[4] K. Riemer, S. Stieglitz and C. Meske, "From Top to Bottom: Investigating the Changing Role of Hierarchy in Enterprise Social Networks," *Bus. Inf. Syst. Eng.*, vol. 57, no. 3, pp. 197–212, 2015.

[5] C. Herzog, A. Richter, M. Steinhüser, U. Hoppe and M. Koch, "Methods and metrics for measuring the success of Enterprise Social Software – what we can learn from practice and vice versa," *Proc. 21st Eur. Conf. Inf. Syst. (ECIS 2013)*, pp. 1–12, 2013.

[6] P. Schubert and S. P. Williams, "The Concept of Social Business: Oxymoron or Sign of a Changing Work Culture?," in *Proceedings of the 26th Bled Conference*, 2013, pp. 222–235.

[7] T. Kuettner, R. Diehl and P. Schubert, "Change factors in Enterprise 2.0 initiatives: Can we learn from ERP?," *Electron. Mark.*, vol. 23, no. 4, pp. 329–340, Oct. 2013.

[8] D. MacKenzie and J. Wajcman, *The social shaping of technology*, 2nd ed. Buckingham, UK: Open University Press, 1999.

[9] T. H. Davenport, "Putting the Enterprise into the Enterprise System," *Harv. Bus. Rev.*, vol. 76, no. 4, pp. 121–131, 1998.

[10] M. L. Markus and C. Tanis, "The Enterprise System Experience - From Adoption to Success," in *Framing the Domains of IT Management: Projecting the Future Through the Past*, R. W. Zmud, Ed. Pinnaflex Education Resources, Inc, 2000, pp. 173–207.

[11] J. Becker, O. Vering and A. Winkelmann, *Softwareauswahl und -einführung in Industrie und Handel: Vorgehen bei und Erfahrungen mit ERP- und Warenwirtschaftssystemen*. Berlin Heidelberg: Springer, 2007.

[12] R. Diehl, T. Kuettner and P. Schubert, "Introduction of enterprise collaboration systems: In-depth studies show that laissez-faire does not work," in *Proceedings of the 26th Bled Conference*, 2013, pp. 236–250.

[13] A. Richter and A. Stocker, "Exploration & Promotion: Einführungsstrategien von Corporate Social Software," in *Proceedings of the Tenth Internationale Tagung Wirtschaftsinformatik (WI 2011)*, 2011.

[14] P. Dourish, "The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents," *Comput. Support. Coop. Work CSCW An Int. J.*, vol. 12, no. 4, pp. 465–490, 2003.

[15] K. Riemer and R. B. Johnston, "Place-making: A Phenomenological Theory of Technology Appropriation," in *International Conference on Information Systems*, 2012.

[16] M. Dahan, P. Shoval and A. Sturm, "Comparing the impact of the OO-DFD and the Use Case methods for modeling functional requirements on comprehension and quality of models: A controlled experiment," *Requir. Eng.*, vol. 19, no. 1, pp. 27–43, 2014.

[17] I. Jacobson, "Object-oriented Development in an Industrial Environment," in *Conference Proceedings on Object-oriented Programming Systems, Languages and Applications*, 1987, pp. 183–191.

[18] I. Jacobson, "Use cases – Yesterday, today, and tomorrow," *Softw. Syst. Model.*, vol. 3, no. 3, pp. 210–220, 2004.

[19] M. Ratcliffe and D. Budgen, "The application of use case definitions in system design specification," *Inf. Softw. Technol.*, vol. 43, pp. 365–386, 2001.

[20] B. Dobing and J. Parsons, "How UML is Used," *Commun. ACM*, vol. 49, no. 5, pp. 109–113, 2006.

[21] OMG, "OMG Unified Modeling LanguageTM (OMG UML), Superstructure, Version 2.4," 2010. [Online]. Available: http://www.omg.org/spec/UML/2.4/Superstructure/PDF/.

[22] I. Jacobson, I. Spence and K. Bittner, "Use-Case 2.0: The Guide to Succeeding with Use Cases," 2011.

[23] G. Irwin and D. Turk, "An Ontological Analysis of Use Case Modeling Grammar," *J. Assoc. Inf. Syst.*, vol. 6, no. 1, pp. 1–36, 2005.

[24] D. Osimo, K. Szkuta, P. Foley, F. Biagi, M. Thompson, L. Bryant, D. Bradshaw and J. Ritzek, "Enterprise 2.0 study: D4 Final Report," Bruxelles, 2010.

[25] N. Drakos, "Survey Analysis: Use of Social Software in the Workplace Is Prevalent but its Impact Is Hard to Quantify," 2013.

[26] N. Bolloju and S. X. Y. Sun, "Benefits of supplementing use case narratives with activity diagrams - An exploratory study," *J. Syst. Softw.*, vol. 85, no. 9, pp. 2182–2191, 2012.

[27] T. Allweyer, *Geschäftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling*. Herdecke/Bochum: W3L, 2005.

[28] L. Liu and E. Yu, "Designing information systems in social context: A goal and scenario modelling approach," *Inf. Syst.*, vol. 29, no. 2, pp. 187–203, 2004.

[29] J. Kim, S. Park and V. Sugumaran, "Improving use case driven analysis using goal and scenario authoring: A linguistics-based approach," *Data Knowl. Eng.*, vol. 58, no. 1, pp. 21–46, 2006.

[30] A. Richter, S. Mörl, M. Trier and M. Koch, "Anwendungsszenarien als Werkzeug zur (V) Ermittlung des Nutzens von Corporate Social Software," *10. Int. Tagung Wirtschaftsinformatik*, 2011.

[31] C. Houy, P. Fettke and P. Loos, "Einsatzpotenziale von Enterprise-2.0-Anwendungen - Darstellung des State-of-the-Art auf Basis eines Literaturreviews," Working Report of the Institut für Wirtschaftsinformatik (IWi) im DFKI, 2010.

[32] M. B. Rosson and J. M. Carroll, "Scenario-Based Design," in *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, J. A. Jacko and A. Sears, Eds. Hillsdale: L. Erlbaum Associates Inc., 2002, pp. 1032–1050.

[33] J. Niemeier, "Der Wandel der Unternehmen durch Social Media setzt sich fort," 2011. [Online]. Available: http://www.centrestage.de/2011/05/11/der-wandel-der-unternehmen-durch-social-media-setzt-sich-fort/.

[34] O. Stiemerling and A. B. Cremers, "The use of cooperation scenarios in the design and evaluation of a CSCW system," *IEEE Trans. Softw. Eng.*, vol. 24, no. 12, pp. 1171–1181, 1998.

[35] S. R. Haynes, S. Purao and A. L. Skattebo, "Scenario-Based Methods for Evaluating Collaborative Systems,"

Comput. Support. Coop. Work, vol. 18, no. 4, pp. 331–356, 2009.

[36] P. Schubert and J. H. Glitsch, "Adding Structure to Enterprise Collaboration Systems: Identification of Use Cases and Collaboration Scenarios," in *Procedia Computer Science*, 2015, vol. 64, pp. 161–169.

[37] V. Hausmann and S. P. Williams, "Social Business Documents," *Procedia Comput. Sci.*, vol. 64, pp. 360–368, 2015.

[38] V. Hausmann and S. P. Williams, "Issues for the long-term management of Social Business Documents," *International Journal of Information Systems and Project Management*, vol. 4, no. 3, in print, 2016.

[39] J. Webster and R. T. Watson, "Analyzing the Past to Prepare for the Future: Writing a Literature Review," *MIS Q.*, vol. 26, no. 2, pp. xiii–xxiii, 2002.

[40] P. Schubert and R. Wölfle, "The eXperience Methodology for Writing IS Case Studies," in *Americas Conference on Information Systems (AMCIS)*, 2007, no. 2, pp. 1–15.

[41] M. B. Miles, A. M. Huberman and J. Saldaña, *Qualitative Data Analysis: A Methods Sourcebook*, 3rd edition. Thousand Oaks: Sage Publications, 2014.

[42] K. Riemer, P. Overfeld, P. Scifleet and A. Richter, "Oh, SNEP! The Dynamics of Social Network Emergence - the case of Capgemini Yammer," Sydney, WP2012-01, 2012.

[43] D. Miles, "Collaboration and Enterprise 2.0," Silver Springs, MD, 2009.

[44] S. J. Andriole, "Business impact of Web 2.0 technologies," *Commun. ACM*, vol. 53, no. 12, pp. 67–79, 2010.

[45] S. P. Williams and P. Schubert, "Social Business Readiness Survey 2014," Working Report of the Research Group Business Software, May 2010, University of Koblenz-Landau, Koblenz, Germany, 2015.

[46] M. Chui, J. Manyika, J. Bughin, R. Dobbs, C. Roxburgh, H. Sarrazin, G. Sands and M. Westergren, "The social economy: Unlocking value and productivity through social technologies," *McKinsey Global Institute*, no. July. McKinsey Global Institute, 2012.

[47] M. Chui, A. Miller and R. P. Roberts, "Six Ways to Make Web 2.0 Work," *McKinsey Quarterly*, pp. 1–8, 2009.

Appendix A. IRESS Model

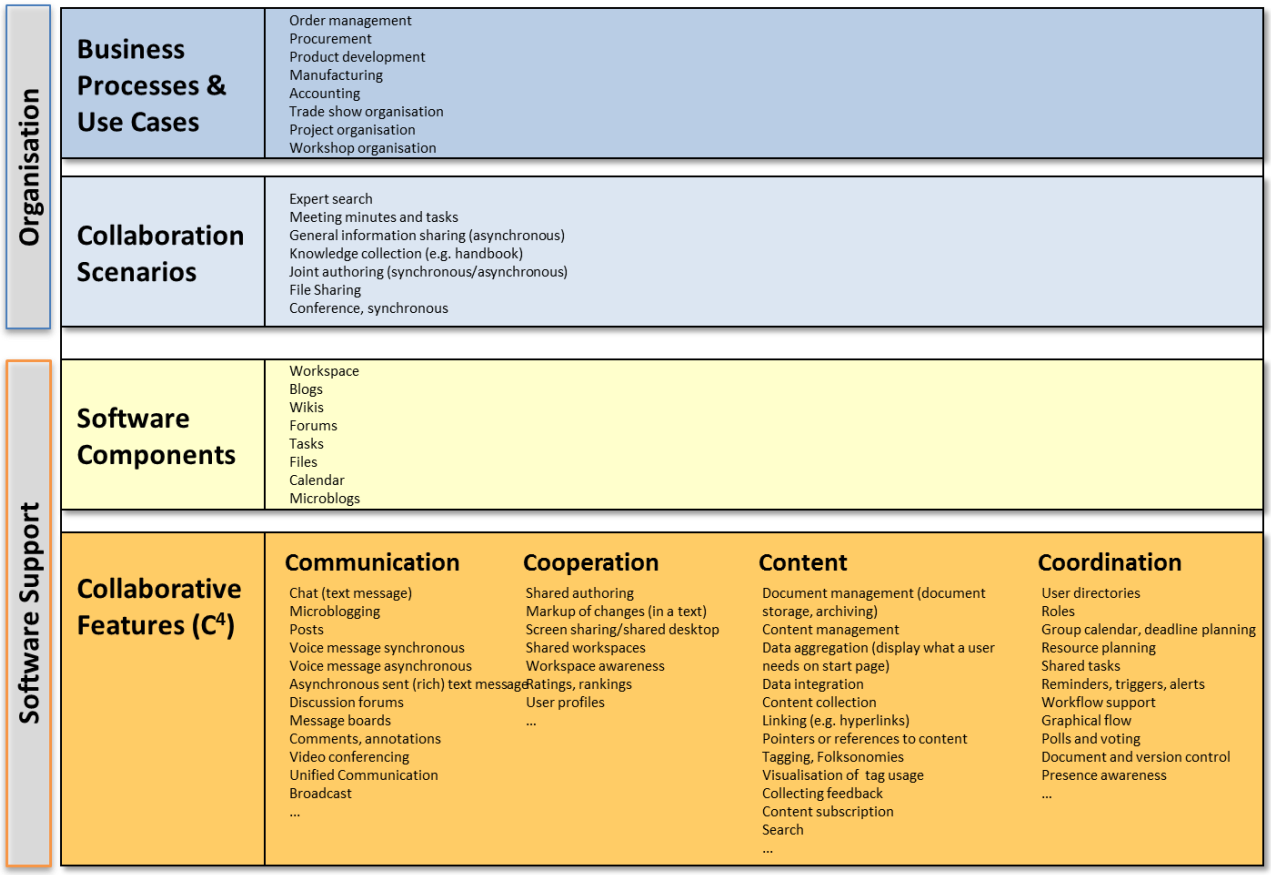

Fig. 6. IRESS Model: Identification of Requirements for Enterprise Social Software [36, p. 164]

Biographical notes

Petra Schubert
Petra Schubert is Professor of Business Software at the Faculty of Computer Science of the University of Koblenz-Landau. She is the Director of the Competence Centre for Collaboration Technologies (UCT). She holds a Ph.D. and a Master's degree in Information and Technology Management from the University of St. Gallen, Switzerland and a CEMS Master's degree (CEMS MIM). She received her "venia docendi" (Habilitation) from the University of Basel, Switzerland. She co-edited more than ten books containing articles and case studies on successful business software projects. Her research interests include all kinds of Enterprise Systems, especially ERP Systems and Enterprise Collaboration Systems (ECS).

www.shortbio.net/schubert@uni-koblenz.de

Johannes H. Glitsch
Johannes Glitsch is a Ph.D. Student and Research Assistant in the Business Software Research Group at the University of Koblenz-Landau, Germany. He holds a Master's degree in Information Management from the University of Koblenz-Landau with stays abroad at the University of South Australia (Adelaide, Australia) and Turku School of Economics (Turku, Finland). His research interests are on Enterprise Collaboration Systems (ECS) and Business Process Management (BPM). His current research focus for the Ph.D. thesis aims at understanding current practices in the use of ECS by means of use cases and collaboration scenarios.

www.shortbio.net/jglitsch@uni-koblenz.de