

Process-Aware Task Management Support for Knowledge-Intensive Business Processes: Findings, Challenges, Requirements

Nicolas Mundbrod, Manfred Reichert
Institute of Databases and Information Systems
Ulm University, Germany
Email: {nicolas.mundbrod, manfred.reichert}@uni-ulm.de

Abstract—The operational support of knowledge-intensive business processes constitutes a big challenge. In particular, this process type is characterized as non-predictable, emergent, goal-oriented, and knowledge-creating. Today, knowledge-intensive business processes are solely driven by professionals utilizing their skills and expertise whereas no support is provided by process-aware information systems. In particular, the management of knowledge-intensive tasks (i.e., defining, updating, distributing, enacting, monitoring and assessing of tasks) is still accomplished manually by the knowledge workers based on pen and paper with the well-known drawbacks. This work presents two case studies of knowledge-intensive business processes. Based on the insights gained from these studies, we derive the core challenges as well as the requirements for an process-aware information system supporting knowledge workers in the management of the tasks emerging in the context of knowledge-intensive business processes.

Keywords—*knowledge-intensive business process, collaborative knowledge work, emergent knowledge process, task management*

I. INTRODUCTION

Peter F. Drucker described the economic relevance of the proper support of knowledge workers, who are involved in knowledge-intensive business processes (KiBPs), as follows: *The single greatest challenge facing managers in the developed countries of the world is to raise the productivity of knowledge and service workers* [1]. Substantiating this statement from 1991, a structural shift from an industrial towards a knowledge-based society has been taking place in developed countries. In particular, knowledge-intensive work has been becoming the predominant type of work [2], [3]. More than before, KiBPs residing in the companies' most sensitive business areas (e.g., research, development or service) determine overall business success. Hence, involved knowledge workers (e.g., researchers, engineers or physicians) utilize their distinguished skills, experiences, and expertise to cope with sophisticated and non-routine tasks every day.

A. Problem Statement

Nowadays, KiBPs and therein involved knowledge workers have not been supported at the operative level by any kind of process-aware information systems so far. Obviously, this drawback is related to the challenging characteristics of KiBPs. In particular, KiBPs are *non-predictable, emergent, goal-oriented, and knowledge-creating* [4]. Several researchers

even consider the problem of supporting KiBPs as the holy grail of information system (IS) research [5]. In fact, the degree of flexibility offered by traditional process-aware information systems (PAISs) is limited as they need to ensure an error-free and sound process execution during run time [6]. Generally, most PAISs rely on the principle of defining process models at design time and enacting corresponding process instances during run time. This principle contrasts with the characteristics of KiBPs, which utterly grave for alternating design and run times and, hence, high flexibility during run time.

B. Contribution

In the scope of a long-term research project, we aim at designing a new type of IS called *proCollab*¹. The latter is supposed to actively *guide* knowledge workers in a process-aware and task-based manner in the scope of the KiBPs they are involved in. To successfully design *proCollab*, we conducted several case studies with the aim to deepen our understanding of real-world KiBPs. Two representative studies, originated in the automotive and health care sectors, are presented in this paper. We condensed our insights considering the coordination habits, procedures, and tools into a set of key findings. Thereby, we humbly relearned something obvious: *tasks* constitute the most central objects for knowledge workers when it comes to coordination issues in KiBPs. The combination of our findings with our existing knowledge about KiBPs enabled us to denote eight key challenges for the successful design of *proCollab*. In particular, these challenges may be leveraged as a valuable foundation for any future research aiming at improving the operational support of KiBPs. To further expedite this undertaking, we introduce a set of requirements directly derived from the presented challenges as well as we unveil the valuable interdependencies between the mentioned findings, challenges, and requirements.

The remainder of this paper is organized as follows: Section II summarizes related work on KiBPs and discusses the applied methodology. In turn, III presents the case studies whereas Section IV summarizes the core findings we could derive from prior work as well as the case studies' analysis. Subsequently, eight key challenges for designing *proCollab* will be discussed in Section V. Drawing upon, Section VI unveils the key requirements of *proCollab* in the same way. Finally, Section VII concludes the paper with a summary and outlook.

¹Process-aware Support for Collaborative Knowledge Workers

II. FUNDAMENTALS

This section deals with related work on KiBPs as well as the methodology applied in the context of our research. Both provides a necessary basis for the subsequent sections III-VI.

A. Previous and Related Work

As known from other research fields, KiBP notions and definitions significantly vary in relation to on KiBPs. Notions also denoting KiBPs include (*collaborative*) *knowledge work* and *emergent knowledge processes*. A detailed discussion of different KiBP notions and related definitions can be found in [7]. For the sake of simplicity, this paper uses the notion of *knowledge-intensive business processes* based on [8]:

“Knowledge-intensive processes (KiBPs) are processes whose conduct and execution are heavily dependent on knowledge workers performing various interconnected knowledge intensive decision making tasks. KiBPs are genuinely knowledge, information and data centric and require substantial flexibility at design- and run-time.”

In previous work [4], we analyzed existing literature as well as three different use cases for KiBPs. In this context we derived core KiBP characteristics, which constitutes as follows:

Non-predictable: KiBPs address complex and dynamic situations comprising an unstable set of unmanageable as well as varying influencing factors intertwined via dynamic correlations. In particular, this circumstance involved process participants are compelled to deal with continuous uncertainty and makes it impossible for them to foresee the process in all details upfront.

Goal-oriented: Coping with uncertainty, a common goal is used by the involved knowledge workers as a solid basis for aligning activities and resources. To ease the achievement of the common goal, sub-goals may be defined.

Emergent: In connection with uncertainty, knowledge workers need to regularly evaluate accomplished work as well as the given situation (e.g., available resources) to plan the actions to be performed in the following. Thereby, KiBPs comprise alternating phases of planning and operative working and, hence, gradually emerge toward the goal the knowledge workers want to achieve.

Knowledge-creating: Knowledge workers, participating in KiBPs, leverage their expertise and experiences to effectively achieve the process goal. Typically, they create explicit knowledge artifacts through intermediate and final work results. Finally, they expand their tacit knowledge base during the process run time.

In relation to the presented characteristics, Markus et al. [5] rely on ten KiBP characteristics whereas Di Ciccio et al. [7] denote eight KiBP characteristics. Nonetheless, the authors believe that these characteristics can be reduced onto the four presented above. The following Figure 1 exposes the KiBPs characteristics and their coherence visually.

In previous work [4], we further proposed a KiBP Lifecycle as an essential foundation for every IS aiming at the support

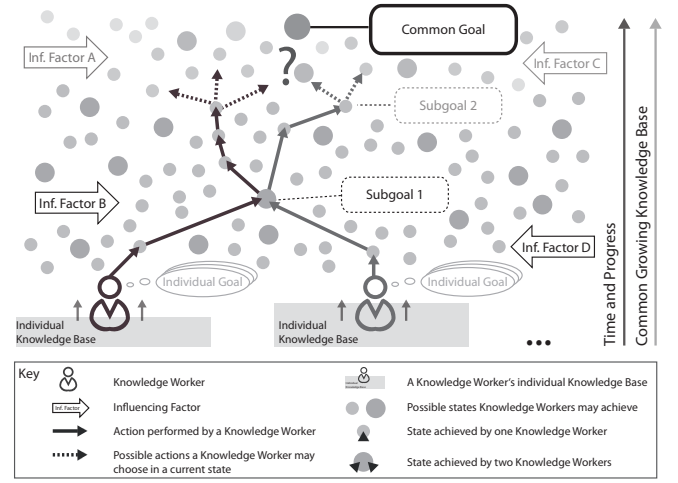


Fig. 1. Characteristics of KiBPs and their Coherence

of knowledge workers involved in KiBPs. Figure 2 exposes the KiBP lifecycle in a compact illustration.

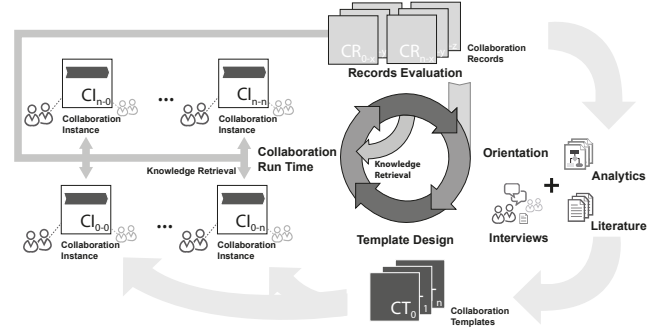


Fig. 2. KiBP Lifecycle

As some of the challenges and requirements directly base on a lifecycle support, the lifecycle phases are recapped in short in the following:

Orientation: In this first phase, information about the KiBP is collected. Based on interviews, analysis and existing literature a process description is compiled.

Template Design: Drawing upon this description, a *collaboration template* (CT) is defined for the respective KiBP. A CT comprises all those coordination artifacts which are likely employed by the knowledge workers during run time. Created CTs are then offered to knowledge workers who are using the supportive IS during the collaboration run time.

Collaboration Run Time: To actively start the guidance by the IS, knowledge workers instantiate a CT to create a *collaboration instance* (CI). The CI ultimately determines the supportive guidance offered by the IS to the involved knowledge workers involved in KiBP. In the context, the CI may be continuously adjusted by the involved knowledge workers.

Records Evaluation: On the one hand, knowledge workers, who are involved in a CI, can make use of insights from *comparable* collaboration records (i.e., archived CIs). On the other hand, the analysis of collaboration records is also employed to increase the guidance during run time as well as to improve the existing CTs.

Based on the presented lifecycle principle, the emergent character of KiBPs can be addressed and collaboration-based knowledge may be maintained or even increased.

Riss et al. [9] present five meta-challenges for business process and task management which are taken into account in Section V. Moreover, Markus et al. [5] introduce ten meta-requirements and Di Ciccio et al. [7] even discuss 25 requirements for a process-aware system support of knowledge workers. Both are directly considered in Section VI in context. Finally, Belotti et al. [10] and Pryss et al. [11] discuss the usage of digital task management in the context of KiBPs. Drawing upon, the definition of the challenges and requirements are also influenced by the results of these related works.

B. Methodology

As this work is part of a long-term project targeting to design proCollab (cf. Section I), we apply the design science approach as research methodology [12]. In particular, regarding the design science research process [13], our work represents an objective-centered solution providing an answer to the question “What would a better artifact accomplish?”. Our work can be further categorized as an *explanation* theory in relation to the theory types of IS research [14].

Based on a study of prior work, we derived a number of research questions we want to investigate in case studies. Specifically, our goal is to foster the definition of challenges and requirements for proCollab:

- How do knowledge workers cope with the constant uncertainty induced by intertwined influencing factors?
- How do knowledge workers deal with goals?
- How do knowledge workers accomplish the frequent change between planning and working?
- How do knowledge workers manage the process-related knowledge?

In order to gain valid results in relation to the stated research questions, the case studies were conducted according to known scientific standards [15]. In particular, the case studies’ results have been gained during many on-premise observations, interviews, and workshops.

III. CASE STUDIES

In this section two case studies are presented based on a predefined comparison scheme. Case study CS1 presents insights from development processes of electrical and electronic (E/E) car components at a big German automobile manufacturer. The insights are gained in the context of on a long-term collaboration: detailed information are available (e.g., [16], [17]). In turn, case study CS2 presents insights from patient treatment processes in a medical unit of a university hospital. Again, insights are based on profound experiences

we have in this domain [18], [19], [11]. In both cases, we categorized the processes as knowledge-intensive since they clearly expose the characteristics presented in Section II-A.

A. Automotive E/E Development Processes (CS1)

Goals: The common goal is to develop a certain E/E car component, defined by its specification, until a fixed release date. Sub-goals (often called *milestones* or *quality gates*) are hierarchically derived based on the used methodology (cf. Figure 3). Further, the sub-goals expose temporal constraints, e.g., “to be achieved 12 months before release”.

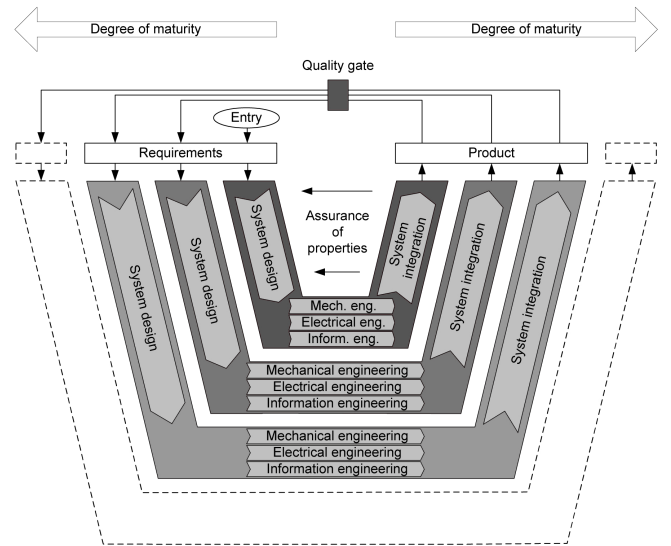


Fig. 3. V-model for Mechatronic Engineering, according to [20]

Duration: Depending on the specific goal, an EE development project lasts between three and four years in average.

Participants: Usually, hundreds of engineers from different disciplines and partners (i.e., OEM and suppliers) are involved in an E/E engineering project throughout the various phases. Due to high specialization, most of the projects are conducted together with associated suppliers. While there are professionals having central and static project roles (e.g., “person with system responsibility”), many people or even entire organizations are assigned to the project on demand. Professionals take care of special tasks (e.g., integration tests). Hence, they may participate in several projects concurrently.

Resources: The knowledge workers involved in an E/E engineering project particularly deal with an abundance of digital information respectively data sources, e.g., best practices, standards, presets, regulations and requirements. Furthermore, intermediate development results, like requirement specifications, circuit diagrams, wiring diagrams, or entire software, need to be shared and managed properly to ensure an effective progress in the project. Additionally, hardware prototypes are developed in the scope of a project as proof-of-concepts to eventually validate digitally simulated concepts by series of physical tests [17].

Process: To ensure effective E/E development involving hundreds of stakeholders, the project members must follow a well-defined development methodology (i.e., the V-Model (cf. Figure 3)). As a development project may expose many sequential phases, the methodology is applied to each phase individually. Usually, each phase is connected to a quality gate (i.e., a sub-goal). Obviously, development phases may again comprise several sub-phases and many different and parallel development processes that have to be managed by the project management as well.

There are numerous coordination meetings to continuously synchronize the work results produced by the project members. In addition, project members asynchronously and frequently communicate with each other (e.g., via email or face to face) to share results. Furthermore, the project manager moderates discussions and keeps track of the project by applying and using project management standards as well as tools to document the progress and to manage exceptions.

To foster the quality of development processes, to ensure compliance with regulations, and to track the development progress, a dedicated *checklist-based guideline* is initially created and, then, continuously managed by a quality assurance officer. Usually, the quality assurance officers regularly discuss the currently relevant items with the engineers in the scope of an interview. As the project proceeds, the checklist symbolizes the project progress. Stepwise passing quality gates, an EE component gradually becomes more mature and finally enters the testing stage before being released for using in a product (i.e., a car).

B. Patient Treatment Processes (CS2)

Goals: A patient treatment process ultimately comprises the goal to cure a patient from his or her ailments. Referring to the different phases of the process, several identifiable sub-goals (e.g., first diagnosis) exist.

Duration: The duration of a patient's treatment in a hospital cannot be always foreseen. Instead, it ultimately depends on the patient's health conditions and the physicians' decisions.

Participants: Based on the patient's diagnoses, various stakeholders are involved in patient treatment processes. However, when admitting patients to the hospital, standard procedures with a predefined set of staff (roles) can be assumed. Afterwards, a patient is allocated to the medical unit that can optimally deal with his or her predominant disease. There, the physicians drive the progress of a patient's treatment process while care attendants are involved as well. In every shift, there always is one physician who is in charge of taking care of a group of patients at a medical unit.

Resources: In the context of a single patient treatment process, the most central information object is represented by the patient's medical record. Ideally, such a medical record contains the complete medical history and care documentation of the patient within a particular health care organization across time. The nursing record, representing the progress in a patient's case, includes an abundance of information entered

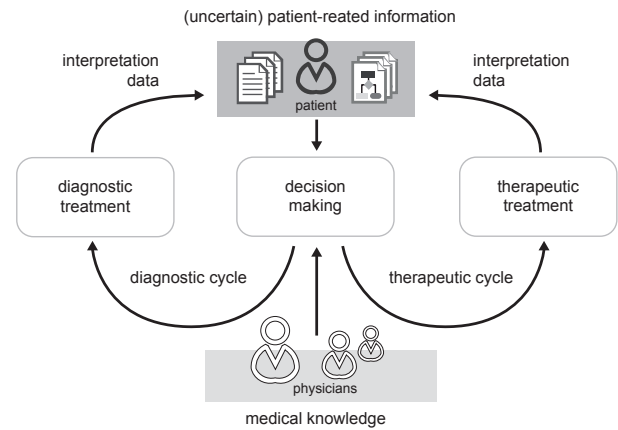


Fig. 4. Diagnostic-therapeutic cycle, according to [19]

over time, like simple notes, observations, administrated drugs and therapies, orders for administration of drugs and therapies, test results, x-rays, reports, etc.

Process: In the very first stages of a patient's treatment process, an initial diagnosis is made through a first anamnesis during admission. Based on this diagnosis, the patient is transported to a specialized medical unit that can address his or her main, most severe ailment best. Once the patient arrives at a specialized medical unit, the physicians on-site then take care of the patient's treatment process. Furthermore, there is always one physician in every shift who directly takes care of one patient. Physicians from other specialized medical units are only involved on demand—this is called consultations.

Typically, physicians implicitly apply a methodology called *diagnostictherapeutic cycle* (cf. Figure 4) to cure the diseases or at least to alleviate the symptoms a patient suffers from. The involved physicians continuously assess the patient-related information based on their medical knowledge. In doing so, they derive decisions regarding further diagnostic and therapeutic treatments of the patient. The application of these treatments then generates new patient-related information and may increase the medical knowledge of involved physicians. Based on new insights gained from treatments, the assessment starts over again.

In most medical units every day, physicians conduct medical rounds to assess the current health state of their patients. Further, they thereby derive medical orders regarding diagnostic and/or therapeutic treatments. These orders and the drug administration are documented in the patient record and the (care) progress report continuously. Additionally, physicians and care attendants hold meetings in which they openly discuss the patients' health conditions without directly facing them. Typically, there is also a scheduled handover to exchange important information among physicians working in different shifts. Thereby, the physicians describe, for example, important patient-related events and insights to the incoming set of physicians. Apart from regular handovers concerning the cases of many patients, physicians schedule individual meeting in relation to a certain patient case on demand.

IV. FINDINGS

This section discusses the key findings we derived from existing literature (cf. Section II-A) and the presented case studies CS1 and CS2 (cf. Section III). In particular, we focus on the aspects relevant for enabling coordination among the knowledge workers involved in the cases CS1 and CS2. Thereby, we respond to the research questions raised in Section II-B. The following Figure 5 depicts the Findings F1-F14 in an illustration based on CS1.

Goals (F1): In both cases, we could observe clear and pretty static common goals the involved knowledge workers align their work to. However, the common goal in CS2 is not communicated explicitly since it is the same for every patient. By contrast, the definition of sub-goals differs: in CS1, stable quality gates and milestones are set early to ensure high quality at the entire E/E engineering project. Both are linked to fixed dates and directly derived from the overall goal. More fine-grained sub-goals are then defined for every development process on demand. The most dynamic, intermediate sub-goals, defined by the development teams themselves, can be found at the operational level of the project. In comparison, physicians do not define any static milestones—they strongly use dynamically created intermediate sub-goals instead.

Methodologies (F2): While in both cases a methodology is used, it is applied only in CS1 explicitly. Due to the amount of stakeholders involved in an E/E component development project, a dedicated project management team takes care of the project coordination. Furthermore, a quality assurance officer, responsible for several projects, interviews project members in order to ensure compliance (cf. F7). In comparison, a patient is usually supervised by the physician in charge who coordinates the necessary actions by applying the aforementioned diagnostic-therapeutic cycle implicitly. In general, this cycle is a robust, reliable and quite applicable way to deal with the challenging characteristics of knowledge-intensive treatment processes in hospitals.

Tasks (F3): As we could observe, tasks are the most central objects for knowledge workers on both the organisational and the personal level. In CS1, dedicated systems and a lot of Excel sheets contain thousands of tasks managed by the project management team and accomplished by the engineers—typically grouped in checklists (cf. F6). Tasks or groups of tasks are semantically connected to the pursued process milestones and are annotated with contextual information, e.g., names of persons supposed to accomplish the tasks. Especially during meetings (observable in both cases), the statuses of current tasks are discussed and updated. Further, new tasks are discussed, defined, updated, subdivided into more fine-grained tasks or assigned to readjust plannings in relation to the pursued sub-goals.

Events (F4): The occurrence of one or several events often triggers the definition of new tasks or the updating of existing ones. An example is the successful accomplishment of several tasks, which, in turn, typically initiates further planning resulting in new tasks. In CS1, the completion of integration tests for developed E/E components may trigger tasks related to the release of the components. However, there

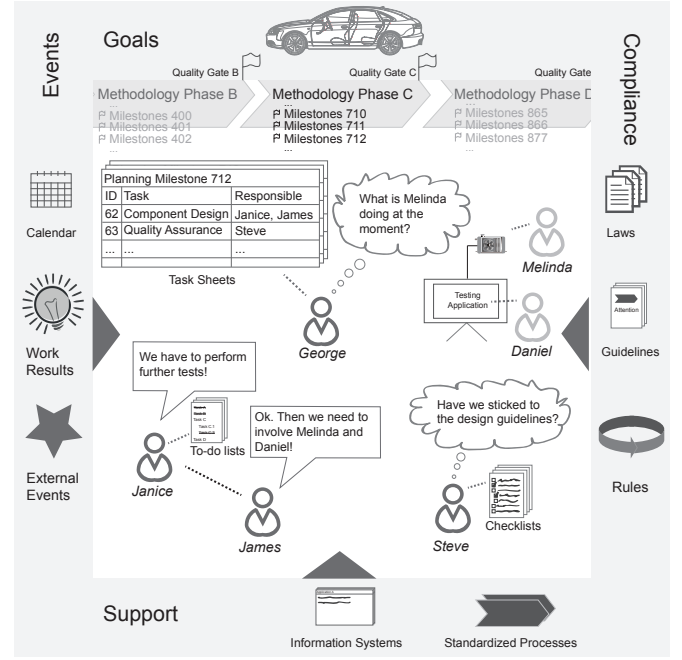


Fig. 5. Overview of Findings F1-F14, based on CS1

exist many external events triggering tasks for knowledge workers. For example, consider CS2: if a patient exposes new symptoms, the physician in charge might trigger new diagnostic actions (encoded as tasks). Finally, even missing events (e.g., missing results from a blood test) might enforce knowledge workers to define new tasks.

To-do Lists (F5): Especially for personal task management, to-do lists are a common task-oriented artifact everybody knows [21]. In both cases we could observe that the involved knowledge workers rely on to-do lists for personal usage. In CS2, e.g., many physicians just jot down their personal tasks during ward rounds on paper-based to-do lists (cf. [11]). After completing the ward round, a physician stepwise accomplishes the defined tasks and usually highlights the accomplishment on the to-do list (e.g., by a check). As soon as all tasks are accomplished and documented in dedicated records, to-do lists are disposed.

Checklists (F6): In both cases, we found checklists as another task-oriented artifact used by the knowledge workers. While to-do list are used for prospective planning, checklists are leveraged to assess work in a retrospective manner. As mentioned in CS1, there is a checklist-based development guideline. The questions are discussed during interviews between a quality officer and the engineers to improve the development quality (cf. F2) and to ensure compliance (cf. F7). In CS2, standardized paper-based checklists are used, e.g., in preparation to surgeries with the goal to reduce the risk of complications during a surgery. In general, checklists are becoming increasingly popular tools for quality assurance in hospitals [22].

Compliance (F7): The compliance to a wide range of different regulative constraints like laws, codes of conducts,

(corporate) guidelines, and best practices is an important aspect for the knowledge workers involved in CS1 and CS2. Thereby, defined tasks and their induced accomplishments are limited by the need of complying with rules which are often only encoded in a narrative way, e.g., in documents. In this context, mentionable regulations include, e.g., security precautions (e.g., signatures), confidentiality or traceability. While engineers and physicians both know numerous rules based on experiences and education, there still exist hundreds of rules they are unaware of regarding details. Hence, the application of a methodology (cf. F3) and checklists (cf. F6), are also strongly motivated by the need for compliance (especially in CS1).

Communication (F8): Communication constitutes an indispensable part of task management in the context of KiBPs. We could observe knowledge workers in CS1 and CS2 frequently communicating with each other in various ways (e.g., face to face, in meetings, via phone or e-mail). The purpose of this communication is to discuss tasks, to refine tasks or to delegate tasks. Furthermore, this informal part of task management substantially increases the knowledge workers' understanding of the tasks' purposes, parameters and constraints, which finally increases effectiveness and efficiency. Considering planning, the communication among knowledge workers is observable in the shape of deliberations and trade-offs regarding alternatives—it therefore strongly influences the *direction* KiBPs emerge to.

Documentation (F9): We observed that documentation is essential for knowledge workers to work and plan effectively. Both in CS1 and CS2, process-related tasks and related artifacts are documented to a certain extent in order to foster the planning of future tasks. In CS1, the project management team centrally takes care of documentation, whereas in CS2 physicians and nurses typically document changes in the respective records. As documentation is performed widely manually, this is an obvious source of errors (e.g., forgetfulness, incompleteness or lack of precision). Moreover, manual documentation is time-consuming (e.g., in CS1, there are huge paper-based wallpapers providing an “overview” on running development processes) and, hence, many knowledge workers just waive fine-grained documentation as they do not recognize any intermediate benefits. Finally, we observed a quite problematic distributed, redundant and sometimes even inconsistent documentation in both cases.

Dynamic Teams (F10): In both CS1 and CS2, we observed knowledge workers joining and leaving processes on demand. Thereby, many knowledge workers concurrently participate in several KiBPs which let them face problematic context switches frequently. Especially in CS2, the inclusion of physicians is mainly determined by the patient's health conditions. If, for example, a surgery is needed based on accordingly diagnosed conditions, a team of surgeons and surgery nurses will be dynamically involved in the patient's treatment process. In addition, we recognized at least one knowledge workers driving a KiBP at any time—in CS1, it is even a stable project management team supervising the KiBPs. In addition, knowledge workers typically act in KiBPs based on functional roles, e.g., chief resident in CS2.

Awareness (F11): Awareness² about where experiences and expertise reside is another crucial factor for knowledge workers. In CS1, the project management team continuously monitors and updates the statuses of processes (and even tasks in the processes) manually. Furthermore, they explicitly manage information about who is responsible for which tasks and who is doing what and where. Moreover, the awareness of events (e.g., the availability of expected work results) also plays an important role in both CS1 and CS2. Nonetheless, the manual gathering of awareness information is very time-consuming and error-prone—this fact generally results in incomplete and fuzzy awareness information automatically leading to ineffective planning and working (i.e., additional and frequent readjustments of existing plans and work).

Supporting Processes (F12): In both CS1 and CS2, the involved knowledge workers invoke many supporting business processes in the context of KiBPs. Further, they receive outputs from respective business processes. For example, there are standardized business processes for blood tests or X-ray examinations in CS2. Unfortunately, knowledge workers are hardly supported regarding status updates (“has the laboratory successfully received a blood sample?”; cf. F11) such that process results might be delivered inefficiently. For example, physician A requests a blood test for patient X. The result will be later send to A automatically although another physician B might take care of X and A might be on leave.

Existing IT Support (F13): There is an abundance of standard software and dedicated ISs provided to the knowledge workers in both CS1 and CS2. For coordination issues, especially communication software, office applications and dedicated ISs (e.g., hospital information system in CS2) are jointly used on a daily basis by the knowledge workers. In general, there is no process-aware integration of the software offered and, hence, knowledge workers manually interconnect functionality. Considering coordination in particular, tasks are often managed on paper-based and electronic systems overlapped or even in parallel. This problem results in media breaks, inconsistencies, redundancy and ineffectiveness.

Reutilization (F14): As far as we could observe, there is no analysis of the KiBPs taking place in CS1 and CS2 based on process-related records. The distributed documentation (cf. F9) of tasks creation, delegation and accomplishment as well as a lack of an integrated task management (cf. F8 and F13) do not allow for any analysis of future optimizations. Especially, the business process instances are not analyzed based on process models, which may provide a retrospective view on the accomplished tasks and their temporal relationship, due to this lack of integrated information. Based on the current situation existing in the context of CS1 and CS2, an effective lifecycle for KiBPs (cf. Section II-A) is provided to the involved knowledge workers.

²(Group) awareness is “the up-to-the-minute knowledge of other peoples activities that is required for an individual to coordinate and complete their part of a group task.” [23]

V. CHALLENGES

This section derives the key challenges for process-aware task management support in the context of KiBPs. Definition and description of the following eight challenges are directly based on the Findings F1-F14 (cf. Section IV) as well as related work (cf. Section II-A). The following Table I presents the relation between the following Challenges C1-C8 and the Findings F1-F14.

TABLE I. RELATION OF CHALLENGES AND FINDINGS

ID	Challenge Name	Related Finding(s)
C1	Meta-Model Design	Goals (F1), Methodologies (F2), Tasks (F3), To-do lists (F5), Checklists (F6), Compliance (F7)
C2	Lifecycle Support	Existing IT Support (F13), Reutilization (F14)
C3	Variability Support	Tasks (F3), Existing IT Support (F13), Reutilization (F14)
C4	Context Support	Tasks (F3), Compliance (F7), Dynamic Teams (F10), Awareness (F11), Existing IT Support (F13)
C5	View Support	Tasks (F3), To-do lists (F5), Checklists (F6), Existing IT Support (F13)
C6	Authorization Support	Tasks (F3), To-do lists (F5), Checklists (F6), Dynamic Teams (F10)
C7	Synchronization Support	Tasks (F3), Events (F4), To-do lists (F5), Checklists (F6), Communication (F8), Documentation (F9), Awareness (F11)
C8	Integration Support	Communication (F8), Documentation (F9), Awareness (F11), Supporting Process (F12), Existing IT Support (F13)

Meta-Model Design (C1): The first and probably most essential challenge is to design an appropriate, sound and powerful meta-model for proCollab enabling knowledge workers to define the kind of support they require. F3, F5, and F6 clearly underlined that the meta-model especially needs to be task-centric. Further, the meta-model symbolizes the most essential foundation for proCollab as it finally comprises a modeling language as well as run time semantics of modelled entities. However, in comparison to meta-models invented by the BPM community (e.g., [24]), the proCollab meta-model has to provide agile support in relation to the characteristics of KiBPs. In particular, this comprises the alternating design and accomplishment of tasks.

Lifecycle Support (C2): Motivated by F14 and strongly connected to C1, another challenge is to establish a continuous and powerful lifecycle support in proCollab with the goal to guide the knowledge workers involved in KiBPs in a sustainable way. Regarding the basic lifecycle presented in Section II-A, this challenge can be separated into four more detailed challenges. The first sub-challenge is the definition of beneficial CTs. In this context note that every KiBP instance is unique regarding its flow of defined and accomplished tasks (cf. Section II-A). Further, the system needs to continuously appraise the current state of a CI during run time (second sub-challenge) to offer appropriate support in the shape of recommendations—this approach directly addresses the emergent character of KiBPs. The analysis of completed CIs (third sub-challenge), which are based on a common CT, can then be leveraged to improve this CT or to even create new CTs. To enable such improvements, future research needs to address the way completed CIs can be effectively archived in the shape of CRs as well as how these CRs can be prepared

and analyzed appropriately (forth sub-challenge).

Variability Support (C3): If lifecycle support is thought to the end, the analysis of completed CRs to possibly improve or create new CTs will result in another challenge—to deal with a high variability of CTs. Furthermore, the granularity of CTs also comes into play as knowledge workers either might have to subdivide tasks a lot during run time or they might feel patronized by too many detailed tasks. Finally, knowledge workers must be enabled to find the right CT in order to be optimally supported during run time—therefore, the challenge need to be addressed how knowledge workers may find the right CT for their purpose.

Context Support (C4): As observed in the context of CS1 and CS2, KiBPs must be assessed and supported in relation to their context. In particular, the KiBPs in CS1 are organizationally integrated into a development project, involving different research departments. In comparison, a certain patient treatment process typically belongs to a medical case of a medical unit in a hospital. To optimally support KiBPs, future research needs to address the challenge of representing and updating contextual parameters and rules in order to provide a specific support and to ensure compliance (cf. F7). Furthermore, the knowledge workers' roles, abilities, and affiliation also need to be represented and updated to allow them for effectively planing collaborative work in correspondence to dynamic teams (cf. F10 and C7) and awareness issues (cf. F11).

View Support (C5): We observed for both CS1 and CS2 (cf. F3, F5, and F6) that knowledge workers struggle with a lack of personalized views on (huge) sets of tasks. As soon as there is a considerable number of tasks with different responsibilities, descriptions, priorities, and other additional information, it might be time-consuming, frustrating, and simply inefficient for knowledge workers to find the tasks personally relevant for them. Hence, future research must address the challenge of defining, updating and providing personalized views on CIs for knowledge workers based on their personal preferences, their roles, and the their context (cf. C4).

Authorization Support (C6): In relation to Challenges C1, C4, and C5, the challenge of an expressive authorization model, coping with dynamic teams (cf. F10) and the possible emergent definition of responsibilities, must be addressed. Since knowledge workers, who may have different educational and vocational backgrounds, need to dynamically define access to tasks while delegating them, an authorization model need to be understandable and expressive at the same time. This obvious contradiction might implicate severe problems for the definition of an authorization model.

Synchronization Support (C7): The synchronisation of CIs with actions performed or events occurred in real-world must be regarded as a severe challenge, too. As highlighted in Findings F3, F8, and F9, tasks are often defined and delegated based on communication among knowledge workers—this results in a lack of documentation. Further, task accomplishment is documented in heterogeneous or distributed records and ISs. As a consequence from these habits, CIs

might suffer from an incomplete task-related information. In turn, this may result in problematic consequences in the shape of deceptive analysis of CRs, irritating recommendations for knowledge workers during run time or in erroneous adjustments of existing CTs.

Integration Support (C8): As discussed in F12, a KiBP may directly correlate with numerous prespecified and standardized business processes. Hence, the challenge is to tightly integrated PAISs to enable knowledge workers to launch standardized business processes, to obtain status updates regarding the progress of corresponding business processes and to receive expected outputs from the standardized business processes.

VI. REQUIREMENTS

This section presents 25 key requirements for process-aware task management support in the context of KiBPs. These requirements were derived from the presented related work (cf. Section II-A), findings (cf. Section IV), and challenges (cf. Section V). As we naturally cannot present a huge set of all requirements necessary regarding the process-aware task management support for KiBPs (e.g., usability aspects), we hence focus on the most essential ones in the following. Providing a further benefit, Table II exposes the challenges in relation to the derived requirements and the discussed findings.

TABLE II. RELATION OF CHALLENGES, DERIVED REQUIREMENTS AND RELATED FINDINGS

ID	Challenge Name	Derived Requirement(s)	Related Finding(s)
C1	Meta-Model Design	R1-R5	F1-F3, F5-F7
C2	Lifecycle Support	R6-R11	F13, F14
C3	Variability Support	R12-R15	F3, F13, F14
C4	Context Support	R16-R17	F3, F7, F10, F11, F13
C5	View Support	R18-R19	F3, F5, F6, F13
C6	Authorization Support	R20	F3, F5, F6, F10
C7	Synchronization Support	R21-R23	F3-F6, F8, F9, F11
C8	Integration Support	R24-R25	F8, F9, F11-F13

Requirements R1-R5 are related to Challenge C1:

Task-centric entities (R1): The proCollab meta-model must cover tasks as knowledge workers rely on them as central entities for planning and performing their work (cf. F3, F5, and F6). Hence, the meta-model must enable knowledge workers to define tasks as well as their relevant parameters, e.g., task goal (cf. F1), task name, task description, task author, task priority, persons in charge, and scope of the task. Additional entities (cf. R6, R16, and R19) are supposed to be aligned to tasks or set of tasks respectively (cf. R2)

Task Dependencies (R2): Based on the Findings F2, F3, F5 and F6, the meta-model should allow for the definition of dependencies between tasks. In particular, hierarchical and temporal task dependencies are relevant in the context of KBPs. The provision of a well-defined set of hierarchical dependencies enable knowledge workers to define *task trees*. In turn, the latter are required as a basis for providing *to-do lists* and *checklists*. Furthermore, temporal dependencies

between tasks need to be covered in order to enable knowledge workers to constrain the execution of their tasks, e.g., to express temporal producer-consumer relationships. As a result, a *task structure* denotes a set of tasks (at least one task) with their hierarchical and temporal dependencies.

Adaptive Task Structures (R3): Due to the emergent character of KiBPs, knowledge workers should be able to continuously change task structures (cf. R2). For this purpose, a set of well-defined operations is required that allow transforming a sound task structure into another sound task structure. To foster the knowledge workers efficiency, low-level change operations (e.g., the removal of a task) should be aggregated to high-level change operations (e.g., to swap tasks) to increase the knowledge workers' convenience.

Meta-Model Comprehensibility (R4): As discussed, every knowledge workers should be able to adjust CIs (cf. R3). Hence, the meta-model need to be comprehensible for them. This requirement suggests limiting the number of the meta-model's entities and relations significantly. However, a well-maintained modularity of meta-model might result in the right balance between the expressiveness and comprehensibility.

Meta-model Extensibility (R5): The meta-model must be built in a modular and extensible way to allow for (domain-related) customizations in order to cover newly needs of knowledge workers as well as to cope with emergent requirements. In particular, extensions proposed by the following requirements R6, R16, and R19 can be regarded as modules on the meta-model as well.

Requirements R6-R11 are related to Challenge C2:

Lifecycle Entities (R6): In order to enable full lifecycle support of KiBPs, the proCollab meta-model needs to be enriched with entities for CTs, CIs, and CRs. Based on this, the meta-model can offer the necessary foundation for the advanced lifecycle analysis (e.g., the improvement of CTs). Furthermore, appropriate functions for instantiating CTs as well as for archiving completed CIs as CRs (cf. R7) need to be provided as well.

Collaboration Instance Archiving (R7): In order to effectively plan upcoming tasks, knowledge workers need the history of performed changes (cf. R11) in relation to task structures as well as CTs and CIs. Hence, proCollab needs to provide a powerful concept for archiving CTs and CIs in order to enable knowledge workers to access valuable information on demand.

Collaboration Schema Evolution (R8): proCollab must enable schema evolution as known from adaptive PAIS [25]: i.e., as soon as there is an update performed of a CT, proCollab shall try to migrate depending CIs, derived from this CT, to its new version. However, the current state of an CI need to be taken into account and, hence, some CIs might expose a state which does not allow any updates any more.

Collaboration Instance Generalization (R9): proCollab should enable knowledge workers to generalize one CI or a collection of CIs into a CT. For example, some knowledge

workers adjust a CI to such an extent during a KiBPs that they want to transform this CI into a CT for future endeavors. In turn, this enables knowledge workers to directly deposit templates for future purposes and thereby might increase the commitment of the knowledge workers to the system (cf. R21).

Run-time Recommendations (R10): Based on the current state of a CI and the knowledge gathered from comparable CRs (e.g., CRs derived from the same CT as the current CI), proCollab shall provide planning support to knowledge workers in the shape of task recommendations. However, future research needs to address how these recommendations can be offered without patronizing users instead. *Version Control (R11):* As KiBPs gradually emerge, CTs and CIs will be subject to frequent changes conducted by the knowledge workers involved. As discussed in the context of R7, the planning of new tasks may comprise the assessment of historical data. Therefore, CTs and CIs need to be managed by version control. Further, the logged information is required for the analysis of CRs with the aim to continuously improve CTs. Finally, it allows knowledge workers to undo changes, too.

Requirements R12-R15 are related to Challenge C3:

Collaboration Template Families (R12): proCollab should enable the definition of CT families to address variability issues. In particular, a CT family comprises related CTs (e.g., based on semantic parameters) including a reference CT together with related variants. The latter may only contain differences in relation to reference CT. Hence, updates on CTs may then be performed more effectively by adjusting only the reference CT instead of a huge set of CTs individually.

Collaboration Template Configuration (R14): proCollab needs to provide a CT configuration concept to allow knowledge workers to configure a template according to their needs before instantiating it. This might be solved by contextual parameters deposited in the templates. Furthermore, proCollab may automatically configure a CT based on existing context information in the system before instantiating it.

Collaboration Template Guidance (R13): Knowledge workers should be assisted in defining CTs. Regarding the granularity and reuse of existing CRs, proCollab must provide proper wizards, tutorials, and documentation to ease the definition of CTs.

Collaboration Template Annotation (R15): To enable knowledge workers to effectively and efficiently search for the CTs needed, proCollab must allow knowledge workers to annotate CTs with additional meta information. In turn, this meta information helps categorizing and filtering templates according to search parameters.

Requirements R16-R17 are related to Challenge C4:

Context Model (R16): In order to enable context-specific support of knowledge workers, the proCollab meta-model (cf. R1 and R5) must be extended by a dedicated context model. Thereby, contextual information must be captured to enrich the support of KiBPs and involved knowledge workers. As an example, the context model can extend the organisational

model, which is likely part of the basic meta-model, with awareness-related information.

Context Rules (R17): According to F7, knowledge workers should be enabled to deposit contextual rules in proCollab. Based on the current context of knowledge workers, the latter may be leveraged to adjust the provided guidance provided by a CI. Therefore, proCollab need to allow knowledge workers to define context rules for CTs and to enable knowledge workers to change these rules on demand during run time.

Requirements R18-R19 are related to Challenge C5:

Personal Views (R18): Comparable to views on business processes [26], proCollab must provide personal views on CTs, CIs, and CRs. A personal view may provide only those tasks relevant for a particular knowledge worker. For example, a personal view may comprise exactly those tasks a knowledge worker is formally responsible for. However, related tasks (e.g., preceding tasks offering inputs for personal tasks) may have to be taken into account and be displayed appropriately. Finally, the access to tasks and other entities must be limited based on an authorization model (cf. R20).

Context-specific Views (R19): proCollab should further support context-specific views to optimally support knowledge workers. Thereby, context parameters and context rules are leveraged to adjust provided views on CTs, CIs, and CRs. Therefore, views may be initially annotated with configuration parameters directly influencing the view and representing the complement to contextual parameter (cf. R16 and R17).

Requirement R20 is related to Challenge C6:

Authorization Model (R20): To ensure privacy, confidentiality and compliance, the proCollab meta-model (cf. R1 and R5) must be enriched with a dedicated authorization model. In turn, this model shall enable knowledge workers to define the access to the entities of CTs, CIs, and CRs based, e.g., on roles. Generally, the authorization model is strongly interconnected with the context model (cf. R16). Furthermore, the authorization model must be comprehensible since every knowledge workers may have to configure authorization rules during run time.

Requirements R21-R23 are related to Challenge C7:

User Encouragement (R21): proCollab should encourage knowledge workers to steadily update the status of their work, i.e., their tasks. Although this sounds odd, concepts regarding this requirement are requested since frequent status updates are indispensable for a successful lifecycle (cf. C2). As many knowledge-intensive tasks are solely performed by knowledge workers manually, the status updates can be only provided by themselves.

Mobility (R22): Mobile access to proCollab is an essential requirement to further increase user commitment as well as to maintain lifecycle functionality (cf. R21 and C2). As knowledge workers may work at different locations, platform-independent applications for pervasive mobile devices (e.g., *smartphones*) are required.

Events (R23): With the goal of increasing the awareness of knowledge workers (cf. F11), proCollab must filter and finally dispatch occurring events (e.g., arrival of work results) to CIs and therein involved knowledge workers. Hence, proCollab must integrate event sources and complex event processing technology (e.g., cf. [27]).

Requirements R24-R25 are related to Challenge C8:

Application Programming Interface (R24): To enable ISs to integrate proCollab functions and entities (e.g., tasks), a well-defined application programming interface is requested. Especially for engineers and developers, the integration of asks with existing development tools may provide benefits (e.g., a task-focused interface [28])

Process Integration (R25): Based on F12, a tight integration of existing PAISs is required. In the context of CIs, such integration should allow knowledge workers to launch standardized business processes, to monitor these processes and to obtain valuable results from them.

VII. CONCLUSION

Altogether, this paper presented a study of related work as well as two case studies providing valuable insights into KiBPs from the automotive and health care domains. In turn, this allowed us to derive 14 key findings regarding the coordinative aspects of KiBPs as well as to address the central research questions raised in Section II-A. These findings are then leveraged for the definition and description of eight key challenges and 25 key requirements regarding the process-aware task management support of KiBPs.

Both challenges and requirements are utilized by us for describing a meta-model comprising the entities, operations, the authorization and context models (cf. R16 and R20) as well as for the development of a first prototype. These results and insights gained from evaluations will be subject to future work.

REFERENCES

- [1] P. F. Drucker, "The new productivity challenge," *Harvard Business Review*, vol. 69, no. 6, pp. 69–79, 1991.
- [2] M. Pfiffner and P. Stadelmann, *Wissen wirksam machen*. Bern: Haupt Verlag, 1998.
- [3] I. Brinkley, "Defining the knowledge economy," *London: The work foundation*, 2006.
- [4] N. Mundbrod, J. Kolb, and M. Reichert, "Towards a system support of collaborative knowledge work," in *Business process management workshops*, ser. LNBIP, vol. 132. Springer, 2013.
- [5] M. L. Markus, A. Majchrzak, and L. Gasser, "A design theory for systems that support emergent knowledge processes," *MIS Quarterly*, vol. 26, no. 3, 2002.
- [6] M. Reichert and B. Weber, *Enabling flexibility in process-aware information systems: Challenges, methods, technologies*. Berlin and Heidelberg: Springer, 2012.
- [7] C. Di Ciccio, A. Marrella, and A. Russo, "Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches," *Journal on Data Semantics*, 2014.
- [8] R. Vaculin, R. Hull, T. Heath, C. Cochran, A. Nigam, and P. Sukaviriya, "Declarative business artifact centric modeling of decision and knowledge intensive business processes," in *15th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2011)*, pp. 151–160.
- [9] U. V. Riss, A. Rickayzen, H. Maus, and van Der Aalst, Wil MP, "Challenges for business process and task management," *Journal of Universal Knowledge Management*, vol. 2, pp. 77–100, 2005.
- [10] V. Bellotti, B. Dalal, N. Good, P. Flynn, D. G. Bobrow, and N. Ducheneaut, "What a to-do: studies of task management towards the design of a personal task list manager," in *CHI '04 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, E. Dykstra-Erickson and M. Tscheligi, Eds., 2004, pp. 735–742.
- [11] R. Pryss, N. Mundbrod, D. Langer, and M. Reichert, "Supporting medical ward rounds through mobile task and process management," *Information Systems and e-Business Management*, 2014.
- [12] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
- [13] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [14] S. Gregor, "The nature of theory in information systems," *MIS Quarterly*, pp. 611–642, 2006.
- [15] R. K. Yin, *Case study research: Design and methods*, 4th ed., ser. Applied social research methods series. Los Angeles and Calif: Sage Publications, 2009, vol. 5.
- [16] D. Müller, M. Reichert, and J. Herbst, "A new paradigm for the enactment and dynamic adaptation of data-driven process structures," in *Advanced Information Systems Engineering*, 2008, pp. 48–63.
- [17] J. Tiedeken, M. Reichert, and J. Herbst, "On the integration of electrical/electronic product data in the automotive domain," *Datenbank-Spektrum*, vol. 13, no. 3, pp. 189–199, 2013.
- [18] P. Dadam, M. Reichert, and K. Kuhn, "Clinical workflows - the killer application for process-oriented information systems?" in *BIS 2000*, W. Abramowicz and M. E. Orłowska, Eds. Springer-Verlag, 2000, pp. 36–59.
- [19] R. Lenz and M. Reichert, "It support for healthcare processes – premises, challenges, perspectives," *Data & Knowledge Engineering*, vol. 61, no. 1, pp. 39–58, 2007.
- [20] VDI, "Richtlinie 2206 entwicklungsmethodik fuer mechatronische systeme," *Düsseldorf, VDI-Verlag*, 2004.
- [21] D. Allen, *Getting things done: The art of stress-free productivity*. New York: Viking, 2001.
- [22] B. M. Hales and P. J. Pronovost, "The checklist—a tool for error management and performance improvement," *Journal of critical care*, vol. 21, no. 3, pp. 231–235, 2006.
- [23] C. Gutwin and S. Greenberg, "Support for group awareness in real-time desktop conferences," in *Proceedings of the Second New Zealand Computer Science Research Students' Conference*, 1995, pp. 18–21.
- [24] M. Reichert and P. Dadam, "Adeptflex - supporting dynamic changes of workflows without losing control," *Journal of Intelligent Information Systems*, vol. 10, no. 2, pp. 93–129, 1998.
- [25] S. Rinderle, M. Reichert, and P. Dadam, "Flexible support of team processes by adaptive workflow systems," *Distributed and Parallel Databases*, vol. 16, no. 1, pp. 91–116, 2004.
- [26] M. Reichert, J. Kolb, R. Bobrik, and T. Bauer, "Enabling personalized visualization of large business processes through parameterizable views," in *27th ACM Symposium On Applied Computing (SAC'12), 9th Enterprise Engineering Track (EE'12)*. ACM Press, 2012, pp. 1653–1660.
- [27] C. Janiesch, M. Matzner, and O. Müller, "Beyond process monitoring: a proof-of-concept of event-driven business activity management," *Business Process Management Journal*, vol. 18, no. 4, pp. 625–643, 2012.
- [28] M. Kersten, "Focusing knowledge work with task context," Ph.D. dissertation, The University of British Columbia, 2007.