Knowledge-intensive Processes: An Overview of Contemporary Approaches*

Claudio Di Ciccio, Andrea Marrella, and Alessandro Russo

Sapienza Università di Roma, Rome, Italy
{cdc,marrella,arusso}@dis.uniroma1.it

Abstract. Engineering of knowledge-intensive processes is far from being mastered. Processes are defined knowledge-intensive when people/agents carry them out in a fair degree of "uncertainty", where the uncertainty depends on different factors, such as the high number of tasks to be represented, their unpredictable nature, or their dependency on the scenario. In the worst case, there is no pre-defined view of the knowledge-intensive process, and tasks are mainly discovered as the process unfolds. In this work, starting from three different real scenarios, we present a critical comparative analysis of the existing approaches used for supporting knowledge-intensive processes, and we discuss some recent research techniques that may complement or extend the existing state of the art.

Keywords: Knowledge-intensive Processes, Process Management Systems, Health Care, Process Adaptation, Process Mining

1 Introduction

Process management systems (PMSs) hold the promise of facilitating the everyday operation of many enterprises and work environments. However, PMSs remain especially useful in a limited range of applications where business processes can be described with relative ease. Current modeling techniques are used to codify processes that are completely predictable: all possible paths along the process are well-understood, and the process participants never need to make a decision about what to do next, since the workflow is completely determined by their data entry or other attributes of the process. This kind of highly-structured work includes mainly production and administrative processes. However, most business functions involve collaborative features and unstructured processes that do not have the same level of predictability as the routine structured work [58].

In [29] processes have been classified on the basis of their "degree of structure". Traditional PMSs perform well with fully *structured processes* and controlled interactions between participants. A major assumption is that such processes, after having been modeled, can be repeatedly instantiated and executed

* This work has been partly supported by the SAPIENZA grant TESTMED and by the EU Commission through the project SmartVortex

in a predictable and controlled manner. However, even for structured processes, the combination and sequence of tasks may vary from instance to instance due to changes in the execution context such as user preferences, or modifications in the environment such as exceptions and changes in the business rules. In such cases (*structured processes with ad hoc exceptions*), processes should be adapted accordingly (e.g. by adding, removing or generating an alternative sequence of activities). In general, structured processes can be described by an explicit and accurate model. But in scenarios where processes are to a large extent unclear and/or unstructured, process modeling cannot be completed prior to execution (due to lack of domain knowledge a priori or to the complexity of task combinations). Hence the classical axiom "first model, then execute" – valid for the enactment of structured processes – fails. As processes are executed and knowledge is acquired via experience, it is needed to go back to the process definitions and correct them according to work practices. This is the case of *unstructured processes with predefined fragments*, where processes cannot be anticipated, and thus cannot be studied or modeled as a whole. Instead, what can be done is to identify and study a set of individual activities, and then try to understand the ways in which these activities can precede or follow each other. At the end of the classification lies the category of *unstructured processes*, where it is impossible to define a priori the exact steps to be taken in order to complete an assignment. Since there is no pre-defined view of the process, process steps are discovered as the process scenario unfolds, and might involve decisions not based on some "codified policy", but on the user expertise applied on the scenario at hand.

The class of *knowledge-intensive processes* is transversal with respect to the classification proposed in [29]. In the literature, different definitions have been proposed about what does "knowledge-intensive" mean for a business process. In [24] a process is defined as knowledge intensive if its value can only be created through the fulfillment of the knowledge requirements of the process participants, while Davenport recognizes the knowledge intensity by the diversity and uncertainty of process input and output [11]. In our view, a knowledge-intensive process is characterized by activities that can not be planned easily, may change on the fly and are driven by the contextual scenario that the process is embedded in. The scenario dictates who should be involved and who is the right person to execute a particular step, and the set of users involved may be not formally defined and be discovered as the process scenario unfolds. Collaborative interactions among the users typically is a major part of such processes, and new process steps might have to be defined at run time on the basis of contextual changes. Despite the popularity of commercial PMSs, there is still a lack of maturity in managing such processes, i.e., a lack of a semantic associated to the models or an easy way to reason about that semantic.

In this paper, starting from three different real application scenarios, we present a critical and comparative analysis of the existing approaches used for supporting knowledge-intensive processes, and we discuss some recent research techniques which may complement or extend the existing state of the art. The rest of the paper is organized as follows. Section 2 discusses the role of knowledge-intensive processes in the health-care domain, mainly focusing on how different modeling approaches can contribute to the process representation and execution. Section 3 discusses the use of knowledge-intensive processes for supporting the work in highly dynamic scenarios, by focusing on the challenging aspect of process adaptation. Section 4 traces the evolution of process mining, from the beginnings up to the current open challenge of discovering flexible models for knowledge-intensive partially structured processes, along with the graphical models proposed for presenting them to the user. Finally, Section 5 concludes the paper.

2 Modeling Approaches for Healthcare Processes

Healthcare is widely recognized as one of the most promising, yet challenging, domains for the adoption of process-oriented solutions able to support both organizational and clinical processes [10,31,46,30]. Organizational processes, which also include administrative tasks (patient admission/discharge, appointment scheduling, etc.), are typically structured, stable and repetitive, and represent the ideal setting for the application of traditional approaches for process automation and improvement. On the other side, the knowledge-intensive nature and flexibility requirements of medical treatment processes [3,37] pose challenges that existing process management approaches are not able to adequately handle. Although BPM solutions can potentially support these processes, in practice their uptake in healthcare is limited, mainly due to a generally perceived lack of flexibility [30]. Clinical decision making is highly knowledge-driven, as it depends on medical knowledge and evidence, on case- and patient-specific data, and on clinicians' expertise and experience. Patient case management is mainly the result of knowledge work, where clinicians act in response to relevant events and changes in the clinical context on a per-case basis, according to so-called diagnostic-therapeutic cycles based on the interleaving between observation, reasoning and action [31]. Clinical practices can not be captured by process models that require a complete specification of activities and their control/data flow, with the risk of constraining the clinicians and undermining the acceptance of proposed tools.

Despite these characteristics, in the last years the medical community has introduced Clinical Guidelines (CGs), in an attempt to improve care quality and reduce costs. CGs are "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances"[21] and act as blueprints that guide the care delivery process and provide evidence-based recommendations. Consequently, many research groups have focused on computer-interpretable clinical guidelines (CIGs) and different languages have been proposed [49,42,61], which can be broadly classified as rule-based (e.g., Arden Syntax), logic-based (e.g., PROforma), network-based (e.g., EON) and workflow-based (e.g., Guide). Most of them follow a task-based paradigm where modeling primitives for representing actions, decisions and patient states are linked via scheduling and temporal constraints, often in a rigid flowchart-like structure, and many representation models are supported by systems that allow the definition and enactment of CGs [27]. This rapid evolution in medical informatics has occurred mainly independently of the advances in the BPM community. However, the recent shift in the BPM domain towards process flexibility, adaptation (see Section 3) and evolution [47,30] has led to reconsider the link with CIGs and investigate the benefits coming from the application of process-oriented approaches in the healthcare domain [36]. On the one side, pattern-based analyses of CIG languages have shown that the expressiveness of these models, although specifically developed for the medical domain, is comparable with (or even lower than) the expressiveness of process modeling languages [39]. On the other side, emerging declarative constraint-based approaches [40,32] have been investigated as a possible solution to achieve a high degree of flexibility, taking advantage of loosely specified process models. In this direction, the combination of procedural and declarative models is under investigation, in order to support healthcare processes with different degrees of structuredness.

After more than a decade of research activities, researchers and practitioners agree on three main points: *(i)* clinical procedures, based on semi-structured and unstructured decision making, can not be completely specified in advance nor fully automated; *(ii)* deviations and variations during the care process (as well as uncertainty and changes in the clinical context) represent the rule rather than the exception; *(iii)* process- and activity-centric models can not adequately represent and support clinical case management. One of the main limitations of existing approaches is that they often underestimate the knowledge and data dimension. As patient treatment is knowledge-driven, the focus should be not on automating the decision making process, but rather on supporting the clinician during this process, according to a "system suggests, user controls" approach [62] that makes available the appropriate data and relevant knowledge when needed or required. Any system intended to support CGs should allow for representing and integrating at a semantic level evolving medical knowledge, patient-related data (including conditions, medical history, prescribed treatments and medications, etc.), and the existing (sometimes unpredictable) interactions between patient conditions, treatments and medications. This focus on data and knowledge is producing a shift from a process management approach to a more flexible case management approach, well understood by clinicians (although mostly in the form of paper-based processes) but only partially investigated in the BPM area [60]. Process support requires object-awareness in the form of a full integration of processes with patient data models consisting of object types and object relations [30,5]. Domain-relevant objects (such as medical orders, clinical and lab reports, etc.), their attributes and their possible states need to be explicitly represented, along with their inter-relations, so as to define a rich information model. This data model enables the identification and definition of the activities that rely on the object-related information and act on it, producing changes on attribute values, relations and object states. As a result, a tight integration between data objects and process activities can be achieved. As object-awareness requires a data-driven process modeling and execution approach, based on object behavior and object interactions, process/activity-centric methodologies are being replaced by data-centric models evolving over time [7]. In the context of a CG, patient's clinical situation (referred to as patient state, scenario, or context [49]) is central and represent the shared knowledge that drives the decision making and evolves as a result of performed actions, made decisions and collected data. Conditions defined over patient state, along with temporal constraints, are typically used as entry/exit points for a guideline [61] and as eligibility criteria for specific actions [49]. During the collaboration-based patient management activities, clinicians have to react to internal (e.g., a change in patient's state) and external (e.g., availability of lab test results) events, that can occur in any sequence. Moreover, it is often not possible to predetermine which activities have to be executed and in which order when an event occurs: according to the diagnostic-therapeutic cycles mentioned before, the clinician first assesses and evaluate the situation and then acts or plans the actions to be performed. This suggests an interleaving and overlapping of modeling and execution, where the process is "created at the time it is executed". Any modeling and execution approach for supporting this view has to consider that the clinician should be guided by what *can* be done and not restricted by what *has* to be done [35]. Although the path to be followed can be initially unclear and is gradually determined by clinician decisions, the care process evolves through a series of intermediate goals or milestones to be achieved (e.g., bring a parameter back to a normal level) that can again be expressed as conditions or constraints over patient state.

Given the above scenario, a promising and emerging approach for modeling CGs and supporting their execution and management is the artifact-centric paradigm, which considers data and knowledge as an integral part of business processes [51]. It is based on the concept of business artifacts as an abstraction for business-relevant entities and data that evolve according to a lifecycle and drive the activities in a business setting. Activities are defined in the context of interrelated artifacts and become enabled as the result of triggering events (internal or external) constrained by conditions defined and evaluated over the artifacts. Events and conditions over artifacts can also be used to set specific goals and evaluate the progress towards their achievement. The scheduling of actions is thus event- and data-driven, rather than induced by direct control flow dependencies. Under this perspective, it emerges a clear correspondence between artifact-centric concepts and clinical case management, in particular if considering the Guard-Stage-Milestone (GSM) meta-model [51] as a representative example of the artifact-based paradigm. GSM builds on the concepts of information model and lifecycle model, where the latter includes milestones to be achieved, hierarchically organized stages as clusters of possible activities to be performed to achieve milestones, and guards, timed events and conditions that control the stages and determine milestones' achievement. The patient and his/her state, a diagnostic test, a treatment course can all be considered as artifact types and represented by an information model that evolves according to a lifecycle and captures all relevant data and relations (e.g., as a relational model or domain ontology). CGs could be seen as progressing through a set of stages, where each performed action, made decision or event occurrence is driven by (eligibility criteria mentioned before) and has an impact on patient state, as reflected in the underlying information model. The data-driven nature of the model facilitates the integration between process control knowledge and the patient-related and medical knowledge; in addition, the distinction between data attributes and status attributes can directly support an integrated and explicit representation of both patient and execution states, not provided by all CIG models [61,49]. Although artifact-centric models can open the way for a new generation of flexible and adaptive case management systems in healthcare, further investigation is needed to understand the contribution that these models can bring in solving well-known problems for CIGs; among them: *(i)* how to reconcile the decision-action nature of CGs with a declarative modeling approach than can be used and understood by clinicians and is able to represent the evidence-based knowledge contained in the CGs; *(ii)* how to define an information model that is able to capture all clinically relevant data and takes into account existing standards, models, and ontologies used in Electronic Medical Records (EMRs) for patient and medical data; *(iii)* to what extent clinical events and medical knowledge can be represented and encoded by rules and conditions; *(iv)* how can an artifact-centric model address the problems of guideline acquisition, verification, testing, tracing and evolution, and how to turn or customize abstract models in executable models that take into account additional information, such as resource availability, roles and local services, in a collaborative multi-user environment.

3 Process Adaptation in Highly Dynamic Scenarios

A recent open research question in the BPM field concerns how to tackle scenarios characterized by being very dynamic and subject to higher frequency of unexpected contingencies than classical scenarios, e.g., scenarios for emergency management. There, a PMS can be used to coordinate the activities of first responders on the field (e.g., reach a location, evacuate people from collapsed buildings, extinguish a fire, etc.). The use of processes for supporting the work in highly dynamic contexts has become a reality, thanks also to the growing use of mobile devices in everyday life, which offer a simple way for picking up and executing tasks. These kinds of processes are also named *dynamic processes*. A dynamic process usually includes a wide range of knowledge-intensive tasks; as the process proceeds, the sequence of tasks depends so much upon the specifics of the context (for example, which resources are available and what particular options exist at that time), and often it is unpredictable the way in how it unfolds. This is due to the high number of tasks to be represented and to their unpredictable nature, or to a difficulty to model the whole knowledge of the domain of interest at design time. If we refer again to the classification shown in [29], dynamic processes can be classified between structured processes with ad hoc exceptions and unstructured processes with predefined fragments.

Research efforts in this field try to enhance the ability of dynamic processes and their support environments to modify their behavior in order to deal with contextual changes and exceptions that may occur in the operating environment during process enactment and execution. On the one hand, existing PMSs like YAWL [50] provide the support for the handling of expected exceptions. The process schemas are designed in order to cope with potential exceptions, i.e., for each kind of exception that is envisioned to occur, a specific contingency process (a.k.a. exception handler or compensation flow) is defined. On the other hand, adaptive PMSs like ADEPT2 [65] support the handling of unanticipated exceptions, by enabling different kinds of ad-hoc deviations from the pre-modeled process instance at run-time, according to the structural process change patterns defined in [64].

However, traditional approaches that try to anticipate how the work will happen by solving each problem at design time, as well as approaches that allow to manually change the process structure at run time, are often ineffective or not applicable in rapidly evolving contexts. The design-time specification of all possible compensation actions requires an extensive manual effort for the process designer, that has to anticipate all potential problems and ways to overcome them in advance, in an attempt to deal with the unpredictable nature of this kind of processes. Moreover, the designer often lacks the needed knowledge to model all the possible contingencies, or this knowledge can become obsolete as process instances are executed and evolve, by making useless his/her initial effort. In general, for a dynamic process there is not a clear, anticipated correlation between a change in the context and corresponding process changes, since the process may be different every time it runs and the recovery procedure strictly depends on the actual contextual information. For the same reason, it is also difficult to manually define an ad-hoc recovery procedure at run-time, as the correctness of the process execution is highly constrained by the values (or combination of values) of contextual data. Dealing with dynamic processes require that PMSs provide intelligent failure handling mechanisms that, starting from the original process model, are able to adapt process instances without explicitly defining at design time all the handlers/policies to recover from exceptions and without the intervention of domain experts.
Recently, some techniques from the field of artificial intelligence (AI) have been applied to process management, with the purpose of improving the degree of automatic adaptation of dynamic processes. In [23], the authors present a concept for dynamic and automated workflow re-planning that allows recovering from task failures. To handle the situation of a partially executed workflow, a multi-step procedure is proposed that includes the termination of failed activities, the sound suspension of the workflow, the generation of a new complete process definition and the adequate process resumption. In [28], the authors take a much broader view of the problem of adaptive workflow systems, and show that there is a strong mapping between the requirements of such systems and the capabilities offered by AI techniques. In particular, the work describes how planning can be interleaved with process execution and plan refinement, and investigates plan

patching and plan repair as means to enhance flexibility and responsiveness. A new life cycle for workflow management based on the continuous interplay between learning and planning is proposed in [20]. The approach is based on learning business activities as planning operators and feeding them to a planner that generates the process model. The main result is that it is possible to produce fully accurate process models even though the activities (i.e., the operators) may not be accurately described. The approach presented in [45] highlights the improvements that a legacy workflow application can gain by incorporating planning techniques into its day-to-day operation. The use of contingency planning to deal with uncertainty (instead of replanning) increases system flexibility, but it does suffer from a number of problems. Specifically, contingency planning is often highly time-consuming and does not guarantee a correct execution under all possible circumstances. Planning techniques are also used in [22] to define a self-healing approach for handling exceptions in service-based processes and repairing faulty activities with a model-based approach. During the process execution, when an exception occurs, a new repair plan is generated by taking into account constraints posed by the process structure and by applying or deleting actions taken from a given generic repair plan, defined manually at design time.

An interesting approach for dealing with exceptional changes has been proposed in [13,34]. Here, it is presented SmartPM (Smart Process Management), a model and a proof-of-concept PMS featuring a set of techniques providing support for automatic adaptation of processes. In SmartPM, a process model is defined as a set of n task definitions, where each task t_i can be considered as a single step that consumes input data and produces output data. Data are represented through some process variables whose definition depends strictly on the specific process domain of interest. The model allows to define logical constraints based on process variables through a set F of predicates f_j. Such predicates can be used to constrain the task assignment (in terms of *task preconditions*), to assess the outcome of a task (in terms of *task effects*) and as guards into the expressions at decision points (e.g., for cycles or conditional statements). Choosing the predicates that are used to describe each activity falls into the general problem of *knowledge representation*. To this end, the environment, services and tasks are grounded in domain theories described in Situation Calculus [48]. Situation Calculus is specifically designed for representing dynamically changing worlds in which all changes are the result of the tasks' execution. Processes are represented as IndiGolog programs. IndiGolog [12] allows for the definition of programs with cycles, concurrency, conditional branching and interrupts that rely on program steps that are actions of some domain theory expressed in Situation Calculus. The dynamic world of SmartPM is modeled as progressing through a series of *situations*. Each situation is the result of various tasks being performed so far. Predicates may be thought of as "properties" of the world whose values may vary across situations. SmartPM provides mechanisms for adapting process schemas that require no pre-defined handlers. Specifically, adaptation in SmartPM can be seen as reducing the gap between the *expected reality*, the (idealized) model of reality that is used by the PMS to reason, and the *physical reality*, the real world with the actual values of conditions and outcomes. The physical reality Φ_s reflects the concept of "now", i.e., what is happening in the real environment whilst the process is under execution. In general, a task t_i can only be performed in a given physical reality Φ_s if and only if that reality satisfies the *preconditions* Pre_i of that task. Moreover, each task has also a set of *effects* Eff_i that change the current physical reality Φ_s into a new physical reality Φ_{s+1}. At execution time, the process can be easily invalidated because of task failures or since the environment may change due to some external event. For this purpose, the concept of *expected reality* Ψ_s is given. A recovery procedure is needed if the two realities are different from each other. An execution monitor is responsible for detecting whether the gap between the expected and physical realities is such that the original process δ_0 cannot progress its execution. In that case, the PMS has to find a recovery process δ_h that repairs δ_0 and removes the gap between the two kinds of reality. Currently, the adaptation algorithm deployed in SmartPM synthesizes a linear process δ_h (i.e., a process consisting of a sequence of tasks) and inserts it at a given point of the original process - specifically, that point of the process where the deviation was first noted. This means that such technique is able to automatically recover from exceptions without defining explicitly any recovery policy.

4 Mining

Process Mining [54], also referred to as *Workflow Mining* [53], is the set of techniques that allow the extraction of process descriptions, stemming from a set of recorded executions. Throughout this Section, we will investigate the techniques adopted, along with the notations used to display the results, i.e., the mined processes. To date, ProM [55] is one of the most used plug-in based software environment for implementing workflow mining techniques. The idea to apply process mining in the context of workflow management systems was introduced in [1]. There, processes were modelled as directed graphs where vertices represented individual activities and edges stood for dependencies between them. Cook and Wolf, at the same time, investigated similar issues in the context of software engineering processes. In [8] they described three methods for process discovery: *(i)* neural network-based, *(ii)* purely algorithmic, *(iii)* adopting a Markovian approach. The authors considered the latter two as the most promising. Although, the results presented in [8] were limited to sequential behavior only. The nowadays mainstream process mining algorithms and management tools model processes with a graphical syntax derived from a subset of Petri Nets, i.e., Workflow Nets (WfN [53]), explicitly designed to represent the control-flow dimension of a workflow. See [41] for a history of Petri nets and an extensive bibliography. From [1] onwards many techniques have been proposed, in order to address specific issues: pure algorithmic (e.g., α algorithm [59] and its evolution α^{++} [67]), heuristic (e.g., [66]), genetic (e.g., [38]). Heuristic and genetic algorithms were introduced to cope with noise, that the pure algorithmic techniques were not able to manage. Whereas algorithmic processes rely on footprints of traces (i.e., tables reporting whether events appeared before or afterwards, if decidable) to determine the workflow net that could have generated them, heuristic approaches build a representation similar to causal nets, taking frequencies of events and sequences into account when constructing the process model, in order to ignore infrequent paths. Genetic process mining adopts an evolutionary approach to the discovery and differs from the other two in that its computation evolves in a non-deterministic way: the final output, indeed, is the result of a simulation of a process of natural selection and evolutionary reproduction of the procedures used to determine the final outcome. A very smart extension to the previous research was achieved by the two-steps algorithm proposed in [52]. Differently from previous works, in which the proposed approaches provide a single process mining step, it splitted the computation in two phases: the first built a Transition System that represents the process behavior and the tasks causal dependencies; the second made use of the state-based "theory of regions" [9,15] to construct a Petri Net bisimilar to the Transition System. The first phase was made "tunable", so that it could be either more strictly adhering or more permissive to the analyzed log traces behavior, i.e., the expert could determine a balance between "overfitting" and "underfitting". Indeed, past execution traces are not the whole universe of possible ones that may run: hence, the extracted process model should be valid for future unpredictable cases, on one hand, nevertheless checking whether the latter actually adhere to the common behavior, on the other hand. This issue reveals to be particularly relevant in the field of knowledge-intensive processes.

To date, the majority of research relating to processes coped with structured business processes. [26] discusses about a particular class of knowledge-intensive processes, named "artful business processes"; they are typically carried out by those people whose work is mental rather than physical (managers, professors, researchers, etc.), the so called "knowledge workers" ([63]). With their skills, experience and knowledge, they are used to perform difficult tasks which require complex, rapid decisions among multiple possible strategies, in order to fulfill specific goals. In contrast to business processes that are formal and standardized, informal processes are not even written down, often, let alone defined formally, and can vary from person to person even when those involved are pursuing the same objective. Knowledge workers create informal processes "on the fly" to cope with many of the situations which arise in their daily work. While informal processes are frequently repeated, because they are not written down, they are not exactly reproducible, even by their originators, nor can they be easily shared. [63] described the "ACTIVE" EU collaborative project, coordinated by British Telecom. Such project addressed the need for greater knowledge worker productivity by providing more effective and efficient tools. Among the main objectives, it aimed at helping users to share and reuse informal processes, even by learning those processes from the user's behavior. Basing on the work of [6] and [56], [19] investigated the challenge of mining these processes out of semi-structured texts, i.e., the email conversations exchanged among knowledge workers, through the interplay of text mining, object matching and process mining techniques. It provided an architectural overview of the application (named MailOfMine) able to fulfill the objective.

The need for flexibility in the definition of some types of process, such as artful business processes, leads to an alternative to the classical "imperative" approach: the "declarative". Rather than using a procedural language for expressing the allowed sequences of activities, it is based on the description of workflows through the usage of constraints: the idea is that every task can be performed, except what does not respect them. [58] showed how the declarative approach can help in obtaining a fair trade-off between flexibility in managing collaborative processes and support in controlling and assisting the enactment of workflows. DecSerFlow [57] and ConDec [43], now under the name of Declare [44], define such constraints as formulations in Linear Temporal Logic. [33] outlines an algorithm for mining Declare processes, integrated in ProM (namely, Declare Miner). The tool is based on the translation of Declare constraints into automata, and works in conjunction with the optimization techniques described in [68]. [4] describes the usage of inductive logic programming techniques to mine models expressed as a $\mathcal{S}$CIFF theory. $\mathcal{S}$CIFF theory is thus translated into the ConDec notation [43]. [2] differs from both [4] and [33] in that it does not directly verify the candidate constraints over the whole set of traces in input. It prepares an ad-hoc knowledge base of its own, instead, which specific queries are further submitted to. The model is determined on the base of the result of such queries. MINERful, proposed in [18], exploits this two-steps technique too, in order to improve the efficiency of the mining procedure. [17] proves the complexity of the algorithm to be polynomial w.r.t. the size of both the alphabet of constraints and the input traces. Differently from [33], [4] and [2], it is independent of the formalism adopted for representing constraints.

Declare provides a graphical model for representing declarative processes, useful to depict the constraints that hold between activities as a graph where nodes are activities and arcs are constraints among them. [25] and [16] presented a different approach to the graphical modelling. The former describes an event-based model, namely DCRGraph, showing the current state of the workflow at run-time, through the listing of tasks that can (either optionally or mandatorily) or can not be executed at the moment. A section describing the mapping of that notation to Büchi Automata is provided as well. The latter provides multiple graphical syntaxes, respectively depicting the process from two viewpoints: *(i) global*, i.e., focused on the representation of constraints between tasks, represented all together in a single graph and *(ii) local* i.e., focused instead on the constraints directly related to one single activity at a time. The first is then divided into a *base* and an *extended* version, in order to respectively depict less or more details about the nature of constraints that hold in the process – following the so called "map metaphor" [14]. The second is also twofold. The *static* view shows the constraints affecting an activity, which is put on the origin of a cartesian-like diagram. There, the implication and the temporal succession are aligned on orthogonal axes. The tasks involved in constraints related to the activity under analysis are put on different coordinates accordingly. In the *dynamic* view, the graph evolves as new tasks are executed. Starting from the initial, the enacted task is chained down to the previous. On the basis of the execution trace, the consequent next tasks are shown below the chain, in compliance with the constraints that hold at the moment.

5 Conclusions

In this work, we provided a critical and comparative analysis of the existing approaches used for supporting knowledge-intensive processes, and we showed some recent research techniques that may complement or extend the existing state of the art to this end.

In the health care domain, several challenges still need to be addressed and an interdisciplinary research effort is required. In this direction, the existing gap between the general evidence-based knowledge contained in CGs and the knowledge and information required to apply them to specific patients in local healthcare organizational contexts needs further investigation. Similarly, modeling approaches should allow to capture all "knowledge layers" and their possible interactions, including the procedural knowledge contained in CGs, the declarative knowledge representing domain- or site-specific constraints and properties, and clinicians' basic medical knowledge.

In highly dynamic environments, commercial PMSs are not able to deal with knowledge-intensive processes sufficiently, due to the static and only implicitly defined meta models of those systems. Basically, a dynamic process is largely dependent on the scenario at hand, and the result of process modeling is often a static plan of actions, which is difficult to adapt to changing procedures or to different business goals. In order to devise intelligent failure handling mechanisms for dynamic processes there is the need to define enriched workflow models, possibly with a declarative specification of process tasks, i.e., comprising the specification of input/output artefacts and task preconditions and effects. In general, the use of AI techniques for adapting dynamic processes seems very promising.

In the area of process mining, the declarative model proves to be very effective in allowing flexibility required by knowledge-intensive processes. Although, it has to be verified with people involved in those processes. E.g., the graphical notation proposed in [16] has to be implemented and its readability tested with real actors of those processes. A graphical notation representing the level of severity of a constraint in the process still misses. In the area of declarative workflow mining, it might be useful to determine the tightness of the discovered constraints on the basis of the frequency with which a constraint did not hold in the past. Moreover, a study on the impact of noise in such analysis could be done.

References

1. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: EDBT'98 (1998)

2. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: The sciff framework. ACM Trans. Comput. Log. 9(4) (2008)
3. Ammon, D., Hoffmann, D., Jakob, T., Finkeissen, E., Detschew, V., Wetter, T.: Management of Knowledge-Intensive Healthcare Processes on the Example of General Medical Documentation. In: BPM Workshops (2008)
4. Chesani, F., Lamma, E., Mello, P., Montali, M., Riguzzi, F., Storari, S.: Exploiting inductive logic programming techniques for declarative process mining. T. Petri Nets and Other Models of Concurrency 2, 278–295 (2009)
5. Chiao, C.M., Künzle, V., Reichert, M.: Towards Object-aware Process Support in Healthcare Information Systems. In: eTELEMED 2012 (2012)
6. Cohen, W.W., Carvalho, V.R., Mitchell, T.M.: Learning to classify email into "speech acts". In: EMNLP. pp. 309–316. ACL (2004)
7. Combi, C., Gambini, M., Migliorini, S., Posenato, R.: Modelling temporal, data-centric medical processes. In: ACM SIGHIT IHI 2012 (2012)
8. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Trans. Softw. Eng. Methodol. 7(3), 215–249 (1998)
9. Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving petri nets from finite transition systems. IEEE Trans. on Computers 47(8), 859 –882 (1998)
10. Dadam, P., Reichert, M., Kuhn, K.: Clinical Workflows - The Killer Application for Process-oriented Information Systems? In: BIS'00 (2000)
11. Davenport, T.H.: Improving knowledge work processes. In: Sloan Management Review, vol. 37 (1996)
12. De Giacomo, G., Lespérance, Y., Levesque, H., Sardina, S.: IndiGolog: A High-Level Programming Language for Embedded Reasoning Agents. In: Multi-Agent Prog.: Languages, Platforms and Applications (2009)
13. de Leoni, M., Marrella, A., Mecella, M., Sardina, S.: SmartPM – Featuring Automatic Adaptation to Unplanned Exceptions. Tech. rep., Sapienza Università di Roma (2011), http://ojs.uniroma1.it/index.php/DIS_TechnicalReports/article/view/9221/9141
14. de Leoni, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Visual support for work assignment in process-aware information systems. In: BPM'08 (2008)
15. Desel, J., Reisig, W.: The synthesis problem of petri nets. Acta Informatica 33, 297–315 (1996)
16. Di Ciccio, C., Catarci, T., Mecella, M.: Representing and visualizing mined artful processes in MailOfMine. In: HCI-KDD (2011)
17. Di Ciccio, C., Mecella, M.: MINERful, a mining algorithm for declarative process constraints in MailOfMine. Tech. rep., Sapienza Università di Roma (2012), http://ojs.uniroma1.it/index.php/DIS_TechnicalReports/issue/view/416
18. Di Ciccio, C., Mecella, M.: Mining constraints for artful processes. In: BIS'12 (2012)
19. Di Ciccio, C., Mecella, M., Scannapieco, M., Zardetto, D., Catarci, T.: MailOfMine - analyzing mail messages for mining artful collaborative processes. In: SIMPDA'11 (2011)
20. Ferreira, H., Ferreira, D.: An Integrated Life Cycle for Workflow Management Based on Learning and Planning. Int. J. Coop. Inf. Syst. 15 (2006)
21. Field, M.J., Lohr, K.N.: Clinical Practice Guidelines: Directions for a New Program. Institute of Medicine, Washington, DC (1990)
22. Friedrich, G., Fugini, M., Mussi, E., Pernici, B., Tagni, G.: Exception Handling for Repair in Service-Based Processes. IEEE Trans. on Soft. Eng. 36 (2010)
23. Gajewski, M., Meyer, H., Momotko, M., Schuschel, H., Weske, M.: Dynamic Failure Recovery of Generated Workflows. In: DEXA'05 (2005)

24. Gronau, N., Weber, E.: Management of knowledge intensive business processes. In: BPM'04 (2004)
25. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. In: PLACES'10 (2010)
26. Hill, C., Yates, R., Jones, C., Kogan, S.L.: Beyond predictable workflows: Enhancing productivity in artful business processes. IBM Syst. J. 45(4), 663–682 (2006)
27. Isern, D., Moreno, A.: Computer-based execution of clinical guidelines: a review. Int. J. of Medical Informatics 77(12) (2008)
28. Jarvis, P., Moore, J., Stader, J., Macintosh, A., du Mont, A.C., Chung, P.: Exploiting AI Technologies to Realise Adaptive Workflow Systems. AAAI Workshop on Agent-Based Systems in the Business Context (1999)
29. Kemsley, S.: The Changing Nature of Work: From Structured to Unstructured, from Controlled to Social. In: BPM'11 (2011)
30. Lenz, R., Peleg, M., Reichert, M.: Healthcare Process Support: Achievements, Challenges, Current Research. IJKBO (2012)
31. Lenz, R., Reichert, M.: IT support for healthcare processes - Premises, challenges, perspectives. Data & Know. Eng. 61(1) (2007)
32. Lyng, K.M., Hildebrandt, T.T., Mukkamala, R.R.: From Paper Based Clinical Practice Guidelines to Declarative Workflow Management. In: BPM (2008)
33. Maggi, F.M., Mooij, A.J., van der Aalst, W.M.P.: User-guided discovery of declarative process models. In: CIDM. pp. 192–199. IEEE (2011)
34. Marrella, A., Mecella, M.: Continuous Planning for Solving Business Process Adaptivity. In: BPMDS'11 (2011)
35. de Man, H.: Case Management: A Review of Modeling Approaches. BPTrends, www.bptrends.com (2009)
36. Mans, R.S., van der Aalst, W.M.P., Russell, N.C., Bakker, P.J.M., Moleman, A.J.: Process-Aware Information System Development for the Healthcare Domain - Consistency, Reliability, and Effectiveness. In: BPM Workshops (2009)
37. Marjanovic, O.: Improving Knowledge-Intensive Health Care Processes beyond Efficiency. In: ICIS'11 (2011)
38. Medeiros, A.K., Weijters, A.J., Aalst, W.M.: Genetic process mining: an experimental evaluation. Data Min. Knowl. Discov. 14(2), 245–304 (2007)
39. Mulyar, N., van der Aalst, W.M., Peleg, M.: A Pattern-based Analysis of Clinical Computer-interpretable Guideline Modeling Languages. JAMIA 14(6) (2007)
40. Mulyar, N., Pesic, M., Van Der Aalst, W.M.P., Peleg, M.: Declarative and procedural approaches for modelling clinical guidelines: addressing flexibility issues. In: BPM'07 (2007)
41. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE 77(4), 541 –580 (1989)
42. Peleg, M.e.a.: Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. JAMIA 10(1) (2003)
43. Pesic, M., van der Aalst, W.M.P.: A declarative approach for flexible business processes management. In: BPM Workshops (2006)
44. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: Declare: Full support for loosely-structured processes. In: EDOC. pp. 287–300 (2007)
45. R-Moreno, M.D., Borrajo, D., Cesta, A., Oddi, A.: Integrating planning and scheduling in workflow domains. Expert Syst. with App. 33(2) (2007)
46. Reichert, M.: What BPM technology can do for healthcare process support. In: AIME'11 (2011)
47. Reichert, M., Rinderle-Ma, S., Dadam, P.: Flexibility in Process-Aware Information Systems. In: Trans. on Petri Nets and Other Models of Concurrency II (2009)

48. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. The MIT Press (2001)
49. Sonnenberg, F.A., Hagerty, C.G.: Computer-Interpretable Clinical Practice Guidelines. Where are we and where are we going? Yearbook of Medical Inf. 45 (2006)
50. ter Hofstede, A., van der Aalst, W., Adams, M., Russell, N.: Modern Business Process Automation: YAWL and its Support Environment. Springer (2009)
51. Vaculin, R., Hull, R., Heath, T., Cochran, C., Nigam, A., Sukaviriya, P.: Declarative business artifact centric modeling of decision and knowledge intensive business processes. In: EDOC '11 (2011)
52. van der Aalst, W.M.P., Rubin, V., Verbeek, H., van Dongen, B., Kindler, E., Günther, C.: Process mining: a two-step approach to balance between underfitting and overfitting. Software and Systems Modeling 9, 87–111 (2010)
53. van der Aalst, W.M.P.: The application of petri nets to workflow management. Journal of Circuits, Systems, and Computers 8(1), 21–66 (1998)
54. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
55. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: Prom: The process mining toolkit. In: BPM'09 (Demos) (2009)
56. van der Aalst, W.M.P., Nikolov, A.: Mining e-mail messages: Uncovering interaction patterns and processes using e-mail logs. IJIIT 4(3), 27–45 (2008)
57. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a truly declarative service flow language. In: WS-FM. LNCS, vol. 4184, pp. 1–23. Springer (2006)
58. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Comp. Sc. - R&D 23(2), 99–113 (2009)
59. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. K. D. Eng. 16(9), 1128–1142 (2004)
60. van der Aalst, W.M.P., Weske, M.: Case handling: a new paradigm for business process support. Data & Know. Eng. 53(2) (2005)
61. Wang, D., Peleg, M., Tu, S., Boxwala, A., Greenes, R., Patel, V., Shortliffe, E.: Representation Primitives, Process Models and Patient Data in Computer-Interpretable Clinical Practice Guidelines: A Literature Review of Guideline Representation Models. Int. J. of Medical Informatics 68 (2002)
62. Wang, D., Peleg, M., Tu, S.W., Boxwala, A.A., Ogunyemi, O., Zeng, Q., Greenes, R.A., Patel, V.L., Shortliffe, E.H.: Design and implementation of the GLIF3 guideline execution engine. J. of Biomedical Informatics 37(5) (2004)
63. Warren, P., Kings, N., Thurlow, I., Davies, J., Buerger, T., Simperl, E., Ruiz, C., Gomez-Perez, J.M., Ermolayev, V., Ghani, R., Tilly, M., Bösser, T., Imtiaz, A.: Improving knowledge worker productivity - the Active integrated approach. BT Technology Journal 26(2), 165–176 (2009)
64. Weber, B., Reichert, M., Rinderle-Ma, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-aware Information Systems. Data Knowl. Eng. 66 (2008)
65. Weber, B., Wild, W., Lauer, M., Reichert, M.: Improving Exception Handling by Discovering Change Dependencies in Adaptive Process Management Systems. In: BPI'06 (2006)
66. Weijters, A., van der Aalst, W.: Rediscovering workflow models from event-based data using little thumb. Integrated Computer-Aided Engineering 10, 2003 (2001)
67. Wen, L., van der Aalst, W.M.P., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. Data Min. Knowl. Discov. 15(2), 145–180 (2007)
68. Westergaard, M.: Better algorithms for analyzing and enacting declarative workflow languages using LTL. In: BPM'11 (2011)