ProSyWis: Concept and Prototype for Managing Knowledge-Intensive Processes

Rüdiger Buck-Emden

Publisher: Dean Prof. Dr. Wolfgang Heiden

University of Applied Sciences Bonn-Rhein-Sieg,
Department of Computer Science

Sankt Augustin, Germany

November 2014

Technical Report 04-2014

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

ISSN 1869-5272

ProSyWis:
Concept and Prototype for Managing Knowledge-Intensive Processes

Rüdiger Buck-Emden

ruediger.buck-emden@h-brs.de

University of Applied Sciences Bonn-Rhein-Sieg
Department of Computer Science

Technical Report 04-2014
ISSN 1869-5272

Abstract — **Business process infrastructures like BPMS (Business Process Management Systems) and WfMS (Workflow Management Systems) traditionally focus on the automation of processes predefined at design time. This approach is well suited for routine tasks which are processed repeatedly and which are described by a predefined control flow. In contrast, knowledge-intensive work is more goal and data-driven and less control-flow oriented. Knowledge workers need the flexibility to decide dynamically at run-time and based on current context information on the best next process step to achieve a given goal. Obviously, in most practical scenarios, these decisions are complex and cannot be anticipated and modeled completely in a predefined process model. Therefore, adaptive and dynamic process management techniques are necessary to augment the control-flow oriented part of process management (which is still a need also for knowledge workers) with flexible, context-dependent, goal-oriented support.**

This paper addresses the demand for flexible business process support infrastructures and presents ProSyWis (Process Support System for Knowledge Workers), an approach for dynamic and adaptive management of knowledge-intensive processes, that combines classic control-flow support with declarative process modeling, rule-based activity identification, activity prioritization, complex event processing, case-based reasoning, and inter-personal collaboration. First ProSyWis prototype components have been implemented based on Software AG's webMethods suite[1].

Keywords: Process Automation, Knowledge-intensive Process, Knowledge Worker, Adaptive Case Management, Dynamic Case Management, Declarative Process Modeling, Collaborative Work, Rule-based System, Case-Based Reasoning, Complex Event Processing, BPMS, WfMS, CMMN

[1] Many thanks to Sascha Alda, Robert Czerwinski, Stefan Schulz and Christina Veit for their contributions to this research project.

I. Introduction

Since the 1990s, when organizations started to move from function-oriented business organization to business process orientation [1], a multitude of workflow management systems (WfMS) and business process management systems (BPMS) has been developed (for definition and scope see for instance [2]). Among them are WebSphere (IBM), jBPM (JBoss), webMethods (Software AG), YAWL (Open Source) and others. The promise of these systems is to effectively support pre-specified routine business processes by liberating workers from coordination tasks, improving maintainability of business control flows (by extracting the control flow definition explicitly into the BPMS/WfMS) and finally leading to higher quality and efficiency of business operation.

Although today's BPMS/WfMS approaches are sufficient for routine work (carried out repeatedly with predictable results in a standard way according to a pre-defined control flow), the question how to best support creative knowledge workers (like business decision makers, health professionals, company lawyers, merger & acquisition consultants, incident managers) has still to be answered.

Many if not most of the core business problems companies are facing today are of knowledge work nature[2], meaning being goal and data-driven, non-repeatable and, to some degree, unpredictable [3]. Knowledge work is typically embedded in a socio-technical context of human experts and information technology, thereby relying heavily on collaboration and continuous communication among people. In addition, this type of work is not driven by pre-specified control structures, but by dynamically evolving, work-related circumstances comprising e.g. goals, received events,

[2] Already in 2009 a growing proportion of 25% to 40% of the workforce was classified as knowledge workers ([86]).

executed and pending tasks, as well as all current and historic data/information/documents, which influence the flow of the work in question (figurative, all these elements have traditionally been collected in a “case folder”).

In short, contemporary BPMS/WfMS lack to support knowledge workers in highly dynamic scenarios where decisions about next process steps depend on the current situation and where changing circumstances require the flexibility to adapt behavior dynamically at runtime. On the other hand, flexible CSCW (Computer Supported Collaborative Work) tools [4] provided by the groupware community generally miss the process perspective. This is where concepts like Knowledge-Intensive Processes (KiPs [5] [6] [7]), Adaptive Case Management (ACM) [8], Dynamic Case Management (DCM) [9]), intelligent Business Process Management (iBPM [10]) and others (e.g. [11] [12] [13] [14] [15] [16]) come into play. In context of these concepts, a “cases” or a “knowledge-intensive process” can be roughly seen as a set of work assignments driven by evolving circumstances, requiring the execution of situation-dependent activities, and controlled by knowledge-workers in order to achieve a reasonable outcome or goal (for instance handling of lawsuits, insurance claims, research activities, or business management tasks). “Adaptive” refers to the ability to dynamically adjust behavior to changing circumstances. Although the details of how to implement these concepts are subject of various ongoing research and development initiatives, the goal seems to be clear: “to provide enough structure to make knowledge work manageable, but not to provide too much structure as to strangle it” [17].

This document contributes a holistic, design-oriented perspective on supporting knowledge work with ProSyWis (Process Support System for Knowledge Workers). The document is organized as follows: After a short overview on related work in industry and academics, we discuss briefly the key concepts ProSyWis is built on: events, rules, goals, milestones, decisions, and cases. We than describe the ProSyWis architecture in more detail, give an overview on first findings from the prototype implementation, and briefly sketch next steps.

II. Related Work

Since the seminal work of van der Aalst, Reijers et al. on case handling (see [18] [19]), numerous scientific papers have been published to address the various aspects of flexibility and adaptability in managing knowledge-intensive processes. Good overviews on latest scientific findings are available e.g. in [7] [20] [21]. In addition, as follow-up to Swenson’s popular book on mastering the unpredictable [8], a couple of industry-influenced books and papers have been published (e.g. [22] [23] [24] [10] [25]).

As a strong signpost for further academic thinking and commercial product development, OMG lately released the beta specification of Case Management Model and Notation (CMMN) [26] [27]. This specification allows the modeling and exchanging of weakly defined processes, thereby going far beyond ad-hoc processes known from BPMN (Business Process Model and Notation [28]). Nevertheless, BPMN can complement CMMN: to integrate strictly defined process fragments, CMMN allows to reference prescriptive (a.k.a. imperative) sub-processes modeled e.g. in BPMN.

On the commercial side, some product offerings are available to address knowledge worker demand for flexible, adaptive process management. Among them are the products from Appian, ECM, IBM, ISIS Papyrus, Kofax, Open Text, and Pegasystems [9]. These offerings either have their roots in the field of content and document management or in workflow and business process management. But the market penetration of dedicated products still seems to be limited. One reason might be that knowledge work in most cases also includes some routine work. Therefore we expect that ACM and the likes will be rather assimilated by established BPMS products than become a separate market segment of its own [29].

III. Dimensions of Process Modeling

A core characteristic of a BPMS is the availability of explicit process models, which represent the given process knowledge and drive the process execution. Process models have to take various perspectives into account. These are, among others, the functional perspective (which are the possible activities performed during process execution), the flow perspective (which activities are performed in which sequence under which conditions), the data perspective (which data elements are produced, consumed and exchanged during process execution), and the resource perspective (which people, systems, and services with which capabilities execute the tasks). Details are discussed e.g. in [30] [7]. If it comes to knowledge-intensive processes, there is no general agreement in the community how to model them. In the following, we put some emphasis on declarative process models to describe weakly defined processes.

IV. Declarative Process Flow Modeling

One essential finding of the community’s research and development activities so far is the fact that the activity flow performed by knowledge workers cannot be modeled completely by predefined control flows. Beside prescriptive workflow fragments, there is a need for weakly defined sub-process which give knowledge workers the flexibility to decide at run-time on the execution details. This is where e.g. declarative models[3] come into play [12] [13] [31]. In order to distinguish from pure ACM approaches, we use the term *knowledge-intensive business process* (see e.g. [7] [32]) for business processes consisting of predefined, imperative work sequences as well as non-predictable, weakly defined parts as needed. An overview on the whole spectrum of

[3] Declarative models use e.g. constraints, business rules or event conditions to define properties of and dependencies between activities in a business process. Declarative models specify what to do in order to achieve given business goals instead of prescribing how to reach a certain end state [31].

knowledge-intensive processes from structured to unstructured is presented in [7].

According to Goedertier et al. a business process can be modeled declaratively by describing its state space and the set of business rules that constrain the movements within this state space [31]. In this model, the business process state reflects a specific configuration of the facts about entities and activities of the process. In ProSyWis, a knowledge-intensive business process can be seen as a goal-directed, data and event-driven trajectory in the state space with business rules constraining the knowledge worker's decisions for next steps (see also [33]).

V. EVENTS

Along with rules and constraints, events constitute a principal concept to drive knowledge-intensive business process flow in ProSyWis. According to [34] an event is an occurrence which takes place at a specific time and initiates or triggers a predetermined response from the system., We use a declarative, rule-based approach to model event responses (see section VI). According to these rules, services based on the underlying Service Oriented Architecture (SOA) are triggered at runtime (see also [35]).

In ProSyWis, we distinguish between raw events with sender semantics (which arrive at a dedicated event adapter), and derived TEvents with internal semantics generated by the ProSyWis event processor factory ([36], see section X). The event processor factory can combine raw events to more complex events (complex event processing, CEP [37]) if needed by the business logic.

Various raw event categories can be handled by ProSyWis: External events (which occur outside the system boundary), internal events (which occur inside the system boundary), temporal events (internal or external events which occur at a pre-specified time), expected events (e.g. temporal event), unexpected events, and non-events (expected events that do not happen).

VI. RULES

One of the benefits of prevalent BPMS/WfMS is their ability to separate process flow specifications from application functions. On the other hand, they suffer from the fact that decisions (see section VII) and the rules they are based on are still hardcoded as part of the process and application logic making them complex and hard to maintain. To resolve this issue, rule engines (a.k.a. reasoning engines or reasoners) have been integrated in contemporary BPMS/WfMS infrastructures like WebSphere (ILOG), jBPM (Drools/Enterprise BRMS), and webMethods (FICO Blaze Advisor/Business Rules).

Goedertier and Vanthienen define business rules as atomic, formal expressions of business policies and regulations that define or constrain some aspect of a business [38]. According to Taveter and Wagner [39] [40], we classify business rules in four categories: integrity constraints (assertions that must be satisfied in all evolving states and state transition histories of an enterprise), derivation rules (statements of knowledge that are derived from other knowledge by inference or mathematical calculation, a.k.a. deduction rules), reaction rules (which state conditions under which certain actions are invoked, e.g. event-condition-action (ECA) rules or state-based condition-action (CA) rules (a.k.a. production rules), and deontic assignments (which assign rights and duties to (types of) internal agents).

Rules can have various origins: Integrity constraints are hard constraints which stem from the basic conditions of the business and must never be ignored (e.g. "goods cannot be shipped before the receiver's delivery address is known"). Business policies (and the respective rules) can be individually defined by business organizations (e.g. "accept orders from local customers only"). Finally, rules can be derived from knowledge and experience of acting persons or organizations (e.g. organizations can learn that customers with certain characteristics often don`t pay the bill and should not be accepted).

Rules can be exploited by rule engines. Most commercial rule engines support different rule categories like reaction rules (e.g. ECA rules or production rules), and inference rules. Notation for ECA rules: ON event IF condition THEN action. Notation for production rules: IF condition THEN action. From inference rules (IF premises THEN conclusion), two commonly used reasoning approaches can be distinguished: One is forward chaining, which can be described logically as repeated application of modus ponens ("A implies B; A is asserted to be true; therefore B must be true"). The other is backward chaining (or backward reasoning), which is an inference method that can be described as working backward from the goal(s).

In ProSyWis, we currently evaluate the details of how to use rules and rule engines to support the selection and prioritization of subsequent process steps. Other than with pure reaction rules, currently we do not trigger business actions directly, but always let the knowledge worker decide which of the possible actions should be executed next.

Further related questions we are working on: (i) What is the right time to find and define rules? In dynamic scenarios, it may not be sufficient to define rules at design time only. Rules may e.g. evolve by experience during run-time. And rules can be extracted from process history logs after a sufficient number of cases has been finalized (see [41]) using e.g. process mining techniques. (ii) How to decide which applicable ECA rules should fire in case of a new incoming TEvent? All applicable rules? In which sequence? Or only one? Which? This is where conflict resolution strategies come into play (see e.g. [42]). (iii) How to model rules. Currently we are using decision tables and decision trees. Related approaches to be further evaluated are e.g. OMG's OCL (Object Constraint Language [43]), JEXL (Java Expression Language [44]), SWRL (Semantic Web Rule Language [45]), OMG's SBVR (Semantic Business Vocabulary and Rules [46]), and OMG's DMN (Decision Model and Notation [47]). In Drools, domain specific languages (DSL) are introduced for rule modeling [48].

For further research on OMG's SBVR and DMN, it has to be recognized that rules in SBVR are complementary to

DMN. SBVR addresses the topic from an overall organizational level by providing a formalized vocabulary to document the semantics of business rules. DMN, on the other hand, defines how to model decision requirements and decision logic in concrete business processes. In addition, SBVR doesn't include any means to structure large sets of business rules. SBVR also doesn't address the transformation of business rules into Business Rule Engines. For this DMN has to be recognized.

VII. DECISIONS

Decisions are a basic constituent of knowledge-intensive processes which determine e.g. next process steps, data elements to be affected and resources to be used [4]. To separate (i) the modeling of decision tasks (for instance in BPMN), (ii) the modeling of decisions themselves including their interrelationships (e.g. as decision requirements diagrams (DRD)), and (iii) the definition of decision logic (e.g. as decision tables), OMG published the beta version of DMN (Decision Model and Notation) in 2014 [47]. According to OMG, DMN has the purpose to provide the constructs that are needed to model decisions, so that organizational decision-making can be readily depicted in diagrams, accurately defined by business analysts, and (optionally) automated[5]. Further DMN details are discussed in [10] [49].

According to DMN and also discussed e.g. in [50] [51] [52] [53], a process model should not include direct mappings of decision trees. In ProSyWis, we therefore separate decisions logic regarding next process steps (and the decision rules they are based on) from the remaining process model by using decision tables, which are evaluated by a rule engine. This allows a much higher level of operational flexibility, traceability, and maintainability [50] [53].

VIII. GOALS AND MILESTONES

Goals are the ultimate regulatory entities when it comes to decision making in business processes (e.g. decision on next process step). (Note: process goals can address expected process outcomes as well as non-functional process aspects (see e.g. [54])). List et al. [55] define goals as one of the constituting elements of the business process meta-model. Jander et al. and Burmeister et al. postulate that goals should directly impact the activity flow within a process[6] in order to eliminate the divergence between goals as part of the process documentation and the process execution semantics [15] [56]. This requirement cannot be fulfilled by traditional process modeling methods like BPMN or EPC (Event-driven Process Chains, [57]), which do not support explicit modeling of process goals. Instead, they only represent goals implicitly by the process flow definition. This limits flexibility in case of changing or multiple, possibly conflicting process goals [58] [15]. In addition, goals may only be achieved to a certain degree. This establishes the need for performance metrics and KPIs (Key Performance Indicators), which measure the degree of goal achievement and are not adequately reflected by activity-centered languages like BPMN (to be further evaluated as part of ProSyWis).

A possible approach to explicitly represent process goals and their lifecycle is described in [15] (see also [56] for a related approach). The idea is that abstract or complex process goals can be decomposed into a hierarchy of sub-goals, and that the overall goal is achieved by meeting all related sub-goals. On the lowest level of the hierarchy, each sub goal can be identified with one or more elementary tasks, which together constitute the plan to achieve the particular (sub-) goal. Depending on context and given business rules, tasks belonging to an "active" (sub-) goal can become candidates for next-step execution. Looking on CMMN, goals and sub-goals can be related to milestones in the case plan model.

Some goal-related concepts which will be further evaluated in ProSyWis stem from the work on intelligent agent systems (see e.g. [15] [56] [59]) and on practical reasoning (dealing with goal-directed reasoning to select concrete actions). Of special interest are BDI (Belief, Desire, Intention) agents, whose practical usage is worked out in [60]. Practical reasoning includes the main activities "deliberation" (goal recognition) and "means-end reasoning" (plan to achieve goal) and is simply explained by Wooldridge as "reasoning to figure out what to do" [61] instead of what to believe. According to Bratman [62], "practical reasoning is a matter of weighting conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes". In case of multiple eligible (sub-) goals linked to a process, the goal selection/activation decision establishes an additional dimension of flexibility to business processes (see for instance [15]). A related visionary product offered by Whitestein[7] is the Living Systems Process Suite (LSPS, former called LS/ABPM [63] [64] [65]), a goal-oriented, agent-based iBPMS.

[4] OMG defines decisions as "the act of determining an output value from a number of input values, using decision logic defining how the output is determined from the inputs" [47]. Decision logic might be weakly defined and decisions can be based on incomplete knowledge. Decisions can relate for instance to next process step, artefacts to be maintained or (human) actors to be contacted.

[5] One possible automation scenario given by OMG is the use of "decision services" deployed from a Business Rules Management System (BRMS) and called by a Business Process Management System (BPMS).

[6] One approach to be further evaluated in ProSyWis is to extent the process definition sketched in chapter IV consisting of state space (activities, entities, business) and business rules by a process goal entity type

[7] http://www.whitestein.com

IX. Case and Case Context

In CMMN, a case is defined as a "proceeding that involves actions taken regarding a subject in a particular situation to achieve a desired outcome" [26]. CMMN does not formalize the terms "activity", "process" and "sub-process". A task is defined as atomic unit of work that can be a human task, a process task or a case task. In contrast, BPMN defines an activity as work that a company or organization performs using business processes. An activity can be atomic or compounded. Activity types in BPMN are processes, sub-processes, and tasks. A sub-process is a process that is included within another process. A task is an atomic activity included within a process, which cannot be broken down to a finer level of detail. Tasks can be performed by end-user, applications, or both.

In ProSyWis we see the handling of a case as a knowledge-intensive process. Cases are structured as depicted in Figure 1. Each case instance is associated with a case file which includes the case history with all relevant log entries, the case state, a declarative case plan defining basic conditions for case processing behavior (this includes possible next-steps), as well as all business artifacts and entities (or references) relevant for the behavior of the case instance.

We use the term *context* to summarize everything that influences the behavior of a case instance. Context can be divided in internal and external context. Internal context refers to context elements which can be identified only by looking inside the case instance. This is also called the *state* of a case instance. External context refers to all context elements outside the case instance [66].

The case file is the principal information source for next step recommendations in ProSyWis. In addition, each case may be associated with patterns for process step sequencing, artifact selection etc. (based e.g. on best practices).

The ProSyWis case modeling concept is strongly influenced by OMG's CMMN[8], which in turn inherited its data-centricity and artifact-orientation from GSM (Guard-Stage-Milestone [27] [67] [68]). CMMN building blocks are tasks, stages, sentries, milestones, event listeners, connectors, case plan models, case file items, and others. Process drivers are business artifacts (meaning a business entities with lifecycle), which are a tight combination of data and process, incorporating both an information model and a lifecycle model. The core behavioral model of CMMN is derived from the GSM business artifact lifecycle comprising tasks, hierarchical stages, milestones and events. An important feature of CMMN, influenced by Cordys[9] [69], is the ability for case workers to dynamically alter the runtime plan (planning at runtime).

[8] Another promising case modeling approach not evaluated yet in detail could be based on abstract state machines (ASM) as discussed by Börger et al. [85]

[9] Cordys was acquired by OpenText in 2013

X. ProSyWis Architecture Overview

ProSyWis is designed as a management system for knowledge-intensive business processes, which can combine static, control-flow directed sub-processes with dynamic, non-predictable work assignments. Although ProSyWis is planned as a solution to be deployed on premise as well as in the cloud, the current phase of the project focusses on the on premise version.

ProSyWis follows a data and event-driven architecture approach. The principal building blocks of the architecture are depicted in Figure 2.

Event management with <u>event adapters</u> and <u>event processor factory</u> allows ProSyWis to take action on various types and classes of (streams of) events. For instance, an incoming e-mail can be analyzed semantically and combined with other events in the sense of complex event processing (CEP) [37]. The final result is a transformed TEvent object (see Figure 3 for a first draft [70]), which is forwarded to the knowledge process engine to drive the next processing steps. The ProSyWis event adapter architecture is the foundation for rich external and internal interaction including interpersonal collaboration based e.g. on email and social networks.

The <u>knowledge process engine</u> is the central ProSyWis component to execute knowledge-intensive business processes. This event-driven component maintains state for each knowledge process instance, manages process flow and activates next process steps. The knowledge process engine cooperates tightly with the rule engine (to exploit declarative process models) and the prioritizer (to prioritize possible next process steps). Based on services provided by these components, the knowledge process engine triggers UI management which displays pending (recommended, prioritized) next process steps as a selection list (see Figure 4 for a first draft [71]). From this screen, the knowledge worker makes the final process flow decisions.

The state of a knowledge process instance can be derived from the history of performed activities, received events, and further instance-dependent context data/information (to be further evaluated).

The knowledge process engine allows users to dynamically modify imperative and/or declarative process specification elements according to changing circumstances (to be further evaluated).

<u>Rule engine</u> is used by the ProSyWis knowledge process engine to identify possible next process steps. For this, business rules/policies, goals, process state, and possibly further context information are evaluated. In addition, the rule engine may also be instrumental to prioritize the list of possible next process steps (see prioritizer, details to be evaluated) and to execute derivation rules (e.g. individual price calculation).

Rules have to be defined in some formal structure/syntax (e.g. as decision table), which is transformed and afterwards executed by a rule engine using a specific rule execution algorithm (e.g. RETE [72]).

Context handler is the ProSyWis component which manages all internal and external context information (or references thereon) relevant for the behavior of a given knowledge process instance. This information source is accessible for all ProSyWis components.

Prioritizer is the central component in ProSyWis to prioritize the list of possible next process steps (generated by the rule engine) in order to make educated recommendations to the user. Algorithms for this prioritization can be based e.g. on context information, process instance state and/or process instance goals using techniques like rules[10] and decision tables/trees, utility theory, Bayesian nets and influence diagrams [73] [74] [75] [76] [77], process history analysis and process mining [12], case based reasoning [78], intelligent agents with goal and utility-based approaches (see [15] [56] [79]) and others (to be further evaluated, see for instance [14] [80] [81]). Among others, the benefit of influence diagrams is evaluated in ProSyWis by using the HUGIN Expert tool [82].

The BPM system in ProSyWis provides a framework for defining, executing and monitoring control-flow driven sub-processes based on BPMN. Currently Software AG's webMethods BPM engine is used to support the predefined routine parts of knowledge processes. In a later project phase, it is also planned to evaluate how to adapt control-flow driven processes to changing conditions at runtime (see for instance [11]).

UI management deals with all visualization aspects in ProSyWis. One key element is the knowledge process cockpit, which provides a complete overview of all active process instances as well as the state of these instances and a (prioritized) list of all possible next steps. An early UI prototype is depicted in Figure 4.

Data management is responsible for providing a persistency layer for all artifacts/data objects generated or used by ProSyWis components. Currently a Microsoft SQL database is used as persistency layer.

XI. PROTOTYPE IMPLEMENTATION

First ProSyWis prototype components are being implemented using Software AG's webMethods 9. An early version of the knowledge process engine (oriented towards CMMN and based on webMethods BPM engine and rule engine) and the UI management (based on My webMethods server) are available (see Figure 4 and Figure 5 [71]). In addition, CEP (complex event management) prototypes have been implemented to triggers ProSyWis processes using the open-source product Esper [70]. As well as Software AG's Apama.

The knowledge process engine maintains the case state for each case instance. In the prevailing ProSyWis version, case state reflects the current fulfillment/not-fulfillment (true, false) of required entry or activation conditions for each case plan entity (stage, task, milestone,…). Based on this case state, ProSyWis uses two decision tables (DTs) for the declarative definition of process behavior (see Figure 5 and Figure 6). One decision table corresponds to the behavior on the level of the overall case plan model (with stages, milestones and entry criteria), the other decision table represents the behavior on stage level. At runtime, these tables are evaluated by the knowledge process engine to identify possible next process steps (in a later ProSyWis version, additional facts like received events and other context information[11] may be recognized as well). After prioritizing the list of possible steps (task of the ProSyWis prioritizer), this list is translated into a UI selection screen (function of the translator component). After the knowledge worker has selected a certain task, this task is activated in an active-task window for further user interaction (done by the task producer). As soon as the selected task is completed, relevant state and log data are stored in the ProSyWis case file (see Figure 7).

XII. CONCLUSION AND FUTURE WORK

This paper presented the motivation for automated knowledge-intensive processes support together with the architecture concept of ProSyWis, a system to handle predefined as well as non-predictable activity streams. The implementation of the ProSyWis prototype is under way. Various open questions still have to be evaluated. Among others, these are the details of knowledge process handling, event management, and the context-dependent generation of next-step proposals during process execution.

[10] Ripple-down rules, which allow to define general rules first and their refinement later are a potential research topic for an incrementally growing rule base in ProSyWis (see e.g. [83], [84])

[11] For example rules like „if student's age is below 18, then parent`s signature is required".

REFERENCES

[1] M. Hammer and J. Champy, Reengineering the Corporation: A Manifesto for Business Revolution, New York: Harper Business Books, 1993.

[2] R. K. L. Ko, "A computer scientist's introductory guide to business process management (BPM)," *ACM Crossroads,* vol. 15, no. 4, 2009.

[3] T. Davenport, Thinking for a Living: How to Get Better Performance and Results from Knowledge Workers, Boston: Harvard Business School Press, 2005.

[4] J. Grudin and S. Poltrock, "CSCW - Computer Supported Cooperative Work," in *Encyclopedia of Human-Computer Interaction*, Aarhus, Denmark, The Interaction-Design.org Foundation, Available online at http://www.interaction-design.org/encyclopedia/cscw_computer_supported_cooperative_work.html, 2012.

[5] 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," *5th IEEE International conference on enterprise distributed object computing (EDOC 2011),* 2011.

[6] C. Di Ciccio, A. Marrella and A. Russo, "Knowledge-intensive Processes: An Overview of Contemporary Approaches," *Proceedings of the 1st International Workshop on Knowledge-intensive Business Processes (KiBP 2012),* pp. 33-47, 2012.

[7] C. Di Ciccio, A. Marrella and A. Russo, "Knowledge-intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches," *Journal on Data Semantics,* pp. 1-29, 2014.

[8] K. Swenson, Mastering the Unpredictable. How Adaptive Case Management Will Revolutionize The Way That Knowledge Workers Get Things Done, Tampa: Meghan-Kiffer Press, 2010.

[9] C. Le Clair and D. Miers, The Forrester Wave™: Dynamic Case Management, Forrester, 2011.

[10] L. Fischer (ed.), iBPMS, Future Strategies, 2013.

[11] P. Dadam and M. Reichert, "TheADEPTproject: a decade of research and development," *CSRD,* vol. 23, pp. 81-97, 2009.

[12] I. Rychkova and S. Nurcan, "Towards Adaptability and Control for Knowledge-Intensive Business Processes: Declarative Configurable Process Specifications," *Proceedings of the 44th Hawaii International Conference on Systems Science,* 2011.

[13] W. van der Aalst, M. Pesic and H. Schonenberg, "Declarative workflows:Balancing between flexibility and support," *CSRD (23),* pp. 99-113, 2009.

[14] C. Dorn, T. Burkhart, D. Werth and S. Dustdar, "Self-adjusting Recommendations for People-driven Ad-hoc Processes," *R. Hull, J. Mendling, and S. Tai (Eds.): BPM 2010, LNCS 6336,* p. 327–342, 2010.

[15] K. Jander, L. Braubach, A. Pokahr and W. Lamersdorf, "Goal-Oriented Processes with GBMN," *International Journal on Artificial Intelligence Tools,* vol. 20, no. 6, pp. 1021-1041, 2011.

[16] M. Brenner-Wickner, M. Book, T. Brückmann and V. Gruhn, "Execution Support for Agenda-Driven Case-Management," in *SAC '14 Proceedings of the 29th Annual ACM Symposium on Applied Computing*, Gyeongju, Korea, 2014.

[17] J. Ukelson, Jacob Ukelson's Blog, http://ukelson.wordpress.com/2010/07/, 2010.

[18] H. A. Reijers, J. H. M. Rigter and W. van der Aalst, "The Case Handling Case," *International Journal of Cooperative Information Systems, Vol. 12, No. 3,* pp. 365-391, 2003.

[19] W. van der Aalst, W. Weske and D. Grünbauer, "Case Handling: A New Paradigm forBusiness Process Support," *Data Knowledge Engineering 53(2),* 2005.

[20] M. S. P. La Rosa, BPM 2012 Workshops, Tallinn, Estonia: LNBIP 132, Springer, 2012.

[21] Y. T. Demey and H. Panetto, OTM 2013 Workshops, Graz, Springer, LNCS 8186, 2013.

[22] K. D. Swenson, Taming the Unpredictable: Real World Adaptive Case Management: Case Studies and Practical Guidance, Future Strategies, 2011.

[23] L. Fischer, How Knowledge Workers Get Things Done, Future Strategies, 2012.

[24] L. Fischer, Social BPM, Future Systems, 2011.

[25] A. Martens, A. Slominskii, G. Lakshmanan and N. Mukhi, "Advanced Case management Enabled by Business Provenance," in *10th IEEE Internationals Conference on Web Services (ICWS)*, Honolulu, HI, USA, 2012.

[26] OMG, Case Management Model and Notation (CMMN), Beta 1, http://www.omg.org/spec/CMMN/1.0, 2013.

[27] M. Marin, R. Hull and R. Vaculin, "Data Centric BPM and the Emerging Case Management Standard - A Short Survey," in *BPM 2012 Workshops*, LNBIP 132, Springer, 2013, pp. 24-30.

[28] OMG, Business Process Model and Notation (BPMN), Version 2.0, http://www.omg.org/spec/BPMN/2.0/, 2011.

[29] C. Richardson and D. Miers, The Forrester Wave™: BPM Suites, Q1 2013, Forrester, March 11, 2013.

[30] M. La Rosa, M. Dumas, A. H. ter Hofstede and J. Mendling, "Configurable multi-perspective business process models," *Information Systems,* vol. 36, no. 2, pp. 313-340, 2011.

[31] S. Goedertier, J. Vanthienen and F. Caron, "Declarative business process modelling: principles and modelling languages," *Enterprise Information Systems,* 2013.

[32] S. Kemsley, "The Changing Nature of Work: From Structured to Unstructured, from Controlled to Social," in *9th International Conference on Business Process Management (BPM 2011)*, Clermont-Ferrand, France, 2011.

[33] I. Bider, "Towards a Non-workflow Theory of Business Processes," in *10th International Conference on Business Process Management (BPM 2012)*, Thallinn, Estonia, 2012.

[34] R. Stumpf and L. Teague, Object-Oriented Systems Analysis and Design With UML, Prentice Hall, 2004.

[35] A. Buchmann, S. Appel, T. Freudenreich, S. Frischbier and P. Guerrero, "From Calls to Events: Architecting Future BPM Systems," in *10th International Conference on Business Process Management (BPM 2012)*, Thallinn, Estonia, 2012.

[36] O. Etzion and P. Niblett, Event Processing in Action, Stamford, Connecticut, USA: Manning, 2011.

[37] D. Luckham, The Power of Events, Boston: Addison-Wesley, 2002.

[38] S. Goedertier and J. Vanthienen, "Business Rules for Compliant Business Process Models," *BIS,* vol. 85, pp. 558-572, 2006.

[39] K. Taveter and G. Wagner, "Agent-Oriented Business Rules: Deontic Assignments," in *Proc. of Int. Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations, (OES-SEO2001)*, Rome, 2001.

[40] G. Wagner, "How to Design a General Rule Markup Language?," in *Proc. Workshop XML Technologien für das Semantic Web - XSW 2002*, Berlin, 2002.

[41] S. Schöning, M. Zeising and S. Jablonski, "Supporting Collaborative Work by Learning Process Models and Patterns from Cases," *9th IEEE Int. Conf. on Collaborative Computing (CollaborateCom),* 2013.

[42] J. Boyer and H. Mili, Agile Business Rule Development: Process, Architecture, and JRules Examples, Heidelberg: Springer, 2011.

[43] OMG, Object Constraint Language (OCL), Version 2.4, http://www.omg.org/spec/OCL/2.4, 2014.

[44] ApacheSoftwareFoundation, Java Expression Language (JEXL) Version 2.1.1, http://commons.apache.org/proper/commons-jexl/, 2011.

[45] W3C, "A Semantic Web Rule Language (SWRL) Combining OWL and RuleML," W3C, A Semantic Web Rule Language (SWRL) CombW3C Member Submission, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/, 2005.

[46] OMG, Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.0, http://www.omg.org/spec/SBVR/1.0/, 2009.

[47] OMG, Decision Model and Notation (DMN), Version 1.0 - Beta 2, http://www.omg.org/spec/DMN/1.0/Beta2/, 2014.

[48] J. D. Team, Drools Documentation, Version 6.0.1, http://docs.jboss.org/drools/release/6.0.1.Final/drools-docs/html_single/, last access: May 2014.

[49] J. Taylor, A. Fish, J. Vanthienen and P. Vincent, "Emerging Standards in Decision Modeling - An Introduction to Decision Model & Notation," in *Fischer, L.; iBPMS*, Future Strategies, 2013.

[50] J. Vanthienen, F. Caron and J. De Smedt, "Business Rules, Decisions and Processes: Five Reflections upon Living Apart Together," *SIGBPS Workshop on Business Processes and Service,* pp. 76-80, 2013.

[51] W. De Roover and J. Vanthienen, "On the Relation between Decision Structures, Tables and Processes," *OTM 2011 Workshops, LNCS 7046,* pp. 591-598, 2011.

[52] F. Wu, .. Priscilla, M. Gao, F. Caron, W. De Roover and J. Vanthienen, "Modeling Decision Structures and Dependencies," *OTM 2012 Workshops, LNCS 7567,* pp. 525-533, 2012.

[53] J. Pitschke, "Integrating Business Process Models and Business Logic: BPMN and The Decision Model," *BPMN 2011, LNBIP 95,* p. 148–153, 2011.

[54] P. Kueng and P. Kawalek, "Goal-based business process models: creation and evaluation," *Business Process Management Journal,* vol. 3 , no. 1, pp. 17-38, 1997.

[55] B. List and B. Korherr, "An Evaluation of Conceptual Business Process Modelling Languages," in *SAC'06*, Dijon, France, 2006.

[56] B. Burmeister, M. Arnold, F. Copaciu and G. Rimassa, "BDI-Agents for Agile Goal-Oriented Business Processes," *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Industry and Applications Trac,* pp. 37-44, 2008.

[57] G. Keller, M. Nüttgens and A.-W. Scheer, "Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)"," in

Veröffentlichungen des Instituts für Wirtschaftsinformatik, Saarbrüchen, 1992.

[58] S. Jacobs and R. Holten, "Goal Driven Business Modelling – Supporting Decision Making within Information Systems Development," *Proceedings of Conference on Organizational Computing Systems (COCS),* pp. 96-105 , 1995.

[59] M. Riemsdijk, M. Dastani and M. Winikoff, "Goals in Agent Systems: A Unifying Framework," in *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Estoril, Portugal, 2008.

[60] A. Rao and M. Georgeff, ". BDI Agents: From Theory to Practice," *Proceedings of the 1st Int. Conf. on Multi-Agent Systems (San Francisco, USA),* 1995.

[61] M. Wooldridge, Reasoning about Rational Agents, MIT Press, 2000.

[62] M. Bratman, " What is intention?," in *Intentions in Communication*, P. Cohen, J. Morgan and M. Pollack, Eds., Cambridge, MA, The MIT Press, 1990, pp. 15-32.

[63] G. Rimassa, M. Kernland and R. Ghizzioli, "An Agent-powered Suite for Goal-oriented Autonomic BPM," in *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Estoril, Portugal, 2008.

[64] D. Greenwood, "Goal-Oriented Autonomic Business Process Modeling and Execution: Engineering Change Management Demonstration," in *BPM 2008, LNCS 5240*, 2008.

[65] D. Greenwood and G. Rimassa, "Autonomic Goal-Oriented Business Process Management," in *Third International Conference on Autonomic and Autonomous Systems (ICAS'07)*, Athens, 2007.

[66] S. Wendt, "Der Zustandsbegriff in der Systemtheorie," Univrsität Kaiserslautern, https://kluedo.ub.uni-kl.de/files/35/no_series_7.pdf, 1998.

[67] R. Eshuis, R. Hull, Y. Sun and R. Vaculin, "Splitting GSM Schemas: A Framework for Outsourcing of Declarative Artifact Systems," *F. Daniel, J. Wang, and B. Weber (Eds.): BPM 2013, LNCS 8094,* p. 259–274, 2013.

[68] F. Heath, D. Boaz, M. Gupta, R. Vaculín, Y. Sun, R. Hull and L. Limonad, "Barcelona: A Design and Runtime Environment for Declarative Artifact-Crntric BPM," in *Proc. 11th International Conference on Service-Oriented Computing, ICSOC 2013*, Berlin, 2013.

[69] H. de Man, "Case Management: Cordys Approach," *BPTrends,* February 2009.

[70] C. Veit, Master Project: Architektur-Entwurf eines allgemeinen Event-Adapters für ein CEP-System am Beispiel der prototypischen Entwicklung eines Email-Processing-Adapters, Sankt Augustin: University of Applied Sciences Bonn-Rhein-Sieg, 2014.

[71] R. Czerwinski, Masterprojekt ProSyWis, Sankt Augustin: University of Applied Sciences Bonn-Rhein-Sieg (announced), 2014.

[72] C. Forgy, "Rete; A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence,* no. 19, pp. 17-37, 1982.

[73] U. Kjærulff and A. Madsen, Bayesian Networks and Influence Diagrams: A Guide, New York: Springer, 2013.

[74] R. Howard and J. Matheson, "Influence Diagrams," *Decision Analysis,* vol. 2, no. 3, p. 127–143.

[75] R. Shachter and D. Bhattacharjya, "Solving Influence Diagrams: Exact Algorithms," *Wiley Encyclopedia of Operations Research and Management Science,* 2011.

[76] M. Terek, "Decision Trees and Influence Diagrams in Decision Analysis," *Journal of the Applied Mathematics, Statistics and Informatics (JAMSI,* no. 1, 2005.

[77] F. V. Jensen and T. D. Nielsen, Bayesian Networks and Decision Graphs, Berlin: Springer, 2007.

[78] A. Aamodt and E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Communications,* vol. 7, no. 1, pp. 39-59, March 1994.

[79] M. Salehie and L. Tahvildari, "Towards a goal-driven approach to action selection in self-adaptive software," *Softw. Pract. Exper.,* vol. 42, p. 211–233, 2012.

[80] J. Busemeyer and T. Pleskac, "Theoretical Tools for Understanding and Aiding Dynamic Decision Making," *Journal of Mathematical Psychology,* vol. 53, pp. 126-138, 2009.

[81] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach (3rd Edition), Prentice Hall, 2010.

[82] F. Jensen, U. Kjærulff, M. Lang and A. Madsen, "HUGIN - The Tool for Bayesian Networks and Influence Diagrams," in *Proc. 1st European Workshop on Probabilistic Graphical Models (PGM 2002)*, 2002.

[83] P. Compton and R. Jansen, "A philosophical basis for knowledge acquisition," *Knowledge Acquisition,* vol. 2, no. 3, p. 241–258, 1990.

[84] M. Adams, A. Hofstede, D. Edmond and W. van der Aalst, "Facilitating flexibility and dynamic exception handling in workflows through worklets," in *Proceedings of the CAiSE05 Forum*, Porto, Portugal, 2005.

[85] E. Börger and R. Stärk, Abstract State Machines, Berlin: Springer, 2003.

[86] BPTrends, "Case Management: Combining Knowledge with Process," *www.bptrends.com,* July 2009.

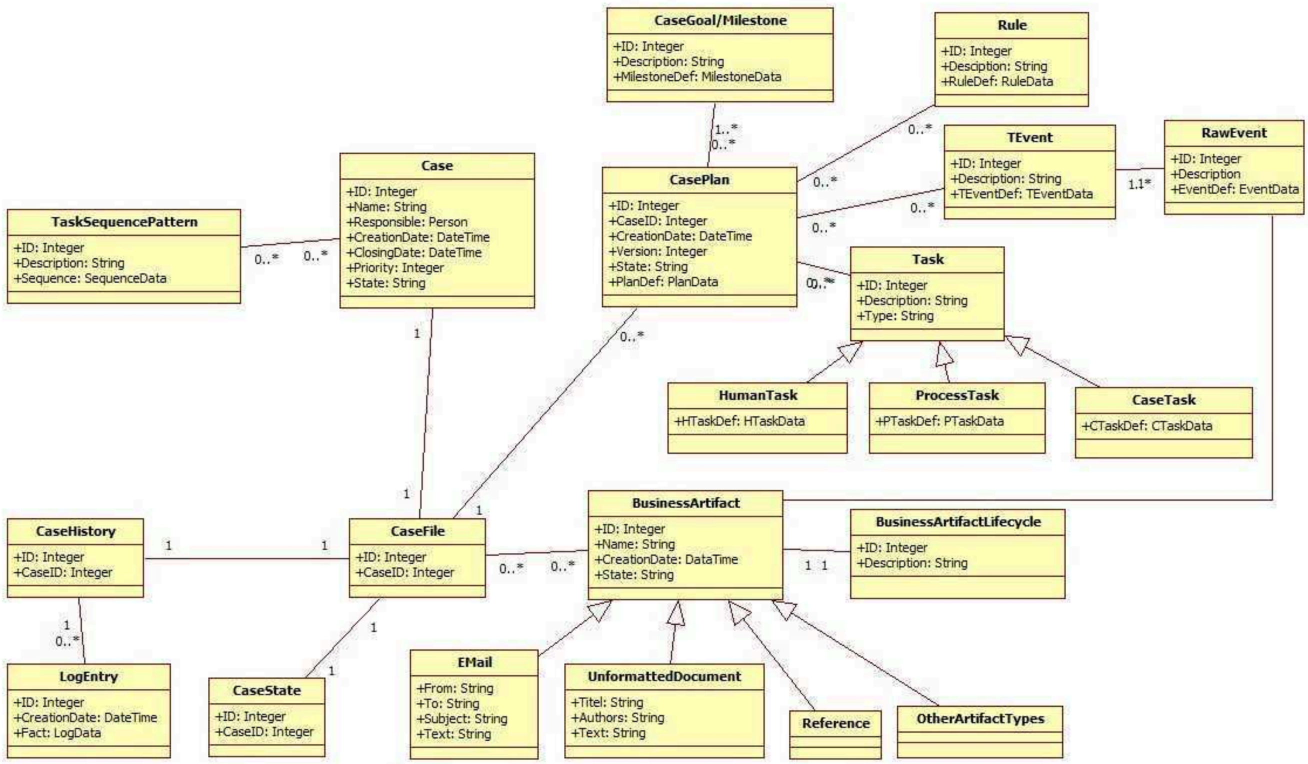

Figure 1: ProSyWis Class Diagram

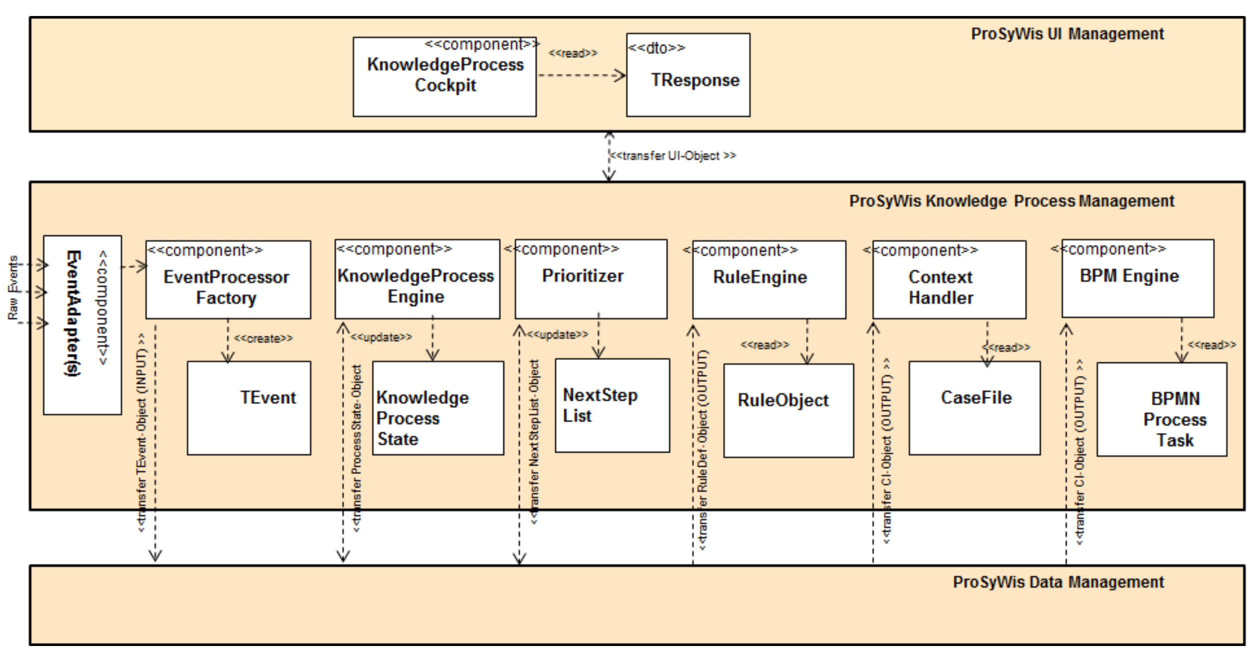

Figure 2: ProSyWis Architecture Overview

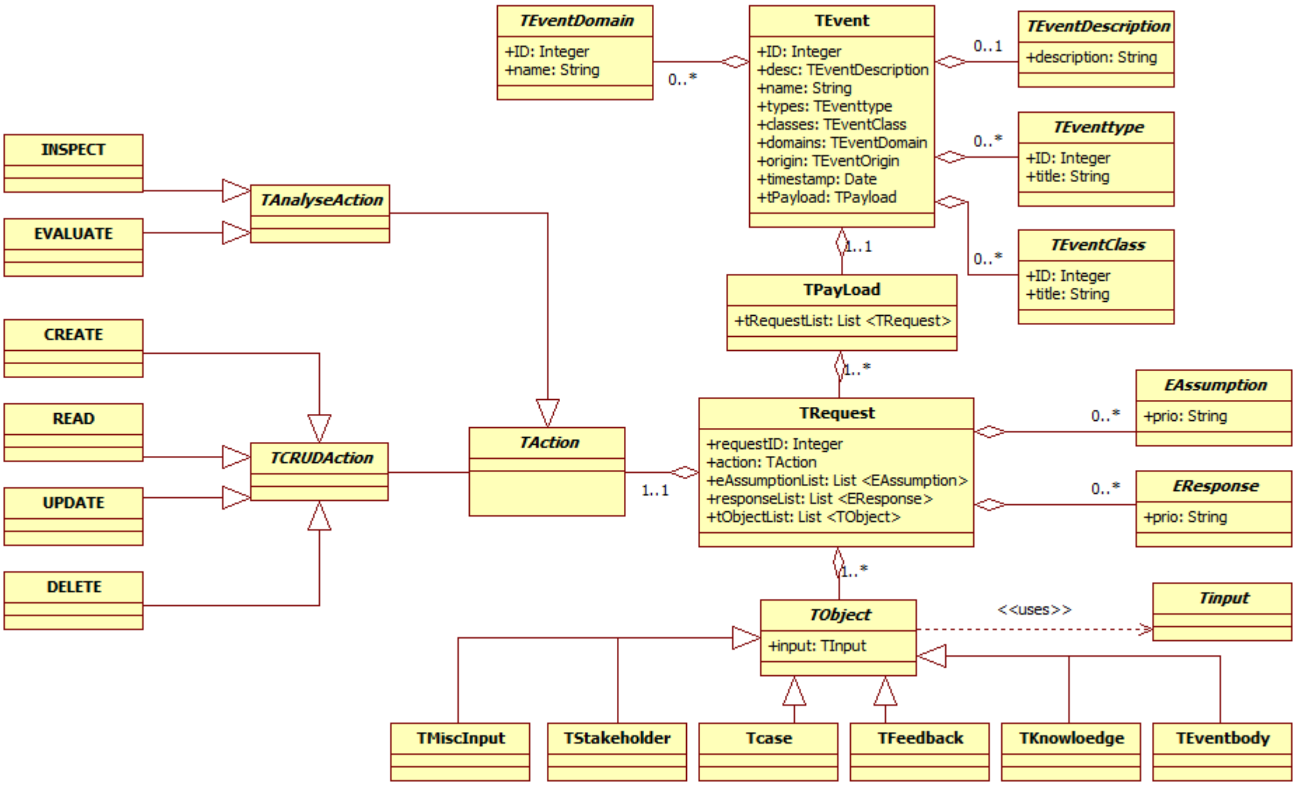

Figure 3: TEvent Structure in ProSyWis (Prototype) [70]

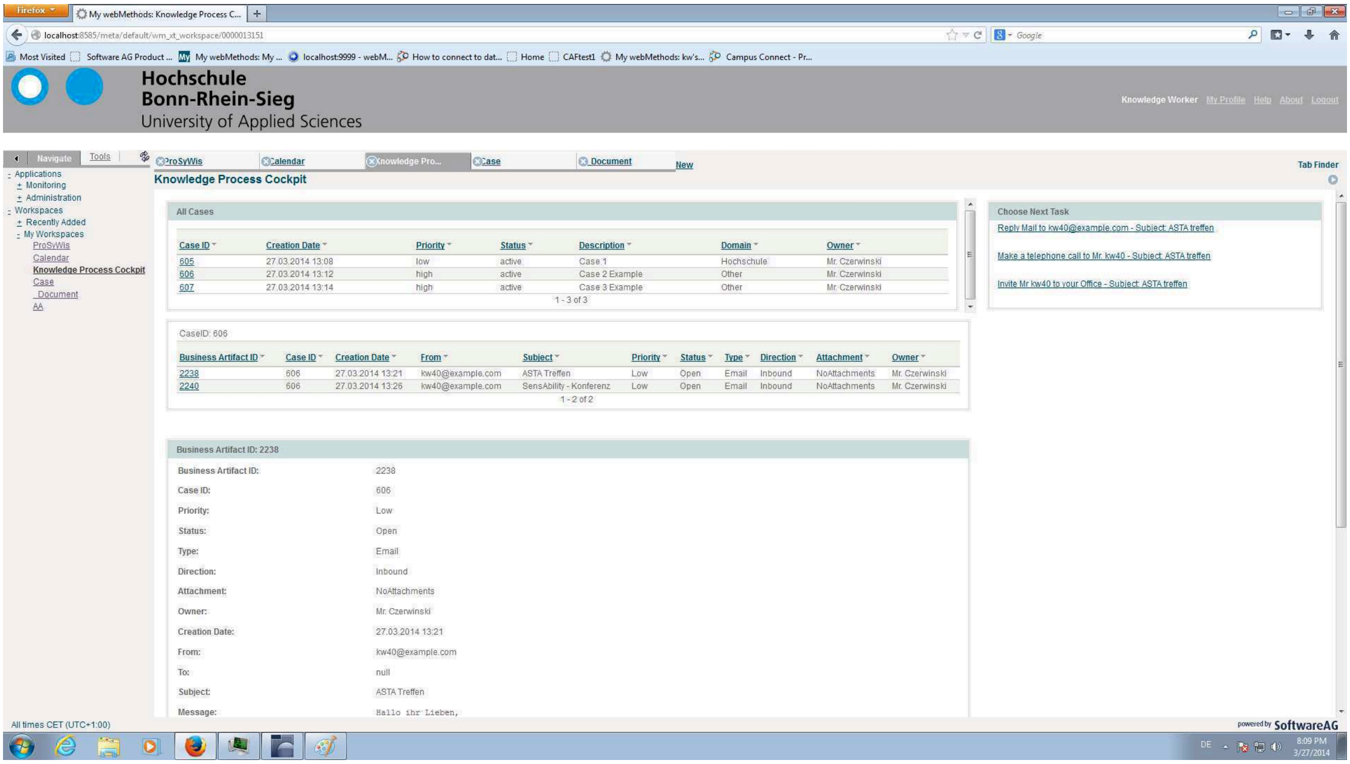

Figure 4: ProSyWis UI Prototype: Knowledge Process Cockpit [71]

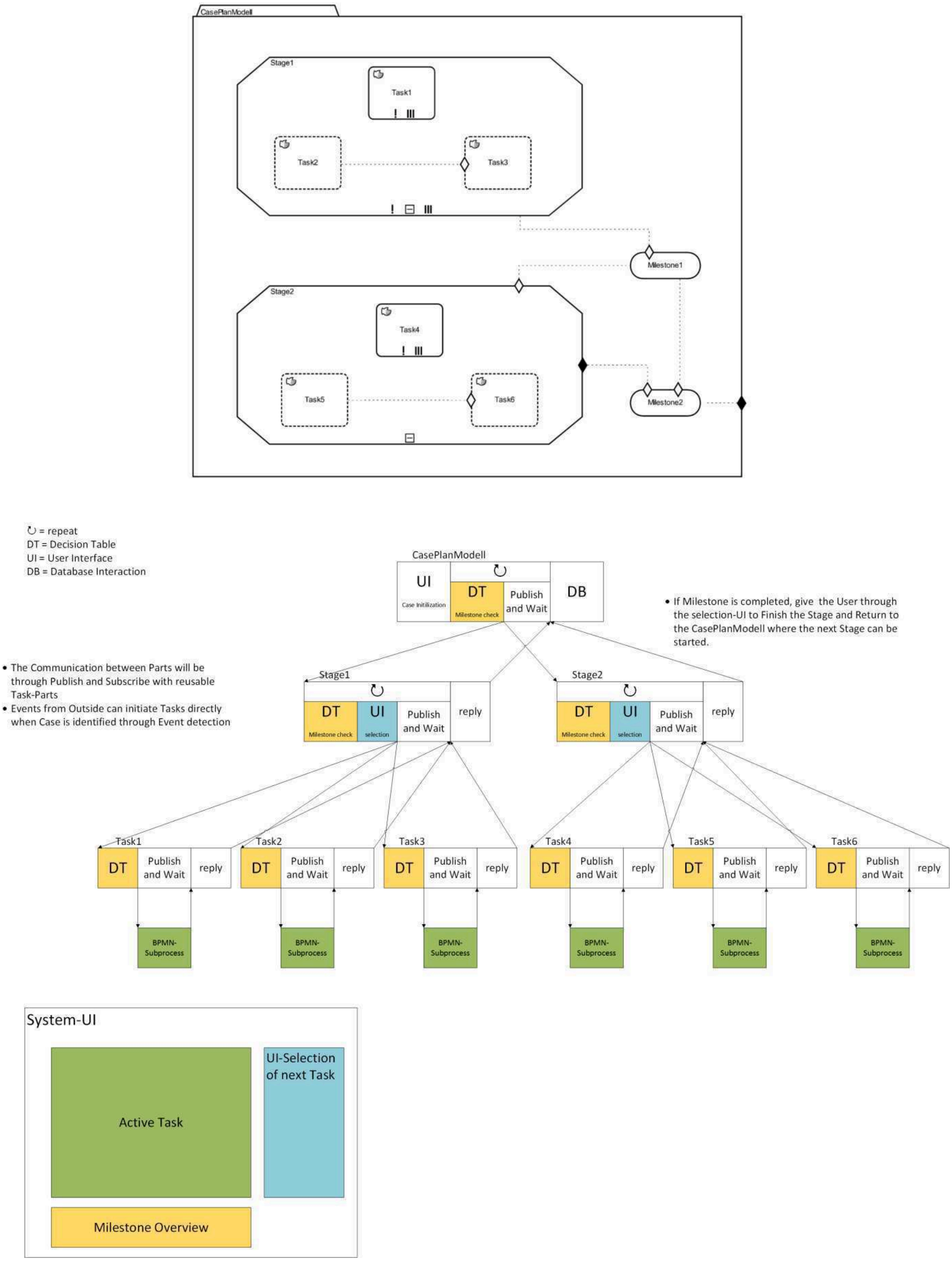

Figure 5: ProSyWis Knowledge Process Handling (Prototype) [71]

	Milestone_1	Milestone_2	Stage_1	Stage_2	TaskSequence	EntryCrit_Milestone_1	EntryCrit_Milestone_2	EntryCrit_Stage_2	Milestone_1	Milestone_2	Stage_1	Stage_2
1					= EMPTY STRING	= FALSE	= FALSE	= FALSE	= FALSE	= FALSE	= TRUE	= FALSE
2	= TRUE										= FALSE	= TRUE

	Stage_1	Stage_2	Task_6	TaskSequence	Task_1	Task_2	Task_3	Task_4	Task_5	Task_6
1	= TRUE				= TRUE	= TRUE	= contains()			
2	= FALSE				= FALSE	= FALSE	= FALSE			
3		= TRUE						= TRUE	= TRUE	= contains()
4		= FALSE						= FALSE	= FALSE	= FALSE

Figure 6: ProSyWis Decision Tables on Case Plan Level and on Stage Level (Prototype) [71]

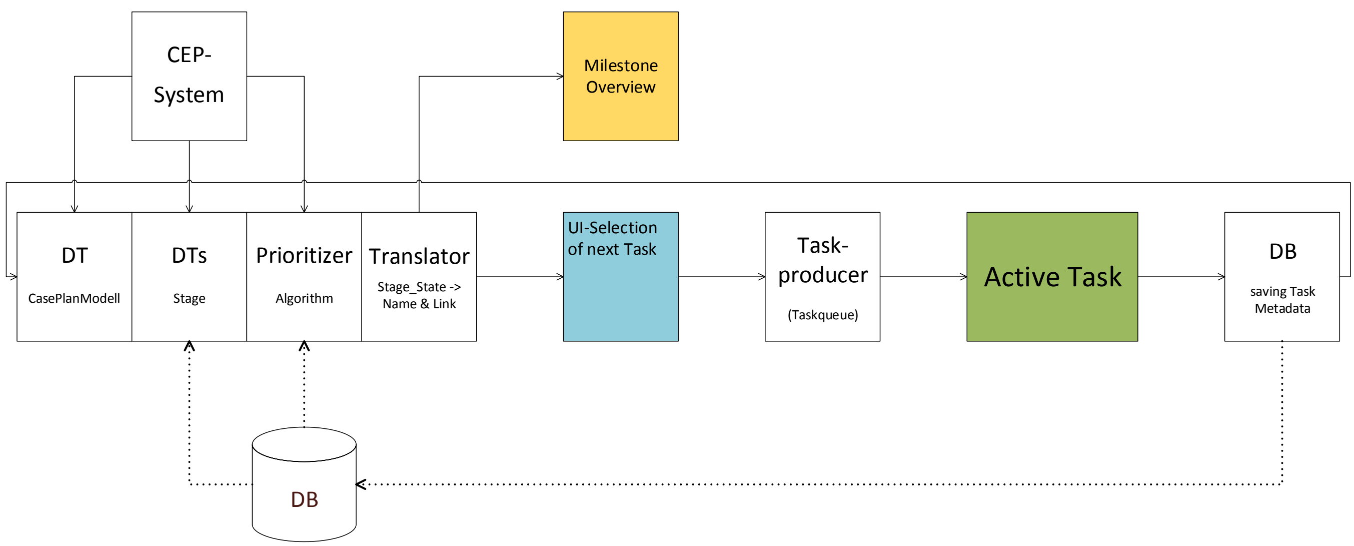

Figure 7: Process Engine Cycle (Prototype) [71]