

Execution Support for Agenda-Driven Case Management

Marian Benner-Wickner, Matthias Book, Tobias Brückmann, Volker Gruhn
paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany

{marian.benner, matthias.book, tobias.brueckmann, volker.gruhn}@paluno.uni-due.de

ABSTRACT

In recent years, mature workflow management technology has become available to support enterprises in engineering and executing well-known business processes. But in the domain of case management, no strict process can be prescribed for tasks due to their high variability. Rather, these tasks require the expertise of case managers who work with information from many different sources. We believe that such processes will benefit from an agenda-driven case management approach. However, implementing an information system supporting such an approach is challenging since it impacts and relies on many aspects of an enterprise's way of handling heterogeneous data. In this paper, we discuss requirements and design decisions arising from the adCM concept and present the results in implementing a corresponding support system.

Categories and Subject Descriptors

H.4.1 [Information Systems Applications]: Office Automation – *Workflow management*.

H.3.4 [Information Storage and Retrieval]: Systems and Software – *distributed systems*.

General Terms

Management, Documentation, Design, Human Factors.

Keywords

Case management, case handling, agenda, business processes.

1. INTRODUCTION

For business processes that are well-understood and highly structured, traditional workflow management (WM) systems can provide comprehensive guidance during process execution. They help an organization's staff to follow the definitions and guidelines described in the form of process models. As a consequence of ongoing industrialization, certain parts of such business processes can be described with such precision and completeness that they become automatable [8]. But not all parts of business processes can be described in such detail. Some are less structured and depend much more on the expertise (i.e. knowledge and experience) of a human expert, a so-called *case manager*. Since a predefined process model often cannot be followed due to the dynamics and uniqueness of each case, a case manager usually follows an

agenda of key process activities or aspects that he or she arranges flexibly as the case demands. Such flexible processes cannot be supported using traditional workflow systems. Instead, systems that support the specific requirements of case handling [2] are required. A system supporting agenda-driven case management (adCM) [3], for example, provides an agenda, i.e. a flexible hierarchically structured list for organizing and keeping track of key activities. Besides the explicit support for case managers' agenda, these systems must also address the fact that a lot of cognitive effort is spent on finding, relating, evaluating, and basing decisions on heterogeneous data. To also support these activities, case managers should be enabled to form a mental model of the information landscape and the relevance and relations of all artifacts – a task that would take a lot of effort and focus to keep in one's head without suitable tool support.

In this paper, we discuss architectural considerations that went into the development of a system that implements key aspects of adCM. In the following section, we will first present the application domain we focused on. We then outline the key concepts of adCM in section 3. Section 4 presents requirements that arise from these concepts and discusses how these requirements are reflected in the system architecture of our prototype. Section 5 presents insights in the implementation of one example design decision. We conclude with a discussion of related work in section 6 and provide an outlook in section 7.

2. THE APPLICATION DOMAIN

Case management (CM) processes can be found in many domains, e.g. in product development [11], customer support, healthcare [7] or claims management [13]. We conducted interviews with case managers and experts especially in the domain of rehabilitation management. We will integrate the resulting insights about their individual CM processes and their specific working environment in this paper.

In rehabilitation management, for example, case managers support patients from the time of an occupational accident through the healing and rehabilitation process up to their re-integration into the workforce. Depending on the past medical history and first diagnostic data, a case manager sets up a healing plan together with the patient and the responsible physician, and then monitors the progress of the different therapies (see Figure 1). Though these CM processes are mostly driven by medical and social responsibility, cost control plays an important role as well. The expenses for therapy measures and especially the process length are key performance indicators. A late re-integration into employment directly affects how long a patient misses social insurance contributions.

In order to improve the probability of a fast and successful vocational re-integration, case managers have to keep track of their cases continuously. They need access to a broad set of different information sources and must be able to extract, evaluate and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'14, March 24-28, 2014, Gyeongju, Korea.

Copyright 2014 ACM 978-1-4503-2469-4/14/03...\$15.00.

<http://dx.doi.org/10.1145/2554850.2554945>

relate that data according to the case. For example, reports of the effectiveness of therapy measures have to be reviewed by the case manager and discussed with the patient and physician.

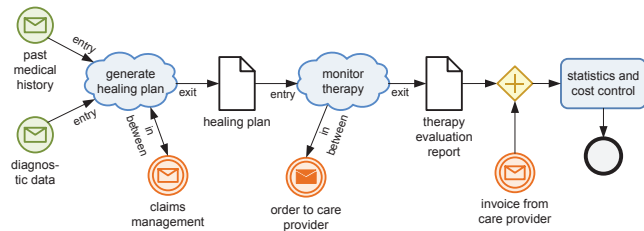


Figure 1. CM processes in rehabilitation management

Our observations in industrial practice have shown that while e.g. a German employer's liability insurance association supports case managers with tailored enterprise information systems such as a digital patient file system including cost estimation functionality, no information systems help the case managers to structure information and to associate the corresponding pieces of information with the main case activities. Consequently, there is still a lot of cognitive effort needed in order to keep track of each case, e.g. by matching incoming reports to the steps in the healing plan. In addition, there is no opportunity to learn from previous case executions. Besides blanket guidelines for the whole domain provided by intranet sources, process knowledge (i.e. knowledge about what and when to do or which information is needed) mainly resides in each case manager's mind. It is hidden to any kind of collaborative knowledge management.

3. AGENDA-DRIVEN CASE MANAGEMENT

Agenda-driven case management has been introduced in literature as a new approach to support case management processes [3]. Due to the high diversity of such processes, the approach reflects the need for high flexibility in both the control-flow perspective as well as the data perspective. It also provides the capability to share process knowledge between case managers. In this chapter, we will introduce the main concepts of adCM.

3.1 Agenda

The pivotal element of agenda-driven case management is a hierarchically structured list containing all entities that a case manager considers important for the case execution – the so-called *agenda*. Ideally, it is derived from a template or best practice for a certain case type. However, the case manager is free to (re-)organize the agenda according to the particularities of each case instance, and may also start with a blank agenda.

To support different levels of abstraction, an agenda can be organized hierarchically. Due to this flexibility, each case manager can use agendas according to his preferences. This is in contrast to the approach of traditional WM trying to bind the process execution tightly to predefined control flow definitions.

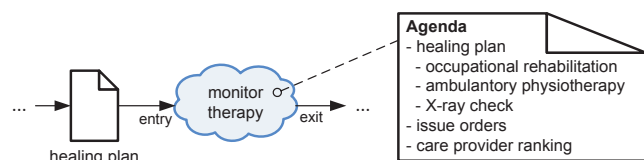


Figure 2. Example agenda in rehabilitation management

Consider the example of the “monitor therapy” process in rehabilitation management (Figure 2). The agenda is initially filled with common important process landmarks like “healing plan”, “issue orders” and “care provider ranking”. They could originate from shared templates containing common agenda-items (see Section 3.3). The subordinate items, however, are specific to the case instance and have been added by the case manager. According to the patient's symptoms, the healing plan contains individual indications (e.g. an ambulatory physiotherapy succeeding an occupational rehabilitation).

3.2 Artifacts

One of the most frequently performed activities of a case manager is to search and explore information related to the case. For this purpose, the case manager will access multiple information sources, both internal and external ones, some of which are structured, while others are not. Based on the information gathered, he builds a mental model of the way he wants to solve the case. We collectively call all information that the case manager gathers *artifacts*. He can work with these by combining, evaluating, rating and relating the information they contain, and structure them by collecting related artifacts in individually defined workspaces.

For example, the healing plan is a key artifact in the rehabilitation domain. This artifact may be based on generally accepted therapy plans, but it may also deviate from these mainstream plans due to certain patient-specific risks or conditions. In this case, a rationale for the deviation from the plan, or a scientific report describing the alternative therapy, can be researched and attached to the healing plan by the case manager.

To organize the task's artifacts in the workspace, the case manager can include information and documents from heterogeneous sources. He can arbitrarily associate artifacts in his workspace with each other, and he can annotate these artifacts and associations with metadata. In the course of the task's completion, the case manager can dynamically re-arrange and relate the artifacts in his workspace as he accesses further sources of information and evaluates them.

3.3 Templates

For certain case instances, the case manager might start on a clean slate, having to figure out for himself how best to proceed and what agenda to follow. In other instances, he may feel the need to adapt process knowledge by means of agenda items or artifacts from similar previous cases that are known to be applicable in a specific context. Consider, for example, a case manager beginning to monitor a therapy. To structure what he has to do in handling the case, it would be useful to identify what is usually done in the “monitor therapy” process, and to check in more detail what other rehabilitation managers recently did in the context of the respective patient's injury. He might take a note of what kind of expertise has to be consulted in the healing steps, which medical datasets need to be checked and what further sources of knowledge should be considered.

In the field of structured business processes, process knowledge is usually well known. It can be acquired by business analysts who design control flow models using notations like Event-Driven Process Chains [12] or Business Process Model and Notation [10]. In the case management domain, control flow models cannot be offered to the case manager due to the high diversity between

cases, as discussed in Sect. 2. Attempts to mine such processes usually result in spaghetti-like models [7].

Hence, providing modular building blocks of process knowledge seem more suitable than trying to identify one “master” process. Consider, for example, the agenda in Figure 2: The top-level agenda items are generic. However, the subordinated items of “healing plan” are very specific to the patient’s indication. And since a case with the same set of indications is very unlikely, these items are not suitable for common process knowledge.

We believe that agenda items and artifacts that are considered to be useful for a common context can be collected in what we call a *template*. Since most cases vary widely and no common process model can describe case management processes in detail, we introduce these small templates as the fundamental process knowledge building block that the case managers can adapt and assemble whenever they seem appropriate.

The templates are defined according to an organization’s documented best practice, according to the case manager’s own experiences, or according to accumulated histories of prior cases. They can provide process knowledge at any level of abstraction - both as a flat agenda skeleton with top-level agenda or as detailed instructions in a deeply elaborated agenda hierarchy with common artifacts. Please also note that a case manager is not required to adopt every item of a template. The defaults may be modified by deleting, adding and revising items according to the situation.

4. REQUIREMENTS AND DESIGN DECISIONS

The notion of case managers working with agendas, artifacts and templates described above results in challenging design requirements for systems that aim to support the agenda-driven case management approach. In this section, we will discuss these requirements. For each aspect, we first motivate the problem using the example scenario from the application domain. We then discuss the requirements and highlight the essence using a bold font. Finally, we describe our design decisions.

4.1 Template Generation

The main goal of the adCM approach is to support case managers during case execution. Besides help to access broad information sources and to keep track of key activities, providing process knowledge from past cases is a key aspect of execution support. For this purpose, the adCM approach introduces templates as described in the previous section. We will now discuss the requirements and resulting design decisions that concern the generation of these templates.

4.1.1 Manual Template Definition

Based on his past case experience, a case manager typically knows best which cluster of agenda items and artifacts are relevant for specific case types. A system supporting the adCM approach shall therefore give him the opportunity to **manually define templates**.

We believe that a case manager’s act of providing a template is spontaneous (similar to the act of sharing content in the context of mobile apps, where a user might deem some sort of information he just found to be interesting to others). The impulse to share content emerges ad hoc and is integrated into the surrounding use case as a simple step.

We encourage adopting this notion and implementing a functionality to share manually defined templates with a community of case managers. We integrated the sharing feature into the agenda’s context menu, so sharing the current state of an agenda (or its parts) as a template can be done in passing during case execution and does not significantly interrupt the case manager’s workflow (see Figure 3). Of course, a dedicated management interface to manage and share templates should also be supported.

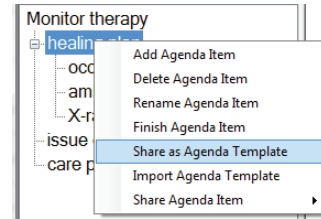


Figure 3. Template generation via sharing function

4.1.2 Automatic Template Generation

However, we also believe that templates can be generated by the system based on accumulated histories of prior case executions. For example, analyzing the sequence in which the case manager works with agenda items may help to find trails in case execution. Also, when mutually comparing completed agendas of past instances with each other, recurring patterns could be identified (e.g. “ambulatory physiotherapy” always follows “occupational rehabilitation”). So the system shall **record the case manager’s steps in completing a case and provide a mechanism that can auto-generate templates** from such instances.

Because of two main challenges, the automatic template generation from past case instances is much more complicated. First, the system needs to monitor user interactions during case execution that are relevant to template generation. For this purpose, we designed an event-driven architecture (EDA) with event-producing client applications and a centralized event-consuming monitoring service component (see Figure 4 for details and Figure 7 for the overall system architecture). In addition, we developed an event data model that defines the events that are communicated via the EDA event channel. If an “agenda item complete” event is broadcasted on the event channel, the monitoring service can store this event in its log file. Later, if all items have been marked as complete, a “case complete” event is sent and the monitoring service can finish the trace within the log. At the same time, the template generator component is notified and checks if the finished case can be used to generate a new template within the knowledge base. The component may also check if there are operating sequences in the knowledge base that are similar to the sequence in which the agenda items have been completed in the recently finished case. Such matching sequences could be isolated as well-known process parts. This is a connection point for process mining techniques as described in [1].

But when monitoring common agenda items, there is a second challenge that concerns template generation: agenda items can be named at will. So they may pollute the knowledge base with similar named entities that are semantically the same. This “conceptual noise” can hamper cluster mining of past instances. In order to cope this noise, we integrate a component into the system architecture that can provide key concept names of the domain. This might be a controlled vocabulary or better yet, a domain ontology containing knowledge about concept relationships (e.g. syno-

nyms). When generating templates by mining common agenda items or artifacts, such a component can be used to filter the noise within agenda names and to find matches based on synonyms that would otherwise remain undiscovered. If a domain ontology is not at hand, it can be mined from document warehouses or intranet web sites [9].

There are various algorithms that can be used to find matches between hierarchical data structures such as the agenda. For example, tree pattern matching algorithms (see [15], [16]) that are usually developed to match different XML files can be applied.

4.2 Template Recommendation

Of course, generated templates stored in the knowledge base anyhow have to be made available to the case manager. In the following, we will describe the challenging requirements on template recommendation. According to adCM, the case manager is supported during case execution by interacting with an agenda and with artifacts aligned on the workspace. So we will discuss how an adCM-supporting system can be designed to recommend templates when the case manager interacts both with the agenda and the workspace.

On the one hand, advertising templates may be conceived as intrusive if they disturb case managers during knowledge-intensive activities. On the other hand, if case managers have to proactively ask for templates, they may miss valuable recommendations some other time. So the system has to **strike a balance between intrusive advertising and missing execution support**. It needs to be sensible to the current context and provide recommendations just in time when they are suitable and when they do not disturb the case manager.

4.2.1 Agenda Perspective

In the field of well-known business processes, traditional WM systems provide process knowledge using a fixed, predefined, and ordered collection of activities. In case management, one might think that an agenda (derived from a template upon beginning work on a case) corresponds to such a sequence of process activities. However, due to the dynamics of CM processes that require a *flexible* agenda, it is unlikely that there is a template that already contains every appropriate agenda item for the specific case, and a matching template may also be identified later during the case execution (e.g. when the case manager re-organizes the agenda).

So, in contrast to the approach of WM systems, an adCM-supporting system needs to provide templates just in time. Recommending templates that are applicable to the current case's context (in terms of the agenda's state and especially the associated artifacts) can help to avoid redundant cognitive effort in managing the agenda. To achieve this, we propose to observe and analyze the case's context during its whole lifecycle.

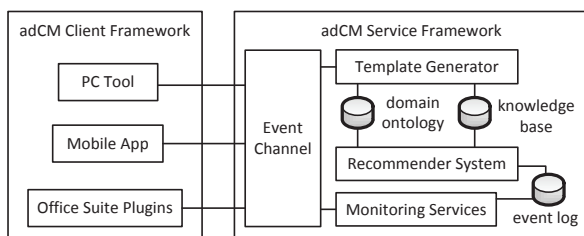


Figure 4. Event-driven architecture for template generation and recommendation

In the previous section, we already introduced an event driven architecture for template generation (see Figure 4). This architecture is also useful to observe and analyze the current context of a case. We designed a recommender system component connected both to the knowledge base and to the event log (see Figure 5). It is responsible to monitor the case execution using the event log and to trigger an event once a matching template can be found in the knowledge base according to the current context. Based on configuration details, the client may or may not react on that event by recommending the template to the user.

4.2.2 Workspace Perspective

To reduce the time spent on finding information relevant to the current context within Terabytes of enterprise data, we also decided to support the case manager in searching for artifacts (see section 4.3 for more information on information access support). To this, the domain ontology as introduced in section 4.1.2 is very useful: Since templates are generated with the help of common concepts stored in such a dataset, it is easier to discover them when the case manager also uses these concepts as search keywords. So, to better find templates common to the current context, we decided to integrate an auto-complete function within the artifact search UI that uses tags from the domain ontology.

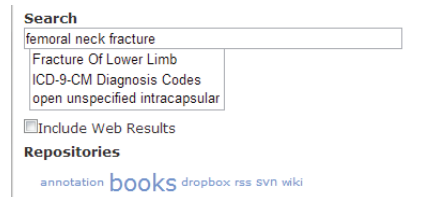


Figure 5. Artifact search support using ontology auto-complete

When the case manager selects an artifact from the result list that is part of any template, the recommender system notifies him. If the case manager decides to use this template, the artifacts and agenda items that may be associated to the selected artifact are integrated into the existing agenda and workspace.

Please also note that, unless templates has been recommended and adopted sometime, they are not verified, i.e. they are not proven to be common. So, when recommending templates, the system learns from user feedback (e.g. whether the user accepts or declines the recommendation).

4.3 Unified Information Access

Due to the diversity of case management processes, a case manager works with a broad range of different data sources, each with different interfaces, access rights and search modalities.

Consider the example process from the application domain. When starting to define the healing plan, any kind of diagnostic data or files of past medical history are needed. In order to find physicians whose expertise matches the patient's diagnosis, the case manager considers the website of the corresponding medical association. In addition, he consults a shared spreadsheet containing individual experience about a physician's expertise or his hospital. All these data sources are accessible via different information systems or applications and stored locally or hosted on remote servers. Some sources may be structured while others are not, and some of them require authentication.

4.3.1 Obtain and Work With Artifacts

To reduce the time spent on switching between different access modalities, the system should **assist the case manager in accessing these sources**.

To simplify the case managers' access to heterogeneous data sources, providing a unified user interface would be helpful. Hence, we integrated enterprise search technology that crawls enterprise data sources and provides a single query interface in the adCM tool's UI (see Figure 6). Since implementing vendor-specific database and cloud storage connectors is expensive, we use a connector framework that can be extended as needed. It also manages direct data access from clients as well as authentication information required to consider both data security and access comfort (e.g. single-sign-on functionality).

But not every found resource can be offered by the search UI as a link to a file. For example, legacy systems dynamically compose information on screen in response to user queries. The results are volatile and cannot simply be included into the workspace. For such resources, virtual documents have to be defined [17]. They are stencils defining a file representation of a query result (e.g. a business card of an expert found in a CRM database).

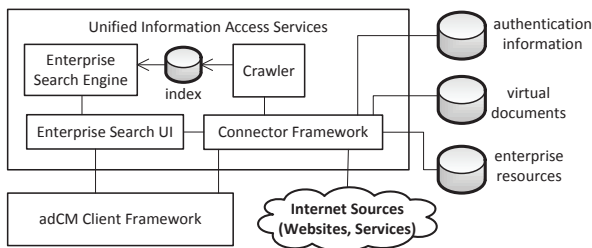


Figure 6. Unified Information Access Service Components

With respect to the workspace concept mentioned in the previous chapter, the system shall also support **storing, relating and annotating artifacts from various sources**. This is challenging since artifacts can be of any type.

In order to fulfill the requirements of filing the case-specific data, the system architecture has to provide a suitable case repository. We use a file-system based repository, because it can handle most resources, including links (e. g. to read-only sources). It is associated with exactly one case and only accessible by its manager.

We also designed a user interface that enables the case manager to define both binary and n-ary relations (groupings) between artifacts. Semantics can be added to these relations by typing keywords like “part of” or “utilizes”. Using drag drop functionality, each artifact can be related to an agenda item. Such a relation, for example, can imply that an artifact is needed to finish the activity, or that it will contain the results after agenda-item completion. Of course, all these relations between artifacts themselves and with agenda items, together with their semantics, are stored in the knowledge base when the case is finished.

Due to page restrictions, we kindly refer to our past work [4] for details about the design of the annotation functionality.

4.3.2 Mobile Access

Many case managers also conduct their case work outside their office. In the rehabilitation domain, they visit patients, physicians or the prospective employer. Often, they consecutively visit locations concerning different cases. In doing so, they **need mobile**

data access to all corresponding patient files and any additional case-specific data.

Since the repository shall also be accessible on mobile devices, there is a need to either synchronize it between all devices or to store it on a server, providing remote access to each device. These alternatives have to be discussed along requirements specific to mobile applications like storage space, bandwidth and performance. Today, many mobile devices can be equipped with additional storage capacities large enough to store a case's workspace. Both bandwidth and performance restrictions depend on the connection: Mobile data is usually very restricted and underperforms compared to Wi-Fi.

We decided to include a file hosting service component that manages the repository on a server and synchronizes it with each local repository. To improve the performance of mobile data access, full workspace synchronization (i.e. downloading all case-specific artifacts) only takes place if the device has Wi-Fi connection. However, the plain list of artifacts is always synchronized independent of the connection type. Only if the case manager wants to access an artifact that has not been fully synchronized yet, it will be downloaded individually.

5. IMPLEMENTATION

In this section, we will introduce our implementation of the design decisions and requirements of an adCM tool. First, we will show the overall system architecture and then describe how the components are involved during usage.

5.1 Overall System Architecture

To implement a case management system that supports the adCM approach, we propose a system architecture as shown in Figure 7.

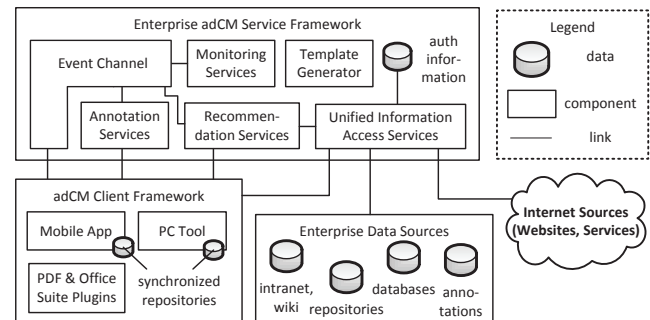


Figure 7. High-level view of the adCM system architecture

The system architecture shows two high-level components: The adCM Client Framework and the Enterprise adCM Service Framework. Containing a mobile application, a PC tool and plugins for office applications, the adCM Client Framework enables multimodal access to the agenda-driven case management services. A plug-in architecture is used to embed adCM-specific functionality (e.g. unified annotation interface) into the case managers' familiar working environment and preferred tools.

The Enterprise adCM Service Framework provides services to search, find, access, and annotate artifacts from both enterprise and external sources. It is also responsible for monitoring the case execution and generating templates from past executions. Using that knowledge, different aspects of execution support can be provided, e. g. artifact recommendation. The event channel is used

to distribute event messages between the adCM services themselves and to the Client Framework components.

5.2 Usage in Practice

We will illustrate how the adCM system prototype can support case managers using an example scenario from the rehabilitation management domain:

Mr. Smith, a case manager in rehabilitation management, has recently become responsible for managing the femoral neck fracture rehabilitation of Mr. White. After logging into the adCM tool, he creates a case instance for monitoring Mr. White's therapy. While planning the agenda and populating the workspace with important artifacts, Mr. Smith notices there are a few steps and artifacts that he routinely includes for all cases of this type, such as typical therapy options or common works of references for femoral neck fractures. Hence, he decides to share the general items as a template, so that he or his colleagues may re-use them in subsequent cases (see Figure 8).

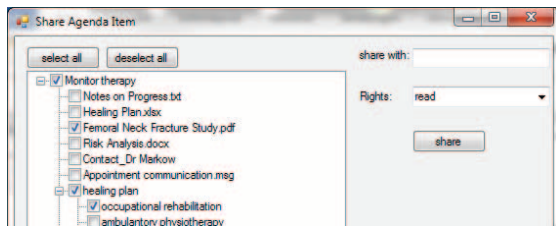


Figure 8. Sharing a template

He continues to work on the case by searching for relevant artifacts. Among other data, he needs detailed information about femoral neck fractures and their therapy options. But such information is spread over several enterprise repositories. He therefore calls up the enterprise search function through a hotkey and enters his search terms. The results show a) pictures from a sharepoint folder, b) Word and PDF documents from a version control repository, c) a link to a wiki page, and d) an annotation added to a source by a colleague (see Figure 9).

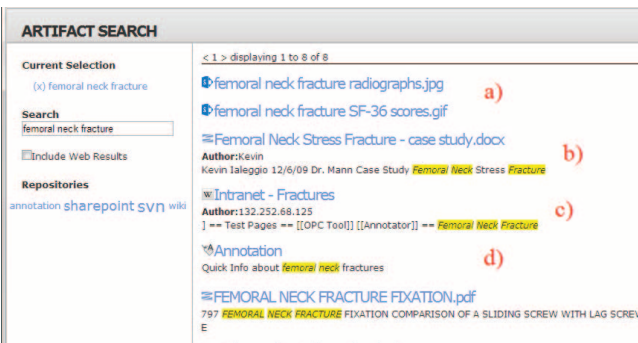


Figure 9. Enterprise search UI exposing heterogeneous data

Most of the results refer to artifacts stored in a file system, so that they can be easily linked to. However, results such as annotations are stored in a database and may involve data drawn from several tables. So when Mr. Smith wants to access or drag an annotation onto the workspace, a virtual document is generated that joins the dispersed information and presents it using XSL transformation (see Figure 10). Mr. Smith follows the document link to see the annotation of his colleague in its context (here: an annotated Wikipedia article of hip fractures containing a femoral neck section).

During his work, Mr. Smith collects artifacts, stores and relates them to each other in the adCM tool's workspace. For example, to keep track of his communication tasks, he manages case-specific contacts and messages in a joint grouping (see Figure 11).

Annotation

Quote: Femoral neck fractures involve the narrow neck between the round head of the femur and the shaft.

Text: Quick Info about femoral neck fractures

User: UserOPC

Document link

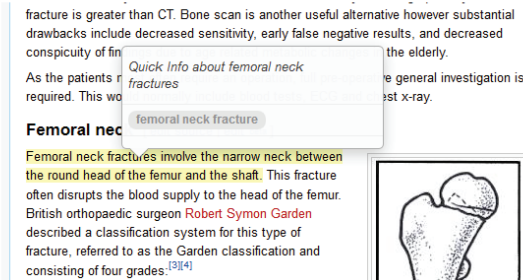


Figure 10. Virtual document representing a website annotation

He also expresses the relationship between the risk analysis document and the femoral neck fracture study by drawing a connection. To describe this connection more precisely, he adds semantics to the relationship, denoting that the risk analysis "utilizes" the study.



Figure 11. Example artifacts in rehabilitation management

Some days later, a similar case is assigned to Mr. Smith. The new patient not only suffers from a femoral neck fracture, but also has low perceived self-efficacy, indicating the need for a more complicated therapy. Mr. Smith remembers the template he shared earlier, and chooses to create the new case instance based on it. The initial agenda shows a subset of the activities and keywords from the original case. On the workspace, the common study of femoral neck fractures is already imported (see Figure 12). Mr. White can therefore start right away with searching for more case-specific information, such as psychological support.

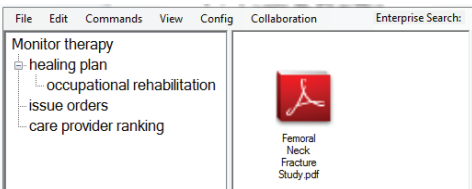


Figure 12. Template with items for femoral neck fractures

6. Related Work

Discovery and enhancement of process knowledge by monitoring process execution is one of the key concepts of process mining [1]. However, process mining workflows and technologies do not focus on mining unstructured processes, and the available tools neglect the data perspective as they concentrate on control flow, organizational and performance perspectives [7].

The KISS project [5] is very promising in the field of agile business process support, with an approach that is very similar to adCM. However, we prefer a method that is straightforward to use for case managers as it does not include any build-time activities. In adCM, process knowledge is either generated automatically or manually defined alongside case execution, using a well-integrated sharing function.

The “case handling paradigm” [2] is an approach to describe processes with certain control flow variability. A case consists of activities, data objects, precedence relations between activities, and relations between them. However, adCM does not prescribe any precedence relation or control flow definition. Instead, we provide an agenda as a guideline to case managers, but no explicit order of agenda items or control flow elements that would introduce some notion of a structured process.

From the process model notation perspective, the BPMN standard [10] can express ad-hoc processes containing unordered activities. So whenever a BPMN model contains an ad-hoc process, it can be supported using the adCM approach.

7. Conclusion

The characteristics of case management processes differ considerably from structured processes. Hence, they require different support than traditional WM tools can provide. Based on agenda-driven case management (adCM) as an alternative approach to support unstructured processes, we therefore identified requirements for systems that aim to support such processes, described the design decisions made during the development of an adCM tool prototype, and finally showed that typical case management use cases such as planning activities or searching, relating and evaluating information can be supported by it.

With the adCM tool architecture established, our ongoing and future work is focusing in improvements of the individual system components. To improve the recommendations, for example, we are working on the integration of a suitable tree mining algorithm. Experiments shall reveal which algorithm configuration fits best to generate templates from past instances. This is challenging because one has to reflect the very specific structure of the knowledge base. We also plan experiments with different recommendation configurations. They shall reveal how templates can be recommended as unobtrusively as possible. Finally, we want to analyze how complex event processing techniques [14] can help to enrich the information about past case instances and templates with statistics. For example, the mean duration of cases derived from a template could be considered in order to recommend templates according to their expected efficiency, too.

We believe that these mechanisms will help case managers to seamlessly receive more targeted recommendations, and are striving to evaluate benefits of these improvements through observations of practitioners working with the prototype on actual cases.

8. ACKNOWLEDGMENTS

We thank Pascal Groß, Alexander Kalinowski, Marcel Kleine-Boymann, Wilhelm Koop, Yordan Terziev and Matthias Verholen for their contribution to implementing the prototype.

9. REFERENCES

- [1] Wil M. P. van der Aalst: *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer 2011, ISBN 978-3-642-19344-6, pp. I-XVI, 1-352
- [2] Wil M. P. van der Aalst, Mathias Weske, Dolf Grünbauer: Case handling: a new paradigm for business process support. *Data Knowl. Eng.* 53(2): 129-162 (2005)
- [3] Marian Benner, Matthias Book, Tobias Brückmann, Volker Gruhn, Thomas Richter, Sema Seyhan: *Managing and Tracing the Traversal of Process Clouds with Templates, Agendas and Artifacts*. BPM 2012: 188-193
- [4] Marian Benner-Wickner, Matthias Book, Tobias Brückmann, Volker Gruhn: *Cloud Storage of Artifact Annotations to Support Case Managers in Knowledge-Intensive Business Processes*. WISE Workshops 2012: 92-104
- [5] Simon Brander, Knut Hinkelmann, Andreas Martin, Barbara Thönssen: *Mining of Agile Business Processes*. AAAI Spring Symposium: AI for Business Agility 2011
- [6] Christian Herrmann, Matthias Kurz: *Adaptive Case Management: Supporting Knowledge Intensive Processes with IT Systems*. S-BPM ONE 2011: 80-97
- [7] Ronny Mans, Helen Schonenberg, Minseok Song, Wil M. P. van der Aalst, Piet J. M. Bakker: *Application of Process Mining in Healthcare - A Case Study in a Dutch Hospital*. BIOSTEC (Selected Papers) 2008: 425-438
- [8] The McKinsey Quarterly, "Eight Business Technology Trends to Watch," December 2007.
- [9] Roberto Navigli, Paola Velardi: *Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites*. *Computational Linguistics* 30(2): 151-179 (2004)
- [10] OMG. *Business Process Model and Notation (BPMN)*. Object Management Group, dtc/2010-06-05, 2010.
- [11] Anne Rozinat, Ivo S. M. de Jong, C. W. Günther, Wil M. P. van der Aalst: *Process Mining Applied to the Test Process of Wafer Scanners in ASML*. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 39(4): 474-479 (2009)
- [12] A.-W. Scheer: *Business Process Engineering: Reference Models for Industrial Enterprises*, Springer-Verlag, 1994
- [13] Etienne Wenger: *Communities of Practice: Learning, Meaning, and Identity*, Cambridge University Press, 1999.
- [14] Eugene Wu, Yanlei Diao, S. Rizvi: *High-performance complex event processing over streams*. SIGMOD 2006: 407-418
- [15] Jingtao Yao, Ming Zhang II: *A Fast Tree Pattern Matching Algorithm for XML Query*. *Web Intelligence* 2004: 235-241
- [16] Mohammed Javeed Zaki: *Efficiently Mining Frequent Embedded Unordered Trees*. *Fundam. Inform.* 66: 33-52 (2005)
- [17] Li Zhang, Michael P. Bieber, David E. Millard, Vincent Oria: *Supporting virtual documents in just-in-time hypermedia systems*. *ACM DocEng* 2004: 35-4