Flexible Business Processes - Evaluation of Current Approaches

Thomas Burkhart, Peter Loos

Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI)

1 Motivation

The way people work, especially in group-based or creative processes, is getting more and more unpredictable. This is based on today's fast-changing business world which is characterized by dynamic conditions. In order to fulfill these flexible business situations, there is a need for the use of support systems. However, classical support systems do not offer the necessary agility to support today's rapidly changing or generally unpredictable workflows due to their static runtime execution. Thus, modern workflow systems have the need to provide a flexible, but process aware execution at runtime that do not force users into a pre-defined way of doing their work. In addition to these new requirements, they must still support users during the execution like classical systems already do (Herbst 2004, p. 4). However, to find the right balance between both tasks is highly challenging.

According to prior research papers (Schwarz et al. 2001; Wargitsch 1998), the examined systems were not appropriate for supporting flexible or so called ad-hoc (cf. section two) workflows. Thus, this paper evaluates the state-of-the-art of current systems based on the viewpoint which requirements have already been achieved and for which requirements technical concepts are still to be developed. Furthermore, the paper clarifies whether there is a comprehensive approach which can be used in every business context or which approach can be used in which specific context.

To achieve such an evaluation, the paper defines the basics of flexibility and details concerning business processes and process support within section two. Subsequently, a framework with six criterions is derived which allows a structural evaluation (cf. section four) of the reviewed approaches and technology (cf. section three).

2 The flexible workflow framework

Before talking detailed about flexibility in terms of business processes or workflows, it is necessary to take a closer look at the meaning or interpretation of the term flexibility. In the sense of business, flexibility could be projected on the ability to change or alter the way things are handled (i.e. processes) without facing serious complexity issues. This definition is closely related to the process-as-a-guide interpretation promoted by Adams et al. (2005), in which processes simply provide a guide-line while the appropriate way of handling single tasks is chosen on an as-needed basis. In Reijers et al. (2003), process models define the normal way of achieving a goal, but still offer the possibility to alter this way based on available case data. Sadiq et al. (2001), on the other hand, describe flexibility as the ability to deal with processes that are only partially defined at build-time. In contrast, Soffer (2005) splits flexibility in a short-term (i.e. deviations from a given model) and long-term (i.e. evolution of processes) view. Another interpretation can be exemplified by the definition given in Greiner and Rahm (2003) which simply uses the term flexibility in association with exception handling capabilities. A further point of view is captured through the concept of adaptive workflow which aims at providing flexible processes through adaption of given, prescriptive process models in case of exceptions or policy changes (cf. chapter three).

To look closer into the term flexibility in business, the first definition that is often found is at the most high-level kind of flexibility, in other words, on the organizational level and thus is called *strategic flexibility*. According to Adamis et al. (2005), strategic flexibility is divided into two dimensions. The first one concerns the variation and diversity of strategies, while the second one describes the capability of a company to rapidly shift from one strategy to another. As changes in business strategies often affect one or more business processes in their structure or resource allocation, it is necessary to take a closer look at flexibility in single business processes. According to Rosemann and Recker (2006), business process flexibility itself consists of two parts. First of all, there is an extrinsic trigger for a change process and secondly the intrinsic process adaption itself. Following this distinction, business process flexibility is the capability to yield to externally triggered changes by modifying the process accordingly. However, both Rosemann and Recker (2006) and Regev and Wegmann (2005) stress that, while being able to react to change, it is of particular importance to also maintain the stability of the business process. Change may only concern the parts of a process that are actually affected (i.e. alter the affected parts without replacing the whole process (Rosemann and Recker 2006)), which can only be achieved by an appropriate structure (not too simple or too complex) with respect to the problems and tasks faced by companies (Regev and Wegmann 2005).

After getting a basic idea on what business process flexibility constitutes, an overview of different process classifications and a more detailed demarcation of flexibility and its characteristics clarify the main concepts in more detail. Based on

the continuum given by Huth and Nastansky (2000) business processes can be classified as follows: *Ad-hoc processes* represent the most flexible type of processes because their actual execution path is completely defined at run-time with no given structure forcing a certain course of action. *Semi-structured processes* or case-oriented processes (Dellen et al. 1997) can be seen as somewhere in between ad-hoc processes and structured processes. They follow certain rules but cannot be entirely standardized. *Structured processes* represent the well-known classical processes with full automation capacity. Structured process-models determine - a-priori - the complete process flow, agents, alternative paths etc. and execute the same process again and again.

The next step when considering flexibility in business is to analyze the characteristics of flexibility and the underlying change processes and more over to demarcate the term flexibility. According to Regev et al. (2006), the taxonomy of flexibility in business processes presents three notions of flexibility; thus, classifying it by the types of changes it enables: *The abstraction level of change* defines whether a given change process concerns the type or instance level. *The subject of change* notion comprises a set of five perspectives: the functional, operational, behavioral, information and organizational perspective. *The properties of change* characterize the enactment of the change process. The proposed notions include the extent, duration, swiftness and anticipation of change. The extent of change can either be incremental (i.e. small changes to a process) or revolutionary (i.e. business process reengineering, for instance). The duration defines if a change is only valid for a short period of time (i.e. temporary) or until the next major change occurs. Swiftness simply states if a change is applied only to new instances (i.e. deferred) or also to all running instances using a migration technique (i.e. immediate change). The last property, anticipation, describes if a change occurs on a planned basis or is implemented ad-hoc.

Using all the criteria discussed above, changes are now describable in a variety of aspect. The last set of properties of business processes reviewed in this paper is a set of three process characteristics introduced by Gebauer and Schober (2005). These criteria were derived to enable a deterministic measure for the expected need of flexibility of a process regarding the implemented supporting system: *The uncertainty* of a process refers to the difficulty of predicting what tasks and resources will be necessary to successfully perform a process. Uncertainty can be subdivided into environmental (i.e. external) and structural (i.e. internal) uncertainty. *The variability* characterizes the versatility of tasks that is necessary to execute a certain business process. *The process time-criticality* measures the amount of steps within a process that depend on a fast and timely execution.

Now that the need for flexibility in business processes and their supporting systems is established, it is now necessary to take a closer look at how to achieve and characterize flexibility features in business process support systems. As useful as flexibility in a business process support system might be, there are two problems one should always be aware of. When looking for a flexible solution, it is impera-

tive to know what kind or degree of flexibility is necessary in the specific context. Excessive flexibility might limit usability or increase complexity, while insufficient flexibility may limit the use of a system in exceptional situations (Gebauer and Schober 2005). As mentioned in section one, another important trade-off that should always be considered is the one between support and flexibility. Flexible systems often trade support for flexibility (Pesic et al. 2007a). Groupware systems, for example, allow people to execute single tasks, but lack the necessary support as they are not at all process aware (Reijers et al. 2003).

After establishing a profound conceptual basis it is now possible to derive a set of criteria (cf. table 1) to evaluate and compare different approaches on providing business process flexibility support in the next section.

Table 1: Evaluation Framework

Criterion	Characteristic
Implicit Flexibility	This criterion refers to the three process classes defined above: ad-hoc (full support), semi structured (medium support) and structured (no support) processes. A classification under this criterion may not always be a hundred percent accurate as a technology may offer support for multiple process classes. In this case it is always assigned with the most flexible class it supports.
Degree of Freedom	The second criterion measures the rigidity of a solution in leading the user along a pre-defined path. A system which imposes a low degree of freedom (no support) may be, for example, an adaptive workflow system which provides a fixed process path with deviation mechanisms only in the case of exceptions. Consequently, it consists of fixed parts that a user has to follow during the execution of a process instance. Medium degree of freedom (medium support) usually occurs in systems offering a-priori flexibility with at least some pre-defined part. These pre-definitions may range from a set of activities with no precedence relation to definitions containing some entirely structured parts. High degree of freedom (full support) is a main characteristic of systems providing ad-hoc process support.
Late Modeling	This criterion describes the extent of late modeling features offered by a system. The extent may range from no support over medium support to full support. Approaches with medium support offer the possibility to at least define some parts of the process model at run-time (e.g. determine the precedence relation of activities; pick activities from a set of alternatives). Full support also enables the execution of processes with no model at all.
Late Binding	This criterion describes the extent of late binding features offered by a system. It may range from no support over medium support to full support. Medium or full support is dependent on two sets of options that might be provided. The first set of options facilitates some basic features like the dynamic assignment of actors or resources (resource binding), while the second set of options on the other hand enables users to e.g. assign implementations to tasks or select another execution alternative (execution binding). If one of those sets is supported, a medium rating is attributed. Full support can only be achieved if features from both sets are provided.
Support	The support (or process-awareness) criterion measures a system's capabilities in supporting the user in his execution of certain processes. The least support is offered by groupware systems or even some systems with support for ad-hoc process execution, as they are not aware of the underlying business logic. Systems are attributed with a medium rating if they are process aware and offer some user support. This might include semi-structured or ad-hoc processes with at least some recommendation, templating, knowledge-management or best-practice support. Full support is offered by adaptive workflow systems with entirely predesigned processes.
Adaptability	Adaptability measures the features of a system to enact changes on predefined process models in case of exceptional or unexpected circumstances. It ranges from no support (exception handling has to be done outside of the system) over medium support (ad-hoc changes or evolutionary changes) to full support (both ad-hoc and evolutionary changes).

3 Flexible workflow technologies

In this section, current flexible workflow technologies, all of which are selected and reviewed by means of an extensive literature research, but without any field test, are presented to provide an overview in conformity with the market (cf. table 2). Some of these technologies are currently available for business use; others are research approaches only.

Table 2: Overview over available flexible workflow technologies and their imple mentation

Technology	Characteristic
Caramba (Dustdar 2004)	Caramba is a process aware collaboration system that aims at supporting the whole process continuum from modeled processes to ad-hoc processes with a distinct focus on team-based ad-hoc processes. To provide these features, a sophisticated object-oriented basis comprising *organizational*, *dynamic* and *business* objects as well as *links* between those objects is utilized.
Collaborative Task Manager (Stoitsev et al. 2007)	This framework aims at an unobtrusive support for unstructured, knowledge-intensive business processes. It enables modeling, exchange and re-use of light-weighted, user-defined task structures. Furthermore, it enables the re-use of distributed process knowledge (best-practices and artifacts) through a shared central repository. To realize this, the framework comprises a few basic concepts (artifacts, human actors, tasks and task patterns) which allow support for full late modeling. These features have been implemented within a prototype which is deployed as a Microsoft Outlook add-in backed by a MySQL server.
Commius (Burkhart et al. 2008)	The Commius approach aims at providing a discrete, but purposeful support for the highly dynamic processes of small and medium enterprises by utilizing already existing email communication facilities. The basic idea is to simply provide adaptable guide-lines and enrich emails with information and recommendations to assist people in their daily work. To achieve its goal the Commius system, once installed, connects to existing email communication facilities automatically to intercept and analyze incoming emails on-the-fly. The system itself comprises a three-layer (interoperability, semantic and process interoperability layer) architecture to provide the necessary functionality and can be further decomposed into a build-time (customization) and a run-time part. So-called *business modules* represents one kind of incoming email, i.e. the necessary keywords to map an email to this particular module as well as the additional information and advice on further proceedings that should be included in the enriched version of the email.
GroupProcess (Huth and Nastansky 2000)	This approach tries to enable effective management of ad-hoc business processes by offering different approaches to generate ad-hoc process support. However, GroupProcess does not aim at replacing existing workflow solutions, but simply tries to extend their scope by adding adjusted techniques. The approach challenges two current ideas of workflow technologies: the distinction between *build-* and *run-time* and *workflow-model* and *-instance*. By merging build- and run-time, just-in-time modeling is intended and therewith dynamic (re-)modeling of ad-hoc processes is allowed. Furthermore, the approach suggest the integration of model and instance into a single object which could then easily be changed on-the-fly or even re-used as a process template in case of a recurring ad-hoc workflow.
Connector-oriented WfMS (Han and Shim 2000)	This approach proposes the implementation of a connector oriented workflow system featuring a connector facility that represents a department's or user's workflow in-box. Due to this connector facility, structured and un-structured processes can be connected (i.e. linking automated and manual execution). A further benefit of this incremental workflow execution is the possibility to gain re-usable process templates by performing process mining. This connector oriented approach allows defining a workflow completely at build-time or parts of it can be defined at run-time. Moreover, the definition can be performed either *activity-* (one defines the flow of activities towards a certain goal) or *actor-based* (a set of actors who are entitled to work on a certain case).
Pockets of Flexibility (Sadiq et al.	Pockets of Flexibility presents a concept to model semi-structured, flexible business processes without losing readability and simplicity of the models themselves or the workflow's adaptability capabilities. The basic idea of this modeling approach is to provide a

Technology	Characteristic
2001)	process model that comprise identifiable (pre-defined) workflow activities and pockets of flexibility within the process with an associated set of workflow fragments (i.e. a single activity or a sub-process) and a special workflow activity called build activity that provides the rules for concretizing the pocket with a valid composition of workflow fragments. Introducing these pockets of flexibility allows for compensating the inability to specify certain workflow fragments at build-time. A pocket is depicted as the above mentioned build-activity containing a (extendable) choice of possible activities.
OPENFlow (Wheater et al. 2000)	OPENFlow is a transactional workflow system implemented as a set of CORBA services. Its primary goals are the support of scalable systems, a flexible composition and reconfiguration of tasks and dependability. Tasks in a given workflow schema are represented as implemented activities comprising an array of *input sets* and *output sets*. This task structure offers a wide range of flexible workflow features. First of all, the task structure described above is an execution flexibility feature in itself as task implementations are chosen by the triggered input set or are completely determined at run-time (genesis tasks). Moreover, the system allows for dynamic changes to stored templates and runtime re-configuration. Possible re-configuration procedures comprise (among others) changing the implementation bound to a task, add/remove tasks from a workflow instance, change the constituent tasks of a compound task or alter the task structure associated with a genesis task.
ConDec / DELCARE (Pesic et al. 2007b; Pesic and van der Aalst 2006)	Traditional workflow modeling languages often force designers to over-specify process models, e.g. by using choice constructs. It is just not possible to simply state that two activities should never occur together. The presented *declarative* approach introduces a new kind of flexible process design by providing models that specify what has to be done, without detailing the how. Relations between activities in a model are not control-flow descriptions as in imperative models, but represent constraints. Two examples of this concept are ConDec, a declarative, constraint-based modeling language, and DECLARE (Pesic et al. 2007a), a workflow management system on a declarative basis.
Flexible Process Graph (FPG) (Polyvyanyy and Weske 2008)	This modeling approach aims at representing large collections of process instances with one single model as well as providing a formal approach for the definition of ad-hoc control flow. Such models can also be executed following formal execution semantics and provide limited late binding and some kind of adaptability. The basic idea of the modeling concept is to generalize process models, from the traditional directed graphs to using *hypergraphs*. The generated models just provide a guideline on what has to be done at which specific process state, but do not force the executing entities to follow a specific path. As a result, they can rather adapt their actions to the current business situation; consequently, FPG provides highly flexible process models and their execution.
Case Handling (Weske and Grünbauer 2005)	Case Handling is a completely new paradigm to handle knowledge-intensive workflows. In contrast to traditional WfMSs, the focus in Case Handling systems is not on what should be done to achieve a business goal, but what can be done. The worker should decide on how to reach the given goal considering a specific case state. The system should be an assistant and not a guide. As a result, this paradigm focuses on the case as a whole and provides its actors all the information available on a certain case. The Case Handling paradigm has been implemented into the FLOWer Case Handling system which is a part of the product BPM|one! (Athena 2008).
Ontology-based workflow handling (Almeida and Casanova 2004)	Workflow execution is often stopped or at least delayed due to unavailable information or resources. This approach tries to enable WfMSs to deal with such situations automatically by basically comprising two handling mechanisms: A mechanism to handle presuppositions that enables a process execution to continue in spite of incomplete information. A mechanism for choosing alternatives for sub-workflows, users or resources if they are unavailable or defined abstractly (applies only to sub-workflows). The choices made by those mechanisms are based on the workflow the user submitted, a workflow ontology and a set of semantic rules. The only weakness of this approach is the missing direct user interaction. Users can just rely upon the correctness of the provided ontology and the semantic derivation mechanism to gain adaptable and flexible workflows.
Extended workflow management through antici-	This approach introduces the concept of anticipation to render workflow execution, especially within team based scenarios, more flexible. It still relies on the well-known process models (directed graphs) but proposes an evolution to the workflow engine itself by enhancing the way a model is interpreted. When utilizing anticipation a workflow system allows users to

Technology	Characteristic
pation (Grigori et al. 2001)	start activities within a process model based on intermediate results provided by their predecessors. To actually support anticipation a workflow engine would have to be extended two new *activity states* (ready to anticipate and anticipating), a few basic *anticipation strategies* and *data flow handling intermediate results*. The described concepts are implemented within the MOTU prototype whose goal is to provide a framework to support cooperative work of virtual teams.
Activity Theory and Worklets (Adams et al. 2003)	Activity Theory is a research area with the intention of understanding human work practice. The concept of worklets is based on Activity Theory and thus aims at supporting people in a more natural way by supporting their way of performing tasks. Following the principles of Activity Theory, a process model is seen as a basic guideline rather than a specific prescription on how to reach the goal of a process. A worklet is defined as a small, self-contained, complete workflow process handling one specific task. The actual selection of the suitable worklet is based on three types of contextual information: generic information, case dependent information with a-priori knowledge and case dependent information without a-priori knowledge. A first implementation is provided as a YAWL Custom Service, collaborating with the YAWL workflow execution engine (Adams et al. 2006).
Taskmodeling (Eichholz et al. 2005)	In contrast to many other approaches already described above, this approach proposes a view of business processes represented by task models. Other than workflow management, which supports work on the organizational and group level, task models rather consider the level of an actor's individual work. The flexible execution and adaption is handled according to the metaphor *order and supply*. Hence, by focusing on the more individual aspect of work, this concept allows for flexible workflow execution. This is accomplished by delegation of tasks and choice of options as well as dynamic adaption of given structures by refining them to fit one's own individual structure, on the instance level.
TIBCO iProcess Suite (TIBCO Software Inc. 2007)	TIBCO iProcess Suite has been designed to deliver a complete Business Process Management solution from planning to execution and monitoring. Besides a wide range of BPM features, it also delivers a set of tools to support process adaptability. The iProcess Suite includes a goal-driven way of composing business processes from re-usable fragments. Once such a goal changes, users are allowed to change the process structure during run-time to adapt the process to its new goal. Moreover, the suite offers mechanisms to handle both expected and unexpected exceptional circumstances. In case of an unexpected exception, the system offers a suspension mechanism to halt process execution allowing for a controlled exception handling. Once an exception has been handled, it may be necessary to alter the process' state.
@enterprise (Groiss Informatics GmbH 2009)	@enterprise is a workflow system based on internet technologies. It supports organizations in the definition, execution and monitoring of business processes. Besides its basic process handling capabilities @enterprise also offers a range of semi-structured workflow techniques. At build-time, actors and alternative process paths can be defined with certain conditions regarding process run-time data. Furthermore, users are enabled to actively alter the process execution path during run-time. @enterprise supports simple changes like step-back or change-actor as well as more complex ad-hoc modifications. These modifications include insertion and skipping of steps and the choice of different execution paths.
Endeavors (Kammer et al. 2000)	Endeavors describes itself as an open, distributed, extensible workflow support environment which improves coordination and management by allowing flexible definition, modeling, and execution of workflow applications. The system is based on an object-oriented storage of business related objects, like *artifacts*, and so-called *activity networks* which associate activities by control flow, data-flow and resource-flow relationships. Activity nets may be changed dynamically during run-time or even used as reference templates. Modifications can either be kept on the instance level, or applied globally to provide incrementally evolving processes. All changes can also be performed in an on-the-fly manner during run-time.
ADAPT (Reichert et al. 2003)	The ADEPT workflow management system is an adaptive-workflow system with the goal to deal with high requirements regarding functionality, flexibility and scalability by delivering a wide range of features: a workflow modeling framework, the usage of temporal constraints, tools to define and coordinate dependencies between workflows, $ADEPT_{distribution}$ to handle distributed workflows on multiple ADEPT servers and mechanisms to handle evolutionary changes on the workflow type level. Furthermore, there is a powerful adaptive workflow feature called $ADEPT_{flex}$ which represents a framework to provide a user-friendly way of

Technology	Characteristic
	performing structural changes on running workflow instances; thus, enabling flexible execution and the ability to adapt to exceptional circumstances.
AgentWork (Müller et al. 2004)	The AgentWork approach aims at presenting a workflow system supporting automated workflow adaptations in a comprehensive way. A first implementation combines the powerful ADEPT workflow definition and execution capabilities with rule-based agent components responsible for the automated adaptations. The basic idea was to facilitate necessary adaptations to running workflow instances in time and with minimal user interaction by providing a WfMS capable of autonomously making the right decisions. Thus, a three-layer architecture (*workflow definition and execution, communication* and *agent layer)* is compromised.
WebFlow (Greiner and Rahm 2003)	The WebFlow project's goal is to provide a web-based workflow management system specialized in handling distributed cooperative workflows. It enables the definition, execution and monitoring of workflows including execution constraints for each integrated service. These execution constraints can be of a temporal or logical nature and allow for a more detailed monitoring and recognition of exceptions. Once an exception has been recognized, it can be handled (semi-) automatically with a focus on advancing the process state, in spite of the exception, or dynamic adaption, if the designed workflow language is based on ADEPT$_{flex}$.

4 Framework comparison and conclusion

To make the presented technologies in the preceding section comparable, they are classified by the framework that has been derived in the second section. An overview of this classification can be seen in following table:

Table 3: Overview of all technology evaluations

Technology / Criterion	Implicit Flexibility	Degree of Freedom	Late Modeling	Late Binding	Support	Adaptability
Caramba	⊕	⊕	⊕	⊙	⊙	⊙
CTM	⊕	⊕	⊕	○	⊙	○
Commius	⊕	⊕	⊕	○	⊙	⊕
GroupProcess	⊕	⊕	⊕	⊙	⊙	○
Connector WfMS	⊙	⊙	⊙	○	⊕	⊙
Pockets of Flexibility	⊙	⊙	⊙	○	⊙	⊙
OpenFlow	⊙	⊙	○	⊙	⊕	⊕
ConDec/DECLARE	⊙	⊙	○	⊕	⊙	⊕
FPG	⊙	⊙	○	⊙	⊙	○
Case Handling	⊙	⊙	○	⊕	⊙	○
Workflow Ontology	⊙	⊙	○	⊕	⊕	○
Anticipation	⊙	⊙	○	⊙	⊕	○
Activit Theory/Worklets	⊙	⊙	○	⊙	⊙	⊕
Taskmodling	⊙	⊙	⊙	⊕	⊙	○
TIBCO iProcess Suite	⊙	○	⊙	○	⊕	⊙
@enterprise	⊙	○	○	○	⊕	⊙
Endeavors	⊙	○	○	○	⊕	⊕
ADAPT	⊙	○	○	○	⊕	⊕
Agent Work	⊙	○	○	○	⊕	⊙
Web Flow	⊙	○	○	○	⊕	⊙

Key: ○ no support, ⊙ medium support, ⊕ full support

This overview clearly shows a polarization of processes with high support and those with a high implicit flexibility. If a technology offers a large set of user support features, this is always accomplished by relying on pre-designed process models. In those cases flexibility can only be provided by more or less dynamic change mechanisms and not by dynamic creation of process models during runtime.

Another extreme point of view is taken by the ad-hoc process support. They naturally have to overcome a lack of support as users might not always know which course of action to choose. However, there are some interesting approaches to overcome this lack of support, for example, the module suggestions provided by the Commius system or the best-practice style task patterns in the CTM framework. A third kind of supporting technique follows the idea of providing guidelines on how to complete a task without defining the "how" (cf. the Case Handling, FPG and Worklet approach).

As the technology reviews show, there is a wide range of different focuses among the systems; some like Caramba or Commius aim at supporting creative or fast-changing processes in smaller enterprises, whereas, e.g. the iProcess Suite or @enterprise offer a full spectrum of business process management features for larger companies. According to these observations, one important conclusion is the fact that there is no "perfect" approach for flexible processes. The only system which is close to offering support for the full spectrum of processes is Caramba as it enables the enactment of all kinds of processes. However, it still lacks the elaborate BPM features that might be desirable in large enterprises. To improve the support in a business it is necessary to carefully analyze the nature of the enacted processes and the kind of work that has to be supported. With the right amount of preparation businesses can already achieve many advantages through the presented approaches as there are technologies suitable for many distinct areas of work.

Nevertheless, future researches have to be focused on combining different approaches to gain a more widely applicable set of technologies. It may, for instance, be beneficial to integrate the email enrichment and tracking functionalities of Commius into Caramba to enhance its process handling with additional information and support capabilities. This combination could be further enhanced by including the worklet approach with its selection rules and repository. The advice on further proceedings provided by Commius could then be improved by providing suggestions on how to complete a certain task based on the gathered business data. If all these features could be integrated in a seamless and easy-to-use way, such a system might be able to yield flexible process support in many business scenarios.

References

Adamides ED, Stamboulis Y, Pomonis N (2005) Modularity and strategic flexibility: a cognitive and dynamic perspective. In: Systems dynamics society conference 2005.

Adams M, Edmond D, Hofstede AHMt (2003) The application of activity theory to dynamic workflow adaptation issues. In: 7th Pacific Asia conference on information systems.

Adams M, Hofstede AHMt, Edmond D, van der Aalst WMP (2006) Worklets: a service-oriented implementation of dynamic flexibility in worklets. In: On the move to meaningful internet systems 2006: CoopIS, DOA, GADA, and ODBASE. Vol. 4275. Springer, Berlin:291-308.

Adams M, Hofstede AHMt, Edmond D, van der Aalst WMP (2005) Facilitating flexibility and dynamic exception handling in workflows through worklets. In: 17th International Conference on Advanced Information Systems Engineering (CAiSE'05) Forum, Porto.

Almeida TSCV, Casanova MA (2004): Flexible workflow execution through an ontology-based approach.

Athena P (2008) History of Pallas Athena. http://www.pallas-athena.com/content/our-history. Retrieved 2009-09-02.

Burkhart T, Werth D, Loos P (2008): An email enabled collaboration tool for SMEs. – URL http://www.commius.eu/index.php?option=com_remository&Itemid=2&func=startdown&id=46.

Dellen B, Maurer F, Pews G (1997) Knowledge-based techniques to increase the flexibility of workflow management. Data & Knowledge Engineering 23(3):269-295.

Dustdar S (2004) Caramba - a process-aware collaboration system supporting ad hoc and collaborative processes in virtual teams. Distributed and parallel databases 15(1):45-66.

Eichholz C, Dittmar A, Forbig P (2005) Using task modelling concepts for achieving adaptive workflows. In: Engineering human computer interaction and interactive systems. Vol. 3425. Springer, Berlin:96-111.

Gebauer J, Schober F (2005) Information system flexibility and the cost efficiency of business processes. Journal of the association for information systems.

Greiner U, Rahm E (2003) WEBFLOW: Ein System zur flexiblen Ausführung webbasierter, kooperativer Workflows. In: 10. Fachtagung Datenbanksysteme für Business, Technologie und Web.

Grigori D, Charoy F, Godart C (2001) Anticipation to enhance flexibility of workflow execution. In: Database and expert systems applications. Vol. 2113. Springer, Berlin:264-273.

Groiss Informatics GmbH (2009) @enterprise web-site. http://www.groiss.com. Retrieved 2009-09-02.

Han D, Shim J (2000) Connector oriented workflow system for the support of structured ad hoc workflow. In: Proceedings of the 33rd Hawaii international conference on system sciences. Vol. 6 : IEEE Computer Society.

Herbst J (2004): Ein induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen. Berlin : Tenea Verlag.

Huth C, Nastansky L (2000): GroupProcess: Partizipatives, verteiltes Design und simultane Ausführung von Ad hoc Geschäftsprozessen.

Kammer PJ, Bolcer GA, Taylor RN, Hitomi AS, Bergman M (2000) Techniques for supporting dynamic and adaptive workflow. Computer supported cooperative work (CSCW) 9(3-4):269-292.

Müller R, Greiner U, Rahm E (2004) AGENT WORK: a workflow system supporting rule-based workflow adaption. Data & Knowledge Engineering 51(2):223-256.

Pesic M, Schonenberg H, van der Aalst WMP (2007a) DECLARE: full support for loosely-structured processes. In: Enterprise distributed object computing conference.

Pesic M, Schonenberg MH, Sidorova N, van der Aalst WMP (2007b) Constraint-based workflow models: change made easy. In: On the move to meaningful internet systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. Springer, Berlin:77-94.

Pesic M, van der Aalst WMP (2006) A declarative approach for flexible business processes management. In: Business process management workshops. Vol. 4103. Springer, Berlin:169-180.

Polyvyanyy A, Weske M (2008) Flexible process graph: a prologue. In: On the move to meaningful internet systems: OTM 2008. Vol. 5331. Springer, Berlin:427-435.

Regev G, Soffer P, Schmidt R (2006) Taxonomy of flexibility in business processes. In: Proceedings of the 7th workshop on business process modeling, development and support (BPMSD'06).

Regev G, Wegmann A (2005) A regulation-based view on business process and supporting system flexibility. In: Proceedings of the CAiSE'05 workshops.

Reichert M, Rinderle S, Dadam P (2003) ADEPT Workflow management system flexible support for enterprise-wide business processes - tool presentation. In: Proceedings of the 1st international conference on business process management (BPM'03), Eidhoven.

Reijers H, Rigter J, van der Aalst WMP (2003) The case handling case. International journal of cooperative information systems(12):365-391.

Rosemann M, Recker J (2006) Context-aware process design: exploring the extrinsic drivers for process flexibility. In: 18th International conference on advanced information systems engineering.

Sadiq SW, Sadiq W, Orlowska ME (2001) Pockets of flexibility in workflow specification. In: Proceedings of the 20th international conference on conceptual modeling: Conceptual modeling. Vol. 2224 : Springer.

Schwarz S, Abecker A, Maus H, Sintek M (2001) Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse. In: WM'2001 - Workshop "Geschäftsprozessorientiertes Wissensmanagement", Baden-Baden.

Soffer P (2005) On the notion of flexibility in business processes. In: Proceedings of the CAiSE'05 Workshops.

Stoitsev T, Scheidl S, Spahn M (2007) A framework for light-weight composition and management of ad-hoc business processes. In: Task models and diagrams for user interface design. Vol. 4849. Springer, Berlin:213-226.

TIBCO Software Inc. (2007) TIBCO iProcess Conductor. http://www.tibco.com/resources/software/bpm/iprocess_conductor_ds.pdf. Retrieved 2009-09-02.

Wargitsch C (1998): Ein Beitrag zur Integration von Workflow- und Wissensmanagement unter besonderer Berücksichtigung komplexer Geschäftsprozesse.

Weske M, Grünbauer D (2005) Case handling: a new paradigm for business process support. Data and knowledge engineering 53(2):129-162.

Wheater SM, Shrivastava SK, Ranno F (2000) OPENflow: a CORBA based transactional workflow system. In: Advances in distributed systems. Vol. 1752. Springer, Berlin:354-374.