Doctoral School on Engineering Sciences
Università Politecnica delle Marche

Extended summary

Innovative method and tool to manage workflow exceptions using process knowledge

Curriculum: Ingegneria Meccanica e Gestionale

Author

Marco Mandolini

Tutor

Prof. Ferruccio Mandorli

Date: 27-January-2011

Abstract. In the last decades, manufacturing industry has experienced important upsets and changeovers as a consequence of the rapid evolution of the global market. Competitiveness and dynamicity demand data integration and effective collaborative environments in order to support interactions among partners, data exchange, unpredicted events detection and solution.

Currently, workflow technology is considered as essential to integrate distributed and heterogeneous applications and information systems to improve business processes effectiveness and productivity. WfMSs (Workflow Management Systems) are used to support the modelling, analysis and enactment of structured business processes. Current systems are unfortunately too rigid and unable to provide adequate answers to exceptions or deviations that differ from the modelled process. It has been demonstrated that such deviations are really common in almost all processes: their handling appears relevant for improving inter and intra business processes.

The research goals could be sensitized as the definition of a new method and software tool for the workflow exceptions management using process knowledge. This thesis is a step toward the development of a novel WfMS enabling the dynamic business processes according to the networked model representing the EE (Extended Enterprise), to the identified process target and to the implemented workflow rules. The developed SW module encompasses the management of dynamic workflow evolution and deviations from the prescribed process model at runtime. In order to implement the abovementioned functionalities, the system extracts information from a knowledge base that increases as the work-

flow moves forward and extends all over the involved chain. Workflow exceptions are handled during process automation thanks to the implementation of artificial intelligence techniques, based on Case Based Reasoning (CBR) algorithms. They analyzes the exceptions domain with the aim to extract the most similar case, compared to the current one, identifying the best solving actions. The system, then, adapts the workflow instance to the retrieved solution and restores the natural process.

Thesis proposes also some case studies relative to industrial and telecommunication environments. Has been tested some processes of an extended enterprise, in order to evaluate advantages of the proposed method and software tools, as well as its limitations.

Keywords. Case-Based Reasoning Algorithms, Dynamic Workflow, Exceptions, JECA Rules, Workflow Management Systems.

1 Problem statement and objectives

The management of design phases of very complex products require more than one actor and more than one company, that are linked inside the Extended Enterprise. In order to manage interactions among all actors of the EE, the analysis and formalization of the Collaborative Development Processes is necessary. While in the past lots of attention was based on the study of new production technologies to reduce time to market and final product price, nowadays the principal researches to satisfy these goals are based on the business process management and optimization. The business process refers to process that coordinates organization and business policy in order to achieve some company goals.

In recent years, workflow management has become a very popular technology for supporting business processes, in which documents, information or tasks are passed between participants according to a defined set of rules to achieve an overall business goal [1]. However, conventional workflow management systems do not provide sufficient flexibility to cope with the broad range of failures that may occur during workflow execution. In particular, not only system failures such as hardware or software crashes need to be dealt with but also logical failures or exceptions. Rather than being considered an error, an exception in a business process is simply a deviation from the expected control flow.

Since fifteen years ago, much researches have been carried out on issues related to workflow exception handling in business process management systems. Such researches were initiated because, generally, commercial workflow management systems provide only basic support for exceptions handling [2], [3]; they typically require the model to be fully defined before it can be instantiated and changes must be incorporated by modifying the model statically. Further, there is minimal support for handling workitem failures and even when that support is offered, they must be manually terminated [4], [5], [6], [7]. Parallel to commercial solutions, a number of academic prototypes have been developed in the last decade. Only few of them have had an impact on the offerings of commercial systems [8], [9], [10], [11], [12], [13], [14], [15], [16].

The latest challenge in business process management is the development of systems able to manage all kinds of workflow exceptions, connected or integrated with tools for process knowledge management. The topic of business workflow exceptions management is tightly connected with process knowledge management, because solutions to problems could be formulated analyzing historical data and information. Knowledge management and process automation even if linked, are often seen as unrelated disciplines, because there are two vastly different perspectives for business processes.

The research goals could be sensitized as the *definition of a new method and software tool for the workflow exceptions management using process knowledge*. To realize these goals the business workflow exceptions should be defined, as well as process knowledge and formalization methods will be discussed or algorithms for exception resolution. This thesis is a step toward the development of a novel WfMS enabling the dynamic business processes according to the networked model representing the EE, to the identified process target and to the implemented workflow rules. The developed SW module encompasses the management of dynamic workflow evolution and deviations from the prescribed process model at runtime. In order to implement the abovementioned functionalities, the system extracts information from a knowledge base that increases as the workflow moves forward and extends all over the involved chain. Workflow exceptions are handled during process automation thanks to

the implementation of artificial intelligence techniques, based on Case Based Reasoning (CBR) algorithms. They allow analyzing the exception domain, extracting the most similar case compared to the current one and identifying the best solving actions. The system then adapts the workflow instance to the retrieved solution and restores the process.

2 Research planning and activities

Today, the research go toward the development of knowledge-based solution for workflow exception handling. The analysis of technical and scientific researches on WfMS allows setting the functional requirements for managing CPD through dynamic workflows:
The present research proposes a new approach to manage CPD by providing the following technical solutions to the above mentioned needs:

- creating a WfMS with a modular architecture, where each module is independent from each other. The exceptions handler can actually be integrated within commercial WfMS. Project data, case studies and solving rules are retrieved from shared databases that act as a bridge with each company repository;
- combining JECA rules and CBR algorithms to apply a knowledge-based strategy. The system extends WfMS functionalities with JECA rules for expected and unexpected events recognition and with CBR to adapt the predefined workflow models to changing circumstances;
- adopting an adaptive computing level for solving actions execution. Solving actions are autonomously applied with the support of the knowledge-based system.

An adaptive approach allows WfMS to meet CPD goals, in order to support continuous process improvement, resulting in more manageable and more efficient business processes over time.

2.1 Dynamic Workflow Manager architecture

The idea of this thesis is to develop and add-in to commercial software, with the aim to overcome their limits regarding exception management. Working on the standard architecture proposed by WfMC, it allow the development of a general application, that should be applied to different commercial systems. Starting from reference architecture proposed by WfMC, we have defined an architecture for the system used for the exception handling; we'll call it Dynamic Workflow Manager (DWM) [17], a plug-in software connected to a commercial Dynamic Workflow Management System, using Interface 2 (Workflow Client Application Interface), see Figure 1.

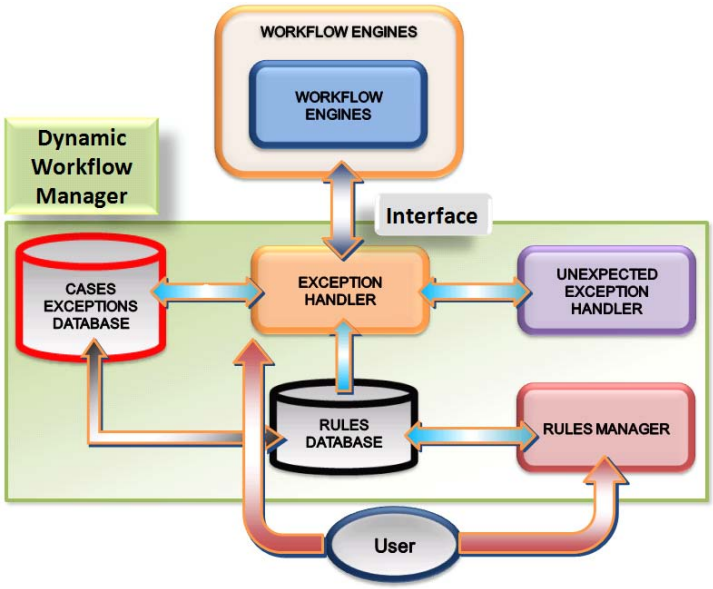

Figure 1. Dynamic Workflow Manager Architecture.

The aim of interface 2 is to communicate to the Dynamic Workflow Manager when an event is arose by WFMS as well as all information required to represent it and the context where is was gathered (current workflow and activity with all information necessary to represent them, as actor, date and time, due date, etc.). These data are then elaborated by DWM in order to determine the actions the WfMS has to perform in order to resolve current situation. The information passed from DWM to WfMS are, then, the action or set of these which should be executed (in this case as action we refer to further activity to perform within current workflow, the start of a new sub-workflow, to terminate current task and start a new one, etc.).

2.2 Unexpected Exception Handler

The Unexpected Exception Handler, Figure 2, instead, it's triggered when the exception handler recognize the event as unexpected. When this module is activated, the event was already defined, then the internal modules can start their tasks. The functions of the Unexpected Exception Handler are:

- Search inside the cases exception database historical cases, similar to current one;
- Provide a searching tool that could be used by user, in order to do customized researches;
- Calculate the solution for current case using proposed method, combining information stored inside rules and exceptions database;
- Provide editing tools to customize (revise activity of the CBR method) resolutive actions retrieved from database;
- Analyze the impact of the action to the current workflow, checking actions correctness and feasibility;
- Retain the solution inside the rules DB;
- Transmit information concerning the solution to the exception handler then to commercial workflow engine.

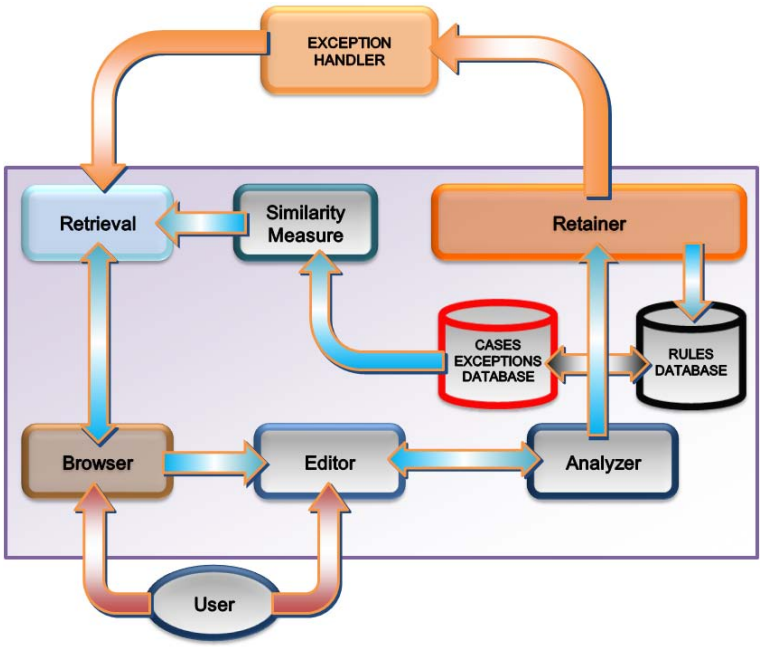

Figure 2. Unexpected Exception handler.

Taking into account the principal exceptions classification, that splits exceptions in two categories, expected (*useful*) and unexpected (*unanticipated*) and how knowledge is generally classified (*tacit* and *explicit*), our idea was to store process knowledge inside two different databases, connected through specific relation. The rule database contains the explicit knowledge used to solve those exceptions classified as expected, where there is a specific rule that defines actions to perform in order to solve the abnormal situation. Whenever arise an unexpected exception (this means that the exception handler didn't find a resolutive rule for current situation), solution should be calculated considering the tacit knowledge combined with explicit one. Even if information contained within both databases are different, there is a link between them in order to indicate the rule used to solve an historical case and vice versa what are the cases solved with a rule.

2.3 Workflow exception management

Whenever, during the workflow instance running, an event is caught, the workflow engine will transmit this one to DWM. Before the exception handler module performs any operations specific of exception management, it has to define what kind of event it should manage. The system can address four events types, *No exception*, *Expected Exception*, *Expected Exception with contradictory rules* and *Unexpected Exception*

- *No exception*: it means that the event is a normal situation and no actions should be chosen to restore current workflow;
- *Expected Exception*: it means that the exception is known, detectable and resolvable because a similar situation was already solved in the past;
- *Expected Exception with contradictory rules*: it means that the exception is known, detectable and resolvable, but more than one resolutive actions could be taken and sometimes they could be contradictory;

– *Unexpected Exception*: it means that the exception could be unknown or even if it's known and detectable, it isn't resolvable.

Each type is defined considering the number of defaults rules and activity defined inside the JECA rule. In order to solve an exception, it is possible to split the problem in two parts: if more than one rules have been retrieved from database, then the event could be solved directly by exception handler, otherwise, it is necessary to invoke the unexpected exception handler. This is a powerful tool able to combine tacit and explicit knowledge in order to define a solution for the current situation, analyzing how similar is current case to each other within cases exception database. The method used for exceptions solving is based on Case-Based Reasoning Algorithms, where the first step consists in the retrieve of similar cases. This phase is also constituted by two sub-steps: the first one for cases filtering, whereas the second one for cases sorting according to similarity between present and historical cases. Sorting phase is done according to specific algorithms used to evaluate similarity between present case and retrieved one. One algorithm used to measure similarity is the "*nearest neighbour matching*"

$$SIM\ (C_I, C_R) = \frac{\sum_{i=1}^{n} w_i \cdot sim(f_i^I, f_i^R)}{\sum_{i=1}^{n} w_i} \quad (1)$$

where C_I is the input case (current case), C_R the retrieved case (historical case) w_i is the importance weighting of a feature, *sim* is the similarity function, and f^I and f^R are the values for feature *i* in the input and retrieved cases respectively

When similar exceptions have been retrieved and sorted, the "*Retrieve*" task of CBR method is completed. Now, the "*Reuse*" and "*Revise*" phases can start and they consist in the editing of rule connected to most similar case (or another one manually identified by user). This is the phase where the human experience (*tacit knowledge*) is formalized in concrete rules (*explicit knowledge*) using numbers and all required to represent reality. User, can define new rules using its knowledge as well as the system knowledge acquired during past exceptions resolution. Possible resolutive actions types defined by DWM are: "*Send email or notification*", "*Jump to Activity*" or *"Run sub-workflows*". Revise phase allow the evaluation of just defined rule, analyzing the impact of defined actions when principal workflow is resumed and it restarts. Finally, a rule is "*Retained*" within rules database.

2.4 Dynamic Workflow Manager implementation

Dynamic Workflow Manager is a Server application, deployed in the same server where WfMS is running, in order to be used by the enterprise workflow administrator. DWM has been developed to work in windows environment (Windows Server 2003). The DWM framework project is used to define data which are then used by each other projects and general function required for their management. This is a shared project, because each other project contains references to this one.

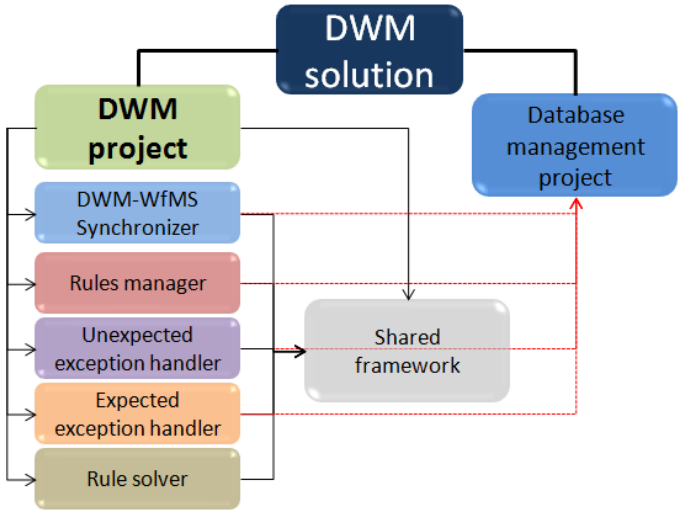

Figure 3. Dynamic Workflow Manager framework.

DWM has been integrated with a commercial WfMS, *SoftFlow* by *Metisoft*. SoftFlow works on the SharePoint platform which is a family of software products developed by Microsoft for collaboration, file sharing and web publishing. Integration between DWM and SoftFlow was done using SharePoint. When SoftFlow arises an event, it is written inside SharePoint lists, where columns represent the information required for event elaboration by DWM. Each information contained within SharePoint lists are the same of those ones defined inside cases exceptions and rules database, required to define an exception. When an event has been saved inside SharePoint list, SoftFlow will wait the answer from DWM, which, after reading event from SharePoint, it elaborates the solution that SoftFlow uses in order to continue workflow. Synchronization between Dynamic Workflow Manager and SoftFlow passes through SharePoint APIs, which are used to read and modify records of its lists.

The idea to choose a WfMS integrated in a platform such as SharePoint, allow the development of a real Collaborative Platform, where the Workflow Management System, with relative exception handler, represents a module of a greater architecture. This tool represents the most useful tool for Collaborative Product Development within an extended enterprises [18], [19].

3 Analysis and discussion of main results

The software tool developed in thesis work has been tested for several test cases related either to business (such as design, industrialization) and maintenance processes applied to industrial and telecommunications fields. Several processes has been implemented within an Extended Enterprise, related to COENV project (COllaborative ENVironments), cofunded by the Economic Development Italian Ministry (2007-2009).

3.1 Product development process test case

For brevity, it is presented only a test case, used to manage a product development process shared among companies of the extended enterprise. This is a long workflow, then any

exception could compromise its successful execution. This workflow manages the activities defined within a gantt diagram according to the project outcomes, and the actors will be the companies participating to the project.

The simplest unforeseen situation concerning this kind of workflow, as well as all workflows, it is represented by temporal exceptions, that is the activity finishing after its due date. Fro instance, an actor doesn't provide the documents representing activity results before the due date. In case of this situation, workflow can't continue as planned and following activities will shift. Solutions for this situation or similar ones are the followings: send a notify and email to PM (*Project Manager*) in order to define a new due date or start a subworkflow for a co-design session between PM and activity owner, in order to analyze problems which have determined this situation.

Another kind of exception is represented by documents coherence during their validation. In case of non-coherent document, followings rules can be defined: send a notify and email to document author, in order to communicate him which modifications are required to re-establish the coherence and jump to the activity where author has to upload the documents once again, modifying its due date, increasing previous due date with an accepted delay.

Also economic exceptions could arise during a workflow instance running. For instance, when an activity is completed if the deviation between final and budget cost is included inside 1 and 5%, then, a notify will sent to PM, whereas, if deviation is greater than 5%, then a collaborative session is required in order to discuss about problems which have determined this situation.

3.2 Results discussion

Test cases presented within thesis allow the achievement of advantages already presented with the aim to overcome limitations of commercial solutions. It's quite difficult for this kind of product to give a quantitative evaluation of benefits perceived by users due to the methods used to estimate advantages. For instance, the flexibility during workflow execution given integrating commercial WfMS with proposed DWM is a qualitative parameter, which should be evaluated providing a score according to a scale, through the comparison of the proposed solution to traditional one. How a system is able to manage workflow exceptions than another one, it is generally another qualitative parameter.

The most important functions which can give a validation to the proposed method and implemented system are the followings. "*Ability to retrieve useful information*" is used to evaluate how DWM is able to retrieve, from cases exceptions database, useful historical cases which solutions could be used to solve current situation. High score for this function means a good chance for user to define a solution for present case in few time and with a good impact on the workflow continuation. "*Adaptability to dynamic changes*", instead, represents the system ability to dynamic routing workflow instance, in order to face workflow exceptions.

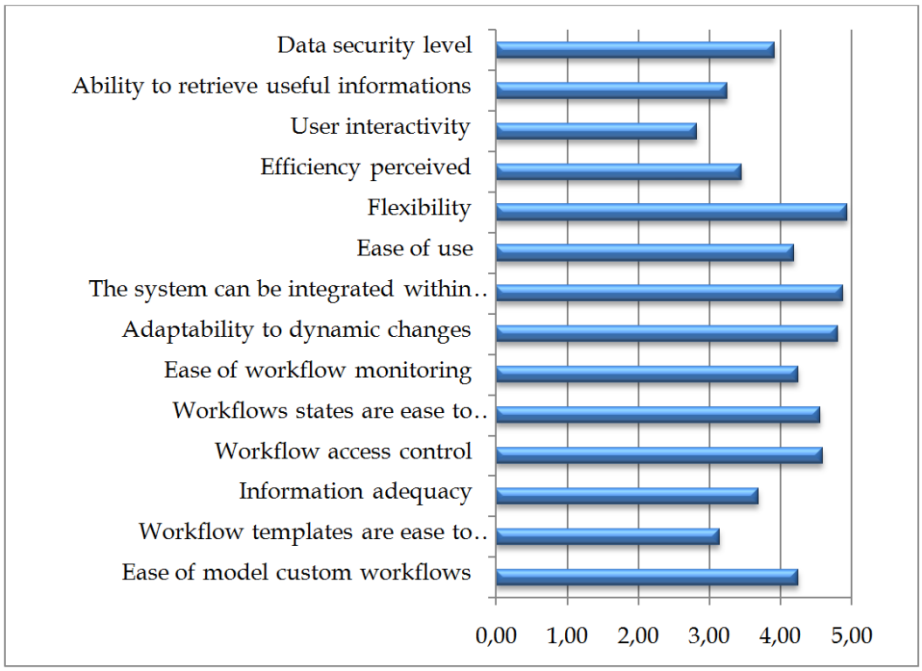

Figure 4. Evaluation of benefits perceived integrating the Dynamic Workflow Manager within collaborative platform.

4 Conclusions

This thesis proposes a new approach for process knowledge formalization and a new DWM module to implement process flexibility, adaptability and efficiency concepts. The proposed approach supports process knowledge formalization as it combines JECA rules and CBR algorithms functionalities to automatically handle workflow exceptions and configure new workflow paths according to cost optimization, timesaving and quality of interaction criteria. This thesis is a step forwards the definition of a dynamic WfMS architecture that will be integrated within a co-design web-portal, opened to all project partners. It can be observed that the dynamic exceptions may provide a more flexible organization to companies because problems may be semi-automatically overcome by the system itself.
The main scientific contributions can be synthesized as follows:

- Workflow adaptability without unexpected crash or loop cycles;
- Automation in the rapid configuration of process models according to the occurred event and the retrieved solving rule;
- Support to the system administrator in rapid decision-making based on previous adopted solutions;
- Automation of numerous operations and consequent reduction of time for handling exceptions to predefined workflow;
- Web-based platform for co-design enhances communication between enterprises at the level of intra and inter chains involving both suppliers, designers and the whole leader company organization;
- A new system architecture to dynamically manage workflows; it's able to formulate different solutions to overcome the unexpected exceptions, allowing the process manager to customize the solution in accord with specific company's needs;

– The proposed knowledge-based approach has been implemented into a commercial WfMS (SoftFlow by Metisoft) by developing a plug-in application for the exception handling;
– Preliminary experimentations have been carried out to support the development of a project within an extended enterprise, the production planning, maintenance processes, etc.

Experimentations, however, point out some system limitations that need to be overcome:

– The adopted similarity measurement techniques do not completely match with the ways the project manager assesses an unpredictable event during the design process and then retrieve the solution thanks to his/her personal experience;
– The techniques for the evaluation of the changes impact on running workflow have to be improved in order to better support decision-making during process reconfiguration;
– Future work will be focused on the complete fulfillment of the knowledge-based repository using additional real case studies in CPD;
– Integrate proposed DWM with other commercial Workflow Management Systems, in order to overcome limitations given by SoftFlow solution.

References

[1] Workflow Management Coalition WMC, "The workflow reference model. Document number TC00-1003, Issue 1.1," 1995.

[2] N. Russell, W. van der Aalst, and A. ter Hofstede, "Workflow exception patterns," in *Proceedings of the 18th International Conference on Advanced Information Systems Engineering (CAiSE 2006)*, Luxembourg, 2006, pp. 288-302.

[3] M. zur Muehlen, "Workflow-based Process Controlling. Foundation, Design, and Implementation of Workflow-driven Process Information Systems," *Advances in Information Systems and Management Science*, vol. 6, 2004.

[4] COSA. (2010) COSA BPM 5.7, Product Description. [Online]. http://www.cosa.nl/docs/EN/EN_COSA_BPM_5.7_ProdDescr.pdf

[5] IBM. (2010) WebSphere MQ Workflow. [Online]. http://www.redbooks.ibm.com/redbooks/pdfs/sg247282.pdf

[6] ORACLE. (2010) WebLogic Process Integrator Release 1.2.1. [Online]. http://download.oracle.com/docs/cd/E13219_01/wlpi/wlpi121/studio/ch7.htm

[7] TIBCO. (2010) TIBCO. [Online]. http://www.tibco.com/multimedia/wp-tibco-iprocess-suite_tcm8-786.pdf

[8] M. Reichert, S. Rinderle, U. Kreher, and P. Dadam, "Adaptive process management with ADEPT2," in *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, Tokyo, Japan, 2005, pp. 1113-1114.

[9] R. Muller, U. Greiner, and E. Rahm, "AgentWork: a workflow system supporting rule-based workflow adaptation," *Data & Knowledge Engineering*, vol. 51, no. 2, pp. 223-256, 2004.

[10] B. Weber, W. Wild, and R. Breu, "CBRFlow: Enabling adaptive workflow management through conversational case-based reasoning," in *Proceedings of the 7th European Conference for Adavances in Case Based Reason ing (ECCBR'04)*, vol. 3155, Madrid, Spain, 2004, pp. 434-448.

[11] S. Hwanga and J. Tangb, "Consulting past exceptions to facilitate workflow exception handling," *Decision support systems*, vol. 37, pp. 49-69.

[12] M. Adams, A.H.M. Ter Hofstede, D. Edmond, and W.M.P. Van Der Aalst, "Facilitating Flexibility and Dynamic Exception Handling in Workflows through Worklets," in *Proceedings 17th International Conference on Advanced Information Systems Engineering (CAiSE'05)*, Porto, Portugal, 2005, pp. 45-50.

[13] H. Mourao and P. Antunes, "Supporting effective unexpected exceptions handling in workflow management systems," in *Proceedings of the ACM symposium on Applied computing table of contents,* Seoul, Korea, 2007, pp. 1242-1249.

[14] Dovilè Vojevdina and Genadijus Kulvietis, "Business Process Unpredictable Exception Handling Using Workflow Based Method," in *Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION*, Corfu, Greece, 2005, pp. 627-632.

[15] Y.C. Ngeow, D. Chieng, A.K. Mustapha, and E. Goh, "Web-based Device Workflow Management Engine," in *International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, Seoul, Korea , 2007.

[16] Kou Weili, Gong Peng, Wang Jing, and Cai Kai, "Workflow exception handling system application in hierarchical modeling," in *Computer Supported Cooperative Work in Design, CSCWD 2008*, Xi'an, 2008, pp. 693 - 698.

[17] Marco Mandolini, Michele Germani, and Maura Mengoni, "A dynamic workflow management system to support cooperation in the extended enterprise," in *International Conference on Product Lifecycle Management*, 12-14 July 2010, Breme, Germany, 2010.

[18] Marco Mandolini, Michele Germani, Maura Mengoni, and Margherita Peruzzini, "Collaborative design system for supporting dynamic virtual enterprises," in *Proceedings of 11th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2010*, St. Etienne, France, 2010, pp. 577-585.

[19] M. Mandolini, M. Mengoni, M. Peruzzini, and S. Graziosi, "A Knowledge-based workflow to dinamically manage product innovation," in *Proceedings of the IDMME-Virtual Concept Conference*, Beijing, 2008.