A Meta-model for Dynamic Workflow Evolution

Berraouna Abdelkader[1], Amirat Abdelkrim[1] and Meslati Djamel[2]
[1]University Mohammed Chérif Messaadia, 41000, Souk Ahras, Algeria
[2]University Badji Mokhtar-Annaba, BP 12, 23000, Annaba, Algeria

Keywords: Meta-model, Workflow, Dynamic Evolution, Components.

Abstract Workflow systems such as they are today do not provide sufficient flexibility needed to deal with certain situations that may arise during the execution, especially in critical environments such as medical systems and banking systems. In this context, this paper treat the adaptive workflow based on Meta-Model approach and components-based software approach. These provide granularity, flexibility and reliability needed for effective and safe development of the workflow. The work is both in terms of the development process and in terms of the support platform, which should ensure the adaptability and propose a generic component-oriented framework to build, run and develop dynamic workflow templates. Our research objectives are: First, the proposal of a proper workflow development approach that takes into account the evolution of these systems appearance. Then the realization of a framework environment that takes advantage of the component approach and the meta-model approach and its features.

1 INTRODUCTION

Companies in different fields have always shown interest for information technology, and this interest is growing. Thus was born an undeniable dependence of computers in this area and it has become impossible for a company to survive without the support of adequate information technology.

One of these technologies is the "workflow" is used to describe the automatic task execution cycle time, validation methods, and to provide each player the information necessary for the execution its task. The workflow application fields are diverse as financial services, telecommunications, the public sector and even creative areas such as the service industry and the media. With changing needs in this wide range of areas, we have identified a gap in research in the adaptation and flexibility of workflow. (Kolar et al., 2013)

Given the level of competitiveness and the environment in which businesses currently it is vital to be able to change workflow systems either statically or dynamically. However, workflow systems such as they are today do not provide sufficient flexibility needed to deal with certain situations that may arise during the execution, especially in critical environments such as health care systems and banking systems. In this context, this paper deals with the adaptive workflow based on both approaches, the Meta-Model approach and the approach based software components (Reichert and Dadam, 1997) (Weske,1997) (López-Fernández et al,2015)(Tiwari and Chakraborty ,2015). These provide granularity, flexibility and reliability needed for effective and safe development of the workflow. The work is both in terms of the development process and in terms of the support platform, which should ensure the adaptability and propose a generic component-oriented framework to build, run and develop dynamic workflow templates. Our research objectives are:

First, we propose a workflow development approach that takes into account the evolution of these systems appearance. Then the realization of a framework environment that takes advantage of the component approach and the meta-model approach and its characteristics. The motivations of this work are many. In fact, dynamic evolve workflow systems will help increase competitive businesses, reduce costs has required the adaptation and evolution of these systems and possibly improve the working environment through active support, reliable and adapted to change as needed. Thus it will be possible for these companies to deal with different types of change and crisis situations by adapting to new contexts.

Abdelkader, B., Abdelkrim, A. and Djamel, M.
A Meta-model for Dynamic Workflow Evolution.
In *Proceedings of the 1st International Conference on Complex Information Systems (COMPLEXIS 2016)*, pages 59-65
ISBN: 978-989-758-181-6

The rest of the paper is organized as follows: section 2 presents some of basic concepts and terminology of the dynamic workflow. Section 3 outlines the similar work and highlights the motivation of this work. Section 4 presents a summary and discussion of existing approaches. Section 5 describes an introduction to the proposed framework and will end with a conclusion and some perspectives.

2 CONCEPTS AND TERMINOLOGY OF DYNAMIC WORKFLOW

Before presenting the existing approaches for the modeling of adaptive workflow we first present several related concepts.

-Workflow: A workflow is a software tool dedicated to the management tool processes. This tool define, manages and executes processes by implementing programs whose execution order is pre-defined in a computer representation of the logic of these procedures (Workflow Management Coalition – Glossary 1996).
- WorkFlow Engine: is a software service that provides all or part of the execution environment of a Workflow(Workflow Management Coalition – Glossary 1996).
-Workflow Management System: is a complete system that is used to define, manage and execute processes by implementing programs whose execution order is pre-defined in a computer representation of the logic of these procedures (Workflow Management Coalition – Glossary 1996).
- Workflow Service: is a software service that consists of one or more workflow engines of the same type used to define, manage and execute workflow process (Workflow Management Coalition – Glossary 1996).
- Dynamism: is characteristic related to evolution of the process model or to changes in the business environment or business process reengineering efforts. These changes are made in the time of conception and involve a significant problem (Sadiq et al., 2001) .In this characteristic the proposed solution should reduce the number of situations needs of change.
-Adaptability: Is the process capability to deal with exceptional cases and a non-standard behavior. This can be partially solved in time by adjusting the design (Adapter accurately) the structure of the process model (Kolar et al., 2013).
- Flexibility: is a characteristic of the process model related to loose or partially defined model structures specified in design-time. The full specification of the process is finalized at the time of execution and may be different for each process instance. Flexibility is the main objective of this work (Kolar et al,2013).

3 EXISTING APPROACHES FOR MODELING DYNAMIC WORKFLOW

An example of a commercial workflow that allows some dynamic adaptations is **Tibco iProcess Suite** (version 10.5), which provides an orchestration component "Orchestrator" which allows dynamic allocation of sub-process variations in the execution time. It requires a construction called dynamic event to be modeled explicitly contains a number of sub-processes listed as an "**Array**". When execution reaches the dynamic event node, it will execute the members of the «Array " on predefined conditional basis, the table must be set statically before the process is instantiated there is no possibility of modifications execution.

Another trading system, **CSR** (Version 5.4) allows an ad hoc manual execution adaptation such as (reorganization of tasks, cancellation, repetition, delay, termination).

The **ADEPT2 prototype** (Reichert et al ,2005) supports the change process at runtime (ie, add, delete and modify the sequence of tasks) to the model level (dynamic evolution) and at Instance (ad hoc changes). These modifications are made to a traditional homogeneous model and must be achieved through manual intervention of an administrator, summarized in a high-level interaction. The system is also responsible for the "jump forward" and "backward jumps" in the process instances, but only by the authorized person.

The **system YAWL** (Yet Another Workflow Language) provides support for flexibility and dynamic exception handling through the concept of worklets, extensible directory of autonomous sub-processes with associated selection rules this approach provides direct a dynamic change and development processes without intervention outside the system or stop.

(Minoret al ,2014) Present a Case-based reasoning (CBR) approach for automated workflow adaptation through the reuse of experience.

This approach helps the expert in the execution of these adaptations by an automated method. The method uses the recorded case workflow adaptation in adaptations of the past. The recorded changes can be automatically transferred to a new workflow that is in a similar situation changes, it uses CFCN the workflow modeling language that was developed by the University of Trier.

(Kolar et al., 2013) Proposes an approach based on the constraints in the design process. This combined approach is for business situations, including a rigid structure to the process model causes a flexible limitation.

It offers design process model that is suitable for the modeling of ad hoc processes within BPMS systems. The pattern may be used to define a process structure in the declarative constraint- based manner. In addition, it presents an application of the approach in a real project, a BPMproject end to end for an insurance company.

In (Tayade and Chavan, 2012), the authors have taken an approach to identify the change as an ongoing process, and integration of the processes of change in the workflow process itself which is predefined and can predefined and can be fixed in execution time. The framework presented introduced the concept of a flexible workflow process comprising a core and one or more types or roles of flexibility in this kernel; the following Figure shows briefly the basic structure of a workflow modeling language:

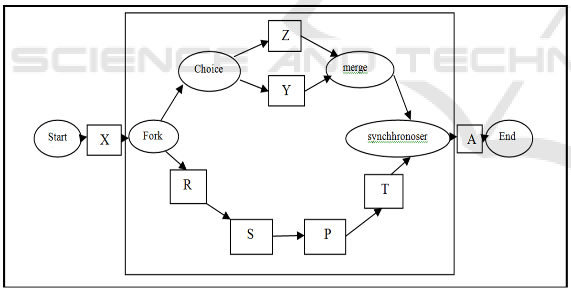

Figure 1: The structure of a workflow modeling language.

The system includes four components as shown in the following Figure :

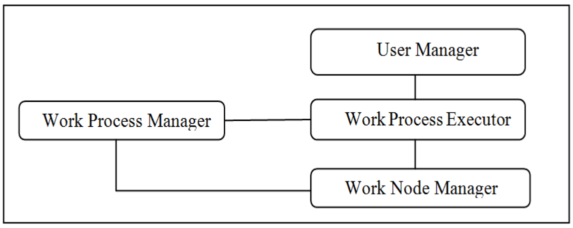

Figure 2: FWMS Architecture.

- **Work Node Manager**

This is the basic component of the work process all working nodes are stored in the nodes of the database with the appropriate classification.

- **Work Process Manager**

Manages the workflow templates and instances, including creating, editing, deleting and querying workflow.

- **User's Task Manager**

This component manages users there's including the verification of the user connections and disconnections.

- **Work Process Executor**

This component represents the system core, includes eight main components - runtime components, Service task, Repository Service, Management service, Service of history, identity service, DAOFactory core.

The **meta-model approach** is used (explicitly or implicitly) to identify structures and types of workflow model of constituent components. A set of primitives is typically defined in which change operations can be performed a certain model or bodies. Examples of this approach include ADEPTflex (Reichert and Dadam ,1997) WASA (Weske ,1997) and WIDE (Casati et al,1997). The syntactic correctness and modifications can be made by users.

A meta-model describes the structure of models and allows reasoning on models like the Knowledge of first Level.

Some object-oriented languages such as Java, describe the same way the classes, meta-classes and objects. In software engineering, it is with the concept of UML meta-model was developed. The meta-model describes the models: classes, objects, attributes, relationships and self-described, also means the model of the model. It can be defined as the representation of a particular view on model.

In the Components-based approach the modeling goal is the construction of more complex models, through reusing consistent and functional models. Each of them is a process component. When modeling components we should consider the following issues:

- Building an information base on reusable components
- Determine the function of each component in the new process model
- Dynamic composition of heterogeneous components.

4 SUMMARY AND DISCUSSION OF EXISTING APPROACHES

We studied several approaches proposed in the literature about workflow flexibility, we can mention some features on the dynamic evolution of workflow:

The Factors of Evolution in Workflow Management Systems (Hanet al., 1998)

- Change in the environment.
- Technical progress.

Ad-hoc Diversions During Workflow Execution (Hanet al ,1998)

- Refinement of dynamic workflow model.
- User intervention.
- An unpredictable event.
- System failure.

Classification of Adaptive Workflow Adaptations Levels (Hanet al ,1998)

- Adaptation to the context level domain.
- Adaptation in process Level
- Adaptation at Resource Level
- Adaptation to the infrastructure level.

Generally, there are two types of changes:

1 ad-hoc changes
2 Evolutionary change.

Ad-hoc changes are handled case by case. To provide customer-specific solutions or handle rare events, the process is suitable for one or more cases. Evolutionary change is often the result of reengineering efforts. The process is modified to improve customer responsiveness and improve efficiency.

Dynamic workflow refers to the extension of the workflow process so that when the change occurs, the process model may be modified or completed, rather than build a new model. The change can be considered (which concern only the current instance) or may need to be applied either temporarily or permanently. The adaptation is done on two levels.

- level process model
- level instances

In instances running must be managed when the process model from which it was instantiated it (the instance should be canceled, restarted) so the workflow must provide support for changes to the process model job but incremental and ad hoc changes can be arranged as required.

From several criteria identified for comparison, we present a subset of them: **the nature of flexibility, formalism and flexibility techniques.** According to (Workflow Reference Model, 1995) **WFMS** can be characterized as a support for three areas functional:

- **The Build-time Functions** involved in the definition and possibly modeling, workflow process and its constituent activities.
- **Run-time Functions** involved in the control of workflow processes in an operational environment and scheduling to handle various activities as part of each process.
- **Run-time Interactions** with human users and its implementation tools available for the various stages of activity.

The study of literature allowed us to distinguish two main types of flexibility based on the ability to cope with changes that may be incorporated into the process definitions when building or running time.

- Posterior Flexibility by adapting the process definition or one of its instances during execution.

The approaches that offer this kind of flexibility are based on prescriptive modeling formalisms. It could be considered that the definition of processes and the resulting are not really flexible, but rather to adapt or evolve. In fact, these approaches can not anticipate the ability to change at time of construction, it is the most common situation found in the literature.

Prescriptive modeling formalisms are well suited to specify business processes that require a high degree of control and prediction and for which the need for change is an exception such as production workflow.

- Flexibility Priori or by Selection based on modeling formalisms that can offer the ability to cope with environmental change without changing process definitions. This means that this capacity should be integrated into the process definitions during construction. The process definition should be specified in a sufficiently flexible to adapt with the environment without breaking execution. Consequently, the workflow service should be able to run the specification "incomplete" process definitions.

5 PROPOSED FRAMEWORK

We focused specifically on workflow modeling approaches based components and meta-model, the control and data integration are implemented from a model based components. Then, the process is seen as a set of inter operative components. Data dependencies are reduced by encapsulating the components behind well-defined interfaces. This allows dynamically changing workflow systems, and proposing a generic component-oriented framework to build, run and develop dynamic workflow templates.

The idea is to see the workflow as an open process and its implementation is not finished.

We used the OPC (Open Process Component) (Sadiq et al., 2001) mechanism to adopt implementation, this mechanism allows:

- Dynamic Evolution of Software Developed.
- Flexibility
- The development of a basic component
- Provides a meta-layer model for software
- Enables configuration management
- Provides meta-process support

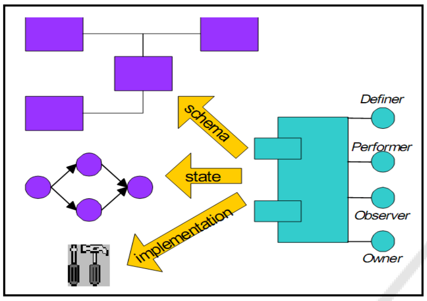

Figure 3: OPC Framework.

The OPC (Gary et al., 1997) provides an environment for production processes based components.

In this approach, the process models are constructed as a set of components that interact through interfaces well defined. It also offers an infrastructure for interoperability and reuse components of heterogeneous processes, built on three levels of abstraction.

1. The ***meta-model level*** that identifies the fundamental entities of the process and their relationships.
2. The ***finite-state machine*** use for representation of the behavior of the process entities
3. An ***object-oriented framework*** related to the process representation formalism and its area of application.

It provides support for dynamic processes, a component is defined with the model and the instantiation information and allows the component manage its own Evolution, a Component has generic information (in the framework) and more specialization (adaptation) in the instantiation.

The first step in the development of our framework is to design a meta-model (***meta-model level***) that can capture all the features and functionality of the various workflows in the ***trade business domain***

The proposed meta-model is shown in Figure 4In this meta-model the class ***Activity*** is the central part. A process contains several ***FlowNode*** that can be ***Avtivity Group***,***Simple Activity*** or ***Control Node***. The class ***FlowNode*** has a reflexive association called ***seq*** to define the sequence relationship.

An activity have tow author relation with ***Resource*** class and ***Actor*** class.

The class Activity has an attribute state assigned to the enumeration type ***State*** that indicates in which execution state the activity is currently, this enumeration has five stats (waiting, running , done ,skipped and failed).

To ensure the control of different activity we have the class ***Control*** that contains three type of control ***Join***, ***Fork*** and ***Sequence***.

The activities in which decisions have to be made are modeled explicitly with the class ***Decision***. The conditions for selecting the right path is modeled by class ***Guard*** which is connected to the subsequent activities. At runtime the user can select the desired Guard with the ***select()*** operation, the Decision activity can either be an XOR that selects exactly one path or an OR that can select more than one or AND that select the both.

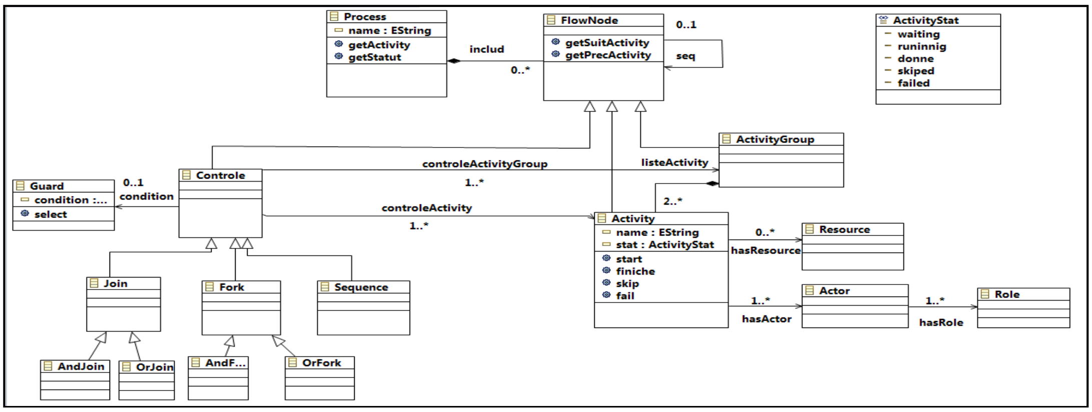

Figure 4: Workflow meta-model.

6 EXAMPLE PROCESS MODELING WITH

In this section the metamodel of the previous section is used to create workflow models. In the first subsection the workflow is described in natural language. In next subsection the described workflow is modeled in ***GMF tool*** with the metamodel elements introduced in this model Resources and data are not considered in the model because they are not part of the metamodel. Only activities and control flow aspects are modeled.

6.1 Process Description

The following example describes globally the workflow for making an order.
We identify three actors in this system, namely the customer, accounting service and delivery service.
The action takes place as follows:
The client begins by make an order, giving rise to the development of a ***Quote*** by the ***Accounting Service*** .making quotation activity itself is decomposed into two sub-avtivity check availability of the ordered products and calculate the order price.
Quote is the name of a class of system, data stream are typed the customer can then change its order (back to the initial activity to order), to cancel (go to final state) or validate the Quote.
If the costumer validate ***Quote*** two actions can be undertaken in parallel:

- the preparation of the order by the delivery service.
- invoice processing and payment.

The ***Accounting Service*** create invoices and send it to customer (the object invoices transmitted in the state sent) this makes the payment and sends the invoice (which is present in the regulated state).

When the order is ready and confirmed the payment of the client (by appointment synchronization of these two activities) the order is delivered to the client and the treatment is finished.

6.2 Workflow in GMF Tool

The proposed modeling of workflow is shown in figure 5

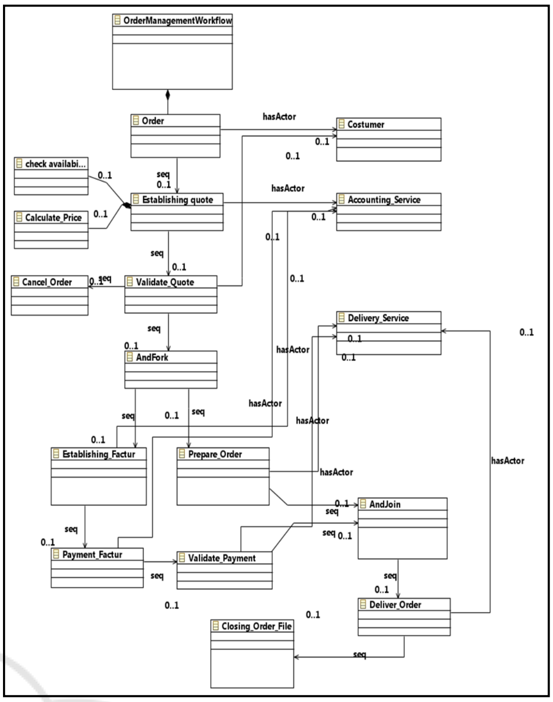

Figure 5: Workflow model.

7 CONCLUSIONS

Workflow management systems must provide high support flexibility with easy and simple ways. This change can manifest in different forms in the workflow process, in this article we presented different approaches for the modeling and the development of flexible and adaptive workflow with a comparison and synthesis.

We also presented research aims to develop a generic component-oriented framework to ensure the dynamic development prospects as workflow.
In further work we aim to develop and implement this framework to ensure the dynamic development workflow

REFERENCES

Kolar, J., Dockal, L., & Pitner, T. 2013" A Dynamic Approach to Process Design: A Pattern for Extending the Flexibility of Process Models. In *The Practice of Enterprise Modeling* "(pp. 176-190). Springer Berlin Heidelberg.

Reichert, M., Rinderle S., Kreher, U., Dadam P. 2005 "*Adaptive Process Management with ADEPT2*". International Conference on Data Engineering, Demo Session, Tokyo, Japan, p. 1113±1114.

Adams, M., ter Hofstede, A., Edmond, D., van der Aalst, W., 2006 *Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows*". International Conference on Cooperative Information Systems, Montpellier, France, pp.450-456.

Minor, M., Bergmann, R., & Görg, S., 2014"Case-based adaptation of workflows". *Information Systems*, 40, 142-152.

Tayade, S. M., & Chavan, V. 2012 "Roles of Flexibility in Workflow Engineering". *International Journal of Modeling and Optimization*, 2(5), 644-647.

Han, Y., Sheth, A., & Bussler, C, 1998." A taxonomy of adaptive workflow management". In *Workshop of the 1998 ACM Conference on Computer Supported Cooperative Work*. pp 150-156.

Workflow Reference Model 1995, WfMC-TC-1003 v1.1, V 1.1.

Sadiq, S., Sadiq, W., Orlowska, M. 2001 "Pockets of flexibility in workflow specification". In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. *LNCS*, vol. 2224, pp. 513–526. Springer, Heidelberg.

Reichert M., Dadam P.1997 "A Framework for Dynamic Changes in Workflow Management Systems" *Proceedings of 8th International Workshop on Database and Expert Systems Applications*, France, September, 42-48.

Weske M. 1998 "*Flexible Modeling and Execution of Workflow Activities*", Submitted to 31st Hawaii International Conference on System Sciences (HICSS-31)pp 201-207.

Casati F., Ceri S., Pernici B., Pozzi G. 1996 "Workflow Evolution", *Proceedings of 15th International Conference on Conceptual Modeling (ER'96),* Germany, October, pp.438-455.

Workflow Management Coalition – Glossary 1996. http://www.aiai.ed.ac.uk/project/wfmc/ARCHIVE /DOCS/glossary/glossary.html.

Workflow Reference Model ,1995, WfMC-TC-1003 v1.1, V 1. January.

Gary, K., Lindquist, T., and Koehnemann, H. 1997 "*Compo-nent-based Process Modeling*". Technical Report TR-97-022, Computer Science Department, Arizona StateUniversity. May.

López-Fernández, J. J., Cuadrado, J. S., Guerra, E., & de Lara, J. 2015. Example-driven meta-model development. *Software & Systems Modeling*, *14*(4), 1323-1347.

Tiwari, A., & Chakraborty, P. S. 2015. Software Component Quality Characteristics Model for Component Based Software Engineering. In: *Computational Intelligence & Communication Technology (CICT*), 2015 IEEE International Conference on pp. 47-51.