Proceedings of GLOGIFT 09
November 12 – 14, 2009
National Institute of Industrial Engineering
Mumbai, India

Different Scenarios of Flexibility in Workflow Technology

Prof. Shweta Tayade
Department of MCA,
G.H.Raisoni College of Engineering
Nagpur,India.
shwetatayade194@rediffmail.com

Abstract

In order to adequately support the execution of business processes in an increasingly turbulent world with shorter cycle times, workflow management systems must provide higher support for flexibility. Workflow technology is currently being deployed in quite diverse domains. However, the element of change is present in some degree and form in almost all domains. A workflow implementation that does not support the process of change will not benefit the organization in the long run. Change can be manifested in different forms in workflow processes. In this paper, we first present a categorization of workflow change characteristics and divide workflow processes into dynamic, adaptive and flexible processes. We define flexibility as the ability of the workflow process to execute on the basis of a loosely, or partially specified model, where the full specification of the model is made at runtime, and may be unique to each instance. To provide a modeling framework that offers true flexibility, we need to consider the factors, which influence the paths of (unique) instances together with the process definition.

Keywords-business; workflow; flexible

1. INTRODUCTION

Workflow management systems (WFMSs) provide useful mechanisms to support users in their enactment of daily business processes. In today's ever changing turbulent business reality BPS systems must provide a high degree of flexibility. Lock in to yesterday's process definitions must be avoided to exhibit continuous improvement and optimization.

We can find in the literature [11], several categories of workflow types, such as production, collaborative, ad-hoc etc. To determine suitability of workflow technology, process characteristics such as functional complexity, predictability and repetitiveness are often considered, especially in the general class of production workflows.

The necessity for the support of change in workflow systems is well recognized. Providing support for changing processes in workflow systems is, and has been for a few years, an active area of research [2], [3], [4], [15]. In the following sections, we will first provide a categorization of change characteristics in workflow processes, dividing them into dynamic, adaptive and flexible workflows.

This paper primarily deals with the last. In the subsequent sections, we will present a unique, generic and practical approach for the handling of flexible workflows. Our approach is based on the principle of recognizing change as an ongoing process, and integrating the process of change into the workflow process

itself. The framework presented in this paper introduces the concept of a flexible workflow comprising of a core process and one or more pockets of flexibility within the core process.

2. Components of Change in Workflows

We first introduce basic terminology for the sake of clarity. By a workflow Model we mean a definition of the tasks, ordering, data, resources, and other aspects of the process. Most, if not all, workflow models are represented as graphs which depict the flow of the process tasks, together with a description of other task properties. A workflow Instance is a particular occurrence of the process. An Instance Type is a set of instances that follow the same execution path within a given process model.

Dynamism

The first component represents dynamism - which is the ability of the workflow process to change when the business process evolves. This evolution may be slight as for process improvements, or drastic as for process innovation or process reengineering. In any case, the assumption is that the workflow processes have predefined models, and business process change, causes these models to be changed.

The biggest problem here is the handling of active workflow instances, which were initiated in the old model, but need to comply now with the new specification. The issue of compliance is rather a serious issue, since potentially thousands of active instances may be affected by a given process change. Achieving compliance for these affected instances may involve loss of work and therefore has to be carefully planned [18].

A typical example of dynamic workflows can be found in university admissions. Consider a scenario of a large tertiary institute that processes thousands of admission applications every year. The procedure for application, review and acceptance is generally well defined and understood. Suppose that the office of postgraduate studies revises the admission procedure for postgraduate students, requiring all applicants to submit a statement of purpose together with their application for admission. To implement this change, there can be two options available; one is to flush all existing applications, and apply the change to new applications only. Thus all existing applications will continue to be processed according to the old process model. This requires the underlying workflow system to at least provide some version management support [7]. The second option to implement the change is to migrate to the new process. It may be decided that all applicants, existing and new, will be affected by the change. Thus all admission applications, which were initiated under the old rules, now have to migrate to the new process. This migration may involve addition of some transition workflow activities as well as rollback activities. Defining the migration strategy is a complex problem and has been the target of extensive research in this area [5], [9], [10], [13].

Adaptability

The second component of change is adaptability - which is the ability of the workflow processes to react to exceptional circumstances. These exceptional circumstances may or may not be foreseen, and generally would effect one or a few instances. Of course the handling of exceptions, which cannot be anticipated, is more complex. However, a large majority of exceptions can be anticipated [6], [17], and by capturing these exceptions, the adaptability of the workflow is promoted. In fact, unless these exceptions are captured within the workflow model, their handling will continue to be done outside of the system, in the form of "system workarounds", the onsequences of which may come in conflict with process goals. However the complete set of exceptions for a given process can never be captured, thus dealing with unanticipated (true) exceptions will always be an issue [12], [14].

Using the same example as before, we can consider dealing with an admission application for a student with a multi-disciplinary background. For example, a student with a background in microbiology may be applying for a degree in IT. The review of this application may require the services of an academic outside the offering department. This may be rare but, if captured within the process model, could be handled within the workflow system.

Another example, which represents a true (unanticipated) exception, can be found in the employment workflow. An employment instance may have reached a stage where even the letter of intent has been issued. If at that time the organization issues a spending freeze, the active instances of the employment workflow will have to be dealt with, requiring rollback and/or compensation activities, even though the original employment workflow remains unchanged.

Flexibility

The third component is flexibility - which is the ability of the workflow process to execute on the basis of a loosely, or partially specified model, where the full specification of the model is made at runtime, and may be unique to each instance.

Processes which depend on the presence of such flexibility for the satisfactory attainment of process goals can be found in many applications:

- A typical example of flexibility is healthcare, where patient admission procedures are predictable and repetitive, however, in-patient treatments are prescribed uniquely for each case, but none-the-less have to be coordinated and controlled.

- Another application is higher education, where students with diverse learning needs and styles are working towards a common goal (degree). Study paths taken by each student need to remain flexible to a large extent, at the same time providing study guidelines and enforcing course level constraints is necessary to ensure a certain quality of learning.

- Web content management is also characterized by flexible processes, where especially in large projects, every development suggests the need for an overall plan to provide the objectives, approvals, and strategy, as well as a flexible means of coordinating the combined efforts of the theme designers, graphic experts, programmers, and project planners.

- Effective Customer Relationship Management (CRM), a critical component in enterprise solutions, also signifies the need to provide a flexible means of composing call center activities according to the available resources and data, by integrating CRM systems with core organizational workflow processes and underlying applications.

The key issue in flexible workflows is the modeling of the loose or partial workflow. Thus rather than enforcing control through a rigid, or highly prescriptive model that attempts to capture every step and every option within the process, the model is defined in a more relaxed or "flexible" manner, that allows individual instances to determine their own (unique) processes. How to achieve such an approach to modeling is the main focus of this paper.

3. FRAMEWORK FOR FLEXIBLE WORKFLOWS

Several workflow products and research prototypes exist. Most, if not all, provide process modeling tools that follow some variation of the workflow modeling language introduced by the workflow coalition [21]. In Figure 1, we briefly introduce the basic structures of a generic workflow modeling language [19], also based on the coalition standards to a large extent. This language will be used in later sections to illustrate various examples.

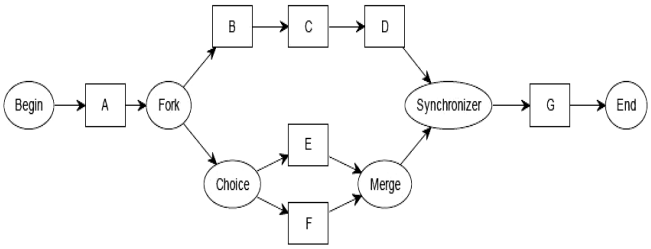

Fig 1. Workflow Modeling Language

In terms of a prescriptive language as above, one can say that the degree of flexibility is indicated by the number of instance types that can be generated from the process model. Number of instance types have a direct correlation with the number of choice constructs found within the process model [20]. For example in the above figure, a choice-merge construct encapsulates two activities, E and F. In any given execution of this process, only one of E or F will be executed. Thus the process has two instance types. This is a straightforward and well-understood concept.

However, consider a scenario where a process generates a very large number of instance types, that is, demands a high degree of flexibility. Suppose that a large number, say k number of paths are present within a choice-merge construct. Each of these paths potentially represents a complex sub-process. There can be several such constructs within the process model, which may include nesting also. One can see that a typical workflow language may not provide a very elegant means of representing such a process. In order to seek some alternative way of modeling, let us start first with some simple approaches.

- Flexibility by Definition: Flexibility may be built into the model through choice merge constructs. Limitations of this approach have already been discussed. This would result in a highly complex model, which in some cases may still be incomplete.

- Flexibility by Granularity: Flexibility may be achieved by encapsulating activity details within workflow tasks, and keeping sub-activities 'internal' (and flexible), or outside the direct control of the workflow [16]. This approach can be applied to a limited extent, but it cannot be used at a generic level without compromising the purpose of deploying workflow technology, namely to coordinate and control the flow of process activities.

- Flexibility by Templates: Flexibility may be achieved by providing separate templates for a given (set of) instance type. This slightly improves the readability and consequently maintainability of the model. However, choosing an instance type from a set of templates rather than one model with many choices will have advantages only if the number of templates can be restricted to a reasonably small number.

A common disadvantage of the above approaches is that they still rely on a prescriptive model. Thus, not only is it cumbersome to model all choices in flexible processes, there may be choices which cannot be

anticipated. Flexibility as we defined it earlier is the ability of the workflow process to execute on the basis of a partial model, where the full specification is made at runtime. To provide a modeling framework that offers true flexibility, we need to consider the factors, which influence the paths of (unique) instances together with the process definition.

4. CONCLUSIONS

Difficulties in dealing with change in workflow systems have been one of the major factors limiting the deployment of workflow technology. At the same time, it is apparent that change is an inherent characteristic of today's business processes. This paper provides a comprehensive categorization of change characteristics in workflow processes, based on which we present an approach that recognizes the presence of change, and attempts to integrate the process of defining a change into the workflow process itself. Our basic idea is to provide a powerful means of capturing the logic of highly flexible processes without compromising the simplicity and genericity of the workflow specification language. This we accomplish through pockets of flexibility in workflow specifications, which allow workflow processes to be tailored to individual instances at runtime.

References

1. W.M.P. van der Aalst, A.P. Barros, A.H.M. ter Hofstede, and B. Kiepuszewski. Advanced Workflow Patterns. O. Etzion and P. Scheuremann, editors, Proceedings Seventh IFCIS International Conference on Cooperative Information Systems, CoopIS 2000, Volume 1901 of Lecture Notes in Computer Science, pages 18-29, Eilat, Israel. Springer-Verlag. September (2000).

2. W.M.P. van der Aalst and S. Jablonski. Dealing with Workflow Change: Identification of issues and solutions International Journal of Computer Systems, Science, and Engineering, 15(5):267-276, (2000).

3. S. Ellis, K. Keddara , G. Rozenberg. Dynamic Changes within Workflow Systems. Proceedings of ACM Conference on Organizational Computing Systems COOCS 95 (1995).

4. J Eder, W. Liebhart. The workflow activity model WAMO. Proceedings of the 3rd international conference on Cooperative Information Systems (CoopIs), Vienna, Austria, May (1995).

5. Fabio Casati, S. Ceri, B. Pernici, G. Pozzi. Workflow Evolution. In Proceedings of the 15th International Conference on Conceptual Modeling, ER'96, Cottbus, Germany. Springer Verlag, Lecture Notes in Computer Science (1996).

6. Fabio Casati, Giuseppe Pozzi. Modeling Exception Behaviors in Commercial Workflow Management Systems. Proceedings of the Fourth IFCIS International Conference on Cooperative Information Systems (CoopIS99). Edinburgh, Scotland. Sep 2-4, (1999).

7. Gregor Joeris, Otthein Herzog. Managing Evolving Workflow Specifications. Proceedings of the third IFCIS International Conference on Cooperative Information Systems (CoopIS98). NewYork, USA. Aug (1998).

8. Mark Klein, Chrysanthos Dellarocas, Abraham Bernstein (eds.) Workshop on Adaptive Workflow Systems. Conference on Computer Supported Cooperative Work (CSCW), Seattle, USA. November (1998).

9. Markus Kradolfer, Andreas Geppert. Dynamic Workflow Schema Evolution based on Workflow Type Versioning and Workflow Migration. Proccedings of the Fourth IFCIS International Conference on Cooperative Information Systems (CoopIS99). Edinburgh, Scotland. Sep 2-4, (1999).

10. Chengfei Liu, Maria Orlowska, Hui Li. Automating Handover in Dynamic Workflow Environments. Proceedings of 10th International Conference on Advances in Information System Engineering (CAiSE 98), Pisa, Italy, June (1998).

11. Mohan C. Tutorial: State of the Art in Workflow Management System Research and Products, 5th International Conference on Extending Database Technology, Avignon, France, March (1996).

12. Manfred Reichert, Peter Dadam. ADEPTflex - Supporting Dynamic Changes of Workflow without loosing control. Journal of Intelligent Information Systems (JIIS), Special Issue on Workflow and Process Management (1998).

13. Shazia Sadiq. Handling Dynamic Schema Change in Process Models. Australian Database Conference, Canberra, Australia. Jan 27 - Feb 02, (2000).

14. Shazia Sadiq. On Capturing Exceptions in Workflow Process Models. Proceedings of the 4th International Conference on Business Information Systems. Poznan, Poland. April 12 - 13 (2000).

15. Amit Sheth. From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Coordination and Collaboration. Siggroup Bulletin, 18(3):17-20, (1997).

16. Keith D. Swensen, Kent Irwin. Workflow Technology: Tradeoffs for Business Process Reengineering. Proceedings of ACM Conference on Organizational Computing Systems (COOCS 95), Milpitas, CA. USA, Nov (1995).

17. Diane M. Strong, Steven M. Miller. Exceptions and Exception Handli-ng in Computerized Information Processes, ACM Transactions on Information Systems, Vol. 13, No 2, Pages 206-233, April (1995).

18. Shazia Sadiq, Olivera Marjanovic, Maria E. Orlowska. Managing Change and Time in Dynamic Workflow Processes. International Journal of Cooperative Information Systems. Vol. 9, Nos. 1 & 2. March - June (2000).

19. Wasim Sadiq, Maria E. Orlowska. On Correctness Issues in Conceptual Modeling of Workflows. In Proceedings of the 5th European Conference on Information Systems (ECIS '97), Cork, Ireland, June 19-21, (1997).

20. Wasim Sadiq, Maria E. Orlowska. Analyzing Process Models using Graph Reduction Techniques. Information Systems, Vol. 25, No. 2, pp. 117-134, 2000. Elsevier Science. June (2000).

21. Workflow Management Coalition. Interface 1: Process Definition Interchange, Process Model, Document Number WfMC TC-1016-p. (1998)