International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation and Computing (ICIAC) (12th&13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Event Driven Approach for Dynamic Business Process Adaptation

Ram Pradhan *

*(Department of Computer Science, P.E.S. Modern College of Engineering, Pune - 05 Email: rampradhan_123@yahoo.com)

ABSTRACT
As the amount of data generated by today's pervasive environments increases exponentially, actually there is a stronger need to decipher the important information that is hidden among it. We can obtain the information that really matters to our organization and use it to improve our business processes by using complex event processing. However, even when this information is retrieved, to adapt to the actual scenario business processes remain static and cannot be changed dynamically, diminishing the advantages that can be achieved. In this paper we present 'Dynamic Adaptation Framework (DAF)' a framework that combines the strengths of complex event processing and dynamic business process adaptation, which allows us to respond to the needs of today's rapidly changing environment. We use a simple online book store scenario to show how 'Dynamic Adaptation Framework (DAF)' this framework could be used to maintain the quality of service of a business process by adapting it according to the situation.
Keywords-Complex Event Processing (CEP); Aspect Oriented Programming (AOP); process adaptation; Quality of Service (QoS).

I. INTRODUCTION

Given the dynamicity of today's business environments, there is a need to continuously adapt the business processes in order to respond to the changes in those environments and keep a competitive level. One of the main concerns for on-line applications is to keep a high Quality of Service (QoS), for which they need to keep a constant monitoring of their processes. By using Complex Event Processing (CEP) we can facilitate the solution of this problem by gathering information about the different steps of the processes in order to determine whether a situation of low QoS is approaching. CEP is an emerging technology which allows us to find real-time relationships between different events using elements such as timing, causality, and membership in a stream of data in order to extract relevant information [1]. CEP can be used, for example, to prevent the theft of merchandise from stores by creating relationships between the kind, movement, and amount of the products inside the store and sending an alert when a suspicious situation is detected. However, there are some occasions in which it is not enough just to be able to obtain this information from simple raw data. or example, when monitoring the QoS, we could alert the administrator when the process is not responding as expected, but an optimal response would be to automatically adapt the business process according to the new context in order to continue in an optimal way, and this is why we developed Dynamic Adaptation Framework (DAF). The purpose of Dynamic Adaptation Framework (DAF) is to create context-aware business processes that are able to adapt dynamically in order to respond to different scenarios. The decisions of how to respond to a specific scenario are done by collecting data from different sources and transforming it into useful information, using CEP. By using an aspect-oriented approach, allowing the business process to adapt in a dynamic way, we can define alternative processes that can be woven into the business process at runtime. In this paper we use an on-line book store application to show how this framework Dynamic Adaptation Framework (DAF) works.

The objectives of this paper are:
-To show why dynamic adaptation is necessary in today's business processes.
-To show integration of CEP into business processes to help in the decision making task.
-To serve a framework that facilitates such integration by allowing the users to create dynamically adaptable business processes.

The rest of this paper is organized as follows. Section II, presents a background of the different domains used in this paper. Section III, explains the Dynamic

Adaptation Framework (DAF) architecture. Section IV presents some of the implementation and challenges. Section V, concludes and discusses some future work.

II. BACKGROUND

2.1 Complex Event Processing (CEP):

CEP is an emerging technology for finding relationships between series of simple and independent events from different sources, using previously defined rules. To enrich the enterprise's existing processes, the CEP technology can be used, among a lot of other things, by introducing rules that will allow the capture of relevant information from the different steps of their business process. For example, let us consider the scenario of a retail store that keeps a record of its inventory in an existing Enterprise Resource Planning (ERP) system and wants to keep a live monitoring of its stocks in order to prevent shortage. For achieving this, the store installs a CEP engine that will monitor the products' movements through their life cycle in the store process by receiving and analyzing all the events generated by every change in time for the state. Since the objective is used to monitor inventory [1], the CEP engine will only keep the events related to changes in the inventory and forget about the other [1]. By preparing the necessary CEP rules, the configuration is set to specify the lowest acceptable stock of product that the store can have to avoid a shortage, e.g., a 5% for some low-demand products and 10% for normal products. The CEP engine alerts the managers so they can make a supply order, whenever a product reaches a minimum. In addition to that, CEP can also be used to predict unexpected situations. To complement the above mentioned example, we can say that because of hand sanitizers and global pandemic alert are very popular and are selling a lot more than usual. Given this demand, the store will run out of hand sanitizer before they can resupply it, even with the minimum stock alert. By adding some specialized CEP rules to analyze the frequency of sales of each product during the last 4 or 5 hours, the engine can polarize these values to know in advance (if the sells rates are kept) that it will need to resupply before the expected time, which will allow them to react in time even before it reaches the minimum level.

2.2 Aspect-Oriented Programming (AOP):

The domain of Aspect-Oriented Programming (AOP) appeared in 1996. The domain of AOP inherits results from other programming approaches, such as reflection, open implementations, meta-object protocols, and generative programming while original and innovative. AOP, as a new programming paradigm, introduces notions such as; aspect, join point, point-cut and advice code. However, these notions do not replace existing ones, such as class, object, procedure or method. Rather, AOP must be seen as a complement to these existing techniques. Furthermore, these notions are not specific to a programming style (e.g., object-oriented or procedural) or a given syntax (Java, C#, Ada, COBOL, etc.). Aspect-oriented extensions exist for many languages, object-oriented or procedural. AOP has been proposed as a technique for improving the separation of concerns in software systems and for adding crosscutting functionalities without changing the business logic of the software. To address concerns, such as security in a modular way, AOP provides specific language mechanisms that make it possible. AOP languages and tools can be applied at compile-time or at run-time, this giving the designer the flexibility to use them in the most appropriate moment.

III. ARCHITECTURE DIAGRAM

The architecture contains:

A. A Log file (a text file) which captures the current events invoked as the users interact with the application

B. CEP engine (implemented as a class) which analyses the current events and using context information decides whether an adaptation condition has arrived.

C. The Aspect Manager which adapts the business process as per the adaptation condition on the basis of advice and point cuts specified at design time.(implemented using XML)

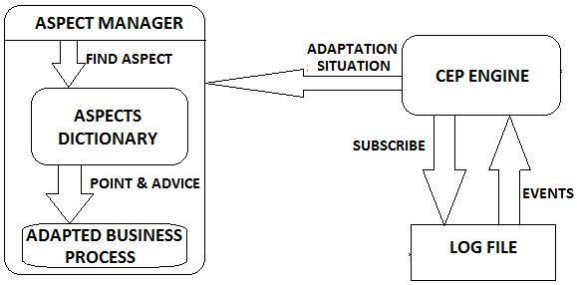

Fig. 1: Dynamic Adaptation Framework (DAF)

Architecture Diagram.

IV. IMPLEMENTATION

4.1 EXAMPLE

To show how CEP and Dynamic Adaptation Framework (DAF) can be used to maintain a high QoS we present an on-line book store service. In this service, the client goes through a process of ten steps to get a book, as shown in Fig. 2. The client starts the process by selecting a book from the available book list, and then client views the details of the selected book, adds it to the cart, enter quantity, modifies cart if required and finally pays the bill through credit card, the system asks the client for satisfaction survey and sends a confirmation with Tracker Id.

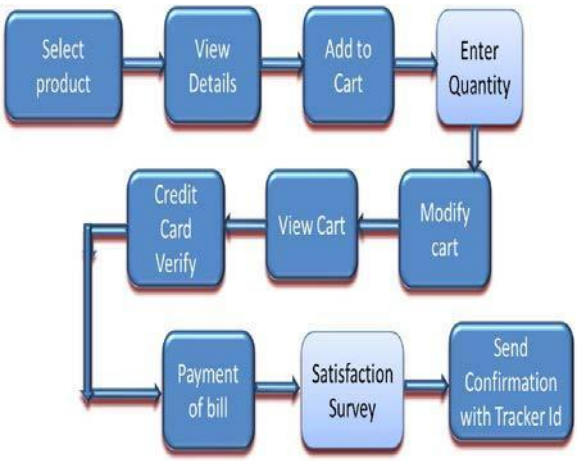

Fig. 2: Online Book Store.

Whenever the application is getting a considerable amount of traffic, which could decrease the response time of the servers, the owners of the on-line application want to avoid the invocation of some optional tasks of the business process. This will help to maintain the QoS and allow the user to complete the shopping process in fewer steps and to spend less time waiting for the application to respond. The optional tasks that could be excluded from the business process of this scenario, without altering the main objective, is: satisfaction survey. To skip them, an additional path from the preceding activity to the next will have to be set. Then, the process will have to be redeployed and the application restarted, for the changes to be considered. In this scenario, the QoS is considered with two parameters: service performance and service availability. The performance of a web service can be measured by considering the time it takes to respond to a user query, while the service availability can be simply measured by the existence or not of a response from the service.

4.2 CHALLENGES

A. Situation identification:

In order to maintain the QoS of business applications we need to identify the situations when the application needs modification to adapt as per the changing business environment. In Dynamic Adaptation Framework (DAF), we use CEP rules in order to identify these special situations. These situations allow us to provide the scenarios and special circumstances that we are interested in and to define how we want to respond to them. Using the information from the process and even other sources, the CEP engine matches the corresponding rules and alerts the system when such a situation is found.

B. Code scattering:

In an application, handling the code for the adaptation decisions and the code for business process's execution together is real challenge. We need to separate the code for making the decision when adaptation is needed from the code which actually executes the business processes in desired manner. To deal with this code scattering, we separated the event definitions from the business process. By doing this, we can focus all the decision making code in a separate file which can be easily maintained without modifying or even interrupting the business process execution. Even when the CEP engine is changed, the adaptation rules only need to be written once.

C. Process redeployment:

The whole process would need to be redeployed, if we needed to add a new condition for adaptation of

business processes, or change the way an existing condition for adaptation is tested. Given the dynamic reconfiguration of the components provided by the component-oriented model, allowing the process to be adapted without redeploying it and without losing any current transaction in the process, once our business processes is transformed to integrate components, these integrations can be altered at runtime.

V. CONCLUSION AND FUTURE WORK

Process adaptation and Complex Event Processing are two topics that are creating a lot of interest in the research community; however there is still no integration of both domains. In this paper we presented Dynamic Adaptation Framework (DAF), a framework that intends to facilitate the integration of CEP into existing business processes and to allow these processes to be dynamically adapted to different circumstances. Before this framework it was not possible to dynamically change the business processes as per the changing environment. And the business process changes were made at the cost of losing all the online clients which is not acceptable in commercial applications. In future we would like to develop a plug-in which can be used as a generalized approach to integrate CEP into any existing business processes.

REFERENCES

[1] D. C. Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley Longman Publishing Co., Inc., 2001.

[2] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut, "Quality of service for workflows and web service processes," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, no. 3, pp. 281 – 308, 2004.

[3] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl, "Qos aggregation for web service composition using workflow patterns," in EDOC '04: Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International. IEEE Computer Society, 2004, pp. 149–159.

[4] M. B. Juric, Business Process Execution Language for Web Services BPEL and BPEL4WS 2nd Edition. Packt Publishing, 2006.