IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 7, September 2014.
www.ijiset.com

ISSN 2348 – 7968

Enhancing Workflow Management System Based On Webservices

Kalaivani. A [1], Saranya. R[2], Indra priyadharshini .S[3]

Assistant Professor, Department of Computer Science and Engineering,

R.M.K. College of Engineering and Technology, Puduvoyal, India.

***Abstract*— Traditional Workflow Management Systems (WfMS) is hard to meet the requirements of the dynamic, flexible modern business process. In recent years, Web services technology has been developing. It's not only makes the activity in workflow by Web services form, but also workflow itself is invoked by Web services, especially semantic web services technology is developing, which makes it possible to compose automatic Web service workflow. This paper proposed a flexible workflow management system based on Web services. We propose workflow engine that handles the overall process flow, calling the appropriate web services, combines two or more web services in dynamic manner and determining the next steps to complete. We also present the design and implementation of a distributed web services workflow system. Workflows are usually used in human-to-human business applications. This means that a task is performed by different users. Information need to be distributed from one employee to a colleague, a customer or a partner. A business application manages these kinds of processes and takes care about messaging, security and the documentation of the whole business process. The Workflow offers an easy to use technology to simplify the development of these kinds of BPM (Business Process Management) solutions. It fulfills the requirements to a scalable, reliable, transactional, robust and simple deployable Java EE Workflow System.**

***Keywords*— workflow management system, Workflow engine, reliable services.**

I. Introduction

This Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. Web services composition is an emerging methodology for building value-added applications by aggregating several exiting web services together according to dynamic business requirements.

Workflow management systems (WfMS) [1] are widely used to manage business processes due to their known benefits such as automation, co-ordination and collaboration between entities. However, the existing commercially available workflow management systems do not offer sufficient flexibility for distributed organizations participating in the global market. In order to enable the enterprises in different places finish their cooperation tasks more fluently, some workflow management mechanism with good flexibility and high self-adaptability is urgent. It is needed to solve the poor interoperation problems and integrate the workflow processes in different enterprises or application systems in each associated enterprise.

The Web Service Architecture (WSA) is intended to provide a common definition of a Web service, and define its place within a larger Web services framework to guide the community. The WSA provides a conceptual model and a context for understanding Web services and the relationships between the components of this model. The idea behind the work flow engine is to create a composite web services and to provide security and privacy in flawless manner.

In the present scenario, there are many composite services are in web which are organized in category wise. The most popular one is travel arrangement for the tourism. The services of a particular application process are defined and worked separately. If one of the service in that application is completed only when one of the processes was completed then the other process will start processing continuously.

There are a number of important technical requirements that must be addressed when designing business Processes involving multiple web services running over a long duration. Knowledge of these requirements will help in positioning the various standards that have been introduced for web services orchestration.

The ability to invoke services in an asynchronous manner is vital to achieving the reliability, scalability, and adaptability required by today's IT environments. With asynchronous support, a business process can invoke web services concurrently rather than sequentially in order to enhance performance. For example, a purchasing system may want to interact with multiple supplier web services at the same time, looking to find the supplier that can offer the lowest price of earliest shipment date. Asynchronous support can be achieved through web services through various correlation techniques. Orchestrated web services that are long-running must also manage exceptions and transactional integrity.

The architecture must take into account how the system will respond if there is an error or if the service invoked does not respond in a given time. Since nearly 80% of the time spent in building business processes is spent in exception management [HUR], one can see why this is a critical component. The architecture must also have a way to manage transactional integrity if something goes wrong. Traditional transactional systems that are ACID-based are typically not sufficient for long-running, distributed transactions. Resources cannot be locked in a transaction that runs over a long period of time. The notion of "compensating transactions" has been introduced to undo an action if a process or user cancels it.

Web services orchestration must be dynamic, flexible, and adaptable to meet the changing needs of a business. Flexibility can be achieved by providing a clear separation between the process logic and the web services used. This separation can usually be achieved through an orchestration engine. The engine handles the overall process flows, calling the appropriate web services and determining the next steps to complete. With this approach, an IT organization can swap out services in the overall process flow. Reusability is also a major benefit with web services, and there is often a need to compose higher-level services from existing orchestrated processes.

We perform the process were run as independent manner. The processes are dynamic in the application, design and create the work flow engine that provides some advantages over the present scenario like less dependency, user satisfaction, efficiency, reliability and less time consume.

II. RELATED WORK

An Traditional Workflow management systems pay more attention to modeling simple and static process, and required all information in the processes should be static. They can't model those dynamic processes and can't support the heterogeneous distributed environment. Currently, some scholars become to propose some Web-based WfMS. The work of Zh. Piefa et al. [2] uses Browser/Server architecture and middleware technology to support the distributed workflow execution, and promote the cooperation among many Web-based workflow servers. But it is unpractical because it requires all workflow servers should be based on Web and lacks of reusability. It can't completely allow the dynamic binding and invocation of processes and is unable to permit the seamless information exchange and cooperation.

The works of Zh. Hongshan [3] use XML-PDL (XML Process Definition Language) as the process definition language and employ J2EE architecture to build the system. But it also lack of dynamic process binding and invocation ability. Web service is a series of standards and developing standards, which are designed and named by W3C, is used to promote communication beyond medium between programmers. These standards include XML (eXtended Markup Language), HTTP, UDDI (Universal Description Discovery and Integration), WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol) and so on. Its characteristics of loose coupling, reusability and cross-platform make it one of the best choices to solve the problems in traditional or Web-based workflow systems. So, this paper tries to use Web services to implement activities or business processes in workflow management system by using workflow engine, which can finally allow the cooperation and integration among the WfMS that cross enterprises or platforms in Internet.

Web services transactions have received much attention recently. Some works (e.g., [4], [5], [6]) give a comprehensive analysis on traditional database transaction models as well as some emerging industrial Web services transaction specifications (e.g., THP, BTP, and WS-Transaction). They point out the pros and cons of these efforts in the Web services environment and, therefore, establish a solid Foundation for further research. Transaction has achieved a great success in the database community [7], [8]. One of the most important reasons is that the operations in database (e.g., insert, update, and delete) have clear transactional semantics. However, this is not the case in Web services. To solve this problem, the extension mechanism of WSDL can be exploited to explicitly describe the transactional semantics of Web services operations [9], [10]. The different approach to orchestration, BPEL4WS primarily focus on the creation of executable business processes, while WSCI is concerned with the public message exchanges between web services. WSCI takes more of a collaborative and choreographed approach, requiring each participant in the message exchange to define a WSCI interface. BPEL takes more of an "inside-out" perspective, describing an executable process from the perspective of one of the partners.

BPML has some complimentary components to BPEL4WS, both providing capabilities to define a business process. WSCI is now considered a part of BPML, with WSCI defining the interactions between the services and BPML defining the business processes behind each service. BPML includes both transactional support and exception handling mechanisms. Both short and long running transactions are supported, with compensation techniques used for more complex transactions. BPML uses a scoping technique similar to BPEL4WS to manage the compensation rules. It also provides the ability to nest processes and transactions, a feature that BPEL currently does not provide. Finally, a robust exception handling mechanism is available within BPML, following many of the constructs in XLANG. Timeout constraints can also be specified for specific activities defined within the process. In our work, we can also adopt this mechanism to support our new workflow Web services.

A. Motivating example

We consider an application for online travel arrangement. In Fig. 1 shows a composite service that aims at planning and reserving travel arrangements for a customer. The composite service starts when it receives a request from the customer. It then invokes two Web services simultaneously: the flight service reserves an appropriate flight while the attraction service recommends some tourist attractions according to the customer's preference. After the flight reservation is done, the composite service sends a request to the hotel service and waits for confirmation. Upon receiving the responses from both the attraction and hotel services, the composite service invokes the computation service to compute the distance between the hotel and the attractions. According to the result, either the bike service or the car service is selected for execution to make the appropriate reservations. The shop service is invoked to get some sales information before the composite service finally sends the arrangement details to the customer.

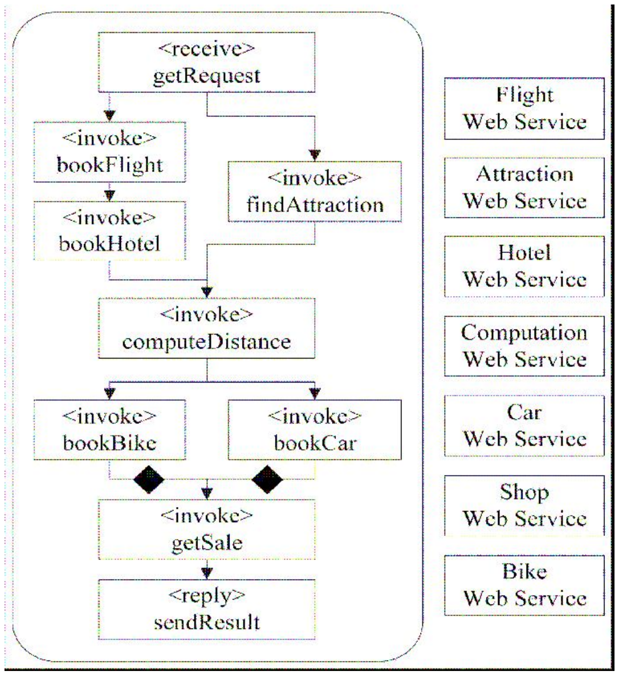

Fig 1 Travel arrangement service

III. PROPOSED SCHEME BASED ON WEB SERVICES

A. *Architectural Model*

Our proposed scheme of flexible WfMS based on Web services is composed of process definition tool, workflow engine and management& monitor tool. Figure 2 depicts the major components and interfaces inside the workflow architecture.

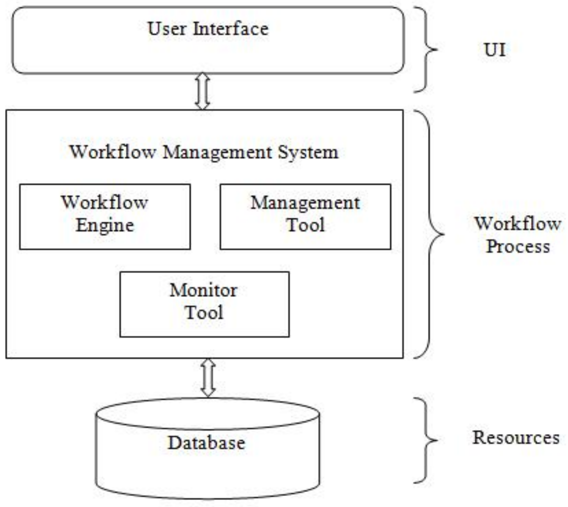

Fig 2 Scheme of flexible WfMS based on web service

In Figure 2, the process definition tool mainly takes charge of process modeling on enterprise workflows which integrate various kinds of Web services in WFMS in different enterprises. Workflow engine provides a runtime execution environment and manages the overall processing and the execution of workflow process instances. The basic functions of this component include the interpretation of business process specification, creation of new process instances, execution and management of instances, navigation between work items, management of workflow information, routing data between instances.

Workflow management & monitor tool manages the Workflow Services, Configuration, Management, and controls the overall workflow environment. Some of the basic functions of this component include establishment of users, assignment of work items, exception and error processing, event generation and notification, auditing, tracking and reporting of results, tracking and reporting of statistics, versioning and change management. For many enterprises, it is unpractical and impossible to totally discard all current business application systems. So, they can use Workflow engine these systems manages the workflow process of Web services and allow the invocation using SOAP through Internet. Workflow Client Applications interact with WFMS to utilize workflow services and the interaction with the workflow engine is done through an interface and a queue of assigned work items.

B. *Work Flow Model*

Web service is a bridge between the client and server. The Client communicates to server through web services. Client sends a request through web service 1 and the web service1 response for the client request to workflow engine. With the response from the workflow engine, a web service sends a request to next web services through the workflow engine then the whole response will be send to Client through workflow engine. Then the whole system is referred as Workflow management system. It manages the whole process, monitor the status, error handling, fault reporting and manages the execution runtime process.

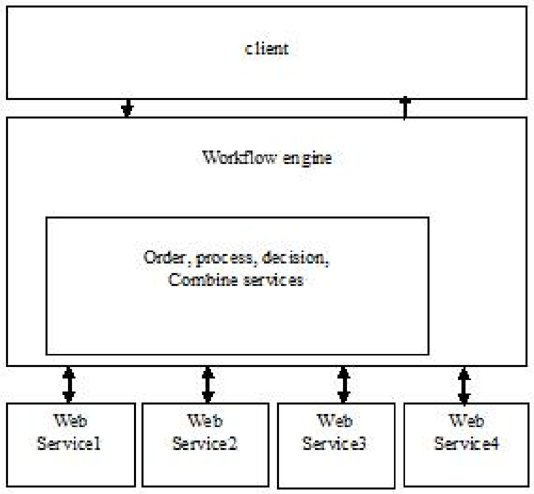

Fig 3 Workflow model1

In figure 3, a client sends a request and access the Webservice1 through the Workflow engine. The Webservice1 output is send to the Workflow engine. Based on the output of webservice1, the workflow engine decides which web services to access next. The output of the service 2 and service3 is integrated in the workflow engine and provide response to the user request. This type refers as workflow chain process. It should satisfy the user needs and reduce work load of users.

C. Work Flow Engine Design

The major functions of workflow engine includes: explaining process definition, debugging the process execution, creating and managing the process instance execution, debugging the activity execution and creating the work item, maintaining workflow control data. The structure of workflow engine is shown as figure 4:

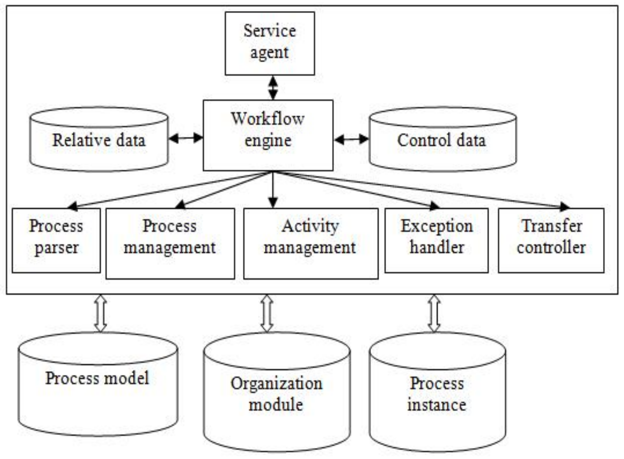

Fig 4 Structure of Workflow Engine

In Figure 4, service agent mainly takes charge of all interaction events with UDDI. It will search UDDI registration center to decide which service will be most suitable one according to the restriction conditions on activity in workflow.

IV. Conclusions

In this paper, we first propose a Web services-based flexible WfMS and present the system architecture. Then, we analyze several key technologies, including workflow model, workflow engine, and Web service invocation and demonstrate how to use Web services to implement the activities and business processes in traditional workflow management to improve the flexibility and self adaptability of workflow system. We propose a solution to ensure reliable and flexible Web services. We propose workflow engine that handles the overall process flow, calling the appropriate web services, combines the two or more web services in dynamic manner. It fulfills the requirements to a scalable, reliable, transactional, and robust. In the future we intend to use security mechanism for high authentication and we integrate workflow ontology and domain-specific ontology to our system and enhance the power of the Semantic Web when using the Web Services.

References

[1] S. Meilin, Y. Guangxin, X. Yong, “Workflow Management Systems: A Survey”. Proceedings of IEEE International Conference on Communication Technology, 1998.

[2] Zh. Piefa, H. Kaihu, Zh. Jinfeng, Q.Liang. chun,“Research on enterprise workflow manager system based on Web,” Development & Innovation of Machinery & Electrical Products, 2006, 19(5), pp.31-33.

[3] Zh. Hongshan, “A design of workflow management system based on Web,”Journal of Capital Normal University (Natural Science Edition), 2006, 27(2):12-14.

[4] S. Dalal, S. Temel, M. Little, M. Potts, and J. Webber, “Coordinating Business Transactions on the Web,” IEEE Internet Computing, vol. 7, no. 1, pp. 30-39, Jan./Feb. 2003.

[5] M.P. Papazoglou, “Web Services and Business Transactions,” World Wide Web, vol. 6, no. 1, pp. 49-91, 2003.

[6] B. Limthanmaphon and Y. Zhang, “Web Service Composition Transaction Management,” Proc. Australasian Database Conf. (ADC ’04), pp. 171-179, 2004.

[7] A. Elmagarmid, Transaction Models for Advanced Database Applications. Morgan-Kaufmann, 1992.

[8] G. Weikum and G. Vossen, Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery. Morgan-Kaufmann, 2002.

[9] T. Mikalsen, T. Tai, and I. Rouvellou, “Transactional Attitudes: Reliable Composition of Autonomous Web Services,” Proc. Workshop Dependable Middleware-Based Systems at the Dependable Systems and Network Conf., 2002.

[10] P.F. Pires, M.R.F. Benevides, and M. Mattoso, “Building Reliable Web Services Compositions,” Proc. Web, Web-Services, and Database Systems, NODe Web and Database-Related Workshops, pp. 59-72,2003..