Research and Design of Lightweight Workflow Engine Based on SCA

Huaizhou Yang, Bowen Lv, Wenbo Shi, Jingxin Sun
Department of computer science, Xi’an Shiyou University, Xi’an 710065, China, lvbowen3@163.com

Abstract. In order to overcome the shortcomings of the traditional workflow engine structure and defects, this study proposes a loosely coupled lightweight workflow engine based on SCA (Service Component Architecture) and BPEL (Business Process Execution Language for Web Service, also known as WS-BPEL or BPEL4WS) engine. The support for traditional workflow engines for heterogeneous is improved by the separation between business logic and transport protocol, so that the engine is no longer limited to a specific transport protocol.

1. Introduction
A workflow is a business activity that can be fully or partially automated. It enables documents, information, or tasks to be flowed and executed between different participants according to certain rules[1]. As a core component of the workflow management system, the workflow engine provides the necessary services and operating environment for the workflow. The services provided include: (1) interpretation of the process model; (2) control of the process instance; (3) travel between the activities of the process; (4) generating work items, notifying the user for processing; (5) workflow control data and workflow-related data maintenance; (6) call external applications and access workflow-related data[2].

With the development of multi-system integration mode, the workflow engine is more "heavyweight". Mainly manifested by the tight coupling between internal modules, web services, workflow engine and application system, which makes the engine lack of flexibility and lightweight, coupled with the expansion of its peripheral components, making the engine structure too bloated and cumbersome. This paper introduces the construction idea of SCA, combines the characteristics of BPEL workflow engine, proposes a low-coupling lightweight workflow engine, and combines the engine architecture to analyze the workflow of the engine.

2. The Combination of SCA and BPEL Engine

2.1. SCA Architecture and BPEL Thoughts
As a service-oriented infrastructure design, SCA is an implementation of SOA thinking. It integrates different services into a single business solution, greatly simplifying the programming model for implementing business service components, and allowing these components to be implemented in different programming languages. The main feature of SCA is that it passes the service-independent content, such as the transport protocol to the SCA’s operating environment for processing. Therefore, the interface described by SCA is a loosely coupled service-independent interface[4]. At the same time, when developing SCA components, the client and server do not need to know the technical details of...
the transport protocol, and the service component can freely choose the transport protocol to achieve the maximum reusability of the service component[3].

BPEL is a business process description language. The business process unit it organizes and orchestrates is a well-defined Web service. Therefore, BPEL can be regarded as a service-oriented business process definition language. BPEL can implement a service-oriented architecture from top to bottom by orchestrating, combining, and coordinating Web services[5]. The BPEL language itself cannot complete the instantiation of business processes, as well as specific Web service calls or business process management functions. It mainly defines and describes business processes. The driver, execution, and specific service calls of the process instance are all done by the BPEL engine. It can be said that the BPEL engine provides the necessary operating environment for BPEL execution.

2.2. BPEL Engine in SCA Architecture

In the SCA architecture, each application consists of one or more components. Each Component can be implemented by a variety of programming techniques, either Java applications or C++ applications, or through BPEL definitions, and even these components can be distributed across machines. Component can be assembled into a larger composite to complete the specific application module[8].

We use the BPEL engine as a Component in the SCA architecture, which provides users with the functions of driving, executing, and calling specific instances of the process instance. Each Component is a parallel hierarchical relationship that can be used as a component alone or as a composite construct[6]. The BPEL engine can orchestrate and call other service components in composite, and the whole can serve as a composite with a certain logic function. The location of the BPEL engine in the SCA architecture is shown in Figure 1[9].

![BPEL architecture diagram in SCA](image)

3. Design of Workflow Engine

3.1. Design Idea of Lightweight Workflow Engine Based on SCA

The lightweight workflow engine should be designed from the perspective of flexibility and low cost, and only implement some essential functions and features according to specific needs. And according to the business process model to complete the process analysis, management and operation of the entire process life cycle tasks. Based on the SCA architecture to implement the BPEL workflow engine design, the BPEL engine exists as a Component in the application, so that it can not only complete the execution and management of processes and other tasks, but also interact and cooperate with other components, so as to maintain the loosely coupled architecture and flexibility of the system.

The working principle of the engine is that when the engine obtains the BPEL process definition file, it will parse out the business process and generate the corresponding process instance, and the operation of the instance is monitored and managed by the BPEL management tool. Generally, a process instance is generated by the corresponding condition of the activity accepted in the process definition, so a process definition template can correspond to multiple process instances[7]. When the engine is executed, the corresponding process instance can be executed through the call to the Web service, and then the corresponding return value is obtained according to the jump condition and the call target in the
process instance, thereby completing the driving and execution of the entire business process. Since the business process instance in the engine may be long-running, a persistence mechanism such as a database is required to store state information of the process instance and management control information[10].

3.2. The overall Architecture of the Engine

Through the analysis in the previous section, we understand the working principle of the workflow engine, combined with the characteristics of the SCA loose-coupling architecture and the top-down discovery of SOA services, we give the architecture diagram of the lightweight BPEL engine. As shown in Figure 2, the overall architecture of the engine given in this paper can be divided into two levels. The topmost part includes the business process management module and the process definition module. The process definition module includes a user interface, a process orchestrator and a parser; the business process management module includes a monitoring manager, a process queue, an executor, a business process instance manager, and an SCA interface module. The bottom layer includes a log management module for saving user single-step operations and a persistence management module for persisting data.

The process orchestrator is responsible for defining the business process by the graphical design tool Active Designer and outputting the process definition file. The BPEL process definition file is parsed by the parser, the corresponding BPEL process definition template is generated and finally the obtained template is persisted into the database. In the business process management module, messages and other service component calls can be obtained through the SCA interface, and then the process instance manager can be used to drive and manage process instances. Then you can choose to execute the process instance or match the process instance in the process queue to the next stage. The monitoring manager displays the status information of the processes and activities being executed, and the status of the processes or activities can be changed visually.

![Fig.2 Lightweight BPEL engine architecture diagram](image)

The biggest feature of the engine architecture is the use of the external interface defined by the SCA standard, which changes the way the BPEL engine interacts externally. The componentization of the BPEL engine greatly enhances the need for flexibility and loose coupling for lightweight engines. At the same time, SCA architecture realizes the separation of service components and transport protocols,
so that service components can freely bind other transport protocols. This loosely-coupled architecture enables us to focus on the implementation of business logic without having to know the interface code of the other side in the inheritance process of the system. Through the combination of the BPEL engine and the SCA architecture, the BPEL engine is not limited by the specific transport protocol, and the loose coupling of the business logic and the transport protocol is realized, thus realizing our demand for the "lightweight" engine.

3.3. Relationship of Workflow Engine Building
We have given the overall architecture of the SCA-based lightweight workflow engine, and analyzed the various functional modules inside the engine. For the workflow engine Component, this paper thinks that it should be composed of three major components, namely Process Definition Component, Business Process Management Component and Data Maintenance Component. The collaborative relationship between them is shown in Figure 3. The process definition build is the input of the entire engine, responsible for providing the process definition file; Business process management is the core of the entire engine. The engine's driver execution is completed by it. It mainly includes the monitoring manager, process instance manager, executor and process queue. The data maintenance build is responsible for maintaining the related operations of the entire process, and process definition templates and persistence of process instances.

In the process definition artifact, user input information is provided to the BPEL process choreography module through the user interface. The BPEL process choreography module completes the description of BPEL business process in XML file format, and then the parser obtains its interface for parsing. In the business process management component, the SCA binding module provides an interface for message interaction with the outside world. The process instance manager is responsible for the driving and management of the business process, and then the process instance is executed by the executor or stored in the process queue for saving. The status information of the execution process and activities is completed by the monitoring management module, and the status of the process or activity can be changed through visual operations. The data maintenance component mainly completes the persistence of data through the persistence manager, and then records the various steps in the business process driving and execution through the log manager.

3.4. Execution mechanism of Workflow Engine
When the engine receives an external message request, it first looks for the presence of the requested business process instance. If it exists, the BPEL engine will perform the matching work according to the
received message request and execute the corresponding business process. If the execution condition of the process is met, it will call the corresponding service component to execute the process instance, then change the state of the process instance in the process queue, return the message and end the process instance; if the execution condition is not met, it will wait for the next process execute and return the message.

If no matching process instance is found, it needs to match the corresponding process definition template according to the message request. If the matching is successful, the business process is instantiated, the process is stored in the process queue, and then the message processing is performed to execute the corresponding business process. If the corresponding process definition template is not found and the instantiation process fails, the message is returned and the process ends. The specific execution process of the engine is shown in Figure 4.

![Execution flow chart of engine response service request](image)

**Fig.4 Execution flow chart of engine response service request**

### 4. Conclusion

Based on the SCA architecture and the BPEL workflow engine, this paper proposes a lightweight workflow engine design. The BPEL engine is componentized and changed through the separation of the business logic and communication details of the components in the SCA architecture. Its external interaction mode makes up for the lack of support of the traditional workflow engine for heterogeneous environments and the tight coupling between functional modules. The language independence and coarse-grained features of the components in the SCA architecture enable the BPEL engine to both arrange and invoke other service components and provide services to the outside world. It is a good technical choice for the reuse of legacy system reuse problems. The current research content mainly includes the transaction processing and exception handling mechanism of the optimization process, as well as the in-depth improvement of the security of the engine.

**Acknowledgments**

This research was sponsored by Xi'an Science and Technology Project of China under the grant No.
201805038YD16CG22(2).

References
[1] Guan Z, Hernandez F, Bangalore P, et al. Grid-Flow: a Grid-enabled scientific workflow system with a Petri-net-based interface[J]. Concurrency & Computation Practice & Experience, 2010, 18(10):1115 -1140.
[2] Yan Mingan. Research on Lightweight Workflow Engine Based on Web Services[D]. Harbin Engineering University, 2007(in Chinese).
[3] Marino J, Rowley M. Understanding SCA (Service Component Architecture) [M]. Addison-Wesley Professional, 2010.
[4] Karmarkar A, Edwards M. Assembly of Business Systems Using Service Component Architecture[C]// International Conference on Service-oriented Computing. Springer Berlin Heidelberg, 2006.
[5] Juric M B. Business Process Execution Language for Web Services BPEL and BPEL4WS 2nd Edition[C]/ Packt Publishing, 2006.
[6] Qi Wei. Design and implementation of lightweight BPEL engine based on SCA[D]. Beijing University of Technology, 2008(in Chinese).
[7] Peng Wei. Research and design of workflow engine based on BPEL[D]. National University of Defense Technology, 2008(in Chinese).
[8] Yu Hao, Zhu Cheng, Ding Peng. SOA Practice: Building Enterprise-Level Applications Based on Java Web Services and BPEL[M]. Electronics Industry Press, 2009(in Chinese).
[9] Wu Yan. Research and implementation of service component combination execution engine based on BPEL[D]. Dalian Maritime University, 2011(in Chinese).
[10] Li Wen Design and Implementation of Web Services combination of WS-BPEL execution engine based on[D]. Northwestern University, 2009(in Chinese).