Context-aware BPEL workflow system using aspect-oriented programming

Minsuk Kim, Donggyu Kuak, Jongsun Choi, Jaeyoung Choi

(*Computer Science and Engineering*, *Soongsil University*, *Seoul 156-743*, *Korea*)

Abstract: Business process execution language (BPEL) is a most recognized standard workflow language. However, it is difficult to be used in the ubiquitous system computing environment because it is difficult to describe the context information in the selection of the flow through the branch. To solve this problem, we propose a new BPEL workflow system with context-awareness by using aspect-oriented programming (AOP). This system is composed of a BPEL system module and a weaving module using AOP for context-aware. The BPEL system module generates a BPEL workflow program. And the weaving module converts a context-aware mark-up language (CAML) document to the aspect-oriented program that is applied to context-aware code without modification of the existing BPEL document. We also define a new document form that is called CAML, which provides a context-aware that is not available in BPEL. The system can generate a context-aware workflow program. It is developed in a way that inserts context information using AOP to provide context-aware services.

Key words: aspect-oriented programming (AOP); business process execution language (BPEL); context-aware workflow

CLD number: TP311 **Document code**: A
Article ID: 1674-8042(2012)02-0119-04 **doi**: 10.3969/j.issn.1674-8042.2012.02.005

Ubiquitous computing has emerged as the core infrastructure because it is possible to build an invisible wireless network infrastructure due to embedded device miniaturization and the development of wireless communications. Therefore, assignments such as reimplementation appropriate to the environment are conducted in many places. The purpose of applying ubiquitous computing environment is to provide appropriate services in appropriate time using context-awareness. Therefore, when we develop a computing system on ubiquitous environments, the most important things are the context-awareness and the workflow system for context-awareness.

Business process execution language (BPEL) is the standard workflow language that defines business process and executes it in a web-service environment[1]. BPEL is used as the core solution to real world problems with associated workflow. And it is applied in various fields especially office automations, led by a business process management system (BPMS), such as health care and education.

In order to develop context-aware workflow system, we may choose two alternative ways: first, developing a new workflow language and a new workflow engine; second, reusing existing workflow languages and adding the syntax that can describe the situation in ubiquitous computing environment.

The first method requires knowledge of a new workflow language. In addition, it has a fatal disadvantage in that it cannot be reused in existing workflow systems and applications. However, except for additional syntax, the second method does not require knowledge of other studies, and it can be reused as the existing workflow systems and applications. Therefore, to develop a context-aware workflow system for ubiquitous computing environment, it is quite easier to reuse existing workflows, applications and developmental environment.

To add a new function while keeping the existing workflow language, a new programming technique is required. Aspect-oriented programming (AOP), which is a program-developing method, suggested by Kiczales in 1997[2] was studied to modularize common and additional requirements that could not be modularized in object-oriented programming.

We suggest a new BPEL system with context-awareness using AOP and a new document model, context-aware mark-up language (CAML), which describes context-aware information. The proposed system has some features that can reuse existing BPEL language and promote context-awareness in BPEL workflow.

This paper suggests related research in section 2 and the composition of the system in section 3. Sec-

* **Received data**: 2011-09-20
Foundation item: Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0025831)
Corresponding author: Jaeyoung Choi (choi@ssu.ac.kr)

tion 4 introduces CAML and section 5 includes a conclusion.

1 Related research

The purpose of this paper is to show how to add context-awareness to BPEL language using aspect-oriented programming techniques. The augmented BPEL with context-awareness is used to design the workflow system that is available in ubiquitous environments. In order to add context-awareness while keeping a BPEL document, other methods such as aspect-oriented programming are needed.

1.1 Aspect J

Aspect-oriented programming method is the programming-development methodology suggested by Kiczales et al. in 1997. Programming-development methodology has been developed from batch process programming to structured programming, then to object-oriented programming. Structured programming and object-oriented programming provide the way to modularize requirements using a function. A developer can classify domain into concerns in a way in which he/she wants to handle. And in aspect-oriented programming, concerns are divided into core concerns and crosscutting concerns. Aspect-oriented programming classifies requirements that are hard to modularize through existing programming methodology such structured programming or object-oriented programming as crosscutting concern, and it modularizes it as an aspect unit by increasing the usability of the module. Each core concern is modularized as the function of the object. Each module has common process factors such as logging, security transaction, and so on. If such common process factors are classified as crosscutting concern and processed, it will be an advantage because when each module is changed or used repeatedly, it can maintain the independence of the module and simultaneously maximize the reusability.

1.2 BPEL to Java (B2J)

B2J is one of the BPEL engines that receives a BPEL document through input, handles the BPEL processes, and provides users with results. B2J engine based on web-services is composed of a coordinator, a worker and a runner. BPEL coordinator is the program that receives a BPEL document as input, transforms it into source code, compiles it and generates a Java target program class file. BPEL runner and worker are programs that execute Java target program generated by coordinator. Coordinator of B2J is the module that transforms a BPEL document into a Java program and compiles it. It is the core of B2J.

1.3 Context4BPEL

Context4BPEL[3] is a famous research work that adds the context-aware function to BPEL workflow. They develop an expanded BPEL engine[3], which can describe the context-aware information. However, it does not provide compatibility with the existing BPEL since the expanded BPEL document for context4BPEL is not reusable for the existing BPEL document.

2 Composition of the system

This paper proposes a context-aware BPEL workflow system using CAML document that includes AOP information and context-aware information. Fig. 1 shows the conceptual image of the suggested system.

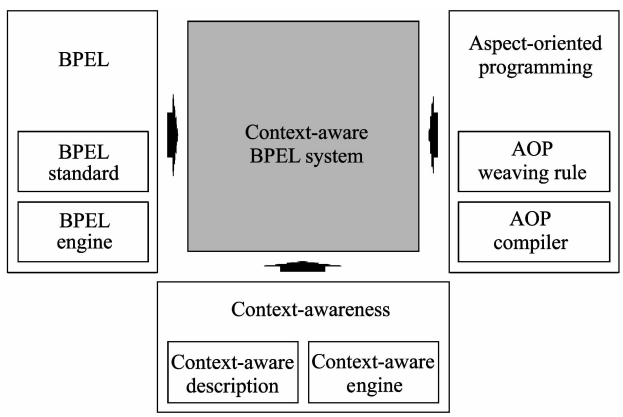

Fig. 1 Conceptual diagram of context-aware BPEL system

The new system consists of BPEL, AOP and context-awareness. BPEL represents BPEL standard. It means that the system can use an existing BPEL document. AOP represents AOP weaving rule. It means that the system has independence between BPEL workflow and context-awareness. Context-awareness represents context-aware description. It means that the description can express a variety of context-aware information.

Fig. 2 shows a context-aware workflow that is generated by the proposed system.

Fig. 2 shows the created workflow program by the suggested system. The workflow consists of BPEL workflow and rule-based program that has context-aware information and AOP information to apply context-awareness while holding an existing BPEL workflow. To perform context-aware function that is newly added without the modification of the existing BPEL document, we define a new document model that describes context-aware information and AOP information.

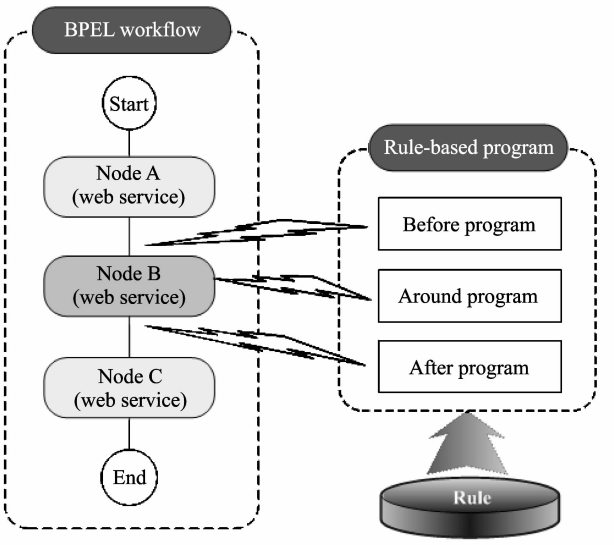

Fig. 2 Workflow that is added to context-awareness

2.1 Context-aware BPEL system architecture

We use B2J[4] and Aspect J engine[5] to implement the system because they are open sources, so that they are easy to modify to different formats. Fig. 3 shows the operating structure of BPEL system, including the processing of the new documents.

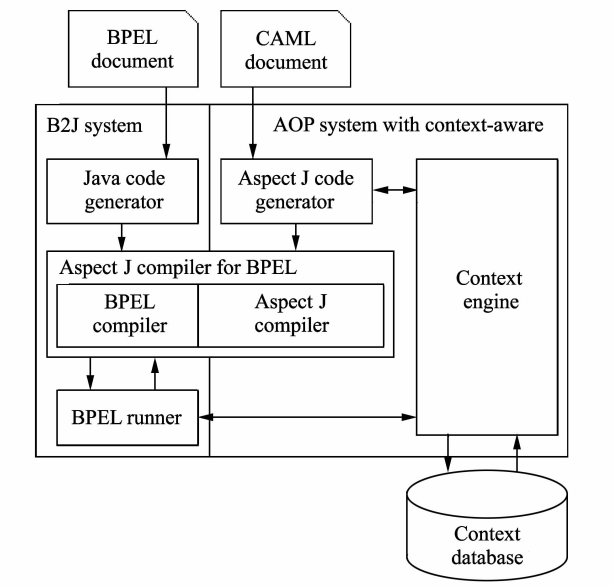

Fig. 3 Context-aware BPEL system architecture

In Fig. 3, a BPEL and a CAML document are translated into workflow Java and Aspect J code respectively, with context-aware information through each generator. And then the generated codes create a workflow program that is woven by context-aware information through the Aspect J compiler. Using the compiler module in which a general Java compiler and Aspect J compiler are combined, it can generate not only context-aware BPEL workflow, but also the general BPEL workflow.

3 CAML

We define CAML as a new mark-up language to describe context-awareness. It is translated into an Aspect J document which includes a context-aware code through an Aspect J code generator. And it is compiled with a translated BPEL document at the B2J engine. Then a workflow with context-awareness facility is finally generated. Fig. 4 shows a simple scenario for context-aware workflow.

1. A person comes in the room;
2. If the person is John, turn on the computer;
3. If the person is Annm, turn on the TV.

Fig. 4 Workflow scenario

When a person enters the room, the system identifies who the person is and operates some actions which are different for each person. In the existing BPEL workflow system, it does not proceed by the scenario that branches each service using context-aware functions such as the identification of people. If the system tries to create a workflow based on the scenario, it only generates a workflow without the context-aware function in Fig. 5.

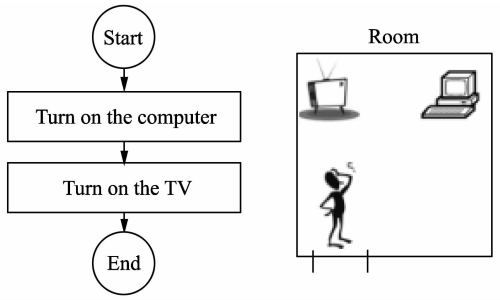

Fig. 5 BPEL workflow without context-awareness

The existing service cannot distinguish turning on the computer and turning on the TV because it cannot identify who the person is. However, the proposed system can generate context-aware workflow by reusing the existing workflow (Fig. 6).

To generate context-aware workflow represented in Fig. 6, we write a CAML document sample for context-aware as shown in Fig. 7.

Fig. 7 shows a part of a CAML document described in Fig. 6 workflow. CAML is composed of three parts: BPEL targeting part, context expression part and the operation part. The lines 2 to 4 of Fig. 7 describe BPEL targeting information. In line 3, "target" describes the application location within the document using the XPath form[6]. And "posi-

tion" describes an aspect location for weaving context information. Lines 5 to 9 are parts for context expression. To describe the context information, it is divided into three parts: "subject", "verb" and "object". In line 5, "target" describes a target of context-aware engine. Line 10 describes the operation that is a proceeding method about the context information. In line 10, "enable" is the operation to activate the target workflow node. Fig. 8 is the resulting program into which the CAML document in Fig. 7 is transformed.

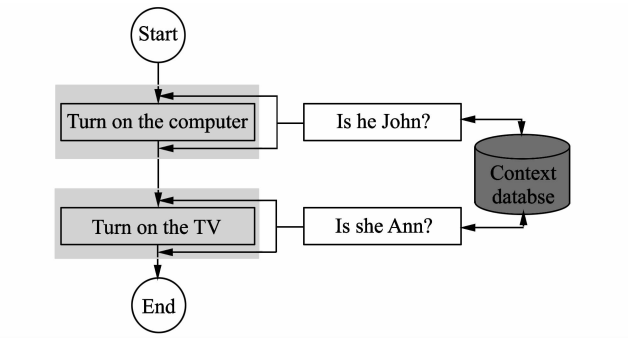

Fig. 6 Context-aware BPEL workflow

```
1  <CAML>
2     <bpel
3      target="/process/invoke[@operation='onCom']"
4      position="around">
5        <context target="/default/">
6           <subject>John</subject>
7           <verb>locatein</verb>
8           <object>Room</object>
9        </context>
10       <operation>enable</operation>
11    </bpel>
12 ...
13 </CAML>
```

Fig. 7 CAML document sample

The line 1 of Fig. 8 is imported to access the context-aware engine. Lines 2 and 3 are imported to get information about the parameter value of the BPEL document and the parameter value of the BPEL document that is currently running. Lines 4 to 17 are aspects which are activity units of aspect-oriented programming method. Line 5 is point-cut of workflow program function, which is the location that is described as location in "BPEL target" of Fig. 7. Lines 7 to 17 are advices of aspect-oriented programming method, and this means program code can be run at the point-cut location. This point-cut method includes accessing and managing target context-aware engine. Using Aspect J compiler, generated Java code about the BPEL document and generated Aspect J code about the CAML document are compiled into the context-aware workflow as shown in Fig. 6.

```
1  import bpel.context.EngineManager;
2  import org.eclipse.stp.b2j.core.…Message;
3  import org.eclipse.stp.b2j.core.…RunnerInterface;
4  aspect onComComponent {
5     pointcut point() : execution(* onCom(*));
6
7     Object around(): point() {
8        EngineManager contextEngine =
9              new EngineManager("/default/");
10
11       if(contextEngine.checkContext(
12                  "locatein", "John", "Room")) {
13          return proceed();
14       }
15       return null;
16    }
17 }
18 …
```

Fig. 8 AspectJ code for context-aware

4 Conclusion

Ubiquitous computing community has emerged as the core infrastructure. For ubiquitous computing, workflow system and context-awareness are needed. BPEL is a most recognized standard workflow language. However, BPEL does not provide context-aware facility. To apply context-aware function to BPEL workflow, Context4BPEL has expanded to BPEL syntax. Therefore, it has context-aware function, while generality and reusability of the existing BPEL documents had disappeared. For the generality and reusability of the existing BPEL documents, we propose a new BPEL workflow system that includes added context-awareness using AOP. The system can generate a context-aware workflow program without modifying the existing BPEL document because it is developed in a way that inserts context information using AOP to provide a context-aware service.

References

[1] BPEL. IBM. [2011-03-09] http://www-128.ibm.com/developerworks/library/ specification/ws-bpel/.
[2] Kiczales G, Lamping J, Mendhekar A, et al. Aspect-oriented programming. Proc. of ECOOP'97, 1997, 1241: 220-242.
[3] Wieland M, Kopp O, Nicklas D, et al. Towards context-aware work-flows. Proc. of the Workshops and Doctoral Consortium, 2007.
[4] Eclipse B2J. Eclipse Foundation, Inc. [2011-02-15] http://www.eclipse.org/stp/b2j/.
[5] Eclipse AspectJ. Eclipse Foundation, Inc. [2011-04-26] http://www.eclipse.org/aspectJ/.
[6] XPath. [2011-06-25] http://www.w3.org/TR/xpath/.