ABSTRACT – This paper proposes a context-aware workflow system that can immediately respond to a user request occurring during execution of context-aware services. Our system has been improved with new features. In order to improve the existing system, we apply it to an external event handling mechanism used in BPEL. Our system focus on handling event handler, included in the execution engine, to deal with user external events, and the event handler is used to invoke various context-aware services in response to a user request. We simulate the event handling process with the event handler of the context-aware workflow system to verify the proposed system.

Keywords: Context-Aware Workflow system, External event, User's request

1. INTRODUCTION

A workflow is a technique integrating all tasks in the management of the changing process and consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information [1,2]. There have been attempts to this technique for the ubiquitous system environments for decades after it was first proposed[3]. We have desiging CAWL, which is a context aware workflow language that expresses the situational information based on information about the surroundings of a user, and studied on a context-aware workflow system based on CAWL[4,5]. CAWL express the transition condition for executing the service as context information. Based on this function, CAWL Engine controls flow of workflow service. CAWL includes various elements that refer to transition conditions of a service for invoking it, and CAWL-based context-aware workflow system can control service flow with various patterns. The existing CAWL system provides context-aware services base on sequential flow pattern. It has a lack of responding capability to user requests that are occurred during running of CAWL system. The lacked capability of CAWL can be solved with the external event handling method of WSBPEL[6]. We apply the method to CAWL system to overcome the problem.

In this paper, we propose a context aware workflow system that can provide context aware services in response to user requests. This paper focuses on the execution process of context-aware services with event handler, which is based on an <eventHandler> tag of several sibling elements in root of CAWL schema. <eventHandler> tag can be used to describe triggering condition for handling an external event.

2. CAWL ENGINE ARCHITECTURE

Figure 1 shows the conceptual architecture for event handling procedure. The context aware workflow engine consists of CAWL parser, Workflow pool, Task manager, Context comparator and Event handler. Event handler consists of Service Requester and Context Checker. The external event processing process of the CAWL engine is as follows.

1. CAWL engine validates input documents to check they are described with proper CAWL-based syntax or not.
2. CAWL parser stores a set of services flow information in Workflow pool and the external event contents in External Event Context Table.
③ Context checker keeps searching External Event Context Table and compares context information from Context handler.

④ The Service requester asks for the service invocation from the Task Manager.

⑤ Task manager finds and executes appropriate target service flow stored in Workflow pool

Figure 1. Conceptual Architecture for Event-Handling mechanism

3. CAWL SCHEMA FOR EVENT HANDLER

Outline of CAWL syntax for handling an external event is shown in Figure 2 (a). Figure 2 (b) shows how to apply \texttt{<eventHandler>} tag. The \texttt{<eventHandler>} tag can be declared as both global element and local element. Different points are each element’s scope that each local \texttt{<eventHandler>} contents must define globally first. Service developers use those \texttt{<event>} tag under the \texttt{<eventHandler>} tag to describe a multiple external events, and uses the target attribute of the event tag to describe the name of the target service scenario. The triggering condition of external event is described under the \texttt{<event>} tag as the \texttt{<condition>} tag, including the RDF-based S-V-O element for the surrounding situation.

Figure 2. Schema for context-aware service scenario

REFERENCES


