

# Using BPEL to Define an Executable CDS Rule Process

Kevin M. Heard<sup>a</sup>, BS, Christine Huang<sup>a</sup>, MS, Laura A. Noiro<sup>b</sup>, BS, Richard M. Reichley<sup>a</sup>, RPh, and Thomas C. Bailey<sup>a,b</sup>, MD

From <sup>a</sup>BJC HealthCare and <sup>b</sup>Washington University School of Medicine, St. Louis, Missouri

## Abstract

*Business Process Execution Language for Web Services (BPEL) is an XML-based language used to define business process and workflow logic. While its original purpose was to provide a method of linking several disparate business applications, we have found that BPEL provides several features and advantages that lend it to incorporation in a clinical decision support (CDS) rule engine.*

## Introduction

An executable rule format is a key element to all CDS applications. When designing a new generic CDS rule processing application, the following factors contributed to our architecture decisions: reusability, portability, maintainability, supporting tools and utilities, separation of rules from application code, and the ability to represent or modify a rule graphically. Based on these criteria we sought an infrastructure that could be used today and in the future. The resulting infrastructure features BPEL as the underlying rule execution engine.

## Methods

We have developed a generic CDS rule and guideline processing application, which uses BPEL as the core of the rule execution engine.<sup>1</sup> A client application, called the event handler, queries the database and constructs a Virtual Medical Record (VMR) for which one or more rules have been triggered. The VMR is translated into an eXtensible Markup Language (XML) message and sent via HTTP to the BPEL engine. The BPEL engine executes the given rule and returns a list of one or more CDS actions, which indicate the result of the rule execution (e.g. alert, no action, schedule future assessment, etc.).

We have used the ActiveBPEL<sup>TM</sup> open source BPEL engine in our implementation.<sup>2</sup> The rules use XPath expression language, a W3C standard for extracting and evaluating XML data, which is standard in all BPEL engines.

## Discussion

Some of the technologies that are used today in CDS applications include Java, SQL, and Medical Logic Modules (MLM). While Java provides many advantages, it still leads to rule logic being embedded in application logic. SQL is not optimal either, as it is not object-oriented and is database dependent. Both

Java and SQL are very syntactical and therefore not easily assimilated by clinical users. MLM, while clinically oriented, is not inherently object-oriented and is geared specifically for condition-action rules.

Service Oriented Architecture (SOA) and XML provide great advantages because they offer language and platform independence, object oriented design patterns, re-usable components, and scalability. BPEL promotes these standards inherently, making it a prime candidate for our architecture. BPEL is rapidly becoming a standard in business process and workflow language. Originally developed jointly by Microsoft, IBM, BEA and SAP, BPEL allows decoupling of rule logic from database and application specific logic, such as event handlers, alerting and notification. BPEL also supports asynchronous and long-running processes, the ability to access other service providers via web services, and pluggable expression languages. This ability to use other expression languages allows us to incorporate more clinically oriented languages such as GELLO (Guideline Expression Language) once a parser becomes available.

The ActiveBPEL<sup>TM</sup> engine, as well as most commercial engines, provides a graphical user interface allowing business users to design and view the processes. Other useful features include tools to test, debug, and view the completed execution path of processes.

## Conclusions

In our development and testing of a new CDS application, BPEL has been a valuable technology for rule execution. It provides vast potential for real-time integration with other systems and the flexibility to evaluate many different types of rules.

## References

1. Business Process Execution Language for Web Services [Internet]. IBM; [updated 2005 Feb 01; cited 2006 Mar 10]. Available from: <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
2. ActiveBPEL<sup>TM</sup> [Internet]. Active BPEL, LLC; c2004-2006 [cited 10 Mar 2006]. Available from: <http://www.activebpel.org>

Supported by NHLBI Grant R01 HL70790-01A1.