uEngine: A Web service based Workflow Management System

uEngine
the universal engine for e-business

2004. 2. 26

Jinyoung Jang
Hyunse Systems, Inc., Korea
pongsor@users.sourceforge.net

Yongsun Choi
Inje University, Korea
yschoi@inje.ac.kr

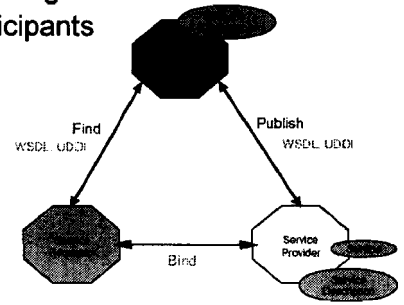
Introduction

- Emerging requirements for EC environments:
  - *At run-time:* Dynamic process composition with various networked resources, i.e., applications or participants
  - *At build & maintenance time:* Agile technical enhancements, e.g., evolution in standards
- Open issues of WfMS (*Bussler, 1999; Sayal et al., 2002*):
  - Heterogeneity & interoperability
  - Adaptability and flexibility

- Features of Web Services

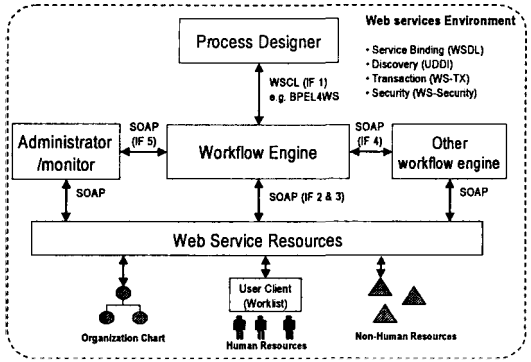
- Information exchange in the interoperable way
- Dynamic service binding & advertisement
- Wide variety of participants
- Multiple suppliers for single service

(*WinterGreen Research, 2002*)



Web Services roles, operations & artifacts (*IBM, 2001*)

- WfMS with Web Services



Web Services-based Workflow Interface Model (*Jang et al., 2004*)

- JavaBeans Framework

- A *framework* is a set of collaborating classes that fosters the separate and adaptive evolution of a system by the plug-and-play style CBD (Johnson, 1997)
- *JavaBeans* (Sun microsystems, 1996) is one of most wide-spread component model that prescribes standard programming conventions and dynamic discovery mechanisms
- *uEngine* employs *JavaBeans Framework* to enhance adaptibility and flexibility in workflow management

uEngine: WS-based WfMS

- *uEngine* is based on the dynamic and flexible property of *Web Services* and the CBD properties of *JavaBeans frameworks*
- Main Features
  - XML based process variables
  - Dynamic web service activities
  - Universal resource management
  - Support for multiple definition languages
  - JavaBean-styled activity components

- XML based Process Variables

- *Interoperable information exchange* with the *XML-type system* for process variable declaration and assignment
- Dynamic generation of XML binding classes to manipulate business documents with XGen (*CommerceOne, 2004*)
- Provides a basis for the *universal resource management*

- Dynamic web service activities

- Web service related Activity types
  - discovery, role binding, invocation, advertisement, etc.
- Supported by
  - Apache Axis (*Apache, 2004*) in invoking Web services or publishing business processes
  - UDDI4J (*IBM, 2004*) in searching and advertising Web services
- Allows dynamic composition of networked applications for a sophisticated internet business

- Universal resource management

- Universally accessible resource management through the simple and comprehensive resource addressing
  - Non-human IT resources: Endpoint URI's
  - Human resources: E-mail addresses or Work Queue Services *(Jang and Choi, 2003)*
- *Universal Resource Binder (URB)*
  - A tool for managing and binding resources

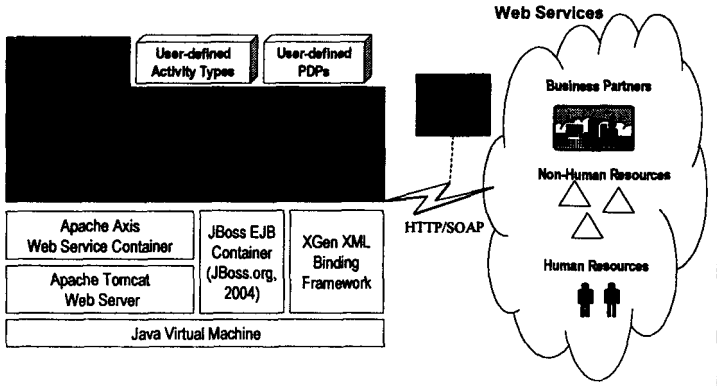
- Support for multiple definition languages

- *Process Definition Publishers (PDP)* lets partners' system understand the process definitions that uEngine hosts
  - In BPEL4WS, WSCI, XPDL, etc.
- Extensible interface of *PDP* allows the adaptation of *new* languages
- May let partners *monitor* process instances with their own system

- JavaBean-styled activity components

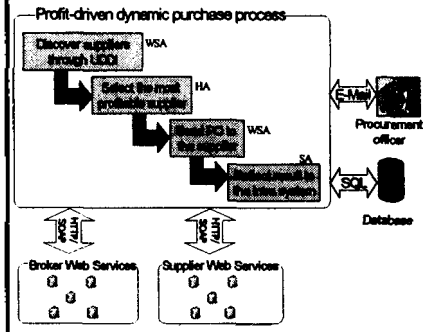
- Component declaration convention of defining property, interfaces, and protocols,
- Support for foundation functions are left to the *uEngine Workflow Framework*
  - e.g. instance data persistence, execution, monitoring, design time GUI, verification, simulation, etc
- Allows easy addition of new activity components without considering the issues of design, instantiation, execution, and monitoring time

Implemented Architecture



Stepwise illustration

Example process



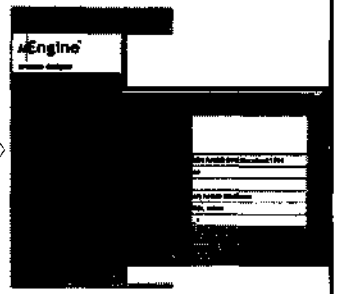
Illustrated features

- Addition of new function of interfacing databases
- Dynamic web service activities for supplier selection
- Universal role binding for resources
- Allows partners to view the process definition
- Human intervention in a process

1) Addition of newly required activity types

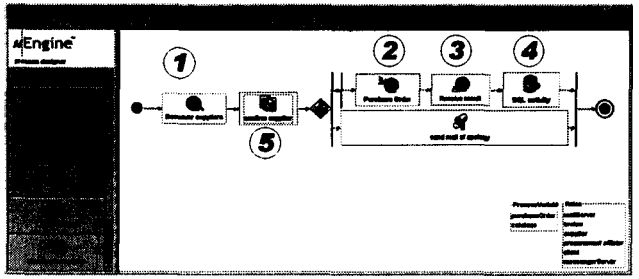
- With JavaBean-styled activity type declaration only

```
public class SQLActivity extends DefaultActivity{
    /* Property declaration */
    String connectionString;
        public String getConnectionString(){
                return connectionString;
        }
        public void setConnectionString(String value){
                connectionString = value;
        }
    String userId;
    String password;
    ...
    /* Logic declaration */
    public void executeActivity(ActivityInstance instance) {...}
    /* Instance data claration */
     public ActivityInstance createInstance(ActivityInstance
    instance) {... }
    /* Status generation logic */
    public Map getActivityDetails(ActivityInstance inst){...}
}
```



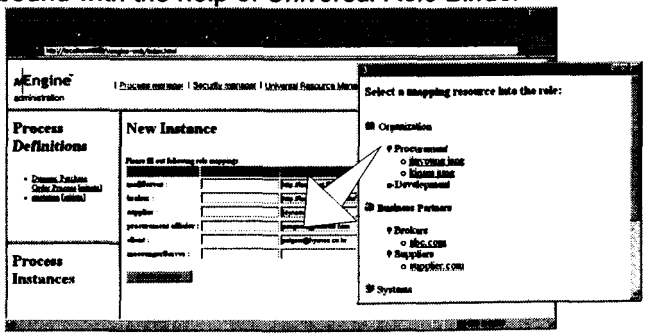
2) Process design and deployment

- Activity types shown in the example process are
  - Dynamic WS Activities, like *DiscoverActivity* (1), *InvokeActivity* (2), *ReceiveActivity* (3),
  - *SQLActivity* (4) added in the previous step,
  - *HumanActivity (5)* for human intervention.



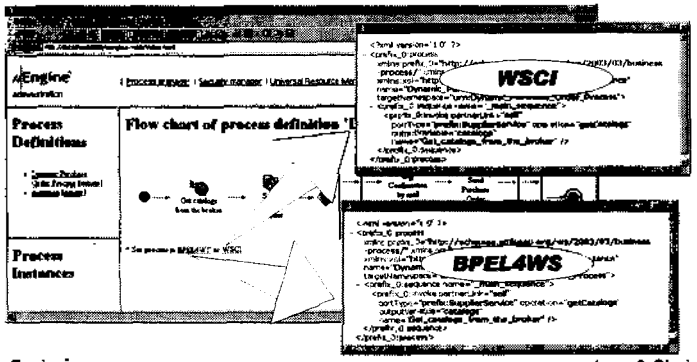
3) Initiation with role binding

- When to initiate, endpoints of appropriate web services or e-mail addresses of human workers are bound with the help of *Universal Role Binder*



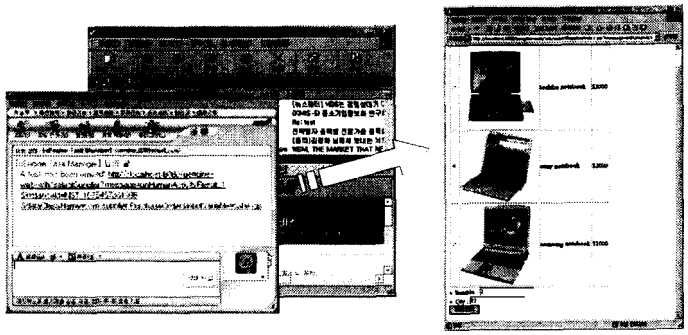
4) Publishing process definitions

- Business partners can view the process definition through the Web



5) Human intervention

- Each human participant is notified of her tasks via e-mail (or instant messenger) or worklist handler



Conclusion

- *uEngine* is a Workflow Management System that allows the dynamic composition of business resources as well as the adaptive functional enhancements with Web Service architecture and Object technology
- *Notable features*: XML type-system, Dynamic WS activities, Universal resource management, Process Definition Publisher, JavaBean-styled activity components
- Developed as an open source software (www.uEngine.org) on the *open architecture of J2EE* and *open source software* (JBoss/Apache Axis)
- Currently working on *Enterprise application builder* to support *small programming*

References

- Apache Software Foundation. (2004). *Apache Axis project*. http://ws.apache.org/axis/
- Commerce One Operations, Inc. (2003). *XGen: XML Schema Code Generation*. http://www.commerceone.com/developers/docsoapxdk/xgen.html
- Bussler, C. (1999). Enterprise-wide workflow management. *IEEE Concurrency, 7*(3), 32-43.
- IBM Corp. (2000). *Web Services Conceptual Architecture (WSCA 1.0)*. http://www.ibm.com
- Jang, J., & Choi, Y. (2003). Web service based universal resource management framework in workflow. *Technical Report SME-TR-2003-02*. Inje University, Korea.
- Jang, J., Choi, Y., & Zhao, J. L. (2004). An Extensible Workflow Architecture through Web Services. *International Journal of Web Services Research 2(1), 1-15*.
- JBoss group, LLC. (2004). *JBoss Application Server*. http://www.jboss.org
- Johnson, R.E. (1997). Frameworks = (components + patterns). *Communications of the ACM, 40*(10), 39-42.
- Sayal, M., Casati, F., Dayal, U., & Shan, M.-C. (2002). Integrating workflow management systems with B2B interaction standards. *The International Conference on Data Engineering*, 287-296.
- Sun microsystems. (1996). *JavaBeans API specification (1.01)*. http://java.sun.com
- International Business Machines Corp. (2004). *UDDI4J project*. http://www.uddi4j.org
- WinterGreen Research, Inc. (2002). *Web Services Markets: Market Strategies, Opportunities, and Forecasts 2002-2007*. http://www.wintergreenresearch.com

www.uEngine.org