IDIOM

BPEL4WS and IDIOM

Overview and Background

One of the key features of the IDIOM Decision Suite is the flexibility in the runtime deployment of the IDIOM decisions and formulas. Deploying IDIOM within a business process managed by BPEL4WS offers further flexibility to the deployment of business rules. IDIOM has been designed from its inception to work in an XML environment and is therefore easily integrated into a web services architecture. By recognising that most business processes have steps that are predominantly rule based (often decisions impact the flow of a business process i.e. approval etc), IDIOM is well placed to contribute to a BPEL4WS managed business process. This document seeks to provide some context and discussion on the approach to integrating IDIOM into a BPEL4WS managed process and details how IDIOM can be deployed as a web service itself.

Business Process Execution Language

The new Business Process Execution Language for Web Services (BPEL4WS), WS-Transaction, and WS-Coordination specifications provides a comprehensive business process automation framework that allows companies to leverage the power and benefits of the Web Services Architecture to create and automate business transactions[1]. The specification defines a mechanism with which a business process can be defined where the components of this business process are web services. The BPEL4WS description language expressed as XML allows collaborations and flows to be defined that express a business process; this process can then be deployed to an execution environment that supports BPEL4WS. The expressive flow language defined by BPEL4WS uses the constructs defined within WSDL and SOAP and extends these to define constructs to support flow activities, message correlation, process instantiation, fault handling, compensation handling and the management of flow dependencies (synchronisation).

BPEL4WS Context

For the purposes of this discussion we have classified some of the services that we believe are important when implementing web services and a business process in BPEL4WS. These classifications help to provide a context for the role and position of the IDIOM Decision Suite in the remainder of this document. Table 1 (overleaf) provides a summary of the classification of web services that we use and the diagram in Figure 1 (overleaf) shows the relationship between these different types of services and the BPEL execution engine.

We see these classifications as important when conceptually defining a collaboration of web services.

The remainder of this document will discuss the role of the IDIOM Decision Suite in providing Decision Services in a BPEL business process execution environment. It is assumed the other services will be built and integrated into a business process using web services development tools (e.g. Glue, Java web services development kit, Eclipse, Visual Studio) and will depend heavily on the technical infrastructure available (i.e. databases, operating systems).

IDIOM as a web service

In an associated discussion paper titled `Web Services and IDIOM', we presented several approaches to implementing IDIOM as a web service. In summary, we presented three alternatives to implement IDIOM as a web service:

- ❑   IDIOM standard web service
- ❑   Document centric web service
- ❑   Type safe message-based web service

The current specification of BPEL4WS is not clear on its support for attachments to the SOAP messages, a feature that is required by the document centric web service approach. WSDL currently provides no mechanism to describe attachments as part of a web service interaction. When invoking a web service directly, the requestor and the service can agree on the presence of a document attachment. While not being described in the WSDL it can be populated and interfaced with directly through SOAP. The insertion of BPEL4WS as an intermediary processor between web services may complicate the support for document attachments. Since WSDL does not describe attachments, and BPEL4WS relies heavily on the WSDL and associated message and container definitions, it is possible that an attachment would not be propagated to collaborating web services. It is assumed that the BPEL4WS container will be responsible for constructing web services interactions (SOAP messages) that are formed by the creation and manipulation of containers and service definitions.

Based on the discussion in the previous paragraph, it has been assumed that for IDIOM to participate in a BPEL4WS managed business process, IDIOM components will need to be implemented as a type safe message based web service.

---

[1]   Referenced from web article http://www-106.ibm.com/developerworks/webservices/library/ws-autobp/

[2]   Please refer to our website http://www.idiomsoftware.com

Table 1   Web Service Classification

| Service Type	Description
Requesting	Initiator of a business process e.g. an application or another web service acting as a client
Decision	Services that make a business decision based on information provided
Storage	Services that integrate with storage systems for the purposes of saving data
Action	Services that perform some business action (e.g. credit card processing)
Query	Services that perform bulk queries for read-only, non-transactional data
Integration	Services that provide an interface to other processes or systems (e.g. SAP or legacy systems)
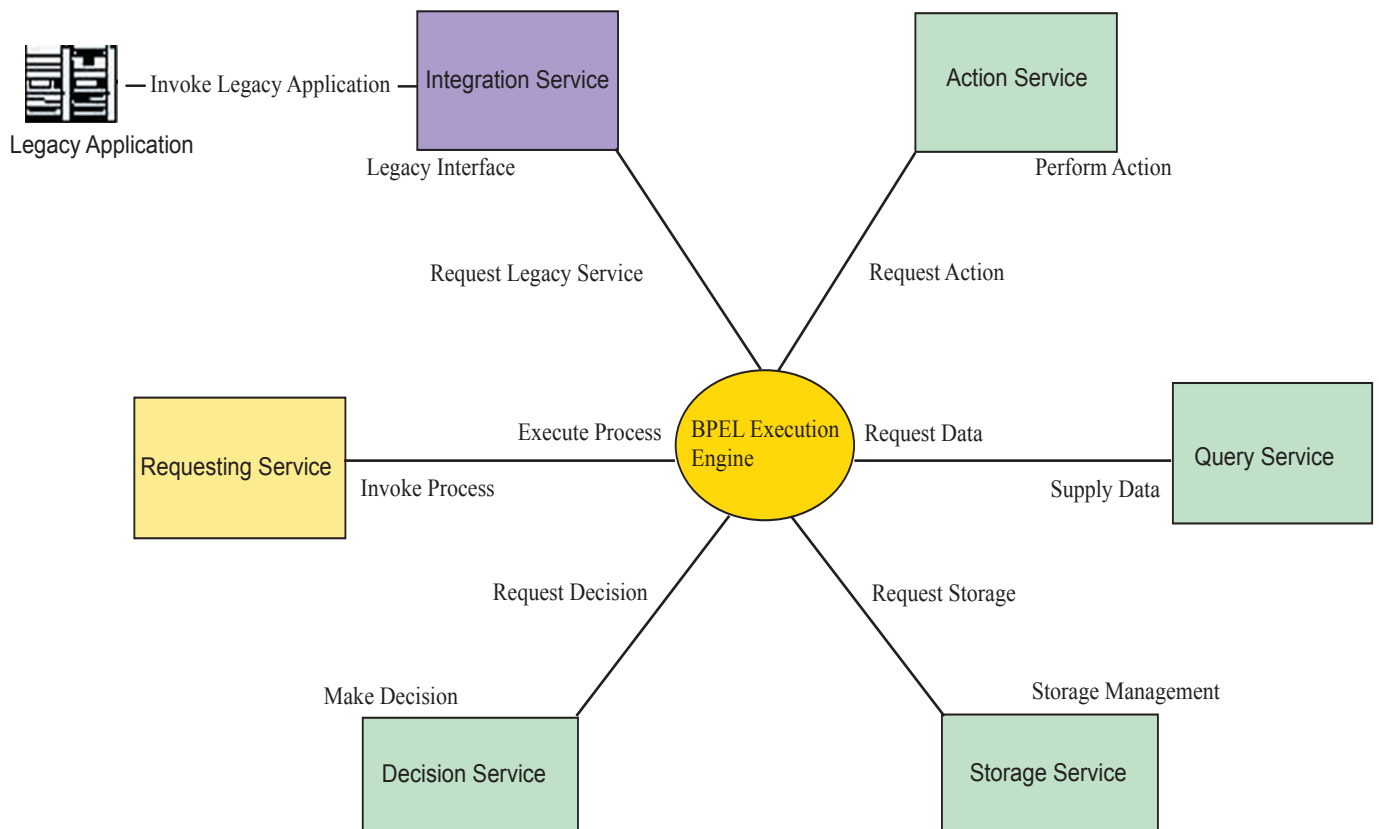
Figure 1   Relationship between services in a BPEL scenario

IDIOM within a BPEL4WS business process

To investigate the use of IDIOM in a managed business process we can consider its role in the example scenario documented by IBM in the following web article

http://www-106.ibm.com/developerworks/webservices/library/ws-bpelcol2/

In this scenario, there is a single collaborating web service managed by a BPEL4WS process. This web service implements the loan approval business rules and is called as part of the managed process (note: a more complex example of this process is packaged with the IBM BPWS4J implementation, which includes two web services, an assessor and an approver). In this scenario, the assessment criteria are trivial and can be documented as follows:

```
<message name="creditInformationMessage">
    <part name="firstName"  type="xsd: string"/>
    <part name="name"  type="xsd: string"/>
    <part name="amount"  type="xsd: integer"/>
</message>
```

A web service implemented with IDIOM may define the messages as follows

```
...
xmlns:idm="http://www.example.com/services/idiom/domain"
<import namespace="http://www.example.com/services/idiom"
location="http://www.example.com/services/domain/DomainDef.wdsl/>
.....
<message name="creditInformationMessage">
    <part name="executionDate"  type="xsd:date"/>
    <part name="customerDetails"  type="idm:CustomerDetails"/>
    <part name="loanDetails"  type="idm:LoanDetails"/>
</message>
```

The main difference in the message format is the use of the CustomerDetails and LoanDetails definitions from the DomainDef.wsdl definitions. These objects would be used during the definition of rules within IDIOM. The only other difference in the format of the message is the execution date; all IDIOM decisions are subject to an effective date (this allows for the easy upgrading of rules and also the execution of historical rules). The web service therefore needs to be told what effective date should be applied to the decisions that IDIOM will execute, otherwise the current date will be defaulted.

The remainder of the BPEL4WS definition remains largely the same, the only difference being the structure of the approval message, which would be the same as the creditInformationMessage as a result of the IDIOM processing model. IDIOM modifies the state of documents as a consequence of the execution of rules; in the case above, the LoanDetails would be modeled to contain the approval status, which would be set appropriately as a result of the execution of IDIOM. The changes to the scenario that are required to manage IDIOM are summarised below.

```
...
<message name="creditInformationMessage">
    <part name="executionDate"  type="xsd:date"/>
    <part name="customerDetails"  type="idm:CustomerDetails"/>
    <part name="loanDetails"  type="idm:LoanDetails"?>
</message>

<containers>
    <container name="request"
            messageType="loandef:CreditInformationMessage"/>
    <container name="approvalInfo"
            messageType="CreditInformationMessage"/>
</containers>
...
```

Conclusion

In this document we have seen how IDIOM can participate in a managed business process defined in BPEL4WS (by way of example). This discussion extends the documented discussion regarding the deployment of IDIOM as a web service. In earlier sections of this document, we discussed the role of IDIOM in providing decision-based services within a business process. We believe that there is a very close relationship between business decisions/rules and business process; this relationship starts during business process modeling (as the business process provides the context within which the business rules/decisions can be expressed) and finishes in the implementation of a business process. By considering BPEL4WS as a 'process execution architecture' (specifically targeted at the collaboration of web services), we can see a concrete example of how IDIOM can be used to augment a business process with rules/decisions. IDIOM in this instance is providing a business decision service within a business process; as such, it has a role to play during the analysis and design of the business process and in the implementation of that process.

About BPEL4WS

For further information regarding BPEL4WS please refer to the following web resources

http://www-106.ibm.com/developerworks/library/ws-bpel/
http://www-106.ibm.com/developerworks/webservices/library/ws-autobp

About IDIOM Decision Manager

For further information regarding IDIOM Decision Manager please refer to

http://www.idiomsoftware.com