WORKFLOW

Workflow Interoperability Standards for the Internet

JAMES G. HAYES
Computer Sciences Corporation
EFFAT PEYROVIAN
ECC Consultants
SUNIL SARIN
TIBCO Software
MARC-THOMAS SCHMIDT
IBM UK Ltd.
KEITH D. SWENSON
MS2 Inc.
RAINER WEBER
SAP AG

Business-to-business integration is transforming the market and has already begun to increase the efficiency of those companies involved. The first step, integration of data, began with Electronic Data Interchange (EDI) in the 1980s. The second step, integration of process information, is only now beginning to emerge.

Consider a fictitious company, Acme Manufacturing, that ships its products via Federal Express. In the past, Acme used a dedicated computer with proprietary software to arrange pickups directly from Federal Express. Today Acme can use the Federal Express website to arrange for shipment and to follow a package's progress in real time. Acme can check whether the package has been delivered to the station, transported to the destination city, delivered to the addressee, or signed for at its final destination. By making process information available through the Web, Federal Express does not need a support department to give this information over the phone.

If Acme wants to order a computer from Dell Computer Corporation, it can do so through the Dell website. Acme can also track the order's progress through five stages: order processing, preproduction, production, delivery prep, and shipping. Again, making this simplified external view of their internal order fulfillment processes available over the Web saves Dell the expense of answering status questions over the phone. Say Dell subcontracts delivery to Airborne Express, a firm with a similar Web interface. Once the Dell process reaches the *shipped* state, users can follow a link directly to the Airborne website, which gives additional detail about the shipment status.

ENABLING E-BUSINESS WORKFLOW

Many companies (such as the examples cited above) have shown that providing process status in a user-readable form over the Web is valuable to business. There is an additional value to be gained from providing that information in a machine-readable form.

Efforts toward defining a standard for workflow interoperability began in 1994 with the Workflow Reference Model from the Workflow Management Coalition (WfMC). From that model other standardization efforts have evolved—from OMG's jointFlow specification to the Simple Workflow Access Protocol. The WfMC's Wf-XML focuses on a simple subset of SWAP for a first version of a standard, with the aim of future extension.

A standard to directly link workflow systems across companies would enable several things. First, it would let Acme configure its purchase order workflow to automatically start the order process at Dell. Second, when the order fulfillment process is done, Dell would send a standard notification back to Acme, allowing the process to advance to the next step, which is to receive the shipment. Finally, at any point during the two processes, Acme employees would be able to follow a link from their internal workflow system to the process at Dell.

This is what we call *Internet-scale* workflow. There is no practical limit to how many different companies and processes can be linked because each server supports only a small part of the overall process, referring to other parts through Uniform Resource Identifier-(URI-)style links. Although many companies have linked their information systems for business processes, the interfaces are nonstandardized and require custom programming. The Wf-XML message set and protocol currently being defined by the Workflow Management Coalition (WfMC) attempt to provide a simple protocol that enables interaction between requesters and providers of workflow-type services.

EVOLUTION OF WORKFLOW STANDARDS

The WfMC, an organization of vendors, users, consultants, and researchers, has become the primary standards body in the workflow management system arena. The WfMC's Workflow Reference Model, defined in 1994, describes the architecture of workflow-based applications and workflow management systems.[1] Another article outlines the WfMC's standardization program.[2]

Workflow Interoperability Specification

The WfMC Workflow Interoperability standard defines an abstract protocol for peer-to-peer interaction of workflow enactment services, potentially across business domain boundaries.[3] The standard supports a scenario where an activity in a workflow process hosted by an enactment service is realized by a subprocess hosted by another enactment service. A typical example is an interenterprise workflow, such as those found in supply-chain interactions. The standard provides interoperability on the workflow-process level only; internals of a process instance need not be known across business domain boundaries.

A workflow process has a set of attributes to define the operating context of the workflow process, a set of attributes to represent results produced during the process's lifetime, and a property to represent the current state of the process. The standard defines a minimal state model that includes the states *running*, *completed*, and *terminated*.

The standard defines the content of *requests* sent by a service requester to a service provider to control and monitor the enactment of the remote process. These requests include

- *Create ProcessInstance requests*, which create new instances of workflow processes to be controlled by the service provider;
- *Get* or *Change ProcessInstance State requests*, which obtain or change the current state of the remote process; and
- *Get* or *Set ProcessInstance Attribute requests*, which provide access to the values of a process's context or result data attributes.

The service provider answers each request with a matching *response*. The service provider can also send *notifications* to the service requester:

- *ProcessInstance State Changed notifications* to indicate that the state of the remote process was changed due to some event in the service provider's domain.
- *ProcessInstance Attributes Changed notifications* to indicate that the context or result data of the remote process was changed.

The Coalition has defined a *binding* of the standard, which uses asynchronous interaction via e-mail as the transport, with MIME encoding of the information to be exchanged. This MIME binding of the standard[4] specifies the protocol for exchanging interoperability requests and responses via e-mail, taking into account e-mail's unreliability as a communication vehicle.

Several workflow management system vendors have applied the MIME-interoperability standard in prototype workflow engines. In addition, the WfMC has defined a set of test cases for proving conformance with the standard. Work to establish an independent conformance test laboratory (for this and other WfMC standards such as Wf-XML) is ongoing at the University of Muenster in Germany.

OMG Workflow Management Facility Standard

The OMG Workflow Management Facility standard (also known as the jointFlow specification) is based on WfMC standards.[5,6] The specification defines a framework for distributed workflow applications in the world of business objects. It enables interoperability of workflow process components, monitoring of workflow execution, and association of workflow components to resources involved in a workflow process. The basic model features a workflow service requester and a workflow service provider. Several primary interfaces are relevant to our discussion.

- *WfRequester* requests work to be done. It can register its interest in a workflow and provide operations to be used by the process to propagate status updates.
- *WfProcess* performs work requests, often by delegating them to other entities. It provides operations to control execution of the work request and to observe its state.
- *WfProcessMgr* represents a workflow process definition and serves as a factory for workflow processes, allowing a WfRequester to register with a WfProcess when it is created.
- *WfActivity* provides an adapter for integrating existing business objects into a workflow process. In conjunction with the WfRequester interface, it also allows for interaction between a main workflow (containing the workflow activity) and another workflow application (implementing the task defined by the workflow).

The standard also defines interfaces for worklist handling (WfAssignment and WfResource) and process auditing (WfEventAudit). It does not attempt to standardize workflow process definitions; however, a request for proposal (RFP) for extensions to the Workflow Management Facility to standardize these definitions is in development.

The OMG workflow standard defines a unified object model covering the various WfMC standards (except process definition). This object model provides the base for future evolution of the WfMC standards. The simple workflow access protocol (SWAP) initiative and the Wf-XML message set are examples of this evolution. Figure 1 illustrates how the workflow systems of the three companies in our example scenario could be integrated using these standards.

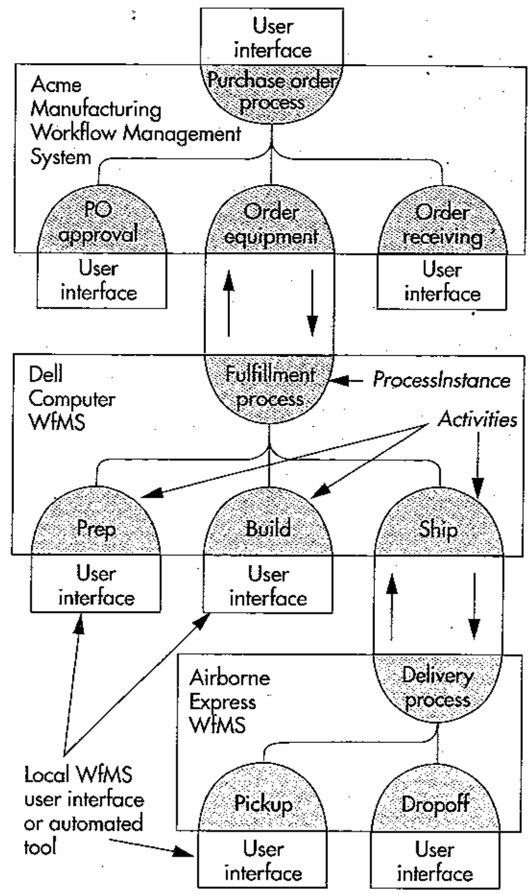

Figure 1. The workflow processes of Acme, Dell, and Airborne integrated by workflow interoperability standards.

Simple Workflow Access Protocol

The SWAP proposal attempts to define an Internet-based workflow access protocol to instantiate, control, and monitor workflow process instances.[7] SWAP was envisioned as a binding of the jointFlow object model and related WfMC standards to an HTTP-based interaction protocol. The basic idea of the SWAP proposal—rendering interaction between components of workflow applications as XML-encoded messages—provides an excellent base for adapting the WfMC standards in the area of message-based enterprise application integration.

The SWAP object model defines four types of Internet resources: Observers, ProcessInstances, ProcessDefinitions, and Activities. These correspond to the WfRequester, WfProcess, WfProcessMgr, and WfActivity interfaces of the OMG standard.

A workflow system represents a process instance as a collection of tasks. Conceptually there are two parts of a task (shown as halves of a circle in Figure 1): the upper part, representing the request for and description of the task; and the lower part, representing the fulfillment of the task.

In a workflow system coordinating human work, the system describes the request for the task, while a person performs the task. For automated tasks, the workflow system makes a request for a task to an external program. Invoking a subprocess on another workflow engine is like invoking a task on an external program—the fact that accomplishing a task involves a set of finer grained tasks is not an issue at the upper level. Therefore, to achieve workflow and tool integration on the Internet, SWAP defines a protocol such that the two parts of the task can be at different Internet locations.

All SWAP interactions involve HTTP request/response pairs with XML encoding of the information in both directions. The proposal builds on the WebDAV extensions to HTTP to communicate workflow operation requests.[8]

The SWAP proposal was presented to the Internet Engineering Task Force at the 43rd IETF meeting in December 1998. SWAP was successfully used to process vendor information requests in a prototype linking the U.S. Department of Defense Joint Computer-aided Acquisition and Logistics Support program (JCALS) workflow engine and the IBM MQSeries workflow engine in different areas of the General Dynamics Electric Boat Division offices.[9]

Other vendors have also implemented SWAP. Netscape implemented a preliminary version in its Process Manager. If requested using a particular HTTP method, Process Manager returns an XML structure describing the process's state. Fujitsu implemented a version of SWAP in an EDI-based product. This product extended the basic process-to-process interaction to include asynchronous creation of process instances, a batch request to create processes, and a "follow-on" process specifier for chaining processes.

Wf-XML SPECIFICATION

The lessons learned from early SWAP implementations led to the initial submission of the Wf-XML specification to the WfMC in April 1999. The specification represents the next step in the evolution of workflow interoperability standards.

The Wf-XML specification enhances some of its predecessors' capabilities, providing

- a structured and well-formed XML body protocol that consists of message structures containing headers and data;
- synchronous or asynchronous message-handling capability;
- independence from transport mechanisms; and
- easy extensibility through the use of XML and dynamic workflow context data.

For interoperation between systems using Wf-XML, an *interoperability contract* (which is outside the scope of the specification) must define each system's expectations and requirements. Typically, the contract addresses issues of connectivity, security, and the context and result data format associated with a specific workflow process definition, as well as the process definitions whose invocation can be requested.

Logical Resource Model

The concept of interoperability among workflow systems can be extended to include interactions among other types of systems and services. These systems are considered generic services that consist of a number of different resources. The resources can be implemented in any fashion, so long as they are identifiable and can interact with other resources uniformly, receiving requests for services and returning appropriate responses.

Individual interoperable functions are termed *operations*. Each operation receives a set of request parameters and returns a set of response parameters. Operations are divided into groups based on their context. As in SWAP, the primary groups of operations required for interoperability are ProcessDefinition, ProcessInstance, and Observer.

- A *ProcessDefinition* creates an instance of a service. It describes the basic functions of the service and is uniquely identifiable by interoperating services or service requesters.
- A *ProcessInstance* represents the actual enactment of a given ProcessDefinition and has its own resource identifier. Each ProcessInstance is unique and exists only once. Once created, a process instance is started and eventually is completed or terminated.
- An *Observer* allows a process to communicate status changes such as its completion or termination.

In a typical interaction, the requester, or Observer, invokes the CreateProcessInstance function on a ProcessDefinition resource, which returns a handle for a new ProcessInstance resource. During execution of the process, the Observer can make requests to GetProcessInstanceData on the ProcessInstance, or can invoke ChangeProcessInstanceState to interrupt or terminate the execution. The ProcessInstance invokes the ProcessInstanceStateChanged notification to inform the Observer of any changes of state (including process completion) and to pass result data. The interaction among resources via these operations is shown in Figure 2.

Definitions and example XML message content for these operations can be found in the sidebar "Wf-XML Operations" on the following pages.

Wf-XML Message Set

Every Wf-XML message is an XML document instance, conforming to the XML 1.0 specification. The root element is the WfMessage tag, which can contain three elements: WfTransport, WfMessageHeader, and WfMessageBody.

WfTransport contains transport-specific and user-defined correlation information, which is returned without modification in the response messages. This element is needed to support asynchronous messaging but is not required with synchronous messaging.

WfMessageHeader contains tags useful to all operations (such as whether the message is a request or a response) and contains the following elements:

- Request—a request message.
- ResponseRequired—attribute defined as enumerated type {No | IfError | Yes}.
- Response—a response message.
- Key—identifies the resource that is the target of a request or source of a response.
- ResponseLang—an optional attribute indicating the language being used (such as English) in language-specific data such as subject or description. This indicator is based on ISO Standard 639.

WfMessageBody provides operation-specific data. For a request message, it contains the OperationName.Request element, which defines request parameters. For response messages it contains the OperationName.Response element, which holds potential response parameters or an exception returned by an operation.

Wf-XML has some built-in data types for describing parameters. In particular, Wf-XML enforces representation of all date and time values as Coordinated Universal Time-(UTC-)based timestamps to ensure interoperability between workflow engines in different time zones. Instance-specific data types can be binary, Boolean, integer, unsigned, float, double, string, date, URI, or XML.

When a process definition is instantiated, the context of the resulting process instance is initialized using the ContextData element. The ResultData element represents the output parameters of a process instance.

The Wf-XML specification leaves the precise syntax of the context data and result data elements to the parties involved in workflow interoperation. The standards described in the sidebar, "E-Business Standardization Efforts," are example specifications that could be used within the ContextData and ResultData sections. WfMC will synchronize with and use standards such as these as they mature.

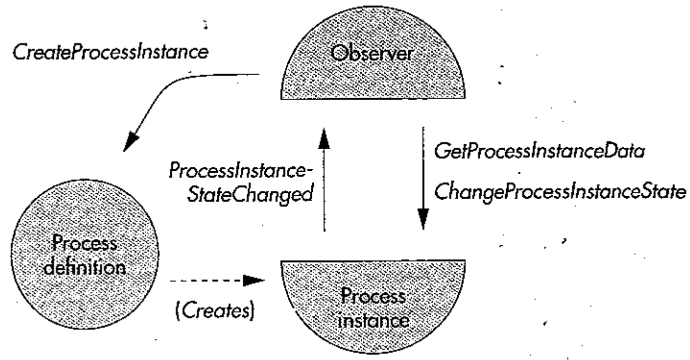

Figure 2. Wf-XML logical resources and their operations: ProcessDefinition, which creates an instance of a service; ProcessInstance, the actual enactment of a ProcessDefinition; and Observer, which allows a process to communicate status changes.

Actual business use of Wf-XML will require security mechanisms to protect the data from snooping and to prevent unauthorized manipulation of the processes. The exact mechanism for safeguarding Wf-XML interactions will depend on the transport chosen and will be specified as part of the interoperability contract between requester and service. In HTTP-style interactions, secure sockets layer (SSL) is sufficiently strong for most uses.

Transport Bindings

Although the Wf-XML specification defines the XML data content required to communicate effectively between workflow engines, the method of transport for these messages (for example, HTTP, e-mail, CORBA, or IIOP) is left to the solution providers.

Because of the ubiquitous nature of HTTP, the WfMC has specified HTTP as a core transport mechanism for Wf-XML messages. When using HTTP, the communicating workflow engines are considered HTTP servers. The Wf-XML messages for all operations are integrated as input or output data with respect to HTTP interactions. Specifically, an operation is encoded in the HTTP method Post. For Wf-XML, Post is directed to some URI and uses exactly one MIME body part for input and exactly one for output.

- The resource key is the URI to which a Post method is directed.
- The Wf-XML request is the request message body for input.

Wf-XML Operations

The following operations are defined in the current version of the Wf-XML interoperability specification[1]:

CreateProcessInstance, in the ProcessDefinition group, is used to instantiate a known process definition. The instance will be created with a context-specific data set according to the input data, and started.

Consider a company Acme Manufacturing ordering a computer through another company, Dell Computer Corporation. When the Acme Manufacturing workflow engine generates a message to the Dell workflow engine, it starts a known process. The message is encoded in XML using the CreateProcessInstance operation as indicated below.

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
  <WfTransport/>
  <WfMessageHeader>
    <Request ResponseRequired ="Yes">
    </Request>
    <Key>http://www.dell.com/Wfengine?id=1199827
    </Key>
  </WfMessageHeader>
  <WfMessageBody>
    <CreateProcessInstance.Request StartImmediately
    ="true">
      <ObserverKey>http://www.Acme.com/wfx456
      </ObserverKey>
      <ContextData>
        <Computer>
          <Type>notebook</Type>
          <Series>Inspiron</Series>
          <Number>7500<Number>
          <Option>DVD</Option>
        </Computer>
      </ContextData>
    </CreateProcessInstance.Request>
  </WfMessageBody>
</WfMessage>
```

If the CreateProcessInstance operation is successful, then a resource identifier for the new process instance will be returned as the response parameter. This response parameter is called **ProcessInstanceKey**. The response message will be encoded as follows.

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
  <WfTransport/>
  <WfMessageHeader>
    <Response/>
    <Key>http://www.dell.com/Wfengine?id=1199827
    </Key>
  </WfMessageHeader>
  <WfMessageBody>
    <CreateProcessInstance.Response>
      <ProcessInstanceKey>
      http://www.dell.com/WfcXML1199
      </ProcessInstanceKey>
    </CreateProcessInstance.Response>
  </WfMessageBody>
</WfMessage>
```

If there is an error, an exception message is returned. For example, if the resource does not exist, the exception message encoded will be

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
  <WfTransport/>
  <WfMessageHeader>
    <Response/>
    <Key>http://www.dell.com/WfEngine?id=1199827
    </Key>
  </WfMessageHeader>
  <WfMessageBody>
    <CreateProcessInstance.Response>
      <Exception>
        <MainCode>502</MainCode>
        <Type>F</Type>
        <Subject>Invalid Process Definition</Subject>
        <Description>Cannot create instance
        </Description>
      </Exception>
    </CreateProcessInstance.Response>
  </WfMessageBody>
</WfMessage>
```

The **GetProcessInstanceData** operation is used to retrieve the values of properties defined for the given process instance resource. To check the priority data element of the process instance in the scenario above, the following request message will be encoded in XML.

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
  <WfTransport/>
  <WfMessageHeader>
    <Request ResponseRequired ="Yes">
    </Request>
    <Key>http://www.dell.com/WfcXML1199</Key>
  </WfMessageHeader>
```

```
    <WfMessageBody>
        <GetProcessInstanceData.Request>
            <ResultDataAttributes>
                <Priority/>
            </ResultDataAttributes>
        </GetProcessInstanceData.Request>
    </WfMessageBody>
</WfMessage>
```

The response message returned is

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
    <WfTransport/>
    <WfMessageHeader>
        <Response/>
        <Key> http://www.dell.com/WfcXML1199</Key>
    </WfMessageHeader>
    <WfMessageBody>
        <GetProcessInstanceData.Response>
            <Priority>5</Priority>
        </GetProcessInstanceData.Response>
    </WfMessageBody>
</WfMessage>
```

The **ChangeProcessInstanceState** operation modifies the status of a given process instance. For example, a user may decide to terminate the process before it is completed. This means changing the state of the process instance to closed.abnormalCompleted.terminated as follows.

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
    <WfTransport/>
    <WfMessageHeader>
        <Request ResponseRequired ="Yes">
        </Request>
        <Key>http://www.dell.com/WfcXML1199</Key>
    </WfMessageHeader>
    <WfMessageBody>
        <ChangeProcessInstanceState.Request>
            <State>closed.abnormalCompleted.terminated
            </State>
        </ChangeProcessInstanceState.Request>
    </WfMessageBody>
</WfMessage>
```

The response message will return the resulting state of the process instance as follows.

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
    <WfTransport/>
    <WfMessageHeader>
        <Response/>
        <Key>http://www.dell.com/WfcXML1199</Key>
    </WfMessageHeader>
    <WfMessageBody>
        <ChangeProcessInstanceState.Response>
            <State>closed.abnormalCompleted.terminated
            </State>
        </ChangeProcessInstanceState.Response>
    </WfMessageBody>
</WfMessage>
```

The **ProcessInstanceStateChanged** operation from the Observer group notifies the observer of state changes of a process instance. In the example above, when the process is terminated, a message will be encoded in XML to notify the Observer of the termination of the process and supply the results of the process:

```
<?xml version="1.0"?>
<WfMessage Version="1.0">
    <WfTransport/>
    <WfMessageHeader>
        <Request ResponseRequired ="No">
        <Request/>
        <Key>http://www.Acme.com/wfx456</Key>
    </WfMessageHeader>
    <WfMessageBody>
        <ProcessInstanceStateChanged.Request>
            <ProcessInstanceKey>
            http://www.dell.com/WfcXML1199
            </ProcessInstanceKey>
            <State>closed.abnormalCompleted.terminated
            </State>
            <ResultData>
                <Order>ACM00456</Order>
                <Account>ACM-400-2460</Account>
                <Amount>150.00</Amount>
            </ResultData>
            <LastModified>1999-12-25T15:10:35Z
            </LastModified>
        </ProcessInstanceStateChanged.Request>
    </WfMessageBody>
</WfMessage>
```

Reference

1. Workflow Management Coalition, "Workflow Standard—Interoperability Wf-XML Binding," WfMC-TC-1023, Version 1.0, May 2000; available online at http://www.aiim.org/wfmc/standards/docs/tc1023v10.pdf.

E-Business Standardization Efforts

No single standard can address all aspects of interactions between business partners in an e-business scenario. A variety of standardization efforts that deal with specific facets of the overall problem are ongoing. Business-to-business protocols are being defined and standardized for capturing different business models and processes.

- The World Wide Web Consortium (W3C, http://www.w3.org/) is working on XML Schema, a common schema definition for complex data.
- The Electronic Business XML Working Group (ebXML, http://www.ebxml.org), initiated by the United Nations body for Trade Facilitation and Electronic Business and the Organization for the Advancement of Structured Information Standards (Oasis, http://www.oasis-open.org), focuses on standardizing XML business specifications. Oasis coordinates related work on the standardization of trading partner agreements.
- The BizTalk (http://www.biztalk.org/) initiative, started by Microsoft, aims to define a set of guidelines for publishing schemas in XML and for using XML messages to easily integrate software programs. The simple object access protocol (Soap) is an XML/HTTP-based protocol developed by Microsoft for accessing services, objects, and servers in a platform-independent manner.
- The Open Buying on the Internet (OBI, http://www.openbuy.org) Consortium is working on a simple, client-server–style protocol for low-value, high-volume transactions over the Internet.
- The RosettaNet (http://www.rosettanet.org) Consortium focuses on standardizing business processes within the IT industry's supply chain, such as defining partner interface processes (PIP) for new product notification, product and technical data interchange, management of purchase orders, and order status.
- The eCo Framework Project, a CommerceNet consortium (http://www.commerce.net) project, promotes integration of common electronic commerce services through XML.
- The Open Trading Protocol (OTP, http://www.otp.org) Consortium is pledged to developing an XML-based open and interoperable standard for purchasing goods and services on the Internet.

- The Wf-XML response is the response message body for output.
- Both request and response specify "Content-type: text/xml" in the HTTP message header. Content-length is determined by the length of the Wf-XML request or response, respectively.
- Authentication is accomplished through the standard HTTP mechanisms, ranging from certificates to "no authentication for 'public' services." Requests that need to be authenticated are normally performed over an SSL connection.

Wf-XML STATUS AND FUTURE DIRECTIONS

The Wf-XML specification was promoted to beta status at the WfMC meeting in December 1999, and version 1.0 will be released as an official WfMC standard in May 2000.[10]

The initial version of the Wf-XML specification defines a minimum interoperability message set only. Future versions may include additional operations to support

- retrieving a list of instances for a given process definition,
- setting the values of any number of properties for a given process instance resource,
- registering a resource with another resource to be notified on status changes and event occurrences,
- retrieving the list of events that have occurred on a particular resource, and
- sending event notifications to registered observer resources.

In addition, protocol bindings will be added for non-HTTP transport mechanisms, such as message-queuing systems.

The current Wf-XML specification supports the actual interaction of business partners. In e-business interactions, this is the final stage of a more complex process that also includes identifying potential business partners (information gathering) and establishing an interaction contract with them (contract negotiation). Future versions of the Wf-XML specification will also attempt to provide support for the information-gathering phase by providing XML messages that support introspection of process definition interfaces.

Wf-XML does not define a standard for business process interfaces; rather, it provides a language and protocol that enables interaction among such processes. We assume that in the near future, process interfaces (such as the structure of context and result data) will be established between individual companies. However, there are many emerging initiatives that attempt to standardize schema for information exchanged in interenterprise business interactions, interfaces of services used in these interactions, or even cross-enterprise business processes. The WfMC expects to synchronize the Wf-XML specification with such standards as they evolve.

Our vision for Wf-XML is reflected in a statement by the JCALS Infrastructure software manager:

"All future workflow integration efforts by the JCALS team with outside workflow vendors are envisioned to utilize the Wf-XML specification in order to simplify integration efforts and to maximize design/code reuse. By capitalizing on the Wf-XML specification, we will be able to provide timely solutions in a more cost-efficient manner. In addition, as more products, such as commercial PDM packages, adopt the standard, we will be able to easily expand our integration efforts and arsenal of features to our end users." ■

ACKNOWLEDGMENTS

We thank our fellow members of WfMC Working Group 4 on Interoperability, especially Mike Anderson, Richard Heim, Mike Marin, Edna Murby, Michael zur Mühlen, Joseph Rogowski, Michael Rossi, the many participants of the SWAP initiative and the OMG Workflow Management Facility submission, and the member organizations who funded this work.

REFERENCES

1. Workflow Management Coalition, "The Workflow Reference Model," WfMC-TC-1003, Version 1.1, Jan. 1995; available online at http://www.aiim.org/wfmc/standards/docs/tc003v11:pdf.
2. M.-T. Schmidt, "Evolution of Workflow Standards," *IEEE Concurrency*, vol. 7, no. 3, July-Sept. 1999.
3. Workflow Management Coalition, "Workflow Standard—Interoperability Abstract Specification," WfMC-TC-1012, Version 1.0, Oct. 1996; available online at http://www.aiim.org/wfmc/standards/docs/if4-a.pdf.
4. Workflow Management Coalition, "Workflow Standard—Interoperability Internet e-mail MIME Binding," WfMC-TC-1018, Version 1.2, Jan. 2000; available online at http://www.aiim.org/wfmc/standards/docs/tc018v12.pdf.
5. "Workflow Management Facility," Object Management Group Joint Submission, revised submission, July 1998; available online at ftp://ftp.omg.org/pub/docs/bom/98-06-07.pdf.
6. M.-T. Schmidt, "Building Workflow Business Objects," in *Business Object Design and Implementation II*, D. Patel, J. Sutherland, and J. Miller, eds, Springer-Verlag, London, 1998, pp. 64-76.
7. K. Swenson, "Simple Workflow Access Protocol (SWAP)," Internet draft, 7 Aug. 1998; available online at http://www.ics.uci.edu/~ietfswap (last accessed 3 May 2000).
8. Y. Goland et al., "HTTP Extensions for Distributed Authoring—WEBDAV," IETF RFC 2518, Feb. 1999; available online at ftp://ftp.isi.edu/in-notes/rfc2518.txt.
9. Computer Sciences Corp., Integrated Systems Division, "JCALS PC Client Simple Workflow Access Protocol (SWAP) Interface Design Document," 28 Dec. 1998; available online at http://199.210.109.10/download/Files/pccv3/wfmc_swap.doc.
10. Workflow Management Coalition, "Workflow Standard—Interoperability Wf-XML Binding," WfMC-TC-1023, Version 1.0, May 2000; available online at http://www.aiim.org/wfmc/standards/docs/tc1023v10.pdf.

James G. Hayes is a development manager at Computer Sciences Corporation where he is responsible for the JCALS Software Infrastructure. Hayes graduated from Temple University with majors in computer science and accounting and holds a master's of engineering degree from Pennsylvania State University.

Effat Peyrovian is president of ECC Consultants, based in New Jersey, and was the editor of Wf-XML 1.0 Beta. She is currently providing consulting services to AT&T. Peyrovian graduated from the University of Kansas with majors in mathematics and economics and holds a master's of science degree in computer science from the same university.

Sunil Sarin is a principal architect with TIBCO Software Inc. He has served as chair of the Workflow Management Coalition Interoperability Working Group, and received an AIIM International Standards Excellence Award. Sarin received an undergraduate degree in engineering from the Indian Institute of Technology, and master's and PhD degrees in computer science from the Massachusetts Institute of Technology.

Marc-Thomas Schmidt is a software architect in IBM's UK software development laboratory. His special interest areas are application development and modeling tools. Schmidt is a vice-chair of the WfMC Technical Committee and leads the WfMC business objects work group. He received an AIIM International Standards Excellence Award, and is a coauthor of the OMG Workflow Management Facility standard, and co-chair of the OMG workflow work group.

Keith D. Swenson is a software architect and development director at MS2 Inc. He has developed groupware and workflow products in the past at Netscape, Fujitsu, and Ashton-Tate. From 1995 to 1997 he served as an officer of the ACM Special Interest Group for Group Support Systems. Swenson has participated in the development of a number of workflow standards and, in 1996, was elected a Fellow of the Workflow Management Coalition.

Rainer Weber joined SAP AG, Walldorf, Germany, in 1994. His principal areas of focus are workflow interfaces to external systems, as well as Web interfaces. He studied computer science at the University of Erlangen-Nuremberg, and holds a Dr. in computer science from the University of Technology, Munich.

Readers can contact Hayes, the current editor of the Wf-XML specification, at jhayes@jcals.csc.com.