RDF-Based Data Integration for Workflow Systems

Jun Shen
Swinburne University of Technology

Yun Yang
Swinburne University of Technology

Follow this and additional works at: http://aisel.aisnet.org/acis2002

Recommended Citation
Shen, Jun and Yang, Yun, "RDF-Based Data Integration for Workflow Systems" (2002). ACIS 2002 Proceedings. 70.
http://aisel.aisnet.org/acis2002/70

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
RDF-Based Data Integration for Workflow Systems

Jun Shen and Yun Yang

CICEC – Center for Internet Computing and E-Commerce
School of Information Technology
Swinburne University of Technology
Hawthorn, Victoria, Australia
jshen@it.swin.edu.au

Abstract

To meet the requirements of interoperability, the enactment of workflow systems for processes should tackle the problem of data integration for effective data sharing and exchange. This paper aims at flexibly describing workflow entities and relationships by innovative ontology engineering, which are emerging in process-centred environments, supported by Resource Description Framework (RDF) based languages and tools. Our novel framework takes into consideration to position the ontology level in the data integration dimension. Having taken a more realistic approach towards interoperability, we present basic constructs of a workflow specific ontology, with a suite of classes and properties selectively created. In particular, we demonstrate an example description of Event Condition Action (ECA) rules by extensions of RDF. As an inter-lingua, the proposed vocabulary and semantics can be mapped onto other process description languages as well as the simple XML-based data representation of our earlier workflow prototype.

Keywords

Workflow systems, data integration, ontology, RDF, process description

INTRODUCTION

As advanced e-services boom, workflow systems are still one of the most promising solutions for process support, such as e-business. XML, an integral language tool, plays important roles in data and application integration. Many kinds of standards and protocols have emerged increasingly, and also redundantly, in favour with XML. Considering the applications in the e-business area, we should tolerate the coexistence of cXML (www.cxml.org), ebXML (www.ebXML.org), xCBL (www.commercenet.com) and so forth. Meanwhile, we face comparable embarrassment when considering development of interoperability issues in workflow systems themselves. Since Hsu (1995) edited an assembly of technical reports from various groups, little substantial work has been done on data integration for data sharing and exchange among workflow systems. In the context of database, certain theories and models have been proposed (Bajaj, 2002), whose work have made semantics and logics of both distributed data and concurrent activities clearer, more formal and explicit. Nevertheless, when we look at current situations of cooperation among dispersed and heterogenous workflow applications, we should emphasize more concerns on simpler representations and interchange of data between Internet/ Intranet wide entities. WfMC (www.wfmc.org) strives to fix this problem and regards XML as a granted bearer by specifying XPDL (XML Process Description Language) (WfMC 2001) and WfXML. Meanwhile, Riempp (1998:78-83) has proposed an additional ‘Interface 6’, or enlargement of Interface 4, for interoperability between workflow engines and/ or managers. Ironically, few implementations were wholly bound to such standards or interfaces due to the inherent lack of sound support for explicit semantics.

In this paper, we take a more realistic approach towards interoperability, this is to say, data integration issues from the lessons we learned. Our basic tool is Resource Description Framework (RDF). Section 2 simply reviews the current ideas. To meet the requirements of ontology construction, which is synthesised in section 3, a new framework has been proposed in section4. Our contribution in the RDF-based representation of workflow elements is detailed thereafter. Section 5 briefly describes our new prototype for the proof of concepts and section 6 concludes our work.
RELATED WORK

Data Integration Using XML

XML has been accepted as a uniform data format widely not only for organisational processes but also for task-specific tools. This will enable distributed software environments, like workflow systems, to be interoperable to a much greater extent when taking advantages of its flexibility and portability. Through evaluating our work and comparing related research, we list the following main drawbacks of XML’s application in workflow environment:

- Extraordinary time consumption for queries and updates: Performance issues of XML query remain at a developing stage and it is not easygoing to get a well defined Document Type Definition (DTD) or Schema. In an experimentation of less than ten concurrent threads, it took a few seconds to refresh the status of work participants and processes in a workflow system, even though read/write operations were optimised by the XPath methods.

- Lack of mechanisms for universal distribution and discovery: We also looked at possible solutions specified by Object Management Group. To the best of our knowledge, legacy systems, which are in line with CORBA-like platform, have not seamlessly integrated XML to store and locate information in a global manner. On the other hand, WfXML or XPDL themselves care more about the information syntax. At this stage, we should make more painstaking efforts to ensure such information at hand in either an online or offline mode, whenever necessary.

- Ubiquitous misunderstanding between different contexts: This is the most significantly problematic issue in data integration of workflow systems. In pursuit of interoperability based on speech-act theory, WfXML and its abortive twin brothers, such as SWAP and MAGI (Bolcer, 2000), kept trying to absorb simplism philosophy of Internet/web and almost regarded the bindings of HTTP/IIOP (Internet Inter-ORB Protocol) of workflow operations as sufficient. We argue that, for workflow state machines, they would fail because of the limited semantic representation capability. This so-called capability, nevertheless, has been enriched by recent work in other laboratories. For example, as reported by Aalst et al. (2001), they incorporate the power of Petri nets and/or XML Algebra. At this stage, we also tend to concern about the similar complexity that has been brought by existing B2B standards, where a lot of things mean nothing, especially to new business process engineering staff.

We believe that workflow specific domain ontology will help us identify problems from a different and also consistent point of view, where XML alone cannot solve the entire problem.

Ontology Construction

Although it comes from knowledge engineering areas, the term of ontology has been cited more and more in information systems nowadays. Uschold (1996) described that, ‘ontology is a formal description of the entities within a given domain: the properties they possess, the relationship they participate in, the constraints they are subject to, and the patterns of behaviour they exhibit’.

Ontology-based data integration has been studied to some extent. Taking Omelayenko et al.’s (2001) work as an example, they introduced a synonym of ‘ontology’, universal catalogue, which acts as a bridge between heterogeneous product information that were described by different standards such as xCBL and cXML. In their implementation, XSLT (eXtensible Style-sheet Language Transformations) plays transforming roles between DTDs. Similar approaches, this is to say, bottom-up methods as Uschold categorised, have been applied to construction of chemistry and art ontologies (Wielinga, 2001).

We prefer Uschold’s middle-out methods in the development of workflow ontology because we should stipulate commonality, stability, and verifiability of consistency and accuracy rather than process description details, which would be hard to manage and handle especially when basic vocabulary and semantics are confusing. Natvig et al. (1999) proposed an informal and general meta-model of shared information spaces and used
ontology to organise and categorise information resources. However, they merely took one workflow prototype as an example of their perspectives. Similar work in Australia can be found on CSIRO Mathematical and Information Science website, where research has been taken on agent-based e-commerce and ontology metadata thesauri in the group of AI in E-Business and Technologies for Electronic Documents (www.cmis.csiro.au/iaebe-commerce.htm or www.ted.cmis.csiro.au/omt/). In this paper, we focus on how to construct workflow ontology to support process organisations.

WfMC specifications indicate shared understanding to a great extent, therefore our candidate of vocabulary stems from their glossary, although highly informal. On the other hand, NIST’s efforts on Process Description Language (PSL) are well structured (Schlenoff, 2000), and formally based on the Knowledge Interchange Format (KIF). The rich semantic expressive power of PSL is a sound skeleton for us. When mapping PSL concepts to XML representation, RDF’s benefits were also explored and compared with KIF (Schlenoff, 2000, appendix).

Decker et al. (2000) once portrayed a portrait of roles of XML and RDF, and by now RDF have been incorporated in some of the ontology projects we mentioned above. We acknowledge that RDF’s power is still limited, even if related foundational theories are work in progress. However, with diversified extensions of RDF, it is more applicable to the ontology of process-centred environments than sole XML, while we are on the way far from a wholly knowledge system supported workflow implementation.

```
<projects>
  <project proj_id="00" proj_name="Co project" />
  <project proj_id="01" proj_name="Plan project" />
  <project proj_id="02" proj_name="Test project" />
</projects>

(a)
<task task_id="1001">
  <proj_id>01</proj_id>
  <task_name>Review</task_name>
  <start_date>18/02/2002</start_date>
  <finish_date>06/03/2002</finish_date>
  <status>unenacted</status>
</task>

(b)
<dependant task_id="1001">
  <dependant_id>1001, 0001</dependant_id>
  <proj_id>01</proj_id>
</dependant>

(c)
<people task_id="1001">
  <username>David Jones</username>
  <proj_id>01</proj_id>
</people>
<tool task_id="1001">
  <tool_name>Java</tool_name>
  <proj_id>01</proj_id>
</tool>

(d)
```

Figure 1: Examples of workflow XML files
REQUIREMENTS FOR WORKFLOW DESCRIPTION LANGUAGES

Based on our previous workflow prototype (Yang 2002), Figure 1 shows some flat syntax structures of separate relations described in XML. Task Review of Plan Project has a unique id as 1001 and an initial status as unenacted, its commencement depends on the completion of task 0001 (Design) and whether David Jones is free and Java tool is available. It’s a primitive shape of WfMC process model including essential semantics that may appear in common contexts. We can simply map parts (a), (b), (c) and (d) onto the correspondent workflow process definition, activity, transition information, and workflow participant specification and application declaration, respectively, although they cannot be easily translated into the XPDL vocabulary and syntax (WfMC, 2001) directly. To define these tags, we have developed both DTDs and schemas.

For interoperability of the workflow system, we are not short of different vocabularies at different levels, but unique requirements for ontology and proper description language are still concerned as follows:

- Basic vocabulary should act as a proper inter-lingua with a well-formed scope, which should be not only necessary but also sufficient. Take XPDL as an example, it can only make out a base for ontological use in its current standardised shape.
- The degree of formality should be reasonably modest. XML certainly fails because of the semantics ambiguity it brings inherently. Formal or semi-formal options such as PSL and KIF are possible options, but over complicated to be used in the implementation and deployment of workflow systems.
- Explicit semantics should be as simple as possible; however, it should provide enough extensibility whenever it is required to express more complex logic or constraints. We would not bind special axioms onto the properties of or relations between entities and/ or classes.
- Unified syntax and universal distribution should be guaranteed intrinsically. This requirement makes communication, discovery and distribution of ontology data become possible to the greater extent.
- Performance can be optimised in comparison to semi-structured XML databases. It should be effortless to represent data dependency and reduce data redundancy as well.

Figure 2: Framework for workflow data integration

ONTOMETRY FOR DATA INTEGRATION IN WORKFLOW SYSTEMS

We propose our framework in Figure 2 to illustrate our contribution as addressed in this section. As for data integration, three levels exist along the vertical dimension, where...
RDF-Based Data Integration for Workflow Systems

Ontology level explicitly describes and represents abstract semantics beyond process. Integrated tools, such as CORBA platform and e-services, such as Simple Object Access Protocol and Web Service Description Language, provide possibilities to ease concrete realisation. According to the degree of formality required, we believe that RDF-based ontology is easier to realise when considering the overall requirements in workflow environments, though PSL or some types of Petri net are complementary.

In this section, we briefly sketch out typical extensions of RDF and the corresponding schema first, with the details of what to be considered and included in our prototype ontology following. A mapping of workflow constructs is exemplified at last.

Landscape of RDF

RDF and RDF Schema (RDFS) have absorbed theories of object-oriented programming, relational database and knowledge representations. The triples of RDF statements, which describe relations between resources and properties, are concise and natural, and most of all, flexible.

The most important enrichment of RDF we should mention here is OIL-Ontology Inference Layer (http://www.ontoknowledge.org/oil) and its extension and integration with an agent language DAML (http://www.w3.org/TR/daml+oil-reference), as well as FIPA-RDF (FIPA, 1999) and RDF Context (http://public.research.mimesweeper.com/RDF/RDFContexts.html). New concepts and primitives have been introduced in OIL and the like, for example, expression (oil:ClassExpression), axiom (daml:TransitiveProperty), rule (fipa:Rule), context (rdfc:asserts) and activity (fipa:Action).

With XML-based syntax, parsers and analysis tools for RDF descriptions have been developed and integrated in a lot of prototypes or systems (www.semanticweb.org). In our work, we avoid making workflow engines as an inference machine but require them capabilities to handle messages and store triples, which are bound to a lightweight ontology rather than a complete knowledge base. Moreover, inline coding of URI helps us to locate workflow related data and entities conveniently. In this paper, we will mainly show how RDF-based process ontology is advantageous over XML-based definitions.

Description of Workflow Ontology Using RDF Tools

Dourish (2001) claimed that process descriptions could be regarded as organisational accounting devices, including dual use of workflow technologies: organising resources and coordinating documents. We add that workflow ontology should be flexible and simple enough, which is the reason why we prefer the middle-out methods of construction for controlling the level of details. The essence of every process description is a combination of basic vocabulary and semantics.

Bajaj et al. (2002) stated their SEAM model captured different aspects of workflow and demonstrated itself as an amalgamation of current models. We should acknowledge that their contribution lies in system development from the database models point of view, despite their considerations cover most constructs that should be included in process description. However, besides Entity, State, and Activity, the most controversially indeterminate construct is temporal modelling and representation. Fortunately, from this point of view, PSL includes rich axiomatic paradigms for describing timing constraints and relationship between activities.

As for complexity, rules are the most difficult to model and manage, especially when the Event Condition Action (ECA) mechanism is well adopted in workflow systems. Actually, on a distributed object platform, Kappel et al. (1995) had also incorporated rules with roles, another important concept. Same as their special handling of class of roles, Fan et al. (2001) argued to keep role concept overlapping with, but also separated from, entities, whose purposes of activities should be represented by proper roles.

After the above comprehensive brainstorming and abstracting, we have reached a would-be agreement on workflow ontology. Basic class hierarchy of rdfs:Resource (rdfs:Class) is shown in Figure 3. There are some meta-classes such as Activity, Entity, Role, Rule and TimePoint, which describe common constructs within a generic workflow system. Further, subclasses such as ComposedActivity, Loop, human, agent, auditor and compensator,
which represent characteristic kinds of things, are regarded necessary. For example, one human-like autonomous agent program may play a role as compensator that is responsible for exception handling.

As for our prototype mentioned in section 3, proj_01 and task_1001 are represented as flat activities, where the subsumption relationship has been hidden. This meta-level method is appropriate for routine processes, so that hence every occurrence or instance of a certain activity takes such a description granted as template. When the processes are ad hoc, designer may choose another option to represent every activity at instance level instead of class level, so are the description of events and states (classes Event and State).

Primitive relations between classes and individuals can be found in Figure 4 using Onto Edit (www.ontoprise.de). Besides the core properties of workflow resources such as activityName, executedBy and participatedBy, the most important relations between process activities are sequences, loops and branches. The basic ordering relations are nextActivity, and its inverse, lastActivity. As depicted in Figure 4, nextActivity is a linear and transitive relation guaranteeing that every occurrence of task_0001 must be prior to that of task_1001. We distinguish transitions into two disjoint classes Join and Split, which are connected to other activities by joinFrom and splitTo. Each class or instance of transition may be a composition of activities by connectors such as AND, OR and XOR. Figure 5 is an excerpt of our RDF model with the OIL syntax.

Table 1 shows the mapping of concepts and relations from our workflow ontology onto XML-based prototype and two other specification languages. The mapping is not one-to-one but overlapped. We would show an example to illustrate the description of Rule in the following subsection.

**An Example of ECA Rules Described by Extended RDF Languages**

When workflow activities are instantiated, they maintain and update their states, and publish and subscribe events. Both states and events are fixed by certain time-stamp, so that semantics, such as ‘state unenacted holds within a time duration of interval’ or ‘event1...
occurs at a time point’, become clearly represented. Moreover, every reified RDF description of activity instance should explicitly include entity-value pairs and related events as well as timing information, if necessary.

Figure 4: Relations among resources

```xml
<!ENTITY b 'http://www.ontoknowledge.org/oil/rdf-schema/oil-standard#'>
<!ENTITY d 'http://www.w3.org/2000/01/rdf-schema#'>
<b:HasValue rdf:about="_anon2">  
  <b:toClass rdf:resource="_anon3"/>
  <b:onProperty rdf:resource="joinFrom"/>
</b:HasValue>
<b:And rdf:about="_anon3">  
  <b:hasOperand rdf:resource="task_1001"/>
  <b:hasOperand rdf:resource="task_1101"/>
</b:And>
<d:Class rdf:about="transition2">  
  <d:subClassOf rdf:resource="Join"/>
  <b:hasPropertyRestriction rdf:resource="_anon2"/>
</d:Class>
```

Figure 5: Description of one AND-Join transition

Above all, RDF and RDFS can hardly be regarded as a strict combination of a type system, entity-relationship model and first-order logic format. Yet, they cannot easily describe logic constraints besides the range and domain, and such relationships as set-element, attribute-value or class-instance, as an ontology specification requires (Decker, 2000).

For example, description of an ECA rule requires more semantic expression capabilities than what we have discussed so far. Figure 6 gives some options, where part (a) introduces some specific namespaces such as rdfq and wfonto, besides fipa. Here, terminateException is an event that may trigger corresponding actions such as updating currentState. To enable the action, certain conditions should be satisfied in advance. As for task_1001, if the activatedInstance has been inactive, say, for more than 48 hours, we might regard it as unrecoverable task and some compensation actions should be initiated perhaps. RDF
queries should be supported in this situation; however, there is no standard solution for it by
now. The other means of description of rules are shown in parts (b) or (c) of Figure 6, which
leave the complex semantic representation to implementation or artefacts such as
ClassExpression or Context. As XPDL and PSL also disregard this expressive mechanism
in their newest version, we would like to mention some efforts such as the XML Declarative
Description, which may denote variables flexibly beyond RDF schema (Wuwongse, 2001).

<table>
<thead>
<tr>
<th>workflow ontology</th>
<th>XML prototype</th>
<th>XPDL</th>
<th>PSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity class</td>
<td>project, task</td>
<td>Process, Activity</td>
<td>activity</td>
</tr>
<tr>
<td>Entity</td>
<td>-</td>
<td>-</td>
<td>object</td>
</tr>
<tr>
<td>human</td>
<td>people</td>
<td>Participant</td>
<td>-</td>
</tr>
<tr>
<td>agent, application</td>
<td>tool</td>
<td>Application, Data</td>
<td>-</td>
</tr>
<tr>
<td>Role</td>
<td>-</td>
<td>ParticipantType</td>
<td>-</td>
</tr>
<tr>
<td>compensator</td>
<td>-</td>
<td>-</td>
<td>Repairable-fluent</td>
</tr>
<tr>
<td>State</td>
<td>status</td>
<td>Attribute, Parameter</td>
<td>fluent, exists-at</td>
</tr>
<tr>
<td>Event</td>
<td>-</td>
<td>-</td>
<td>activity-occurrence</td>
</tr>
<tr>
<td>TimePoint</td>
<td>-</td>
<td>Valid Date</td>
<td>timepoint</td>
</tr>
<tr>
<td>Join, Split, Loop</td>
<td>dependant</td>
<td>Transition Restriction</td>
<td>Junctions</td>
</tr>
<tr>
<td>activityName</td>
<td>id, name</td>
<td>Id, Name</td>
<td>?variable</td>
</tr>
<tr>
<td>FinishTime, StartTime</td>
<td>finish_date, start_date</td>
<td>Valid To/ From Date</td>
<td>endof, beginof</td>
</tr>
<tr>
<td>nextActivity</td>
<td>dependant</td>
<td>From, To</td>
<td>next-activity</td>
</tr>
<tr>
<td>joinFrom, splitTo</td>
<td>dependant</td>
<td>TransitionRef</td>
<td>Junction</td>
</tr>
<tr>
<td>participatedBy</td>
<td>username</td>
<td>Responsible, Performer</td>
<td>participate-in</td>
</tr>
<tr>
<td>Rule</td>
<td>-</td>
<td>Condition, Subflow</td>
<td>Achievement</td>
</tr>
</tbody>
</table>

Table 1. Mappings between languages

PROTOTYPING

As described elsewhere (Yang, 2002), we have based on XML files, instead of relational
database that was used in the previous workflow prototype, as the universally accessible
portable data repository for data integration. In this new prototype, XML files and visual tools
are used to meet the user-friendly requirements of a workflow system. Corresponding to the
deployment described in Figure 1, the demonstration can be seen in Figure 7. Four tasks,
Design (task_0001), Review (task_1001), Editing (task_1101), and Documentation
(task_2001) constitute a simple split-join workflow.

Now with ontology tool (OilEd, www.ontoknowledge.org/oil), we can view and implement
workflow systems from a different point of view. For example, task_1001 is defined as
ComposedActivity in RDF-based workflow ontology (Figure 8). It is followed by an AND-Join
transition2 in parallel with task_1101 (as in Figure 5), participatedBy individual person
Jones, who is also a subclass of both human and auditor, and executedBy individual
application tool Java. The semantics of workflow entities and relationships become clearer
and easier to handle.

RDF files may be stored as triple sets in relational databases, parsed by XML/RDF parser
ingines and analysers, and accessed through URI addresses and anchors
(www.w3.org/RDF/Interest/). We should note that Java-based packages and some open
resource toolkits, which are available within RDF special interest groups, offer greater
interoperability to our implementation of workflow data integration.
CONCLUSION AND FUTURE WORK

Due to the limitations of the XML solution to data integration for data sharing and exchange in workflow systems, we believe that data integration should incorporate ontology engineering. In comparison with other similar research work, we focused on the innovative RDF-based descriptions of entities and relationships for workflow processes. Our novel framework clearly shows the related issues and relations among data integration levels. The concept space that we have sketched out at this stage may form a basis for the construction of most common and conventional workflow ontology, although some constructs such as part of Event Condition Action rules remain ambiguous. In our prototype, diversified tools have been integrated to support the ontology development and analysis. With our framework, mappings among different process description languages are relatively easy to implement.

Figure 6: ECA rule description
As shown in this paper, RDF-based languages and tools are promising options for data integration in a distributed cooperative environment, such as the construction of workflow ontology. RDF mechanisms have been incorporated into existing workflow management system in order to improve interoperability and knowledge sharing among peer entities. RDF schemas may also converge together different vocabularies and semantics from different e-commerce areas or cooperation. With the far-ranging use of web services and semantic web, deploying RDF for data integration in workflow systems becomes inevitable. In the future, we need to refine our ontology and its representation language, especially for unambiguous descriptions of complex elements such as rules and time points. We also need to investigate data-centric application integration further in workflow systems.

REFERENCES


ACKNOWLEDGEMENTS

This work is partly supported by a Swinburne VC Strategic Research Initiative 2002-4. We are grateful for many members of CICEC for constructive discussions.
COPYRIGHT

Jun Shen and Yun Yang © 2002. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.