Association for Information Systems
AIS Electronic Library (AISeL)

ACIS 2002 Proceedings | Australasian (ACIS)

December 2002

RDF-Based Data Integration for Workflow Systems

Jun Shen
Swinburne University of Technology

Yun Yang
Swinburne University of Technology

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 exsiting 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 CSRIO 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/aieb/e-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.

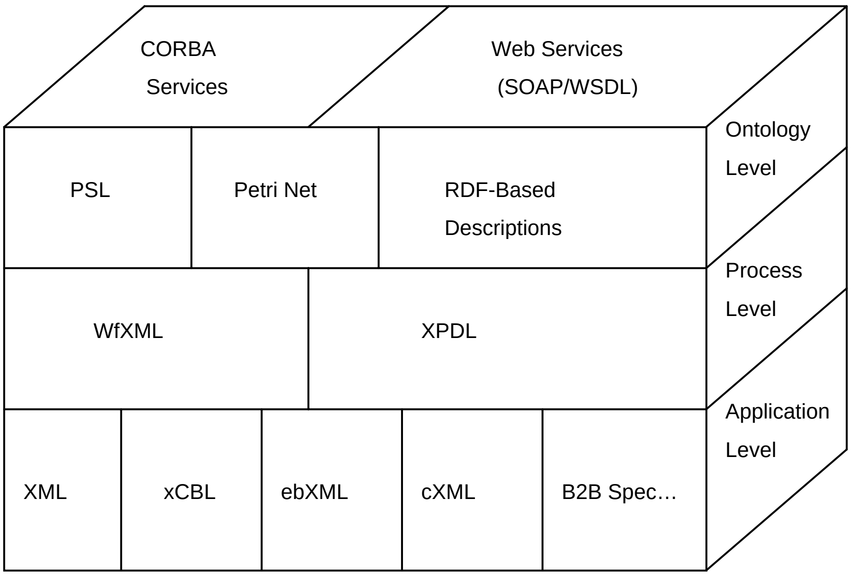

Figure 2: Framework for workflow data integration

ONTOLOGY 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 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.

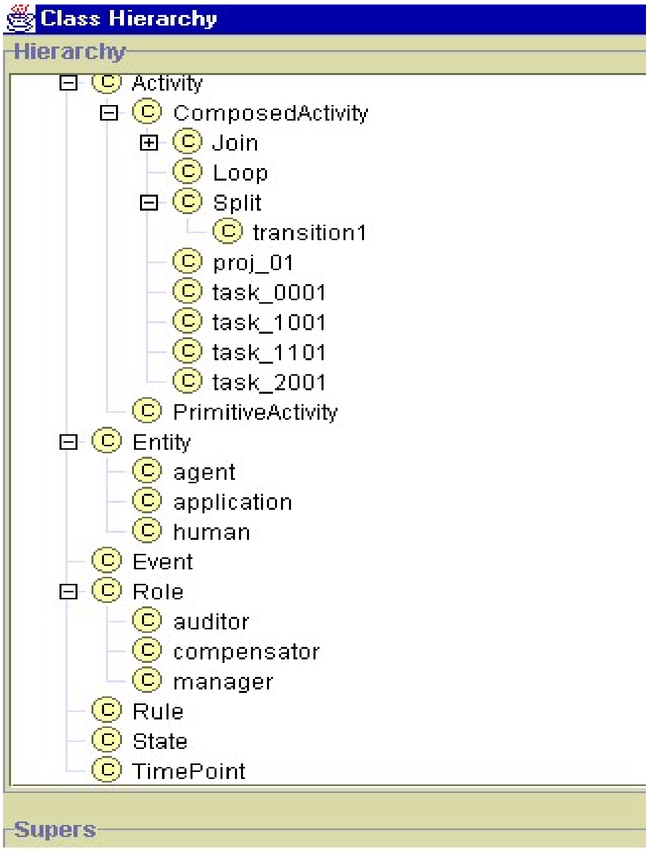

Figure 3: Basic class hierarchy in RDFS

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*, *excutedBy* 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.

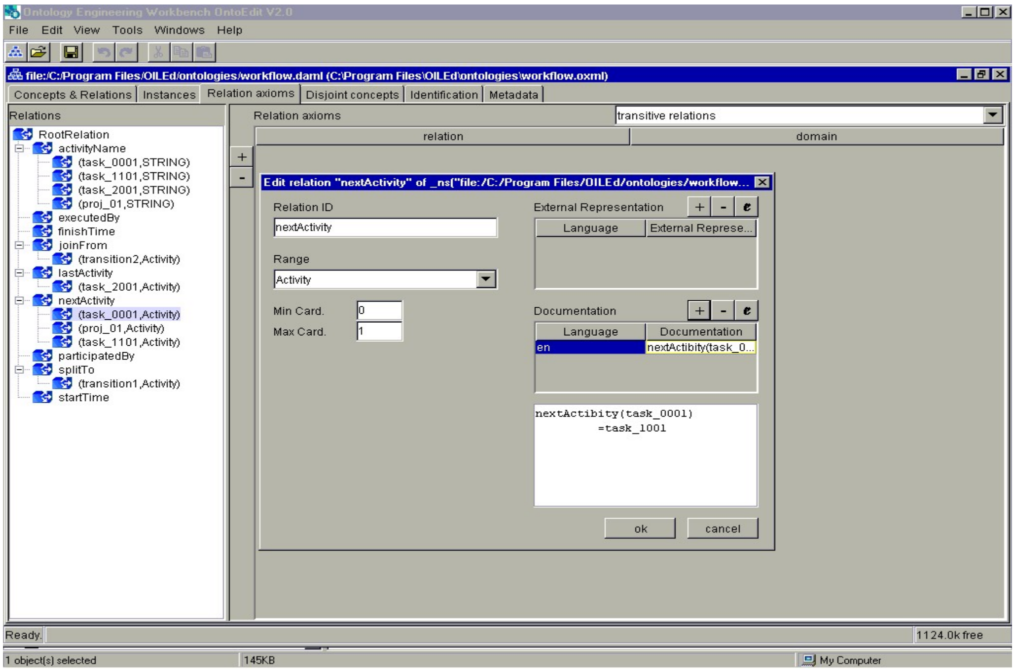

Figure 4: Relations among resources

```
<!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).

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

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 engines 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.

```
<rdf:RDF     xmlns:fipa="http://www.fipa.org/schemas/fipa-rdf1#"
      xmlns:wfonto="http://www.it.swin.edu.au/centres/ciece/workflow-ontology"
      xmlns:rdfq="http://www.w3.org/TandS/QL/QL98/pp/rdfquery.html">
      <fipa:Rule rdf:ID="terminateException">
            <fipa:selection-result rdf:ID="activitedInstance"/>
            <rdfq:rdfquery>       <rdfq:From eachResource="wfonto:currentAcitivities">
                  <rdfq:Condition>
                              <rdfq:equals>
                                    <rdfq:Property name="rdf:type"/>
                                    <rdfq:String>wfonto:task_1001</rdfq:String>
                              </rdfq:equals>
                              <rdfq:greaterThan>
                                    <rdfq:Property name="wfonto:inactiveDuration"/>
                                    <Integer>48<Integer>
                              </rdfq:greaterThan>
                  </rdfq:Condition>
            </rdfq:From>          </rdfq:rdfquery>
            <fipa:manipulation>
                  <rdf:Description rdf:aboutEach="activatedInstance">
                        <wfonto:currentState>wfonto:unrecoverable</wfonto:currentStare>
                  </rdf:Description>          </fipa:manipulation>
      </fipa:Rule>
</rdf:RDF>
```

(a)

```
<fipa:Rule ID="RuleName">
        <fipa:implementedAs>
             <fipa:Code>…</fipa:code>
        </fipa:implementedAs>
</fipa:Rule>
```

(b)

```
<wfonto:Rule ID="RuleName">
        <wfonto:premise>"#ClassExpression"</wfonto:premise>
        <wfonto:conclusion>
             <fipa:Action>…<fipa:Action>
             <rdfc:Context>
                        <rdfc:asserts>…</rdfc:asserts>
             </rdfc:Context>
        </wfonto:conclusion>
</wfonto:Rule>
```

(c)

Figure 6: ECA rule description

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.

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.

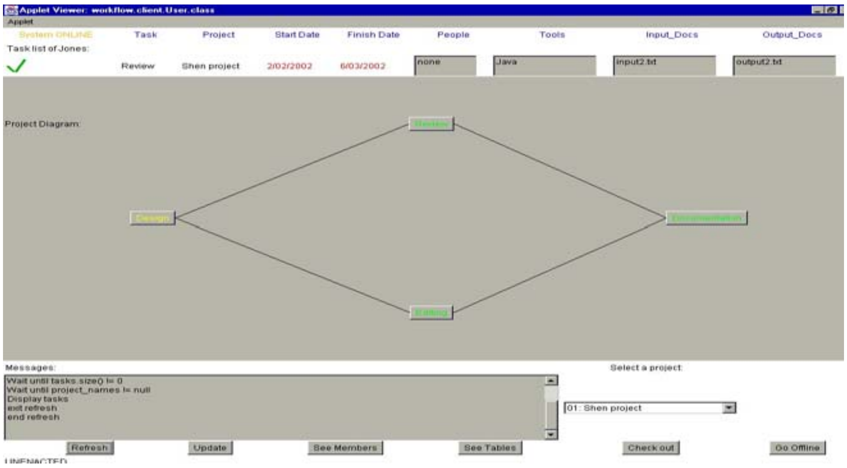

Figure 7: User interface of process enactment

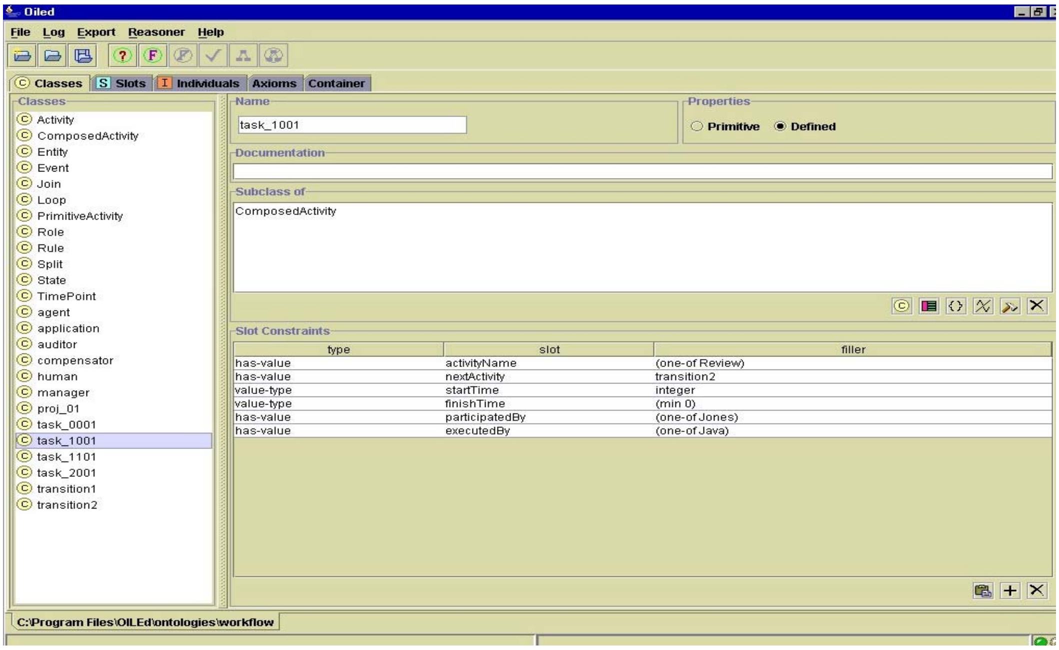

Figure 8: Properties of a to-do activity

REFERENCES

Aalst, W.M.P. van der, Verbeek, H.M.W., Kumar, A., (2001) 'XRL/Woflan: Verification of an XML/Petri-Net Based Language for Inter-Organizational Workflows', In *Proceedings of the 6th Informs Conference on Information Systems and Technology (CIST-2001)*, eds. K. Altinkemer and K.Chari, pp.30-45. Informs, Linthicum, MD

Bajaj, A., Ram, S., (2002) 'SEAM: A State-Entity-Activity-Model for a Well-Defined Workflow Development Methodology', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 14, No. 2, March/April, pp.415-431

Bolcer, G.A., (2000) 'Magi: An Architecture for Mobile and Disconnected Workflow', *IEEE Internet Computing*, Vol. 4, No. 3, May/June, pp.46-54

Decker, S, Melnik, S., Harmelen, F. van, Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I., (2000) 'The Semantic Web: The Roles of XML and RDF', *IEEE Internet Computing*, Vol. 4, No. 5, Sept./Oct., pp.63-74

Dourish, P., (2001) 'Process descriptions as Organisational Accounting Devices: The Dual Use of Workflow Technologies', in: *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, ACM Press, pp.52-60

Fan, J., Barker, K., Porter, B., Clark, P., (2001) 'Representing Roles and Purpose', in: *Proceedings of the First International Conference on Knowledge Capture*, ACM Press, pp.38-43

FIPA (1999) *FIPA Content Language Library*, FIPA 99 Specification, v 0.2, Part 18, Foundation for Intelligent Physical Agents, Geneva, Switzerland

Hsu, M., (ed.), (1995) *Special Issue on Workflow Systems*, Bulletin of the Technical Committee on Data Engineering, IEEE Computer Society, Vol. 18, No.1, March

Kappel, G., Lang, P., Rausch-Schott, S., Retschitzegger, W., (1995) 'Workflow Management Based On Objects, Rules, and Roles', in *Data Engineering Bulletin*: *Special Issue on Workflow Systems*, eds, M. Hsu, Vol. 18, No.1, March, pp.10-17

Natvig, M.K., Ohren, O., (1999) 'Modelling Shared Information Spaces (SIS)', in: *Proceedings of the 1999 International ACM SIGGROUP Conference on Supporting Group Work*, ACM Press, pp.199-208

Omelayenko, B., Fensel, D., (2001) 'A Two-Layered Integration Approach for Product Information in B2B Electronic Commerce', in *Proceedings of 2nd International Conference on Electronic Commerce and Web Technologies (EC WEB 2001),* LNCS 2115, Springer Verlag, Berlin, pp.226-239

Riempp, F., (1998) *Wide Area Workflow Management*: *Creating Partnerships for 21st Century*, in series of Computer Supported Cooperative Work, eds. D. Diaper and C. Sanger, Springer-Verlag, London

Schlenoff, C., Gruninger M., Tissot, F., Valois, J., Lubell, J., Lee, J., (2000) *The Process Specification Language (PSL)*: *Overview and Version 1.0 Specification*, NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD

Uschold, M., Gruninger, M., (1996) 'Ontologies: Principles, Methods, and Applications', *Knowledge Engineering Review*, Vol. 11, No. 2, pp.96-137

WfMC (2001) *Workflow Process Definition Interface-XML Process Definition Language*, WfMC-TC-1025, v 0.03, Workflow Management Coalition, Lighthouse Point, FL

Wielinga, B.J., Schreiber, A.Th., Wielemaker, J., Sandberg, J.A.C., (2001) 'From Thesaurus to Ontology', in: *Proceedings of the First International Conference on Knowledge Capture*, ACM Press, pp.194-201

Wuwongse, V., Anutariya, C., Akama, K., Nantajeewarawat, E., (2001) 'XML Declarative Description: A Language for the Semantic Web', *IEEE Intelligent Systems*, Vol. 16, No.3, May/June, pp.54-65

Yang, Y., (2002) 'Enabling Cost-Effective Light-Weight Disconnected Workflow for Web-Based Teamwork Support', *Journal of Applied Systems Studies*, Cambridge, England, Vol. 3, No. 2

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