PIF Workshop Report, Stanford. July 11-12 '96

Participants:
Adam Farquar (Stanford University), Doug Fridsma (Stanford University), Micheal Gruninger (University of Toronto), Pat Hayes (University of Illinois and KBSI Inc.), Jye-Chen Hwang (KBSI Inc.), Yan Jin (Stanford University), Jintae Lee (University of Hawaii and MIT Center for Coordination Science), Mark Musen (Stanford University), Keith Swenson (Netscape), Austin Tate (AIAI, University of Edinburgh), Mike Uschold (AIAI, University of Edinburgh).

Session 1 Report - Formalization: Syntax and Semantics

==

Summarized by Michael Gruninger, Univ. of Toronto

The discussion in this session focused on two major topics:
1) the syntactic representation of PIF
2) exploration of possible semantics for PIF

1) Syntactic Issues
We clarified and confirmed the use of the Frame Ontology syntax in the specification of PIF.

The class RELATION in the PIF class hierarchy has caused some confusion, since there are other relations in PIF. We agreed to rename the class as PIF-RELATION to identify the distinguished relations in PIF.

We agreed that PIF-SENTENCE was not an object in the language, but rather was a type of a slot-filler.

We noted that we have used symbol as the slot filler for some attributes where we should have used "entity". An example is the agent capabilities. Making the slot fillers be entity where we decided not to commit allows for PSV extension, whereas making them symbol commits to too restrictive a value that is not easily extended in a PSV.

The quantification conventions for PIF Core apply only to free variables in a PIF-SENTENCE. This is consistent with the fact that future PSVs may include explicit quantifiers.

Action items:
a) write down 4 or 5 sentences which describe the syntactic design approach fro PIF, object oriented, that attribute(object)=value is equivalent to a binary property style relation.

b) show equivalence of PIF structures to a flat FOL style of representation.

c) relationship of (primitive and compound) value types to the slots they can fill. Not shown in figure 2, but its IS a restricted relationship. Effectively, primitive and compound value types are in the entity class hierarchy in other words, and value-filler is a relation to these entities from the various attributes.

d) page 7 of PIF 1.1 document first sentence is incorrect. Must add "or another PIF entity" as current sentence is incorrect and too restrictive.

2) Semantic Foundations
Two proposals have been presented for the semantics of activities in PIF.

The first proposal is advocated by Chris Menzel and Jyh-Chen Hwang from KBSI and is based on situation semantics. In this approach, a distinction is made between activities, activity uses, and activations of activities. All of these become classes of objects in the ontology. Activity uses are used to distinguish between different instances of activities which have the same description.

The second proposal is advocated by Michael Gruninger from the University of Toronto and is based on the situation calculus. In this approach, an activity is defined by a set of constraints on the occurrence of its subactions. This allows the definition of different classes of activities (such as nondeterminism, iteration, and conditionals) through the use of different sets of constraints.

The discussion over semantics of PIF is really just beginning at this formal level, and the workshop was not intended to decide on this issue. Rather, we now have two proposals which we can use to explore semantic issues in PIF. The development of a semantics will be facilitated through the use of the examples that will be found on the web pages.

Session 2: Integration with the Workflow Management Coalition
==
Meta-Model
==========
Summarized by Austin Tate, Univ. of Edinburgh.

Introduction

The WfMC Workflow Process Description Language (WPDL) has been defined by WfMC Working Group 1B as part of the specification of Interface 1 of the WfMC reference model. Working Group 1A defined an API for creating, modifying and deleting process definitions in a workflow system which would use the entities defined within the WPDL. At the Orlando meeting of Working groups 1A and 1B in February 1996, the Draft PIF 1.1 document was examined and a statement made that the WfMC Working Groups believed that the approaches at their meta-model were sufficiently similar that convergence should be possible. Syntactic approaches of the WPDL and PIF did differ, but that was not considered crucial.

A half day session of the PIF workshop on 11th July was devoted to a study and comparison of the PIF and WfMC WPDL approaches to process interchange via a language. In particular the WPDL meta-model was compared to the PIF model and approach. The aim was to recommend ways in which PIF could come closer to WPDL, or vice versa in order that the best aspects of each could drawn on one another.

Comparison of WPDL Meta-model and PIF Core

WPDL and PIF share in common the following basic approaches:

a) process, sub-processes and activities have a hierarchical structure;

b) actors or agents have a relationship to the performance of activities;

c) objects and data relevant to processes are related to the processes.

WPDL differs from PIF in having:

a) a separation of manual and automated aspects;

b) a richer actor/agent model.

PIF differs from WPDL in having:

a) a mechanism for extensions in "Partial Shared Views" (PSVs);

b) a more formal approach to the semantics of the terms involved.

A comparison of the WPDL meta-model and the PIF core model showed that there were direct analogues in PIF for each type of entity in the WPDL meta-model with the exception of the transition information.

In general the WPDL separated a number of specialisations of the more generic activity and object entities in PIF. In particular this reflects the differentiation of manually enacted processes (done by people perhaps using tools) and automatic process step enactment.

WPDL Meta-model	PIF
Workflow process definition	Activity
Workflow process activity	Activity
(Sub-)process definition	Activity
Atomic Activity	Activity
Workflow participant definition	Agent
Workflow process relevant data	Object

Workflow application definition	Object
Transition information	Before, Decision

Transition information has a more complicated mapping into PIF core entities. The WPDL designers were trying to find an elegant minimal mechanism to represent a wide range of possible ordering and transition logic between process steps. They utilise a conditional transition from one step to another to do this. This is the subject of some debate in comments on the draft WPDL document. For example, the mapping to the WfMC Glossary and-split/join and or-split/join requires some structure to be used to convert to the less structured transition condition.

PIF's Decision which contains a "if" condition and a "then" and "else" link to (a set of) following activities could map to a (workflow process or other type of WPDL) activity. This activity would record a result in a "flag" that was an element of type workflow process relevant data. There would then be two transition information entities one for the flag in its "if condition is true" state and the other in its "if condition is false" state.

Things PIF Group can do to come closer to WPDL

There are a number of things that a future version of PIF could incorporate to come closer to WPDL.

Simple things:

* Comments could allow same formats as WPDL (i.e. /* */ and // to end of line.

* PIF could consider a less Lisp oriented style and utilise a C++ style.

* WPDL has a small number of required and a large number of recommended (but optional) documentation type attributes on process descriptions and other entities. PIF could consider having a PSV which standardised the documentation type attributes also.

More complex things:

* PIF could utilise the WPDL actor/agent meta-model as the basis for a PSV.

Things WfMC WPDL Group could do to allow for a convergence with PIF
----

* WfMC WG1 could consider adding "CAPABILITY=entity" to WPDL as an example of "role related information" (page 27 of the WPDL draft) as this is in PIF Core.

Things the PIF and WPDL Groups Could Study Together
----

a) PIF has had more extensive consideration of formalisation aspects, and it is part of the plan for the next version of PIF that a formal model will be further developed. PIF is likely to have semantically described entities. This work could usefully be studied along with the WPDL group to ensure the best practice experience of both sides is pooled.

b) The PIF Partial Shared View (PSV) extension mechanism could be a basis for sharing models with the WfMC, and for seeking a compatibility layer. The PSV mechanism might also be something which the WfMC WPDL could adopt.

c) A richer temporal model is to be defined in a PIF PSV. A simple example of this is in the current PIF 1.1 document, but more work is anticipated to define a standard temporal relationships PSV which builds on the simple model in the core. The PIF Core time model allows for more flexible begin/begin and end/end relationships on activities (to express "ladder" activities as found in some project management systems) whereas the WPDL model only admits simpler end/begin activity links (much like PIF's "successor" which is syntactic sugar for "end(a1) is before begin(a2)").

Next Steps

The PIF Group is to consider the following workplan to take PIF and WfMC WPDL compatibility forward:

a) Write a report of the workshop and PIF/WfMC session and provide it to PIF and WPDL (WfMC WG1) group members.

b) Code the WfMC Interoperability demonstration example in PIF core to provide a point of comparison and information interchange with the WfMC, and to feedback to the PIF group what Partial Shared Views (PSVs) may be useful.

c) Consider a WfMC WPDL compliant PSV on PIF Core to give the manual/automated process specialisations, etc.

d) Consider a richer Agent/Actor PSV on PIF Core to be compliant with the WPDL agent model (agent, resource, application program, etc.).

e) Consider a documentation and annotation PSV on PIF Core to give the same structure for process documentation as is provided for in WPDL.

Session 3 Examples and Translators
==
Summarized by Yan Jin, Stanford Univ.

In this session we discussed the purpose of examples, what kind of examples to choose and how to make the examples. We also watched and discussed a tool demo given by John Gennari from Stanford Medical Informatics group. Following is a summary of the discussion and the action items generated from the discussion.

Summary of discussion:
----

* Purpose of Examples:

(1) To explain what is PIF and to demonstrate the capabilities
(2) To demonstrate the usefulness of PIF
--> not only the process model itself but also how the model is used and exchanged among process support systems.
(3) To drive PIF PSV development.

* What kind examples are needed:

(1) PIF-CORE based examples and (PIF-CORE + PSV) based examples ---> We need both.

(2) CAN examples: to demonstrate the capability of PIF, and CAN'T examples: to show what is not covered by the PIF ---> We need both.

(3) Real examples vs. toy examples
---> Small examples are good to demonstrate specific capabilities Large/real scale examples is more powerful to show that PIF is practically useful.

* How to get/create specific examples:

(1) From others (other text books, systems, etc.) ---> one example of this case is that we can use WfMC's examples

(2) From each participating group: VDT example, TOVE example, PH example, and O-PLAN(?) example, etc.

(3) From practitioners: interviews, related engineering and business literatures.

* Example Repository and tools:

(1) We need a mechanism (site and tool) to create and accumulate PIF examples
(2) Stanford CAMIS' Protege(?) can be used for create examples
(3) AIAI's Hardy is another process description tool.

Action Items:
----

1) Jintae and Yan will work on the selected WfMC example with the help from the other members of the group.

2) Michael will set up a web site as PIF example repository.

3) Each group (VDT, TOVE, PH, O-PLAN(?)) will contribute one example to the repository

4) Work out and implement a story of how PIF is used among our groups

Session 4: Extending the PIF-CORE:

=====================================

Summarized by Jintae Lee, Univ. of Hawaii. & MIT

The goal of this last session was to articulate how the Partially Shared View (PSV) mechanism can be used to best extend the PIF-CORE and accommodate expressive needs of different domains/tasks. We proceeded first by articulating the semantics underlying PSV, followed by discussing the logistic aspect of the PSV mechanism, structuring the space of potential PSV modules, and then finally deciding which of the modules should be responsible for the issues yet outstanding. Each of these topics is discussed in the rest of this report.

PSV Semantics

The discussion mostly focused on the nature of a view in PSV. For example, is a view more than a grouping of classes? What is the nature of relation between the views? When one adopts a view, is one also adopting its parent views?

The major conclusion we have arrived at can be summarized as follows. A view is in essence a grouping construct which helps locate/use a set of classes and prevent naming conflicts by providing a local name space, but which does not add anything extra to the semantics underlying classes. The relation between views is not an inheritance relation but a uses relation which provides information about where to find the non-local definition and axioms about the classes currently adopted. When the user adopts a view, i.e. the set of classes that it groups, its parent views are NOT automatically adopted. In particular one cannot use any class in a view that one has not explicitly adopted. The only thing that a parent view provides is the definition and axioms about the attributes not locally present in the classes in its children views being adopted. Once this information is inherited down to the classes in the view being adopted, then its parent views are no longer relevant. Therefore, one cannot use a class in a view even if the view is a parent of another view that has been adopted, but one has to explicitly specify the parent view in the list of adopted views in order to use any of its classes.

PSV Logistic

We agreed that there should be a (Web-based) central PSV registration server which ensures unique id for every PSV module. We also agreed that although the modules could be distributed over the network, it should keep a copy of every module to ensure consistency. We agreed that it would be impossible to attempt quality control on the modules being registered, but that it might still be a good idea to provide a collaborative evaluation mechanism in which people can post their comments on the modules. There was a question about how to make clear what each module is good for. Austin Tate pointed out that one way of doing it is through pictures that make the overall structure of the modules clear, e.g. such as Fig. 3 and Fig. 4 drawn for the IDEF-0 and the temporal-relation-1 PSV modules.

Keith Swenson volunteered to explore the possibility that the ACM SIGOIS (Special Interest Group on Office Information System) will maintain the PIF-related Web pages. Adam Farquhar pointed out that the KSL ontology group is beginning to deal with a similar set of logistic issues and suggested that we do not worry about them much until we actually have to face them. It is possible that the PIF group can simply adopt the solutions that the KSL group might develop by then.

Structure of PSV Space

Based on the discussions in the earlier parts of the workshop, Jintae Lee has compiled the following list of areas where the PIF needs to be more expressive.
control flow: conditional (XOR), iteration

domain-specific: e.g. computer application
  organizational: group
  inter-agent relations: authority, liability, ..
temporal relation (e.g. calendar, duration)
  metric
  relative relation
operations
  arithmetic
  financial
task-specific
  Workflow
documentation (e.g. creator, modified-by, ..)
expression: variable, quantifier

Based on his INOVA model, Austin suggested the following structuring of the PSV space.
Activity
  Timepoint
    Temporal Relation
      Metric
      Relative
        James Allen
Variable Relation (relationship on things defined using variables -- equal, greater than, ..)
Auxilary
  Authority (Control)
  W.S. S.O.A.
  Resources (Mechanism)
  Spatial Relation
  Miscellaneous
Issue
Node

Then we discussed how the two lists match up. It seemed that the following is not captured in the latter, INOVA, model: agent-agent relation, expression, and documentation.

There was also a question about whether all the control flow could be expressed if we had a rich temporal relationship module. Pat Hayes cautioned that we have to be careful about extending the areas of control flow and expression because they can change the semantics of the existing PIF drastically (e.g. recursion, quantifier). We seemed to have the consensus that extending the PIF expression itself, such as introducing quantifiers, cannot be done by a PSV module because it changes the PIF syntax itself. Another issue was whether to make such basic relations or functions as equal, greater-than, or plus, should be a part of the PIF-CORE. We decided that the PIF-CORE should include the following relations explicitly: equal, not equal, greater-than, less-than, greater-than-or-equal-to, less-than-or-equal-to. Arithmetic functions such as plus and minus, however, would be in a separate module used by the PIF-CORE.

We concluded that based on this discussion, we should provide an overall structuring of the PSV modules as a guideline for the people building PSV modules.

Remaining Issues

The major remaining issue was on the scope of PIF-CORE -- whether it should include constructs such as duration and resource. Keith Swenson suggested, and we agreed, that the best way of dealing with these issues is in fact to actually develop the WfMC module, which is to provide the interoperability between the PIF and the WfMC models, and see which parts could be potentially generic.