SWSC F2F Meeting
Thurs-Sat April 10-12, 2003

Minutes of Language Committee - Friday April 11 & Saturday April 12, 2003

Recorded by Robert Ross, Bijan Parsia, Sheila McIlraith and Mike Uschold

Note – appended please find an addendum with notes taken during this meeting by Benjamin Grosof.

Participants

Murray Burke
Stefan Decker
Dieter Fensel
Benjamin Grosof
Michael Gruninger
Jim Hendler
Michael Kifer
David Martin
Sheila McIlraith
Bijan Parsia
Terry Payne
Robert Ross
Karl Aberer

Not present

Jonathan Dell
Frank Leymann
Austin Tate
Frank van Harmelen

Discussion Friday April 11, 2003:

Primary topics discussed during this Language Committee meeting:

1. Process Model for Semantic Language: Generic/Modular approach or more focused using BEPL as an initial base
2. Define scope/mission of Language Committee & understand how it relates to scope/mission of architecture committee
3. Whether to build on top of DAML-S & OWL, w/ extensions or if these are too limited to build upon in an effective way.
4. Technical Presentations
   a. Overview of Process Ontology, Michael Gruninger
   b. Basic Process Modeling for Semantic Web Services, Michael Kifer
   c. Connection Between Rules & Services, Ben Grosof

The presentations can be found here:

http://www.daml.org/services/swsl/materials/F2F-April-2003/

Note – a "?" before the name indicates the statement may not have been accurately recorded. RR

MichaelG: an important issue is the relationship b/w SWSC & other Web services efforts such as BEPL
Sheila: review of web services stack would be a good starting point
JimL: each web services effort has their own stack (eg. WSDL, BEPL)
Michael: how does WSDL, BEPL, etc. deal with semantic Web services?
Jim: important to understand grounding, profile, process

Dieter: this is the task of the architecture group, language committee is responsible for language

Sheila: need to understand what languages exist – language needs to be understood in concert with architecture - DAML+OIL & DAML-S are appropriate tasks & this committee should understand where they don’t work & understand how to expand DAML-S or propose other alternatives

? Murray: a start should be w/ DAML-S, DAML-R. need to understand what it means to get the language out e.g. W3C, OASIS, etc

Ben: re: starting w/ DAML-S – what are the deep procedural issues of process models – sequencing constraints, exception handling, etc. A good approach may be to be modular and not get too restrictive – question – decoupling of KR & procedural

Shelia: KR can represent procedural, etc. – what do you mean?

Ben: where is our core competence – need to define knowledge base relationship to procedural

Sheila: need to figure out if we have the core competence or get it from outside

Murray: we will likely need to accommodate to a few standards

Karl: process model is syntactic part – question is how to integrate semantics into process language construct

Jim: if goal is real output, scientific research & experimental systems are helpful (papers, etc.) – but we need to document the established methods & have this language group establish itself in a documented format – e.g.:

1. Create a W3C note by the DAML-S committee
2. Create a tech report
3. Have reviewers (friends, etc.) review
4. Create a consensus & document the KR around the stuff that’s easy – not necessarily tackle the hard stuff
5. This committee would review the DAML-S documentation

Dieter: DAML-S still has several issues

1. Expressive enough to capture functional
2. State based
3. QoS
4. How to adopt it for architecture w/ architecture group & a couple of European projects

Stefan: need to separate process model (eg. existing models) from SWSC language design

Sheila: the DAML-S original intention was to separate process from language

David: variability of process modeling but look at some “defacto standard”

Sheila: DAML+OIL – defines vocabulary but not process model & DAML-S process model does not have well defined semantics

Stefan: this s/b input for arch committee – design architecture to allow for arbitrary process model

Terry: need to explicitly describe how process model binds to semantics

Bijan: modularity isn’t likely the best approach – accommodating everyone w/ complex process model will be really hard & result will be unsatisfying to everyone. May lead to complex scenario – alternative would be design a process model that hits a sweet spot

Terry: a little modularity would be helpful to provide enough generality to address 2-3 scenarios – i.e. 1 sweet spot may not be enough

Jim: we should create a list of requirements – eg. process model

Stefan: requirements to process model – perhaps not as complex as BEPL – it may not be possible to give semantics to BEPL that would be useful

David: we shouldn’t just rubberstamp BEPL, but understand if there are flaws in it

Ben: find a sweet spot that has social/business value to get traction of the whole area; initial approach should be eg. simple composition; another example is rules that trigger action w/o state change – demonstrate a semantics heavy approach rather than just a procedural approach.

Sheila: discovery & invocation was the sweet spot of DAML-S
Stefan: question - what do rules have to do with semantics
Ben: e.g. If you return the item you're entitled to a 90% refund – you can reason it. Group of actions.
Bijan: some procedural elements are more semantically synergistic – e.g. composition & reasoning thru effects
Stefan: what are the requirements?
Dieter: DAML-S needs to fit to architecture committee & for some current European projects; DAML-S may not be functional enough eg. declarative expression of a web service & procedural issues
Jim: OWL is not a representation language. s/n say OWL can't do X, say instead what we will add to OWL
Sheila: OWL as a reasoning language, etc. may not be enough
Jim: suggestion is to build on top of OWL as extensions
David: building on top of OWL may be hard
Jim: e.g. OWL doesn't have a rules model, but you could incorporate one into OWL; building on top of OWL vs. using OWL only; a difficult thing in OWL make it both syntactic & semantic extension to RDF, but it was helpful rather than starting from scratch – entire RDF community built tools that OWL leveraged. OWL can have a service language on top of OWL. If you start from scratch with a process model & presentation model – you lose because of early momentum of OWL
Karl: issue is design vs. implementation
Jim: suggest this group define syntactic & semantic extension to OWL – e.g. OWL-DL
Terry: if there's something we can't do w OWL, we define a new mechanism
Jim: OWL has no notion of process, but it can express it syntactically, above expressivity, eg. quantification
MichaelG: perhaps there is no way to achieve a particular goal w/in OWL
Sheila: xline was based on pi calculus, wsdl was built on top of [something] so why not start with [something] for process model [scribe: I believe latter [something] is BPEL]
Stefan: OWL is an object model
Stefan: this is a side issue – I don't think RuleML is compatible w/ OWL
Ben: RuleML *IS* intended to be compatible w/ OWL
Dieter: OWL & DAML-S are starting points – now, where do we go from here – e.g. functional spec of web services (eg pre/post conditions), describe relationship of inputs & outputs, QoS
? Sheila: use OWL to describe a process model as opposed to OWL being a process model

Presentation: Semantic Web Services by Karl Aberer
http://www.daml.org/services/swsl/materials/F2F-April-2003/

Sheila: interesting thing to look at is what an easy syntax is for a process model
Ben XML & RDF example isn't necessarily good because XML didn't address conceptualization, etc.
Karl: XML vs. RDF is communication vs. conceptualization (i.e. they were built for separate reasons)
Ben: Simple to understand is the key
Karl: agree – workflow products are successful because of simplicity
Sheila: agree

Stefan: giving BEPL semantics directly would be useless because WSFL was workflow & XLANG was programming - need to go up to a higher order of abstraction

Dieter: need to identify meaningful subsets of BEPL to apply semantics

? David: maybe hopeless/useless to do this for BEPL; add semantic expressiveness to [something]

Stefan: Pre/post conditions, post, sequencing – eg. very simple workflow model; Abstraction of BEPL to apply it to

Jim: BEPL is incomplete, so can’t combine it with anything right now; with a simpler process model can do analysis, etc. – even DAML-S process model is too complex; enigmatic uses Pi calculus eg. protocol analysis applied to business process – lots of examples of semantic use w/in business processes

Karl: given a process model, certain things apply;

MichaelK: separation of process and internal/external components

Karl: software verification people perhaps should be invited to participate

Sheila: web service discovery, invocation, integration

Ben: need to distinguish semantics from the reasoning that is done with that semantics

MichaelK: s/w verification may or may not have a strong business case

Ben: could have fewer features in initial version, and grow from there

Jim: requirements vs. goal – requirement defined as: if we can’t do this, we’re not done goal defined as: we’d like to do this, but we don’t know yet how need to have a balance b/w requirements & goals

Presentation: DAML-S Breifing by David Martin, Sheila McIlraith, Terry Payne
http://www.daml.org/services/swsl/materials/F2F-April-2003/

Presentation: Web Services Language: Scope & Objectives, by Sheila McIlraith
http://www.daml.org/services/swsl/materials/F2F-April-2003/

******** minutes scribed below by Bijan Parsia (until section break)

Murray: Consensus was leave Profile and Grounding to DAML-S, go for ProcessModel

Terry: Caveat: But Ben is affecting profile. We can focus on ProcessModel

MichaelK: Agreed

Bijan: Wait, are we wedded to *Process* models for Service Modeling

MichaelK: We need a process model for modeling *composition*. Composing language can be used for building services.

Bijan: Er..I meant composition, not implementation. Like what Ben was talking about.

Ben: You can use my procedure attachment stuff to implement *and* compose

Terry: Composing services produce meta-service. [Argh. Scribe isn't following] Composition vs. something else. There's something about state.

Chaos ensued. What is composition?

Karl: Comp has two dimensions The structure of the composition (BPEL) and the semantics of the composition.

DavidM: Manual vs. Automatic?

Karl: The first is about syntactical, structural compatibility/composition. The second is about reasoning about what is done when why.

Michael: BPEL is like Relational Algebra, other thing like SQL.

Terry: Do we want to articulate the role of semantics and reasoning in processes. Simply to do composition? To combine models?? Role of semantics: I can stick two things together?

MichaelK: Related to planning. What is discovery? Might not match unless you look at some ontologies and pull things together.

Terry: Are we talking profiles or processes? Once we decide what type of reasoning, how does that fit into use cases?

DavidM: Sounds like you're asking us to get clear on requirements?

Terry: Yes, if verbosely.

DavidM: We'll do the requirements tomorrow.

Ben: Is a service description best thought of (grounding is about connecting to base level invocation) something including directly executable portions, if so we need to decide what kind of reasoning goes with which bits. We may need more notions besides process and profile, and more levels to track the different sorts of reasoning with regard.

Karl: Requirement, partial specification, objective, executable ones

Bijan: We should have a requirement for some executable ones

Ben: Lets dissolve the distinction. Can have execution for perhaps limited expressive classes.

MichaelK: Indeed: don't formalize full bepl, but something less, might be ok.

DavidM: Two possible requirement: Our language is fully executable (or, give a formal semantics for all BPEL). Anything in our process model is executed. At least some process model is executable.

Ben (& others): Notion of executable: Heard of two web services and I can do one after the other. one is someone prepares a presentation, the other someone translates from german to english. If I now say that I now have an "executable model": call the first one, then call the second one. Executable is at some point you can get it to run. Thus, executable is easy to achieve. What kind of expressiveness for the executable portions of our language? And no one will write service descriptions that don't tie to execution. No one writes documentations.

Terry: very few people write WSDL, they derive it. We need to do that for semantics.

Karl: This is probably why more semantics is express in XML Schema than in rdf.

An interlude in which our scribe publicly bemoans his lack of scribing when compared to Benjamin "The Notetaking Madman" Grossoff.

MichaelK: [something]

DavidM: Paraphrase: We're going to define the semantic subset of BPEL.

MichaelK: No, I mean bindings. "Think of BPEL as .NET 2.0"

DavidM: But if our language product is at a different level. Ours is semantic analysis, BPEL is syntax.

Bijan: But BPEL has a semantics.

MichaelK: API for writing composite web services.

Bijan: BPEL is just a programming language. We don't want to do denotational semantics for a distributed imperative messy programming language.

MichaelK: Having trouble expressing my issue

Ben: One approach to SWS is to be weak and inclusive. There's such a thing as sws, and here's a language that let's you specify some stuff about how to specify semantic descriptions. Then provide some examples.

Bijan: Ick.

Ben: another approach: RuleML + OWL + What? E.g., transaction logic?

Terry: Let's stop.

Bijan: Yay!

DavidM: More discussion

******** minutes scribed below by Robert Ross

Jim: need to balance functionality w/ usability eg. failed w/ Kif. Tool ease of use is not always straightforward. Some people think that if OWL is difficult therefore an OWL tool is difficult

Ben: rules based agent example – some people like textual or graphical interfaces as long as people can use the underlying functionality & focus on cognitive aspect & formulate the challenging aspects like the underlying spec itself – most people can't deal w/ 1st order constructs, so it can make it very difficult if we don't make it something people can actually use. developers familiar w/ sub-class, column names, query, trigger, case statement

Jim: be careful to not make it so functional that it's not useable by the intended audience

David: complexity & familiarity are 2 different things. People are more accustomed to defining things procedurally rather than declaratively – & declarative languages have not had a lot of success. Expressivity & tractability trade off is known to us – but there is a trade off that needs to be managed between expressivity & people's willingness to use it (ie. if it's too complex to use).

MichaelK: OWL light is simple, yes, but is it useful. People learned Java, because there were a lot of uses.

Jim: DAML – people are using it for application integration

Ian: Does DAML+OIL have a strong enough base + extensions, or will it always be a struggle.

Sheila: Discovery & invocation, OWL is great, perhaps w/ slight extension, it will be good for composition, For execution & monitoring it won't be

Ian: Rules issue, they're sort of a first order logic, so setting up a Rules tower next to an Ontology tower might not be the right way to go – it may require entirely different (new) method; not sure it makes sense to have a rules tower – does this solve the process modeling problem

David: conditional expressions are one thing, algorithms, data typing are all separate things – perhaps we're looking for a combination

Jim: differentiator b/w this group & DAML-S group – profile & grounding would be good for DAML-S to focus on & this group could focus on process (BEPL in particular) and compatibility with OWL; And again, if there are tools out there already to use (eg. OWL) then try to use them, unless they aren't enough

Sheila: OWL good for defining interfaces, but more is needed for declarative, functional & behavioral descriptions.

David: build alongside of OWL

Ian: what is that going to look like, perhaps it's appropriate to create something entirely new

MichaelG: eg. DAML time; depends on what type of process you want to support – eg. contract expiration times & that can determine how high the stack (tower) goes

Bijan: focus on OWL layering

Sheila: puzzled by Dieter's comment that Architecture Comm. direct input to Language Comm.

Michael: eg. constraints for language committee to consider

David: Use cases are something

Presentation: Overview of Process Ontology, Michael Gruninger
http://www.daml.org/services/swsl/materials/F2F-April-2003/

Discussion during presentation:

Sheila: BEPL isn’t declarative, for composition you’d need to have something declarative
Karl: need to provide semantics to BEPL
Stefan: what should be the goal – not sure it should be something that has semantics or logic necessarily per se, but something simple that people can use & that later on can be automated – not necessarily the right goal to say need to have reasoning, process, etc. needs to be used
Keefer: shouldn’t be BEPL light
Sheila: well yes, we should be the semantic org
Stefan: a little semantics goes a long way – perhaps small stuff (jim ref)
Sheila: e.g. some small things that DAML-S can do
Terry: panacea solving all problems, or just something to start w/
MichaelK: little semantics for something big or lot of semantics for something small
Stefan: perhaps little semantics for something little

MichaelG: Presentation - Process Ontologies

More discussion during presentation:

Ben: ontology process is straightforward - spec, translation, authoring, inferencing, execution – does the Process Ontology area make it unnecessarily complex – so then eg. is RuleML executable – ie. these categories could be compressed
MichaelG: an important point to note is whether semantics are internal/external to the language
Ben: default assumption is that there are standard systems
Murray: no one (except a couple) are saying OWL is the end all be all, just that it is a good starting point
Ben: perhaps we should identify all the places that it DOES apply rather than all the places that it doesn’t (ie. will be too difficult to do latter)

Presentation: Basic Process Modeling for Semantic Web Services, by Michael Kifer
http://www.daml.org/services/swsl/materials/F2F-April-2003/

Slide from presentation:

Process Modeling Requirements

1. sequential composition
2. parallel composition
3. alternative executions
4. transition pre/post conditions

-----Supported by many formalisms with more or less the same semantics ------

5. subroutine mechanism process definition

-------------------------required minimum -------------------------------------

Constraints

6. Exceptions
7. Planning
8. Constraint Solving
9. Non-cooperative actors

Process Modeling Requirements (Cont)

1. Executable?
2. Specification only?

3. Both?

Discussion during presentation:

Ben: nice list
?David: is this built on top of BEPL
?MichaelK: BEPL would be built on top of this?
Bijan: we can use BEPL or influence BEPL
Someone: may want to involve Paco Caberos – IBM guy from Yorktown, Richard Goodwin (Ben ref) Yorktown doing something re: UDDI

Benjamin: IBM Watson & Zurich workflow may be appropriate to incorporate
Sheila: workflow would be good to add as that's a requirement & may not be a core competency w/in the current group
Dieter: mentioned evolving algebras in expressing state
MichaelK, MichaelG, Sheila & Ben – scribe didn't understand discussion
Sheila: industry standards should not shackle us - semantic recovery

Presentation: Connection Between Rules & Services, by Ben Grosof
http://www.daml.org/services/swsl/materials/F2F-April-2003/

Benjamin gave an example contract proposal – late delivery penalty exception handler module (see presentation)

Someone: Part of exception handling should be part of executable not just process model Selection, discovery, execution – question is how to map this model into daml-s

---

Discussion Saturday April 12, 2003

Summary of Friday discussions:
SWSL Requirements:
* support analysis
* executable process model
* declarative
* compositional
* well defined semantics
* understandable and useble (easy to build tools)
* supports automated reasoning
* not just interface descriptions, but functional and ??
* lifecycle management, process control
* extensible by others
* usable by ordinary folk, not just developers, possilby via tools, not directly
* get user community quickly
* incomplete service specifications
* easy to write tools to support language use
* process needs
* seque3ntial composition
* parallel composition
* ...

Goals/Objectives:
* guarantee-able post conditions

*   growth path to verification
*   enable multiple approaches for formal semantics
*   mappable to other languages
*   usable as interlingua for process languages
*   language to support:
*   constraints
*   exceptions
*   planning,
*   ...

Constraints on roadmap
*   DAML-S profile, grounded atomic processes: review/refine and get out quickly. Integrate with OWL

One concrete plan of action from MichaelK’s (MK)

- DAML-S will finish grounding and profile and pass it on to SWSL
- We need to figure out the process model and elaborate on whatever we get
- Common process model can be characterized w/ any formalism.
- Add contract to profile with rules (Benjamin’s)

BenG: not decided that profile needs to get out as fast as language. Also, guaranteed postconditions is very related to the contracts area that we discussed.

Murray: need to distinguish between hard requirements and what would be desired. Tradeoff between complexity/simplicit and usablility. Need to make more crisp the definitions of the requirements and desirements.

Sheila: (SM) process class defines class of all possible executions. Worreid about dong amythin much with process model IOPEs in the process model. Not necessarily reflect full complement of preconditions and effects. If process has multiple different trajectories, code compliles IOPEs, there are still complex conditional. (taken offline).
Overlap between profile and process because profile too complex, needed a shorthand.

SM: now, DAML-S describes services as part of a composite, but not other way round. Cannot look at atomic service and see what larger services it is part of.

Ben: easy to do that
MK some work on algebras on constraints.
Suppose want to do travel reservation, not care how.

MikeG (MG) what do you mean by 'more declarative'
MK: less procedural
MG: that's just as bad.

DavidK: (DK) we're talking about BPEL
SM: BPEL is a non-starter,

MG: declarative means anything consistent is possible. Is up to process model to say what semantics is.

BG: Sequential contiguity different from ? Are you saying this should be declarative ?
MG/SM: it has not been decided, but we believe this should be a requirement

MG: point is to stir things up so we know what issues are contentious.
BG: notion of declarative is overloaded,
SM: lets define what we mean.
MK: if you can specify it in some formal logic that maps into reasonable concepts, then it is declarative. But this is not so useful, because you can express BPEL as axioms.
SM: I like MG's definition better: anything that is consistent is possible
BG: MG's definition is fairly weak.
MG: you need to know how to do inference?
Karl: declarative means: introduce relationships among services to express dependencies on executin order, also can be used to specify process up to last detail.

Sum: three views of declarative.

Karl: we are a language committee. In other language committees, you adopt the standard, then map to other languages and semantics. e.g. as petri net, as logic program, whatever.

MK: this is a virtue we should aim for.
BG: at least have a large portion that is easily translatable.
SM: in practice, this virtue is likely to be unattainable, not want to be hampered by having this as a strong requirement, that could prevent us from having the expressive capabilities that we require.

BG: want both, not want to do all the work to get in all the mechanisms that we will ultimatly want.
MG: we need to crystalize this.
BG: work from the middle (in between lite and full)
Bijan: what is upper and lower bound?
MG: also a distinction between language and ontology. Daml-time is written partly in OWL and partly not in OWL. Might want a language extension mechanism, so you could capture more. Independent from that is an extension structure in the ontology. Say could extend linear time to dense time. Two different dimensions of extension.

MK: if we choose a core model, now suppose someone not like logic programs, but they like petri nets. So they want to extend into bigger petri net model. If it is possible for them to do this w/o being messy.

BG: workflow is a low-end point on a process model.

SD: I am concerned that we are going to define the KIF of process representations. Fine and nice, but is it deployable? What is the vision? DO we agree? My vision is: enable ordinary people to create, use and share services on the Web. Which bookshop within 5 miles has Lord of the Rings.

Critical:
*   simple process model
*   sharing Web service specifications (libraries, reuse)

*   get user community quickly
*   not necessarily industry/academics, may be grassroots
*   how to solve chicken/egg problem?

NOT critical

*   automated composition
*   deep semantics

Process primitives

*   dataflow, control flow, datastores, ontologies - as simple as possible
*   conceptual model that supports sharing and building critical mass
*   incomplete service specifications
*   link to remote service specification - import by reference
*   notion of adapter - so I can adapt an existing specification imported by reference

Concern that there is too much talk about complex things, keep it simple.

SM: DAML-S was designed to be used for composition, to do some stuff automatically, some in conjunction with humans.

Bijan: something sexy about sexy-automation, people retain control, also more tractable.
MK: not proposing as a requirement, to do hard things automatically (e.g. find me a hotel).

Bijan: there are different ways that DAML-S solves problems. Potential implementers say this is too complicated, overwhelming, too much concurrency... On other hand, want as much expressivity as possible, but then cannot write planner to handle this. Different restrictions on the language means some things are possilbe, and other things not.

TerryPayne (TP) what user, developer or end user?
MK: users should be developer, in long term
SD: yes, that's right.
MG: I oppose to claim that composition not needed.
SD: I mean initially, sure eventually it is important.
Katia: this is a good set of requirements for now, initial set. Idea is to come up with a set of incremental functionalities, start small, add more. Come up with language that supports the required functionalities.
SM: our job is to define a scripting language. Allow to expres relationships in DAML-S. Specify compositions in a GUI that generates DAML-S code automatically. Maybe we are not as far off from what you are proposing after all. Might be good starting point.
Bijan: two developers: end user developer, and people developing tools for end users. If it is too hard for people to build tools, then infrastructure will never come, unless we build it ourselves. Bad way to get end user simplicity is to put huge burden on tool developers. So make language easy for tool developers too.

DM: can we produce a straw man list of Requirements List?

<Insert Dave Martin’s Straw Requirements List >
(subsumed by final requirements list)

Katia Sycara (KS): DAML-S process model is a server-centric view. We need to fix this

KA: We want semantic interoperability

MG: Do we want to achieve semantic interoperability via writing in that language or as a translation

BG: Authoring language vs. interchange language
Imperialism vs. interoperability.
Do we need to support in a first class way, all these other systems, or assume that it will be encoded in our language initially. RuleML – what's exisitn, what's in common, how do we get things to go back and forth

MG: not necessarily compatible now, just let us get the list, check details later.

Orchestration vs. Choreography

Orchestration: Specifying the control strategy
Choreograpy: how multiple services are used together, specifying the linkages

MG: is language for authoring for everyone? If so, then semantic interoperability comes for free. OR is the language an interlingua to enable mapping to/from other languages.

-------- coffee break -----------

Agenda
- Outbrief
- Summary
- Lunch
- Administrative meeting

Please send slides for presentations to "the list"

<INSERT DAVE MARTIN'S OUTBRIEF SLIDES>
- summary of what we did
- summary of goals and objectives

Admin: Everyone should go to this URL to subscribe to the SWSC mailing list: http://informatik.uibk.ac.at/mailman/listinfo/swsc-list

<INSERT MARK BURSTEIN'S OUTBRIEF SLIDES>

Mark Burstein (MB) talked about interoperation frameworks being built in industry and how they fit nicely into the WSMF framework.

DM: What languages do these existing interoperation architectures use?

Christoph Bussler (CB): Everyone has their own workflow language. Talk about workflow, dataflow and controlflow

All these mappings are NECESSARILY partial (much is private).

Can’t define a universal ontology. Just do it for what you need. Very much like Gio Widederhold’s notion of mediators.

High level conclusion of value-added: Using semantics to reason about and reuse what the processes , what the object being discussed are *between* organizations so that they don’t have to be done over and over again individually.

Whose ontologies are these? Maintaining incrementally developed

Benjamin suggested replacing “ontologies’ with “knowledge” Think we’re endangered of focusing too much on where we’ve been rather than where we’re going.

%%%

% Notes from Face-to-Face meeting on 4/10-12/2003 of
% Semantic Web Services Coalition / Initiative
% by Benjamin Grosof

%%%%%

o organizational issues

SWS Initiative (SWSI)

website: http://swsc.semanticweb.org

mailing list: swsc-list@uibk.ac.at for steering/coord committee in past,
but soon to be more open;   (?also: swsc@semanticweb.org)

SWSL Lang comm:

init SWSL site: www.daml.org/services/swsl

swsl-committee@daml.org
- private archived list started on 3/7/03
- still need public list

%%%

W3C relations:
- W3C Interest Group on SWS has been established;
use www-ws mailing list is proposed;
Hugo Haas is contact, maybe we can have another mailing list with specific name indicating semantic web services, is presenting to W3C team
for now
- WS Arch WG
- BOF at W3C Plenary

TimBl enthusiastic, sugg's we aim to be on path to form WG

Dieter: we should consider alternatives to W3C too, cf. general WS scene

Benjamin: yes, e.g., Oasis; could help us get industry visibility

Dieter: we should follow where IBM and Microsoft want to go,
e.g., have been in touch with Frank Leymann about this

JimH: but we need to have the right stuff first;
W3C Choreography will probably obsolesce our process stuff

Lang.and Arch. committees each now meeting once per week,
plus F2F's at DAML PI Mtgs and OntoWeb Mtgs, plus ISWC

DanC: all can participate in the new SWS Interest Group of the W3C

Benjamin: want to be clearer on:
1. boundary b/ Lang and Arch areas/efforts,
and coord/info mechanisms b/ those, e.g., about use cases
2. near-term collective focus of say 80% of energy,
esp. given this is a complex area and
we're at a relatively early point in terms of fundamental research

Dieter: a way to proceed, cf. past EU SWWS effort, is to create
a framework and architecture
- the one from that is a good starting point for Arch. effort

WSMF

Murray: 3 basic tools for Lang. effort:
1. OWL
2. DAML-S
3. RuleML / Rules -- what can vs. can't handle

JimH: think "down-scope", pick a few things and get moving

Katia: analogy of our efforts to W3C WG's
- WS Description WG: WSDL
- WS Architecture WG

Austin: can start from requirements

Benjamin: let's think in terms of opportunities and challenges,
before focusing on requirements.
- e.g., automated discovery is probably not the most important task in near term for most industry users of web services, rather operations and maintenanc
- but we also need to come from the supply side -- what we know how to do,
and maybe discovery is strong in that respect
- we need strategy

Katia: e.g., can have strategy of piloting auto discovery first in the grid computing community, then take to industry later

Dieter: our main priority is simply to ALIGN the already existing large
efforts in US and EU;
big industry players, e.g., IBM, Iona, etc.,
already see potential importance of SWS;
wrt biz case: to achieve the promise of cheaper and more scalable e-commerce

%%%

SWSL [Michael Kifer]

todo: get Michael K's slides

*see SWSL page -> research resources

- todo: send stuff from me

will be driven by use cases

repair / recovery on-the-fly: Grid services folks really want [Austin Tate]

%%%

SWS Arch. [Chris Bussler]

todo: get Chris B's slides

use cases

*todo: do I need to join the Arch grp -- to do use cases?

see architectural scope doc by W3C from the WS Arch WG

is the notion of process the right abstraction,
e.g., BPEL4WS etc.

do SWS have to include "mediation" as a concept:
discovery, translation, name services, role services;
e.g., maybe it's part of the "intelligent network";
to what extent is it tied to partic org's

what is role of org's, e.g. biz's, on the provision of sets of services

will service descr's for discov be sim to web crawling or "repository-like"

how is security addressed

how is service compensation and recovery to be addressed

%%%%%

Industrial Committee [Mike Uschold] michael.f.uschold@boeing.com

Mike U. is co-chair and sole member so far

nominations open!

*todo: nominate Richard Goodwin, maybe someone from Sun or H-P or SAP

aiming for 5-10 active members, but maybe no limit on affiliated, eg 30+
- some come to f2f's hopefully
- Sheila: let's have two tiers of participation
- Katia: ... yes but informally
- Benjamin: ... yes -- have it be emergent who's more active when

hope to get input from them on requirements etc.

Benjamin: can set up with structure similar to RuleML participants
with appearance of their name/inst/logo on our website

JimH: they would like to get educated -- show demos -- have outreach --
can

Rob Ross: can have them describe/present the biz case / value in their
industry.

Benjamin: could have a webcast event, separate or with

Sheila: could have outreach with big events, e.g, WWW-2003
Benjamin: ... and trade shows/conferences

BOF at WWW-2003?
JimH: we'd have to find our own place and organize it
*todo: help ?organize one

Mike U.: wrt use cases and where to sell:
Boeing responses:
e-commerce, autom cust procurem, prod stds, EE stds,
commercial airline services;
emph open stds and platf-indep, incl ebXML, OMG XMI;
concern about IBM, MS, Sun, etc. backing out of efforts in WS stds
efforts

Dieter: SWWS has a large industrial board of 70 members, it would
be natural to take that over

*todo: talk to Dieter about me helping getting those co's involved

%%%

coord issues b/ difft committees:

Benjamin: would be helpful to have web archives readable by all in
SWSC of:
1. each committee's minutes
2. each committee's next telecon's agenda

JimH: yes, and have coord calls once per month

David: we want to decide here at f2f on public-ness of which mailing
lists
and archives

%%%

SWSL session:

Rob Ross of SF area, robross@yahoo.com, invited by Christoph Bussler,
is note taker for minutes

agenda:
- position statements
- some technical presentations:

. Mike Gruninger process ontol's
. DAML-S
. Michael Kifer on process composition
. Benjamin on rules in relnsh to services

%%
my 4/3 email:

1. here's a proposal of something I think it would be helpful to give people background on:

o how SW rules+ontologies relate to procedural aspects.
    - situated logic programs (SLP) abstraction of event-condition-action rules and OPS5 production rules.
      NB: SLP is supported in RuleML and (basically too in) Jess.
        . actions (invoke external procedures) triggered by inferring of conclusions
        . queries (invoke external procedures) performed during testing of rule antecedent conditions
    - built-ins used in rules and ontologies, e.g., arithmetic and comparison operators/functions
    - exception handling in workflows and service agreements/contracts

I could do this in 10-15 minutes, then we could have a bit of Q+A/discussion.

2. Wrt other stuff: as I and some others said on the call today, I do think it would be nice to have a summary of BPEL4WS, the Choreography-type-stuff glossary Bijan pointed us to, and Pi-calculus. Say at least 15min on each by someone who knows it well enough to summarize the highlights / most-relevant aspects.
%%

Jim H:
curr DAML-S:
grounding
profile
process model

but in last yr, WSDL became the winner on grounding

JimH:
recommend have a W3C Note on better part of DAML-S stuff

Karl:
key is DYNAMIC aspects of processes/services, difft from static semantics
of documents cf. SW

Dieter: DAML-S needs to be extended:
- functional aspects
- state-based
- quality of service -- very impt for biz
- how to adopt it for the kind of architecture in couple EU projects
  . may require restructuring

Benjamin:

want to decouple semantics/KR -- our core competence -- from procedural aspects of process models -- which others will fight out; make the latter pluggability

Stefan: yes, and this is input to the Arch. committee

Sheila: e.g., have a plug-in to Petri Nets,
consistent with the original vision of DAML-S approach to process model

David: envisioned pluggability, but created a default approach,
sort of like SOAP is default approach to use with WSDL

Sheila: problem is that OWL is not expressive enough to define the semantics of the process model; there's been a translation of operational
semantics to Petri Nets, and to situation calculus (?), but...

Terry: we need examples of this with translation to Petri Nets or whatever

Bijan: this modularity is hard to achieve
Sheila: yes, PSL is trying to do this

if want stds, then research should be done

Bijan: let's have one sweet spot

Sheila: BPEL4WS is maybe the best choice now; doesn't have a formal semantics, but have been told it's some combo of Petri Nets and Pi-calculus

Rob Ross: IBM is pushing this really hard to developers (e.g., latest Dev'ers conf)

Terry: some stuff is useful even without this aspect, e.g.,
just discovery to do it better than UDDI

JimH: OK to be partial wrt coverage in a draft release

Benjamin:

let's have a part of initial version
that's initially shallow wrt procedural aspects,
can still get value and reuse for several basic kinds of tasks;
e.g., negotiation, what-if'ing, monitoring, discovery

- Sheila: yes, and invocation

be semantics-heavy and procedural-light

don't let the deep be the enemy of the good

Bijan: but procedural aspects are central for composition, which gets people excited

JimH: analogy: was valuable to get OWL to be a semantic and syntactic extension of RDF, e.g., so RDF tools get something out of OWL;
would like our lang to be an extension of OWL

%%%

Requirements:

Dieter:
1. functional specifications of services: rich predicates and statements about relationship of inputs to outputs
   - OK if cf. program verification or process modeling approaches
   - be layered: e.g., in terms of OWL, in terms of more
   - be state based
2. quality of service

Michael Kifer: is OWL extendable, or broken for this purpose

Bijan: is rules layer going to be rich enough? is the logic layer going to be rich enough?

Michael Kifer: don't like the syntax of DL -- is completely broken in that FOL has been around for a long time; is unfamiliar to most, thus makes it hard to represent processes; not as nicely compositional either

Karl Aberer of EPFL:

*requirements:
give expressive semantics of state in whatever process model;
function, state, input and output should be Web-accessible resources;
distinguish clearly between syntactic and semantic levels of descriptions of processes

how can we carry on the XML vs. RDF concerns
in the WS area -- XML for exchange, RDF for data model

note that most of the current process model approaches for the Web
do not explicitly model state, but only sequences of calls or messages;
so is similar to XML's emph on serializ;
we need to give explicit semantics to state, somewhat similar to RDF in spirit

stateless WS:

functions addressible and accesssible thru Web stds,
URI's, msg i/f's

f(I):O synt repn given eg in WSDL

sem descr:
- f, I, O resources, e.g., in RDF classes

stateful WS:

state acccesible thru Web-addressable functions

functions have side effects on state

issues:
- how model + sem of states and transitions

- is a synt proc model req'd, ir is this just internal to the WS
- serializ of the descr

ex. of sem of states:
- need use cases
- e.g., travel tells if reach state s, then the airline booking has been done

todo: get Karl's slides

Benjamin: I have deep misgivings about us getting deep into semantics of process models -- decades of work on formalizing that and verification
and automatatic programming and semantics of programming languages
-- to 1st order has no practical impact, it's just a very hard/complex area;
OWL is a pimple on an elephant in comparison;
we should focus on quite simple stuff

Karl: workflow is an exception, since is simpler
Benjamin and All: yes

Sheila: it's simple for a user to understand

David: does its formal semantics have impact

Michael: not much, it's simple enough that users don't need to know it

Michael Kifer:

WSFL, XLang based on Pi-calculus, is really a programming lang

BPEL has both, it's like Java; giving it semantics would be a useless exercise unless we go up to a pretty high level (of abstraction)

but maybe we can find a nice simple subset of BPEL;
thinking about:
- precondns;
- postcondns;
- sequence and parallelism;
- perhaps also: looping

JimH: biz proc automation analysis at choreography level,
e.g., showing deadlock, livelock, guarantee of a deliverable or account decrementing will occur,
is growing in importance

semantics of a process   vs.   reasoning with it

Sheila: let's put up a list of what tasks like discovery, monitoring, etc.
- David: will be doing this aft.

%%% (lunch break)

let's use the www-ws mailing list for most technical stuff, posting links
to our briefings, etc.
- will be public, help get involvement, help convince W3C

%%%

DAML-S Briefing [David, Sheila, Terry]:

3 basics:
service profile - used for advertising, discovery, selection
service grounding - how to access it
service model - how it works

then gets mapped into concrete msg formats, specified in WSDL

profile: [Terry]
- non-functional properties, incl.
. serviceCategory, provenance, qualityRating, text description,
. serviceParameters that such as response time, geog. radius
   -- intent is that they're the set of negotiable parameters
      - Benj: not really the right set, tho' -- e.g., quality or actor
              are often negotiable param's in practice
. actors: requester, provider, third parties, ...
- functionality descr:
. param's: inputs, outputs, pre-conditions, effects (= post-conditions)

each param has:
. param name
. OWL/RDF type restriction
. refersTo pter into the process model

there's a sample profile hierarchy ontology

effects:
o should hold true if the service is invoked successfully
o often refers to real-world effects, e.g., book being delivered,
or credit card being debited

knowledge conditions?
A [Sheila]: output is knowledge effect;
post-conditions are world effects; pre is about world too

inputs and outputs: these in the profile are interpretations of
"salient" inputs and outputs
of the underlying process model

issue of what are the necessary knowledge conditions for the grounding i.e., to actually invoke/execute the service; these are distinct from the aspects represented (information-modeled) in the profile

issue of do we even need to have a representation of how the profile relates to the grounding, e.g., how the profile inputs/outputs relate to the grounding's inputs/outputs
- for some tasks / applications it's needed, for some it's not

Karl Q: can user tweak all this specific ontology
A: phil: it's an exemplar upper ontology
to show overarching approach, with example ontologies that flesh out, but no real firm expectation that a user has to stick to the ontology; have maybe dev'd more than nec detail for some, e.g., Actor

Benjamin: should be clearer on where it's OK to tweak, i.e., treat as an upper ontology part that's required and part that's offered as example;
this one includes some details that are too/problematically to be required

Terry: yes

DL-ish matchmaking can be made tricky by too detailed ontology;
want some test cases
- e.g., ad is more general in some regards, less general in other regards

```
. e.g., query "overnight ship from Canada to US" vs.
        ad    "ship from B.C. to California"
```

- often in practice matchmaking is done in two phases with one being in opposite generality direction or similarity
- a trick is to leave some parts of the query more general by saying don't care for a property

process model: [Sheila]

workflow-y + some planning

info providing or world changing

e.g., amazon.com

data flow and control flow are not well rep'd in DL

could use FOL, could use Petri Nets

rules seem reasonable

need/want a tool to present in a simple way to user authors
- much less risky if the "deep UI" of the KR is
already conceptually cognitively familiar to lots of developers/people

e.g., OWL Lite, Horn rules, relational database table columns,
basic DB queries, case statements, if-then-else

David: but people learn the relatively complex Java because it's so useful

Michael K: but how useful is OWL?

for discovery and invocation it's good;
for composition -- maybe;
for monitoring, it's not good

wrt whether need more than OWL:

Benjamin: developers use DEFAULT inheritance a lot

Michael K: generally, nonmon in rules

todo: we need a version of the WS Stack diagram that shows separately the interface I/O descr, rest of profile, grounding, and process model

%%%

Process Ontologies [Michael Gruninger] <gruning@nist.gov>

lang
model th (sem)
 - want at least soundness, sometimes completeness
axiomatiz
classif
reln b/ ontol and proc descr's

Process Ontol's:

o Common Plan Repn (with Austin Tate OPLAN) became ...
o Shared Planning and Activity Repn (SPAR)
o Planning Domain Descr. Lang (PDDL) - std used for AI Planning conf's
    - gets turned into other KR's, give grammars for various aspects
    - no formal semantics (?)
o Wkflow Proc Defn Lang (non-AI community)
o Biz Proc Modeling Lang (BPML)
    - claim to be formalizable in Pi-Calculus or something
    - claim BPEL is subset of BPML
o DAML-S
o Golog
    - has some spiritual relnsh to DAML-S -- FOL, has axiomatiz but no ontol
o Proc Spec Lang (PSL) -- latest, Michael G working a lot on this
    - has aspect that's identical to sit calc FOL

different application scenarios motivate different degrees of depth in formalizing semantics
- info sharing and interoperab:
    driver for PSL: common access to info w/ heterog vocab / format;
    ontol as basis for translation b/ them
- discovery: motivator for OWL in DAML-S
- neutral authoring: to support reuse in sharing and interoperab
- ontol as (non-exec'able) spec , e.g., some use PDDL this way in AI Planning

*interesting thing we should do as a community: [Murray]
find the expressive overlap of DL/OWL with PSL or other good can proc models,
esp their axiomatizations

PSL core theories
- subactivity, atomic activities, complex activities,
  occurrence trees, discrete state, activity occurrences

can distinguish b/:
precondns, conditional activities, triggers;
eg look at relnsh b/ activities and agreements on states, or at times

proc descr's are boolean combo's of stuff like precondns (incl. cf. Golog)

intended uses: planning, process modeling
- e.g., translated IDEF3 workflow-y model from Procap tool into
    a C++ ILOG scheduler -- applied for manufacturing, airline flights

all axiomatizations so far are in FOL in KIF syntax
- found that need pretty full expressiveness,
  partly to have interoperability

issues:
- how heavy does the "sem machinery" have to be?

%%%

Basic Process Modeling for SWS [Michael Kifer]:

req's:

1. sequential composn
2. parallel composn
3. alt exec's
4. transition precondns/postcondns

many formalisms support 1-4 w/ more or less same sem's

5. subroutine -- in manner of Prolog, e.g., as a predicate + its defn

1-5 is the req'd min in Michael K's opinion; is supported in his
Concurrent Transaction Logic formalism;
1-5 supported by many approaches with some restrictions

6. constr's
7. exceptions
8. planning
9. constr solving, e.g., achieve w/in a cost constr
10. non-cooperative actors, e.g., when contracts+laws, game-theoretic beh

6-10 is more stuff: but not as clear, not as near term;
some of it can be accomp'd with the same expressiveness as 1-5

overall wrt 1-10:

exec'able? and/or for spec only?

nice example: (contractor-) bid evaluation workflow control flow
- uses Constraint Transaction Logic (CTR)

CTR: w-c expon time, but linear time after compiling spec to w-c expon space

can view this as formal semantics and alt repn equivalent to
a fragment of BPEL (probably)

WSFL is fairly close to this -- lacks while statements and if statements

XLang has while statements and if statements

then the combo to form BPEL -- was it newly expressive or more political

wrt BPEL:
Frank Leymann and IBM Yorktown person (Paco Kobaris (sp?))
are interested, and have promised
to do a presentation on one of the SWSL telecons;
Frank is a member of the SWSL committee (?);
- Me: also Richard Goodwin, and there are others at IBM Zurich and Watson
who work on workflow

David: issue of how useful it is to provide a formal semantics for
only a subset of BPEL (or some successor wrt popularity)

Michael K: come at that issue from use cases

Stefan D: let's not surrender the process model space to BPEL, since
that will mean everyone pays license fees to IBM and MS

recursion -- hard to reason with it, reasoning becomes 2nd order

Sheila: work around is to do bounded iteration (sim to timeout)

%%%

***todo: think about the lots of good points/discussion below

How Rules+Ontologies relates to Procedural aspects of SWS: [Benjamin]

**todo: add a few SweetDeal scenario slides to the slide set

precondns, postcondns

rules+aproc as light workflow

process model view must take in aproc's in rule/SW settings
(both KB's and inferencing)

issue of state and time

**implic's of a partially executable contract ruleset use case
at paradigm level -- for
possibly modifying the DAML-S paradigm of division between service profile
vs. service process model

- at least would want to be able to point from profile to rulebase,
  and from process model to rulebase
- they overlap

exception handling can be viewed as part of a rule-based spec
at profile level, not just at process model level

contracts can be viewed as service descriptions

user/customer conceptual view of business processes / services today
is often cf. light workflow

light workflow also used in pub-sub, or info flow type applications
(news, workflow, info dissemination)

rule-b descr's/services as early low-hanging approach to
distinctively semantic WS; same rulebase descr useful for several SWS tasks
incl discovery, selection, negotiation, monitoring, exception handling

Michael K: verification: issue of tractability in reasoning to detect
whether a postcondn would be implied by composition plus precondns and
rule-based service descr's for the service and sub-services

Dieter: can use rules for "evolving algebras" for state-transition oriented
process descriptions - i.e., represent the states and time explicitly,
can have sequences of actions

Benjamin: yes for quite a bit, but rules lack some perspicuity for
representing some of that, compared to workflow languages

Dieter: error recovery w/ compensation is tough

- Michael K: should do in WS level not SWS

Karl: can view: a requirement for our SWS service description: is that
service descriptions are partial

also want executable for at least some simpler expressive cases

Benjamin: as consider different tasks for SWS, the aspects of
overall service descr that help with each task may overlap properly,
e.g., rule-based service descr's for e-contracts,
thus may want to define even more aspects of service descriptions
than just "profile" vs. "process model"

Bijan: we want to have an executable portion of the service description

Benjamin: executability in basic sense is easy -- e.g.,
composition "A;B" where have groundings of A and B WS's available;
so issue is really:
- what expressiveness do we want for executable aspect of our service descr's

Benjamin: for adoption by developers, it's critical that our service descr's
are part of the main spec that generates executable, otherwise dev'ers
won't invest the effort to gen the spec's and maintain them in the life cycle,
because they'll be viewed as extra

%%%

Terry:
interaction protocol vs. composition -- difft process models be needed?
i.e., will it be coherent to compose distributed processes into a
larger process, and present an interface to it as a single invokable process

Karl: reasoning to generate possible compositions vs. specifying the structure of compositions

%%%