Reason about learning resources sequencing

Module B can be selected only if the user knows also something about Mutual Exclusion; the user knows nothing about it.

The system can search for a second module that teaches the missing competence.

Building complete reading sequences

To this aim AI planning techniques can be used. A plan is a sequence of atomic actions (resources).

Two main approaches:

- Combine atomic actions (e.g., A*, graphplan)
- Produce plans that respect a schema: e.g., procedural planning (interesting in educational domains: procedures as learning strategies)

What is a learning strategy?

The organization of the material in a lesson or a course is not only up to prerequisites and effects but also to the experience of the lecturer.

Learning strategy: overall schedule of the topics, the view of the teacher of how topics should be sequenced.

My learning strategy for today:

- Know your teacher (briefly introducing myself)
- Motivation to this lesson: why should you care about adaptation, personalization and reasoning?
- Technicalities
  - User modeling (can you personalize the teaching to your needs?)
  - Adaptation by reasoning about actions and changes
    - Agents
    - Curriculum sequencing/planning
  - Web services / interaction protocols
- Lunch
Moving to the web ...

Task of adaptation and reasoning on the semantic web

identify a set of learning resources that fit in my learning strategy and that are the most suitable to the specific user

A specific case: using DyLOG to specify learning strategies

DyLOG, by Baldoni et al., is a language for programming agents, based on a modal approach for reasoning about actions and change

- Primitive actions: preconditions and effects
- Sensing actions: interaction with the world
- Prolog-like procedures: complex actions

A domain description consists of a set of primitive actions, a set of sensing actions, a set of complex actions, together with a set of initial observations.

Procedural planning

The search space is constrained by allowing only sequences of actions that are executions of a given procedure

plan: procedure execution

procedure: behavior schema

A specific case:

using DyLOG to specify learning strategies

DyLOG, by Baldoni et al., is a language for programming agents, based on a modal approach for reasoning about actions and change

- Primitive actions: preconditions and effects
- Sensing actions: interaction with the world
- Prolog-like procedures: complex actions

A domain description consists of a set of primitive actions, a set of sensing actions, a set of complex actions, together with a set of initial observations.

Learning resource as a DyLOG atomic action

processes concurrency Module A deadlock

moduleA possible if processes, concurrency
moduleA causes deadlock

processes concurrency mutual exclusion Module B deadlock deadlock avoidance

moduleB possible if processes, concurrency, mutual_exclusion
moduleB causes deadlock, deadlock_avoidance
Learning strategy as a procedure

The meaning of “basic notions of Comp. Sci.” depends on the kind of students you need to explain things to...

comp_sci_4_biolgists is db, statistical_tools
comp_sci_4_web_designers is web_programming, computer_networks

Given a specific student: the task is to produce a specific plan, i.e. sequence of actual learning resources, that will allow him/her to acquire the desired expertise.

Another educational application validation of the user’s choices

Procedural planning in DyLOG

Scenario

A user can browse a repository of learning resources and choose some that he/she likes and considers useful for achieving a given learning goal.

The user would like to know if the chosen material will lead him/her to the target.

Forms of reasoning

Plan validation + Explanation of failure
Plan validation + Explanation of failure

A set of assumptions is collected along the way; the plan is correct given that those assumptions hold.

The domain is monotonic:
- Each state includes the previous one.
- There is no forgetting.

LRi possible if a, b
LRi causes c

Conclusion:
- LRi cannot be applied unless user already knows b.

Links

Web Services

The Web as a provider of services

The web is evolving into a provider of services, resources that allow one to check/change the state of the world:
- Information providing services: weather forecast, flight arrival schedule, ...
- World-altering services: ticket booking, purchase, ...
- Web services: hard-/soft-ware devices accessible over the internet

There's a need for all these devices to interoperate.

So far, interoperation provided by code hardwired in the APIs for information extraction. APIs are strongly dependant on the structure of documents/resources they must handle, if the latter changes they are to be changed.

More flexibility is desirable ...
The Web as a provider of services

The Semantic Web initiative deals also with the problem of representing and remotely invoking web services

In order to obtain flexibility:
need for a computer-interpretable description of the services

Two main initiatives

- AI agent community: DAML-S / OWL-S (based on semantics)
- Commercial initiative WSDL / BPEL4WS (not based on semantics yet)

Once again the action metaphor

Some possible tasks

What do we wish to do (in an automatic way) with web services?

- Discovery
- Execution
- Composition & interoperation

Currently, none of these tasks can fully be executed (do you agree?)

Example: organize the journey to Aussois

Services I can find on the web:
- train ticket / flight
- hotel reservation

Supplied information: school period (June, 21st-25th), ticket class (economy)

My expectation: after task execution I will have a ticket that will allow me to arrive before the beginning of the school, and leave right after its end, and a reservation at some nice hotel

possible problems ...

Example: organize the journey to Aussois

Would you trust an agent to do it for you?

- The system finds a place called Aussois in Canada, thus it reserves a flight to Ottawa ... a little far!
- The system does not find a free seat on the 21st, but it finds a free seat on the 19th, it immediately makes a reservation; luckily the hotel has a free room starting from the 21st ... where to sleep before this date?
- Days of arrival and hotel reservation match, however the train arrives in Modane at 21:00 ... school is over
- ...
What kind of markup do we need for ...

Discovery? What does the service provide to users? Properties that classify the service

Execution? dataflow model / IOPE (inputs, outputs, prerequisites and effects)

Composition? business logic, how the service affects the world

---

Info in the profiles

- human-readable:Service name, text description and contact information
- functionality description: inputs and outputs (they can be conditional and refer to inputs and outputs of the process model), preconditions and effects (they can be conditional)
- profile attributes: category (ontology-based description)
- ...

---

OWL-S descriptions

Kinds of service

- primitive: a single device is invoked, little interaction with the user, a simple response is returned
- complex: composed of multiple services, a conversation with the user is required, the user can make choices

A service is described by

- service profile: used for advertising (and discovering) services
- process model: describes the service’s operations
- grounding: describes how to interoperate with the service by means of messages

---

http://www.daml.org/services/owl-s/1.0/BravoAirProfile.owl

```xml
  ...<specification of quality rating for profile ...
</profile:qualityRating>
<profile:rating rdf:ID="BravoAir-goodRating">
</profile:ratingName>
</profile:ratingValue>
</profile:qualityRating>

<specification of the service category using NAICS -->
<profile:serviceCategory>
<addParam:NAICS rdf:ID="NAICS category">
</profile:value>
Airline reservation services
</profile:value>
</profile:serviceCategory>
```

...
Info in the process model 1

- input/(conditional), output: information that is necessary for the computation to take place (may be provided by other processes), and that will be processed + info that is produced
- preconditions/(conditional) effects: related to changes occurring to the world by the execution of the service
- conditions: no mandatory language to be used (best candidates SWRL and DRS)
- ...

http://www.daml.org/services/owl-s/1.0/BravoAirProfile.owl

<| Descriptions of IOPEs |>
<profile hasInput rdf:resource="@ba_process;#DepartureAirport_In"/>
<profile hasInput rdf:resource="@ba_process;#ArrivalAirport_In"/>
<profile hasInput rdf:resource="@ba_process;#OutboundDate_In"/>
<profile hasInput rdf:resource="@ba_process;#InboundDate_In"/>
<profile hasInput rdf:resource="@ba_process;#PreferredFlightItinerary_In"/>
<profile hasOutput rdf:resource="@ba_process;#AvailableFlightItineraryList_Out"/>
<profile hasInput rdf:resource="@ba_process;#Password_In"/>
<profile hasInput rdf:resource="@ba_process;#ReservationId_In"/>
<profile hasOutput rdf:resource="@ba_process;#PreferredFlightItinerary_Out"/>
<profile hasOutput rdf:resource="@ba_process;#ReservationId_Out"/>
<profile hasOutput rdf:resource="@ba_process;#HaveSeat_Out"/>

Info in the process model 2

- Atomic processes: directly invocable, they execute in a single step
- Simple processes: not invocable, they are elements of abstraction that present a service as having a one-step execution
- Composite: decomposable, possible control constructs:
  - Split (concurrent execution), Unordered, Split+join
  - Sequence
  - Unordered
  - Choice, If-then-else
  - Iterate, repeat-until
Emphasis on the seeking agent

Once again these representations are thought for the automatic retrieval, invocation, etc performed by artificial entitle (agents)

Agent main characteristics

- autonomous
- has a state
- can sense the world
- decides how to act based on its state and its perceptions

  e.g. a thermostat!

- It is not necessarily rational :-(

The seeking agent

Two-steps search

use the profile
profiles stored / handled by registries for quick focus

use the model
used when selection is based on how the service is executed, and for composition

It requires reasoning !!
Example
I live out of town, need to move by bus
Agent Jeeves: search if a book is available in some library of Mytown,
and reserve it; please, pay attention to the bus schedule!

University library
requires: you are a student
pick up during opening
time: 9:00 – 16:00
reservation?

Town library
requires: you are registered
pick up during opening
time: 9:00 – 12:00 / 15:00 – 18:00
reservation?

Bus schedule: 6:00 or 13:00

The seeking agent: how to build it?
Requisite: it must be able to draw conclusions about programs described in a declarative way (the web services) and/or by interacting with them

Web services descriptions based on the action metaphor

Techniques for reasoning about actions and change are necessary

Examples: situation calculus, agent programming languages such as Golog and DyLOG

Example
I live out of town, need to move by bus
Agent Jeeves: search if a book is available in the libraries of Mytown,
and reserve it; please, pay attention to the bus schedule!
Agent task: build a plan that will allow my master to achieve her goal

Two library services are available, both have the book and my master can go to both ... which to choose?

Interaction with the bus information service leads to choose between the two libraries

The seeking agent: a sketch

Is this a multi-agent system?
The agent community is highly variable, organized on the fly depending on needs.

Agents let their behaviour be visible to the others.

They can reason on each other's behavior and foresee the effects of their actions.

Actions: world-affecting / information-providing

In order to foresee effects of information-providing actions the agents must be able to make assumptions on the “mental state” of their interlocutors.

Many approaches in the literature are based on variant of modal logic, in which mental attitudes, such as beliefs, goals and intentions, as well as communicative acts are represented by modalities.

Only recently the attention has been moved to formalize those aspects of communication that are related to the conversational context in which communicative acts occur.


What kind of reasoning is interesting?

Proof of existential properties

I can execute a protocol p in way such that afterwards the provider will not have this piece of information:

\( \langle p \rangle \models B \models \neg B_{\text{other info}} \)

Nested belief

I believe the “other” does not know info.

There is a possible execution of p such that...
DyLOG, the interaction protocols

Communication protocol: procedure that defines the communicative behavior of an agent (or a web service)

It is defined on the basis of predefined speech acts (atomic, communicative actions), for instance FIPA speech acts

Conversation: an execution of a communication protocol, a specific sequence of speech acts

What kind of reasoning is interesting?

Proof of universal properties

Properties of the multi-agent system as a whole
The overall behavior of the agent system is modeled
Among the various approaches:
use of Dynamic Linear Time Temporal Logic

Links

- S. A. McIlraith, T. C. Son, H. Zeng, Semantic Web Services, IEEE Intelligent Systems, 2001
- S. A. McIlraith, T. C. Son, Adapting Golog for composition of semantic web services
- The OWL Services Coalition, OWL-S: Semantic Markup for Web Services, white paper (http://www.daml.org)
- M. Baldoni, C. Baroglio, A. Martelli, V. Patt, Reasoning about self and others: communicating agents in a modal action logic, LNCS 2841 2003.
- Links to web service resources
  http://lsdis.cs.uga.edu/lib/presentations/SWSP-tutorial-resource.ht