DAML2Soar

A DAML+OIL Ontology to Soar Knowledge Translator

Sean Lisse

Soar Technology

The problem: Knowledge Representation

- How to represent abstract and/or long-term knowledge in Soar?
- How to share knowledge between agents, or between an agent and a non-Soar mechanism?
- How to allow SMEs to add or modify knowledge in an agent system?

Current Solution: Productions

- Create productions that elaborate declarative structures in Soar
  - Doesn't allow explicit processing of knowledge
    - No introspection, limited inferencing
  - Needs a new production for every information piece
    - SMEs will not and should not write Soar
    - Expensive to code and error-prone

Generated Productions?

- Maybe we should automate productions via TCL Script?
- Need a new TCL script for every data type - no standard representation of types
  - Still looks like/is code – not user-friendly
  - Difficult to tie different categories of data together (F/A-18 carries GBU-12)
  - No explicit relationships between data types (F/A-18 is a Plane, is a Vehicle)
  - Adding new types is difficult
    - New TCL Procedure needed
    - No inheritance (reuse)

What could be used to represent knowledge?

- *on·tol·o·gy*

  "The branch of metaphysics that deals with the nature of being."

1. <philosophy> A systematic account of Existence.
2. <artificial intelligence> (From philosophy) An explicit formal specification of how to represent the objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them.

The Opportunity

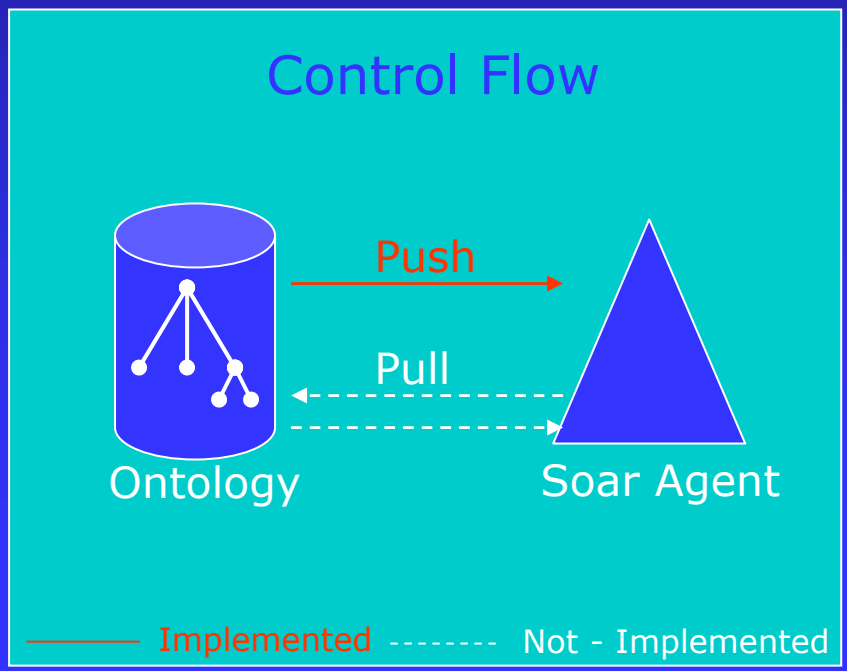
- Active research community in the area of Ontologies
- Many pre-existing Ontologies
- Many Ontology editors available

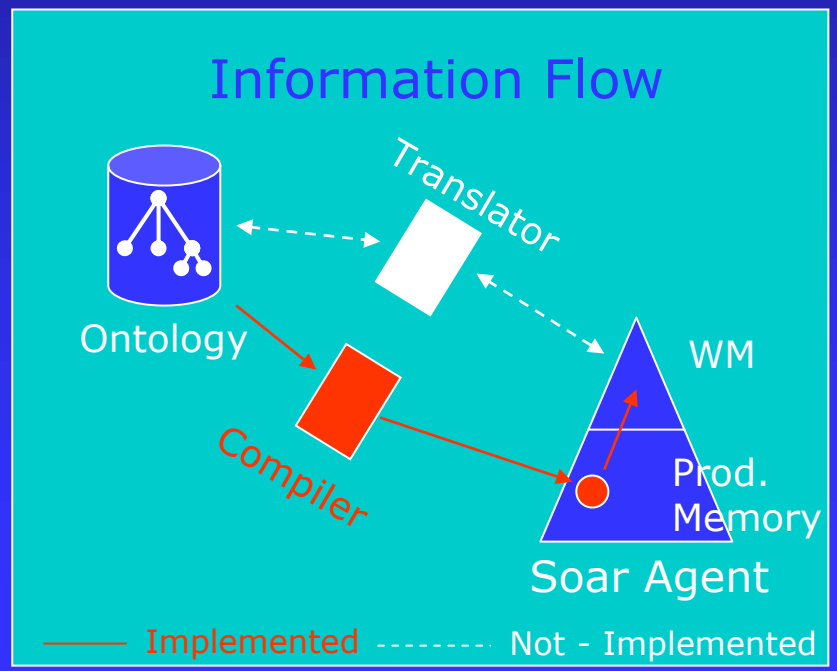
Which Ontology Format?

- DAML+OIL
  - Darpa Agent Modeling Language + Ontology Inference Layer
  - Has formalisms for
    - Classes + Class Hierarchies
    - Properties + Property Restrictions
    - Instances
    - And more…

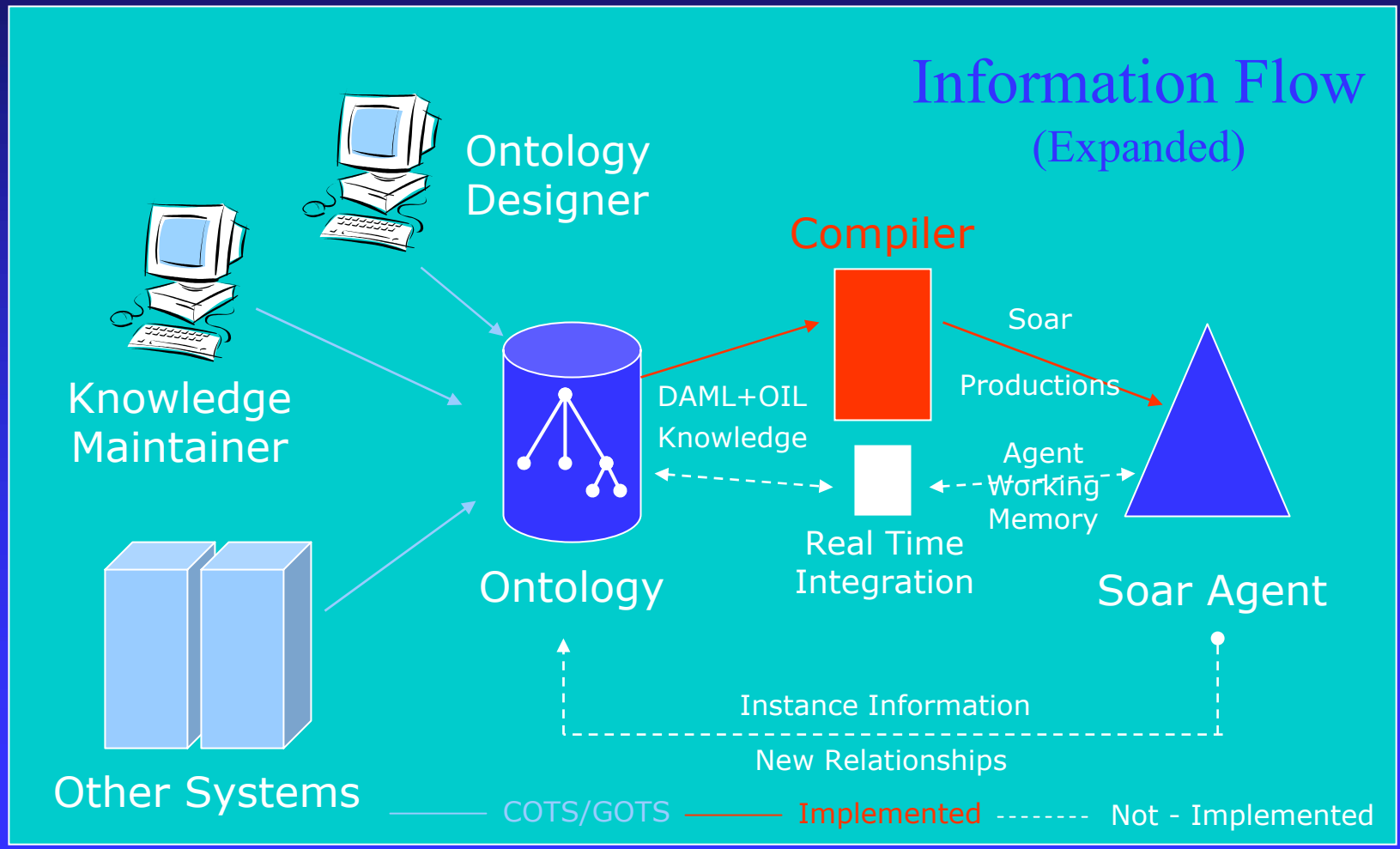
Translation Mechanism?

- One-way or Two-way?
- Compile-Time or Run-Time?





DAML2Soar v. 0.5

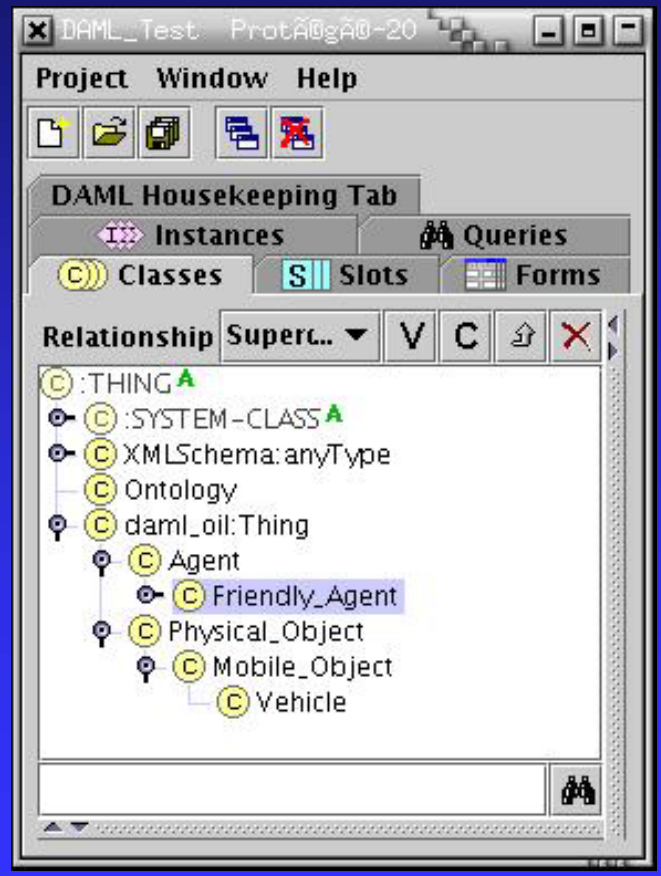


DAML2Soar v. 0.5

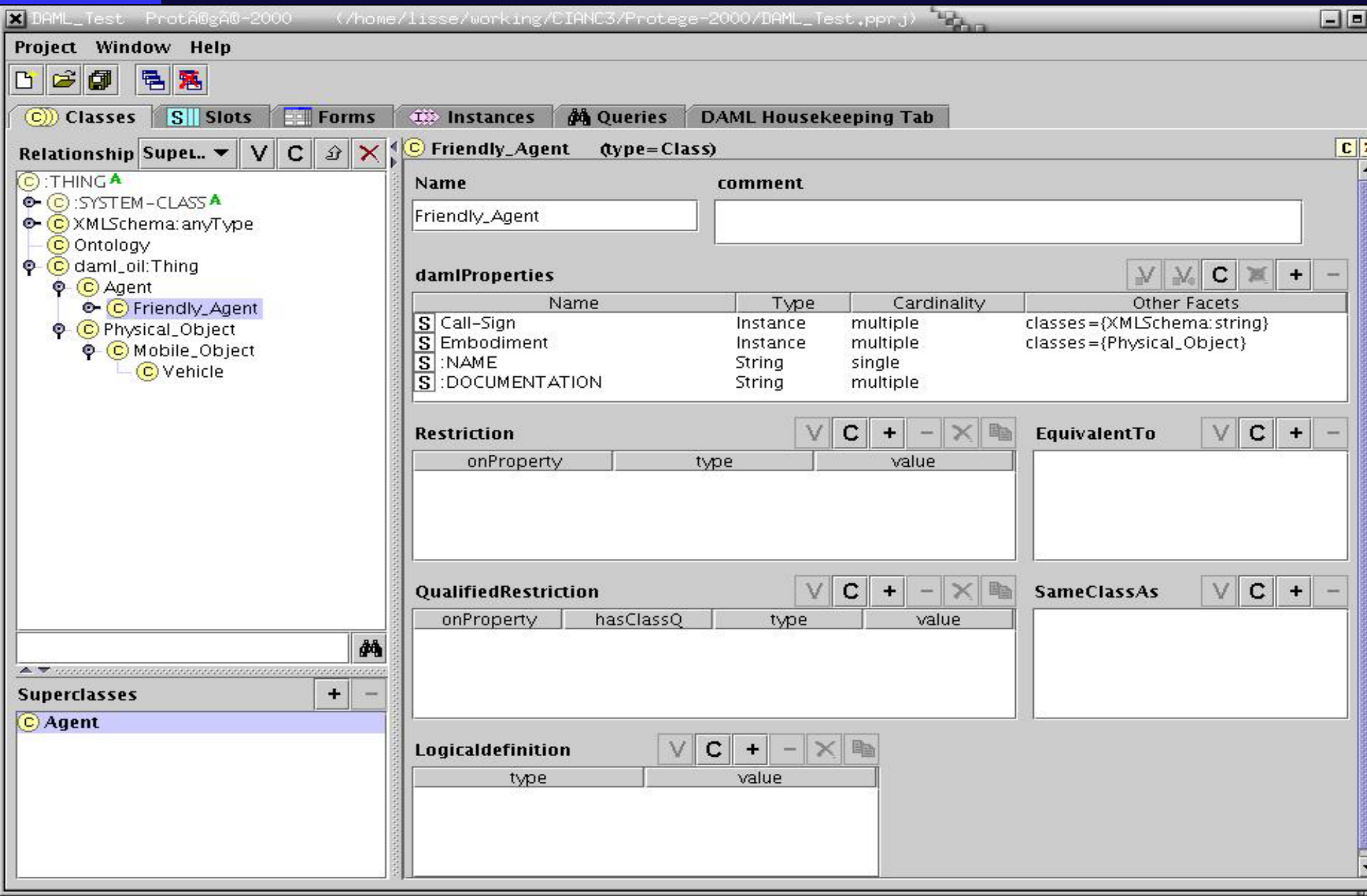
- Support for:
  - Classes
  - Properties
  - Superclasses

Protégé 2000

- Tree-based Ontology editor
- Can read & write DAML+OIL
- Under active development at Stanford
- Has customizable data-entry forms



Example Use – Input (Protégé)



Example Use – Output

```
sp {daml2soar*class*Friendly_Agent
(state <ts> ^problem-space.name top-ps)
(state <ts> ^ontology <ontology>)
-->
(<ontology> ^class <generated0>)
(<generated0> ^name Friendly_Agent)
(<generated0> ^superclass Agent)
(<generated0> ^superclass Thing)
(<generated0> ^property <generated1> )
(<generated1> ^name Embodiment)
(<generated1> ^type Physical_Object)
(<generated0> ^property <generated2> )
(<generated2> ^name Call-Sign)
(<generated2> ^type string)
}
```

DAML2Soar: Next steps

- Whole ontology active in WM
  - Scalability issues?
    - What is impact on matcher when there are many WMEs in memory?
- First step towards a possible LTM solution
  - How do Ont. sentences get to LTM?
  - How do Ont. productions get activated?
  - What are the correct retrieval cues?
  - Functional reasons for spreading activation?

Nuggets

- It works!
- 530 Soar Productions created in 5 minutes from the SUMO ontology, downloaded from the web

Coal and possible coal

- Prototype – needs further development
- No namespace support
- All we have are classes and properties
- Licensing TBD

Possible Future Directions

- Ontological reasoning knowledge for agents
- DAML Instances
- Property Restrictions
- Dynamic interface to agent
- Ontology transmission/sharing between agents
- Procedural knowledge representation formalisms
- Integration with other tools
- Additional relationship types beyond Property?

Acknowledgements

- Jon Beard and Bob Wray, Soar Technology
  - For coming up with the idea in the first place and enlightening me upon the Secret Path of Ontologi-do
- The CIANC3 project, funding by ARI
  - For paying for DAML2Soar initial research and development
- Jacob Crossman, Soar Technology
  - For convincing me that I didn't just want to write a back-end to Protégé