FIPA Ontology TC
Pleasanton – FIPA 23

Patricia Charlton

Dominic Greenwood

Ontology Aims in FIPA?

What are we/agents going to use ontologies for?

- To share common understanding of conceptual models
- To enable knowledge re-use
- To make assumptions about knowledge/domain models explicit

Abstract Semantic trees

To separate operational knowledge from domain knowledge

Referencing of multiple ontologies within content language expressions.

Ontology management: Ontology servers, Ontology maintenance,

Work Plan Milestones

2001/10 Issue Call For Information

2002/01 Document drafting relevant information to the process. Will reflect issues relating to existing FIPA specifications, FIPA semantic framework and external community projects.

2002/04 Development of specification.

2002/07 Design of a test suite to highlight the semantic interoperability stack relating to ontological representation. This should include the work of DAML+OIL and other technologies as appropriate.

2002/10 Pragmatic testing of "Multi-ontology Tool Bake-off" with FIPA implementations but also with the Semantic Web and other relevant architectures.

2003/02 Completion of testing and report of results.

2003/06 Feedback to the FIPA community and completion of the revised Ontology Service Specification.

Specific Call:

- *Requirements on ontology representation languages from an agent perspective.*
- *Requirements on agents to make use of ontologies*
- *The manipulation of ontological structures from a communication perspective, that is how to decompose the content into semantic trees or other relevant structures/representations.*
- *The identification and description of architectural mechanisms required to support ontologies.*
- *The definition of a technology stack including such representations as RDFS, DAML+OIL and DAML-S.*
- *Referencing of multiple ontologies within content language expressions.*
- *Discovery and management of public ontologies, revision management of published ontologies and Verification of ontological objects.*
- *Identification of appropriate use cases that highlight relevant issues.*
- *Solicitation of comments on FIPA00086, the legacy Ontology Service specification and any other related issues*
- *Reference models illustrating the relationship between ontologies and other FIPA models and entities, this may include meta-modelling utilising meta-languages and upper-ontology concepts*

Objectives

There are five main objectives of this work-plan:

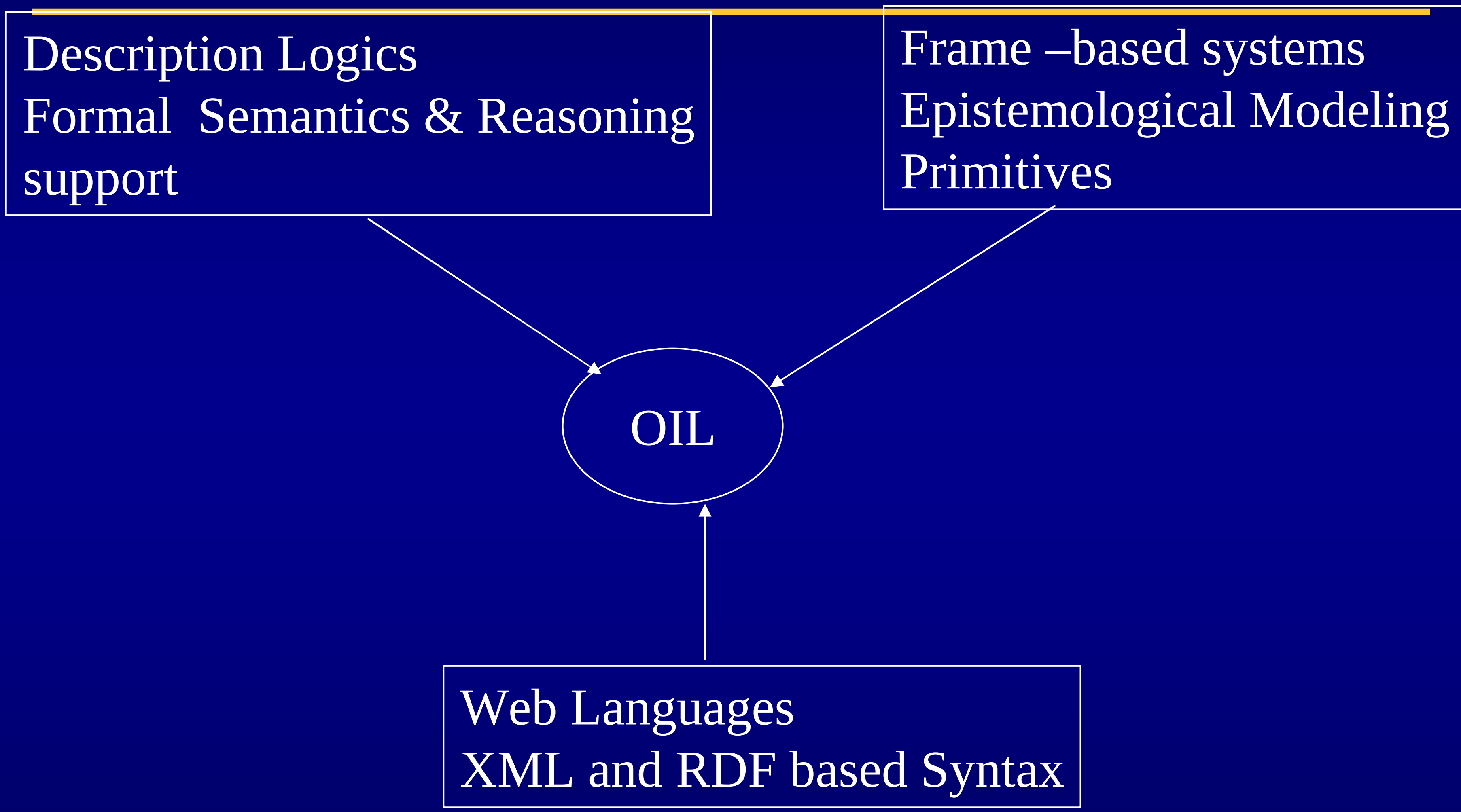
1. ***Review the former work on ontologies by FIPA [FIPA00086]. This material should be updated to include FIPA position statement and full FIPA2000 compliance.***
2. ***Develop an understanding of, and working relationships with, other projects and standards in this area (such as OKBC, DAML and OIL), in order to design an ontology stack enabling semantic interoperability.***
3. ***To issue a Call For Information to the community.***
4. ***To identify ontology specific architectural abstractions, [use case scenarios] illustrating any impact on other FIPA standards. Areas to be addressed will be:***
   - Relation of ontologies to the FIPA communication stack e.g. Multi-ontology referencing within unitary content expressions.
   - Manipulation of ontological structures from a communication perspective, that is how to decompose the content into semantic trees.
   - Sharing of ontologies.
   - Guidelines on how to test ontological interoperability.
   - To work closely with semantic web efforts, such as DAML+OIL.
5. ***Design of a test suite for a multi-ontology tool 'bake-off' between agent platforms and other knowledge sharing services.***

Frame based representation

Elements of a frame representations:

- Class frames (primitive and defined classes)
- Individual frames
- Slot frames (own slots and template slots)
- Facets (Each slot has a set of facets like :domain and :range)

OIL Ontology Interface Layer



Abstract Semantic Trees

Form and composition of semantic trees

- Capturing operational semantics of ontologies
- Clear level of abstraction
- KIF potentially useful representation for an abstract syntax

Web standards

XML a basis to add semantics

- DTD
- Easily readable and understood
- Can be embedded in web pages

RDF (xml based) to represent meta-data

- Data model consists of three object types: resource, predicate and object.
- Terminological part in RDF Schema
  - `Class, SubClassOf, type`
  - `Property, subPropertyOf,`
  - `domain, range`

Requirements for an ontology representation language

- Possibility to represent basic features: concepts, relations, instances;
- Possibility to develop meta ontologies;
- Xml based language to use the parser facility;
- Possibility to have other inference support than subsumption by representing axioms/rules.

XML and RDF based languages

XML based languages for domain description

- XOL
- SHOE

RDF based languages for domain description and inference processing

- OIL
- DAML +OIL

Ontologies

"formal specification that defines the representational vocabulary in some domain of discourse"

"An ontology defines a common vocabulary for agents who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them"

Already used for years in AI applications:

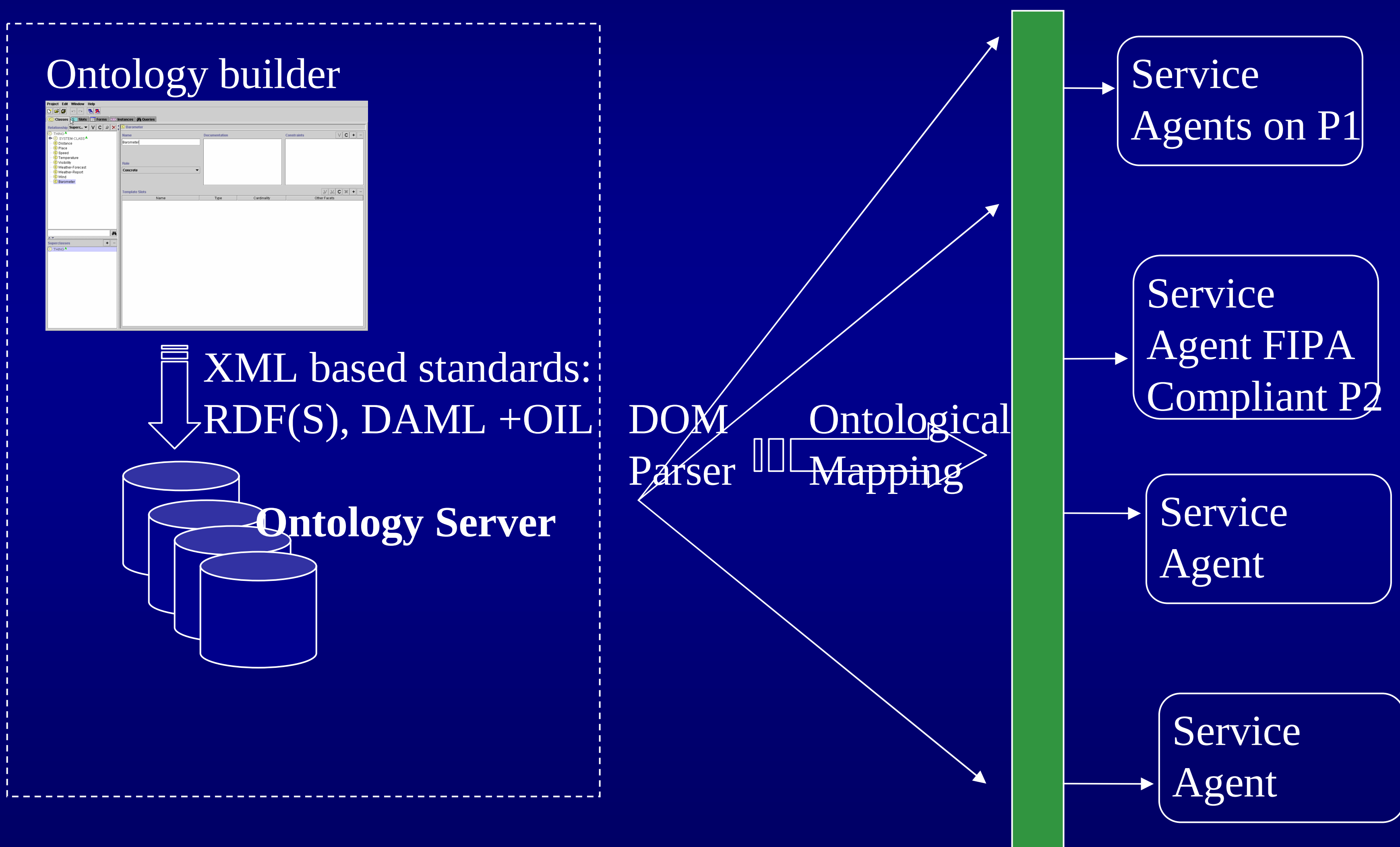
- Knowledge based systems
- Knowledge management…

Abstract Semantic Trees

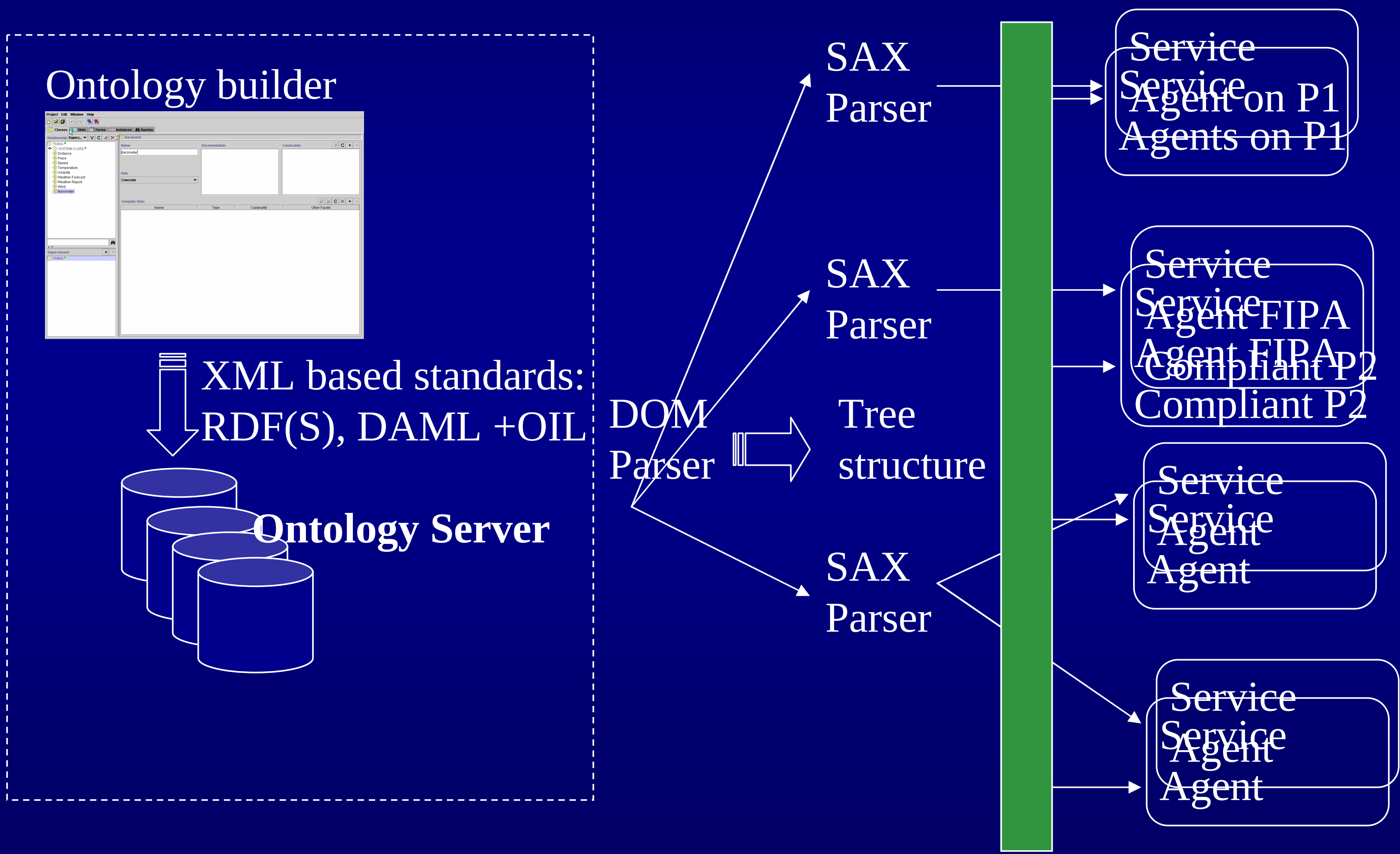
Form and composition of semantic trees

- Capturing operational semantics of ontologies
- Clear level of abstraction
- KIF potentially useful representation for an abstract syntax

Open Ontologies



Open ontologies



Ontology Formalisation

Using 5 elements:

- Concepts
- Relations
- Functions
- Axioms
- Instances and facts

Traditional languages: Ontolingua (KIF), OKBC, OCML, Flogic, LOOM, CG,

Web based ontology languages: XOL, SHOE, OIL, DAML+OIL based on XML and RDF(S), DAML-S

Tools for ontology building

- Ontolingua (KIF representation)
- GKB editor (OKBC-compliant knowledge bases)
- Protégé Ontology Builder (OKBC compliant, RDF, OIL plugins)
- OntoEdit (OIL)
- WebOnto (OCML)
- WebKB-GE (Conceptual graphs)
- Ontology builder and server from VerticalNet
- Jena: Java RDF parser

Ontolingua

Logical semantics on KIF

Ontolingua extends KIF syntactically for defining object-oriented and frame terms

Set of KIF expressions defined as a frame ontology

High expressive power

OIL

Inheriting values can not be over written

Only a fixed number of algebraic properties for Rules/Axioms –no composite definitions of relationships

Namespace limitations;

Descriptive logic limitations; using instances in class definitions is not supported

Limited second order expressivities; only classes are provided not meta-classes; cannot treat statements of the language as objects in their own right.

DAML + OIL

Name space and import; need to check uniqueness and accessibility

Datatypes for concrete domains; not individual objects

Better Class model

RDFS limitations

OKBC and XOL

Restrictive representation of class

No mechanisms for defining disjoint

No slot hierarchy or transitive relations

Requirement

Function

- Agents and service descriptions referencing functions in the context of multiple ontologies

Policy

- Referencing of >1 ontology in permissions and obligations

Implication…

In conversation:

- May need to state content expressions and conversational semantics in the context of different ontologies

Multiple References

Referencing of several ontologies within content expressions:

FIPA 2000 → ***one ontology reference per message content***

Proposal → ***to allow segments of content to be associated with different ontologies***

Finest granularity:

Individual symbols can be tagged with ontology references

Scoped granularity:

Content is scoped and tagged in terms of relevant ontology

Namespaces:

Tags belong to namespaces, such as with XML associations

Example 1

ACL level namespace

```
(accept-proposal
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :ontology
    (set (ontology-identifier :name multimedia,
          ns: “http://www.xyz.org/multimedia#"),
      (ontology-identifier :name contract,
          ns: “http://www.xyz.org/contract#"))
  :content
    ((action (agent-identifier :name j)
       (multimedia:stream-content movie bladerunner))
     (B (agent-identifier :name j)
       (contract:payment-made customerX)))
  :language FIPA-SL
```

Example 2

Content level namespace

```
(accept-proposal
  :sender (agent-identifier :name i)
  :receiver (set (agent-identifier :name j))
  :ontology

  :content
    ((ontology-identifier :name multimedia,
           ns: “http://www.xyz.org/multimedia#"),
      (ontology-identifier :name contract,
           ns: “http://www.xyz.org/contract#")
           (action (agent-identifier :name j)
               (stream-content movie bladerunner))
       (B (agent-identifier :name j)
           (payment-made customerX)))
  :language FIPA-SL
```

Architecture Issues

Identify and describe the abstract architectural elements required to support ontologies:

Discovery and management of public ontologies.

Revision management of published ontologies:

Unlikely that any Ontology will remain static over time

=> some guarantees of consistency

=> reasonable and acceptable sources of change

=> registration required for updates

Ontology objectives

Use ontological framework to model services. The framework needs to allow the models to be extended and maintained in some form of autonomous manner, without having to take down the services.

Offers infrastructure which allows added value to enabling a concept of service aggregation and new service identification.

Coordination of service ontologies autonomously (where a service ontology can be a user-model, set of stereotypes a particular domain etc.). The ontology supports mapping an ontology model of one type to another using the concept of types, relationships and schemas to set contexts. The matching may be fuzzy that is an object or meta-object only belongs to a context because of a particular filter or boundary but in the normal classification of meta-level to object level may fail. Hence the meta-meta-level of types, relationships and schemas are important.

To assist in the generation of more re-usable components

To enable more autonomous use of components, through the semantics a component can be integrated automatically. This is supported via a set of default configuration rules. Again the rules can be modelled as ontology objects which can be refined through meta-meta-schema to enable adaptable configurations.

Offers infrastructure which allows added value to enabling a concept of service aggregation and new service identification.