Coalition Search and Rescue - Task Support

Intelligent Task Achieving Agents on the Semantic Web

Final Report

Austin Tate & Jeff Dalton
AIAI, Informatics, University of Edinburgh

Jeff Bradshaw & Andrzej Uszok
IHMC, Pensacola, FL

Project Summery

- To provide capabilities linking:
  - models of organizational structures, policies, and doctrines
  - with intelligent task support software
- The project integrates:
  - AIAI's I-X planning and collaboration technology
  - IHMC's KAoS policy and domain services
  - Semantic Web Services of various kinds
- Search and rescue operations - rapid dynamic composition of available policy-constrained services - good use case for Semantic Web
- Other participants in the application include: BBN Technologies, SPAWAR, AFRL, and CMU

Project Goals

- **Development of base technologies:**
  - **I-X/I-Plan**
  - **KAoS Policy and Domain Services,**
- **Deployment of the technology in a realistic CoAX agents demonstrator scenario,**
- **Integration of these two technologies with a perspective of a uniform tool release in the future.**

Project Yearly Outline

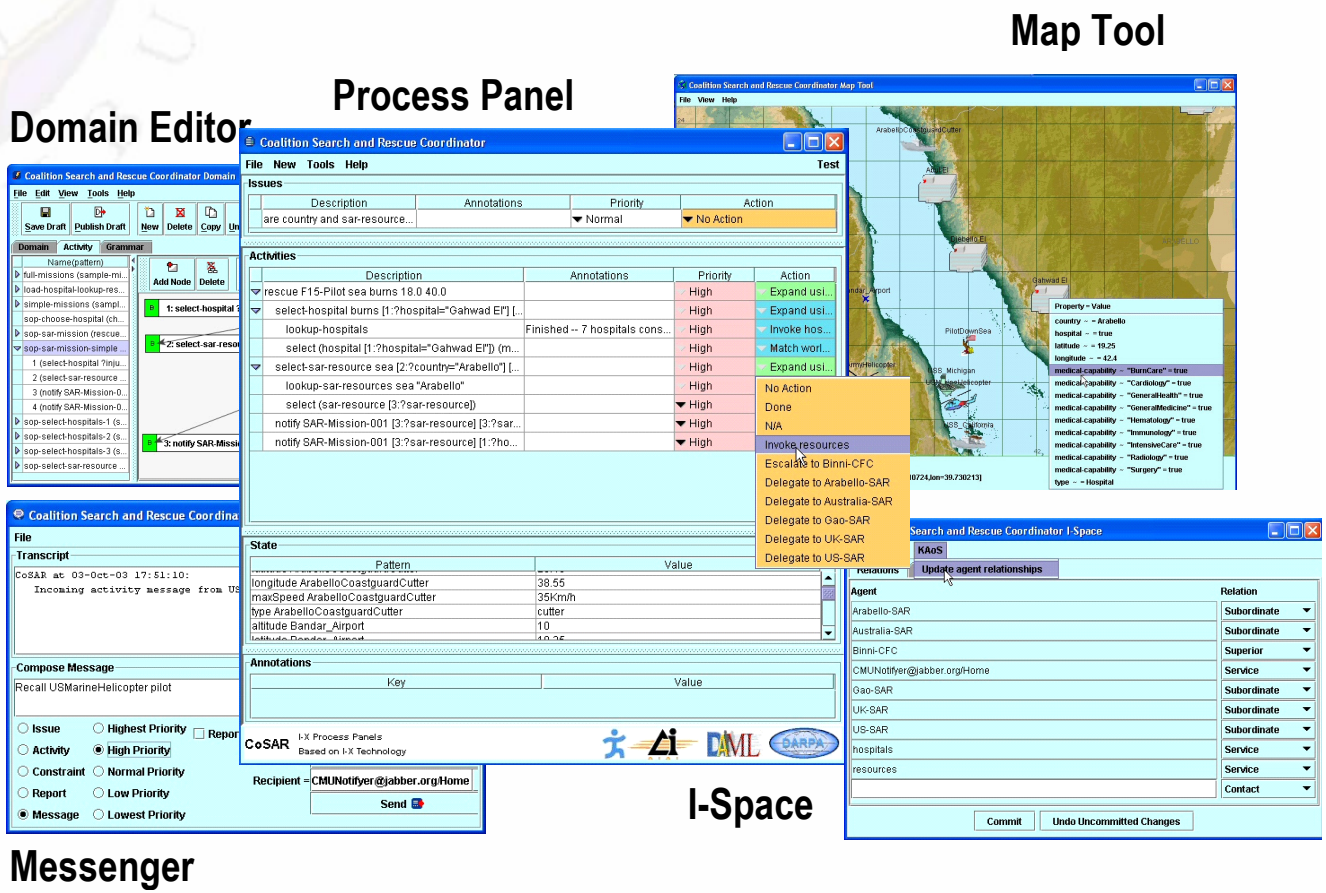
- ***Year 1:*** **Distributed multi-agent systems were developed and integrated with the semantic web in a realistic coalition search and rescue scenario:**
  - **AAAI-2004 Intelligent Systems Demonstrator for CoSAR-TS**
- ***Year 2:*** **An initial web services composition and policy analysis tool for semantic web services (I-K-C) was implemented:**
  - **IEEE Intelligent Systems journal article and an ISWC 2004 conference paper**

Details of developed technology

I-X Technology

- Reasoning about and exchanging with other agents and services any combination of Issues, Activities, Constraints and Annotations
  - represented in the <I-N-C-A> ontology.
- Collaborative task support and exchange of structured messages related to plans, activity and the results of such activity.
- Information can be exchanged with other tools via OWL, RDF or other languages.
- The system includes an AI planner

I-X Process Panel and Tools for a Coalition Search and Rescue Task



I-X Process Panels

- Intelligent ‘to-do’ list for its user
- In conjunction with other users’ panels, it can become a workflow,
  - reporting and messaging ‘catch all’
  - allowing the coordination of activity
- Presentation of the current items of each of the four sets of entities comprising the <I-N-C-A> model
- Can take requests to:
  - Handle an issue
  - Perform an activity
  - Add a constraint
  - Note an annotation

Policies and Semantic Web Services

- Semantic Web Services to be used by people but also by software agents
- Policy ensure that human-imposed constraints on agents interactions are respected
- Policy-based controls can also be used to govern interaction with traditional (non-agent) clients
- Proposals for SOAP-based message security and XML-based languages for access control (e.g., XACML2) have begun to appear recently
- However only declarative ontology-based policy semantics can fulfill the SWS requirements

Use of Ontology in KAoS

- Descriptions of actors, actions, situations at different levels of abstraction, policies
- Possibility to dynamically calculate relations among policies and current situation, as well between policies themselves based on ontological relations of used concepts
  - Dynamic extension of the policy framework by specifying platform ontology and linking it with generic KAoS framework ontology
  - Extension of the framework itself by adding new ontologically-described components
  - See: http://ontology.ihmc.us/

KAoS Policies

- **Main types of supported policies:**
  - Authorization **– Negative and Positive**
  - Obligation **– Negative and Positive**
    - » **Associated with a Trigger Specifying Conditions Activating thisObligation**
- **Policy controls** actions
  - **Includes a description of the *action template/class***
  - **Constitutes a** test **for the applicability of the policy**
- **Policy posses a** priority**, which enables it to take precedence above contradicting ones**
  - **Will be replaced by a more general** precedence **mechanism**
    - » **Encoded in OWL**

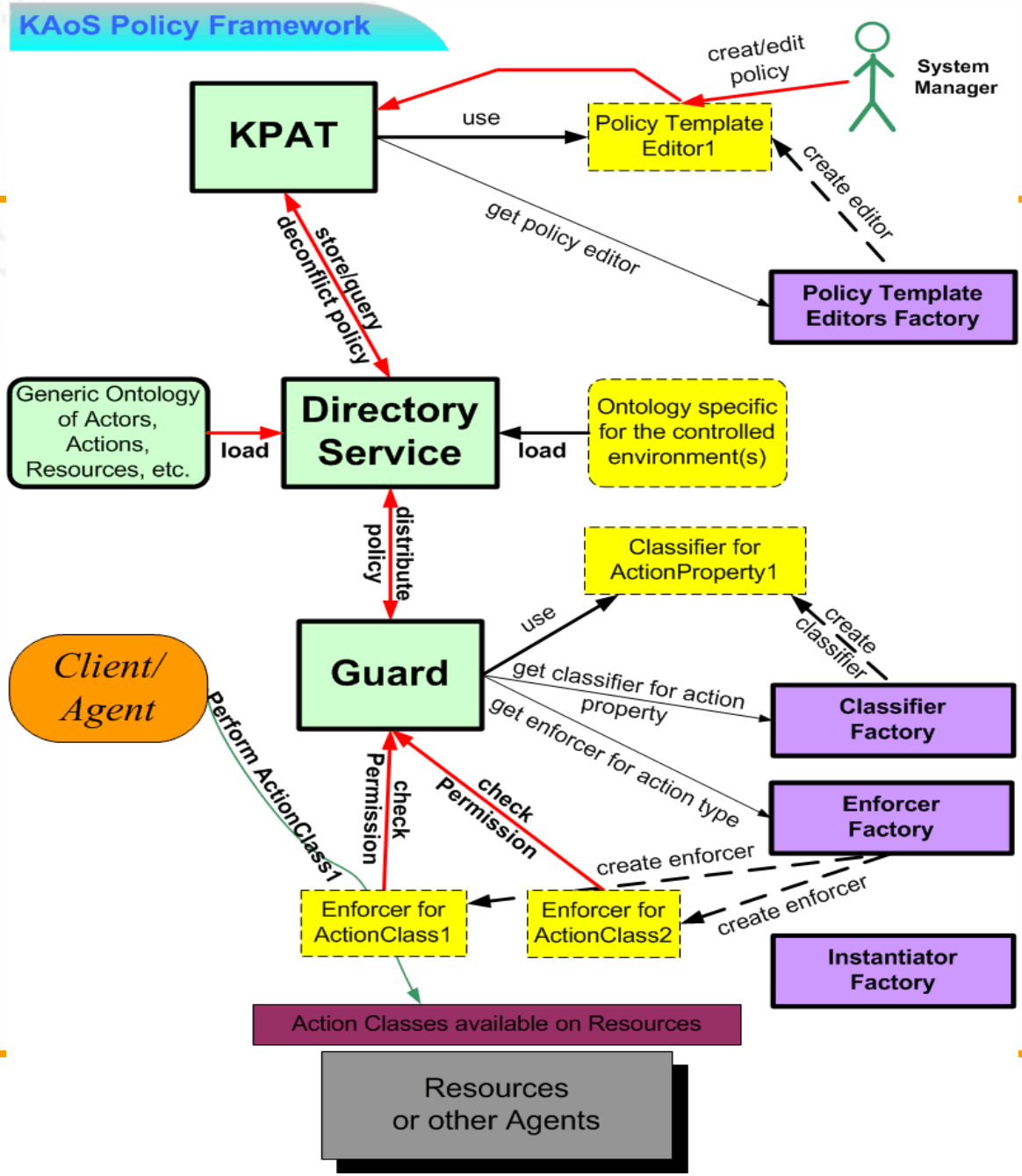
Policy Syntax Example

```
<?xml version="1.0" ?>
<!DOCTYPE P1 [
      <!ENTITY policy  "http://ontology.ihmc.us/Policy.owl#" >
      <!ENTITY action  "http://ontology.ihmc.us/Action.owl#" >
      <!ENTITY domains  "http://ontology.ihmc.us/ExamplePolicy/Domains.owl#" >
 ]>
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.owl.org/2001/03/owl+oil#"
      xmlns:policy="http://ontology.ihmc.us/Policy.owl#"
>
<owl:Ontology rdf:about="">
      <owl:versionInfo>$ http://ontology.ihmc.us/ExamplePolicy/ACP1.owl $</owl:versionInfo>
      <owl:imports rdf:resource="http://www.owl.org/2001/03/owl+oil" />
      <owl:imports rdf:resource="http://ontology.ihmc.us/Policy.owl" />
      <owl:imports rdf:resource="http://ontology.ihmc.us/Action.owl" />
      <owl:imports rdf:resource=" http://ontology.ihmc.us/ExamplePolicy/Domains.owl" />
</owl:Ontology>

<owl:Class rdf:ID="OutsiteArabelloCommunicationAction">
      <owl:intersectionOf rdf:parseType="owl:collection">
          <owl:Class rdf:about="&action;NonEncryptedCommunicationAction" />
          <owl:Restriction>
               <owl:onProperty rdf:resource="&action;#performedBy" />
               <owl:toClass rdf:resource="&domains;MembersOfDomainArabello-HQ" />
          </owl:Restriction>
          <owl:Restriction>
              <owl:onProperty rdf:resource="&action;#hasDestination" />
              <owl:toClass rdf:resource="&domains;notMembersOfDomainArabello-HQ" />
          </owl:Restriction>
      </owl:intersectionOf>
</owl:Class>

<policy:NegAuthorizationPolicy rdf:ID="ArabelloCommunicationPolicy1">
      <policy:controls rdf:resource="# OutsiteArabelloCommunicationAction " />
      <policy:hasSiteOfEnforcement rdf:resource="&policy;ActorSite" />
      <policy:hasPriority>10</policy:hasPriority>
      <policy:hasUpdateTimeStamp>446744445544</policy:hasUpdateTimeStamp>
</policy:NegAuthorizationPolicy>
```

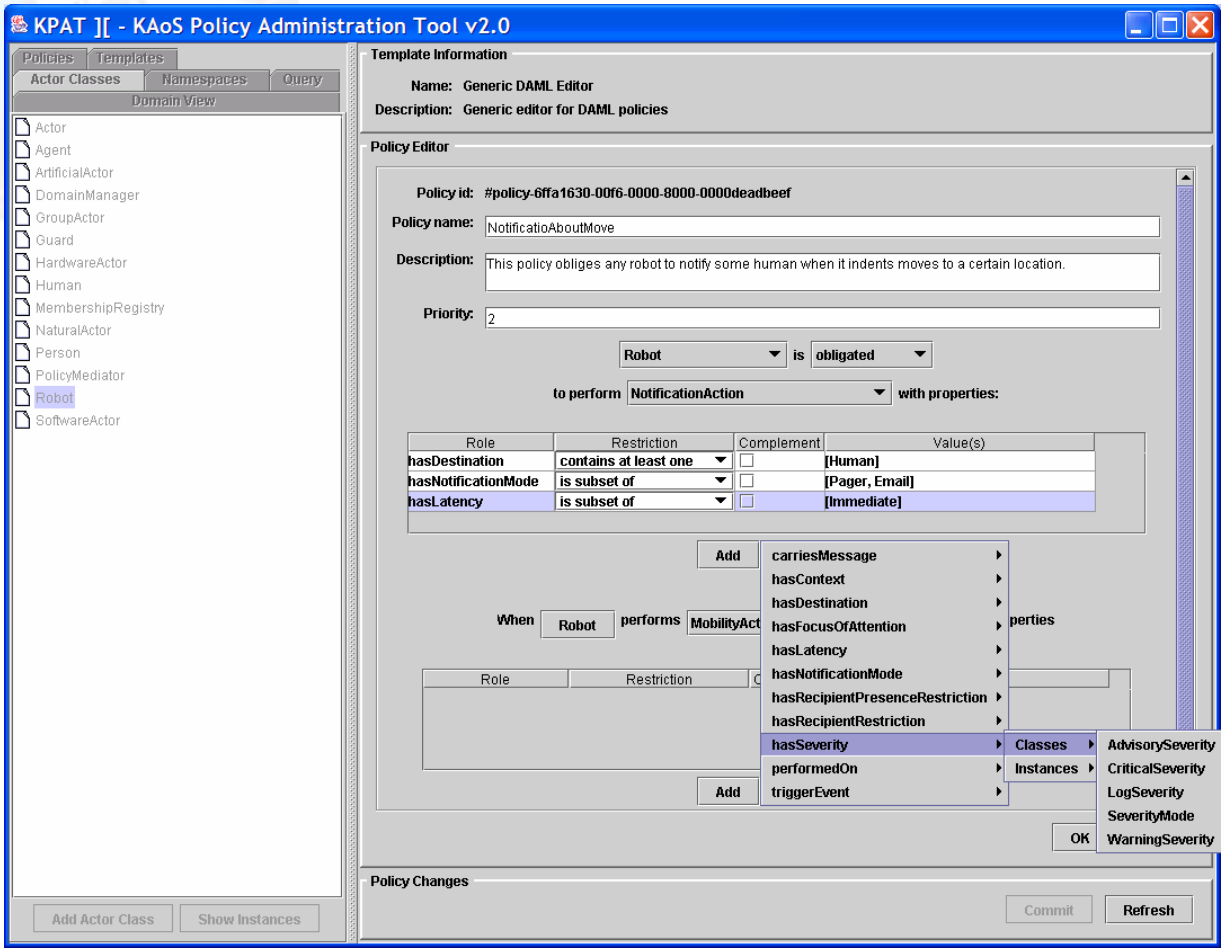
Framework Overview



Description Logic Reasoning

- **Subsumption-based reasoning used for determination of disjointness:**
  - **Finding policy conflicts by determining if two classes of controlled actions classes are disjoint**
  - **Harmonization of policies**
- **Instance classification:**
  - **Policy exploration, disclosure, and distribution**
- **Usage of Stanford inferencing engine –** JTP

KPAT Hides Complexity



Dynamically obtains list of selections from the ontology repository based on the current context.

Uses **Jena** - Java OWL manipulation library to build policies.

Beyond Description Logic for Policy Representation

- **Originally KAoS used only OWL-DL (initially DAML)**
- **Limited in situations when needed to define policies where one element of an action's context depended on the value of another part of the context:**
  - **Example – Loop Communication Action**
  - **Relation between Trigger Action and Obliged Action**
- **These requirements can be fulfill by** role-value-map semantics
  - **maps allow policy to express equality or containment of values that has been reached through two chains of instance properties**
- **KAoS was equipped with mechanisms adding role-value-map semantics to defined policy actions when necessary**

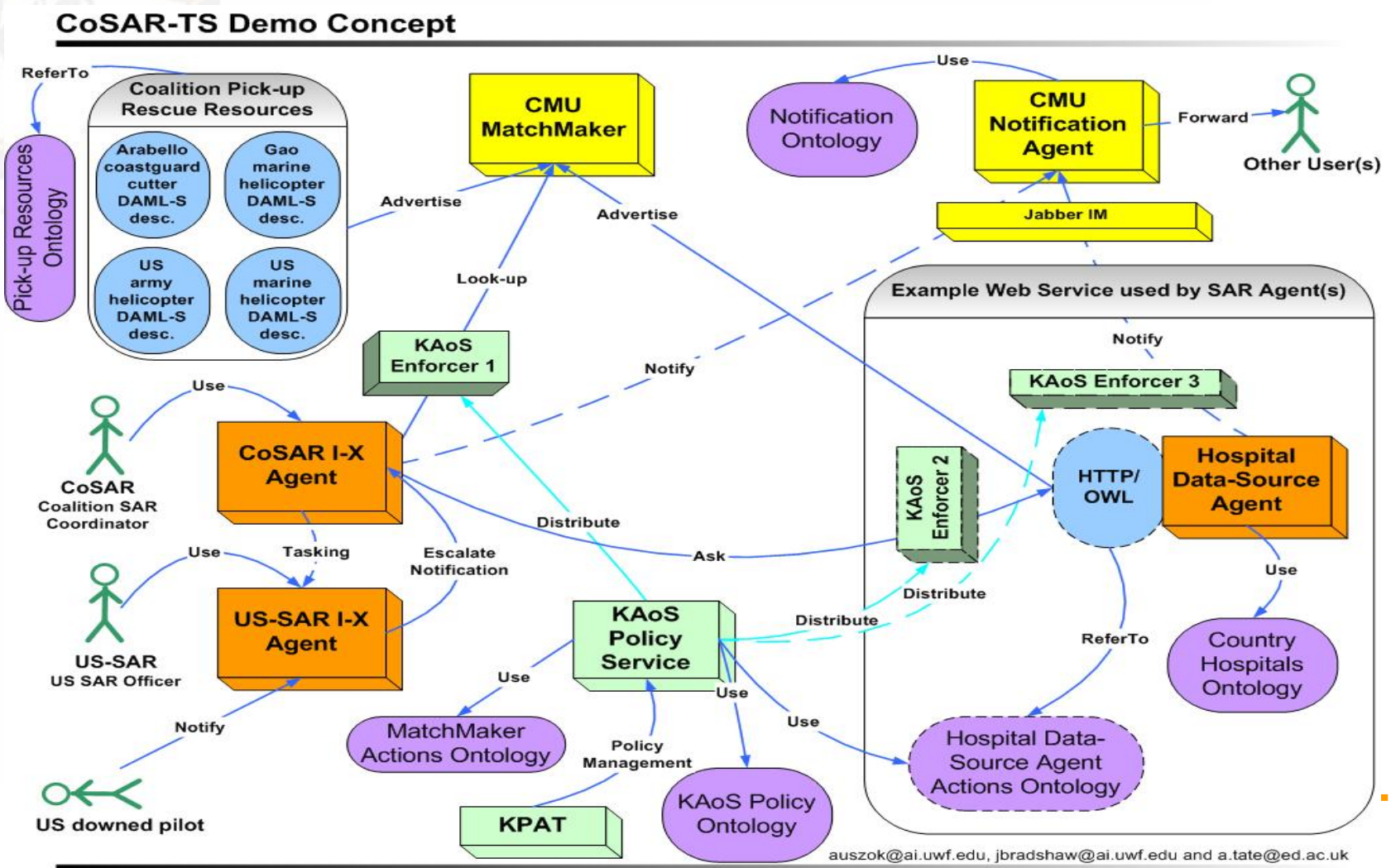
Generic Semantic Web Service Policy Enforcer

- Intercept SOAP messages
- Understanding arbitrary Semantic Web Service invocations:
  - Follows annotations from WSDL interface to OWL-S interface
- Apply appropriate authorization policies to request – filtering these forbidden
- It is equipped with a mechanism to perform obligation policies,
  - which is in a form of other Web Service invocations

CoSAR-TS Scenario

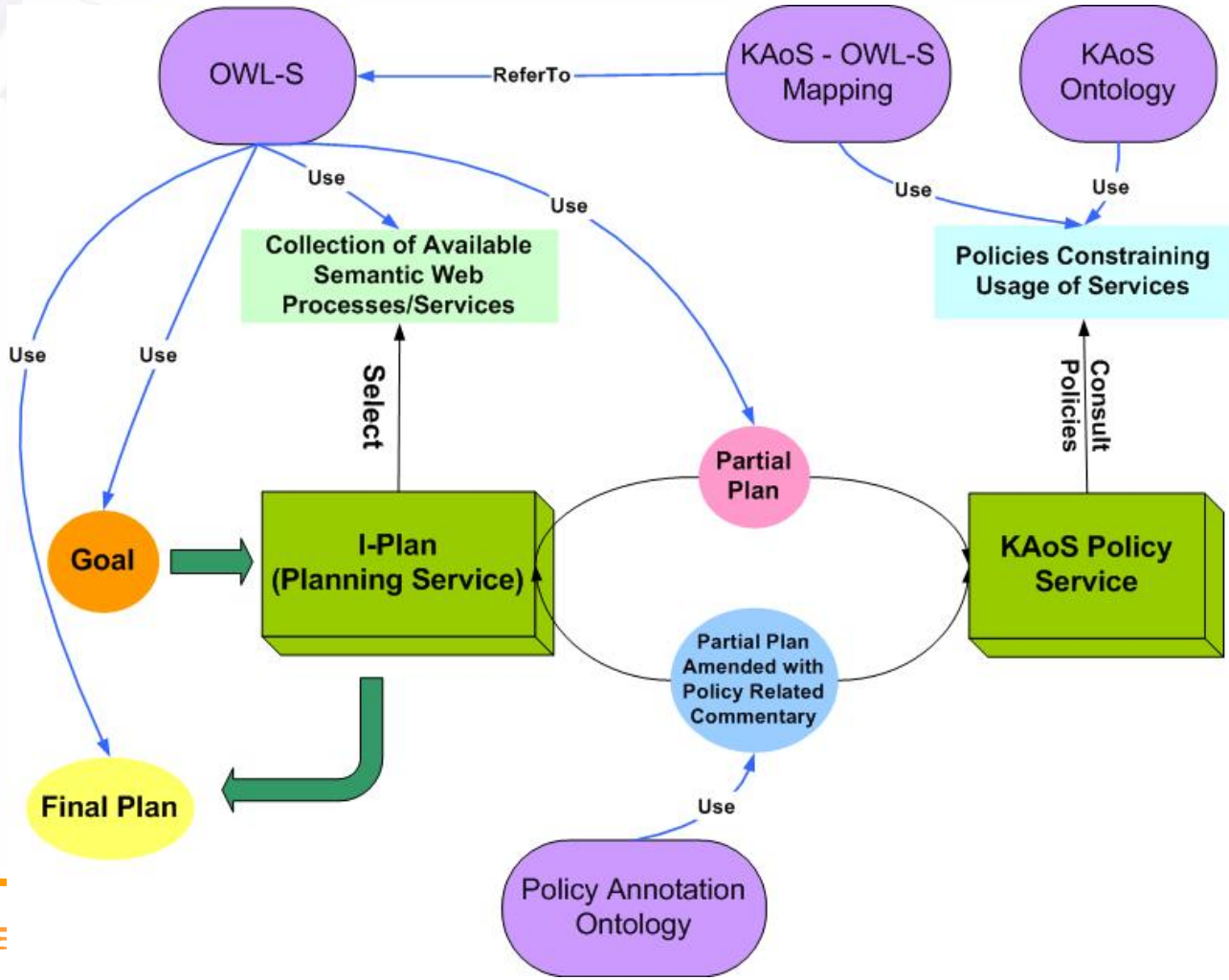
- Based on the Arabello military scenario from the CoAX (Coalition Agents eXperiment) project
- The story begins with an event that reports a downed airman in the Red Sea
- Rescue resources (transportation, medical, notification) represented as dynamic Semantic Web Services
  - Description based on ontology developed for the DARPA SONAT experiment
- The selection of a SAR resource is made using the CMU Semantic Matchmaker to find a suitable service
- These lookups comply with KAoS policies

CoSAR-TS demo details



Constraining/Advising Service Workflow Composition

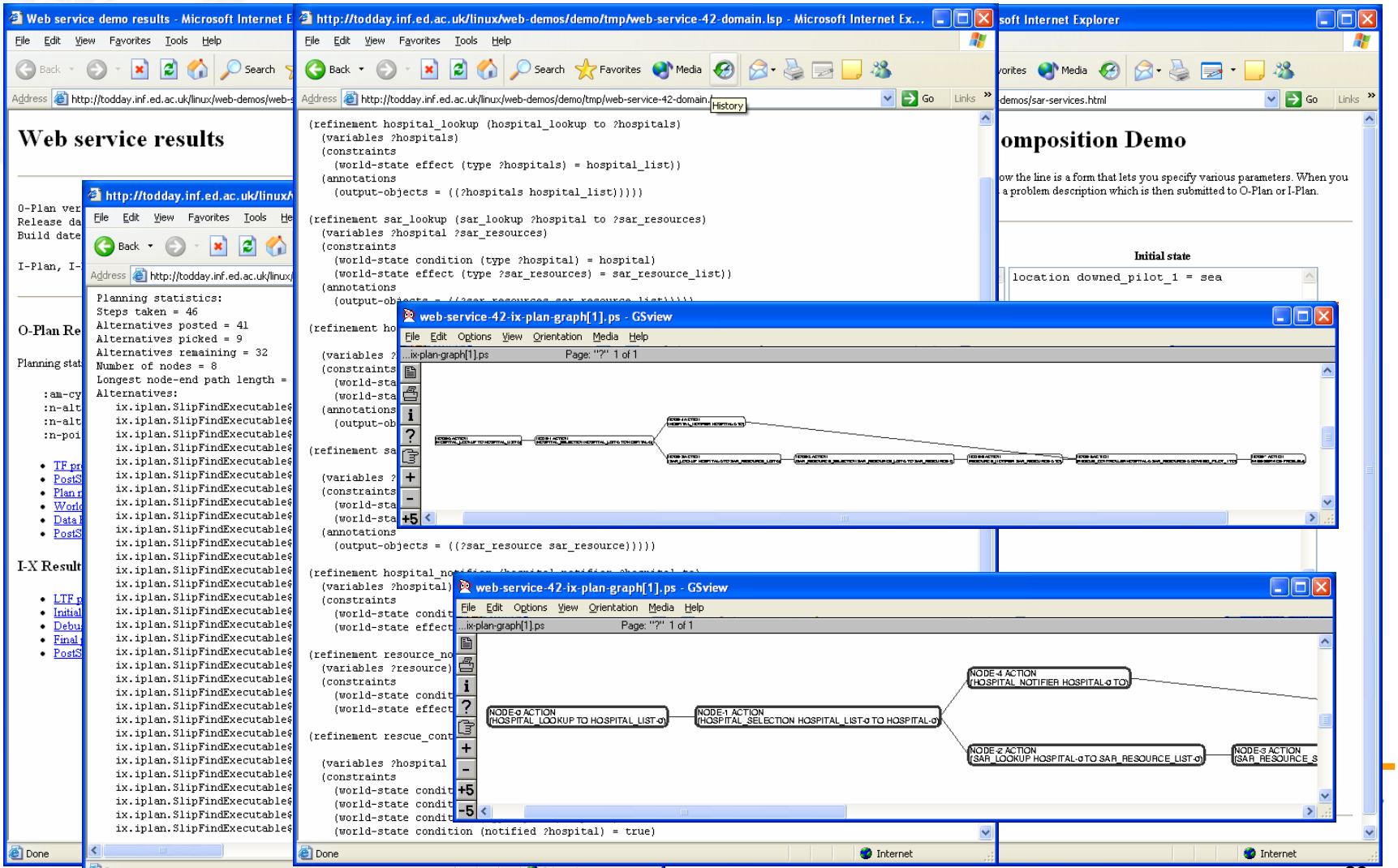
I-Plan – KAoS integration



I-X new capabilities

- **Extend the I-Plan planning elements to allow for the creation of composed workflows ahead of execution**
- **Import of services described in OWL-S to be used within the planner**
  - ***Dealing with Inputs & Outputs***
  - ***Recovering Data flow from Plan Goal Structure***
- ***I-Plan as a web service***

I-Plan Web Service Workflow Composition



Workflow Compositions

- **Incremental plan built by I-Plan defined using combination of processes expressed using OWL-S**
- **KAoS analyzes the proposed plan and annotates it with policy decisions:**
  - **Currently considers individual workflow actions**
  - **In the near future, will take into account action context within the workflow; e.g. actions preceding the given action**

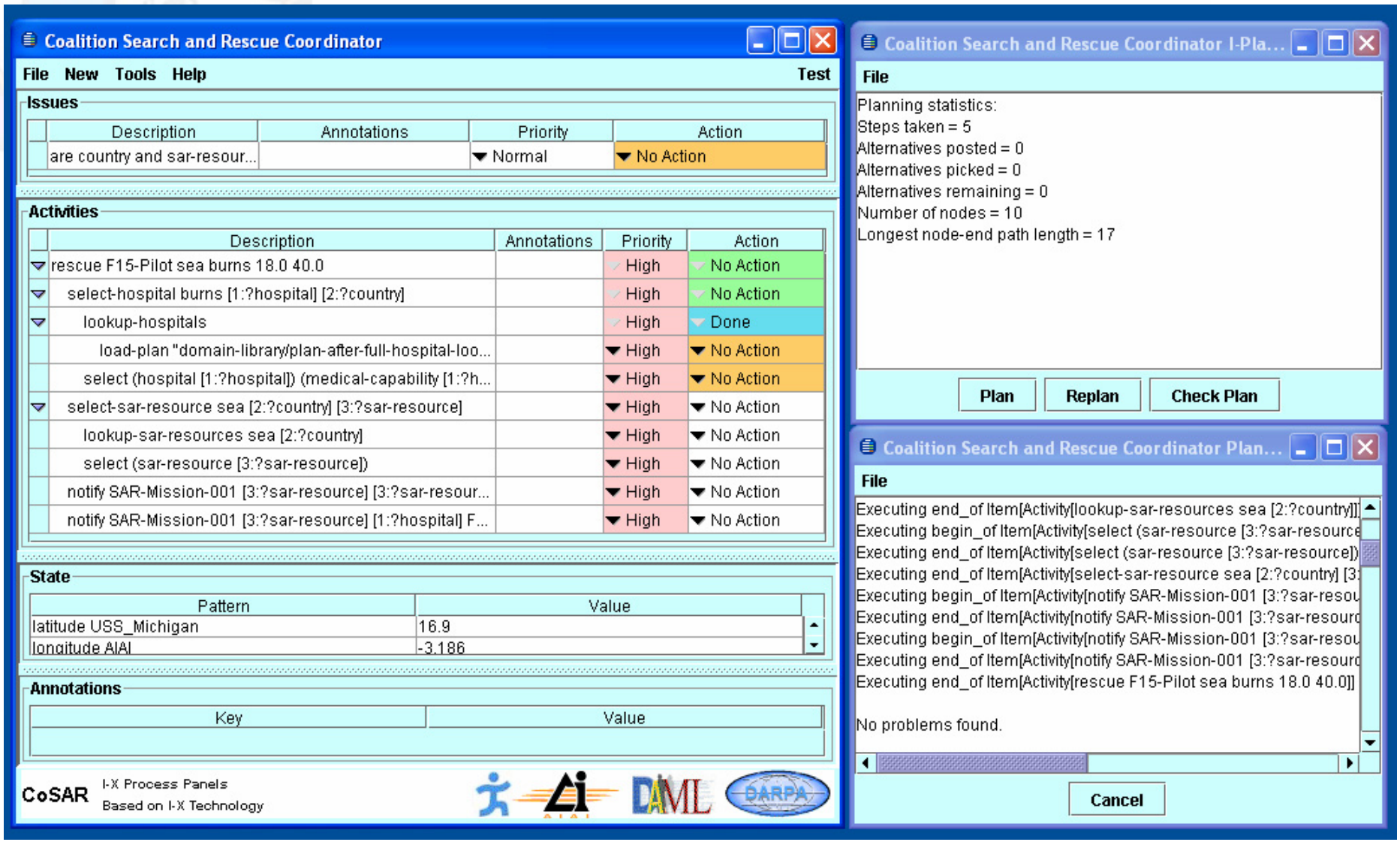
Mapping the OWL-S Process to KAoS Concept Action

- OWL-S concept of Process maps semantically to the KAoS concept of Action
- OWL-S represents Processes as instances, KAoS represents Actions as classes
- Need to create an OWL class based on the OWL-S process definition instance
- OWL-S API is used to:
  - load OWL-S process workflows,
  - find all processes within a workflow
  - get detailed definitions about each of them,
- Using Jena, KAoS builds the OWL class that corresponds to a subclass of the KAoS Action class beign eithr authorize or obliged by policies

KAoS Workflow Analysis

- Action class extracted from the workflow is analyzed for policy compliance:
  - Action authorization and possible additional obligations
- Using subsumption reasoning KAoS finds relations between the current action class and action classes associated with policies:
  - deterministic conclusions – when checked action fully subsumes policy action
  - nondeterministic conclusions – when checked action is neither fully subsumed nor fully disjoint with policy action
  - KAoS builds a representation of the new action class by computing the difference between the current action class and the relevant policy action class

I-Plan Java Tool



On-line resources

- **CoSAR-TS AAAI-2004 Intelligent Systems Demonstrator**
  http://www.aiai.ed.ac.uk/project/cosar-ts/isd/
- **KAoS KPAT Java Web Start demonstration**
  http://norma.coginst.uwf.edu:8080/coalition/KPAT-TCP.jnlp
  http://ontology.ihmc.us
- **I-K-C tool demonstrations**
  http://www.aiai.ed.ac.uk/project/i-k-c
  http://projects.semwebcentral.org/projects/i-k-c
- **Web service composition examples**
  http://todday.inf.ed.ac.uk/linux/web-demos/web-service-demos/web-service-examples.html
- **Demonstration on-line web services composer running via a SOAP interface**
  http://todday.inf.ed.ac.uk/linux/web-plan/web-plan.html

Conclusions

- New sophisticated functionalities in AIAI's intelligent planning technology and IHMC's KAoS services
  - fully OWL compliant
- The cooperation between AIAI and IHMC was significantly strengthened
  - collaborate on future projects
  - release tool integrating both technologies
- The project deepened understanding of the Semantic Web technology
  - realistic military scenarios
- Tested for technologies developed by other DAML program participants
- Communication of the value of lessons learned on the project to the OWL and OWL-S committees and forums