Applying Semantic Rules for Dynamic Service-Oriented Architecture

Suzette Stoutenburg, Leo Obrst, Deborah Nichols, Ken Samuel, and Paul Franklin
Research Team Members

Principal Investigators

Leo Obrst, Ph.D.
E544/McLean

Suzette Stoutenburg
E540/Colo Spgs

Ontology and Logic Environment

Deborah Nichols
E544/McLean

Ken Samuel, Ph.D.
E544/McLean

Rules and Application Development

Paul Franklin
E540/Colo Spgs

Karen Fox
E540/Colo Spgs
Objectives

- Identify sponsor-specific requirements for semantic rules and ensure that they are captured in the emerging World Wide Web Consortium (W3C) recommendation
  - Rules separated from executable code
  - Rules integrated with ontologies to leverage logic
- Demonstrate a **dynamic** Service-Oriented Architecture using rules that allow adaptive change in standard operating procedures
  - Rules supporting situation awareness
  - Rules adjusting in response to real-time events
- Optimization of the Logic Programming environment that integrates and reasons over rules and ontologies
Mission Use Case Key Concepts

- Friendly GMTI
- Hostile GMTI
- Unknown GMTI
- Intel Summary and ROI
- Region of interest (ROI) provides "safety zone" around convoy
- Convoy route
- Path of UAV
- A track is said to be the "focal object" of an ROI
Scenario Ontology Overview

TheaterObject
- hasFocalObject
- DynamicRegionOfInterest
- subclassOf RegionOfInterest
- subclassOf Convoy
- hasCurrentRoute
- describedBy ConditionsAndAlerts

MilitaryUnit
- hasFocalObject
- DynamicRegionOfInterest
- subclassOf RegionOfInterest

ObservationArtifact
- subclassOf GMTIObservation
- subclassOf IntelSummary
- subclassOf VMTIObservation

Convoy
- hasCurrentRoute

ConvoyRoute

PrimaryOntology

Majorclasswithinontology
Applications of Rules

- Support intelligent processing of real-time events
  - Synthesize information from multiple sources
  - Apply data integrity constraints
  - (Future work) Mediate between conflicting sources, and
  - Derive new knowledge based on source type and reliability
- Construct conceptualization of the battlespace for enhanced situational awareness
  - Transfer attribute values from ObservationArtifacts to TheaterObjects
  - Establish ‘safety zones’ and ‘threat areas’ (ROIs) for objects
  - Detect and report threats to friendly forces
- Assist decision makers in applying Rules of Engagement to developing situations
Year 2 Prototype Modifications

- **Modifications to the ontologies**
  - Added properties to support new rules and represent additional information from messages
  - Added subclasses of weapons and vehicles
  - Classes and relations for other domains (missile, maritime, airspace)

- **Rules made more robust**
  - Detect when new TheaterObjects should be created
  - Some geospatial rules to detect enemy approaching, overtaking, etc.
  - Added “extensions” to SWRL, by creating predicates to:
    - Create new instances
    - Apply message data to the set of instances
    - Support non-monotonic reasoning

- **Dynamic rule changes supported in a SOA with communications across Enterprise Service Bus (ESB)**
**Rules for Dynamic Services**

**Goal:** Dynamic service behavior based on real-time events

Abstract from service behavior and express in the form of rules

Create a generic service, e.g., “What are the implications of information updates for convoy security?”

Applicable Rule Set produces appropriate response:

In normal daytime conditions, Rule Set 1 is in effect

Real-time weather event causes Rule Set 2 to take effect

- If good visibility, send out UAV to take closer look at unknown mover
- If poor visibility, avoid encounters with unknown movers
Sample Rules (Day)

IF an unknown mover is within the convoy’s ‘safety zone’
THEN alert the commander that there is a nearby unknown mover

IF an unknown mover is within the convoy’s ‘safety zone’
THEN recommend that the convoy deploy UAV reconnaissance
Sample Rules (Night)

IF an unknown mover is within the convoy’s ‘safety zone’
THEN alert the commander that there is a nearby unknown mover

IF [above] & the convoy has a viable alternate route
THEN recommend that the convoy change to the route
Logic Programming Environment

- Developed SWORIER (Semantic Web Ontologies & Rules for Interoperability with Efficient Reasoning)
  - Reasons over ontologies and rules and answers queries with Logic Programming
- Designed general Prolog predicates to implement the semantics of OWL (e.g., hasPropertyWith, disjointClasses, logicNot)
- Addressed major challenges
  - Prolog and OWL define negation differently
  - Prolog assumes closed world, while OWL assumes open world
  - Disjoint and complementary classes
    - Previous work claimed this was not possible [Volz et al., 2003]
    - Simpler approach for equivalence
- Initial performance of SWORIER slow; improvements gained with extensionalization, caching, and code minimization
Logic Programming Challenges

- Logical Negation
- Open/Closed World
- Complementary/Disjoint Classes*
- Disjunctive Conclusions*
- Enumerated Classes*
- Equivalent Individuals*

- Error Messages
- Existential Quantification*
- Cardinality Restrictions*
- Cyclic Hierarchies
- Anonymous Classes
- Duplicate Facts

* Solutions to be implemented
Rules not expressible in SWRL were represented in Prolog directly; now in the process of specifying extensions on SWRL
Design Decisions

- Enterprise Service Bus (ESB) technology selected for underlying communication infrastructure
  - Supports integration with other applications and prototypes
- Google Earth used as client
  - Provides realistic maps and zoom-in capabilities
  - Supports integration with other applications
- Ground facts for instances stored in the Knowledge Base (instead of database) for simplicity
- Separate simulation for Convoy used, making Convoy behavior independent of message injection
- Issue: How should rules be swapped?
  - Since instances are stored in the Knowledge Base, how are they persisted after the rule swap?
  - Decision: Instantiate multiple Knowledge Bases, apply updated instances to each
Architecture Options for Rules

Option 1: Instances in Knowledge Base
Instances must be persisted across multiple KBs

Option 2: Instances in Database layer
Tools must support links from KB to database

Option 3: Preferred
Meta-rules control application to KB instances in database
Prototype Architecture

1. Logic Servers instantiated with Rule Sets
2. Message injection starts
3. Adapter performs schema-based transformation of messages
4. New input applied to Knowledge Bases – positions displayed on Client
5. Rules fire to relate messages to Theater Objects
6. Sit’n Service detects events on ESB, queries for alerts
7. Rules fire to produce “Single Integrated Picture”
8. Alerts and Recommendations based on battlefield movement and Intel are displayed on client

Google Earth Client

Adapter

Enterprise Service Bus

AMZI Logic Server₁

AMZI Logic Server₂

Situational Awareness Service

Message Injector

Optimized Knowledge Base 1

Optimized Knowledge Base 2

Simulated Position of Friendly Convoy

Events (GMTI, INTSUM)
ALERT: Intelligence report of unknown force in vicinity
RECOMMENDATION: Take alternate route
Summary of Findings

- OWL is expressive and meets the majority of identified DoD requirements
- SWRL has robust integration with ontologies
- SWRL does not fully support the DoD requirements identified in this use case
- Translation of OWL to executable (Prolog) environment difficult
- Translation of rule languages fairly straightforward
- Integrated framework of tools and capabilities needed to support semantic web development
Future Work

- Develop meta-rules to select rule set
- Completion of SWORIER
- Storage of instances in a database
- Extend to new domains
Year 2 Scenario: Handling Dynamic Events

Convoy moving through enemy territory

Convoy nears sniper position, changes route

Unknown GMTI reported along route

UAV retasked to investigate GMTI

UAV confirms enemy GMTI

Enemy GMTI in proximity Convoy alerted

Convoy CDR reports unexpected sandstorm

New rules of engagement applied

No need for UAV to investigate; route must change regardless, per new rules of engagement
Two Queries (Online)
What are the locations and speeds of all known units?
What are all of the alerts and recommendations?

Desired Time: a few seconds
Baseline: 33 minutes
With extensionalization: 110 milliseconds
& Dynamic Changes: 58 minutes
& Code Minimization: 130 milliseconds

Extensionalization (Offline)
Desired Time: a few hours
Baseline: CRASH
With Avoiding Reanalysis: 13 hours
& Code Minimization: 6.5 hours

Two Dynamic Changes (Online)
Apply the location and speed of a convoy
Apply the location and speed of an enemy motorized infantry unit

Desired Time: a few seconds
Baseline: 25.2 minutes
With Code Minimization: 10 milliseconds
Project Products

Stoutenburg, Suzette; Leo Obrst; Deborah Nichols; Ken Samuel; Ken Laskey; Adrian Johnson; Jason Peterson; Karen Fox. 2006. *Dynamic Service Oriented Architectures through Semantic Technology*. Submitted to 4th International Conference on Service Oriented Computing (ICSOC 2006).

Samuel, Ken; Leo Obrst; Suzette Stoutenburg; Deborah Nichols; Adrian Johnson; Ken Laskey; Jason Peterson; Karen Fox. 2006. *Transforming OWL and Semantic Web Rules to Prolog: Towards Description Logic Programs*, Federated Logic Conference, August 2006, Seattle, Washington.


Obrst, Leo; Patrick Cassidy; Steve Ray; Barry Smith; Dagobert Soergel; Matthew West; Peter Yim. 2006. The 2006 Upper Ontology Summit Joint Communiqué. *Journal of Applied Formal Ontology*. Forthcoming.
