SAWA: An Assistant for Higher-Level Fusion and Situation Awareness

Christopher J. Matheus, Mieczyslaw M. Kokar, Kenneth Baclawski, Jerzy A. Letkowski, Catherine Call, Michael Hinman, John Salerno, Douglas Boulware
SAW Process

1. **SAW Ontology**
   - Commander
   - Knowledge Engineer

2. **Ontologies** (classes, relations, rules)
   - Created by Ontologies

3. **Meta Annotations Process**
   - Meta Annotation of Rules
   - Determine Relevance
   - Relevant Relations

4. **Final Step**
   - Fusion Ontology
   - Uncertainty Ontology
   - New Relations
Supply Logistics Scenario

• Scenario for supplying units using ground transports via roads that may not be under friendly control
• Configuration files control types and quantities of resources, transports, suppliers and consumers
• Generates events based on our SAW Core, Supply Logistics and Event Ontologies
OWL: Web Ontology Language

- W3C’s ontology language for the Semantic Web
- Mainly intended to provide means for describing web content in a form amiable to automated reasoning
- Used to construct OWL ontologies that define domain specific classes and properties along with the inherent constraints among them
- OWL ontologies are then used to describe specific instances or situations in the given domain
- Built on top of RDF and XML
- Three flavors: Full, DL and Lite
SWRL

• W3C’s Semantic Web Rule Language
• Extends representational power of OWL by adding implication in the form of Horn Clauses (i.e., a form of if-then rules)
• Leverages the descriptive capabilities of OWL DL
• Leverages the rule and variable syntax of RuleML
SWRL Pros and Cons

• Pros:
  – Formal Foundation
  – W3C Effort
  – Based on RuleML
  – Can connect to OWL Ontologies

• Cons:
  – Limited to Binary Relations (makes higher-order relations difficult to represent)
  – Verbose/complex syntax
  – No Existential Quantification in rule heads (makes higher-order relations impossible to infer – we thus are ignoring this constraint with the expectation it will be removed)
  – Still evolving
Event Ontology

Event
- time: &xsd;dateTime
- source: &rdfs;Resource

Object
- id: &rdfs;Resource
- type: &rdfs;Resource

Attribute
- name: &rdfs;Resource
- value: &rdfs;Literal
- units: &rdfs;Literal
- certainty: &xsd;float

Event
- time (max:1)
- object

Object
- source (max:1)
- id (max:1)
- type (max:1)

Attribute
- name (max:1)
- value (max:1)
- units (max:1)
- certainty (max:1)

rdfs:Resource
rdfs:Literal
xsd:float
SAWA Architecture

SAWA

Knowledge Management
- ConsVISor
  - Consistency Checker
- Protégé
  - Ontology Editor
- SWRLed
  - Rule Editor

Runtime System
- SMC
- RMA
- EMC
- TDB

GUI

Events

Standing relations (goals) & queries
ConsVISor Consistency Checker

ConsVISor Report
A service of Versatile Information Systems, Inc.

For http://vistology.com/ont/tests/military3.owl

Consistency Verification Summary

Test Outcome: The result of the consistency check of http://vistology.com/ont/tests/military3.owl at level full is: inconsistent.

Messages

<table>
<thead>
<tr>
<th>Info</th>
<th>Warn</th>
<th>Error</th>
<th>Fatal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CardinalityConstraint</td>
<td>No warning messages!</td>
<td>DisjointnessFailure</td>
<td>No fatal error messages!</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DisjointnessFailure</td>
<td></td>
</tr>
</tbody>
</table>

Consistency Verification Detail
Resource Locations Are Shown in Brackets[row.col]

Symptom: CardinalityConstraint
Message: A cardinality constraint with cardinality 1 was not satisfied.
Axiom Violated: owl_cardinality
Severity: info

Statements:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>sym:property</td>
<td>b_anon004 [a:22]</td>
<td>owl:property</td>
<td>b_containedIn [a:51]</td>
</tr>
<tr>
<td>sym:restriction</td>
<td>b:Unit [a:19]</td>
<td>rdfs:subClassOf</td>
<td>b_anon004 [a:22]</td>
</tr>
<tr>
<td>sym:constraint</td>
<td>b_anon004 [a:22]</td>
<td>owl:cardinality</td>
<td>1</td>
</tr>
<tr>
<td>sym:instance</td>
<td>b:Easy2 [a:71]</td>
<td>rdf:type</td>
<td>b:Unit [a:19]</td>
</tr>
<tr>
<td>sym:unsatisfied</td>
<td>2</td>
<td>sym:equal</td>
<td>1</td>
</tr>
</tbody>
</table>
RuleVISor Rule Editor

- Graphical SWRL Editor
- Support for
  - all RuleML capabilities (everything in SWRL from ruleml: namespace)
  - all new SWRL elements (from swrlx: namespace, e.g., swrlx:builtin)
- Does not support arbitrary embedded OWL constructs
  - OWL Ontologies are required to be external
- Ontologies used as basis for rule building blocks
RuleVISor GUI
Supply Logistics Rule Set

```xml
<rule rlab="has Supply Line">
  <body>
    <hsl:inRegion sub="?unit" data="?region"/>
    <hsl:isSuppliable sub="?region" data="true"/>
  </body>
  <head>
    <hsl:hasSupplyLine sub="?unit" data="true"/>
  </head>
</rule>

<rule rlab="isSuppliable">
  <body>
    <hsl:hasSupplyStation sub="?region" data="true"/>
    <hsl:underFriendlyControl sub="?region" data="true"/>
  </body>
  <head>
    <hsl:isSuppliable sub="?region" data="true"/>
  </head>
</rule>

<rule rlab="isSuppliable2">
  <body>
    <hsl:connects sub="?road" data="?region1"/>
    <hsl:connects sub="?road" data="?region2"/>
    <swrlb:notEqual arg1="?region1" arg2="?region2"/>
    <hsl:isPassable sub="?road" data="true"/>
    <hsl:isSuppliable sub="?region2" data="true"/>
  </body>
  <head>
    <hsl:isSuppliable sub="?region1" data="true"/>
  </head>
</rule>

<rule rlab="underFriendlyControl">
  <body>
    <hsl:inRegion sub="?unit" data="?region"/>
    <hsl:memberOf sub="?unit" data="?force"/>
    <hsl:FriendlyForce ind="?force"/>
  </body>
  <head>
    <hsl:underFriendlyControl sub="?region" data="true"/>
  </head>
</rule>

<rule rlab="isPassable">
  <body>
    <hsl:connects sub="?road" data="?regionA"/>
    <hsl:connects sub="?road" data="?regionB"/>
    <swrlb:notEqual arg1="?regionA" arg2="?regionB"/>
    <hsl:underFriendlyControl sub="?regionA" data="?force1"/>
    <hsl:underFriendlyControl sub="?regionB" data="?force2"/>
  </body>
  <head>
    <hsl:isPassable sub="?road" data="true"/>
  </head>
</rule>

<rule rlab="hasSupplyStation">
  <body>
    <hsl:inRegion sub="?X" data="?region"/>
    <hsl:SupplyStation ind="?X"/>
  </body>
  <head>
    <hsl:hasSupplyStation sub="?region" data="true"/>
  </head>
</rule>
```
SAWA Runtime

- SMC
- RA
- TDB
- EMC
- GUI

Events:

Sensors
Triple Data Base

- Stores RDF/OWL triples
  - E.g., (predicate subject object)
- Built on Jess (Java Expert System Shell based on CLIPS)
- Infers implicit triples from events and OWL axioms
- Detects inconsistencies
- Tracks performance metrics of inference engine
- Supports OWL-QL (OWL Query Language) formerly known as DQL
Query Capabilities

• Full support of OWL Query Language – DARPA sponsored effort
• Permits Queries over patterns in triples
  – e.g., (consumes ?user “food”) (type ?user “company”)
  – Results returned as variable bindings
• “What If” Query capability
  – assumptions posited and then retracted after query returns
• Writing queries and interpreting results can be challenging
• Prompted move to implement simple GUI
Query Interface

- Simplifies query construction
- Initial version based on static templates with fill-in slots
- Demo
- Extensions:
  - constraints between slot values enforced by GUI
  - automatic generation of candidate templates
  - free-form query wizard
Supply Logistics Ontology
SAWA Runtime GUI
Conclusion

- SAWA is a general purpose assistant for situation awareness:
  - monitors the evolution of relevant higher-order relations within a situation.
  - supports formal reasoning techniques for level-2 fusion.
  - based on the Semantic Web languages OWL and SWRL.
  - performs relevance reasoning.
- The domain ontology and rules are constructed and checked using an ontology editor, rule editor and consistency checker.
- At runtime events are processed to determine relevance and to infer higher-order relations.
- As higher-order relations are detected they are passed to the GUI, which displays them in both tabular and graphical forms.
- The query capability allows for both ordinary and “what if” queries.