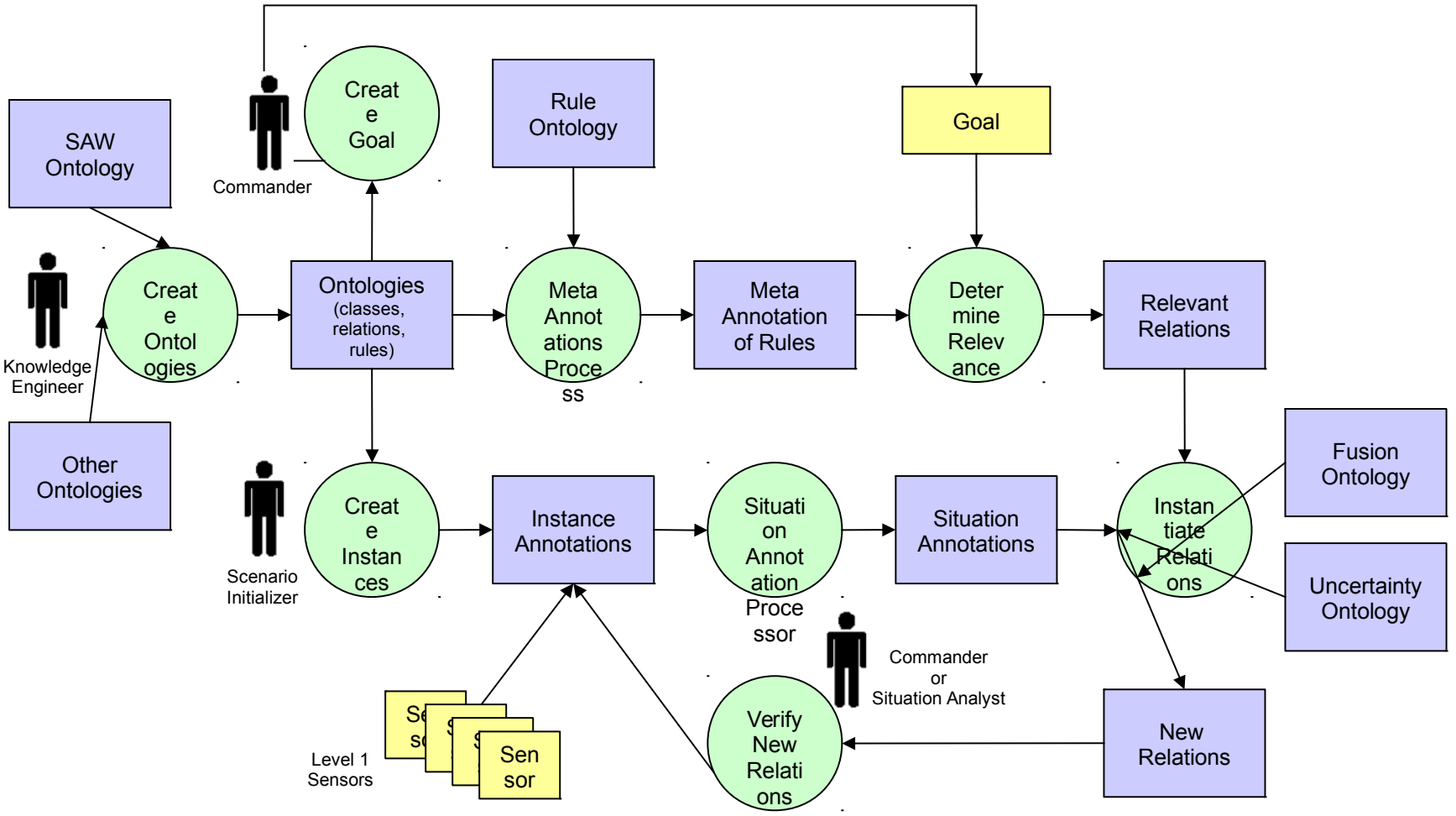
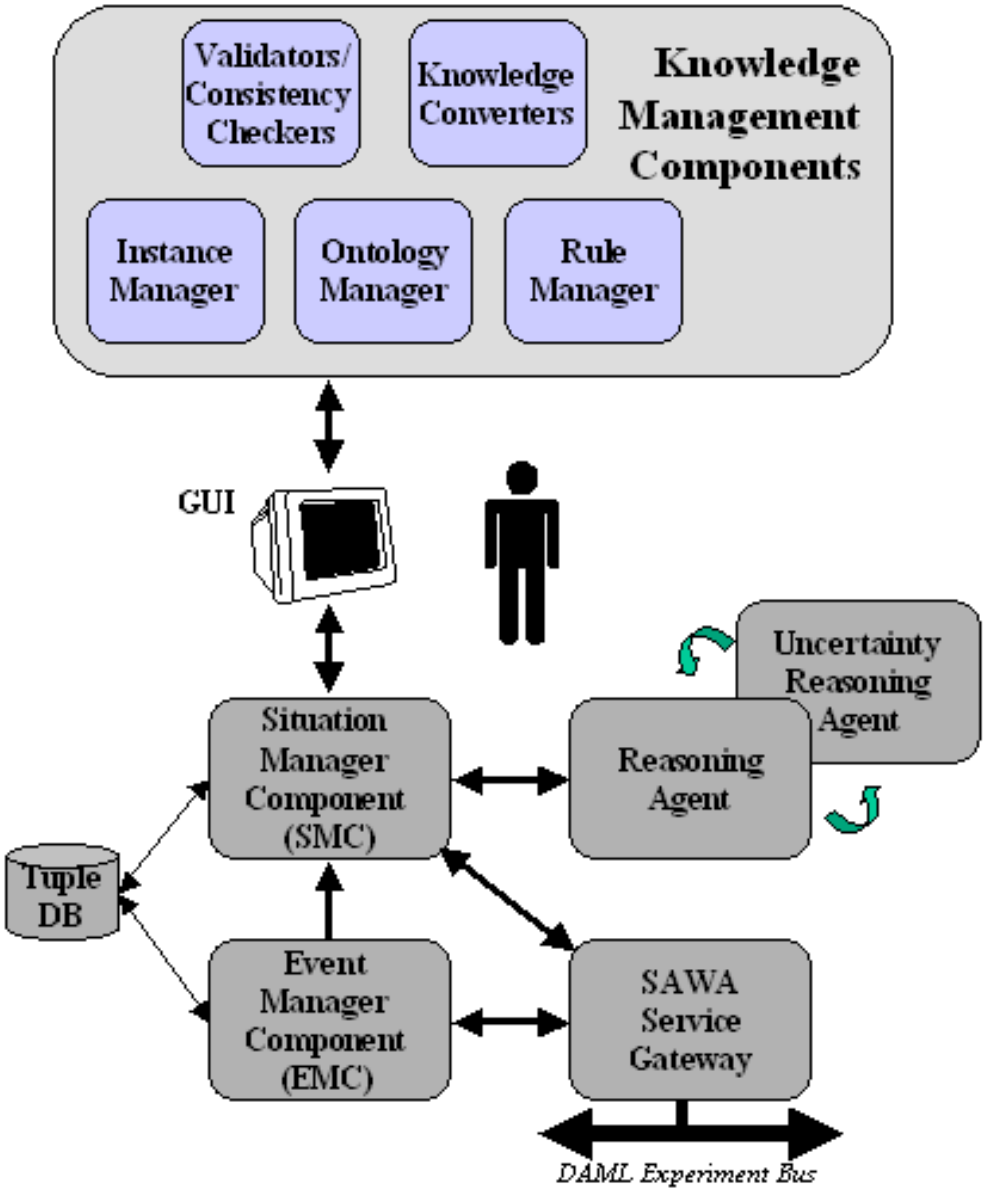
# SAWA: An Assistant for Higher-Level Fusion and Situation Awareness

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

# SAW Process


SAW Ontology
Commander
Create Goal
Rule Ontology
Goal
Knowledge Engineer
Create Ontologies
Ontologies (classes, relations, rules)
Meta Annotations Process
Meta Annotation of Rules
Determine Relevance
Relevant Relations
Other Ontologies
Scenario Initializer
Create Instances
Instance Annotations
Situation Annotation Processor
Situation Annotations
Instantiate Relations
Fusion Ontology
Uncertainty Ontology
Commander or Situation Analyst
Level 1 Sensors
Sensor
Verify New Relations
New Relations

# SAW Assistant (SAWA)



# 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

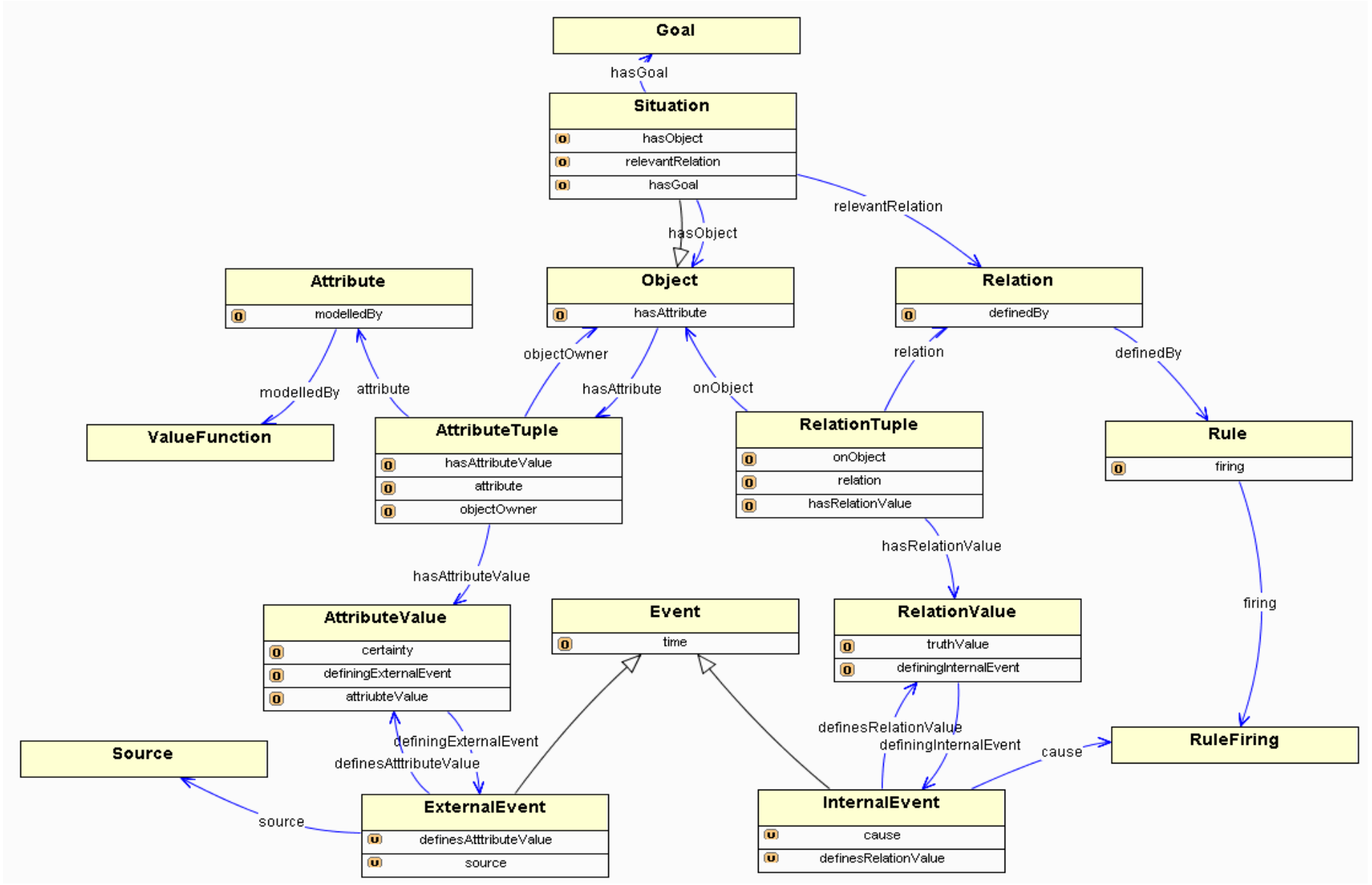
# SWRL

- 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

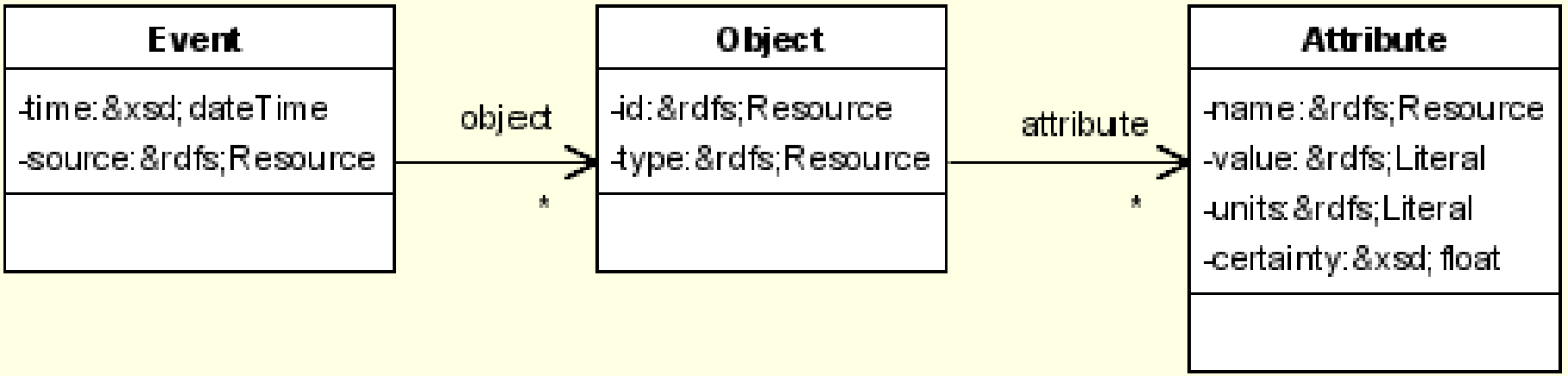
# SWRL Pros and Cons

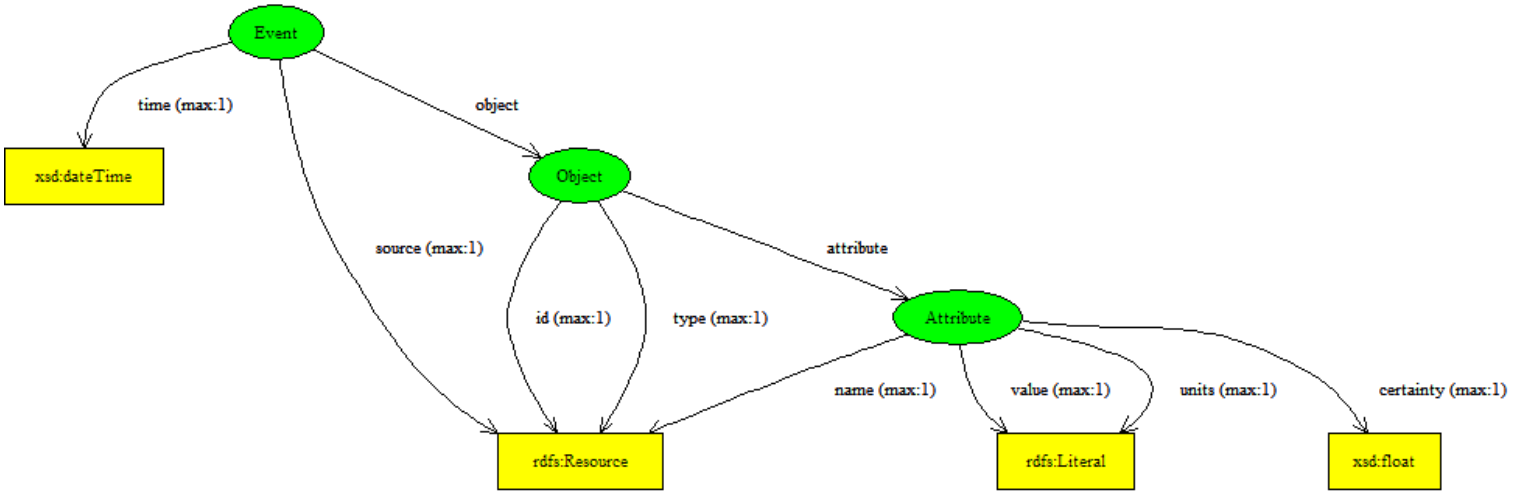
- Pros:
  - Formal Foundation
  - W3C Effort
  - Based on RuleML
  - Can connect to OWL Ontologies
- Cons:
  - Limited to Binary Relations (makes higher-order relations difficult to represent)
  - Verbose/complex syntax
  - No Existential Quantification in rule heads (makes higher-order relations impossible to infer – we thus are ignoring this constraint with the expectation it will be removed)
  - Still evolving

# SAW Core Ontology

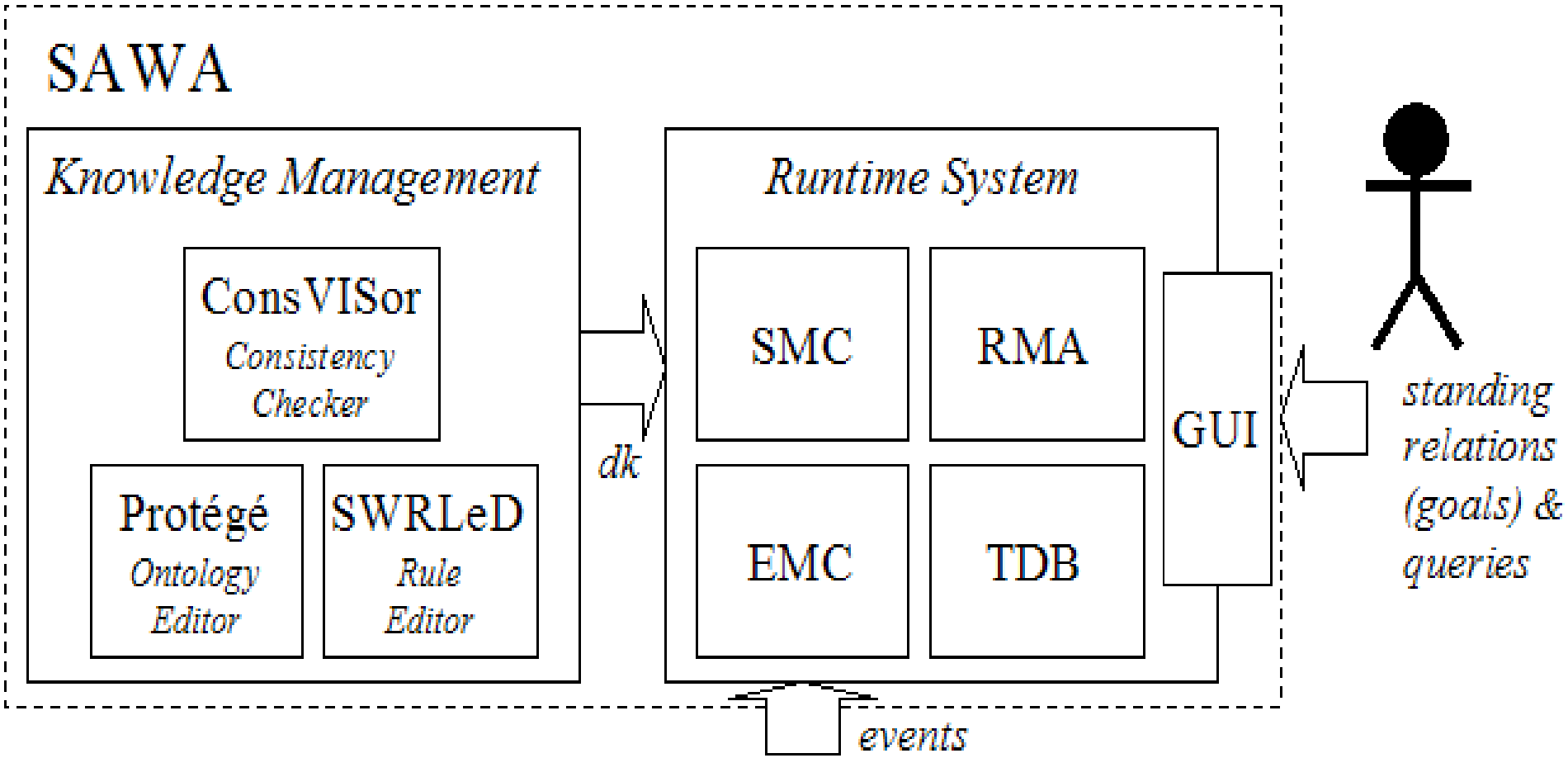


# Event Ontology

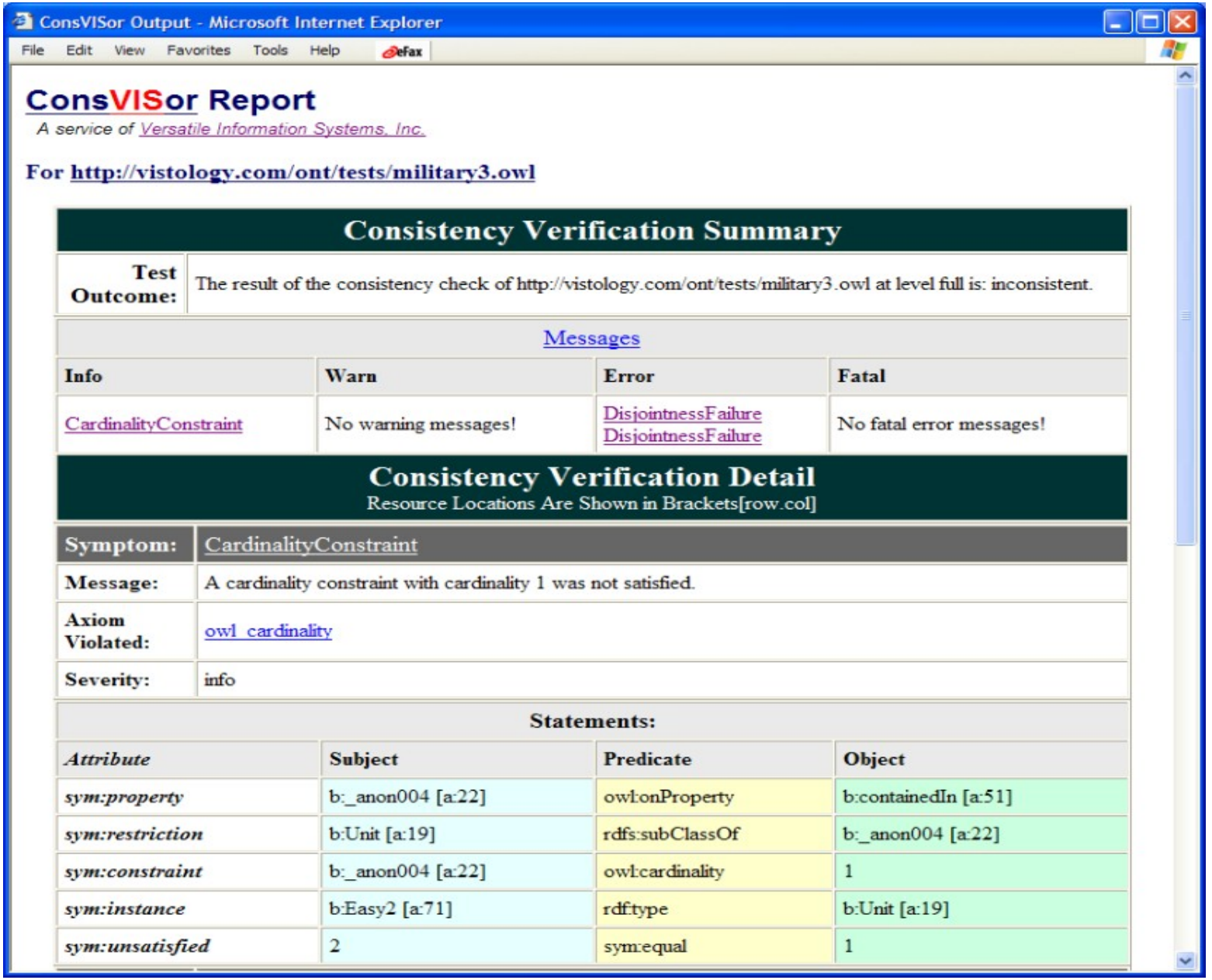


# SAWA Architecture



# ConsVISor Consistency Checker



ConsVISor Output - Microsoft Internet Explorer

File Edit View Favorites Tools Help eFax

## 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

| Info | Warn | Error | Fatal |
|---|---|---|---|
| CardinalityConstraint | No warning messages! | DisjointnessFailure<br>DisjointnessFailure | No fatal error messages! |

### 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 |

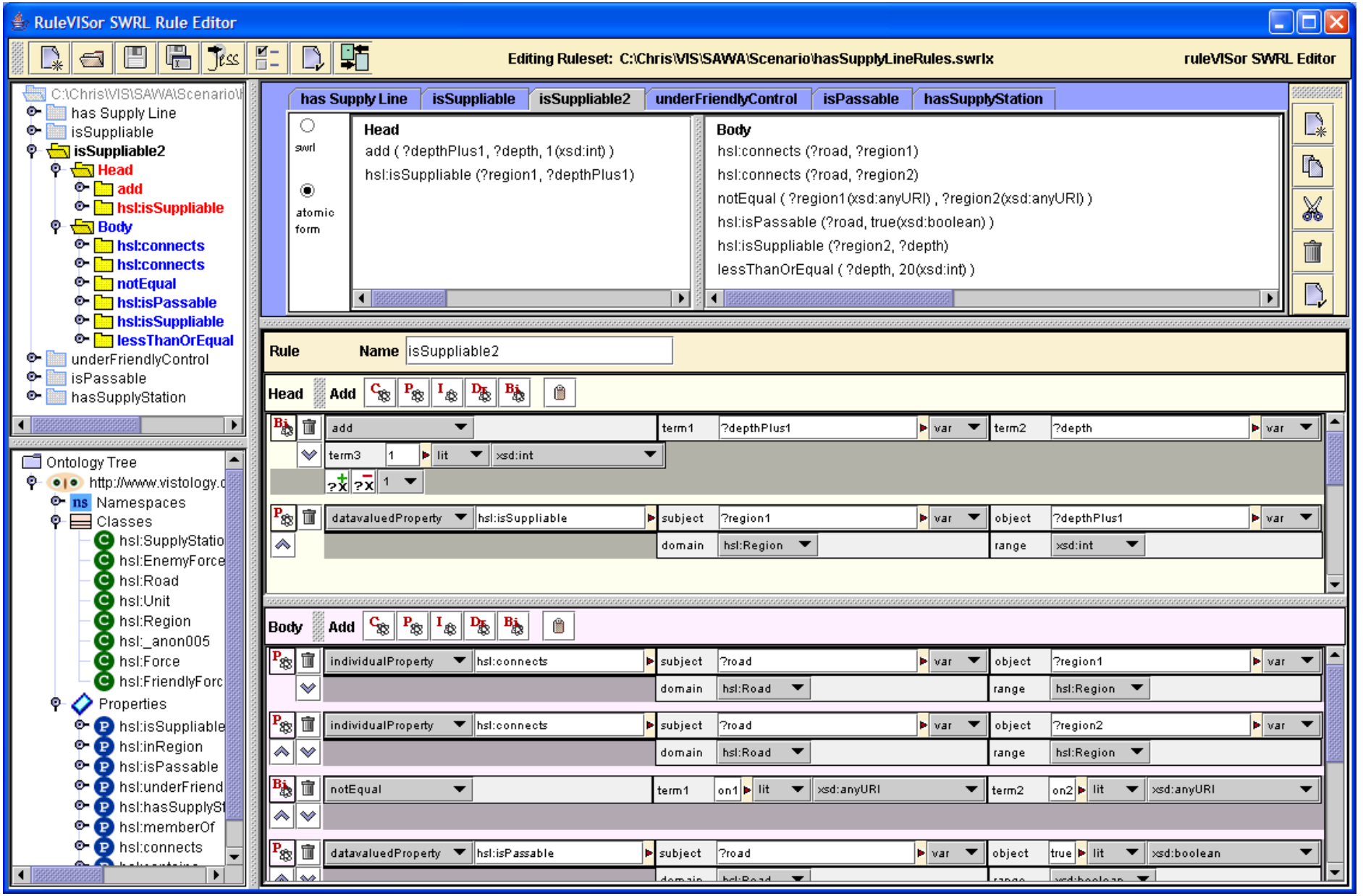
**Statements:**

| *Attribute* | Subject | Predicate | Object |
|---|---|---|---|
| *sym:property* | b:_anon004 [a:22] | owl:onProperty | b:containedIn [a:51] |
| *sym:restriction* | b:Unit [a:19] | rdfs:subClassOf | b:_anon004 [a:22] |
| *sym:constraint* | b:_anon004 [a:22] | owl:cardinality | 1 |
| *sym:instance* | b:Easy2 [a:71] | rdf:type | b:Unit [a:19] |
| *sym:unsatisfied* | 2 | sym:equal | 1 |

# 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

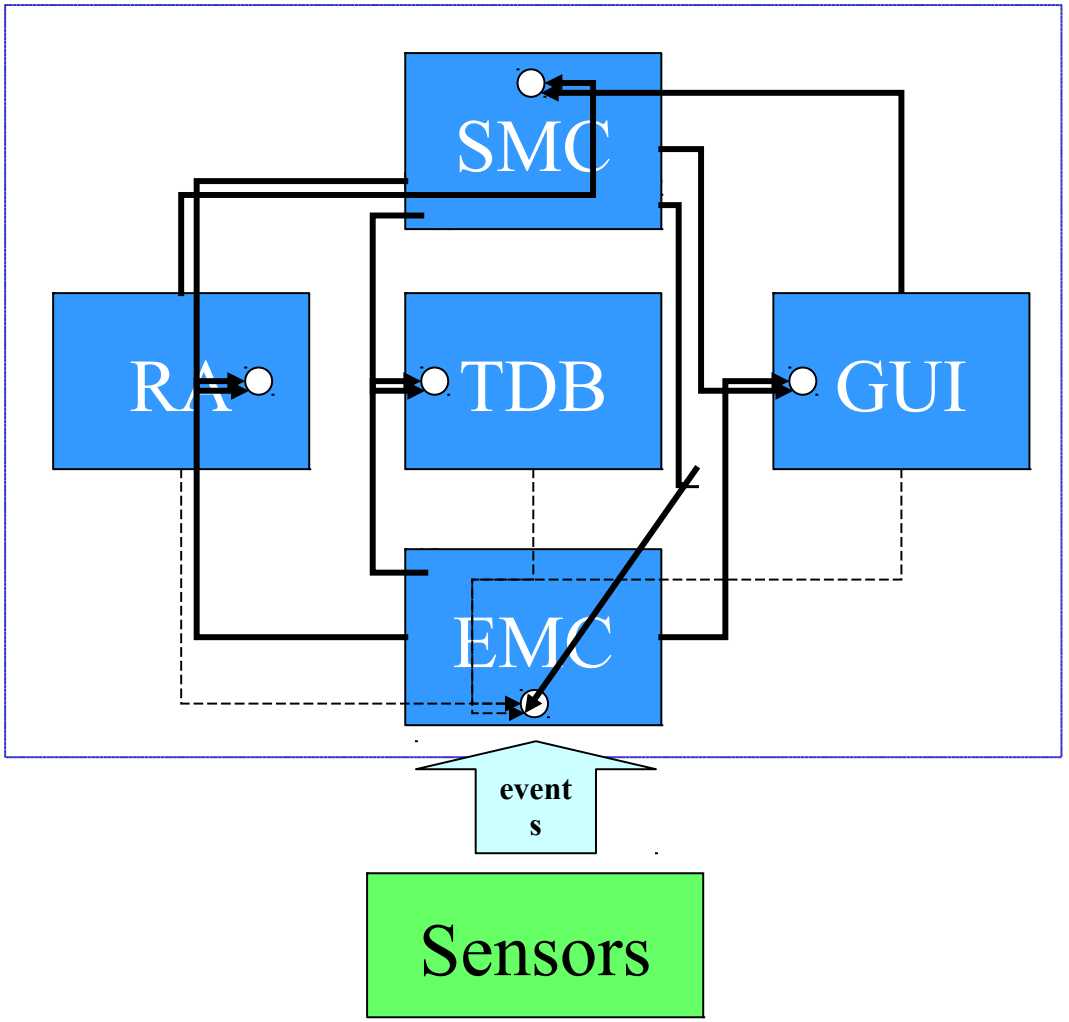


# Supply Logistics Rule Set

```
<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



# 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

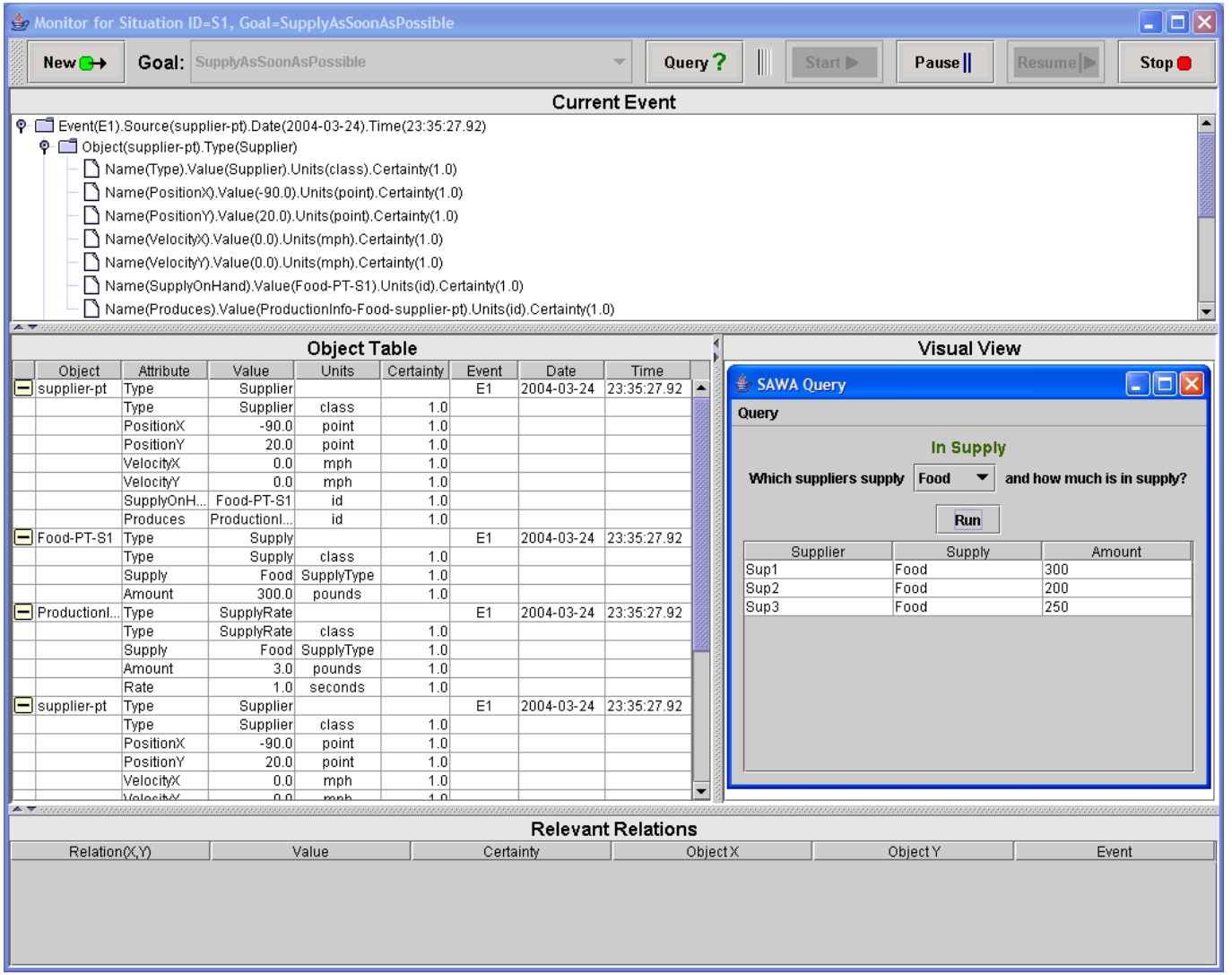
# Query Capabilities

- 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

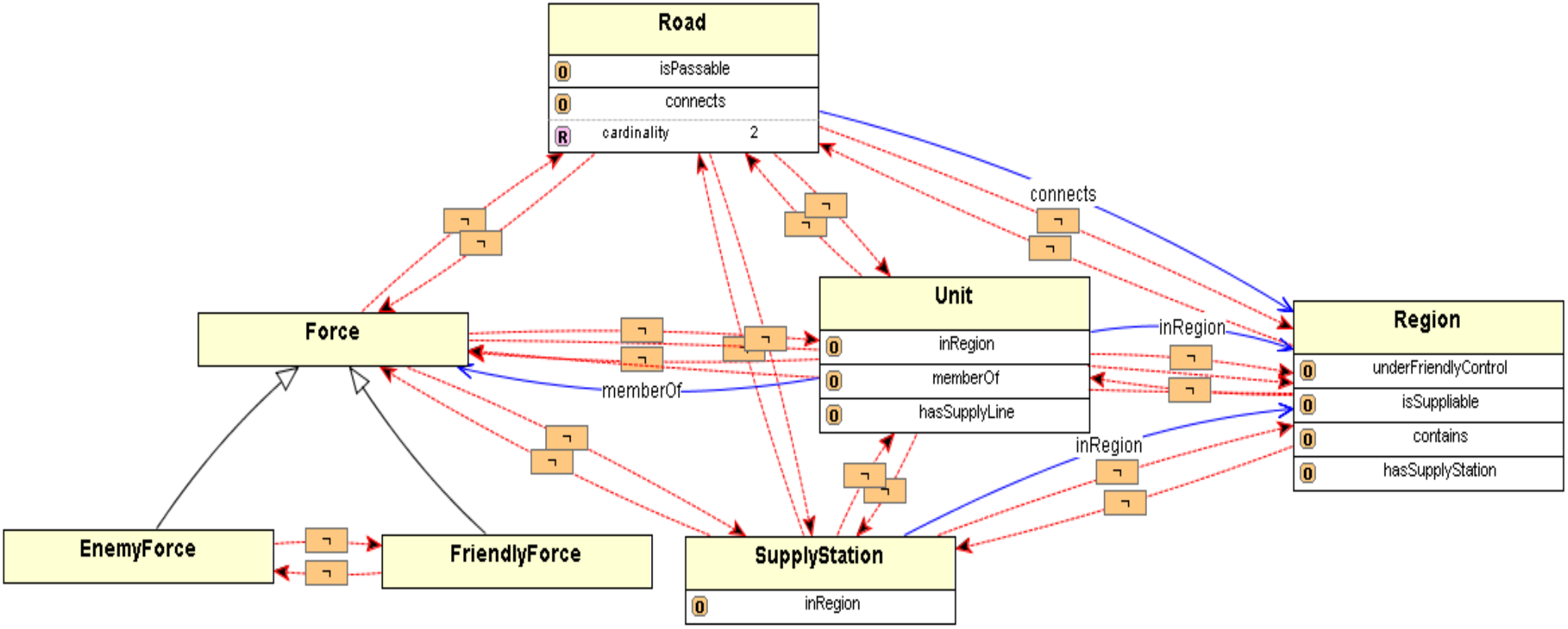
# Query Interface

- 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

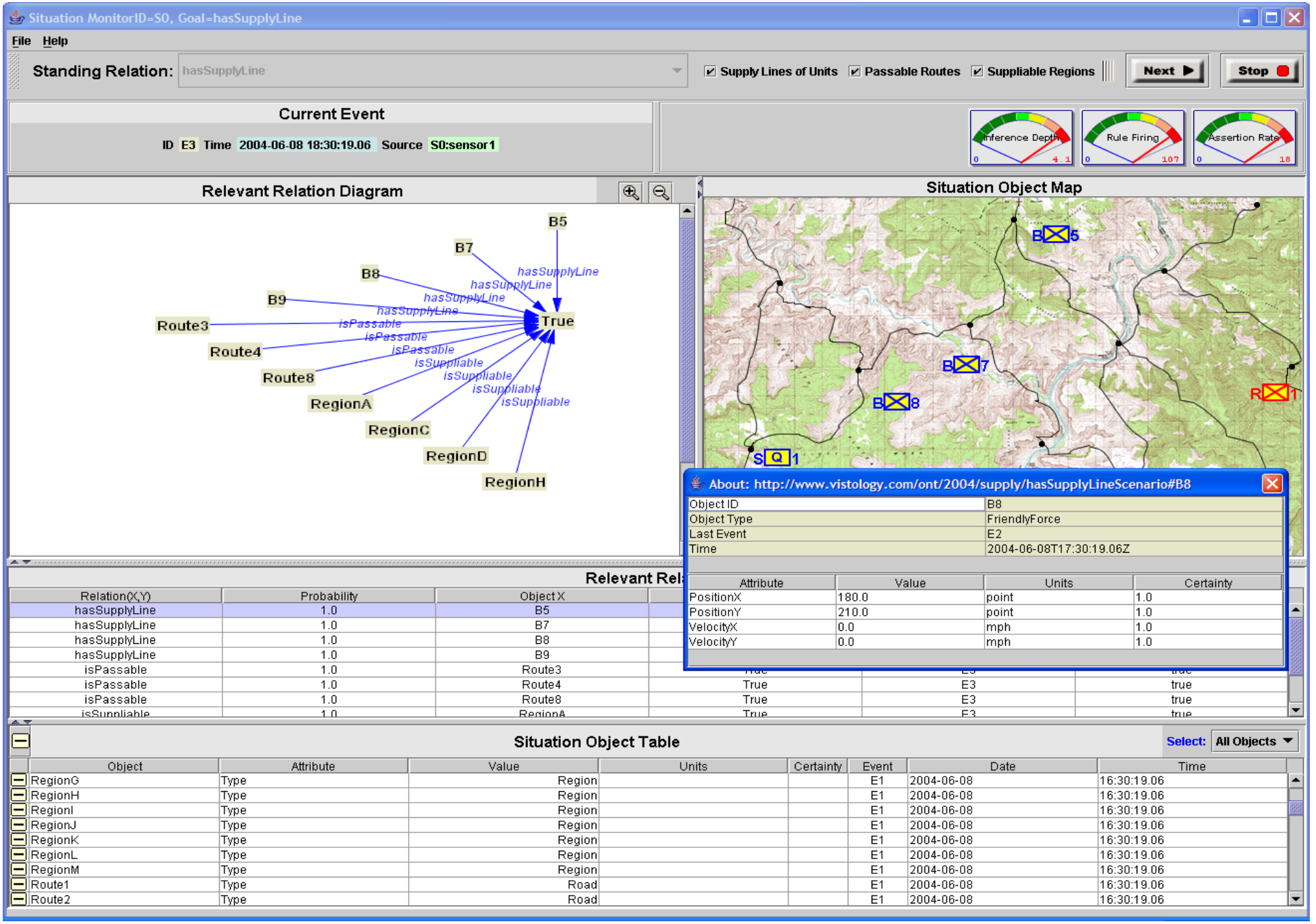
# Query GUI Screenshot



# 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.