SWRL

Semantic Web Rule Language

ή

Γλώσσα Κανόνων Σημασιολογικού Ιστού

Τμήμα Μηχανικών Η/Υ &
Πληροφορικής

Μια γενική ιδέα…

- Η SWRL(Semantic Web Rule Language) ή στα ελληνικά Γλώσσα Κανόνων Σημασιολογικού Ιστού) είναι μια πρόταση για μια γλώσσα αναπαράστασης γνώσης υπό μορφή κανόνων στο Σημασιολογικό Ιστό, που συνδυάζει τις υπογλώσσες της γλώσσας οντολογίας Ιστού OWL (OWL DL και Lite) με εκείνες της γλώσσας σήμανσης κανόνα (μοναδιαίο/δυαδικό Datalog).

Κανόνες (Rules)

- Ένας κανόνας σε SWRL έχει τη μορφή
  - $\mathbf{B_1, \ldots, B_n \rightarrow A_1, \ldots, A_m}$
  - Τα κόμματα δείχνουν τη σύζευξη και στις δύο πλευρές
  - $\mathbf{A_1, \ldots, A_m, B_1, \ldots, B_n}$ μπορεί να είναι μορφής **C(x)**, **P(x,y)**, **sameAs(x,y)**, ή **differentFrom(x,y)** όπου το **C** είναι μια περιγραφή της OWL, η **P** μια OWL property , και **x**, **y** είναι μεταβλητές Datalog , σταθερές OWL, ή τιμές δεδομένων

SWRL Properties

- Εάν η κεφαλή ενός κανόνα έχει περισσότερα από ένα άτομα, ο κανόνας μπορεί να μετασχηματιστεί σε ένα ισοδύναμο σύνολο κανόνων με ένα άτομο στην κεφαλή
- Οι εκφράσεις, όπως οι περιορισμοί, μπορούν να εμφανιστούν στην κεφαλή ή το σώμα ενός κανόνα
- Αυτό το χαρακτηριστικό γνώρισμα προσθέτει τη σημαντική εκφραστική δύναμη στην OWL, αλλά με κόστος την υψηλή τιμή αναποφασιστικότητας

SWRL Σύνταξη

Η σύνταξη στην SWRL καθορίζεται ως εξής:

Atom ← C(i) |D(v) |R(i, j) |U(i; v)| builtIn(p, v1,…, vn) | $i = j$ | $i \neq j$

- C = κλάση
- R = ιδιότητα αντικειμένου
- i; j = μεταβλητές ονομάτων του αντικειμένου ή μεμονομένα ονόματα μεταβλητών
- v_1,… v_n = τύποι μεταβλητών και δεδομένων
- p = Built-in name

- D = Data type
- U = Data type Property

Παράδειγμα κανόνας

hasParent(?x1,?x2) ^ hasBrother(?x2,?x3) → hasUncle(?x1,?x3))

Ένα δεύτερο παράδειγμα

has_father(?a,?b) ^ has_married(?b,?c) → female(?c) ^ has_mother(?a,?c)

Αξίζει να σημειωθεί πως ένας κανόνας με πολλαπλά άτομα στην κεφαλή του ισοδυναμεί με πολλαπλούς κανόνες που έχουν το ίδιο σώμα με τον αρχικό κανόνα και ένα άτομο στην κεφαλή τους

ισοδυναμεί με τους ακόλουθους κανόνες :

has_father(?a,?b) ^ has_married(?b,?c) → female(?c)

και

has_father(?a,?b) ^ has_married(?b,?c) → has_mother(?a,?c)

Σημείωση

- Ο συνδυασμός κανόνων μαζί με οντολογίες παρέχει μεγάλη εκφραστικότητα, καθώς προσπαθεί, και εν μέρει καταφέρνει, να συνδυάσει τα πλεονεκτήματα της κλασικής λογικής και του λογικού προγραμματισμού.
- Όμως, κάποια βασικά μειονεκτήματα παραμένουν. Θα πρέπει να τονιστεί πως η γλώσσα SWRL αποτελεί επέκταση της OWL και αυτό έχει ως αποτέλεσμα να λειτουργεί σε συνέπεια με την υπόθεση του ανοικτού κόσμου (Open-World Assumption), όπως και η OWL.
- Έτσι, η SWRL δεν υποστηρίζει τελεστή *not*, σε αντίθεση με τα περισσότερα περιβάλλοντα κανόνων που λειτουργούν σε περιβάλλοντα κλειστού κόσμου.
- Τέλος, θα πρέπει να σημειωθεί πως αφού οι κανόνες SWRL έχουν μορφή κανόνων-Horn, δεν επιτρέπουν διάζευξη ούτε στο σώμα ούτε στην κεφαλή τους.

Λεκτική Περιγραφή Κανόνα
Vs
Λογική Περιγραφή Κανόνα:

Εάν ένας καλλιτέχνης x έχει υιοθετήσει μια τεχνοτροπία y και έχει ένα δημιούργημα z τότε το δημιούργημα z είναι της τεχνοτροπίας y.

Η λεκτική περιγραφή του κανόνα

Η λογική περιγραφή του κανόνα

artist (x) ^ style(y) ^ artist Style (x,y) ^ creator(x,z) → artistfactStyle(z,y)

Οντολογική Περιγραφή Κανόνα (σε SWRL σύνταξη):

Το προηγούμενο παράδειγμα Λογικής περιγραφής Κανόνα μπορεί να μετατραπεί με μεγάλη ευκολία σε μία οντολογική περιγραφή κανόνα για την οποία η σύνταξη αλλά και η εξήγηση φαίνεται παρακάτω

SWRL σύνταξη

RDF Concrete Syntax

```
<swrl:Variable rdf:ID="x"/>
<swrl:Variable rdf:ID="y"/>
<swrl:Variable rdf:ID="z"/>
<ruleml:Imp>
<ruleml:body rdf:parseType="Collection">
<swrl:ClassAtom>
  <swrl:classPredicate rdf:resource="Artist"/>
  <swrl:argument1 rdf:resource="#x" />
</swrl:ClassAtom>
<swrl:ClassAtom>
  <swrl:classPredicate rdf:resource="Style"/>
  <swrl:argument1 rdf:resource="#y" />
</swrl:ClassAtom>
<swrl:IndividualPropertyAtom>
  <swrl:propertyPredicate
      rdf:resource="artistStyle"/>
  <swrl:argument1 rdf:resource="#x" />
  <swrl:argument2 rdf:resource="#y" />
</swrl:IndividualPropertyAtom>
```

```
<swrl:IndividualPropertyAtom>
  <swrl:propertyPredicate
      rdf:resource="creator"/>
  <swrl:argument1 rdf:resource="#x" />
  <swrl:argument2 rdf:resource="#z" />
</swrl:IndividualPropertyAtom>
</ruleml:body>
<ruleml:head rdf:parseType="Collection">
<swrl:IndividualPropertyAtom>
   <swrl:propertyPredicate
      rdf:resource="artifactStyle"/>
   <swrl:argument1 rdf:resource="#z" />
   <swrl:argument2 rdf:resource="#y" />
</swrl:IndividualPropertyAtom>
</ruleml:head>
</ruleml:Imp>
```

XML Concrete Syntax = Συνδυασμός OWL XML synatx και RuleML XML syntax

Αναλυτικά

```
<swrl:Variable rdf:ID="x"/>
<swrl:Variable rdf:ID="y"/>
<swrl:Variable rdf:ID="z"/>
```

Ορισμοί μεταβλητών

Ορισμός και αντιστοίχιση κατηγορημάτων του σώματος
Συγκεκριμένά:
artist (x)
style(y)
artist Style (x,y)
creator(x,z)

```
<ruleml:body rdf:parseType="Collection">
<swrl:ClassAtom>
  <swrl:classPredicate rdf:resource="Artist"/>
  <swrl:argument1 rdf:resource="#x" />
</swrl:ClassAtom>
<swrl:ClassAtom>
  <swrl:classPredicate rdf:resource="Style"/>
  <swrl:argument1 rdf:resource="#y" />
</swrl:ClassAtom>
<swrl:IndividualPropertyAtom>
  <swrl:propertyPredicate rdf:resource="artistStyle"/>
  <swrl:argument1 rdf:resource="#x" />
  <swrl:argument2 rdf:resource="#y" />
</swrl:IndividualPropertyAtom>
<swrl:IndividualPropertyAtom>
  <swrl:propertyPredicate rdf:resource="creator"/>
  <swrl:argument1 rdf:resource="#x" />
  <swrl:argument2 rdf:resource="#z" />
</swrl:IndividualPropertyAtom>
</ruleml:body>
```

Αναλυτικά(2)

```
<ruleml:head rdf:parseType="Collection">
<swrl:IndividualPropertyAtom>
  <swrl:propertyPredicate rdf:resource="artifactStyle"/>
  <swrl:argument1 rdf:resource="#z" />
  <swrl:argument2 rdf:resource="#y" />
</swrl:IndividualPropertyAtom>
</ruleml:head>
```

Ορισμός και αντιστοίχιση κατηγορημάτος της κεφαλής Συγκεκριμένά:

artistfactStyle(z,y)

```
<ruleml:Imp>
…
</ruleml:Imp>
```

Αυτό το element επιτρέπει να πούμε ότι κάθε σύνδεσμος που ικανοποιεί το body του κανόνα πρέπει επίσης να ικανοποιήσει το head του ίδιου κανόνα

SWRL σύνταξη

XML Concrete Syntax

```
<swrlx:datarangeAtom>
...
</swrlx:datarangeAtom>
```

Η περιγραφή σε ένα άτομικό datarange μπορεί να είναι μια ταυτότητα datatype, ή μπορεί να είναι ένα σύνολο literals.

Παράδειγμα:

```
<swrlx:datarangeAtom>
  <owlx:Datatype owlx:name="&xsd;int" />
  <ruleml:var>x1</ruleml:var>
</swrlx:datarangeAtom>
```

```
<ruleml:var>xsd:string</ruleml:var>
```

Καθορίζει απλά την ύπαρξη μιας μεταβλητής. Αυτό λαμβάνεται από το RuleML namespace.

Παράδειγμα:

```
<ruleml:var>x1</ruleml:var>
```

SWRL και Querying: SQWRL

- Η SWRL είναι μια γλώσσα κανόνων, όχι μια γλώσσα διατύπωσης ερωτήσεων
- Εντούτοις, ένας κανόνας προηγουμένως μπορεί να αντιμετωπισθεί ως προδιαγραφή ταιριάσματος σχεδίων, δηλ., μια ερώτηση.
- Με built-ins, γλωσσικές συμμορφώσεις των query extensions είναι δυνατές.
- Έχει αναπτυχθεί μια SWRL-based query γλώσσα που καλείται SQWRL

Παράδειγμα στην SQWRL Query

Ερώτημα:
«Επιστρέψτε όλους τους ενηλίκους σε μια οντολογία»

Person(?p) ^ hasAge(?p,?age) ^ swrlb:greaterThan(?age,17)

→ sqwrl:select(?p, ?age)

Άλλη SQWRL Query

«Επιστρέψτε όλους τους ενηλίκους σε μια οντολογία διατεταγμένα»

Person(?p) ^ hasAge(?p, ?age) ^ swrlb:greaterThan(?age, 17) →
sqwrl:select(?p) ^ sqwrl:orderBy(?age)

Γιατί χρήση SWRL ως βάση για τη γλώσσα διατύπωσης ερωτήσεων είναι ελκυστική

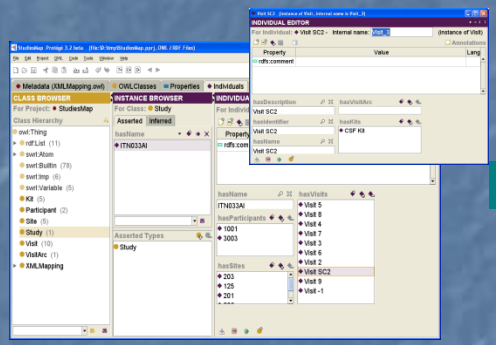
- Καθαρότερη σημασιολογία από την SPARQL
- OWL-based, όχι RDF-based
- Πολύ εκτατή μέσω του built-ins, π.χ., χρονικές ερωτήσεις που χρησιμοποιούν το temporal built-ins

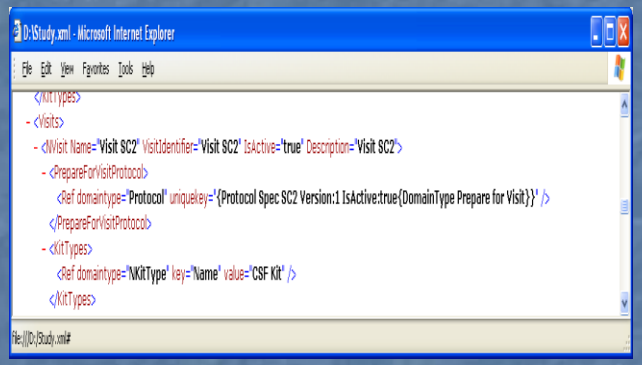
XML Mapping με την βοήθεια SWRL Mapping Rules

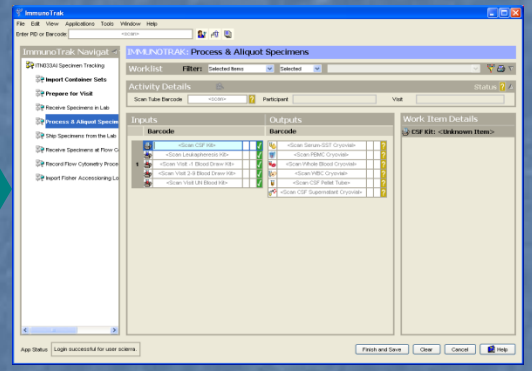
Ontology

Application

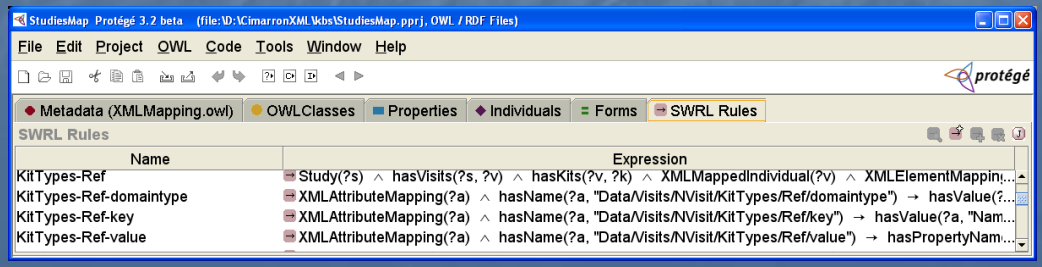
XML Document







SWRL Mapping Rules



Μηχανές της SWRL
SweetRules v2.0

Γενικά :

- SweetRules v2.0 – μηχανή εκτέλεσης κανόνων SWRL
- http://projects.semwebcentral.org/forum/forum.php?forum_id=232
- http://xml.coverpages.org/ni2005-04-25-a.html
- Το SweetRules είναι ένα pluggable σύνολο εργαλείων κανόνων για RuleML και SWRL που χαρακτηρίζουν την διαλειτουργικότητα μεταξύ Prolog, των κανόνων παραγωγής, της OWL, CommonRules, Jena-2, και διάφορων άλλων γλωσσών και inferencing με την άρνηση, τις προτεραιότητες, και τις διαδικαστικές συνδέσεις.

Άλλες Μηχανές και Validators

- swrl2clips
- Hoolet
- VIS use with JESS
- Rennes use with Protege and JESS

- SWRL Validator

Χρήσεις και Εφαρμογές της SWRL

- Αναπαράσταση των διαφόρων κανόνων που απαιτούνται για την επιλογή του καταλληλότερου μονοπατιού.
- Οι χρήστες μπορούν επίσης αν θελήσουν να περιορίσουν την εκφραστικότητα των OWL κλάσεων και περιγραφών που εμφανίζονται στους κανόνες. Ένας χρήσιμος περιορισμός στην εκφραστικότητα είναι προγράμματα λογικής περιγραφής: Description Logic Programs [Grosof et al 2003] που, π.χ. απαγορεύει την existentially-quantified γνώση στα επακόλουθα. Οι κατάλληλα ορισμένοι κανόνες SWRL μπορούν να επεκταθούν απευθείας για να επιτρέψουν τις διαδικαστικές συνδέσεις ή/και το μη μονοτονικό συλλογισμό (negation-as-failure ή/και prioritised conflict handling) των ειδών που υποστηρίζονται σε CCI συστήματα κανόνων και σε RuleML που διευκολύνουν τη διαλειτουργικότητα μεταξύ εκείνων των CCI συστημάτων κανόνων. Με τέτοια δυνατότητα μπορούμε να διευκολύνουμε το συνδυασμό της γνώσης SWRL με τη γνώση από άλλες γλώσσες κανόνων. Οι κατάλληλοι περιορισμοί μπορούν επίσης να βελτιώσουν την εμπειρική πειθώ του συλλογισμού με τους κανόνες.

APIs

test1.swrlx: sample input for jaxbtest.java

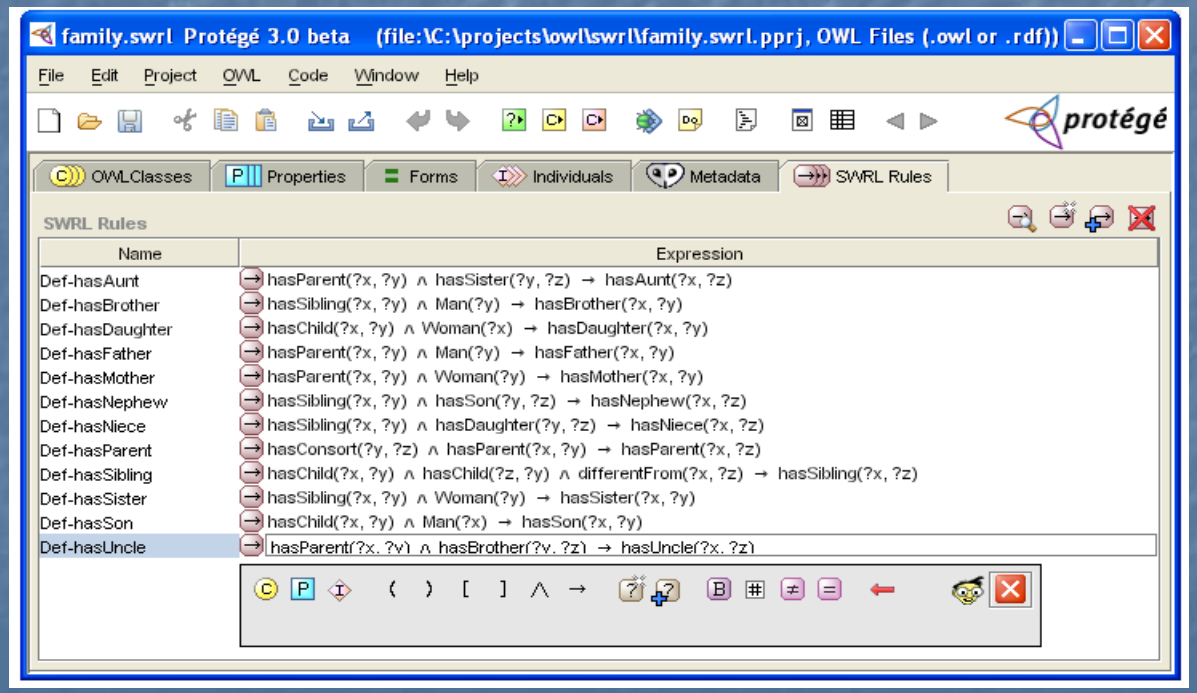
Βρίσκονται στην σελίδα:
http://www.daml.org/rules/proposal/jaxb/

Εργαλεία σχετικά με την SWRL

- Free, open source; download at: protégé.stanford.edu
- SWRLTab, Datamaster: v3.3.1
- SQWRL, XML querying: v3.4 beta
- Dynamic relational querying: 2-3 months
- Extensive documentation: http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab

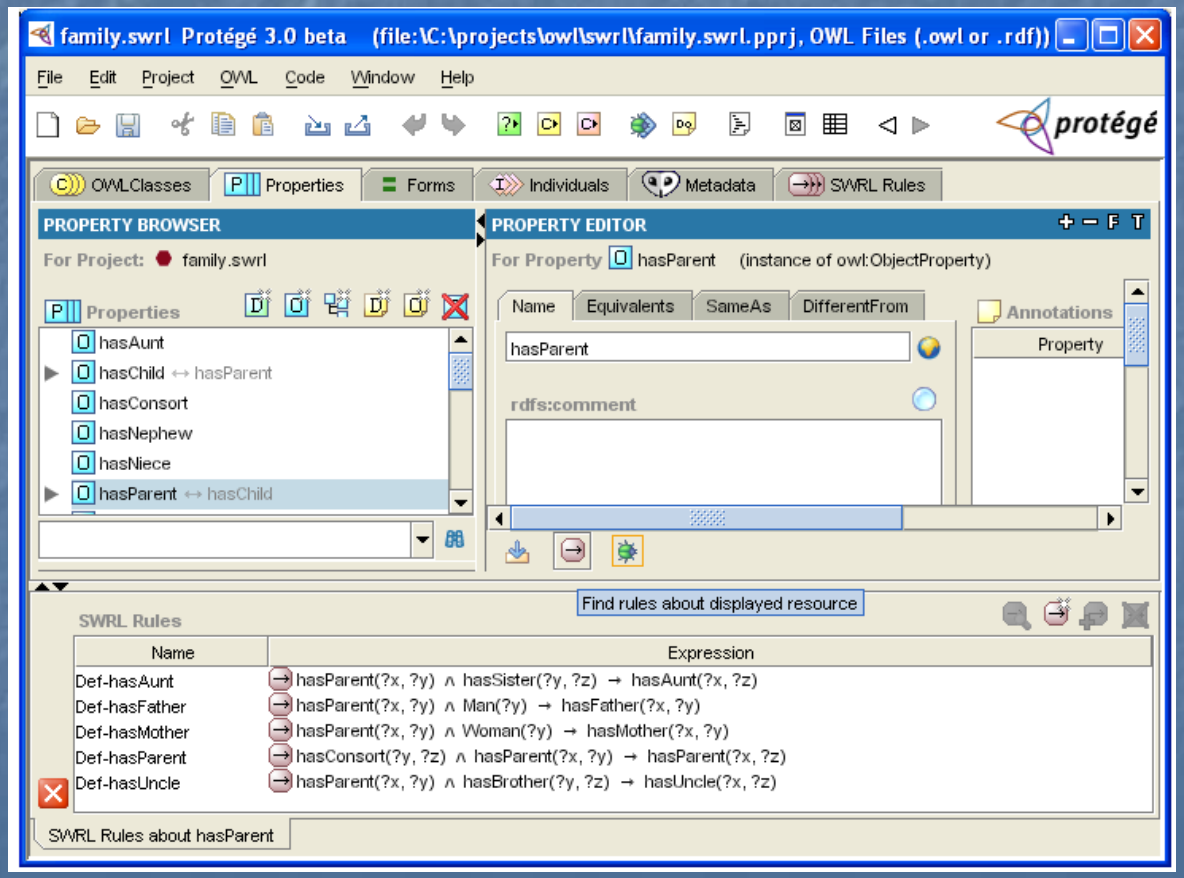
SWRL Editor

Ο SWRL Editor είναι μία επέκταση του Protege-OWL SWRLTab που υποστηρίζει συγγραφή κανόνων SWRL. Μπορεί να χρησιμοποιηθεί για την δημιουργία SWRL κανόνων, να ανοίξει τους υπάρχοντες SWRL κανόνες, να διαβάσει και να γράψει SWRL κανόνες



SWRL Editor

Το επόμενο screenshot παρουσιάζει τον κατάλογο κανόνων που περιέχουν μια αναφορά στη hasParent property.



Μερικοί ακόμη Editors

- VIS RuleVISor
- Protege OWL Plugin, since build 215
- SWeDE

RuleML και SWRL

- Η RuleML είναι σχεδιασμένη να είναι μια XML-based γλώσσα ανταλλαγής για τους κανόνες στο Web. Οι SWRL κανόνες, για παράδειγμα, θα μπορούσαν να χαρτογραφηθούν σε μια ισοδύναμη RuleML αντιπροσώπευση.
- Η ομάδα εργασίας W3C Rule Interchange Format (RIF) ασχολείται τώρα με αυτό το πρόβλημα ανταλλαγής.
- Ο αποκαλούμενος πυρήνας RIF θα είναι πιθανώς αρκετά κοντά σε SWRL από την άποψη της εκφραστικότητάς του.
- Η SWRL είναι OWL-specific, ενώ η RuleML και η RIF στοχεύουν να αντιμετωπίσουν το πολύ ευρύτερο πρόβλημα της ανταλλαγής κανόνων.
- Δεν θα είχε πολύ νόημα, για παράδειγμα, να χρησιμοποιήσουμε την RIF ή την RuleML άμεσα προτιμώντας την SWRL, αλλά η χρήση τους μπορεί να έχει νόημα εάν θελήσουμε να εξάγουμε ή να εισάγουμε σύνολα κανόνα.