ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων

Θέμα Διπλωματικής:

ΑΠΟΤΥΠΩΣΗ ΥΦΙΣΤΑΜΕΝΩΝ ΣΥΣΤΗΜΑΤΩΝ ΟΡΓΑΝΩΣΗΣ ΚΑΙ ΑΠΟΘΗΚΕΥΣΗΣ ΣΗΜΑΣΙΟΛΟΓΙΚΩΝ ΣΧΗΜΑΤΩΝ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΟΥΣ, ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΠΡΟΤΥΠΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΣΥΓΧΩΝΕΥΣΗΣ ΟΝΤΟΛΟΓΙΩΝ.

Επιβλέποντες καθηγητές: Παπαδάκης Νικόλαος

Σταμούλης Γεώργιος

Μελέτη της φοιτήτριας: Ψύρρα Ειρήνης

ΚΑΤΑΛΟΓΟΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1 Μέρος 1: Ο ρόλος των οντολογιών στο σημασιολογικό ιστό και αλγόριθμοι σύμπτυξης σημασιολογικών σχημάτων

Αυτή η εργασία αναφέρεται στο σημασιολογικό ιστό , προσπαθεί να τον ορίσει, να δείξει τη χρησιμότητά του, και να αποσαφηνίσει το πως οργανώνονται οι πληροφορίες σε αυτόν. Πιο συγκεκριμένα αποσκοπεί στο να κατανοηθεί ποια είναι τα πλεονεκτήματα του σημασιολογικού σε σχέση με τον συντακτικό ιστό, ποια είναι η ανάγκη για ύπαρξη του και τα οφέλη του για το φιλτράρισμα των πληροφοριών που είναι πάρα πολλές πλέον, και πως οι οντολογίες μπορούν να βοηθήσουν στην οργάνωση των πληροφοριών που δίνονται στα δεδομένα.

Εκτός από τον ορισμό των οντολογιών στο διαδίκτυο αναφέρονται και διάφοροι τρόποι αναπαράστασης μιας οντολογίας και οι διάφορες γλώσσες στις οποίες μπορεί να γραφεί, και το πώς μπορεί να αντιστοιχιστεί με μια βάση γνώσης. Όμως η δημιουργία και η ανάπτυξη οντολογιών εξελίσσεται και αναπτύσσεται συνεχώς λόγω της απότομης ανάπτυξης του όγκου πληροφοριών στο διαδίκτυο, με αποτέλεσμα όμοια γνώση να αναπαρίσταται από πολλές οντολογίες και με διάφορους τρόπους, και είναι αναγκαία η σύμπτυξη οντολογικών σχημάτων για την καλύτερη αναπαράσταση της γνώσης. Εξηγούνται διάφοροι αλγόριθμοι σύμπτυξης οντολογικών σχημάτων, που χωρίζονται σε δυο κατηγορίες, αυτούς που ελέγχουν κόμβους σε οντολογικά σχήματα και αυτοί που ελέγχουν ονόματα. Έπειτα περιγράφονται κάποιες εφαρμογές των οντολογικών σχημάτων στον σημασιολογικό ιστό και το πως αυτοί λειτουργούν ως αναβαθμισμένα μοντέλα παλιών εφαρμογών λόγω των οντολογιών.

1.1. Ο συντακτικός ιστός (syntactic web)

Ο συντακτικός ιστός ήταν η πρώτη διεπαφή της μορφής επικοινωνίας της παγκόσμιας κοινότητας μέσω του διαδικτύου. Αποτελείται από σελίδες που χρησιμοποιούνται ως βιβλιοθήκες δεδομένων, βάσεις δεδομένων πολυμέσων και εφαρμογών, τη δυνατότητα σύνδεσης με άλλες σελίδες , καθώς και τη διαδραστικότητα του χρήστη με το περιεχόμενο της σελίδας, δηλαδή όλα όσα μπορεί να περιέχει μια απλή html σελίδα. Με μία τυπική web page σήμερα, η markup πληροφορία αποτελείται από πληροφορίες παρουσίασης πχ. μέγεθος γραμματοσειράς και χρώμα και διασυνδέσεις σε σχετικό περιεχόμενο. Το σημασιολογικό περιεχόμενο αυτό είναι προσβάσιμο στους χρήστες, αλλά όχι εύκολα στους υπολογιστές, οι οποίοι απαιτείται τουλάχιστον να κατανοούν τη φυσική γλώσσα, και να κάνουν συγκρίσεις ανάμεσα σε αλφαριθμητικά τα οποία είναι και η μόνη βάση γνώσης τους. Ένα άλλο στοιχείο στο οποίο στηρίζονται είναι η μετάβαση από μια ιστοσελίδα σε άλλη, κάτι που επίσης αναγνωρίζουν ως παράθεση γραμμάτων αλλά το αντιστοιχίζουν σε μια μοναδική και καθολική παρουσίαση των δεδομένων που ο administrator ή κάποιοι χρήστες έχουν αντιστοιχίσει σε αυτή. Είναι επακόλουθα ένας τρόπος στον οποίο ο υπολογιστής παρουσιάζει τα περιεχόμενα και ο χρήστης τα ερμηνεύει. Άρα η πιο δύσκολη διαδικασία επαφίεται στη γνώση και διάθεση του χρήστη αφού η ερμηνεία των περιεχομένων των σελίδων δεν είναι κατανοητή από τη μηχανή. Ως συνέπεια αυτού, η μηχανή δε μπορεί να διαχειριστεί πολλά δεδομένα για να βρει αυτό που πραγματικά αναζητά ένας χρήστης. Με τη ραγδαία αύξηση των πληροφοριών που διατίθενται στο web, τα οποία υπολογίζονται ως 10^18 bytes, η προσωποποίηση (personalisation) της πληροφορίας έχει γίνει αναγκαία για τους διαδικτυακούς οργανισμούς και για τους τελικούς χρήστες. Η ικανότητα ενός site να οδηγεί επιτυχώς τους χρήστες σε χρήσιμες γι' αυτούς πληροφορίες αποτελεί πλέον ένα παράγοντα-κλειδί για την επιτυχία ενός site.

Μια λύση που δίνει χρηστικότητα στους υπολογιστές όσον αφορά τα δεδομένα είναι τα μεταδεδομένα (metadata), δηλαδή δομημένα δεδομένα που εμπεριέχουν πληροφορία για τα δεδομένα και υποστηρίζουν λειτουργίες συσχετιζόμενες με αυτά, οπότε οι μηχανές δεν αναγνωρίζουν μόνο τις λέξεις, αλλά και κάποια σημασία που αυτές έχουν.

1.2. Ο σημασιολογικός ιστός (semantic web)

«Ο Σημασιολογικός Ιστός δεν είναι ένας ξεχωριστός ιστός αλλά η επέκταση του συντακτικού ιστού, στον οποίο η πληροφορία είναι καλά καθορισμένη ώστε να διευκολύνεται πιο αποτελεσματικά η συνεργασία ανθρώπων και υπολογιστών»

Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific

American, May 2001

Το πέρασμα από το syntactic web στο semantic web γίνεται με τη διαδικασία συμπλήρωσης πληροφορίας στα περιεχόμενα του web ώστε να είναι πιο άμεσα προσβάσιμα σε αυτοματοποιημένες διαδικασίες και εφαρμογές. Η πληροφορία αυτή είναι κάποια μεταδεδομένα (metadata) που χαρακτηρίζουν το περιεχόμενο και τη σημασία της πληροφορίας για ένα δικτυακό δεδομένο.

Η χρήση του semantic web μπορεί να χρησιμοποιηθεί για εφαρμογές όπως:

Σημασιολογική αναζήτηση σε μηχανές αναζήτησης

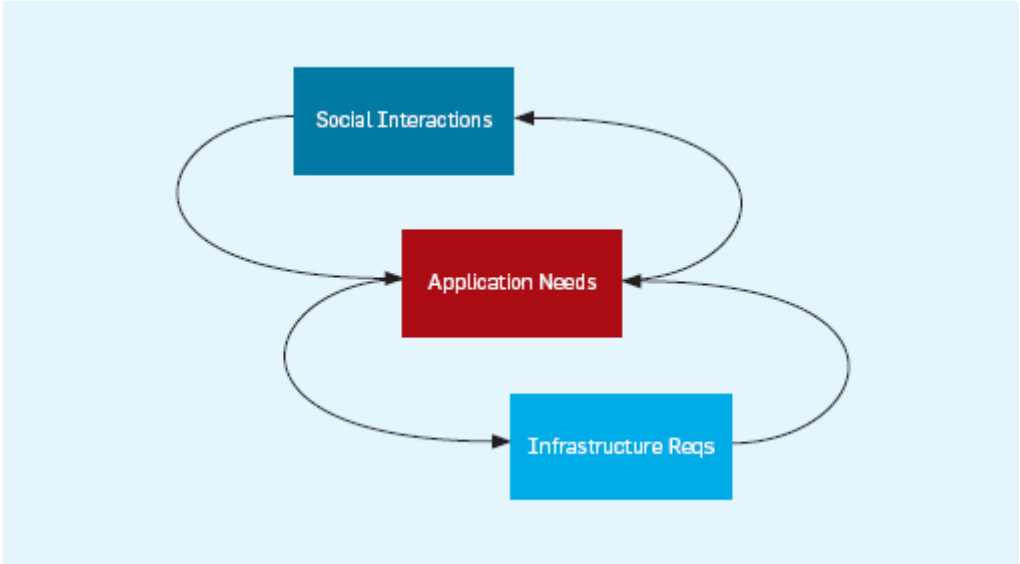
Χρήση πρακτόρων (agents) για εύρεση πληροφοριών

Σημασιολογικά portals

Ο Σημασιολογικός ιστός ευελπιστεί να γίνει ο κύριος μοχλός μετεξέλιξης

του σημερινού παγκόσμιου ιστού όπου η διαδικτυακή πληροφορία θα είναι κατάλληλα οργανωμένη ώστε να μπορούν μηχανές να τη συλλέγουν, να την κατανοούν και να την επεξεργάζονται με αυτόματο και χωρίς επίβλεψη από τον άνθρωπο τρόπο. Στην κατεύθυνση αυτή οι οντολογίες αναμφισβήτητα θα διαδραματίσουν βασικό ρόλο.

Η φιλοσοφία του semantic web στηρίχτηκε στην παραδοχή ότι για καλύτερη επικοινωνία των ανθρώπων και την καλύτερη εξυπηρέτησή τους μέσω του διαδικτύου, πρέπει να δοθεί έμφαση στις εφαρμογές του διαδικτύου και στη βάση γνώσης για να εξαχθούν συμπεράσματα. Μια καλύτερη βάση γνώσης, που θα στηρίζεται σε μεταδεδομένα και όχι σε σύγκριση αλφαριθμητικών, θα κάνει τους πράκτορες αναζήτησης δεδομένων πιο εύχρηστους και πιο αποτελεσματικούς. Το διάγραμμα παρακάτω αποτυπώνει την παραπάνω φιλοσοφία ως αμφίδρομη σχέση των τριών σταδίων αναγκών που πρέπει να βελτιωθούν στο διαδίκτυο.



Σχήμα 1: οι κοινωνικές αλληλεπιδράσεις που γίνονται στο web

Οι κοινωνικές αλληλεπιδράσεις που γίνονται στο web, θέτουν απαιτήσεις στις web εφαρμογές που βρίσκονται κάτω από αυτές, οι οποίες με τη σειρά τους θέτουν αιτήσεις στις υποδομές που βρίσκονται ένα στάδιο πιο κάτω. Έπειτα επιστρέφονται απαντήσεις από το πιο κάτω επίπεδο προς τα παραπάνω.

1.3. Οι Οντολογίες ως κύριο συστατικό του semantic web

Ο όρος **Οντολογία** χρησιμοποιήθηκε για πρώτη φορά στην αρχαία Ελλάδα αναφερόμενη στο *λόγο* περί του όντος ή στην επιστήμη του όντος, τη φιλοσοφική και θρησκειολογική αναζήτηση που εξετάζει τις αρχές της ύπαρξης και συγκρότησης του Όντος. Στην επιστήμη της πληροφορικής χρησιμοποιείται για να ορίσει ένα διαφορετικό τρόπο αναπαράστασης των υποστάσεων των πληροφοριών, να ορίσει δηλαδή πληροφορία για την πληροφορία που υπάρχει στο διαδίκτυο.

Για την περιγραφή ενός αντικειμένου χρειάζεται ένα όνομα και ένα υπόβαθρο γνώσης για αυτό.

Οι οντολογίες είναι μια σαφής περιγραφή ενός αντικειμένου.

“An explicit specification of a conceptualisation” [Gruber93]

Αποτελούνται από ένα συγκεκριμένο λεξιλόγιο που περιγράφει μια πραγματικότητα, και μια σειρά υποθέσεων που γίνονται με σκοπό να κατανοηθεί το νόημα του λεξιλογίου, και αναπαριστούν μια εννοιολογική μορφοποίηση σε μία στερεότυπη μορφή .Ο στόχος είναι να παραχθεί γνώση για ένα αντικείμενο, ώστε να καθίστανται οι οντολογίες ένας τυπικός και επεξεργάσιμος από τις μηχανές ορισμός του αντικειμένου.

Η χρησιμότητα των οντολογιών έγκειται στην οργάνωση των πληροφοριών του διαδικτύου και την εύκολη περιήγηση στον παγκόσμιο ιστό, και κυρίως στην αναζήτηση πολύπλοκων δεδομένων από τις μηχανές αναζήτησης.

Οι οντολογίες αποτελούνται από όρους(κλάσεις και αντικείμενα) και τις σχέσεις μεταξύ τους, τις ιδιότητες και τον αριθμό πληθυκότητας.

Πιο συγκεκριμένα μια οντολογία αποτελείται από ένα σύνολο από έννοιες (concepts), ρόλους (roles), αξιώματα (axioms), και οντότητες (entities ή individuals).

O = (S, A, E), όπου

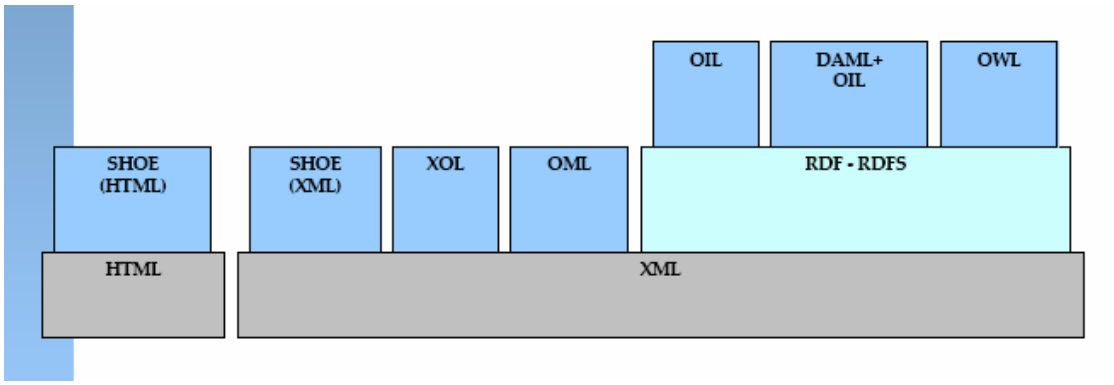
S (Signature): το σύνολο των εννοιών και των ρόλων της οντολογίας, περιγράφουν σύνολα από οντότητες του σύμπαντος

A (Axioms): τα αξιώματα περιορίζουν το νόημα των εννοιών και των ρόλων του S

E (Entities): οι οντότητες κατηγοριοποιούνται από τις έννοιες

Πρακτικά οι όροι που αναπτύσσουν μια οντολογία περιέχουν ορισμούς κλάσεων σε μια οντολογία, ταξινόμηση των κλάσεων ιεραρχικά, ορισμό, περιορισμούς και τιμές για τα χαρακτηριστικά των κλάσεων , και απόδοση τιμών για τα αντικείμενα(instances) των κλάσεων.

Οι γλώσσες στις οποίες μπορεί να περιγραφεί μια οντολογία είναι οι παραδοσιακές γλώσσες, και πιο συγκεκριμένα γλώσσες κατηγορηματικής λογικής όπως prolog, με λογική πλαισίων ή με περιγραφική λογική. Κάποιες από αυτές είναι οι Carin, Flogic, Loom, OCML και Ontolingua. Ο πιο διαδεδομένος τρόπος περιγραφής οντολογιών είναι οι web-bases γλώσσες που η σύνταξή τους βασίζεται στην xml. Τέτοια παραδείγματα είναι οι SHOE, XOL, OML, KML, RDFS, DAML, OIL, η OWL και η OWL 2.



Σχημα 2: οι γλώσσες που περιγράφουν οντολογίες

Η xml δημιουργούσε ένα χαμηλού επιπέδου μοντέλο δεδομένων και δεν μπορούσε να χρησιμοποιηθεί για δημιουργία οντολογιών εξειδικευμένου πεδίου ή οντολογικών λεξιλογίων και να χρησιμοποιήσει βασικές οντολογικές αρχές μοντελοποίησης ούτε ήταν κατάλληλη για διαμοιραζόμενες πηγές στον

παγκόσμιο ιστό καθώς και δε διαθέτει μηχανή συμπερασματολογίας. Όπότε δημιουργήθηκαν οι άλλες γλώσσες για να μπορούν να παρακάμσουν αυτές τις αδυναμίες.

Έπειτα η rdf προοριζόταν για την αναπαράσταση μεταδεδομένων. Συστήνεται από τη W3C, μπορεί να περιγράψει ένα τυποποιημένο μοντέλο δεδομένων και τυποποιημένο XMLs συντακτικό (+namespaces), υπάρχουν parsers και Api's, και κάνει για την δημιουργία εκτεταμένων λεξιλογίων. Όμως εισάγει και κάποιους περιορισμούς όπως το γεγονός ότι είναι πολύ αδύναμη στη σημασιολογική αναπαράσταση, δεν περιγράφει καλά το νόημα της πληροφορίας και δεν διαθέτει μηχανή συμπερασματολογίας.

Η DAML+OIL δημιουργήθηκε από την US Defense Advanced

Research project agency (DARPA) σε συνεργασία με την ομάδα της ευρωπαϊκής ένωσης για τις agent markup γλώσσες. Έχουν δημιουργηθεί πολλές οντολογίες με την DAML+OIL. Αυτή η γλώσσα βασίζεται στην RDF Schema και θεωρείται πρόγονος της OWL (standard web ontology language).

Η OWL είναι Ιδανική όταν πρέπει οι πληροφορίες που περιλαμβάνονται σε έγγραφα να επεξεργαστούν από εφαρμογές, σε αντιδιαστολή με τις περιπτώσεις όπου το περιεχόμενο πρέπει να παρουσιαστεί μόνο σε ανθρώπους. Παρέχει περισσότερες ευκολίες για νοηματική και σημασιολογική έκφραση από τις XML, RDF και RDF(S)

και, συνεπώς, η OWL υπερβαίνει αυτές τις γλώσσες χρησιμοποιώντας τη δυνατότητά της να αναπαριστά το αναγνώσιμο από μηχανή περιεχόμενο στον Ιστό. Όπως αναφέρθηκε και προηγουμένως η owl χρησιμοποιεί τη μεθοδολογία της xml για τη συγγραφή της, όμως μπορεί να συγγραφεί και με uml σχήματα.

Τέλος υπάρχουν τρεις εκδόσεις της γλώσσας για κάθε χρήση:

η OWL LITE για συγγραφείς που δε χρειάζονται μεγάλη γκάμα εντολών για να περιγράψουν τις οντολογίες τους ή νέους συγγραφείς

η OWL DL που υποστηρίζουν εντολές με τις οποίες ένας συγγραφέας μπορεί να περιγράψει κλάσεις που εκδίδουν αποτέλεσμα σε πεπερασμένο χρόνο σε ότι αφορά τις λογικές σχέσεις μεταξύ δεδομένων

η OWL FULL για έμπειρους συγγραφείς οι οποίοι έχουν περισσότερες και χωρίς όρια περιγραφικές ανάγκες, αυτή η γλώσσα όμως δεν εγγυάται αποτίμηση αποτελέσματος σε πεπερασμένο χρόνο!

Ενώ οι πιο δημοφιλείς περιγραφικές εντολές τις γλώσσα φαίνονται στον παρακάτω

πίνακα:

<table>
<tr>
<td>RDF Schema Features:
Class (Thing, Nothing)
rdfs:subClassOf
rdf:Property
rdfs:subPropertyOf
rdfs:domain
rdfs:range
Individual</td>
<td>(In)Equality:
equivalentClass
equivalentProperty
sameAs
differentFrom
AllDifferent
distinctMembers</td>
<td>Property Characteristics:
ObjectProperty
DatatypeProperty
inverseOf
TransitiveProperty
SymmetricProperty
FunctionalProperty
InverseFunctionalProperty</td>
</tr>
<tr>
<td>Property Restrictions:
Restriction
onProperty
allValuesFrom
someValuesFrom</td>
<td>Restricted Cardinality:
minCardinality (only 0 or 1)
maxCardinality (only 0 or 1)
cardinality (only 0 or 1)</td>
<td>Header Information:
Ontology
imports</td>
</tr>
<tr>
<td>Class Intersection:
intersectionOf</td>
<td rowspan="2">Versioning:
versionInfo
priorVersion
backwardCompatibleWith
incompatibleWith
DeprecatedClass
DeprecatedProperty</td>
<td rowspan="2">Annotation Properties:
rdfs:label
rdfs:comment
rdfs:seeAlso
rdfs:isDefinedBy
AnnotationProperty
OntologyProperty</td>
</tr>
<tr>
<td>Datatypes
xsd datatypes</td>
</tr>
</table>

Σχημα 3: οι εντολές της γλώσσας owl

Τέλος, η owl 2 είναι μια μετεξέλιξη της owl που δημοσιεύτηκε τον Οκτώβρη του 2009. Έχει τα πλεονεκτήματα ότι μπορεί να διαχειριστεί δεδομένα τα οποία περιγράφονται με γλώσσα xsd (xml schema definition language) και ότι μπορεί να διαχωρίσει την απλή δήλωση (abstract) μιας οντότητας από την περιγραφή της, η οποία μπορεί να γίνει με δυο τρόπους, τον συναρτησιακό (direct semantics) και το γραφικό όπως με rdf (rdf based semantics).

1.4. Οντολογίες αντί των βάσεων δεδομένων

Στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένα. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει, βάση του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Αποτελείται από ένα σύνολο από πίνακες στους οποίους περιγράφονται αντικείμενα και οι μεταξύ τους σχέσεις. Οι βάσεις δεδομένων πρέπει να δέχονται μεγάλες ποσότητες δεδομένων αλλά και να είναι εύκολα προσβάσιμες και απόλυτα ασφαλείς.

Μία οντολογία περιέχει συντακτικά και σημασιολογικά πλουσιότερη

πληροφορία από τις βάσεις δεδομένων. Η πληροφορία που περιγράφεται από μία οντολογία περιέχει ημιδομημένο κείμενο σε φυσική γλώσσα και όχι πληροφορία ενός σχεσιακού πίνακα, και αυτό δίνει τη δυνατότητα στο συγγραφέα της οντολογίας να προσθέσει σχόλια, περιορισμούς για τις κλάσεις, για ορίσει έναν ποιο ιδιαίτερο τύπο δεδομένων από τους κλασσικούς που χρησιμοποιούν οι βάσεις δεδομένων ή ακριβή αριθμό τιμών, να ορίσει αντίθετες κλάσεις, και πληθώρα άλλων δυνατοτήτων. Μία οντολογία πρέπει να είναι δικτυακής αρχιτεκτονικής και να συνδέεται με άλλες οντολογίες διαδικτυακά, γιατί χρησιμοποιείται για το διαμοιρασμό και τη μετάδοση της πληροφορίας σε όλους τους χρήστες.

1.5. Τα οντολογικά σχήματα

Τα οντολογικά σχήματα περιγράφουν οντολογίες και εισάγουν επιπλέον μια πληροφορία, αυτή της σχέσης μεταξύ των δεδομένων (classification).Η περιγραφή των οντολογιών προγραμματιστικά γίνεται με γλώσσες RDF (Resource Description Framework), και εμπεριέχει XML συντακτικό και σημασιολογία, χρησιμοποιείται για την αναπαράσταση δεδομένων και περιγράφει τη σημασία των δεδομένων με τρόπο αναγνωρίσιμο από τις μηχανές.

Το RDFShema εμπλουτίζει την RDF με «σχηματικά χαρακτηριστικά, όπως κλάσεις και ιδιότητες, τύπους, υποκλάσεις και ιδιότητες τους, το πεδίο ορισμού και την έκτασή τους (range).

<AcademicStaff>Kotis</AcademicStaff>

<professor> Vouros</professor>

<course name="Knowledge Repres.">

<isTaughtBy>K. Stergiou</isTaughtBy>

</course>

Οι άνθρωποι καταλαβαίνουν ότι το μάθημα διδάσκεται και από τους τρεις ανθρώπους, αλλά οι μηχανές όχι. Αυτό το πρόβλημα λύνεται με τα RDFShemas.

"All Professors are Academic Staff"

<rdfs: Class rdf: about="Professor">

<rdfs:subClassOf rdf:resource="#AcademicStaff"/>

</rdfs: Class>

"All courses are taught by Academic Staff"

<rdf: Property rdf: ID="taughtBy">

<rdfs: domain rdf: resource="course"/>

<rdfs: range rdf: resource="AcademicStaff"/>

</rdf: Property>

Παρακάτω περιγράφεται σχηματικά ο παραπάνω κώδικας και η επέκταση που δίνει η χρήση του rdfs.

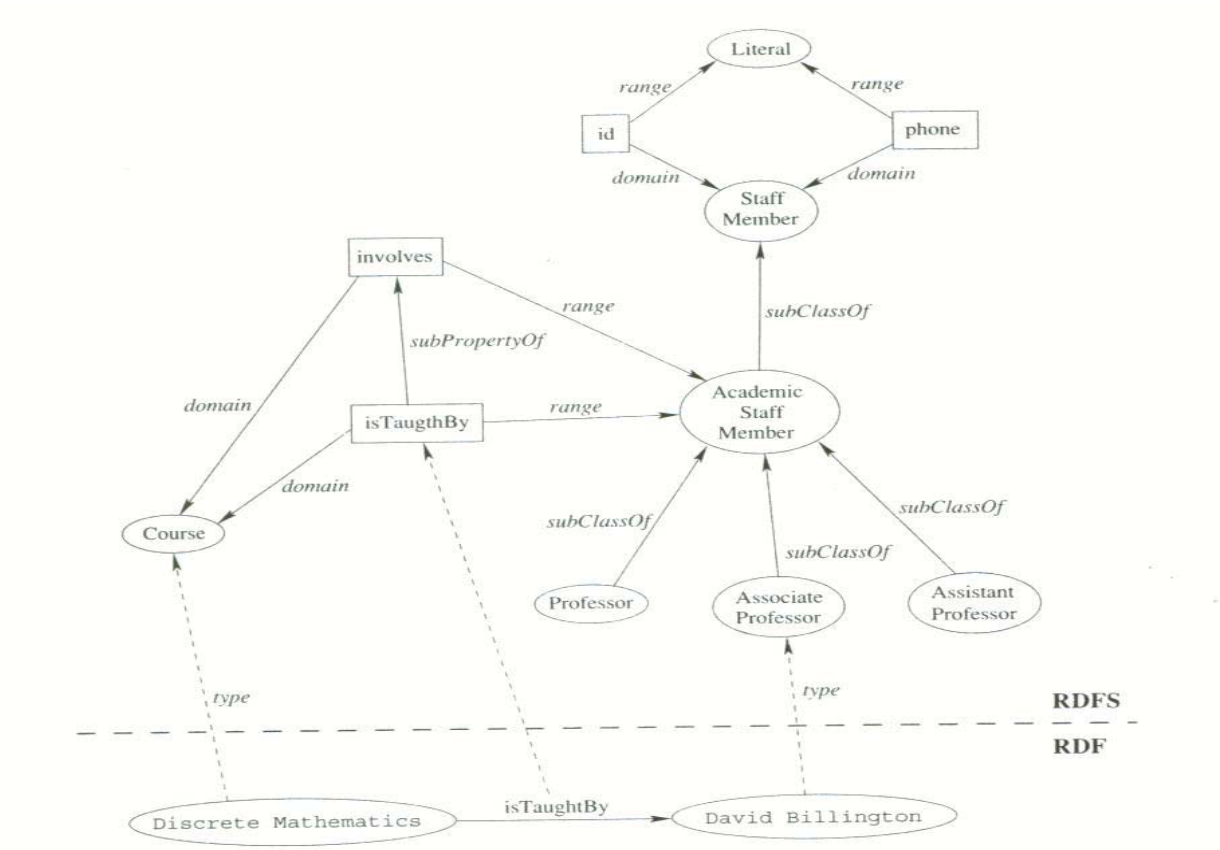


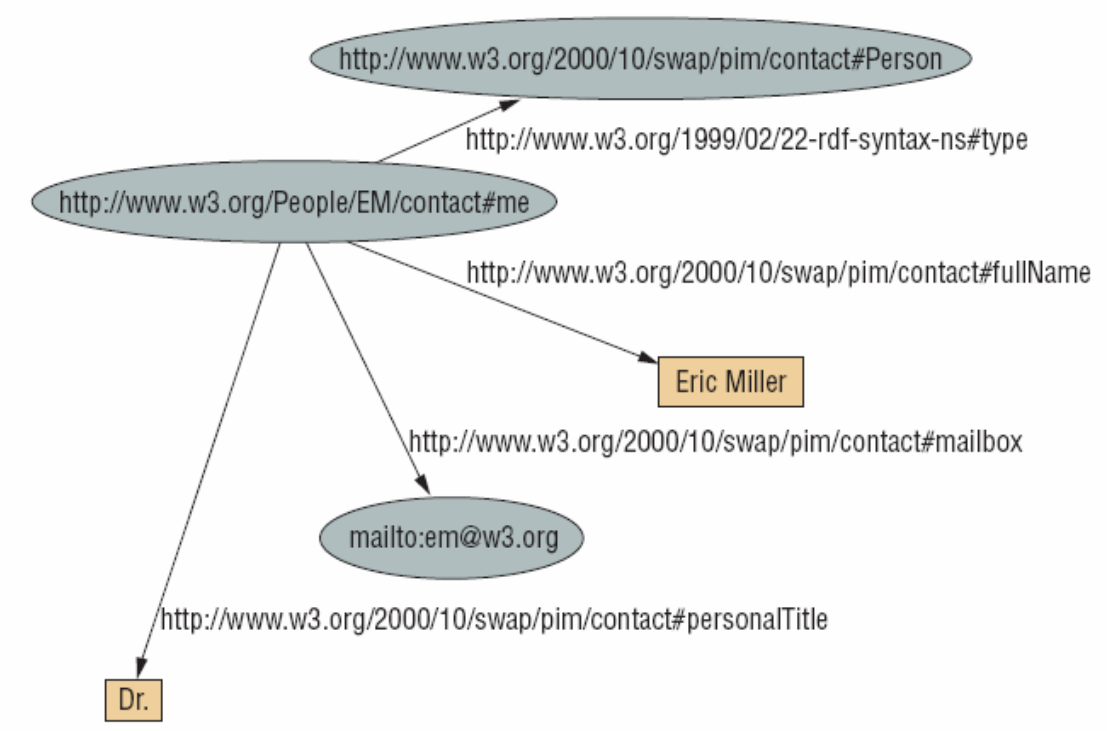
Figure 3.6 RDF and RDFS layers

Σχημα 4: ένα παράδειγμα rdfs

Ένα άλλο παράδειγμα που περιγράφει το γεγονός «πληροφορίες για τον Eric Miller»:

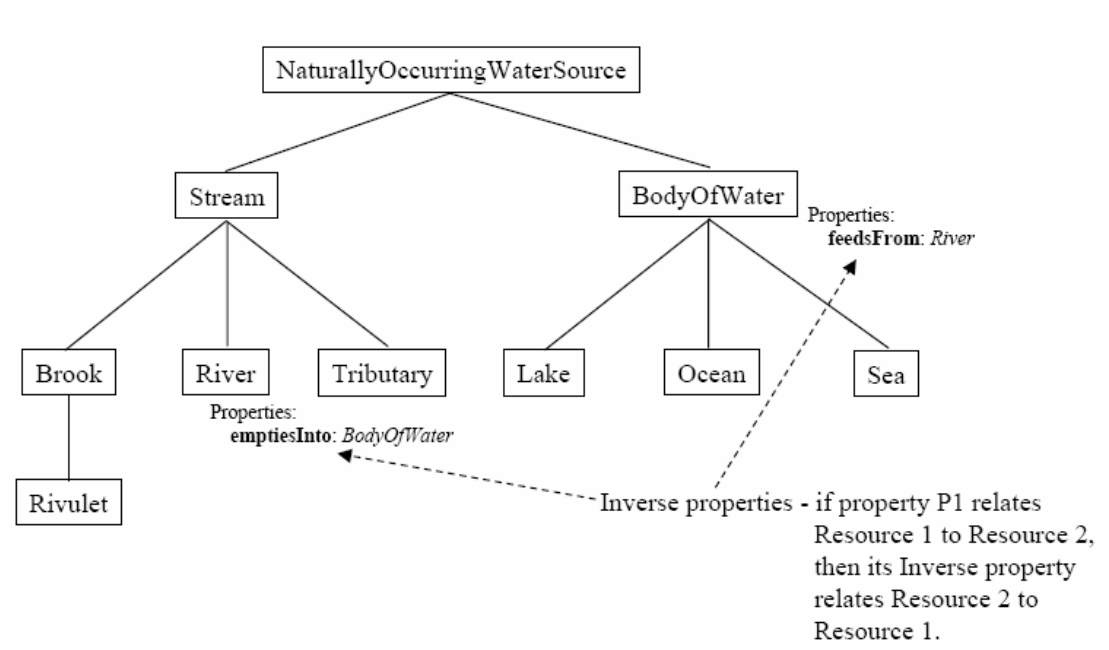
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

   <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
      <contact:fullName>Eric Miller</contact:fullName>
      <contact:mailbox rdf:resource="mailto:em@w3.org"/>
      <contact:personalTitle>Dr.</contact:personalTitle>
   </contact:Person>

</rdf:RDF>
```

Σχήμα 5: παράδειγμα στο οποίο φαίνεται ο κώδικας rdf

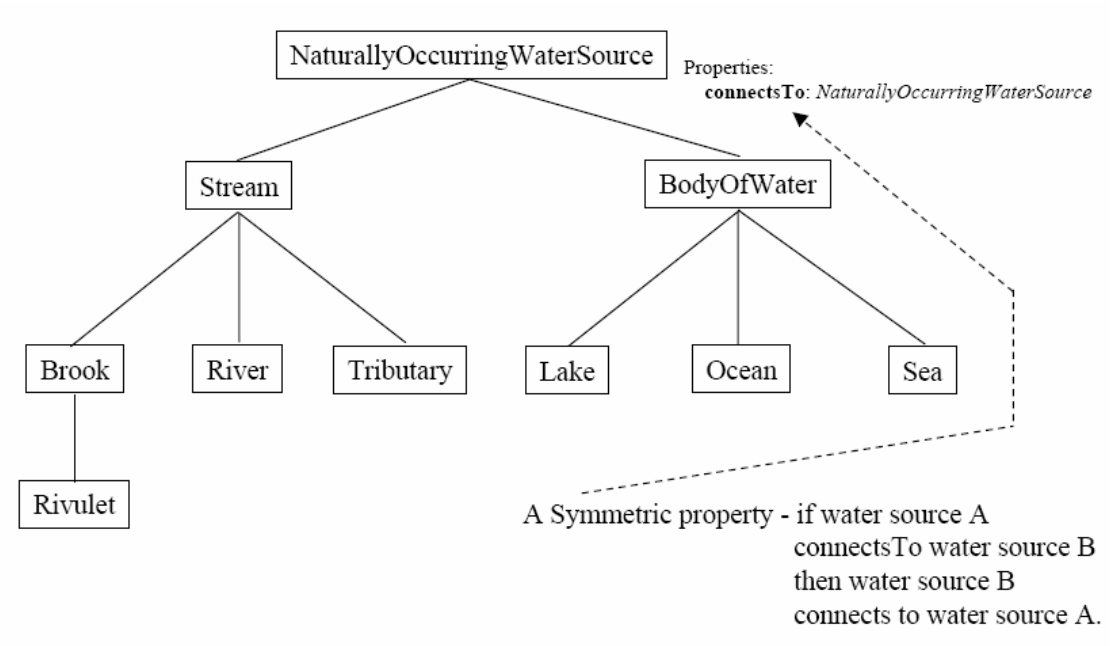


Σχήμα 6: ο γράφος που περιγράφει τον παραπάνω κώδικα

Επίσης στα οντολογικά σχήματα είναι εμφανής η σχέση και οι ιδιότητες που μπορεί να έχει μια οντολογία, όπως στο παρακάτω παράδειγμα:



Σχήμα 7: παράδειγμα rdf γράφου στο οποίο φαίνονται αντίστροφες ιδιότητες



Σχήμα 8: παράδειγμα rdf γράφου στο οποίο φαίνονται συνδεόμενες ιδιότητες

Η περιγραφή μιας οντολογίας με rdfs όμως περιορίζει το συγγραφέα. υπάρχουν περιγραφικά σχήματα που δε μπορούν να αποτυπωθούν, όπως για παράδειγμα:

Δε μπορεί να οριστεί το εύρος και ο τύπος κάποιων κλάσεων.

Πχ δε μπορεί να περιγραφεί ότι το αντικείμενο του «άνθρωπος» hasChild είναι επίσης άνθρωπος, ενώ για τον ελέφαντα είναι επίσης ελέφαντας.

Δε μπορεί να οριστεί η ακρίβεια και κάποιοι περιορισμοί στα αντικείμενα

Πχ δε μπορεί να περιγραφεί ότι ένας άνθρωπος έχει γονείς, 2 ακριβώς στον αριθμό, και ότι είναι και αυτοί άνθρωποι.

Δε μπορούν να οριστούν μεταβατικές αντίθετες ή συμμετρικές ιδιότητες.

Πχ δεν υπάρχει σχέση που να περιγράφει ότι η isPartOf είναι μεταβατική ιδιότητα, ότι η hasPart is είναι αντίθετη με την isPartOf ή ότι αυτές οι δύο είναι συμμετρικές.

Μια γλώσσα που έχει αυτές τις δυνατότητες, επεκτείνοντας τα rdfs είναι η owl, κάποια χαρακτηριστικά της οποίας έχουν αναφερθεί προηγουμένως. (Αυτή η γλώσσα θα χρησιμοποιηθεί στα αρχεία που δέχεται ως είσοδο το πρόγραμμα αυτής της εργασίας.)

1.6. Ποια η ανάγκη για σύμπτυξη οντολογικών σχημάτων

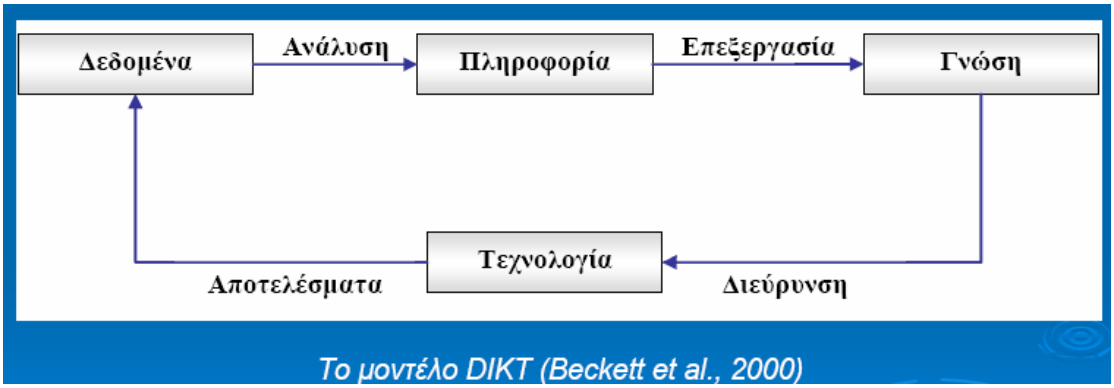
Οι πληροφορίες στο διαδίκτυο αυξάνονται συνεχώς και με εκθετικούς ρυθμούς, με αποτέλεσμα να αυξάνεται και η γνώση που χρειάζεται ο υπολογιστής να κατανοήσει και άρα και οι οντολογίες που την περιγράφουν. Ο σκοπός είναι όταν εμφανίζεται καινούρια γνώση, να μπορούν οι χρήστες να τη διαχειριστούν και να την ενσωματώσουν στην ήδη υπάρχουσα. Έτσι η βάση γνώσης μεγαλώνει και είναι εύκολα διαχειρίσιμη. Άρα εμφανίζεται η ανάγκη να οργανωθούν οι καινούριες οντολογίες και να ενσωματωθούν στις ήδη υπάρχουσες για να επεκτείνουν τη γνώση που θα χρησιμοποιείται ως δεδομένο σε μελλοντική χρήση της.

Σχήμα 9: το μοντέλο διαχείρισης γνώσης

Το κόστος δημιουργίας οντολογιών μόνο από τους ανθρώπους που διαχειρίζονται τη γνώση στο διαδίκτυο είναι πολύ μεγάλο, με επακόλουθο να

δίνεται η δυνατότητα στους απλούς χρήστες να αναπτύξουν την οντολογική γνώση, όπως και τη γνώση στο διαδίκτυο (web2). Όμως σε ένα κατανεμημένο σύστημα, όπως το internet, ο κάθε δημιουργός μπορεί να δώσει το δικό του ορισμό για μια ετικέτα, ανάλογα με τη δική του γνώση και την ομάδα χρηστών στην οποία ανήκει, οπότε να υπάρχουν παραπάνω του ενός ορισμού για το ίδιο πεδίο ενδιαφέροντος, με διαφορετικά χαρακτηριστικά και κάποια από αυτά ίσως να είναι λάθος. Στο επόμενο στάδιο, για την αναζήτηση δεδομένων στις οντολογίες χρησιμοποιούνται προγράμματα με την ιδιότητα των «λογικών πρακτόρων», που δε λειτουργούν ντετερμινιστικά αλλά αυτόνομα, οπότε οι πολλαπλοί ορισμοί μιας ετικέτας επιφέρουν κάθε φορά διαφορετικά αποτελέσματα. Επομένως δημιουργείται η ανάγκη για συνεχή παρακολούθηση, έλεγχο και διαχείριση αυτών των αλλαγών και επιπλέον η ανάγκη για συγχώνευση οντολογιών και επαναχρησιμοποίηση τους. Αυτή η ανάγκη μεγαλώνει από την αποδοχή ότι οι οντολογίες χρησιμοποιούνται από το διαδίκτυο και κάθε αρχείο έχει μια αναφορά σε οντολογικό σχήμα που ανήκει στο διαδίκτυο, οπότε μπορεί να υπάρξει σύγκρουση κατά την αποθήκευση οντολογιών, όπως αν τύχει να έχουν το ίδιο όνομα. Από τη μεριά τους οι δημιουργοί πρέπει να είναι σε συχνή συνεργασία ώστε να αποσαφηνιστούν οι έννοιες στην επικοινωνία μεταξύ τους.

Το σημαντικότερο πρόβλημα πέραν της «κοινής σημασίας » στα δεδομένα είναι ότι όλοι οι δημιουργοί δεν είναι εξειδικευμένοι στην ανάπτυξη σχημάτων οντολογιών, ή χρησιμοποιούν διαφορετικές τεχνικές δημιουργίας και απόδοσης νοήματος στις οντολογίες. Ένα άλλο θέμα είναι ότι κάποιες οντολογίες δημιουργούνται από πλέον ειδικούς και η υλοποίησή τους περιέχει πολλά μεταδεδομένα και πολλούς κόμβους αντικειμένων, ενώ από την άλλη μπορεί να δημιουργηθεί μια οντολογία από ανθρώπους που έχουν μια πιο γενική εικόνα για το ίδιο θέμα με λιγότερους κόμβους. Αν κάποιος αναζητήσει το συγκεκριμένο θέμα, πρέπει να πάρει πληροφορίες και από τις δυο οντολογίες, άρα είναι αναγκαία η σωστή σύμπτυξη τους για πιο σωστά αποτελέσματα . Συνεπώς πρέπει να δίνεται η δυνατότητα ανάπτυξης σχημάτων οντολογιών με φυσικό τρόπο (εύκολη και περιγραφική γλώσσα όπως η xml), η δυνατότητα αξιολόγησης των οντολογιών που ενέχει όμως και τη λάθος απόδοση αξίας από χρήστες χωρίς αρκετή γνώση, και η δυνατότητα συνένωσης δυο διαφορετικών οντολογιών για να εμπλουτίζεται η διαθέσιμη γνώση.



Σχήμα 10: το μοντέλο DIKT

Ακόμη στην ένωση της γνώσης που προσφέρουν τα δεδομένα οδηγεί και η μεγάλη ετερογένεια των δεδομένων που υπάρχουν στο διαδίκτυο. Αυτή μπορεί να αφορά τα υπολογιστικά συστήματα, την ετετογένεια στη σύνταξη, στη σημασιολογία και η ετερογένεια των χρηστών. Αυτές οι ετερογένειες προκαλούν προβλήματα στην πρόσβαση στη γνώση, είτε από την ομάδα χρηστών, είτε λόγω των περιορισμών που δίνει μια γλώσσα περιγραφής οντολογιών, είτε λόγω των περιορισμών στην επεξεργασία που ενέχει ένα υπολογιστικό σύστημα. Η συνένωση δυο οντολογικών σχημάτων, μεγαλώνει τη γνώση αλλά το τελικό μέγεθος είναι μικρότερο από το άθροισμα των αρχικών, κάποιοι μπορούν να χρησιμοποιήσουν τη βάση γνώσης που κάποιος άλλος με καλύτερο υπολογιστικό σύστημα μπορεί να δημιουργήσει, καθώς και οι ειδικοί μπορούν να μεταφέρουν μια οντολογία από μια γλώσσα σε άλλες ώστε κάποιος χρήστης που ξέρει μόνο μία να είναι σε θέση να βρει όποια γνώση χρειάζεται.

Ως αποτέλεσμα των παραπάνω είναι εμφανής η ανάγκη οργάνωσης των οντολογιών, με απώτερο σκοπό να γίνουν ένα εργαλείο «καθολικής αποδοχής» , που να χρησιμοποιείται από όλα τα προγράμματα-πράκτορες, από αναγνώστες, ή προγράμματα που θέλουν να το χρησιμοποιήσουν για την αναζήτηση αποτελεσμάτων, και αν κάποιος θέλει να βάλει νέα πληροφορία αυτή πρέπει να ενσωματώνεται μετά από κρίση στην ήδη καθολική υπάρχουσα.

1.7. Ποιοι αλγόριθμοι χρησιμοποιούνται για τη σύμπτυξη οντολογικών σχημάτων

Οι αλγόριθμοι που χρησιμοποιούνται για τη σύμπτυξη οντολογικών σχημάτων βασίζονται σε δυο ιδέες:

Σύμπτυξη με βάση το όνομα(ή ετικέτα), στην οποία τα δεδομένα οργανώνονται σε κλάσεις ανάλογα με το όνομά τους, και οι αλγόριθμοι που ακολουθούν αυτή την ιδέα συγκρίνουν τα ονόματα αυτά για να ενώσουν τυχόν όμοια στοιχεία, και

Σύμπτυξη με βάση τον κόμβο, στην οποία τα δεδομένα ενώνονται σε ένα κόμβο, δεδομένου ενός ονόματος και μιας συγκεκριμένης θέσης στο οντολογικό σχήμα.

Για να περιγραφεί η σχέση μεταξύ των ονομάτων και των κόμβων των οντολογικών σχημάτων χρησιμοποιούνται οι παρακάτω λογικές πράξεις:

ισότητα (=)

αντίθεση (¬)

υποσύνολο ή υπερσύνολο(≤,≥) και

όχι ταίριασμα (<>)

ενώ αν δε βρεθεί κάποια σχέση μεταξύ των δεδομένων θεωρούνται ανεξάρτητα.

Συνήθως οι πιο πολλοί αλγόριθμοι προσεγγίζουν και τις δυο παραπάνω τακτικές , οι οποίες συνοψίζονται στον παρακάτω αλγόριθμο.

Ο παρακάτω αλγόριθμος δέχεται ως είσοδο δυο οντολογικά σχήματα και υπολογίζει ως έξοδο μια σειρά σχέσεων μεταξύ των δεδομένων, είναι δε ο σκελετός πάνω στον οποίο είναι χτισμένοι όλοι οι αλγόριθμοι που συμπτύσσουν οντολογίες :

1: για όλες τις ετικέτες(labels) L στα δυο δέντρα, υπολόγισε τις έννοιες των ετικετών

2:για όλους τους κόμβους στα δυο δέντρα υπολόγισε αν υπαρχει ταίριασμα μεταξύ των κόμβων

3: για όλα τα ζευγάρια ονομάτων (labels) L στα δυο δέντρα, υπολόγισε τις σχέσεις των ονομάτων

4:για όλα τα ζευγάρια κόμβων στα δυο δέντρα υπολόγισε τις σχέσεις μεταξύ των κόμβων

1.8. Οι 4 φάσεις απόφασης του ταιριάσματος

Οι πιο πολλοί αλγόριθμοι απόφασης ταιριάσματος δυο οντολογιών ακολουθούν τις τέσσερις παρακάτω φάσεις. Η πρώτη αποτελεί την είσοδο και η τελευταία την έξοδο των αλγορίθμων, ενώ οι δύο μεσαίες αποτελούν τον τρόπο εξαγωγής απόφασης.

Απόκτηση χαρτογράφησης οντολογίας. Σε αυτήν την φάση, αποκτάμε τις χαρτογραφήσεις που παράγονται με τη χρησιμοποίηση ενός συστήματος χαρτογράφησης οντολογιών. Το σύστημα ταιριάσματος μπορεί να στηριχθεί σε συντακτικές, δομικές ή ακόμα και σημασιολογικές τεχνικές ταιριάσματος.

Συγκεχυμένη ερμηνεία των χαρτογραφήσεων. Σε αυτήν την φάση, οι επίκτητες χαρτογραφήσεις ερμηνεύονται ως συγκεχυμένοι ισχυρισμοί.

Συγκεχυμένος συλλογισμός πέρα από τις χαρτογραφήσεις. Σε αυτή τη φάση, η οντολογία που λαμβάνεται από τον εμπλουτισμό της δεύτερης οντολογίας με τη χαρτογράφηση με συγκεχυμένους μεμονωμένους ισχυρισμούς που παράγονται με τη βοήθεια των χαρτογραφήσεων, ελέγχεται για τη συνέπεια της με τη βοήθεια ενός συγκεχυμένου συστήματος συλλογισμού.

Επικύρωση και αναθεώρηση χαρτογράφησης. Σε αυτήν την φάση, οι χαρτογραφήσεις είναι αναθεωρημένες συμφώνα με τα αποτελέσματα συλλογισμού. Οι χαρτογραφήσεις που προκαλούν τις ασυνέπειες μέσα στη νέα οντολογία διώχνονται και δίνεται νέο βάρος.

1.9. Τεχνικές σύμπτυξης οντολογικών σχημάτων

Παραπάνω διατυπώθηκε η διαδικασία που ακολουθείται για να αποφασιστεί αν δυο οντολογίες ταιριάζουν η όχι. Αυτή η απόφαση παίρνεται μετά από τη χρήση διάφορων τεχνικών που ελέγχουν τις ετικέτες και τους κόμβους και ψάχνουν για ομοιότητες.

Παρακάτω θα παρουσιαστούν διάφορα ήδη τεχνικών σύμπτυξης οντολογικών σχημάτων, που διαφέρουν μεταξύ τους στην είσοδο (κόμβος ή ετικέτα) καθώς και στον τρόπο που επεξεργάζονται αυτά.

1.9.1 Τεχνικές σύμπτυξης των ετικετών (string and language based)

1.9.1.1 Τεχνικές σύμπτυξης των ετικετών με βάση τους χαρακτήρες των λέξεων

1.9.1.1.1 Prefix

Δέχεται ως είσοδο δυο αλφαριθμητικά και ελέγχει αν υπάρχει μερικό ταίριασμα στην αρχή των λέξεων πχ net=network

1.9.1.1.2 Suffix

Δέχεται ως είσοδο δυο αλφαριθμητικά και ελέγχει αν υπάρχει μερικό ταίριασμα από το τέλος προς την αρχή των λέξεων πχ. Pword=password

1.9.1.1.3 Edit distance

Δέχεται ως είσοδο δυο αλφαριθμητικά και υπολογίζει τον αριθμό των πράξεων χαρακτήρων(ένθεση, διαγραφή και αντικατάσταση) που πρέπει να γίνουν, ώστε τα αλφαριθμητικά να είναι ίδια, πχ edit_distance(nkn, nikon)=0,4. Δημοφιλείς αλγόριθμοι που υπολογίζουν την απόσταση είναι οι S-Match, OLA, και Ankor-Prompt. πάντα όμως υπάρχει ο κίνδυνος δυο λέξεις να μοιάζουν αλλά το νόημά τους να είναι εντελώς διαφορετικό, κι αυτό λύνεται μόνο με τη χρήση άλλων οντολογιών ή με την παρέμβαση του χρήστη.

1.9.1.1.4 Tokenization

Τα ονόματα των κόμβων μπαίνουν σε tokens τα οποία αναγνωρίζουν τα χαρακτηριστικά μια γλώσσας ,πχ η ν«History and Philosophy of Science» γίνεται «history, and, philosophy, of, science»Επίσης είναι αναγνωρίσιμα και τα μόρια ένωσης τομης και άρνησης. Για παράδειγμα η φράση Earth and Atmospheric Sciences γίνεται «earth sciences, and, atmospheric, sciences».

1.9.1.1.5 Lemmatization

Τα παραπάνω tokens διαμορφώνονται κατάλληλα ώστε να ταιριάζουν με τις «βασικές » λέξεις, για παράδειγμα η λέξη sciences γίνεται science, ώστε η σύγκριση μεταξύ λέξεων να γίνεται πιο εύκολα.

1.9.1.1.6 Gloss based: WordNet gloss comparison

Ο αριθμός των ίδιων λέξεων που βρίσκονται σε δυο ίδιες εισόδους μεγαλώνει την τιμή της ομοιότητας σχέση ισότητας επιστρέφεται αν το αποτέλεσμα της τιμής ομοιότητας ξεπερνά ένα δεδομένο άνω όριο. Πχ «το τάβλι είναι ένα παιχνίδι που παίζεται με δύο παίκτες και κάποιος είναι νικητής» «το τένις είναι ένα άθλημα, με δύο παίκτες, και κάποιος είναι νικητής».

1.9.1.1.7 δημιουργία μοναδικών consepts

Αναγνωρίζονται όλα τα δεδομένα που «ανήκουν» σε κάποιο ήδη δημιουργημένο token με συγκεκριμένο λήμμα.

1.9.1.1.8 δημιουργία σύνθετων consepts

Αναγνωρίζονται όλα τα μόρια και οι λέξεις που έχουν λογική σημασία, και επομένως από ατομικά consepts και αυτή τη σημασία γίνονται σύνθετα consepts.Για παράδειγμα οι λέξεις εκτός και χωρίς γίνοται άρνηση, οι μέσα και του γίνονται λογική τομή, το κόμμα και οι ενώσεις γίνοται λογική τομή. Άρα για παράδειγμα η ετικέτα History and Philosophy of Science υπολογίζεται ως CHistory and Philosophy of Science = (CHistory CPhilosophy) CScience, όπου CScience = science, {sensesWN#2}

1.9.1.2 Τεχνικές που βασίζονται σε περιορισμούς

Είναι αυτές που ελέγχουν τα χαρακτηριστικά των κλάσεων και των ιδιοτήτων μιας οντολογίας, όπως τον τύπο δεδομένων, τα κλειδιά και τις μεταβλητές, και προσπαθούν να ταιριάξουν δυο οντολογικά σχήματα με βάση αυτά. Πιο συγκεκριμένα στοχεύουν σε δυο συγκρίσεις:

1.σύγκριση των τύπων δεδομένων

Συγκρίνει τους τύπους δεδομένων των χαρακτηριστικών δυο κλάσεων και το πεδίο τιμών τους, και καλείται να αποφασίσει αν ένας τύπος ή μια τιμή «μοιάζει» με κάποιον άλλον-η, αν δεν ταιριάζει απόλυτα. Πχ ο τύπος «έτος» ταιριάζει περισσότερο με τον τύπο «χρονολογία» παρά με τον τύπο «integer».

2.σύγκριση πολλαπλών τιμών των χαρακτηριστικών

Συγκρίνει τις τιμές και τη συμβατότητα των τύπων δυο χαρακτηριστικών που συλλέγει από τις κλάσεις. Πχ η τιμή «6-12 μήνες» είναι πιο κοντά στην τιμή «1 χρόνος» από ότι στην τιμή «6-12 πόλεις», γιατί παρόλο που οι τιμές είναι ίδιες οι τύποι δεν είναι συμβατοί.

Οι γλωσσικοί και λεξικογραφικοί πόροι ως κοινή γνώση χρησιμοποιούνται με σκοπό να ταιριάξουν οι λέξεις, βασισμένες στο κοινό νόημα που έχουν(συνώνυμα ή αντώνυμα)

1.κοινή βάση γνώσης.

Χρησιμοποιείται κοινό λεξικό για να βρεθεί η κοινή σημασία των λέξεων. Πχ το WorldNet είναι μια ηλεκτρονική βάση δεδομένων που χρησιμοποιείται ως αγγλικό λεξικό (και άλλων γλωσσών), όπου δίνονται τα νοήματα όλων των λέξεων μαζί με τα συνώνυμά τους.

2.οργανωμένη γνώση ειδικών ορισμών.

Μια ειδική βάση δεδομένων εμπλουτίζεται με γνώση ειδικών ορισμών, που δεν είναι διαθέσιμη στις κοινές βάσεις γνώσης, με τη μορφή ετικέτας με

συνώνυμα, καταλήξεις, και άλλες σχέσεις. Πχ οι ετικέτες «μπαμπάς», «μπαμπούλης», και «πατέρας» αν περάσουν μέσα από μια τέτοια βάση δεδομένων θα φανεί ότι ταιριάζουν.

1.9.1.3 Τεχνικές επαναχρησιμοποίησης διάταξης

Οι τεχνικές επαναχρησιμοποίησης διάταξης των αντικειμένων αναπαριστούν έναν εναλλακτικό τρόπο εύρεσης εξωτερικών πηγών, που περιέχουν διατάξεις από παλιότερα, ήδη ταιριασμένα, οντολογικά σχήματα. Πχ αν σε παλιότερα μελετημένο σχήμα που αναπαριστά ένα γενεαλογικό δέντρο, η ετικέτα «πατέρας» είναι συνώνυμη με την ετικέτα «μπαμπάς», ανάγεται το συμπέρασμα ότι στο προς μελέτη δέντρο αν η ετικέτα «γονιός» είναι ανεξάρτητη, μπορούμε να την αντικαταστήσουμε με την ετικέτα «μπαμπάς» και να ξαναψάξουμε για τυχούσα σχέση.

1.9.1.4 Τεχνική χρήσης οντολογιών υψηλότερου επιπέδου

Οι ήδη υπάρχουσες οντολογίες υψηλότερου επιπέδου επίσης μπορούν να χρησιμοποιηθούν ως εξωτερική υπάρχουσα κοινή γνώση, που μπορεί να δώσει συμπεράσματα για κάποιες αμφιλεγόμενες κοινές οντολογίες. Για παράδειγμα μια οντολογία που περιγράφει συνώνυμα μπορεί αν χρησιμοποιηθεί να ταιριάξουν λέξεις που δε θα ταίριαζαν επειδή δε μοιάζουν τα γραμματά του ή το ύψος τους στα σχήματα. Τέτοιες οντολογίες είναι οι Suggested Upper Merged Ontology(SUMO) και Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE). Το χαρακτηριστικό τέτοιων οντολογιών είναι ότι αποτελούν λογικά συστήματα που χρησιμοποιούνται ως διερμηνείς για άλλες οντολογίες. Παρά την προφανή αναγκαιότητα τέτοιων χρήσεων αυτή η τεχνική δεν είναι πολύ διαδεδομένη προς το παρόν ίσως επειδή είναι δύσκολο να δημιουργηθούν πολλές τόσο μεγάλες οντολογικές βάσεις γνώσης, γιατί πρέπει να γίνονται από ειδικούς στη γλώσσα ανθρώπους. Παρακάτω αναφέρεται μια οντολογία το wordnet στην οποία ο δημιουργός της καθηγητής ψυχολογίας George A. Miller προσπάθησε να περιγράψει σε αυτή όλες τις αγγλικές λέξεις και τα συνώνυμά τους.

1.9.2 Τεχνικές σύμπτυξης των κόμβων (structure-graph based)

Αυτές οι τεχνικές εξετάζουν ένα οντολογικό σχήμα από τη γραφική του σκοπιά, και παρατηρούν σ ένα κόμβο τη θέση του στο γράφημα και τις σχέσεις με του με τους άλλους κόμβους, με σκοπό να βγάλουν συμπεράσματα για το πόσο ταιριάζουν δυο κόμβοι που ανήκουν σε διαφορετικά οντολογικά σχήματα.

Υπάρχουν πολλοί αλγόριθμοι που δίνουν λύση στο πρόβλημα ταιριάσματος δυο γραφημάτων, οι οποίοι έχουν μεγάλο χρονικό κόστος, και συνήθως χρησιμοποιούν μεθόδους βελτιστοποίησης για τον ορισμό του προβλήματος, όπως για παράδειγμα την εύρεση ταιριάσματος του γραφήματος εκεί που ελαχιστοποιείται η ανομοιότητα δυο κόμβων.

Κάποια άλλα ταιριάσματα αφορούν τα παιδιά ενός κόμβου, τα φύλλα και τις σχέσεις μεταξύ κόμβων.

Α. Η ομοιότητα δυο εσωτερικών κόμβων ενός γραφήματος υπολογίζεται βασιζόμενη στο αν τα (άμεσα) παιδιά τους είναι όμοια λεξικογραφικά.

Β. Η ομοιότητα δυο εσωτερικών κόμβων ενός γραφήματος υπολογίζεται βασιζόμενη στα φύλλα του γραφήματος. Δύο κόμβοι λοιπόν, είναι όμοιοι αν τα φύλλα τους είναι όμοια λεξικογραφικά, ακόμη και αν τα παιδιά τους(στο ακριβώς κάτω επίπεδο) δεν είναι όμοια.

Γ. Η ομοιότητα δυο εσωτερικών κόμβων ενός γραφήματος υπολογίζεται βασιζόμενη στις σχέσεις τους. Αν δυο κόμβοι σχετίζονται με έναν ίδιο, τότε οι δυο σχέσεις είναι όμοιες, ενώ αν δυο κόμβοι σχετίζονται με ίδιες σχέσεις με άλλους δυο κόμβους, τότε αυτοί είναι όμοιοι.

Πχ.«Εργάτες ->(εργάζονται)-> 8ώρες» και «εργαζόμενοι->(δουλεύουν)->8 ώρες» συνεπάγεται

ότι εργάζονται = δουλεύουν

και ότι εργάτες = εργαζόμενοι

Τεχνικές ταξινόμησης για σύμπτυξη των κόμβων (taxonomy based)

Οι τεχνικές ταξινόμησης είναι επίσης αλγόριθμοι γράφων που ελέγχουν μόνο τη σχέση εξειδίκευσης. Η διαίσθηση πίσω από τις τεχνικές ταξινόμησης είναι ότι δεδομένων κάποιων όμοιων κόμβων που συνδέονται με κάποιες σχέσεις με άλλους, τότε και οι γειτονικοί κόμβοι είναι πιθανό να είναι όμοιοι μεταξύ τους. Αυτή η λογική μπορεί να εκφραστεί με διάφορους τρόπους.

Α. ταίριασμα συγκεκριμένων μονοπατιών

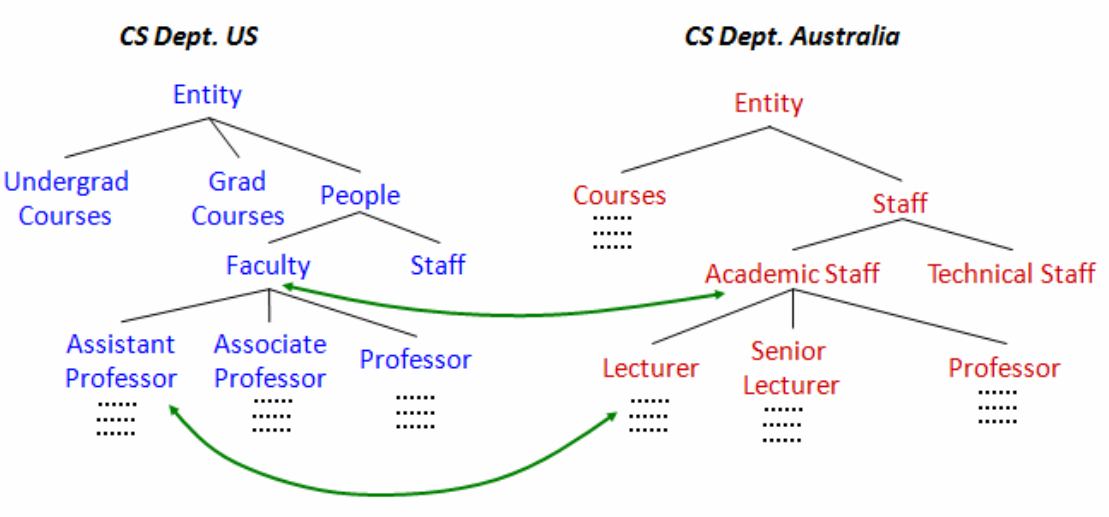
Δεδομένων δυο μονοπατιών με ακμές ανάμεσα σε κλάσεις ορισμένες με ιεραρχικές σχέσεις, συγκρίνει τους κόμβους και τη θέση τους σ αυτά τα μονοπάτια, και αναγνωρίζει όμοιους κόμβους.

Β. Κανόνες των υπέρ/υπό συνόλων

Το ταίριασμα βασίζεται σε κανόνες που καλύπτονται από την παραπάνω λογική. Πιο συγκεκριμένα, αν δυο κόμβοι υπερκλάσης δυο οντολογιών είναι όμοιοι, τότε είναι όμοιοι και οι κόμβοι των κλάσεων, και αν οι κόμβοι δυο κλάσεων είναι όμοιοι τότε είναι όμοιοι και οι αντίστοιχοι κόμβοι των υποκλάσεων, κοκ.

Αποθήκευση γραφημάτων με τα χαρακτηριστικά τους

Με αυτή την τεχνική , τα γραφήματα αποθηκεύονται μαζί και τα χαρακτηριστικά τους, όπως ο αριθμός των κόμβων, το ύψος κα. Οπότε όταν δυο σχήματα ελέγχονται αν ταιριάζουν , γίνεται αρχικά έλεγχος των χαρακτηριστικών τους και αν αυτά δεν ταιριάζουν, τότε δεν ταιριάζουν και τα σχήματα. Οπότε εξάγεται πιο γρήγορα το συμπέρασμα για το αν είναι δυνατόν να ταιριάζουν ή όχι, σε αντίθεση με την αναζήτηση κόμβο προς κόμβο.



Σχήμα 11: παράδειγμα πιθανού ταιριάσματος των κόμβων δύο γράφων

Τεχνικές σύμπτυξης κόμβων βασισμένες σε μοντέλα αλγορίθμων

Αυτοί οι αλγόριθμοι περιορίζουν την είσοδο(κόμβους) στην ερμηνεία του νοήματός της και έτσι βγάζουν συμπεράσματα. Τέτοιοι αλγόριθμοι είναι οι propositional satisfiability (SAT) και description logics(DL).

A. propositional satisfiability

Με αυτό τον αλγόριθμο, το πρόβλημα ταιριάσματος δυο κόμβων ανάγεται σε πρόβλημα ταιριάσματος ενός συνόλου κόμβων. όλοι οι κόμβοι που πιθανώς ταιριάζουν μπαίνουν σε clauses , πχ (node 1.1, node2.1) , και έπειτα επιστρέφονται οι σχέσεις τους, δηλαδή αν ταιριάζουν ή όχι. Αυτοί οι αλγόριθμοι κάνουν εξαντλητική αναζήτηση, και έπειτα αποφαίνονται για το αν δυο σχήματα ταιριάζουν. Η πολυπλοκότητά του όμως γίνεται πολύ μεγάλη αν σε ένα clause υπάρχουν παραπάνω από δυο κόμβοι (SAT-3) γιατί το πρόβλημα γίνεται NPComplete.

B. DL τεχνικές

Οι τεχνικές sat δε μπορούν να υπολογίσουν ιδιότητες η σκοπούς των κόμβων. στην περιγραφική λογική (DL) οι σχέσεις (=, !=, <,>) εκφράζονται με μια συνάρτηση ένταξης (subsumption). Στην πραγματικότητα, στην αρχή γίνεται αναζήτηση ταιριάσματος δυο οντολογιών, μετά τη μετονομασία, και μετά εξετάζεται κάθε ζευγάρι κόμβων και σκοπών μήπως βρεθεί κοινή ερμηνεία. Πχ μια οντολογία περιέχει τις κλάσεις εταιρία, εργαζόμενος και μικρό-εταιρία

ως εταιρία με το πολύ 5 εργαζόμενους, και μια άλλη περιέχει τις κλάσεις βιομηχανία , εργάτης και βιοτεχνία ως μια βιομηχανία με το πολύ 10 εργαζόμενους. Αν είναι γνωστό ότι εργαζόμενος= εργάτης, τότε ανάγεται το συμπέρασμα ότι εταιρία= βιομηχανία και μικρό-εταιρία= βιοτεχνία.

Πιο τυπικά, μια αναπαράσταση (mapping) M η οποία περιέχει τις δυο έννοιες των δυο οντολογιών που συγκρίνονται και τις σχέσεις μεταξύ τους είναι η

mi=(Ci, C`I,ni,Ri)

Ci,C`i είναι οι δυο συγκρινόμενες έννοιες των δυο οντολογιών O και O'.

Ni είναι η τιμή που δείχνει το σθένος της σχέσης, και είναι κομμάτι του διανύσματος (D,≤,0,1), όπου, d E D: 0≤d≤1.

Ri είναι η σχέση μεταξύ των δυο εννοιών Ci, C`i , που καθορίζεται με ένα από τα παρακάτω σύμβολα R={≤ ≥ ≡}.

Table 1. Fuzzy DL Descriptions and Axioms

Abstract Syntax	DL Syntax	Semantics
Bottom	$\bot$	$\bot^{\mathcal{I}}(a) = 0$
Top	$\top$	$\top^{\mathcal{I}}(a) = 1$
Intersection	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}}(a) = t(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
Union	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}}(a) = u(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
Complement	$\neg C$	$(\neg C)^{I}(a) = c(C^{\mathcal{I}}(a))$
Existential Restriction	$\exists R.C$	$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{I}}} t(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))$
Universal Restriction	$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in \Delta^{\mathcal{I}}} J(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))$
Min Cardinality Restriction	$\geq nR$	$(\geq nR)^{\mathcal{I}}(a) = \sup_{b_1,\ldots,b_p \in \Delta^{\mathcal{I}}} t(\mathop{\mathrm{t}}_{i=1}^{p} R^{\mathcal{I}}(a,b_i), \mathop{\mathrm{t}}_{i<j} \{b_i \neq b_j\})$
Max Cardinality Restriction	$\leq nR$	$(\leq nR)^{\mathcal{I}}(a) = \inf_{b_1,\ldots,b_{p+1} \in \Delta^{\mathcal{I}}} \mathcal{J}(\mathop{\mathrm{t}}_{i=1}^{p+1} R^{\mathcal{I}}(a,b_i), \mathop{\mathrm{u}}_{i<j} \{b_i = b_j\})$
SubClass	$C \sqsubseteq D$	$C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$
Equivalent Classes	$C \equiv D$	$C^{\mathcal{I}}(a) = D\mathcal{I}(a)$
SubRole	$R \sqsubseteq S$	$R^{\mathcal{I}}(a,b) \leq S^{\mathcal{I}}(a,b)$
Class Individual	$o : C \triangleright\triangleleft n$	$C^{\mathcal{I}}(o^{\mathcal{I}}) \triangleright\triangleleft n$
Role Individual	$(o,o') : R \triangleright\triangleleft n$	$R^{\mathcal{I}}(o^{\mathcal{I}}, o'^{\mathcal{I}}) \triangleright\triangleleft n$
Disjoint Classes	$C \sqsubseteq \neg D$	$C^{\mathcal{I}}(a) \leq 1 - D^{\mathcal{I}}(a)$
Transitive Object Property	$\mathsf{Trans}(R)$	$\sup_{b \in \Delta^{\mathcal{I}}} t(R^{\mathcal{I}}(a,b), R^{\mathcal{I}}(b,c)) \leq R^{\mathcal{I}}(a,c)$

Σχήμα 12: πίνακας που δείχνει την περιγραφική λογική και τους κανόνες της

Η γλώσσα owl έχει σχεδιαστεί έτσι ώστε να μπορεί να ανταποκριθεί στην περιγραφική λογική και να αναπαραστήσει λογικές σχέσεις ανάμεσα σε κλάσεις ή αντικείμενα. Στον παρακάτω πίνακα φαίνονται (αριστερά) κάποιες ιδιότητες που ορίζονται από την owl και ποια είναι η αντιστοιχία τους με τις πράξεις της περιγραφικής λογικής.

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	¬Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	∀hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	⩽1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	⩾2hasChild	$\exists^{\geqslant n} y.P(x,y)$

Σχήμα 13: πίνακας με ιδιότητες της owl που αντιστοιχίζονται με κανόνες περιγραφικής λογικής που φαίνονται παρακάτω

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human ⊑ Animal ⊓ Biped
equivalentClass	$C_1 \equiv C_2$	Man ≡ Human ⊓ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male ⊑ ¬Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} ≡ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} ⊑ ¬{peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter ⊑ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost ≡ price
inverseOf	$P_1 \equiv P_2^-$	hasChild ≡ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ ⊑ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1P$	⊤ ⊑ ⩽1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1P^-$	⊤ ⊑ ⩽1hasSSN$^-$

Σχήμα 14: αντιστοιχία ιδιοτήτων της γλώσσας owl και η αντιστοιχία τους με τους κανόνες περιγραφικής λογικής

1.10. Το web2 και το ταίριασμα οντολογιών

Μια προσέγγιση που προτείνεται σε αρκετές δημοσιεύσεις χρησιμοποιεί ένα μοντέλο ενός web2 διαδικτυακού εργαλείου το οποίο οι χρήστες μπορούν να χρησιμοποιούν σε τρία στάδια:

Manager οντολογιών και συνόλων δεδομένων. Το μοντέλο δίνει το δικαίωμα στο χρήστη να δημοσιεύσει τη δική του οντολογία είτε δίνοντας ένα url, είτε ανεβάζοντας στο server του μοντέλου το δικό του αρχείο που περιγράφει την οντολογία είτε να διαβάζεται το αρχείο από το χώρο του χρήστη με το SPARQL πρωτόκολλο. Επίσης δίνει το δικαίωμα στο χρήστη να δημοσιεύσει την οντολογία του με συγκεκριμένα δικαιώματα, όπως να τη δημοσιεύσει μόνο για ανάγνωση ή και να μπορούν να την αλλάξουν οι άλλοι χρήστες.

Περιβάλλον ταιριάσματος της οντολογίας. Σε αυτή τη φάση το μοντέλο αφού διαβάσει την οντολογία του χρήστη μπορεί μόνο να προτείνει τις ομοιότητες που βρίσκει με άλλες οντολογίες και να επιδείξει το πώς ο χρήστης (ή οι άλλοι χρήστες) μπορεί να αλλάξει ή να εμπλουτίσει την οντολογία του με υποκλάσεις ή υπερκλάσεις ή περιορισμούς. Σε αυτά τα αποτελέσματα καταλήγει χρησιμοποιώντας πολλές από τις τεχνικές που περιγράφηκαν παραπάνω και συγκρίνοντας την οντολογία με τις ήδη υπάρχουσες στη συλλογή δεδομένων του manager. Έτσι εναπόκειται στην κρίση του εκάστοτε χρήστη αν τα αποτελέσματα του μοντέλου-εργαλείου είναι ασφαλή και ποια από όλα θα χρησιμοποιήσει.

Κοινωνικό περιβάλλον αλληλεπίδρασης. Σε αυτό το στάδιο το μοντέλο δίνει το δικαίωμα στους χρήστες να συζητήσουν πάνω σε μια οντολογία και στην ορθότητα των αποτελεσμάτων του εργαλείου, και χωρίζεται σε τρία στάδια. Το στάδιο ontology view έχει μια ενισχυμένη μηχανή αναζήτησης που βρίσκει άλλες οντολογίες που ταιριάζουν με την επιλεγμένη κοινή οντολογία. Οι επαυξήσεις της οντολογίας που είναι δραστηριότητες χρηστών και έχουν επιπτώσεις στις κοινές έννοιες, απεικονίζονται τις πρόσθετες πληροφορίες. Επιπλέον, η διεπαφή επιτρέπει να ελεγχθεί το σύνολο ετικετών που χρησιμοποιούνται για τις ισοδύναμες έννοιες. Όταν ο χρήστης επιλέγει μια έννοια που συνδέει μερικές χαρτογραφήσεις χρηστών με αυτή, μπορεί να μεταπηδήσει στο στάδιο mapping view το οποίο επιδεικνύει πληροφορίες για τις τοπικές χαρτογραφήσεις για τη συγκεκριμένη έννοια. Επίσης υπάρχει και στο forum view στο οποίο ο κάθε χρήστης μπορεί να πει τη γνώμη του πάνω στη σύγκριση δυο οντολογιών και /ή να ψηφίσει για το ταίριασμα τους.

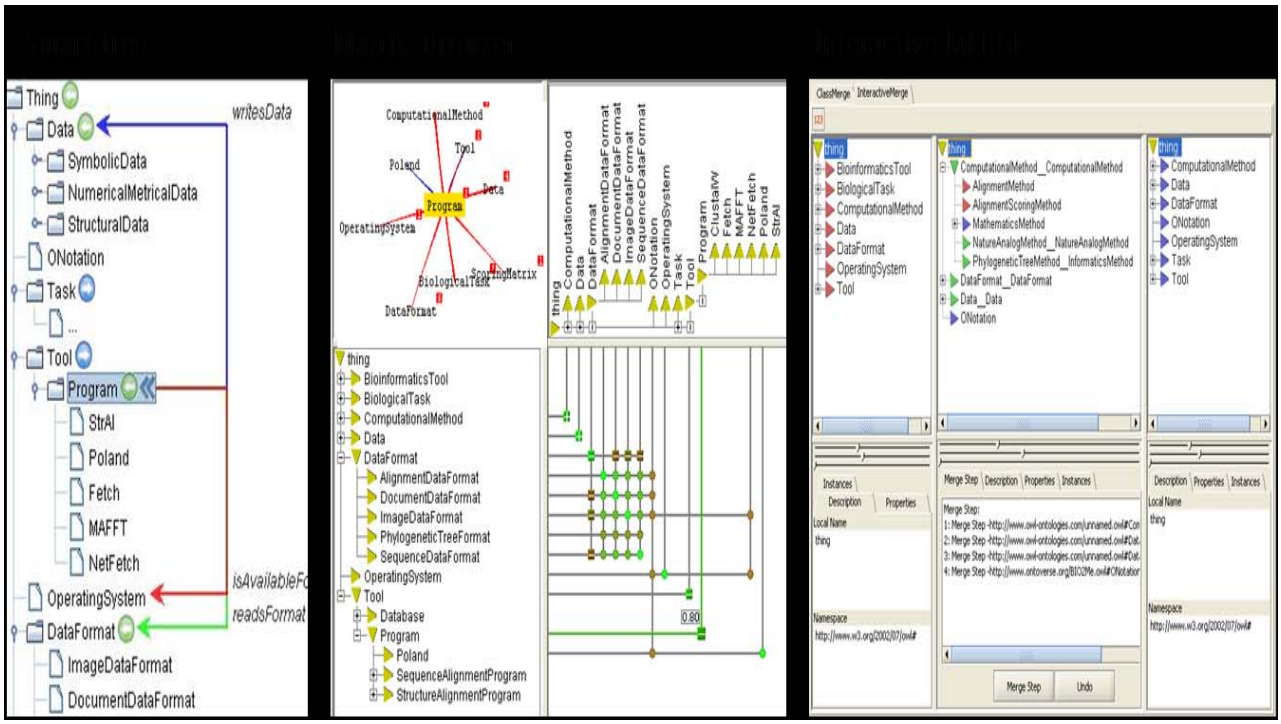
Μια τέτοια χρήση του web2 και ενός τέτοιου μοντέλου κρίνεται αρκετά αποδοτική γιατί έχει το πλεονέκτημα ότι το ταίριασμα και η αλλαγή μιας οντολογίας δεν επαφίεται στη μηχανή αναζήτησης ταιριαστών οντολογιών

αλλά στην κρίση των χρηστών του ιστοχώρου, και άρα οι απαιτήσεις των χρηστών δεν καλύπτονται αποκλειστικά από το λογισμικό αλλά και οι ίδιοι έχουν τον τελευταίο λόγο.

Τα πλεονεκτήματα αυτής της άποψης είναι:

- Υποστήριξη εξερεύνησης μιας οντολογίας και χειρωνακτική δημιουργία των χαρτογραφήσεων
- Παροχή μιας οπτικής αντιπροσώπευσης της πηγής και των στόχων μιας οντολογίας
- Παροχή μιας μεθόδου για το χρήστη ώστε να δεχτεί/απορρίψει μια προτεινόμενη χαρτογράφηση
- Παροχή πρόσβασης σε ορισμούς οντολογιών
- Παρουσίαση ενός όρου που δείχνει πότε ένας χρήστης επιθεωρεί μια προτεινόμενη χαρτογράφηση
- Παροχή διαλογικής πρόσβασης στις πηγές και τους στόχους των οντολογιών
- Υποστήριξη της διαλογικής πλοήγησης και προνόμιο στο χρήστη να δεχτεί/απορρίψει τις χαρτογραφήσεις
- Παροχή ανάδρασης

Κάτι παρόμοιο φαίνεται στην παρακάτω εικόνα, και δείχνει τις τρεις φάσεις της διαδικτυακής ανάπτυξης και διαχείρισης οντολογιών.



Σχήμα 15: παράδειγμα των τριών φάσεων διαδικτυακής ανάπτυξης και διαχείρισης οντολογιών

Το πιο μεγάλο μειονέκτημα όμως τέτοιων εφαρμογών είναι ότι χρειάζονται την απόλυτη επίβλεψη από τον άνθρωπο, κάτι που αντιβαίνει στον αρχικό στόχο, δηλαδή την απαλλαγή του ανθρώπου από το ψάξιμο της πληροφορίας. Ο μελλοντικός στόχος είναι αυτές οι εφαρμογές να προγραμματιστούν ώστε να ενεργούν αυτόνομα, και να κάνουν τις ενώσεις οι πράκτορες-προγράμματα (μια δουλειά που τώρα κάνουν οι άνθρωποι).

1.11. Η απόδοση των αλγορίθμων

Οι διάφορες τεχνικές και αλγόριθμοι είναι σχετικά απαιτητικοί σε πολυπλοκότητα, δεδομένου ότι οι περισσότεροι χρησιμοποιούν την εξαντλητική αναζήτηση και πολλά μοντέλα οντολογιών είναι αρκετά εκτενή (για παράδειγμα τα μοντέλα που περιγράφουν μια πρωτεΐνη η οποία περιέχει χιλιάδες συνδυασμούς αμινοξέων), αλλά δε θα αναπτυχθεί περεταίρω το συγκεκριμένο ζήτημα, γιατί πιο σημαντικό για αυτό το είδος αλγορίθμων είναι να είναι αποδοτικοί από λειτουργικής άποψης.

Το πιο σημαντικό στην απόδοση των αλγορίθμων και των τεχνικών εντοπίζεται στο να καταφέρουν να συνδυάζουν οντολογίες διάφορων χρηστών σε ένα κατανεμημένο σύστημα με αυτόνομο τρόπο (πράκτορες) ώστε να ενώνουν να βρίσκουν σχέσεις σε αντικείμενα και περιορισμούς των τύπων και των τιμών στα οντολογικά σχήματα. Τις περισσότερες φορές μία μόνο τεχνική καλύπτει μικρό ποσοστό των οντολογικών σχημάτων, για αυτό το λόγο προτείνεται η χρήση περισσότερων τεχνικών και σχετικής αυτονομίας του προγράμματος στις αποφάσεις για πιο σωστά αποτελέσματα. Επίσης η αυτονομία των προγραμμάτων εισάγει πολλές περιπτώσεις λάθους ακόμη κι αν χρησιμοποιείται οντολογία που περιέχει βάση δεδομένων για όλες τις λέξεις και τα συνώνυμα τους, πράγμα ουτοπικό, οπότε χρειάζεται και στα περισσότερα υπάρχοντα προγράμματα χρησιμοποιείται η λήψη απόφασης από τον άνθρωπο, για την ερμηνεία των αποτελεσμάτων των τεχνικών.

Η διαλειτουργικότητα είναι μια ισχυρή απαίτηση στα ανοικτά κατανεμημένα συστήματα και στο σημασιολογικό Ιστό. Η ανάγκη για την ολοκλήρωση οντολογιών δεν ικανοποιείται πάντα από τη διαθέσιμη τεχνολογία ταιριάσματος οντολογιών επειδή, στις περισσότερες περιπτώσεις, η σημασιολογία των συγκρινόμενων οντολογιών δεν εξετάζεται, οδηγώντας κατά συνέπεια σε ασυμβίβαστες χαρτογραφήσεις. Οι πιθανολογικές προσεγγίσεις έχουν προταθεί για να επικυρώσουν τις χαρτογραφήσεις και να λύσουν τις ασυμβατότητες. Άρα προτείνεται μια προσέγγιση επικύρωσης βασισμένη στη συγκεχυμένη ερμηνεία των χαρτογραφήσεων και η απόδοση μιας πιθανότητας ταιριάσματος δυο κλάσεων, η οποία μοντελοποιεί καλύτερα την έννοια του βαθμού ομοιότητας μεταξύ των στοιχείων μιας οντολογίας, αλλά πάλι ακόμη κ αν η πιθανότητα ένωσης είναι μεγάλη, το πρόγραμμα δεν εγγυάται ότι δεν υπάρχει περίπτωση λάθους.

1.12. Εφαρμογές σημασιολογικού ιστού

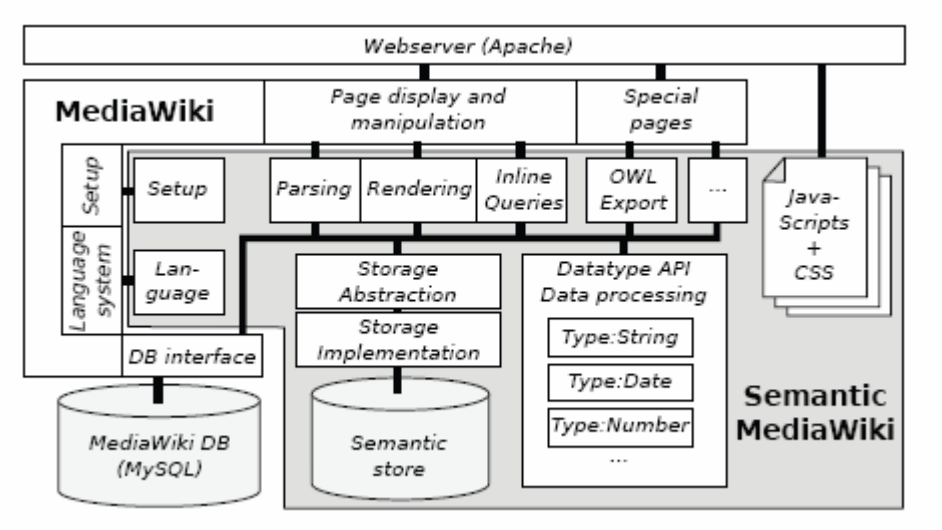
ΕΦΑΡΜΟΓΕΣ ΠΟΥ ΣΤΗΡΙΖΟΝΤΑΙ ΣΕ ΟΝΤΟΛΟΓΙΕΣ

Η αρχική εφαρμογή που θα αναφερθεί είναι αυτή της μεταφορά των δεδομένων του προγράμματος mediawiki από το συντακτικό στο σημασιολογικό ιστό. Η αρχική είσοδος πληροφοριών σε ένα wiki είναι σε μορφή κεμένου. Συνεπώς, το κείμενο wiki παρέχει ήδη πολλές πληροφορίες για την περιγραφή της μορφοποίησης, και ακόμη και μερικές για τη δόμηση του περιεχομένου. Για τον καθορισμό της αμοιβαίας σχέσης των σελίδων μέσα σε ένα wiki, οι σύνδεσμοι υπερ-κειμένου είναι αμφισβητήσιμα το σημαντικότερο χαρακτηριστικό γνώρισμα. Είναι ζωτικής σημασίας για την περιήγηση, και μερικές φορές ακόμη χρησιοποιείται για να ταξινομηθούν τα άρθρα ανεπίσημα. Στη Wikipedia, για παράδειγμα , ένα άρθρο μπορεί να περιέχει τις συνδέσεις με τις σελίδες της μορφής «από το 2005» για να δηλώσει ότι οι δεδομένες πληροφορίες να χρειαστούν αναπροσαρμογές μετά από εκείνο το έτος. Με αυτές τις υπερσυνδέσεις ως είσοδο και την δημιουργία ενός σημασιολογικού οντολογικού σχήματος, οι σελίδες wiki μπορούν να μετασχηματιστούν από συντακτικές σε σημασιολογικές με αποτέλεσμα την καλύτερη πληροφόρηση των αναγνωστών.

Το SemanticMediaWiki επιτρέπει στους χρήστες να προσθέσουν δομημένα στοιχεία στις σελίδες wiki μέσω της απλής σήμανσης wikitext , και τις συνδέει με άλλες σελίδες που δίνουν στα δεδομένα σημασία και σημαντικές ιδιότητες. Με αυτές τις πληροφορίες, το SMW βοηθά κάποιον να ψάξει, να οργανώσει, να αξιολογήσει, και να μοιραστεί τα περιεχόμενα των wikis.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το mediawiki είναι η *http://www.mediawiki.org/wiki/MediaWiki* .

Στην παρακάτω εικόνα φαίνονται τα στάδια μετασχηματισμού, κάποια από τα οποία έχουν ήδη περιγραφεί παραπάνω.

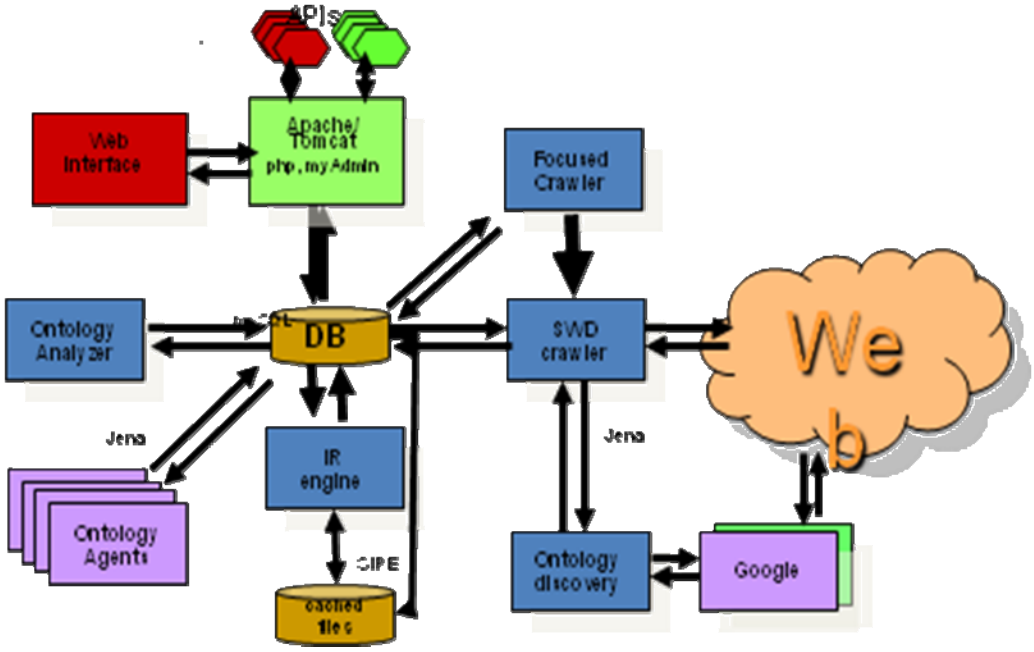


Σχήμα 16: τα τμήματα οργάνωσης του semantic mediawiki

Το δεύτερο παράδειγμα εφαρμογής που αναφέρεται είναι αυτό του semantic google ή SWOOGLE. Το εγχείρημα της μετατροπής των δεδομένων από λεξικογραφικά σε οντολογικά προϋποθέτει μια μηχανή αναζήτησης να δίνει πληροφορίες στους χρήστες όχι ταιριάζοντας λέξεις, αλλά ψάχνοντας σε βάσεις δεδομένων οντολογικών σχημάτων. Το Swoogle είναι βασισμένο στο σύστημα αναζήτησης & ανάκτησης πληροφορίας από τα σημασιολογικά έγγραφα Ιστού (SWDs) σε RDF, και DAML. Ανακαλύπτει SWDs και υπολογίζει τα μεταδεδομένα και τις σχέσεις τους, και τις αποθηκεύει σε ένα σύστημα IR. Το Swoogle χρησιμοποιεί δύο τρόπους για να ανακτήσει σημασιολογικά έγγραφα του Ιστού και διάφορους πράκτορες ανάλυσης για να υπολογίσει τα μεταδεδομένα και τις σχέσεις μεταξύ των εγγράφων και των οντολογιών. Τα μεταδεδομένα αποθηκεύονται σε ένα συγγενικό πρόγραμμα διαχείρισης βάσεων δεδομένων. Με αυτό τον τρόπο είναι δυνατή η απάντηση σε πολύπλοκα ερωτήματα που η μηχανή αναζήτησης του δε μπορεί να απαντήσει, και εξειδικευμένες απαιτήσεις.

Το Swoogle v1 έχει 12K SemanticWebDatas(το οποίο περιέχει οντολογίες και αντικείμενα αυτών) & 100K σχέσεις. Το Swoogle v2 θα καταχωρήσει επίσης τις κατηγορίες και τις ιδιότητες και τα μεταδεδομένα τους και θα έχει παραπάνω από 1.6M SWDs. Παρακάτω φαίνεται η κατανομή των projects που θα χρησιμοποιεί για την εξαγωγή συμπερασμάτων και πώς αυτά συσχετίζονται μεταξύ τους.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το swoogle είναι η *http://swoogle.umbc.edu/* .



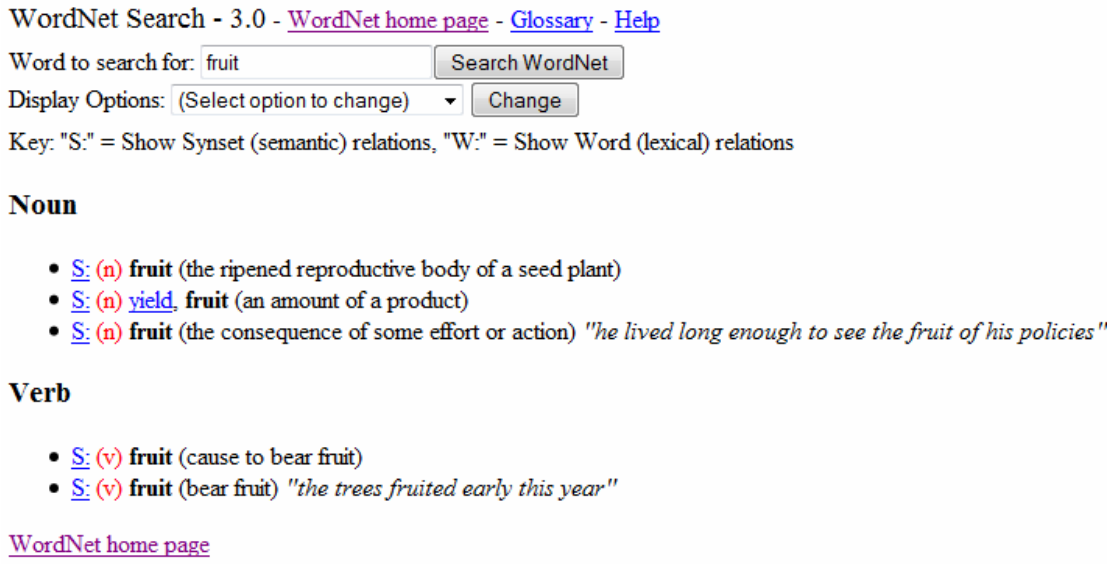
Σχήμα 17: τμήματα οργάνωσης του semantic google

Η τρίτη εφαρμογή που θα μελετηθεί είναι το WordNet. Το WordNet είναι ένα σημασιολογικό λεξικό για τη αγγλική γλώσσα. Ομαδοποιεί τις αγγλικές λέξεις σε σύνολα συνωνύμων, αποκαλούμενων synsets, παρέχει σύντομους γενικούς ορισμούς, και κρατάει διάφορες σημασιολογικές σχέσεις μεταξύ αυτών των συνόλων συνωνύμων. Ο σκοπός είναι διπλός: να παραγάγει έναν συνδυασμό λεξικού και θησαυρού που είναι πιο διαισθητικά χρησιμοποιήσιμος, και να υποστηρίξει εφαρμογές αυτόματης ανάλυσης κειμένων και τεχνητής νοημοσύνης. Η βάση δεδομένων και τα εργαλεία λογισμικού είναι ελεύθερα κατόπιν άδειας BSD. Η βάση δεδομένων μπορεί επίσης να χρησιμοποιηθεί online. Το WordNet δημιουργήθηκε και διατηρείται στο εργαστήριο επιστήμης του Πανεπιστήμιου του Princeton υπό την καθοδήγηση του καθηγητή ψυχολογίας George A. Miller. Η ανάπτυξη άρχισε το 1985. Κατά τη διάρκεια ετών, το πρόγραμμα έλαβε τη χρηματοδότηση από τα κυβερνητικά πρακτορεία ενδιαφερόμενα στην αυτόματη μετάφραση. Από το 2009, η ομάδα WordNet περιλαμβάνει τα ακόλουθα μέλη του γνωστικού εργαστηρίου επιστήμης: George A. Miller, Christiane Fellbaum, Randee Tengi, Pamela Wake. Οι George Miller και Christiane Fellbaum απονεμήθηκαν με το βραβείο Antonio Zampolli του 2006 για την εργασία τους στο WordNet.

Οι σχέσεις hypernym/hyponym μεταξύ των synsets μπορούν να ερμηνευθούν ως σχέσεις ειδίκευσης μεταξύ των εννοιολογικών κατηγοριών. Με άλλα λόγια,

WordNet μπορεί να ερμηνευθεί και να χρησιμοποιηθεί ως λεξικολογική οντολογία υπό την έννοια πληροφορικής. Εντούτοις, μια τέτοια οντολογία πρέπει κανονικά να διορθωθεί πριν χρησιμοποιείται δεδομένου ότι περιέχει εκατοντάδες βασικές σημασιολογικές ασυνέπειες όπως (ι) η ύπαρξη των κοινών ειδικεύσεων για τις αποκλειστικές κατηγορίες και (ii) τους πλεονασμούς στην ιεραρχική ειδίκευση. Επιπλέον, ο μετασχηματισμός του WordNet σε μια λεξικολογική οντολογία χρησιμοποιήσιμη για την αντιπροσώπευση γνώσης πρέπει κανονικά να περιλάβει (ι) σχέσεις ειδίκευσης όπως οι σχέσεις subtypeOf και instanceOf , και (ii) μοναδικά προσδιοριστικά με κάθε κατηγορία. Αν και τέτοιες διορθώσεις και μετασχηματισμοί έχουν εκτελεσθεί και έχουν τεκμηριωθεί ως τμήμα της ένταξης WordNet 1.7 στη συνεταιριστικά αναθεωρήσιμη βάση γνώσεων webKB-2. Επίσης τα περισσότερα προγράμματα που υποστηρίζουν ότι κάνουν χρήση του WordNet για δικές τους εφαρμογές βασισμένες στη γνώση (χαρακτηριστικά, γνώση-προσανατολισμένη στην ανάκτηση πληροφοριών) απλά το επαναχρησιμοποιούν άμεσα.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το wordnet είναι η *http://wordnet.princeton.edu/* .

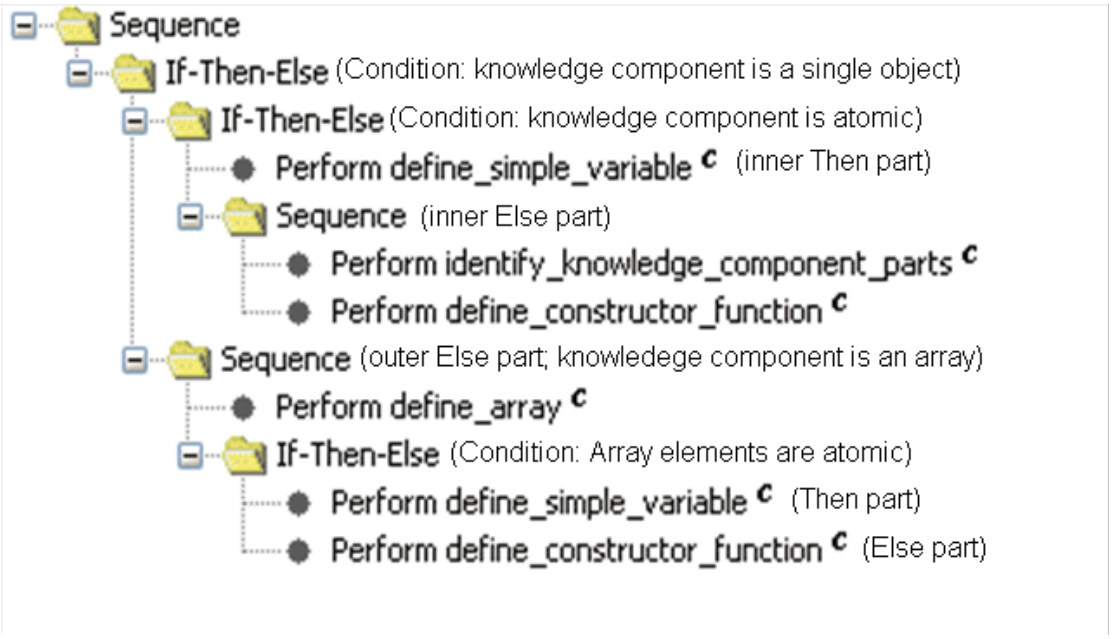


Σχήμα 18: παράδειγμα του wordnet

Μια ακόμη εφαρμογή των οντολογιών είναι μια ελληνική εφαρμογή με το όνομα mathesis. Πρόκειται για μια οντολογία που περιγράφει κανόνες των μαθηματικών, και δημιουργήθηκε με το πρόγραμμα protege με σκοπό την εύκολη εκμάθηση των μαθηματικών και την ανάπτυξη της ηλεκτρονικής μάθησης. Ολοκληρώθηκε τον Ιανουάριο του 2009 και δημοσιεύτηκε στο 14ο διεθνές συνέδριο για την τεχνητή νοημοσύνη στην εκπαίδευση. Το mathesis

Algebra tutor είναι μια δικτυακή εφαρμογή, και πιο συγκεκριμένα μια ιστοσελίδα που βοηθά στην εκμάθηση επίλυσης μαθηματικών εξισώσεων.

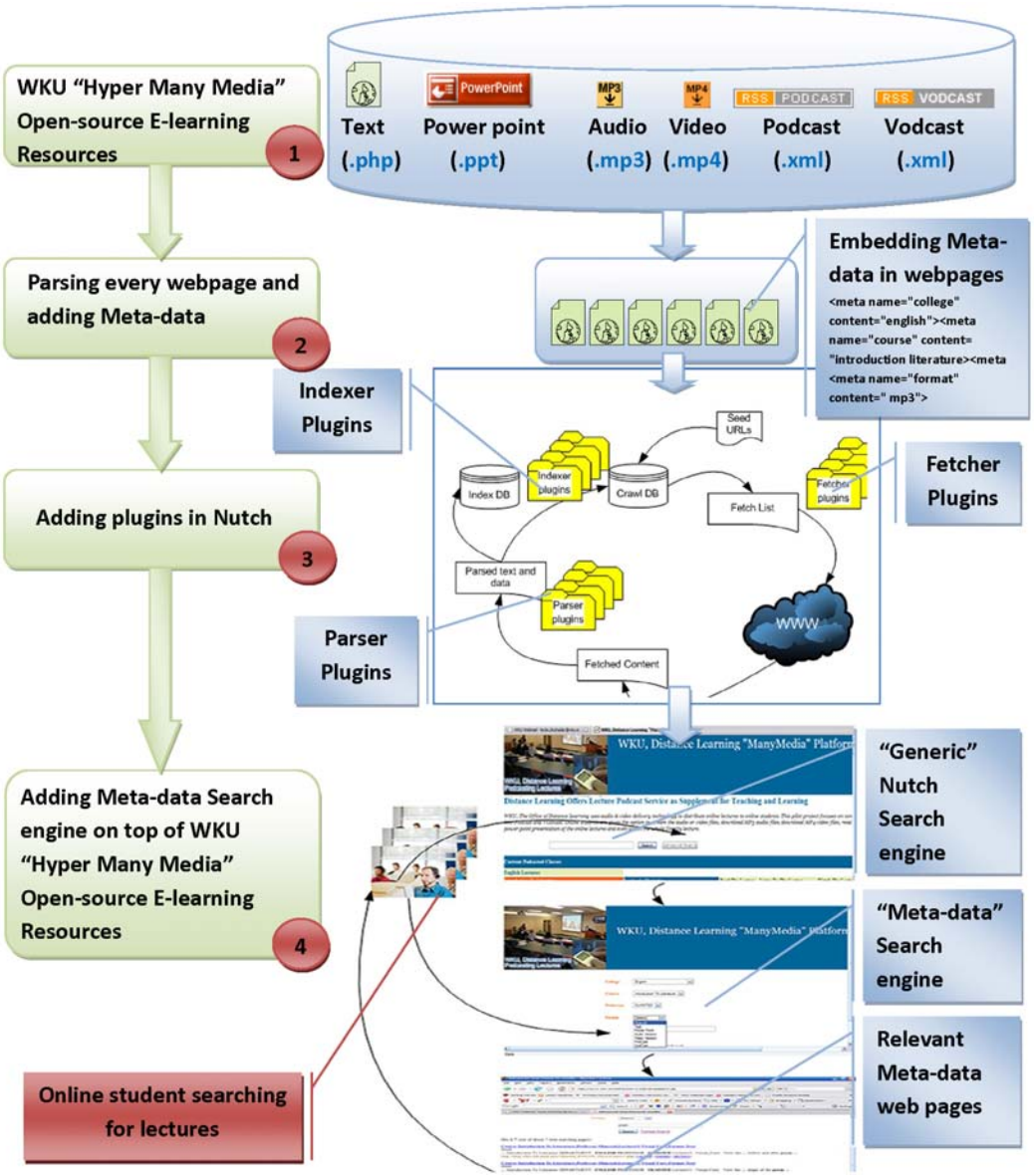
Η ιστοσελίδα από όπου μπορεί να βρεθεί το mathesis είναι η *http://www.ics.forth.gr/mathesis* .



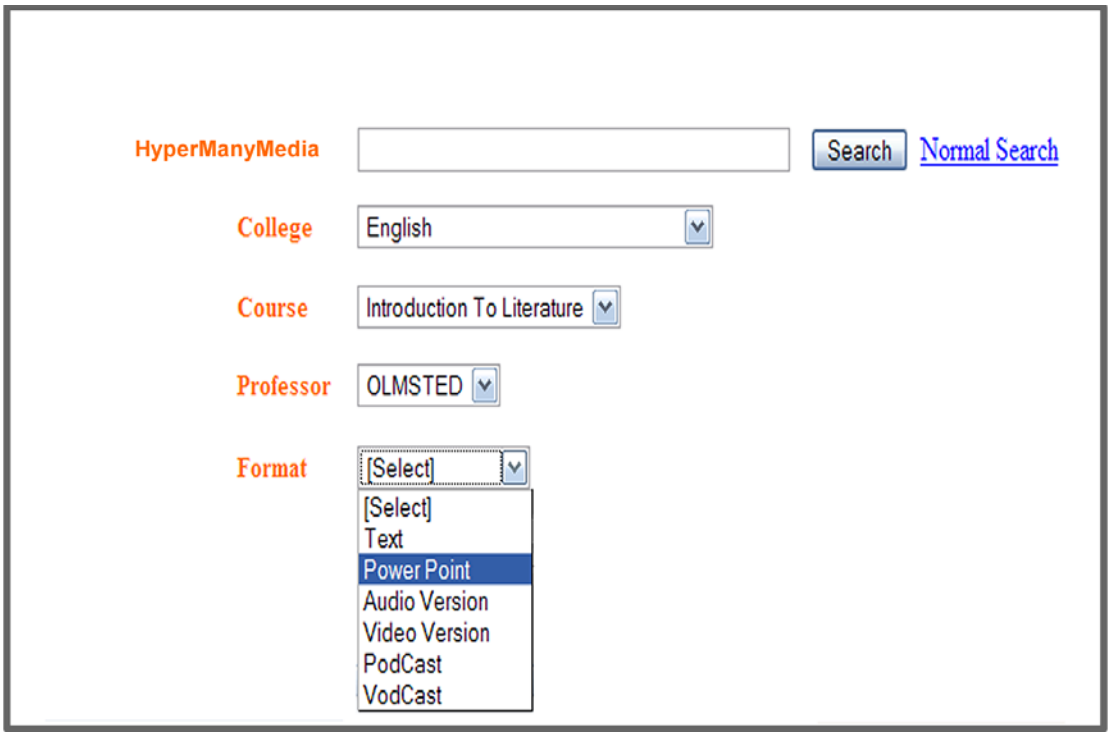
Σχήμα 19: στιγμιότυπο αλγορίθμου της οντολογίας mathesis

Μια διαφορετικής φύσης εφαρμογή που βασίζεται σε οντολογίες είναι η HyperManyMedia. Πρόκειται για μια εφαρμογή που χρησιμοποιείται επίσης για εκπαιδευτικούς λόγους, προσφέροντας στους εκπαιδευόμενους γνώση σε μορφή κειμένου, ήχου, εικόνας και βίντεο, με σκοπό να γίνει μια διαδικτυακή εξατομικευμένη μηχανή αναζήτησης πληροφοριών. Ο χρήστης μπορεί να δώσει μια απλή ερώτηση για το θέμα που ψάχνει, και το πρόγραμμα κάνει αναζήτηση στην τοπική βάση γνώσης του και δίνει αποτελέσματα, επίσης ψάχνει σε όλη τη διαδικτυακή βάση γνώσης, βρίσκει δεδομένα, τα δίνει στο χρήστη και προτείνει μεταδεδομένα για αυτά τα δεδομένα. Έτσι μπορεί κάποιος να δημιουργήσει μεταδεδομένα για οποιαδήποτε πληροφορία που προϋπήρχε αλλά δεν υπήρχε μεταπληροφορία για αυτή.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το hyperManyMedia είναι η *http://www.serpanalytics.com/sites/hypermanymedia.wku.edu* .



Σχήμα 20: παραπάνω φαίνεται γραφικά η λειτουργία του HyperManyMedia



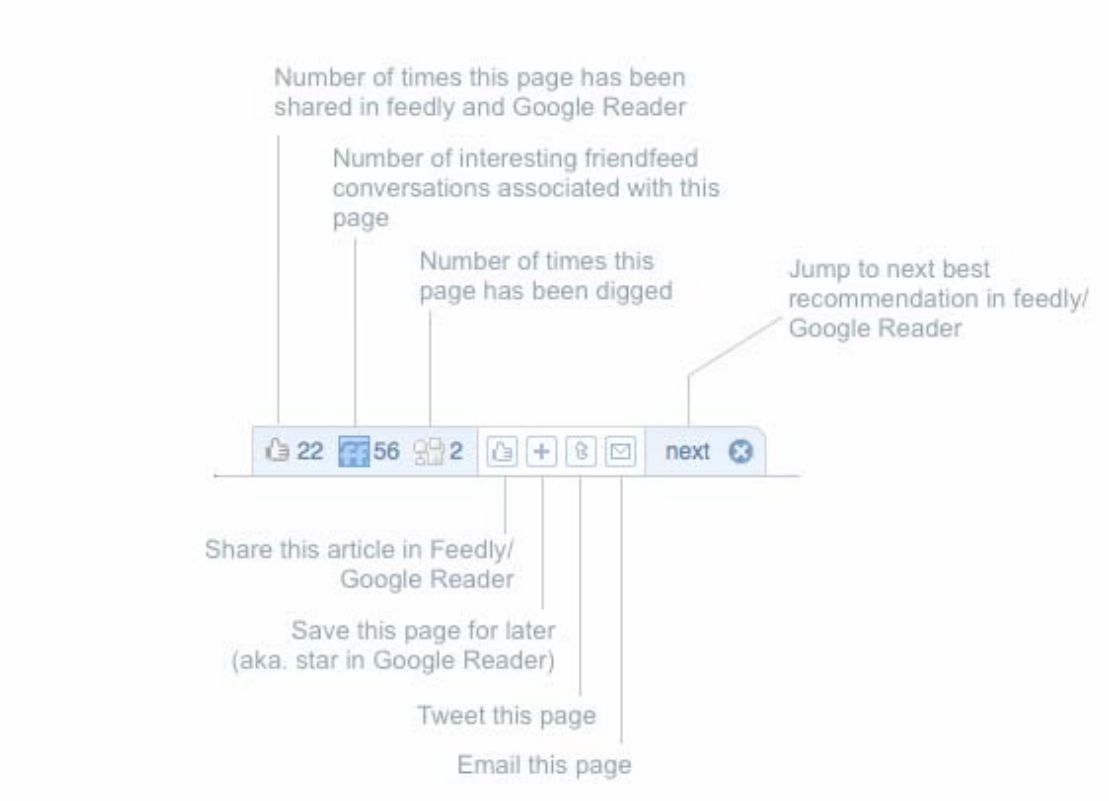
Σχήμα 21: παραπάνω φαίνεται ένα παράδειγμα της μηχανής αναζήτησης του HyperManyMedia

ΕΦΑΡΜΟΓΕΣ ΠΟΥ ΣΤΗΡΙΖΟΝΤΑΙ ΣΕ ΑΛΛΕΣ ΤΕΧΝΙΚΕΣ

Αρχικά θα αναφερθεί το Feedly. Το Feedly περιγράφεται ως «ηλεκτρονικό πρωτοσέλιδο περιοδικού» Όταν εμφανίστηκε, τον Αύγουστο του 2008, ονομάστηκε ως «μια εναλλακτική διεπαφή για τον αναγνώστη της Google». Εντούτοις με την έναρξη του mini Feedly, ένα μικρόπαράθυρο που εμφανίζεται στο κατώτατο σημείο της οθόνης μέσω των blogs στον Ιστό, η υπηρεσία έχει αναβαθμίσει την αναζήτηση των πληροφοριών που μπορεί να ενδιαφέρουν έναν ανάγνωστη blog. Επείσης, στο mini Feedly έχουν ενσωματωθεί τα Twitter, FriendFeed, Google Search, Mozilla's Ubiquity, και άλλα.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το Feedly είναι η *http://www.feedly.com/* .

Παρακάτω φαίνονται κάποιες από τις πληροφορίες που προσφέρει:

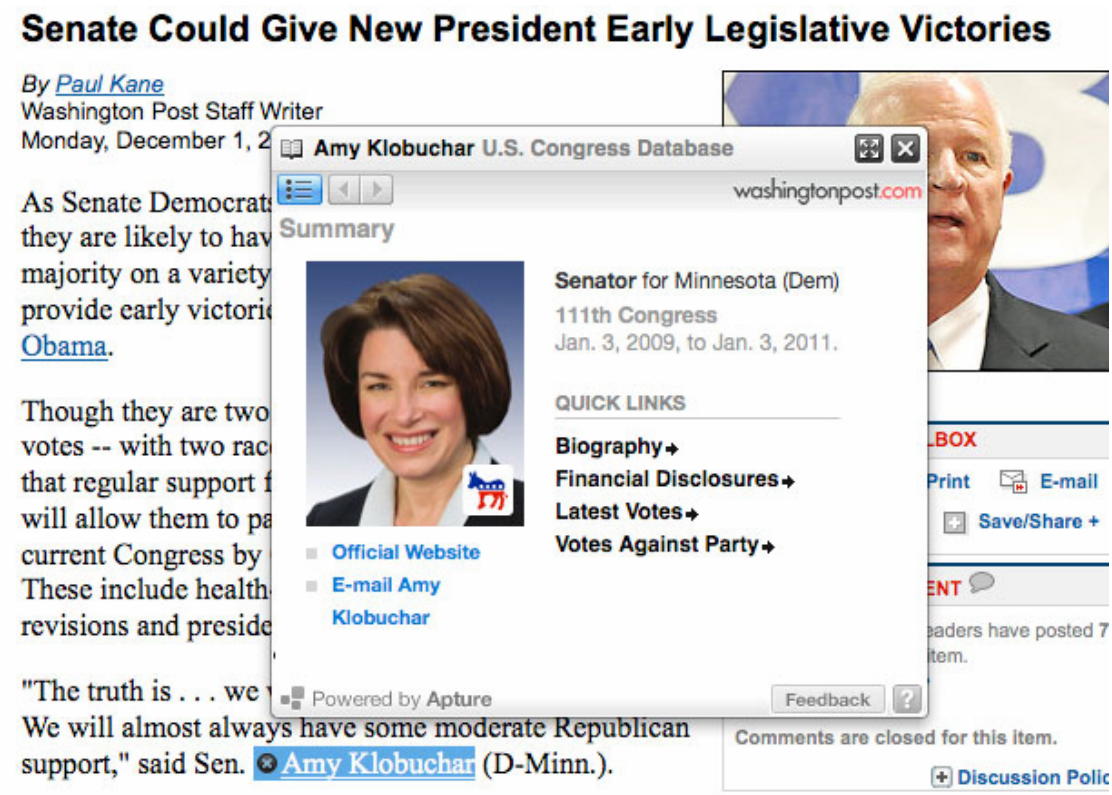


Σχήμα 22: κάποιες από τις πληροφορίες που δίνει το Feedly

Η δεύτερη διαδικτυακή εφαρμογή που θα αναφερθεί είναι το Apture, το οποίο είναι ένα plug-in Javascript επιτρέπει στους εκδότες να προσθέσουν

πληροφορίες στις συνδέσεις, μέσω παραθύρων που ανοίγουν όποτε οι χρήστες περνάν από πάνω το ποντίκι ή κάνουν «κλικ» σε αυτές. Οι χρήστες του δηλώνουν εντυπωσιασμένοι από τον όγκο των δεδομένων και του ποσού των πολυμέσων που μπορεί να χωρέσει ένα τόσο μικρό (pop-up) αναδυόμενο παράθυρο. Για παράδειγμα, οι αναγνώστες στο washingtonpost.com και σε άλλες περιοχές που χρησιμοποιούν Apture μπορούν να δουν πλούσιο, σχετικό, βασισμένο στα συμφραζόμενα περιεχόμενο από τα δημοφιλή του Wikipedia, το YouTube και το Flickr χωρίς να φύγει από τον ιστοχώρο στον οποίο βρίσκεται. Παρακάτω φαίνεται ένα αναδυόμενο παράφυρο του Apture ως παράδειγμα.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το Apture είναι η *http://www.apture.com/* .



Σχήμα 23: παράδειγμα του Apture

Τέλος θα αναφερθεί το Calais 4.0 που είναι η τελευταία ανανέωση του Calais.Το Calais είναι ένα εργαλείο που επιτρέπει στους εκδότες να ενσωματώσουν τη σημασιολογική λειτουργία ενός προϊόντος μέσα στις ιδιότητές του, επιτρέποντας τους να ταξινομήσουν τα περιεχόμενα ως ανθρώπους, θέσεις, επιχειρήσεις, γεγονότα, και άλλα.

Το Calais 4.0 υπερέβη και επέτρεψε στους εκδότες να ενσωματώσουν τα περιεχόμενα τους δεδομένα από το Wikipedia, το GeoNames, τη βάση δεδομένων κινηματογράφων Διαδικτύου (IMDB), το Shopping.com και άλλα. Το Calais 4.0 επίσης άφησε τους εκδότες να μοιραστούν τα σημασιολογικά μεταδεδομένα για το περιεχόμενό τους με μηχανές αναζήτησης, πρακτορεία ειδήσεων, και υπηρεσίες σύστασης ιστοχώρων με παρόμοιο περιεχόμενο.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το Calais είναι η *http://www.opencalais.com/news/calais-40-has-arrived* .

(τα στοιχεία από το μέρος 2 είναι μεταφορά από τον ιστοχώρο http://www.readwriteweb.com/archives/top_10_semantic_web_products_of_2009.php και το άρθρο έχει γραφτεί από τον Richard MacManus τον Δεκέμβριο του 2009.)

ΔΗΜΟΦΙΛΕΙΣ ΣΗΜΑΣΙΟΛΟΓΙΚΕΣ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΕΙΣ

Η ιδανική μηχανή αναζήτησης θα ήταν σε θέση να ταιριάξει με τις ερωτήσεις αναζήτησης το ακριβές πλαίσιο και να επιστρέψει τα αποτελέσματα μέσα σε αυτό. Ενώ το Google, το Yahoo και το Live συνεχίζουν να «κρατούν τα ινία» στην αναζήτηση, παρακάτω θα αναφερθούν μηχανές που υιοθετούν μια βασισμένη στη σημασιολογία μέθοδο όπου το τελικό αποτέλεσμα είναι πιο σχετικό, πιο ακριβές και μη εξαρτώμενο από σχηματισμούς ομάδων λέξεων ή αλγορίθμους μέτρησης συνδέσεων, οι οποίοι κάνουν παραδοσιακότερη τη μηχανή αναζήτησης αλλά και εμφανίζουν πολλά μη επιθυμητά αποτελέσματα.

Το κύριο παράδειγμα σε αυτή την κατηγορία είναι το swoogle το οποίο βασίζεται σε οντολογίες και αναπτύχθηκε παραπάνω, οπότε παρακάτω θα αναφερθούν μηχανές που βασίζονται σε άλλες τεχνικές.

Το πρώτο παράδειγμα σε αυτόν τον τομέα είναι το “Hakia”. Σε αυτό, ο μαθητής του Δρ Riza, C. Berkan, προσπαθεί να εντοπίσει τις ερωτήσεις που θα μπορούσαν να υποβληθούν σχετικά με ένα έγγραφο, και τις χρησιμοποιεί ως τίτλους για το περιεχόμενο του. Οι ερωτήσεις αναζήτησης συνδέονται με αποτελέσματα και ταξινομούνται χρησιμοποιώντας έναν αλγόριθμο που σημειώνει στην ανάλυση της πρότασης και πόσο πολύ ταιριάζουν το αποτέλεσμα με την ερώτηση.

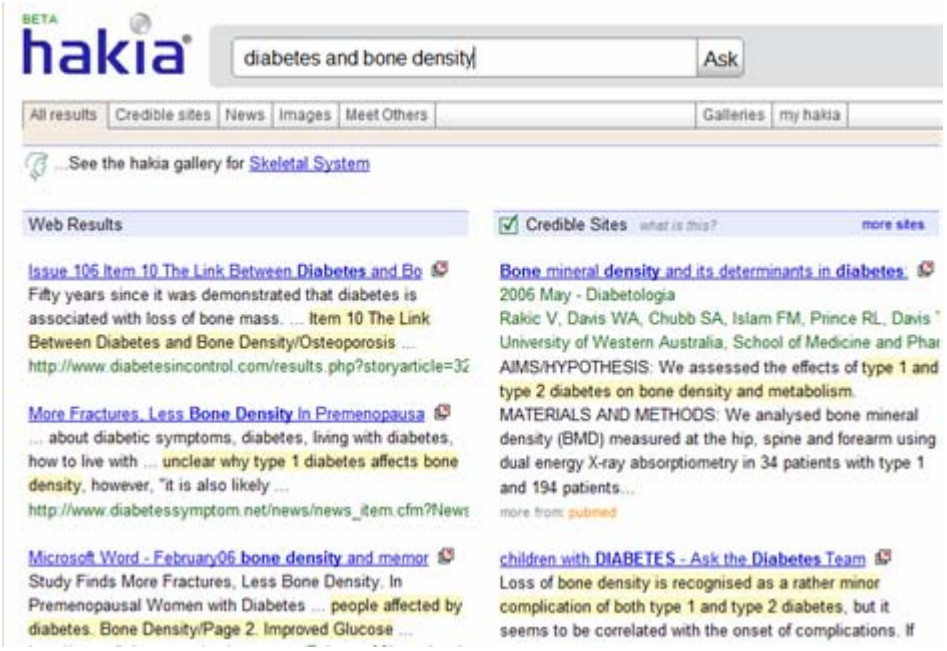
Το Hakia χρησιμοποιεί τρεις τεχνολογίες για την εύρεση της σημασίας των δεδομένων:

OntoSem - sense repository(γλωσσική βάση δεδομένων όπου οι λέξεις είναι ταξινομημένες σε διάφορες κατηγορίες).

QDEX - Query indexing technique (αντικατάσταση για το δείκτη που οι περισσότερες μηχανές χρησιμοποιούν για να σώσουν το περιεχόμενο με δείκτη για τις ερωτήσεις στις οποίες απαντά το περιεχόμενο, μειώνοντας έτσι τον όγκο πληροφορίας που πρέπει να ψάχνουν).

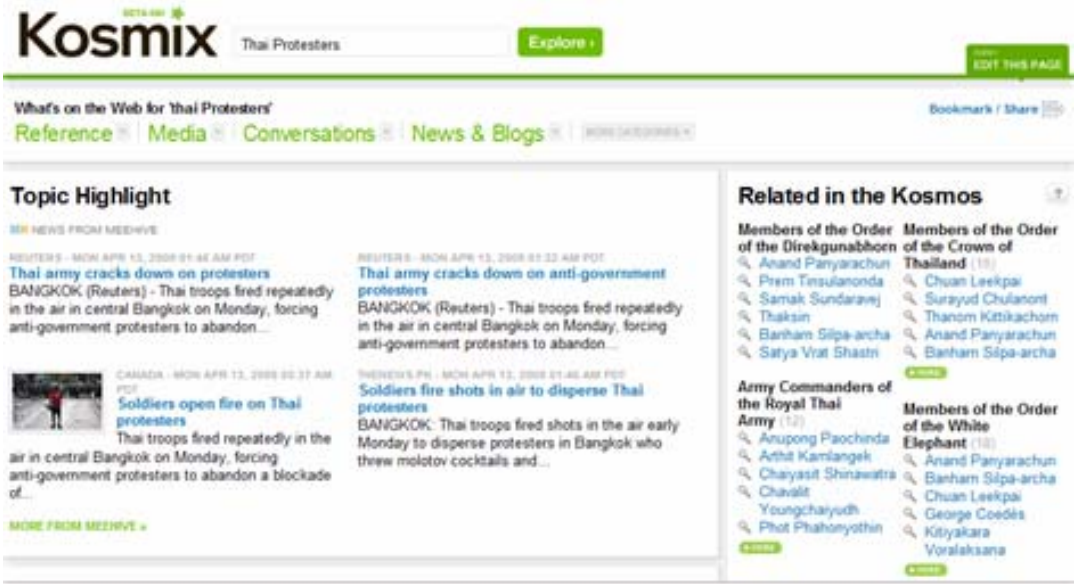
SemanticRank algorithm (ταξινομεί το περιεχόμενο βάσει περισσότερης ανάλυσης της πρότασης. Η αξιοπιστία και η ηλικία του περιεχομένου χρησιμοποιούνται επίσης για να καθορίσουν τη σχετικότητα).

Η ιστοσελίδα από όπου μπορεί να βρεθεί το Hakia είναι η *http://www.hakia.com/* .



Σχήμα 24: η επιφάνεια εργασίας του hakia

Η δεύτερη μηχανή αναζήτησης είναι η “Kosmix”.



Σχήμα 25: η επιφάνεια εργασίας του Kosmix

Η επιχείρηση αναζήτησης στην Kosmix δεν έχει ως αποτέλεσμα ένα url από τα πολλά το οποίο θα επιλέξει ο χρήστης, αλλά μιας σελίδας, αποκαλούμενης ως «ο οδηγός σας για τον Ιστό». Σε αυτή τη σελίδα υπάρχει κράμα πληροφοριών από κομμάτια άλλων ιστοσελίδων, τα οποία απαντούν στο ερώτημα του χρήστη. Παραδείγματος χάριν, η αναζήτηση της ανταλλαγής

πιστωτικής προεπιλογής παρείχε ένα μεγάλο κράμα συνδέσεων, βίντεο και άρθρων. Η τεχνική που χρησιμοποιεί είναι η meaning in search queries(απόδοση πληροφορίας στις ερωτήσεις του χρήστη), μια τεχνική την οποία χρησιμοποιεί και η μηχανή αναζήτησης “Cognition”.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το Cosmix είναι η *http://www.kosmix.com/* .

Μια ακόμη επιτυχημένη μηχανή αναζήτησης είναι και η “exalead”, η οποία κάνει αναζήτηση ανάλογα με τις εικόνες μιας σελίδας..

Σχήμα 26: επιφάνεια εργασίας του exalead

Η μηχανή αναζήτησης εικόνας ήταν μοναδική για τον πλήθος επιλογών της να βελτιώσει την αναζήτηση ανάλογα με το μέγεθος, το χρώμα και το περιεχόμενο της εικόνας. Πολλά από αυτά τα χαρακτηριστικά γνωρίσματα έχουν εμφανιστεί από τότε σε άλλες μηχανές αναζήτησης εικόνας. Η επιχείρηση έχει εστιάσει στην αγορά επιχειρηματικής αναζήτησης, προσπαθώντας ουσιαστικά να λύσει το πρόβλημα της αναζήτησης του περιεχομένου όπου το page rank είναι μικρό.

Η ιστοσελίδα από όπου μπορεί να βρεθεί το Exalead είναι η *http://www.exalead.com/search/* .

(τα στοιχεία από το μέρος 3 είναι μεταφορά από τον ιστοχώρο http://www.searchenginejournal.com/semantic-search-engines/9832/ και το άρθρο έχει γραφτεί από τον Arun Radhakrishnan τον Απρίλιο του 2009.)

1.13. Σύνοψη

Αυτή η εργασία ξεκίνησε διατυπώνοντας την ανάγκη για τη επέκταση του Syntactic web σε Semantic web, που διευκολύνεται με τη χρήση οντολογικών σχημάτων με τα οποία γίνεται αναπαράσταση των εννοιών των περιεχομένων του web. Στη συνέχεια αναφέρθηκαν αλγόριθμοι που καθιστούν ένα τέτοιο εγχείρημα εφικτό, καθώς και οι τεχνικές που χρησιμοποιούν. Αυτές οι τεχνικές χρησιμοποιούν την έννοια ή την τοπολογία των μορίων ενός οντολογικού σχήματος με σκοπό να αναγνωρίσουν όμοια οντολογικά σχήματα, τα οποία θα ταιριάξουν ή/και θα επεκτείνουν. Έπειτα αναφέρθηκε ένας πρότυπος τρόπος χρήσης όλων των ανωτέρω με σκοπό την καθολική και κατανεμημένη παρακολούθηση και προσαρμογή οντολογιών κατά το δοκούν ώστε να γίνουν πιο εύχρηστες και να συνδυαστούν όσο το δυνατόν περισσότερο. Ως αποτέλεσμα της παραπάνω συζήτησης αναφέρονται τρία παραδείγματα εφαρμογής των οντολογικών σχημάτων στην οργάνωση πληροφοριών. Τέλος έγινε μια αναφορά στην απόδοση των τεχνικών, με την προσέγγιση ότι είναι πιο εύχρηστες όταν συνδυαστούν.

2 Μέρος 2ο: ανάπτυξη λογισμικού διερεύνησης ομοιοτήτων και συγχώνευσης δυο οντολογικών σχημάτων

Εισαγωγή

Σε αυτό το μέρος της εργασίας περιγράφονται κάποια προγράμματα που κάνουν σύμπτυξη οντολογικών σχημάτων (ontology merging) και σε ποιους τομείς είναι διαδεδομένη η χρήση τους. Επίσης γίνεται ανάπτυξη και σχεδίαση λογισμικού με στόχο την σύμπτυξη οντολογικών σχημάτων με την παρεμβολή του χρήστη για τη λήψη αποφάσεων.

Έπειτα δημιουργείται η φιλοσοφία, και καθορίζονται οι συντακτικοί και γλωσσικοί κανόνες τους οποίους πρέπει να διέπει η καινούρια οντολογία, και γενικά όλες οι απαιτήσεις για τη δημιουργία μιας οντολογίας. Πέρα από αυτό όμως πρέπει να καθοριστούν και οι απαιτήσεις του χρήστη από το πρόγραμμα. Η πιο βασική από αυτές έγκειται στις λειτουργικές απαιτήσεις του προγράμματος και δηλώνει ότι ο χρήστης πρέπει να συμμετέχει στη λήψη αποφάσεων για τη δημιουργία της νέας οντολογίας, και να απαντά στις προτάσεις για ένωση υποδέντρων από το πρόγραμμα.

Τέλος εντοπίζονται εκ των προτέρων κάποιοι περιορισμοί, που βάζουν όρια στο πρόγραμμα, όπως στη γλώσσα και της έκδοσή της καθώς και του συντακτικού, υποθέσεις και διευκρινήσεις για τις εισόδους αλλά και την έξοδο.

2.1 Σχετικά προγράμματα

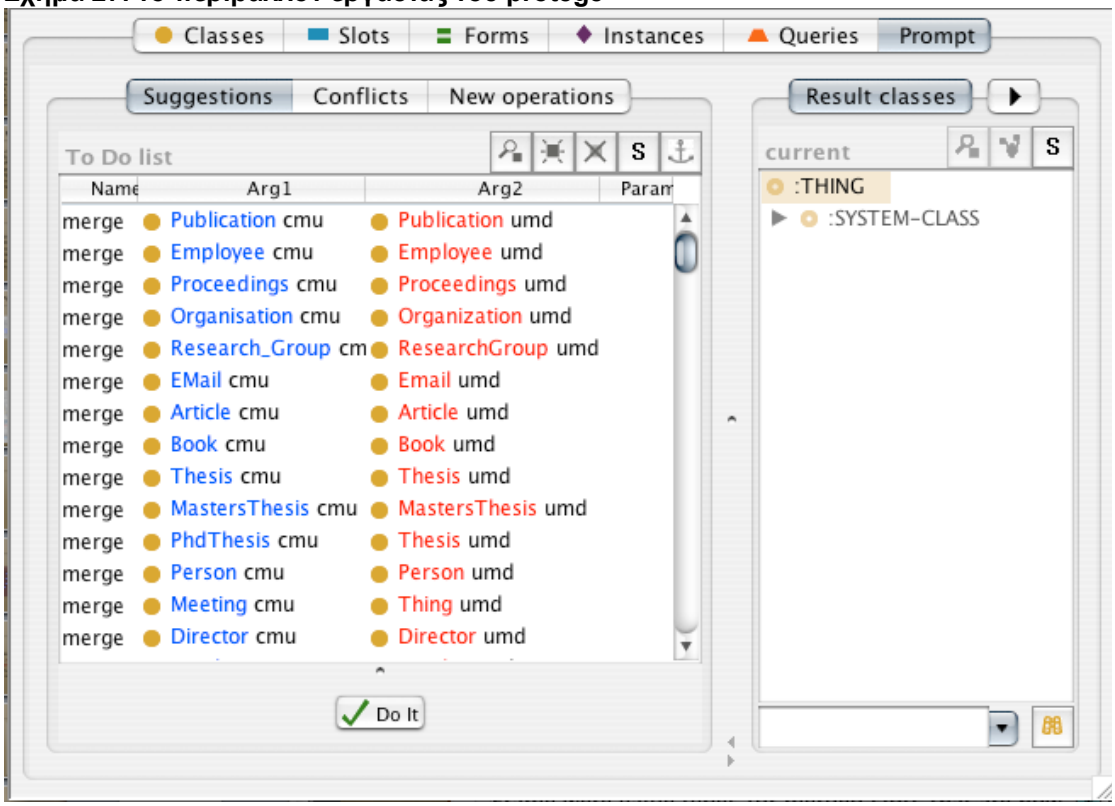
Υπάρχουν αρκετά προγράμματα τα οποία δίνουν στο χρήστη τη δυνατότητα να δημιουργήσει οντολογίες γραφικά και να ενώσουν ή να συσχετίσουν δυο οντολογικά σχήματα, αλλά ενέχουν πολλούς περιορισμούς καθώς δε χρησιμοποιούν ποικιλία γλωσσών αναπαράστασης οντολογιών και κάποια από αυτά δε χρησιμοποιούν διαδραστικά παράθυρα με το χρήστη για τη συνένωση των οντολογιών. Αντίθετα σε κάποια από αυτά ο χρήστης έχει τον τελευταίο λόγο για το ποια κομμάτια θα ενωθούν, αν δεν είναι βέβαιη η συσχέτιση δύο οντολογικών κλάσεων ή αν υπάρχει σύγκρουση αποτελεσμάτων.

Το πιο δημοφιλες πρόγραμμα που ασχολείται με αυτή τη διαδικασία είναι το prompt, ένα plug-in του protege. Το protege είναι ένα ανοιχτού κώδικα πρόγραμμα δημιουργίας και αλλαγής οντολογιών με πολλές δυνατότητες και μεγάλη βάση δεδομένων από οντολογίες, στις οποίες ο χρήστης έχει πρόσβαση, και λειτουργεί ανεξαρτήτως λειτουργικού συστήματος. Πιο συγκεκριμένα το Protégé εφαρμόζει έναν μεγάλο αριθμό από δομές και ενέργειες, που υποστηρίζουν την δημιουργία, την απεικόνιση και τον χειρισμό των οντολογιών με διάφορες μορφές αναπαράστασης. Οι οντολογίες, που έχουν συνταχθεί με το Protégé, μπορούν να εξαχθούν με ποικίλους μορφές όπως οι RDF(s), OWL και XML. Μπορεί να προσαρμοστεί, ώστε με εύκολο τρόπο να υποστηρίζει την δημιουργία των προτύπων γνώσης και την είσοδο των στοιχείων. Επιπλέον, το Protégé μπορεί να επεκταθεί μέσω μιας plug-in αρχιτεκτονικής και ενός API (Προγραμματιστικής Διασύνδεσης) βασισμένο σε Java – Application Programming Interface(API) για την δημιουργία εργαλείων και εφαρμογών βασισμένων σε γνώσεις. Πρέπει να σημειωθεί ότι και το protégé έχει επιλογή συνδυασμού οντολογικών σχημάτων, αλλά τις πραγματοποιεί αυτόνομα και χωρίς την παρέμβαση του χρήστη.

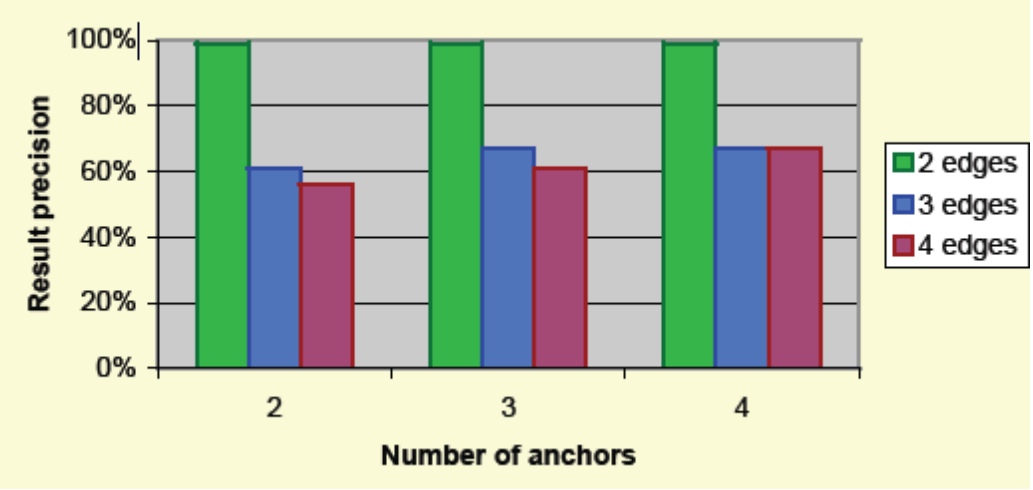
Οι πληροφορίες βρέθηκαν από τη σελίδα *http://protege.stanford.edu/plugins/prompt/prompt.html* .

Παρακάτω φαίνεται το αλληλεπιδραστικό παράθυρο του prompt.

Σχήμα 27: το περιβάλλον εργασίας του protege



Παρακάτω φαίνονται τα στατιστικά στοιχεία των επιτυχών συνδέσεων ανάλογα με τον αριθμό των συσχετιζόμενων κόμβων σε δύο οντολογικά σχήματα. Από τα αποτελέσματα παρατηρείται ότι όταν γίνεται έστω λίγο πιο πολύπλοκη η σχέση των δύο οντολογιών από τις δύο ακμές, τότε το ποσοστό επιτυχία μειώνεται σημαντικά. Αυτό συμβαίνει γιατί όσο καλοί κι αν είναι οι πράκτορες εύρεσης ομοιότητας, δε μπορούν να κατανοήσουν την ανθρώπινη γλώσσα, άρα φαίνεται ότι ακόμη και με την παρούσα τεχνολογία, στο συγκεκριμένο τομέα, για πιο σωστά αποτελέσματα είναι απαραίτητη η ανθρώπινη απόφαση.



Σχήμα 28: στατιστικά στοιχεία που δείχνουν την αποτελεσματικότητα του ontology merging με το protege

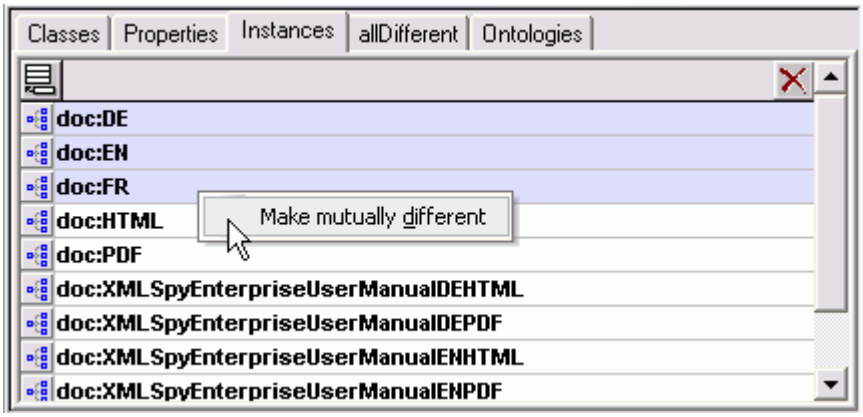
Ένα ακόμη πρόγραμμα με παρόμοιες δυνατότητες και απόδοση είναι το Chimaera. Χρησιμοποιείται για την ένωση μεγάλων οντολογικών σχημάτων, προτείνει στο χρήστη κάποια σημεία ένωσης, και με την καθοδήγησή του κάνει την ένωση. Έχει χρησιμοποιηθεί στην εργασία “High Performance Knowledge Base” για την ανάλυση των εισερχόμενων οντολογιών. Επίσης χρησιμοποιείται από πολλές εταιρίες όπως η VerticalNet και η Cisco.

Οι πληροφορίες βρέθηκαν από τη σελίδα *http://www-ksl.stanford.edu/software/chimaera/* .

Σχήμα 29: το περιβάλλον εργασίας του chimaera

Ένα τρίτο πρόγραμμα είναι το Altova Semantic Works, το οποίο μοιάζει πολύ με το protégé, και χωρίζει τα αρχεία owl σε κομμάτια, ανάλογα αν περιγράφουν κλάσεις αντικείμενα ή ιδιότητες. Κάποια στοιχεία του μπορεί να καταλάβει ο χρήστης αν δεί το γραφικό περιβάλλον του. Στο πάνω παράθυρο φαίνονται οι κλάσεις της οντολογίας, αν πατηθεί το κουμπί δίπλα φαίνονται οι ιδιότητες, μετά η υλοποίηση των οντολογιών, ποιες είναι οι αντιθέσεις μεταξύ κλάσεων ή ιδιοτήτων, και όλες οι «ανοιγμένες» οντολογίες. Στο παρακάτω παράθυρο φαίνονται όλες οι διεπαφές μιας κλάσης και όλες οι ιδιότητες που συνδέονται με αυτή.

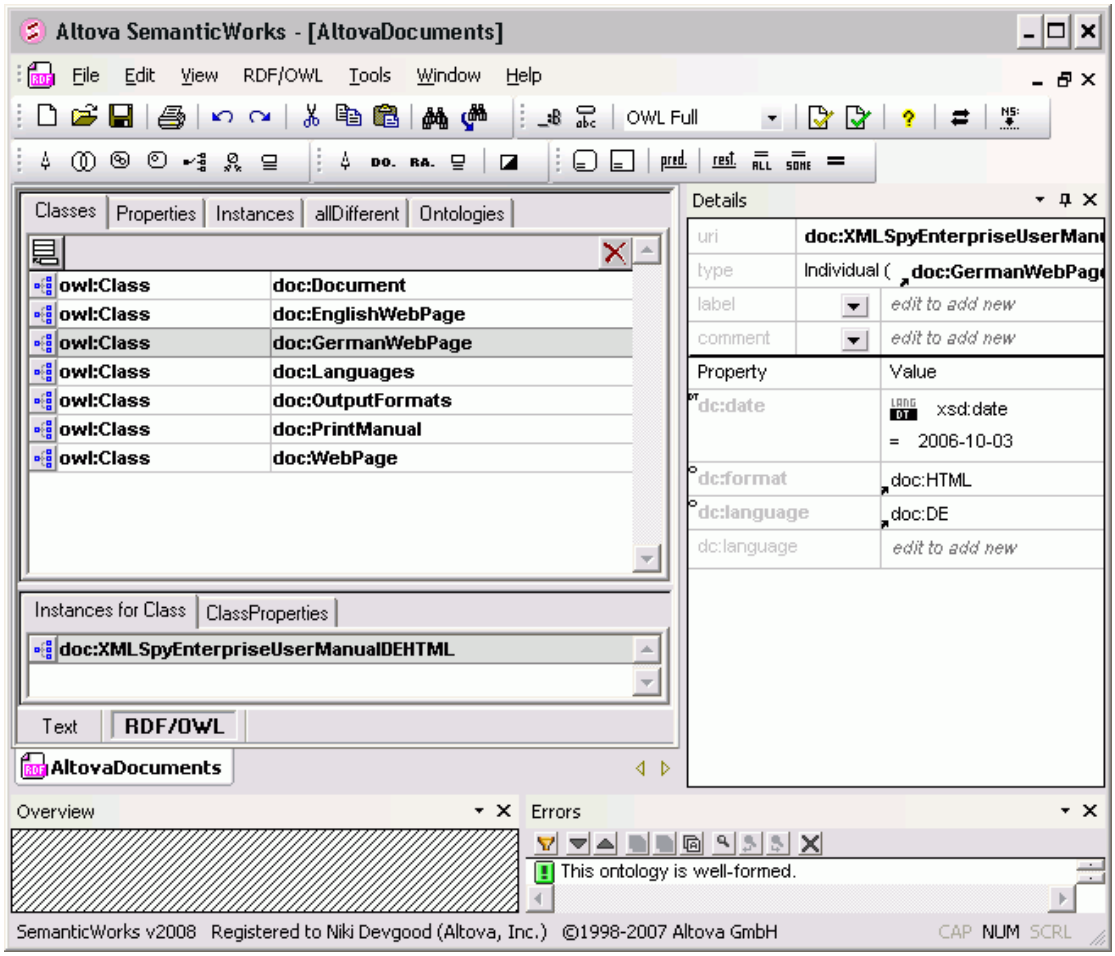
Εδώ φαίνονται όλες οι κατηγορίες στις οποίες χωρίζονται οι πληροφορίες που εξάγονται από ένα αρχείο owl (classes, properties, instances, allDiffenrent, Ontologies).



Σχήμα 30: οι κατηγορίες στις οποίες χωρίζει το Altova Semantic Works μια οντολογία

Παρακάτω μπορεί να φανεί το δέντρο γραφικά αλλά και ο κώδικας ανάλογα με το ποια λειτουργία βολεύει το χρήστη τη δεδομένη στιγμή. Για παράδειγμα κάποιος μπορεί να χρησιμοποιεί τη γραφική απεικόνιση για ανάγνωση της οντολογίας και τον κώδικα για να τη διορθώσει. Σε αυτό το σημείο πρέπει να παρατηρηθεί ότι η γραφική απεικόνιση σε αυτό όπως και στα υπόλοιπα προγράμματα, αλλά και σε αυτό που θα υλοποιηθεί, δε γίνεται με βάση την ιεραρχία των κλάσεων αλλά με βάση τον κώδικα του xml αρχείου. Το tag που ενέχει ένα tag, στη γραφική αναπαράσταση γίνεται το πρώτο παιδί του δεύτερου αντίστοιχα. Η περιγραφή μιας οντολογίας με διαγράμματα uml είναι βεβαίως δυνατή, αλλά αυτή η αναπαράστασή της δεν είναι εύχρηστη στη συσχέτισή της με τον κώδικα, ούτε στην επεξεργασία των δεδομένων που περιέχουν τα xml αρχεία, πράγμα που χρειάζονται αυτά τα προγράμματα.

Οι πληροφορίες βρέθηκαν από τη σελίδα *http://www.altova.com/* .

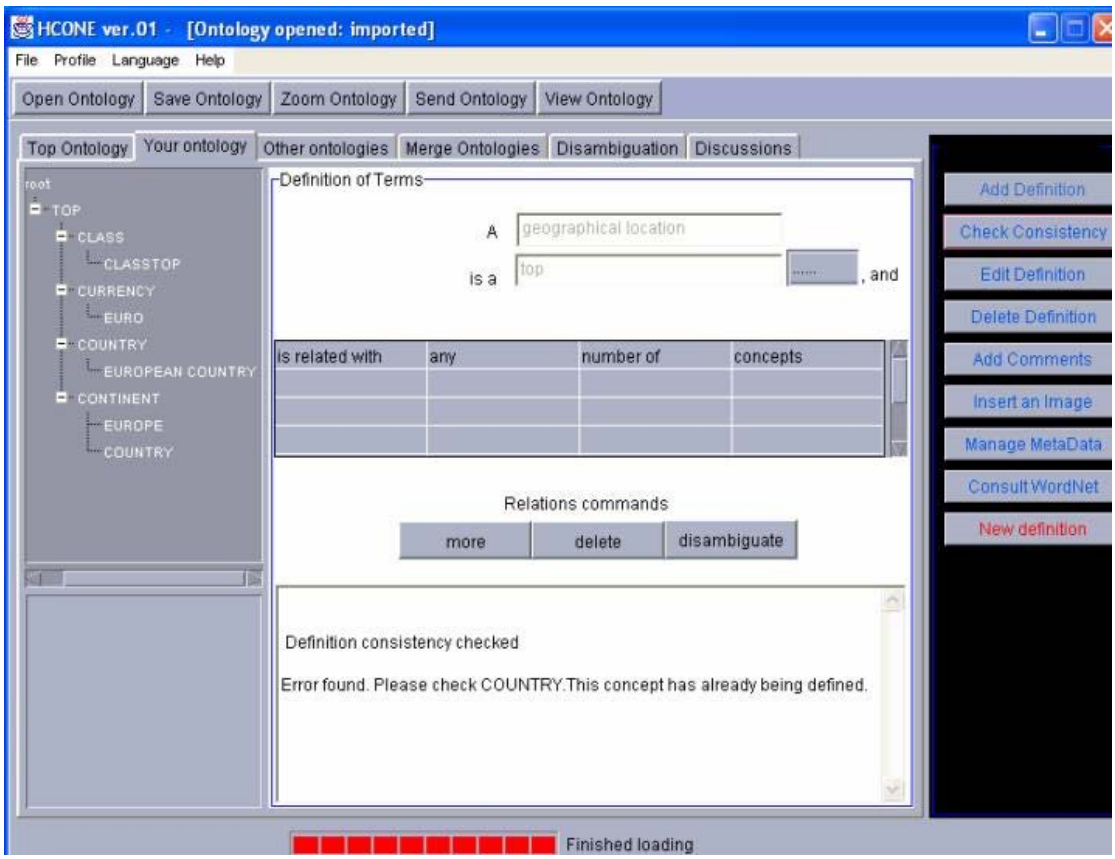


Σχήμα 31: περιβάλλον εργασίας του Altova Semantic Works

Ένα ακόμη πρόγραμμα συγχώνευσης οντολογιών που θα αναφερθεί είναι το DAG-Edit. Χρησιμοποιείται πολύ από ανθρώπους που ασχολούνται με βιολογικές βάσεις δεδομένων, και δίνει μεγάλη έμφαση στους τύπους και τα πεδία ορισμού των κλάσεων και των αντικειμένων τους. Χαρακτηριστικό του είναι ότι ανταποκρίνεται σε οντολογίες μεγάλου βάθος δέντρου (οι οντολογίες μπορεί να περιγράφουν πρωτεΐνες που συνήθως περιγράφονται από παραπάνω των χιλίων αμινοξέα) αλλά το λεξιλόγιό που χρησιμοποιούν ως εξωτερική βάση γνώσης είναι καλά καθορισμένο και σχετικά ορισμένου πλήθους (καθώς οι μεταλλάξεις δημιουργούν καινούρια αμινοξέα). Το πρόγραμμα μπορεί να βρεθεί στη σελίδα http://www.geneontology.org/GO.tools.shtml.

Τέλος ένα πολύ καλό πρόγραμμα για δημιουργία διαχείριση και συσχέτιση οντολογιών είναι το Hcone (Human Centered Ontology Enviroment), ένα πρόγραμμα που δημιουργήθηκε ως διδακτορική διατριβή του κυρίου Κότη Κωνσταντίνου στο πανεπιστήμιο Αιγαίου. Στο κομμάτι της συγχώνευσης οντολογιών χρησιμοποιούνται πολλοί αλγόριθμοι για αναζήτηση κοινών στοιχείων αλλά η συγχώνευση γίνεται αυτόματα με απόφαση μόνο του

προγράμματος και όχι με την παρέμβαση του χρήστη. Η ιστοσελίδα του προγράμματος είναι η *http://www.icsd.aegean.gr/kotis/hconeWeb/* .



Σχήμα 32: το περιβάλλον του Hcone.

2.2 Ανάλυση απαιτήσεων του παρόντος προγράμματος

Σε αυτό το βήμα ο προγραμματιστής καλείται να κατανοήσει, να καταγράψει και να κατηγοριοποιήσει τις απαιτήσεις του χρήστη. Οι απαιτήσεις χωρίζονται σε λειτουργικές, δηλαδή σε απαιτήσεις που αν δεν καλύπτονται το πρόγραμμα είναι ελλιπές, και σε μη λειτουργικές, που είναι αυτές που ακόμη κι αν παραληφθούν το πρόγραμμα θα υποστηρίζει τη λειτουργικότητά του, αλλά ο χρήστης δε θα είναι απόλυτα ευχαριστημένος. Η ανάλυση των απαιτήσεων είναι το πιο σημαντικό βήμα για να καθοριστεί η χρησιμότητα και η ποιότητα ενός προγράμματος, αλλά και για να εντοπιστούν λειτουργικά λάθη και να γίνει προσπάθεια επέκτασή του αν μεγαλώσουν οι απαιτήσεις. Οι απαιτήσεις δίνονται από το χρήστη και καταγράφονται από το σχεδιαστή του λογισμικού, ο οποίος ξεχωρίζει τις απαιτήσεις, απορρίπτει τις περιττές ή όμοιες, και τις ιεραρχεί. Ο χρήστης μπορεί να ανήκει σε οποιαδήποτε κατηγορία, για παράδειγμα μπορεί να είναι ένας χρήστης μιας ιστοσελίδας, μια μεγάλη εταιρία που είναι πελάτης. Σε αυτή την περίπτωση οι χρήστες είναι πολλοί, και χωρίζονται σε πολλές ομάδες πάνω στο διαδίκτυο. Αυτή η κατανομή των χρηστών κάνει πιο δύσκολο τον καθορισμό των απαιτήσεων, παρόλα αυτά, οι δημιουργοί οντολογιών μετά από συνεργασία μπορούν να καθορίσουν τις απαιτήσεις ενός τέτοιου προγράμματος, και τα πρότυπα (standards) που υπάρχουν καθορίζουν τις κινήσεις των προγραμματιστών. Παρόλα αυτά όμως οι απαιτήσεις δε μπορούν να καλύπτουν όλους τους δημιουργούς οντολογιών(για παράδειγμα μπορεί να μη μπορούν να υποστηριχθούν όλες οι γλώσσες δημιουργίας οντολογιών), και κάποιες φορές μπορεί να μην είναι συγκεκριμένες.

Για την ανάλυση των απαιτήσεων αυτού του προγράμματος χρησιμοποιήθηκε το πρότυπο αρχείο της IEEE για την περιγραφή των απαιτήσεων, το οποίο επισυνάπτεται στο τέλος της εργασίας. Οι απαιτήσεις συλλέχθηκαν με βάση την εκφώνηση της πτυχιακής, τα υπάρχοντα προγράμματα που προαναφέρθηκαν, την εμβέλεια των γλωσσών δημιουργίας οντολογιών τόσο στην ευκολία άρα και στο γεγονός ότι είναι δημοφιλείς όσο και στη μεγάλη λειτουργικότητα, και στο γεγονός ότι αυτή η γλώσσα (owl) χρησιμοποιεί τους συντακτικούς κανόνες της xml. Επίσης η απαίτηση να έχει καλό ποσοστό απόκρισης, δηλαδή η οντολογία που δημιουργείται να ταιριάζει αρκετά με αυτή που είχε στο μυαλό του ο χρήστης, οδηγεί στην απαίτηση το πρόγραμμα να είναι user-based, δηλαδή το πρόγραμμα να προτείνει και ο χρήστης να παίρνει την τελική απόφαση.

2.3 Περιγραφή της ιδέας

Με βάση τις απαιτήσεις του προγράμματος, μπορεί να χτιστεί βήμα-βήμα η δομή του. Πρώτο βήμα είναι ο καθορισμός της εισόδου. Το πρόγραμμα πρέπει να δέχεται από το χρήστη δυο αρχεία, να τα διαβάζει και να αποθηκεύει τα δεδομένα σε μια δομή δεδομένων για σύγκριση και περεταίρω επεξεργασία. Ο κώδικας του προγράμματος θα είναι σε java ενώ τα αρχεία πρέπει να είναι γραμμένα βάση του πρωτοκόλλου της owl και της rdfs και σε σύνταξη xml, επειδή οι γλώσσες αυτές είναι εύχρηστες, διαδεδομένες και δίνουν περισσότερες περιγραφικές δυνατότητες από τις άλλες, όπως περιγράφτηκε και παραπάνω. Για το διάβασμα του αρχείου θα χρησιμοποιηθεί ο dom parser, που διευκολύνει την εξαγωγή των δεδομένων για xml αρχεία. Επίσης τα αρχεία υποτίθεται ότι περιγράφουν από την πιο γενική στις πιο ειδικές κλάσεις, οπότε μια ακολουθιακή ανάγνωση γίνεται άμεσα κατά πλάτος αναζήτηση του υποτιθέμενου οντολογικού γράφου.

Έπειτα η βασική διεργασία του προγράμματος αποτελείται από κράμα αλγόριθμων οι οποίοι βρίσκουν κοινά ανάμεσα στις οντολογικές κλάσεις. Αυτοί οι αλγόριθμοι πρέπει να είναι δύο ειδών, αλγόριθμοι που συγκρίνουν ονόματα και αλγόριθμοι που συγκρίνουν κόμβους. Επειδή ο χρήστης επεμβαίνει στη διαδικασία εξαγωγής συμπερασμάτων, οι αλγόριθμοι πρέπει να βρίσκουν κοινά στοιχεία κλάσεων πολύ συντηρητικά, και να ενώνουν μόνο όσες κλάσεις είναι σίγουρα όμοιες και όσες τους επιτρέπει ο χρήστης.

Μετά την εξαγωγή των συμπερασμάτων για την ομοιότητα των κλάσεων μένει να δημιουργηθεί ένα αρχείο στην ίδια μορφή με τα αρχεία εισόδου, το οποίο θα περιγράφει το συγκερασμό των δύο οντολογιών. Ένα πρόβλημα που προκύπτει με τα αρχεία είναι ποιο από τα δύο ονόματα των οντολογιών είναι το πιο γενικό για να χρησιμοποιηθεί. Οπότε μια λύση είναι να ερωτηθεί ο χρήστης, ή να δοθεί ως όνομα το όνομα της πρώτης κλάσης. Στο παρόν πρόγραμμα θα θεωρηθεί η σύμβαση ότι το πρώτο αρχείο είναι και το πιο γενικό οπότε το όνομα της οντολογίας θα βρεθεί από το αυτό..

Τέλος θα ήταν χρήσιμο για το χρήστη να δει γραφικά το αποτέλεσμα του προγράμματος, αντί να διαβάζει μια μεγάλη σε μέγεθος οντολογική περιγραφή. Οπότε στο γραφικό περιβάλλον ο χρήστης θα ήταν θεμιτό να μπορεί να δει σχηματικά , και όχι σε αρχείο xml, και τα τρία αρχεία. Έπειτα θα ήταν χρήσιμο για αυτόν να μπορεί να επέμβει σε αυτά διαγράφοντας, δημιουργώντας ή μετονομάζοντας κόμβους. Να σημειωθεί ότι στο αποτέλεσμα του προγράμματος δεν αλλάζουν τα αρχεία εισόδου, αλλά ο χρήστης πριν προβεί στη συνένωση μπορεί να αλλάξει παροδικά τα οντολογικά δέντρα, και τα αποτελέσματα να φαίνονται στο τελικό δέντρο και στο αρχείο.

2.4 Περιορισμοί του προγράμματος

Οι περιορισμοί του συγκεκριμένου προγράμματος αφορούν κυρίως τα αρχεία εισόδου, καθώς πάνω σε αυτά στηρίζεται το πρόγραμμα για την εξαγωγή αποτελέσματος. Τα αρχεία εισόδου πρέπει να έχουν ακριβώς το ίδιο format, και να είναι σωστά γραμμένα καθώς δεν γίνεται έλεγχος για λάθη. Σε αυτό το σημείο πρέπει να δηλωθεί ότι τα αρχεία εισόδου που χρησιμοποιήθηκαν για τον έλεγχο ορθότητας και ποιότητας του προγράμματος είναι αρχεία ανοιχτού κώδικα που παρέχει το protégé, είναι γραμμένα σε γλώσσα owl / rdfs, και έχουν παρόμοιο τρόπο γραφής.

Το όνομα της οντολογίας και της κλάσης πρέπει να περιγράφονται βε βάση τους κανόνες του rdfs, όπως παρακάτω:

<owl:Ontology rdf:about= ...

<owl:Class rdf:about= ...

Για παράδειγμα:

<owl:Ontology rdf:about="&ns_transport;TransportOntology">

<owl:Class rdf:about="&ns_transport;Transport">

Επίσης είναι καλό τα αρχεία να είναι οργανωμένα σε < ontologies, classes, properties, objects > γιατί οι αλγόριθμοι ενώνουν μόνο στοιχεία που είναι όμοια και ανήκουν και στην ίδια ομάδα. Για παράδειγμα, αν δύο στοιχεία είναι κοινά, αλλά το ένα ανήκει σε class και το άλλο σε property, δεν τα προτείνει για ένωση, καθώς δεν είναι ορθό να ενώνεται μια κλάση με μια ιδιότητα αυτής.

Η εκδοχή και η κωδικοποίηση του xml αρχείου πρέπει να είναι η παρακάτω:

<?xml version="1.0" encoding="UTF-8"?>

Ακόμη, για να ξεχωρίζουν τα απλά πεδία της xml από αυτά που η γλώσσα επιτρέπει να έχουν συγκεκριμένο νόημα, πρέπει τα αρχεία να ορίζουν τις παρακάτω οντότητες, ώστε να μπορεί να χρησιμοποιεί και τις δυνατότητες των γλωσσών αυτών:

```
<rdf:RDF
    xmlns:xsd="&xsd;"
    xmlns:rdf="&rdf;"
    xmlns:rdfs="&rdfs;"
    xmlns:owl="&owl;"
    xmlns:ns_transport="&ns_transport;"
>
</rdf:RDF>
```

3 Μέρος 3ο: Εγχειρίδιο χρήσης του πρότυπου προγράμματος σύμπτυξης οντολογιών περιγραφόμενων σε γλώσσα owl.

Εισαγωγή

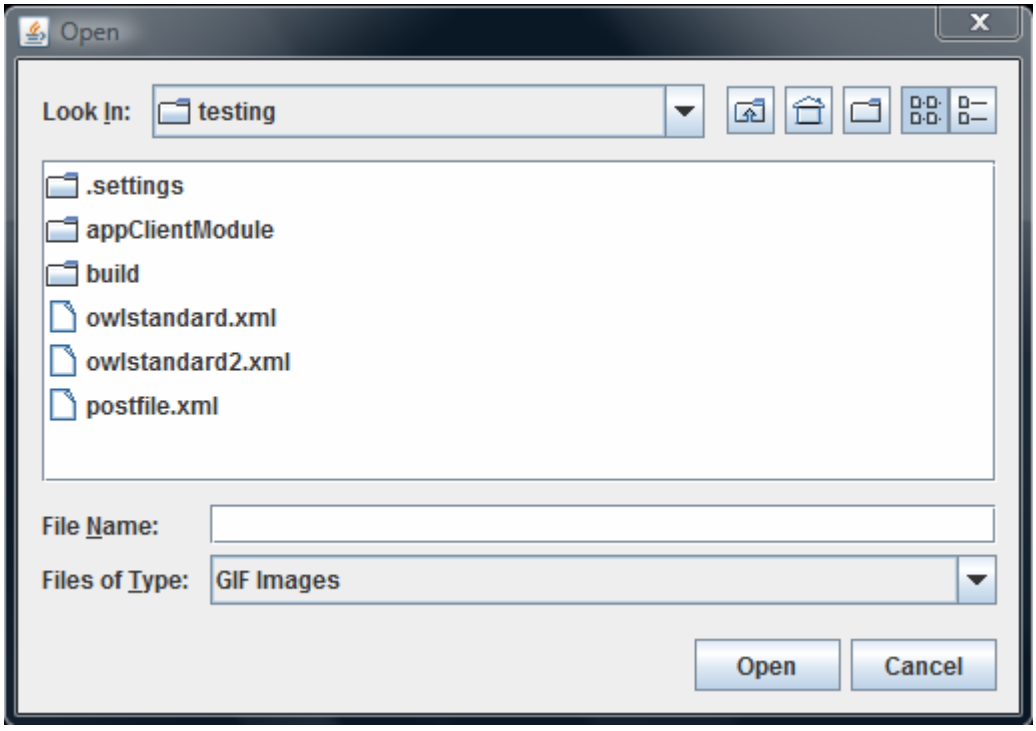
Όπως προαναφέρθηκε, το πρόγραμμα που παρουσιάζεται είναι μια java εφαρμογή, για την ανάπτυξή της οποίας χρησιμοποιήθηκε το πρόγραμμα eclipse, ένα πρόγραμμα ανοικτού κώδικα που μπορεί να βρεθεί στον ιστοχώρο *http://www.eclipse.org/* . Το eclipse δημιουργήθηκε από τη IBM το Νοέμβρη του 2001 και υποστηρίχθηκε από μια κοινοπραξία προμηθευτών λογισμικού. Το ίδρυμα του eclipse δημιουργήθηκε τον Ιανουάριο του 2004 ως ανεξάρτητη μη κερδοσκοπική εταιρία για να ενεργήσει ως διαχειριστής της κοινότητας eclipse.

Αυτό που το πρόγραμμα κάνει σε γενικές γραμμές είναι να δέχεται δύο αρχεία συγκεκριμένης μορφής, και να προσπαθεί να τα ενώσει, ψάχνοντας για κοινά στοιχεία. Είναι χρήσιμο για ανθρώπους που ξέρουν xml και owl, και θέλουν να ενώσουν δύο οντολογικά σχήματα.

Δομικά, το πρόγραμμα, έχει ένα προστάδιο επιλογής αρχείων, τρία στάδια της κυρίως εργασίας, και μια έξοδο, που είναι ένα xml αρχείο που περιγράφει μια οντολογία. Με περισσότερα λόγια, στην αρχή του προγράμματος ο χρήστης καλείται να ανοίξει τα δύο αρχεία xml που είναι οι αρχικές οντολογίες, και από εκείνη τη στιγμή ξεκινά το κυρίως πρόγραμμα. Το κυρίως πρόγραμμα έχει τρία στάδια, τα οποία πρέπει να περάσει ο χρήστης για να φτάσει στο επιθυμητό αποτέλεσμα, το στάδιο στο οποίο βλέπει και μπορεί να βελτιώσει τα δέντρα, το στάδιο που καθοδηγεί το πρόγραμμα για τη συνένωση των οντολογιών, και το σημείο όπου και δημιουργείται το τελικό αρχείο με την καινούρια οντολογία.

3.1 Εισαγωγή αρχείων

Στο πρώτο στάδιο του προγράμματος εμφανίζονται δυο παράθυρα το ένα μετά το άλλο στα οποία ο χρήστης επιλέγει τα αρχεία εισόδου, δυο αρχεία .xml τα οποία υπάρχουν τοπικά στο δίσκο του χρήστη. Με αυτή την επιλογή φορτώνονται τα δεδομένα τους στο πρόγραμμα ώστε να αρχίσει η διαδικασία συνένωσης. Αν δεν ανοιχτούν δύο αρχεία αλλά ένα τότε η διαδικασία σταματά εκεί, ενώ αν έστω ένα από τα αρχεία δεν είναι .xml ή .owl τότε η διαδικασία σταματά δημιουργώντας exception, επειδή δε μπορεί να βρει δεδομένα του είδους που ψάχνει. Να τονιστεί σε αυτό το σημείο ότι το συγκεκριμένο γραφικό περιβάλλον διεπαφής με το χρήστη που χρησιμοποιήθηκε, δίνεται έτοιμο από τη γλώσσα με την κλάση JFileChooser της java.



Σχήμα 33: το περιβάλλον εργασίας του προγράμματος που δημιουργήθηκε στην παρούσα πτυχιακή. Το πρώτο παράθυρο εισαγωγής αρχείου

Ως αποτέλεσμα των παραπάνω, αν επιλεγούν τα σωστά αρχεία στα δυο παράθυρα που θα ανοίξουν , τότε θα ανοίξει το κυρίως παράθυρο το οποίο φαίνεται παρακάτω.

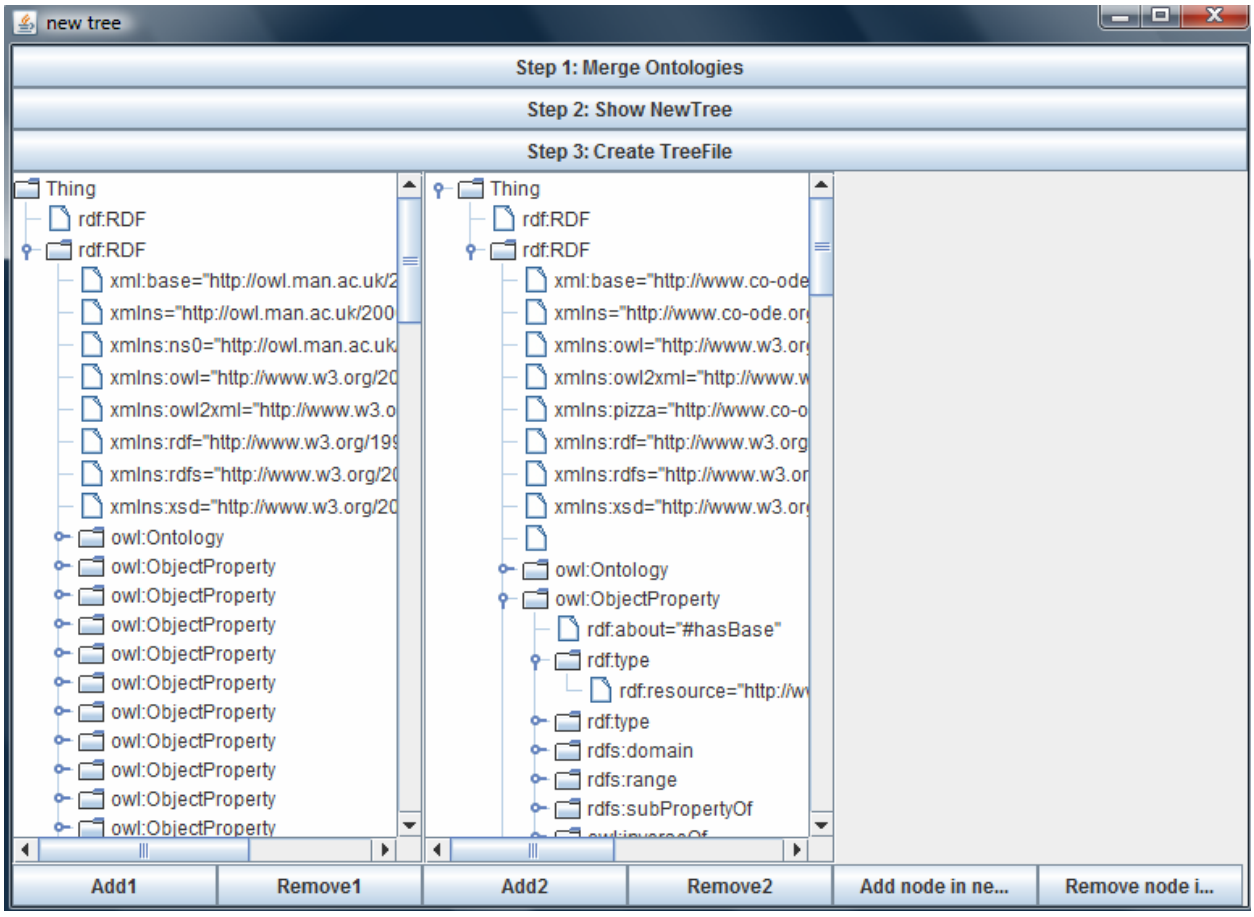
Οπτικοποίηση και επέμβαση του χρήστη στα δέντρα

Στο κυρίως αρχείο φαίνονται γραφικά, και με μορφή δέντρου σύμφωνη με τον τρόπο συγγραφής του αρχείου, οι οντολογίες που περιγράφονται στα αρχεία xml. Πριν προχωρήσει ο χρήστης στη συνένωση των αρχείων μπορεί να επέμβει στα δέντρα ως εξής:

προσθήκη κόμβου: επιλέγοντας το μονοπάτι που θέλει κάποιος και πατώντας το κουμπί ADD 1 για το πρώτο δέντρο και ADD2 για το δεύτερο.

διαγραφή κόμβου: επιλέγοντας το μονοπάτι που θέλει κάποιος και πατώντας το κουμπί REMOVE 1 για το πρώτο δέντρο και REMOVE 2 για το δεύτερο. Μαζί με τον κόμβο διαγράφονται και τα παιδιά του, αν αυτός δεν είναι φύλλο.

αλλαγή του ονόματος ενός κόμβου: ο χρήστης μετά από μια προσθήκη κόμβου ή επειδή προτιμά ένα πιο αντιπροσωπευτικό όνομα για ένα κόμβο, μπορεί κάνοντας διπλό κλικ πάνω στον κόμβο μπορεί να τον μετονομάσει. Σημείωση: Η αλλαγή είναι προσωρινή και χρησιμοποιείται μόνο στη δημιουργία του τελικού αρχείου όπως και για το οπτικό αποτέλεσμα, συνεπώς οι αλλαγές δε φαίνονται στα αρχεία εισόδου.

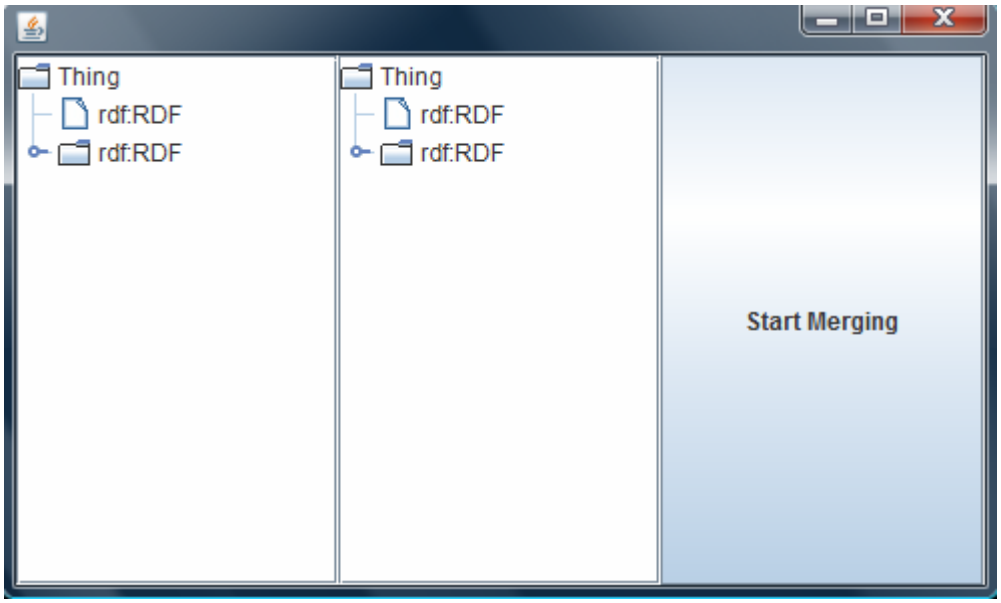


Σχήμα 34: το βασικό παράθυρο εργασίας του χρήστη

Αφού ο χρήστης διαμορφώσει όπως θέλει τα αρχεία, μπορεί να ξεκινήσει η διαδικασία συνένωσης των σχημάτων. Αυτή η διαδικασία χωρίζεται σε τρία βήματα.

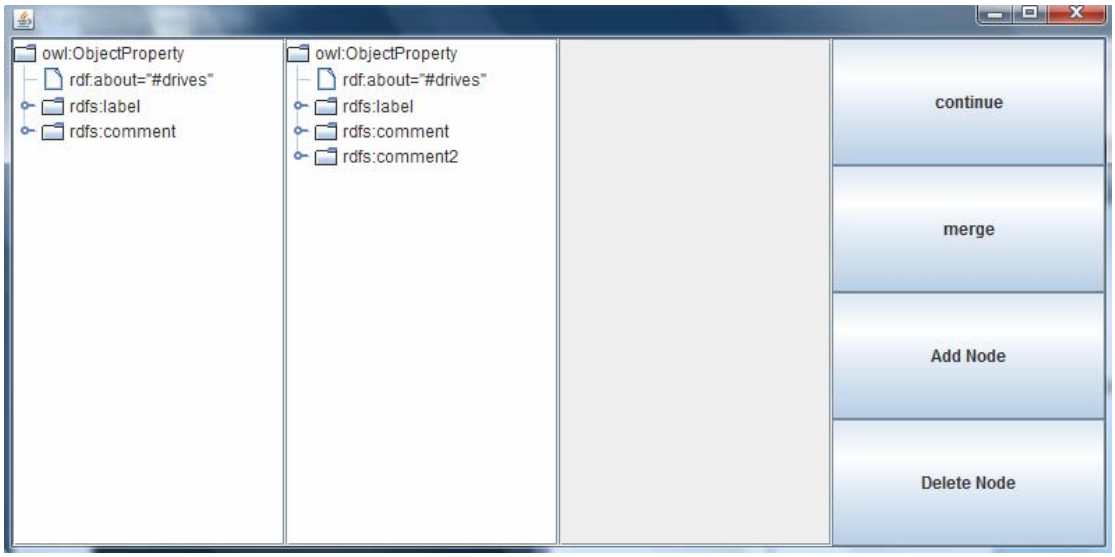
3.2 Βημα1: Απόφαση για συνένωση κόμβων

πατώντας το κουμπί «Step 1» εμφανίζεται το παρακάτω παράθυρο, που δείχνει τις δυο οντολογίες που πρόκειται να συνενωθούν και το κουμπί εκκίνησης «Start Merging» .



Σχήμα 35: το παράθυρο εκκίνησης της συγχώνευσης οντολογιών

Πατώντας το κουμπί εκκίνησης εμφανίζεται το παρακάτω παράθυρο, στο οποίο το πρόγραμμα προτείνει πιθανά σημεία ομοιότητας, και ο χρήστης καλείται να αποφασίσει αν θέλει και πως θέλει τα δυο σχήματα να ενωθούν.

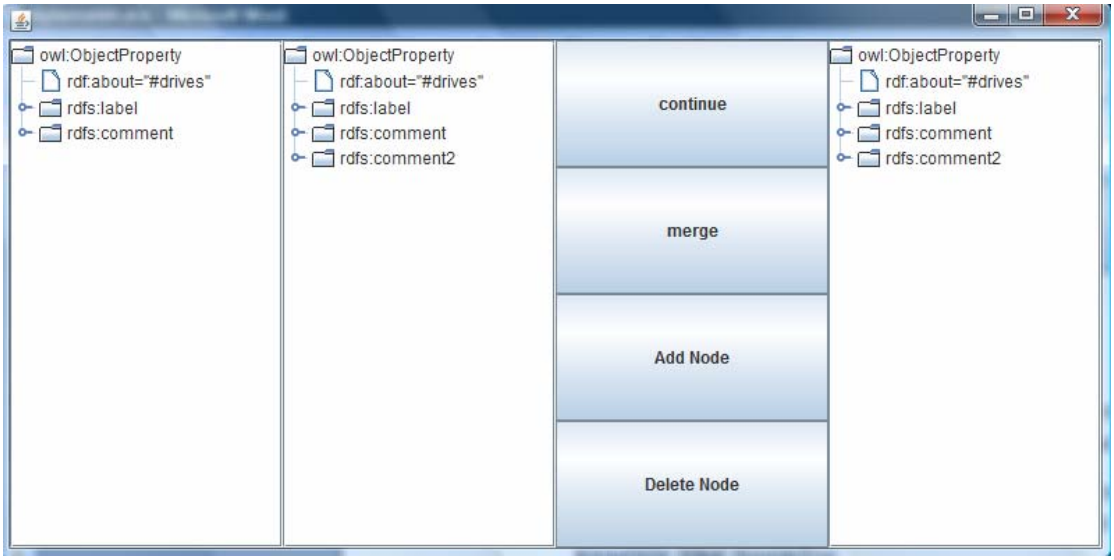


Σχήμα 36: το παράθυρο απόφασης συνένωσης δύο κόμβων

Τα κουμπιά:

πατώντας το κουμπί «continue» ο χρήστης θα δει το επόμενο παράθυρο όπου το πρόγραμμα θα προτείνει άλλο σημείο συνένωσης.

Πατώντας το κουμπί «merge» τα δύο υποδέντρα θα ενωθούν αυτόματα από το πρόγραμμα. Το καινούριο δέντρο θα φανεί δεξιά των κουμπιών όπως παρακάτω:



Σχήμα 37: το παράθυρο απόφασης συνένωσης δύο κόμβων, αφού έχει πατηθεί το κουμπί merge.

Το κουμπί «add node» λειτουργεί μόνο εφόσον έχει πατηθεί το κουμπί «merge» πιο πριν. Αν ο χρήστης δεν είναι απόλυτα ικανοποιημένος από το αποτέλεσμα του προγράμματος ως προς το νέο δέντρο, τότε μπορεί να επέμβει ενθέτοντας νέους κόμβους και μετονομάζοντας τους (διπλό κλικ) όπως και μετονομάζοντας τους ήδη υπάρχοντες.

Το κουμπί «delete node» διαγράφει ένα κόμβο και τα παιδιά του εφόσον αυτός έχει, έχει επίσης την ίδια χρηστικότητα και περιορισμούς με τον κόμβο «add node».

Εφόσον ένας χρήστης πατήσει το κουμπί συνένωσης και αλλάξει το νέο δέντρο, μπορεί να συνεχίσει τη διαδικασία πατώντας «continue». Έτσι να αναδύεται καινούριο παράθυρο εάν υπάρχει και νέο σημείο συνένωσης και η διαδικασία είναι η ίδια, αλλιώς ο χρήστης περνά στο βήμα 2.

Ακόμη μένει να σημειωθεί ποιοι από τους αλγόριθμους σύμπτυξης οντολογικών σχημάτων χρησιμοποιούνται από το πρόγραμμα για να βρεθούν

όμοια υποδέντρα. Αυτοί όπως και παραπάνω ειπώθηκε χωρίζονται σε αλγόριθμους κόμβων και αλγόριθμους λεξικού.

Αλγόριθμοι λεξικού: αρχικά όλοι οι κόμβοι συγκρίνονται μεταξύ του ως προς τη λεξικογραφική τους ομοιότητα. Η java δίνει κάποιες συναρτήσεις σύγκρισης αλφαριθμητικών που ταιριάζουν με τους αλγόριθμους που αναφέρθηκαν στο πρώτο μέρος της εργασίας και είναι οι παρακάτω:

.equals, η οποία απαντά με μηδέν αν δύο αλφαριθμητικά είναι όμοια.

.startWith, η οποία απαντά με true κάποιο αλφαριθμητικό είναι η αρχή ενός δεύτερου, πχ sub και subtitle.

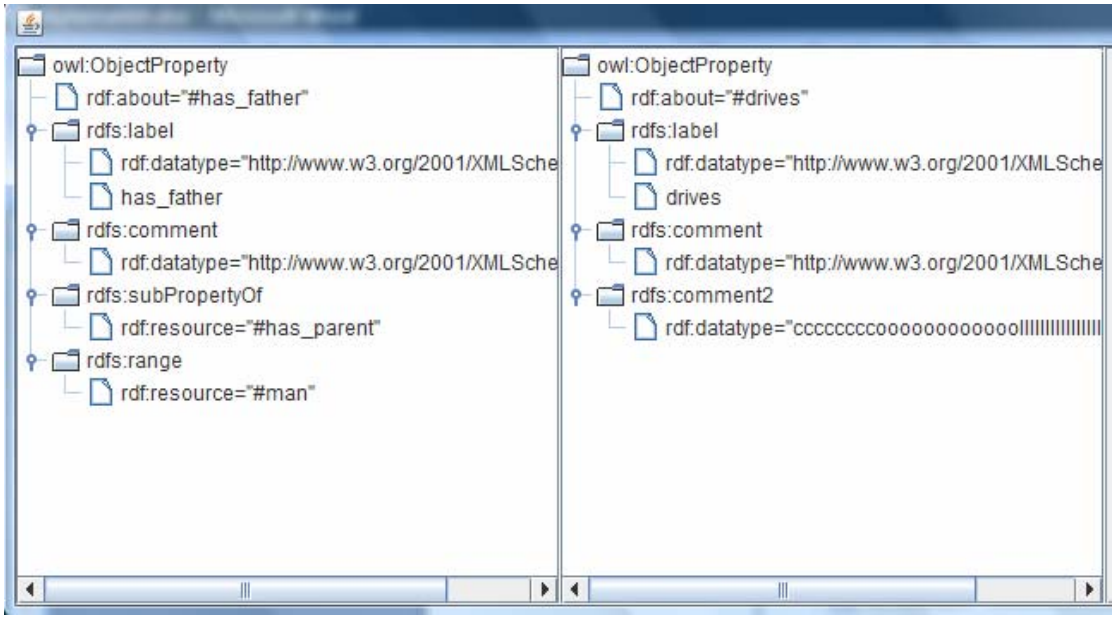
.endsWith, η οποία απαντά με true κάποιο αλφαριθμητικό είναι το τέλος ενός δεύτερου, πχ width και bandwidth.

.contains, αν το ένα αλφαριθμητικό περιέχει ολόκληρο το άλλο, πχ for unfortunately.

Αυτό που φαίνεται και παραπάνω είναι ότι με μόνο αυτές τις τεχνικές καταρχάς τα συνώνυμα δε μπορούν να εντοπιστούν, και κατά δεύτερον μπορεί να προταθούν δύο υποδέντρα ως πιθανόν όμοια τα οποία δεν έχουν καμία σχέση. Το γεγονός όμως ότι ο χρήστης έχει τον τελευταίο λόγο οδηγεί στο συμπέρασμα ότι το πρόγραμμα χρειάζεται να βρει όσες περισσότερες (σε πλαίσια για να μη χαθεί η χρηστικότητα του προγράμματος) πιθανές ενώσεις γίνεται. Για παράδειγμα, αν οι πραγματικές ομοιότητες είναι 20 και το πρόγραμμα δίνει 10, χάνονται σίγουρα οι 10 πιθανές. Αν όμως οι πραγματικές ομοιότητες είναι 20 και το πρόγραμμα δίνει 50, ο χρήστης μπορεί να δει και παραπάνω από δέκα σωστές, δεδομένου ότι μπορεί πάλι να μην εντοπιστούν όλες οι σωστές, ενώ αν δίνει 100 ο χρήστης θα κουραστεί να βρίσκει σημεία ένωσης καθώς είναι χρονοβόρο και κουραστικό.

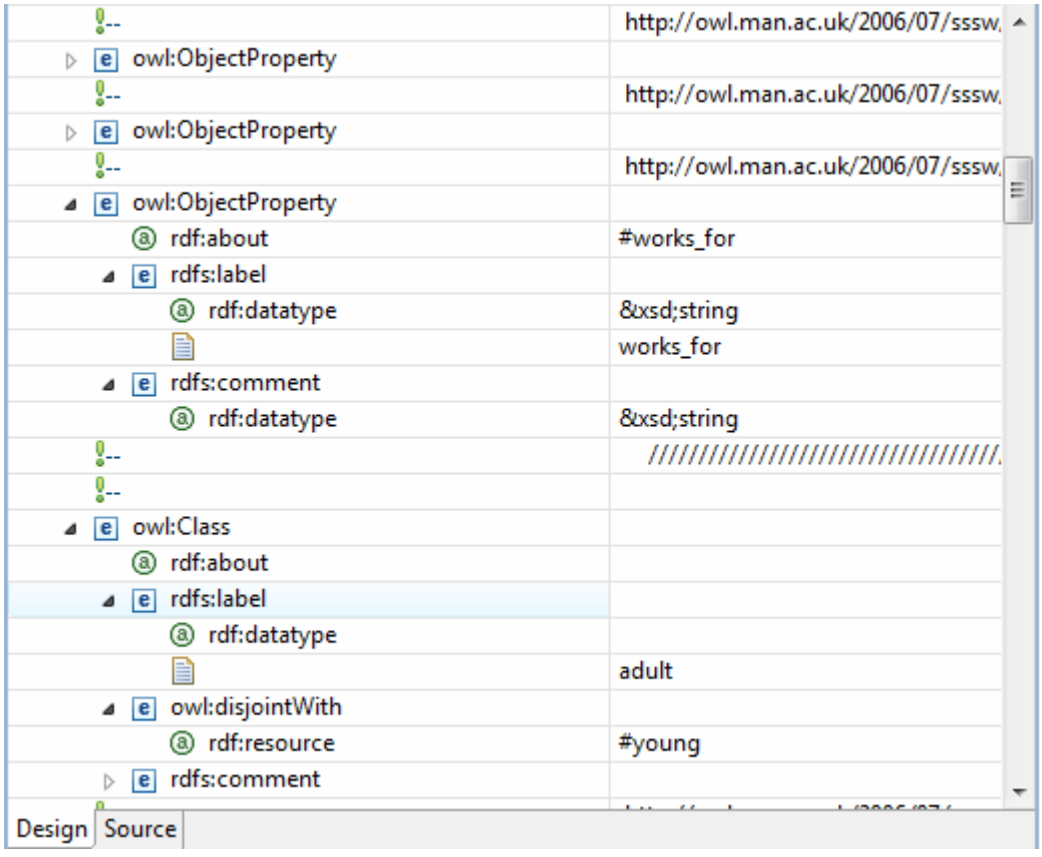
Αλγόριθμοι κόμβων: αν με τους λεξικογραφικούς αλγόριθμους δύο κόμβοι βρεθούν ως όμοιοι χρησιμοποιούνται δύο περιορισμοί κόμβων ώστε να καθοριστεί αν είναι μεγάλη η πιθανότητα ένωσης και αν γίνεται τελικά τα δύο υποδέντρα να ενωθούν, και είναι οι παρακάτω:

Τα όμοια στοιχεία πρέπει να είναι φύλλα. Αυτός ο περιορισμός κατανοείται αν παρατηρηθούν τα υποδέντρα προς ένωση. Σχεδόν όλα τα δέντρα που περιέχουν το στοιχείο rdf:Comment, όμως δεν έχουν κανένα σημείο σχέσης. Το θέμα είναι ότι στους ενδιάμεσους κόμβους υπάρχουν εντολές/ μεταβλητές/ συναρτήσεις ενώ στα φύλλα υπάρχουν οι τιμές. Αυτό φαίνεται γραφικά στο παράδειγμα παρακάτω:

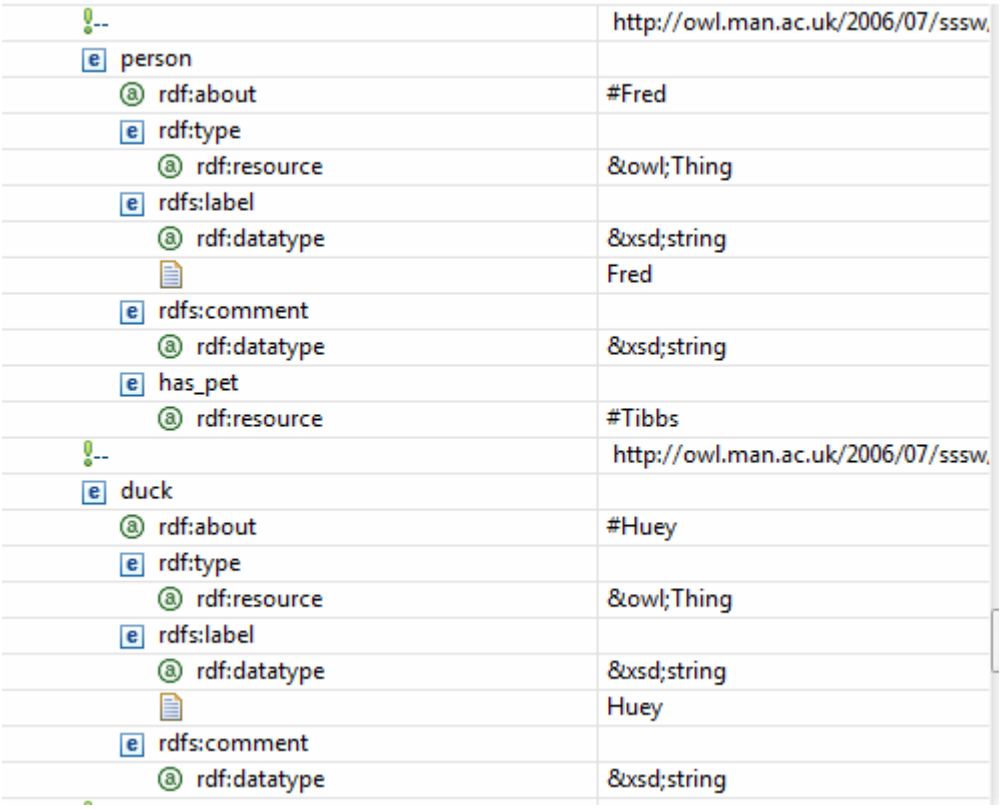


Σχήμα 38: δύο κόμβοι με όμοιο όνομα αλλά με διαφορετικά χαρακτηριστικά

Τα υποδέντρα πρέπει να έχουν όμοιες ρίζες. Η αναφορά δε γίνεται για την αρχική ρίζα των δύο οντολογιών (Thing) αλλά ένα σημείο παρακάτω, όπου και περιγράφεται αν το υποδέντρο περιγράφει κλάση ιδιότητα ή αντικείμενο.



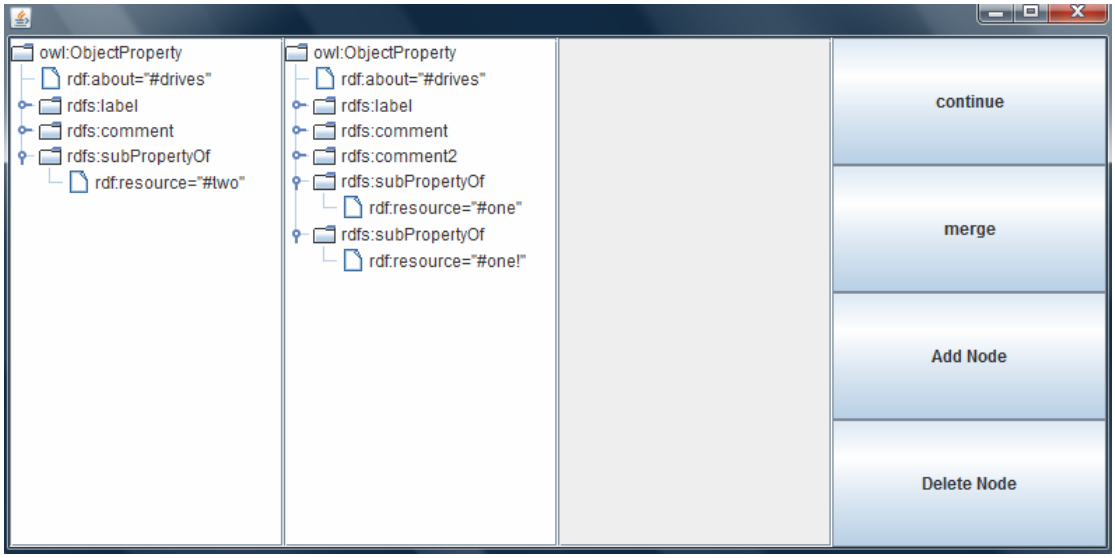
Σχήμα 39: τα στοιχεία μιας οντολογίας όπως φαίνονται στο αρχείο εισόδου xml



Σχήμα 34: φαίνονται δύο αντικείμενα από το xml αρχείο.

Περιορισμοί που εισάγει ο χρήστης για τις εντολές subClassOf και subPropertyOf

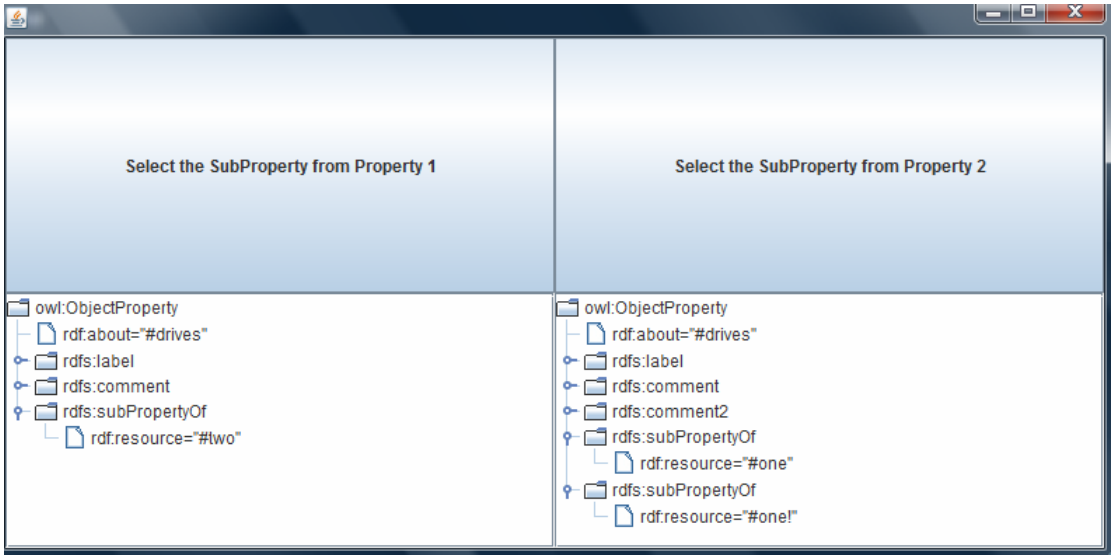
Κατά την ένωση των κόμβων υπάρχει η περίπτωση να θέτονται σε σύγκριση δύο κλάσεις που η καθεμία είναι υποκλάση διαφορετικής κλάσης, ή αντίστοιχα δύο ιδιότητες που είναι παιδιά διαφορετικών ιδιοτήτων. Σε αυτό το σημείο τίθεται το ερώτημα εάν η νέα συμπτυγμένη μορφή θα είναι υποκλάση της κλάσης στο πρώτο αρχείο, ή στο δεύτερο. Ένα παράδειγμα δύο κόμβων με διαφορετικό πατέρα που δύνανται να υποβληθούν σε σύμπτυξη φαίνεται παρακάτω:



Σχήμα 41: δύο όμοιες ιδιότητες, με διαφορετικό πατέρα

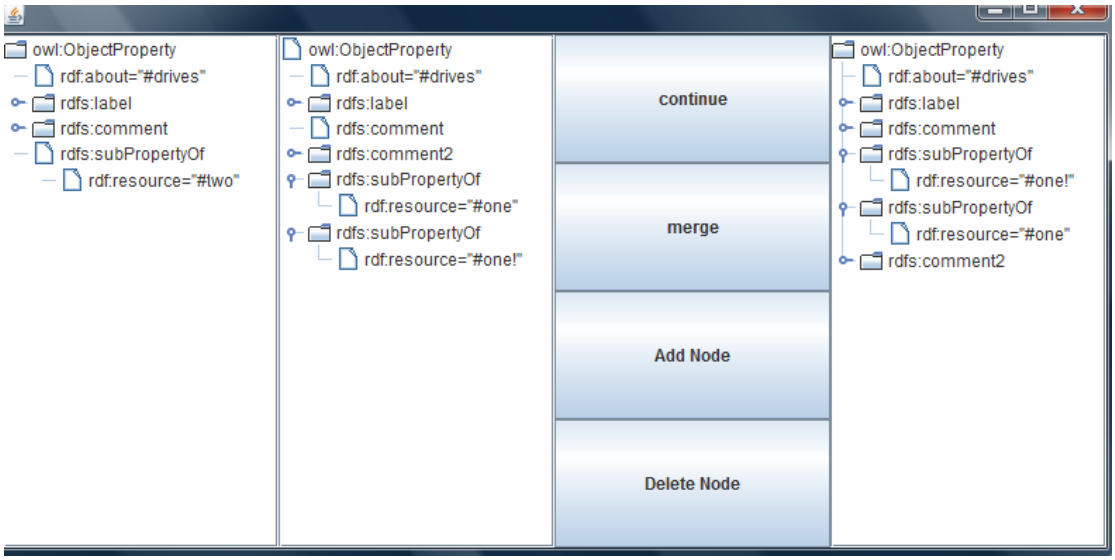
Στο παραπάνω σχήμα φαίνονται δυο όμοιες ιδιότητες, και όπως κ εδώ συμβαίνει και κάποιες υποκλάσεις να έχουν παραπάνω από έναν πατέρα (να σημειωθεί ότι τα αρχεία που χρησιμοποιούνται είναι πρότυπα αρχεία του προγράμματος protege). Αν το πρόγραμμα αποφασίζει αυτοτελώς σε πιο κόμβο θα αναθέσει το κάθε παιδί, μπορεί το αποτέλεσμα να μην είναι ικανοποιητικό για το χρήστη καθώς το σφάλμα είναι αρκετά μεγάλο. Επομένως αυτό είναι το πιο σημαντικό σημείο στο οποίο η απόφαση του χρήστη κρίνεται απαραίτητη.

Με βάση τα παραπάνω, όταν ο χρήστης πατήσει merge στο παραπάνω παράθυρο, ανοίγει το παρακάτω παράθυρο στο οποίο ο χρήστης αποφασίζει από ποια κλάση θα «κληρονομηθεί ο πατέρας».



Σχήμα 42: το παράθυρο στο οποίο ο χρήστης επιλέγει ποιον πατέρα θα έχει η νέα ιδιότητα

Και παρακάτω φαίνεται το αποτέλεσμα της ένωσης αφού έχει πατηθεί το κουμπί «select subPropertyOf from property 2»:



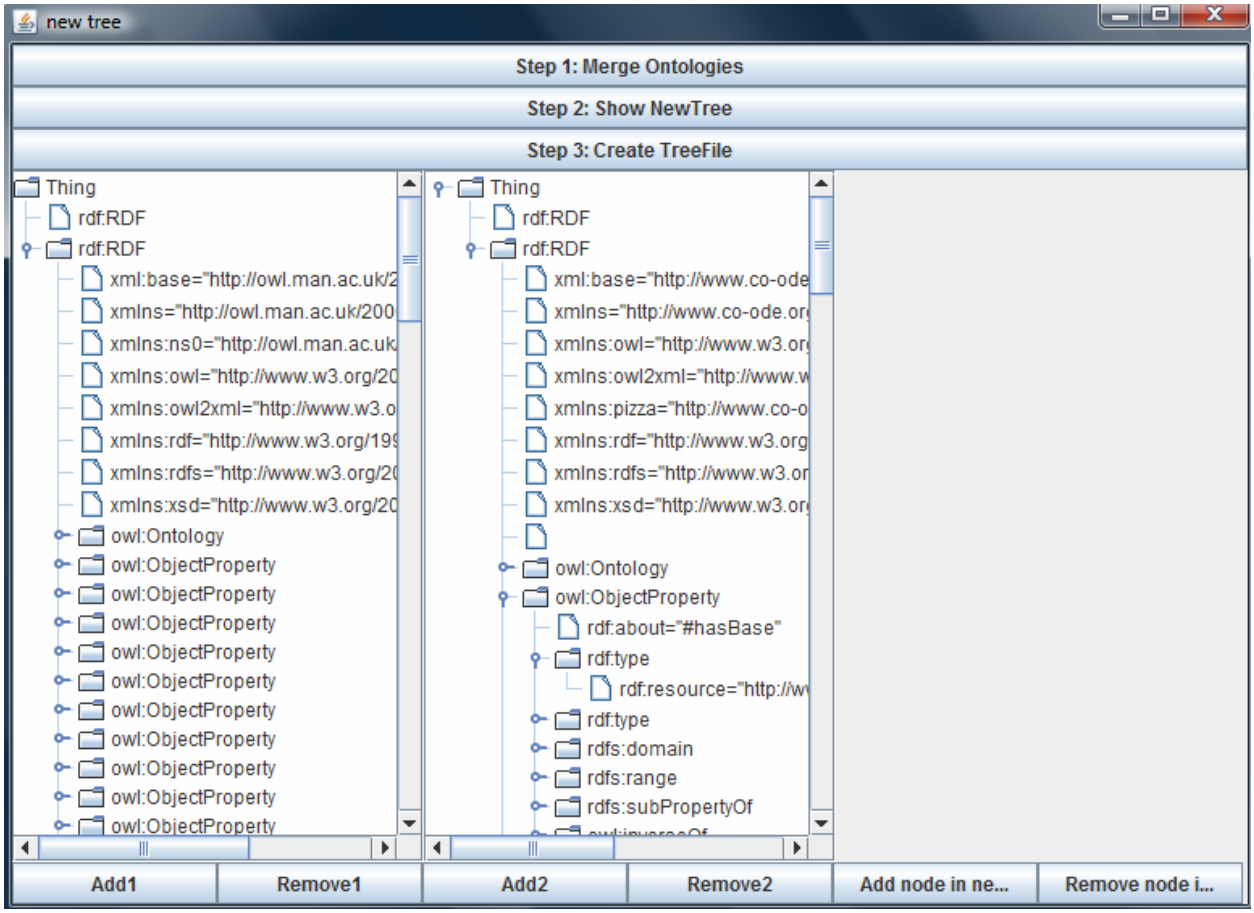
Σχήμα 43: φαίνεται η νέα ιδιότητα και ο πατέρας που ο χρήστης έχει επιλεξει

Έπειτα ο χρήστης αλλάζει το αποτέλεσμα αν δεν είναι ικανοποιημένος με τον τρόπο που αναφέρθηκε παραπάνω και πατάει «continue» για να περάσει στους επόμενους προς ένωση κόμβους.

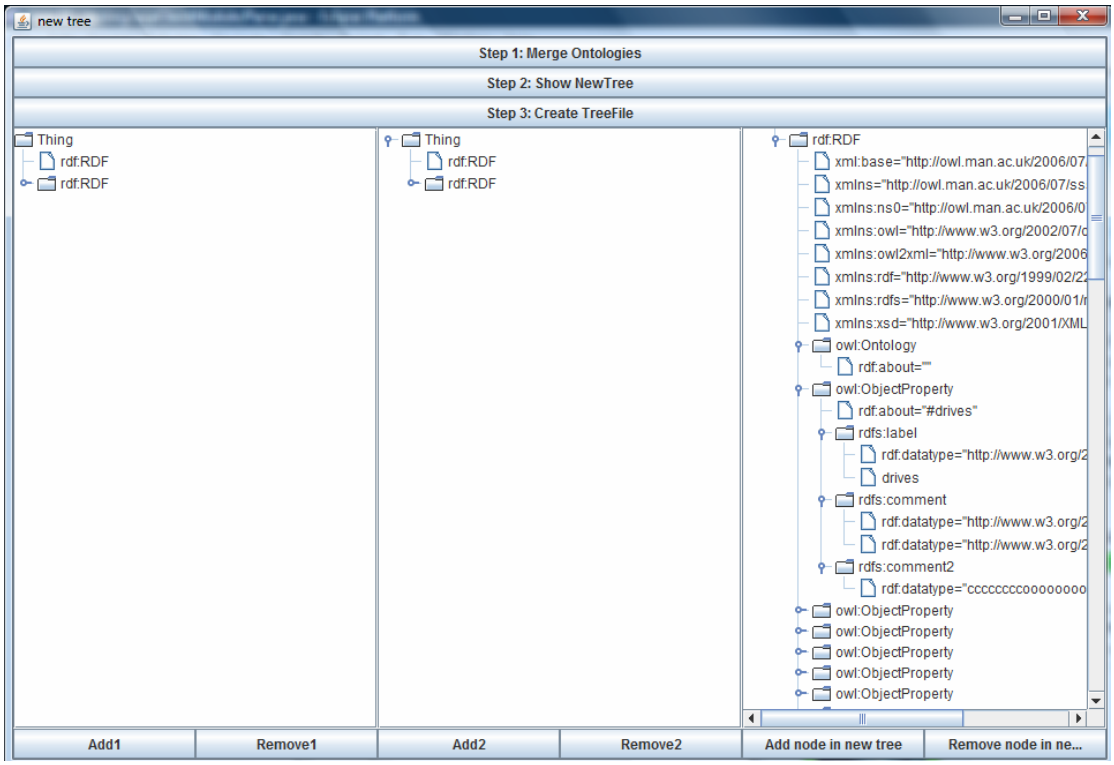
Μετά το τέλος των προτάσεων του προγράμματος για ένωση κόμβων, οι κόμβοι που βρέθηκαν διαφορετικοί περνάν αυτόματα από το δεύτερο δέντρο στο πρώτο απαράλλαχτοι. Οι μόνοι κόμβοι που δεν περνάν είναι αυτοί που περιγράφουν τις xml πληροφορίες του αρχείου καθώς και ο κόμβος που περιγράφει τη δεύτερη οντολογία και αυτό γιατί έχει υποτεθεί ότι η νέα οντολογία έχει το όνομα της πρώτης οντολογίας, όπως επίσης έχει υποτεθεί ότι το πρώτο αρχείο περιγράφει πιο γενική πληροφορία από το δεύτερο (η με άλλα λόγια ότι η δεύτερη οντολογία είναι πιθανό υποδέντρο του πρώτου).

3.3 Βήμα 2: Οπτικοποίηση νέου δέντρου

Σε αυτό το στάδιο έχει ήδη δημιουργηθεί το νέο δέντρο, με την καθοδήγηση του χρήστη, όμως ο χρήστης δεν το έχει δει ακόμη. Πατώντας το κουμπί «Step 2 : Show new tree» στο παρακάτω παράθυρο που είναι ακόμη ανοιχτό, εμφανίζεται το νέο δέντρο.



Σχήμα 44: το κύριο παράθυρο πριν την εμφάνιση του νέου δέντρου



Σχήμα 45: το κύριο παράθυρο μετά την εμφάνιση του νέου δέντρου

Αφού εμφανιστεί το δέντρο, ο χρήστης μπορεί να το αλλάξει, αν κάποια σημεία δεν του αρέσουν, με τα κουμπιά «Add node in new tree» και «remove node in new tree» και τη μετονομασία που είναι λειτουργία που υποστηρίζει η java για τα δέντρα και γίνεται με διπλό κλικ. Όταν ο χρήστης τελειώσει και με τη διαμόρφωση του δέντρου μπορεί να περάσει στο βήμα 3.

3.4 Βήμα 3: Δημιουργία αρχείου

Σε αυτό το στάδιο ο χρήστης είναι έτοιμος να δημιουργήσει το αρχείο xml που περιέχει την περιγραφή του νέου οντολογικού σχήματος που βλέπει γραφικά. Αυτό μπορεί να γίνει πατώντας το κουμπί «Step 3: Create TreeFile» που βρίσκεται στο κύριο παράθυρο, οπότε και δημιουργείται τοπικά στο δίσκο το αρχείο, στο σημείο που βρίσκεται και το πρόγραμμα, ένα αρχείο που ονομάζεται «postfile.xml» και περιγράφει την νέα οντολογία, η οποία έχει το όνομα της πρώτης οντολογίας που θεωρήθηκε ως η πιο γενική.

Λόγω της πολυμορφικότητας των xml αρχείων που περιγράφουν οντολογίες, είναι πιθανό να υπάρχουν λάθη σύνταξης που μπορεί να οφείλονται σε διαφορετικές εκδόσεις της xml, στην πολυπλοκότητα των αρχείων εισόδου, ή σε μεγάλη διαφορετικότητα στον τρόπο συγγραφής των αρχείων. Ο χρήστης πρέπει να διορθώσει «με το χέρι» τα λάθη επεμβαίνοντας στο τελικό αρχείο, ώστε να έχει το επιθυμητό αποτέλεσμα το οποίο ορίστηκε από την αρχή και είναι συνοπτικά ένα νέο αρχείο xml το οποίο περιέχει την σύμπτυξη δύο αρχείων xml τα οποία περιγράφουν οντολογίες.

4 Πρότυπο κείμενο ανάλυσης απαιτήσεων του παραπάνω λογισμικού

Εισαγωγή

Στο κείμενο αυτό περιγράφεται η δημιουργία του προγράμματος συγχώνευσης οντολογικών σχημάτων, ενός προγράμματος που δέχεται δυο αρχεία που περιγράφουν οντολογικά σχήματα και παράγει ένα αρχείο που να περιέχει όλα τα δομικά στοιχεία των δύο σχημάτων και να δείχνει τη σχέση των δυο οντολογικών σχημάτων.

Σκοπός

Η δημιουργία του προγράμματος συγχώνευσης οντολογικών σχημάτων πρέπει να δέχεται δυο αρχεία που περιγράφουν οντολογικά σχήματα σε συγκεκριμένη μορφή και να παράγει ένα αρχείο που να δείχνει τη σχέση των δυο οντολογικών σχημάτων, στην ίδια μορφή με τα προηγούμενα. Για την αποδοτικότερη εύρεση κοινών στοιχείων θα πρέπει να δέχεται την παρέμβαση του χρήστη.

Στόχος

Ο χρήστης θα πρέπει να μπορεί να δει τα κοινά σε δυο οντολογικά σχήματα, ή να κατανοήσει ότι δεν υπάρχει δυνατή σύνδεση μεταξύ τους. Το τελικό αρχείο θα περιγράφει ένα γράφο, που από τη συνεκτικότητά του θα αποφαίνεται αν οι δυο οντολογίες που εισήχθησαν είναι παρεμφερείς.

Επιδίωξη του χρήστη

Ένας χρήστης θα μπορεί με το πρόγραμμα να ενώνει οντολογικά σχήματα. Έτσι θα μπορεί να έχει πληρέστερη περιγραφή τους και να μεγαλώνει τη βάση δεδομένων του και τη βάση γνώσης του καθώς μπορεί να ενώνει μια παλιά πληροφορία με μια νέα , ή μια πληροφορία που δεν έχει αρκετές γνώσεις για αυτή με μια πιο εμπεριστατωμένη.

Ορισμοί

Οι οντολογίες είναι η σαφής περιγραφή ενός αντικειμένου. τα οντολογικά σχήματα περιγράφουν τη σχέση που έχουν οι κλάσεις που περιέχει μια οντολογία καθώς και τα πεδία ορισμού τους. Οι ορισμοί περιγράφονται αναλυτικά στο άλλο επισυναπτόμενο κείμενο.

Πηγές

Παραπομπή στη βιβλιογραφία.

Περιορισμοί

Το πρόγραμμα πρέπει να δέχεται αρχεία που η γλώσσα περιγραφής τους είναι η OWL. Αυτή η γλώσσα είναι αρκετά διαδεδομένη και χρηστική για την περιγραφή οντολογικών σχημάτων. Στην ίδια γλώσσα πρέπει να περιγράφεται και το παραγόμενο αρχείο της εξόδου. Οι περιορισμοί περιγράφονται εκτενέστερα και στο μέρος δεύτερο.

Αποτέλεσμα

Στο τέλος, λόγω της δημιουργίας μιας καινούριας οντολογίας ο χρήστης πρέπει να μπορεί να δει τη νέα οντολογία γραφικά αλλά και σε νέο αρχείο xml, αλλά και να μπορεί να την αλλάξει για να τη φέρει στην τελική επιθυμητή για τον ίδιο μορφή.

Γενική Περιγραφή
Προεπισκόπηση του μοντέλου που χρησιμοποιεί ο χρήστης

Το πρόγραμμα διαβάζει δυο αρχεία που περιγράφουν οντολογικά σχήματα, εξάγει τις κλάσεις, τα χαρακτηριστικά τους, τα αντικείμενά τους, και τον τρόπο που αυτές συνδέονται, τα αποθηκεύει σε μια δομή δεδομένων, τα συγκρίνει και φτιάχνει ένα νέο αρχείο. Αυτό το αρχείο περιέχει τη σύμπτυξη των δυο αρχείων εισόδου, που προκαλείται από την εκτέλεση διάφορων αλγορίθμων και τη χρήση των δεδομένων.

Υποθέσεις και περιορισμοί

Τα αρχεία εισόδου υποτίθεται ότι δεν λαμβάνονται από το διαδίκτυο αλλά είναι τοπικά αποθηκεμένα. Επίσης περιγράφουν ένα οντολογικό σχήμα από τα πιο γενικά του στοιχεία προς τα πιο ειδικά (πχ αν είναι δέντρο γίνεται κατά πλάτος ανάγνωση). Τα παραπάνω αρχεία δεν ελέγχονται για λάθη, πρέπει να τηρούν τη σύμβαση του τρόπου περιγραφής ο οποίος αναφέρει ότι πρέπει να περιγράφουν γράφο, με κλάσεις, αντικείμενα, και το πώς αυτά συνδέονται. Ο γράφος αυτός δεν πρέπει να απαραίτητα να είναι άκυκλος, αλλά το αρχείο xml που τον περιγράφει προσφέρει τη δυνατότητα δενδρικής άκυκλης αναπαράστασης των δεδομένων.

Αναφορά στις λειτουργικές απαιτήσεις του χρήστη

Το παραγόμενο αρχείο πρέπει να ακολουθεί τους κανόνες της γλώσσας xml owl. Πρέπει να φαίνονται οι σχέσεις γονιού και παιδιού όπου υπάρχουν, και πρέπει να μην αντιτίθενται με τις σχέσεις των δυο εισαγόμενων αρχείων.

Επίσης πρέπει να φαίνονται όλα τα χαρακτηριστικά μιας κλάσης που υπάρχουν στα αρχεία εισόδου και στο τελικό αρχείο.

Συμπληρωματικές απαιτήσεις

Θα ήταν θετικό το αρχείο της εξόδου να έχει κάποιους κανόνες και εξηγήσεις που υπάρχουν και στα άλλα αρχεία.

Περιεχόμενο των δεδομένων

Το περιεχόμενο των δεδομένων αρχείων έχει εντολές της owl και των σχημάτων rdf και σύνταξη xml.

Μη λειτουργικές απαιτήσεις

Το πρόγραμμα θα πρέπει να είναι απλό και χρηστικό, και να ανταποκρίνεται σε μεγάλο ποσοστό οντολογιών και για μεγάλο μέγεθος οντολογιών, να αναγνωρίζεται από τη χρηστικότητά του ότι απευθύνεται σε έμπειρους και εξειδικευμένους χρήστες που θέλουν να κάνουν κάτι πολύ συγκεκριμένο εύκολα και γρήγορα.

Το σύστημα με το οποίο αλληλεπιδρά ο χρήστης

Ο χρήστης εκτός από το xml αρχείο που παράγεται θα ήταν καλό να μπορεί να δει και γραφικά το οντολογικό σχήμα. Επίσης ο χρήστης θα πρέπει να μπορεί να δει τα δέντρα και να μπορεί να τα αλλάξει ή να τα μεγαλώσει πριν γίνει η ένωση, αλλά και μετά την ένωση και πριν δημιουργηθεί το τελικό αρχείο θα ήταν θεμιτό να μπορεί να παρέμβει στο γραφικό αποτέλεσμα.

Πληροφορίες υποστήριξης

Το πρόγραμμα ελέγχει για κοινά ονόματα η κόμβους. Θα μπορούσε να επεκταθεί ελέγχοντας και τους τύπους και τα πεδία ορισμού των κλάσεων καθώς και θα μπορούσε να καθοδηγείται από το χρήστη για αμφιλεγόμενες σχέσεις.

Απαιτήσεις διεύρυνσης του συστήματος

Το πρόγραμμα θα πρέπει να περιέχει γρήγορους και απλούς αλγόριθμους εντοπισμού κοινών στοιχείων, και αφού υπάρχει αλληλεπίδραση με το χρήστη, θα πρέπει να ενώνει μόνο τα στοιχεία τα οποία ο χρήστης κρίνει και βρίσκει όμοια. Στο μέλλον το σύστημα μπορεί να βελτιωθεί με πιο ευαίσθητους αλγόριθμους αναζήτησης ομοιότητας, και να υποστηρίζει και άλλες γλώσσες περιγραφής οντολογικών σχημάτων.

Επισύναψη

Το πρόγραμμα πρέπει να περιέχει και μια δομή δεδομένων που να αντιστοιχίζει τα παιδιά με τον πατέρα, και το αντίστροφο. Τα αρχεία περιγράφουν αυτές τις σχέσεις όπως και άλλες σχέσεις μεταξύ των κλάσεων, στην κοινή γλώσσα αυτές περιγράφονται από ρήματα.

5 Βιβλιογραφία – πηγές

1. **A Survey of Schema-based Matching Approaches**

   Pavel Shvaiko and Jerome Euzenat

   University of Trento, Povo, Trento, Italy,

   INRIA, Rhone-Alpes, France

2. **Resolution of conicts among ontology mappings: a fuzzy approach**

   Al_o Ferrara, Davide Lorusso, Giorgos Stamou, Giorgos Stoilos, Vassilis Tzouvaras, Tassos Venetis

3. **Ontology Development 101: A Guide to Creating Your First Ontology**

   Natalya F. Noy and Deborah L. McGuinness

   Stanford University, Stanford, CA, 94305

4. **Semantic Matching: Algorithms and Implementation**

   Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko

   Department of Information and Communication Technology, University of Trento

5. **A Community Based Approach for Managing Ontology Alignments**

   Gianluca Correndo, Harith Alani, Paul Smart

   University of Southampton, Electronic and Computer Science Department

6. **Semantic Integration Research in the Database Community: A Brief Survey**

7. Efficient semantic matching

Fausto Giunchiglia, Mikalai Yatskevich and Enrico Giunchiglia

December 2004

8. An algorithm for matching contextualized schemas via sat

Luciano Serafini, Paolo Bouquet, Bernardo Magnini and Stefano Zanobini

January 2003

9. Semantic Wikipedia

Markus Krötzsch Denny Vrandeci Max Völkel Heiko Haller Rudi Studer

10. παρουσίαση για το Swoogle

Tim Finin, Anupam Joshi, Yun Peng, R. Scott Cost, Jim Mayfield, Joel Sachs, Pavan Reddivari, Vishal Doshi, Rong Pan, Li Ding, and Drew Ogle. Partial research support was provided by DARPA contract F30602-00-0591 and by NSF by awards NSF-ITR-IIS-0326460 and NSF-ITR-IDM-0219649. 20 May 2004.

11. iMERGE: Interactive Ontology Merging

Zoulfa El Jerroudi and J urgen Ziegler

University Duisburg-Essen, 47057 Duisburg, Germany

12. The Semantic Web Revisited

Nigel Shadbolt and Wendy Hall, University of Southampton

Tim Berners-Lee, Massachusetts Institute of Technology

13. Web Science: An Interdisciplinary Approach to Understanding the Web

James Hendler, Nigel Shadbolt, Wendy Hall, Tim Berners-Lee, and Daniel Weitzner

doi: 10.1145/1364782.1364798

14. The Chimaera Ontology Environment

Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder

15. Η Χρήση των Θησαυρών στις Οντολογίες

Σταματίνα Τσάφου, Ματίνα Πολίτη

16. Ανάπτυξη Οντολογίας στο Protégé για την Αναπαράσταση Προϊόντων και Λειτουργιών Τραπεζικών Οργανισμών

Διπλωματική Εργασία της Μαρίας Παπαφώτη

17. Ontological Engineering

Asunción Gómez-Pérez

18. The Case for a Multilingual Upper-Level Electronic Commerce Ontology

Martin Bryan, The SGML Centre

19. ExploitingWeb Service Semantics: Taxonomies vs. Ontologies

Asuman Dogac, Gokce Laleci, Yildiray Kabak, Ibrahim Cingil

Software Research and Development Center Middle East Technical University (METU)

20. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL

Dean Allemang and James Hendler

21. Semantic Web Programming

by John Hebeler, Matthew Fisher, Ryan Blace, and Andrew Perez-Lopez

22. Ontology Management

Hepp, M.; Leenheer, P.; Moor, A.; Sure, Y. (Eds.)

2008, XIX, 295 p. 20 illus., Hardcover

ISBN: 978-0-387-69899-1

23. Model Driven Architecture and Ontology Development

Springer, 2006, ISBN: 3-540-32180-2

24. Ontology Matching

Jérôme Euzenat, Pavel Shvaiko

ISBN: 3-540-49611-4; ISBN13: 978-3-540-49611-3

25. Information Integrating Using Contextual Knowledge and Ontology Merging

Aykut Firat

26. Remarks on automated ontology merging algorithm

August 19-August 21
ISBN: 978-0-7695-3331-5

27. Ontology merging using answer set programming and linguistic knowledge

J urgen Bock1, Rodney Topor2, and Raphael Volz1
1 FZI Forschungszentrum Informatik, Karlsruhe, Germany fbock
2 Gri_th University, Brisbane, Australia

28. Ιστοχώροι

http://en.wikipedia.org/wiki/Ontology_%28information_science%29

http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

http://www.w3.org/2004/OWL/

http://www.jfsowa.com/ontology/

http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/Ontologies#cw

http://semanticweb.org/wiki/Ontology

http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary

http://www.w3.org/TR/webont-req/

http://www.pms.ifi.lmu.de/mitarbeiter/ohlbach/Ontology/

http://www.obitko.com/tutorials/ontologies-semantic-web/

http://ontologyonline.org/

http://protege.stanford.edu/

http://www.ontologymatching.org/

http://ontologies.freebase.com/

Καθώς και πολλά στοιχεία και παραδείγματα χρησιμοποιήθηκαν από σημειώσεις-παρουσιάσεις του Κώτη Κωνσταντίνου(ICSEng. Dept. University of the Aegean - 2007)