ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανακάλυψη κανόνων συσχέτισης στο Σημασιολογικό Ιστό: Μια επαγωγική μέθοδος

Ολυμπία Ν. Νίκου

Επιβλέποντες: **Ευστάθιος Χατζηευθυμιάδης,** Επίκουρος Καθηγητής ΕΚΠΑ
Βασίλειος Παπαταξιάρχης, Υποψήφιος Διδάκτωρ ΕΚΠΑ

ΑΘΗΝΑ

ΑΠΡΙΛΙΟΣ 2012

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανακάλυψη κανόνων συσχέτισης στο Σημασιολογικό Ιστό: Μια επαγωγική μέθοδος

Ολυμπία Ν. Νίκου

Α.Μ.: Μ956

ΕΠΙΒΛΕΠΟΝΤΕΣ: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Βασίλειος Παπαταξιάρχης, Υποψήφιος Διδάκτωρ ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Μανόλης Κουμπαράκης, Καθηγητής ΕΚΠΑ

Απρίλιος 2012

ΠΕΡΙΛΗΨΗ

Με την έλευση του Σημασιολογικού Ιστού (ΣΙ), οι οντολογίες έγιναν ο σημαντικότερος τρόπος αναπαράστασης γνώσης και συμπερασμού. Καθώς, όμως, ο όγκος των δεδομένων τους αυξάνεται ραγδαία, κρίνεται αναγκαία η εφαρμογή μεθόδων μηχανικής μάθησης με σκοπό την παραγωγή νέας γνώσης από αυτές. Η ανακάλυψη γνώσης στα πλαίσια του ΣΙ στοχεύει στη βελτίωση των αποτελεσμάτων της διαδικασίας παραγωγής γνώσης από τον Ιστό χρησιμοποιώντας τις νέες σημασιολογικές δομές των δεδομένων του. Η επίτευξή της απαιτεί την εφαρμογή των μεθοδολογιών της μηχανικής μάθησης, με σκοπό την ανάπτυξη κατάλληλων μεθόδων και εργαλείων για την εκπαίδευση των εννοιών σε Περιγραφικές Λογικές. Παρόλο που, μέθοδοι μηχανικής μάθησης έχουν εφαρμοστεί επιτυχώς για τη δημιουργία δεδομένων του ΣΙ από απλά δεδομένα, δεν έχουν χρησιμοποιηθεί αρκετά για την ανακάλυψη γνώσης από τα ήδη υπάρχοντα δεδομένα του ΣΙ.

Στην παρούσα εργασία παρουσιάζονται συνοπτικά τα συστήματα, που πραγματοποιούν ανακάλυψη γνώσης από έννοιες εκφρασμένες σε Περιγραφικές Λογικές. Επιπλέον, παρουσιάζεται η δημιουργία ενός συστήματος αυτόματης παραγωγής νέας γνώσης από τα δεδομένα του ΣΙ. Η βασική λειτουργικότητα του συστήματος είναι η παραγωγή SWRL κανόνων από τα δεδομένα μιας οντολογίας και η επιλογή των πλέον κατάλληλων κανόνων για την εγγραφή τους σ’ αυτήν. Τέλος, αξιολογήθηκαν η λειτουργία και οι επιδόσεις του συστήματος.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Τεχνολογίες Γνώσης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Σημασιολογικός Ιστός, Ανακάλυψη Γνώσης, Κανόνες Συσχέτισης, Υποστήριξη, Εμπιστοσύνη

ABSTRACT

With the advent of the Semantic Web (SW), ontologies have become the most prominent paradigm for knowledge representation and reasoning. As the volume of data of ontologies is increasing rapidly, the application of machine – learning approaches is judged necessary in order to induce new knowledge from the ontologies. Semantic Web Mining aims to improve the results of Web Mining by exploiting new semantic structures in the Web. The achievement of Semantic Web Mining requires the implementation of machine – learning approaches in order to develop methods and tools for learning concepts in description logics. Although, the machine – learning approaches have been successfully applied to create Semantic Web data from plain data, they have not been used enough to induce models from existing Semantic Web data.

In summary, the present work shows the tools, which induce models from Semantic Web data. Moreover, this thesis provides the implementation of a tool for automatic induction of new knowledge from ontologies. The basic system functionality is the production of SWRL rules from ontology and the choice of the most appropriate of them in order to be saved in ontology. In the end, the operation and the system performance were evaluated.

SUBJECT AREA: Knowledge Technologies

KEYWORDS: Semantic Web, Knowledge Discovery, Association Rules, Support, Confidence

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του Μεταπτυχιακού Προγράμματος Σπουδών (ειδίκευση «Προηγμένα Πληροφοριακά Συστήματα») του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Αθηνών. Το αντικείμενο μελέτης της είναι η εφαρμογή των μεθοδολογιών ανακάλυψης γνώσης για αυτόματη παραγωγή νέας γνώσης – παραγωγή κανόνων – από τα δεδομένα του Σημασιολογικού Ιστού. Η μελέτη των υπαρχόντων συστημάτων ανακάλυψης γνώσης στα πλαίσια του Σημασιολογικό Ιστό ανάδειξε την έλλειψη τους για αυτόματη παραγωγή κανόνων. Η ανάπτυξη του συστήματος περιλαμβάνει την παραγωγή SWRL κανόνων για τα δεδομένα του Σημασιολογικού Ιστού. Οι κανόνες, που προκύπτουν, βασίζονται στη σχεσιακή δομή των δεδομένων λαμβάνοντας, όμως, υπόψη και την υπονοούμενη (inferred) δομή της πληροφορίας που προκύπτει μέσω της διαδικασίας συμπερασμού.

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της εργασίας, επίκουρο καθηγητή κ. Ευστάθιο Χατζηευθυμιάδη για την καθοδήγηση και την πολύτιμη συνεισφορά του σε όλη τη διάρκεια εκπόνησής της. Επίσης, θα ήθελα να ευχαριστήσω θερμά το Βασίλειο Παπαταξιάρχη, υποψήφιο διδάκτορα του Τμήματος Πληροφορικής και Τηλεπικοινωνιών ΕΚΠΑ, για τις χρήσιμες οδηγίες και συμβουλές του κατά τη φάση της σχεδίασης του συστήματος. Η συνεχής παρακολούθηση της προόδου της διπλωματικής εργασίας και οι εύστοχες επισημάνσεις τους συνετέλεσαν στη διαμόρφωση του τελικού αποτελέσματος.

Ολυμπία Νίκου

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Σημασιολογικός Ιστός

Η ανάπτυξη του Παγκόσμιου Ιστού (World Wide Web, WWW) έχει αλλάξει τον τρόπο επικοινωνίας των ανθρώπων και έχει προσφέρει στους χρήστες ένα συνεχώς αυξανόμενο αριθμό από πηγές πληροφορίας. Η τωρινή μορφή του είναι μια μεγάλη βιβλιοθήκη από διασυνδεδεμένα έγγραφα, η ποιότητα των οποίων δεν μπορεί να εγγυηθεί. Το μεγαλύτερο μέρος της πληροφορίας δε γίνεται κατανοητό από τους υπολογιστές. Αυτό συμβαίνει, γιατί οι υπολογιστές εγγυώνται μόνο τη σωστή μεταφορά και παρουσίασή της αδυνατώντας για τη σωστή ερμηνεία της, αφού έχουν πρόσβαση μόνο σε ένα περιορισμένο τμήμα της, κυρίως στην κωδικοποίησή της. Αυτή η διαδικασία έχει ως αποτέλεσμα η ερμηνεία, η επεξεργασία και η κατανόηση της μεταφερόμενης πληροφορίας να βασίζεται αποκλειστικά στους χρήστες. Καθώς, όμως, ο όγκος των δεδομένων του διαδικτύου αυξάνεται συνεχώς, η ανάκτηση του χρήσιμου περιεχομένου γίνεται μια χρονοβόρα διαδικασία. Επομένως, η ανάπτυξη αυτοματοποιημένων λύσεων για πρακτική χρήση κρίνεται αναγκαία λόγω της συνεχής ανάπτυξης του Διαδικτύου.

Λύση στα προβλήματα του Παγκόσμιου Ιστού έρχεται να δώσει μια επέκταση αυτού, εμπνευσμένη από τον Tim Berners Lee, ο λεγόμενος Σημασιολογικός Ιστός (Semantic Web, SW) [1, 2]. Ο Σημασιολογικός Ιστός είναι μια προέκταση του τωρινού Ιστού, στον οποίο η πληροφορία γίνεται κατανοητή από υπολογιστές. Η πληροφορία αποκτά δομή και σημασιολογία, για να υποστηριχθεί η αποδοτική αναζήτηση, επεξεργασία, ενοποίηση και επαναχρησιμοποίησή της από εφαρμογές, με τελικό σκοπό τον εντοπισμό της χρήσιμης πληροφορίας για τους χρήστες.

Ο Σημασιολογικός Ιστός προέκυψε επειδή ο υπάρχων Παγκόσμιος Ιστός ήταν ελλιπής [2]. Δεν αποτελεί ένα καινούργιο είδος πληροφορίας, που υπάρχει παράλληλα με τον ήδη υπάρχον Ιστό, αλλά αποτελεί μια εξέλιξη αυτού. Ενώ ο Παγκόσμιος Ιστός είναι ένα τεράστιο κατανεμημένο σύστημα εγγράφων, ο Σημασιολογικός Ιστός σκοπεύει να δημιουργήσει ένα κατανεμημένο σύστημα γνώσης (βλ. Εικόνα 1.1). Αποτελεί, δηλαδή, έναν ιστό δεδομένων. Ο στόχος του είναι να μοιράζει πληροφορία και όχι έγγραφα. Προσφέρει ένα κοινό πλαίσιο, που επιτρέπει τα δεδομένα να μοιράζονται, να επαναχρησιμοποιούνται ανάμεσα σε εφαρμογές, και να μπορούν να επεξεργαστούν

εξίσου από εφαρμογές όπως και από τους ίδιους τους χρήστες. Τα χαρακτηριστικά που τον καθιστούν προτιμητέο είναι ότι είναι σχεδιασμένος ως παγκόσμιο μέσο για την ανταλλαγή δεδομένων και βασίζεται στον ορισμό και την επαναχρησιμοποίηση κοινών λεξιλογίων.

Βασικές Αρχές

Ο στόχος του Σημασιολογικού Ιστού είναι η προσαρμογή και επαναχρησιμοποίηση της ήδη υπάρχουσας δομής του Παγκόσμιου Ιστού. Η επαναχρησιμοποίηση θα διευκολύνει την ομαλή μετάβαση στον Σημασιολογικό Ιστό και θα αυξήσει τις πιθανότητες επιτυχίας του. Η πληροφορία θα πρέπει να προσπελαύνεται χρησιμοποιώντας τη γενική αρχιτεκτονική του Διαδικτύου. Όλες οι βασικές αρχές του τωρινού Ιστού εμπλουτίζονται για να καλυφθούν οι επιπλέον απαιτήσεις του Σημασιολογικού Ιστού. Αυτές οι εμπλουτισμένες αρχές [3] αναλύονται παρακάτω (βλ. Πίνακας 1.1):

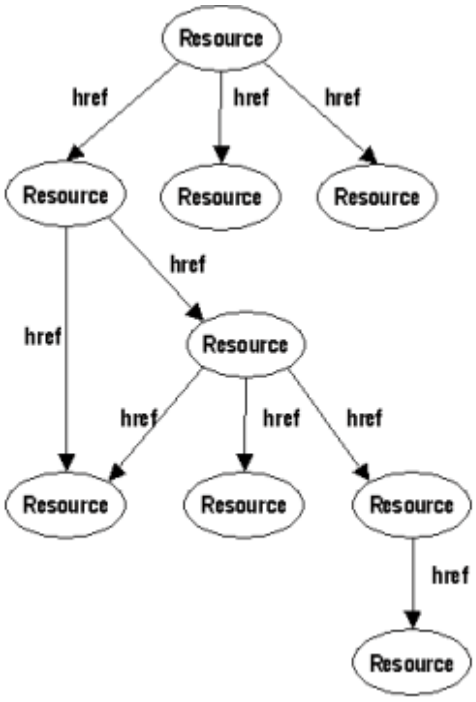
Πίνακας 1.1: Βασικές αρχές του Σημασιολογικού Ιστού

- **Η πληροφορία είναι αναγνωρίσιμη με URIs**: Οποιαδήποτε πληροφορία στο Σημασιολογικό Ιστό αναγνωρίζεται με URIs. Τα αναγνωριστικά επιτυγχάνουν την αποτελεσματική ενοποίηση και κατανόηση της πληροφορίας.
- **Ομαδοποίηση δεδομένων και συνδέσμων:** Ο Παγκόσμιος Ιστός αποτελείται από πηγές πληροφορίας (resources) και συνδέσμους χωρίς κάποια επιπλέον μεταπληροφορία, ώστε να αντιλαμβάνεται ο υπολογιστής το είδος της πληροφορίας ή ποια είναι η σχέση της με άλλες πηγές πληροφορίας του διαδικτύου. Αντίθετα, ο Σημασιολογικός Ιστός περιλαμβάνει την απαραίτητη μεταπληροφορία για τους υπολογιστές, ώστε να κατανοούν την πληροφορία που μεταφέρουν.
- **Ανεκτή η ελλιπής πληροφορία:** Όπως ο Παγκόσμιος Ιστός έτσι και ο Σημασιολογικός Ιστός είναι απεριόριστος: οποιοσδήποτε μπορεί να ορίσει οτιδήποτε δημιουργώντας διαφορετικούς τύπους συνδέσμων ανάμεσα στους πόρους. Μερικοί σύνδεσμοι μπορεί να σταματήσουν να υπάρχουν ή οι συγκεκριμένες διευθύνσεις μπορούν να επαναχρησιμοποιηθούν. Οι εφαρμογές του Σημασιολογικού Ιστού πρέπει να μπορούν να ανέχονται την αλλοίωση των δεδομένων και να μπορούν να λειτουργούν ανεξάρτητα από αυτό το γεγονός.
- **Όχι απαραίτητη η ορθότητα της πληροφορίας:** Οποιαδήποτε πληροφορία μπορεί να μην είναι αληθής. Η αξιοπιστία της κάθε πληροφορίας αξιολογείται από

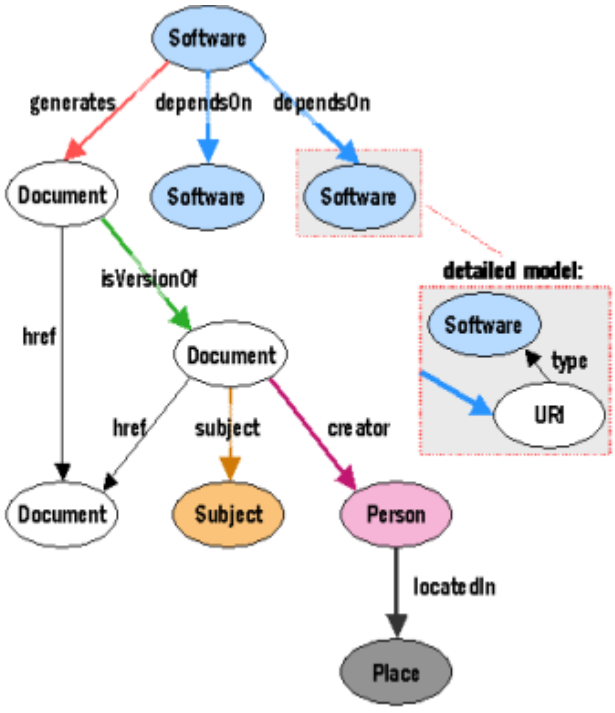
την κάθε εφαρμογή, που την επεξεργάζεται.

- **Δυνατότητα εξέλιξης του Ιστού:** Η ίδια πληροφορία μπορεί να αναπαρίσταται από διαφορετικές ομάδες ανθρώπων με διαφορετικό τρόπο. Πολύ συχνά αυτά τα δεδομένα πρέπει να συνδυάζονται, χρησιμοποιώντας αυτές τις περιγραφές. Ο σημασιολογικός Ιστός παρέχει τη δυνατότητα συνδυασμού των διαφορετικών μορφών της ίδιας πληροφορίας παρέχοντας εργαλεία που επιλύουν τις ασυνέπειες μεταξύ τους. Επίσης, η εισαγωγή νέας πληροφορίας δεν θα πρέπει να επηρεάζει την ήδη υπάρχουσα καταχωρημένη πληροφορία.
- **Απλός σχεδιασμός:** Η πληροφορία σχεδιάζεται απλά. Στόχος είναι να μην προτυποποιηθεί περισσότερη πληροφορία από ότι χρειάζεται και να μην επαναλαμβάνεται χωρίς λόγο.

α)



β)



Εικόνα 1.1: α) Αναπαράσταση Παγκόσμιου Ιστού, β) Αναπαράσταση Σημασιολογικού Ιστού [3]

1.2 Ανακάλυψη Γνώσης

Η παραδοσιακή μέθοδος μετατροπής δεδομένων σε γνώση βασίζεται σε μη αυτόματη ανάλυση και ερμηνεία. Καθώς όμως ο όγκος των δεδομένων και οι πηγές αυτών αυξάνονται η διαδικασία γίνεται χρονοβόρα, με αποτέλεσμα να υπερβαίνει τις ανθρώπινες ικανότητες για ανάλυση τέτοιων δεδομένων. Γι' αυτό το λόγο κρίνεται αναγκαία η δημιουργία εργαλείων, που θα βοηθήσουν τους χρήστες στην εξαγωγή της χρήσιμης πληροφορίας.

Ανακάλυψη γνώσης είναι η διαδικασία, στην οποία ένα μεγάλο πλήθος δεδομένων επεξεργάζεται, με σκοπό την παραγωγή προτύπων που θα αποτελούν τη γνώση για τα δεδομένα. Σύμφωνα με τον Fayyad [4], η ανακάλυψη γνώσης χαρακτηρίζεται ως μια χρονοβόρα διαδικασία εξαγωγής υπονοούμενης, άγνωστης και πιθανώς χρήσιμης πληροφορίας για τα δεδομένα. Ο βασικός στόχος της είναι η αντιστοίχηση των χαμηλού επιπέδου δεδομένων (που είναι συνήθως πάρα πολλά και πολύ δύσκολο να κατανοηθούν) σε μορφές που είναι περισσότερο συμπαγείς, αφαιρετικές (περιγραφικό μοντέλο) και χρήσιμες (μοντέλο πρόβλεψης).

Η γενική διαδικασία μετατροπής απλών δεδομένων σε νέα πληροφορία αποτελείται από ένα σύνολο σταδίων (βλ. Πίνακα 1.2), που μπορούν να χωριστούν σε τρεις κατηγορίες: προεπεξεργασία δεδομένων (data preprocessing), ταξινόμηση ή ομαδοποίηση (data mining algorithm), επεξεργασία παραγόμενης γνώσης (discovered – knowledge postprocessing).

Πίνακας 1.2: Στάδια διαδικασίας ανακάλυψης γνώσης

Προεπεξεργασία Δεδομένων

- **Καθαρισμός Δεδομένων**: Είναι απαραίτητο τα δεδομένα να είναι όσο το δυνατόν πιο ακριβή. Σ' αυτό το στάδιο αποκλείονται δεδομένα με θόρυβο ή άσχετα δεδομένα.
- **Ενοποίηση Δεδομένων**: Αυτό το στάδιο είναι απαραίτητο, αν τα δεδομένα που πρόκειται να χρησιμοποιηθούν στη διαδικασία προέρχονται από διαφορετικές πηγές. Σ' αυτό το στάδιο γίνεται η επίλυση των ασυνεπειών των ονομάτων των χαρακτηριστικών ή/και των τιμών αυτών ανάμεσα σε σύνολα δεδομένων διαφορετικών πηγών.
- **Επιλογή χαρακτηριστικών**: Επιλέγεται το υποσύνολο των χαρακτηριστικών, που θα χρησιμοποιηθεί για την ανακάλυψη γνώσης.

- **Μετασχηματισμός Δεδομένων**: Τα επιλεγμένα χαρακτηριστικά μετασχηματίζονται σε κατάλληλη και αποδεκτή μορφή σύμφωνη με τη μορφή του επόμενου σταδίου.

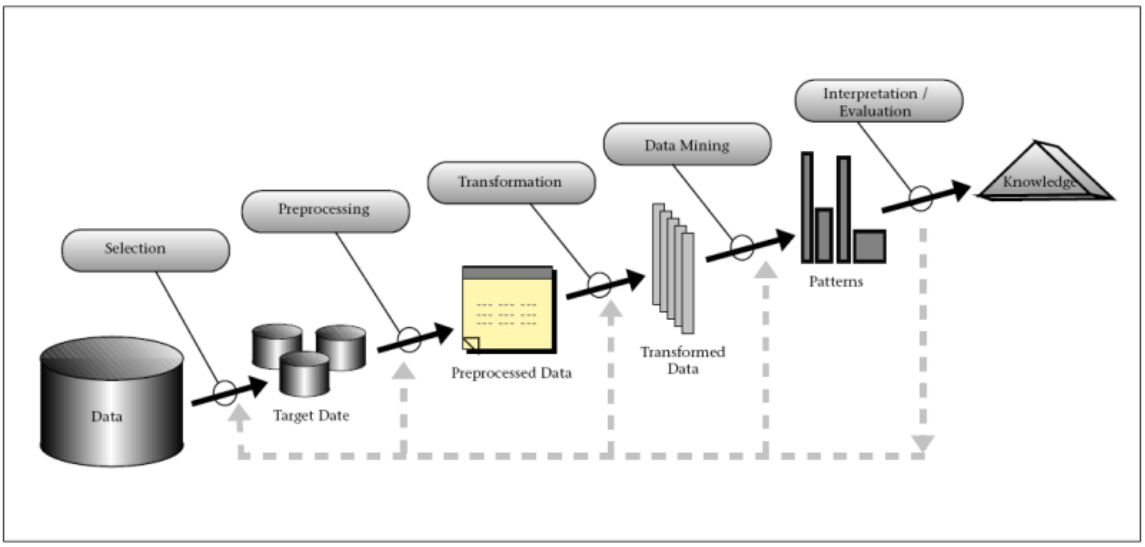
Ταξινόμηση ή Ομαδοποίηση

- **Εξόρυξη Δεδομένων**: Είναι το πιο σημαντικό στάδιο της διαδικασίας, εδώ εφαρμόζονται «έξυπνες» τεχνικές για την εξαγωγή πιθανών χρήσιμων προτύπων. Οι τεχνικές, που χρησιμοποιούνται μπορεί να είναι: Επαγωγικός Λογικός Προγραμματισμός (Inductive Logic Programming, ILP), Bayes Ταξινομητής, Αναγνώριση προτύπων, κ.α.

Επεξεργασία Παραγόμενης Γνώσης

- **Αξιολόγηση προτύπων**: Διακρίνονται τα πιο ενδιαφέροντα πρότυπα σύμφωνα με κάποιες μετρικές.
- **Αναπαράσταση γνώσης**: Η γνώση, που ανακαλύφθηκε, παρουσιάζεται στο χρήστη. Περιλαμβάνονται τεχνικές, που βοηθούν τον χρήση να κατανοήσει και να ερμηνεύσει σωστά τα αποτελέσματα της διαδικασίας.

Η διαδικασία της ανακάλυψης γνώσης είναι επαναληπτική. Αυτό σημαίνει ότι η έξοδος του τελευταίου σταδίου δεν αποτελεί μόνο την παραχθείσα γνώση, αλλά και ανατροφοδότηση σε κάποιο προηγούμενο στάδιο (βλ. Εικόνα 1.2).



Εικόνα 1.2: Τα στάδια της διαδικασίας της ανακάλυψης γνώσης [4]

1.3 Προβλήματα Ανακάλυψης Γνώσης από το Σημασιολογικό Ιστό

Η ανακάλυψη γνώσης είναι, επίσης, μια χρήσιμη διαδικασία και στα πλαίσια του Σημασιολογικού Ιστού. Όμως, η κλασική εφαρμογή της γνώσης αδυνατεί να παράγει κανόνες, που να λαμβάνουν υπόψη τις ιδιαιτερότητες του Σημασιολογικού Ιστού. Η απότομη αύξηση των δεδομένων, η ανομοιογένειά τους, η δυναμικότητά τους καθώς και οι σημασιολογικές συσχετίσεις μεταξύ αυτών είναι οι αιτίες που η κλασική εφαρμογή της διαδικασίας στα δεδομένα του Σημασιολογικού Ιστού δεν είναι εφικτή [5].

Καθώς τα δεδομένα του Σημασιολογικού Ιστού αλλάζουν συνεχώς, νέα χαρακτηριστικά μπορεί να προστίθενται ή να αφαιρούνται, οπότε να αλλάζει κάθε φορά το υποσύνολο των χρήσιμων χαρακτηριστικών για την εξαγωγή της νέας γνώσης. Αυτήν την ιδιαιτερότητα η κλασική μορφή της ανακάλυψης γνώσης δεν την περιλαμβάνει. Οι αλγόριθμοι, που χρησιμοποιούνται, δεν είναι δυναμικοί και επεκτάσιμοι, έτσι ώστε να προσαρμόζονται στα δεδομένα του Σημασιολογικού Ιστού.

Επιπλέον, δεν υποστηρίζεται η σημασιολογική ενοποίηση των δεδομένων. Τα δεδομένα του Σημασιολογικού Ιστού μπορούν, επίσης, να προκύπτουν από διαφορετικές πηγές δεδομένων, όμως εκτός από την απλή ενοποίησή τους, που πραγματοποιεί και η κλασική διαδικασία, είναι απαραίτητη και η σημασιολογική ενοποίησή τους. Είναι χρήσιμο να εντοπιστούν και οι σχέσεις των δεδομένων ανάμεσα στις διαφορετικές πηγές. Η κλασική διαδικασία ανακάλυψης γνώσης δεν υποστηρίζει πολύπλοκες σχέσεις μεταξύ οντοτήτων και η ανακάλυψη αυτών επιτυγχάνεται μόνο μέσω της σημασιολογικής ενοποίησης των δεδομένων.

Τέλος, η ανακάλυψη γνώσης στο Σημασιολογικό Ιστό δεν είναι μια διαδικασία, που πραγματοποιείται μια φορά και οι κανόνες, που προκύπτουν ισχύουν για πάντα. Καθώς τα δεδομένα του Σημασιολογικού Ιστού αλλάζουν συνεχώς, νέες σημασιολογικές συσχετίσεις μπορούν να δημιουργούνται ανάμεσα στα δεδομένα ή και να καταργούνται κάποιες άλλες, αυτό έχει σαν αποτέλεσμα να καταργούνται κάποιοι κανόνες ή να πρέπει να προκύψουν και κάποιοι καινούργιοι. Επομένως, η ανακάλυψη γνώσης πρέπει να πραγματοποιείται συνεχώς, ώστε οι κανόνες, που προκύπτουν να συμβαδίζουν πάντα με τις αλλαγές, που έχουν πιθανώς γίνει.

1.4 Στόχοι Εργασίας

Το αντικείμενο μελέτης της εργασίας αφορά την ανακάλυψη γνώσης στα πλαίσια του Σημασιολογικού Ιστού. Οι στόχοι της εργασίας είναι: η προσαρμογή της διαδικασίας της

εξόρυξης δεδομένων στα δεδομένα του Σημασιολογικού Ιστού και η δυνατότητα αυτόματης παραγωγής νέας γνώσης – παραγωγή κανόνων - από τα δεδομένα του Σημασιολογικού Ιστού.

1.5 Οργάνωση εργασίας

Η οργάνωση της εργασίας έχει ως εξής: στο κεφάλαιο 2 δίνονται στοιχεία σχετικά με την αναπαράσταση γνώσης στο Σημασιολογικό Ιστό. Περιγράφονται συνοπτικά οι Περιγραφικές Λογικές, η πιο γνωστή γλώσσα ανάπτυξης οντολογιών OWL και η γλώσσα κανόνων του Σημασιολογικού Ιστού SWRL. Στο κεφάλαιο 3 παρουσιάζονται μερικά από τα πιο γνωστά εργαλεία αναπαράστασης γνώσης στο Σημασιολογικό Ιστό και μερικά από τα πιο γνωστά εργαλεία συμπερασμού. Οι μηχανές συμπερασμού χωρίζονται στις μηχανές κανόνων και στις μηχανές συμπερασμού. Στο κεφάλαιο 4 παρουσιάζεται η ανακάλυψη γνώσης από το Σημασιολογικό Ιστό, οι εφαρμογές και οι ιδιαιτερότητές της και οι μέθοδοι που θεωρούνται καταλληλότεροι για την ανακάλυψη γνώσης από το Σημασιολογικό Ιστό. Τέλος, παρουσιάζονται και τα ήδη υπάρχοντα εργαλεία, που πετυχαίνουν ανακάλυψη γνώσης από Περιγραφικές Λογικές. Στο κεφάλαιο 5 παρουσιάζεται η μεθοδολογία που ακολουθεί το σύστημα για την αυτόματη παραγωγή νέας γνώσης από το Σημασιολογικό Ιστό. Παρουσιάζεται ο αλγόριθμος της μεθόδου και τα κριτήρια που πρέπει να ικανοποιούν οι προκύπτοντες κανόνες. Στο κεφάλαιο 6 γίνεται η αξιολόγηση του συστήματος και εξάγονται συμπεράσματα σχετικά με την απόδοση του. Η εργασία ολοκληρώνεται με το κεφάλαιο 7 στο οποίο παρουσιάζονται τα συμπεράσματα της όλης μελέτης και κάποια «ανοικτά» θέματα που αφορούν το συγκεκριμένο πεδίο έρευνας.

ΚΕΦΑΛΑΙΟ 2

ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ

Η εξέλιξη της Τεχνητής Νοημοσύνης αποδεικνύει ότι η γνώση είναι ένας σημαντικός παράγοντας για τα «έξυπνα» συστήματα. Σε πολλές περιπτώσεις, η γνώση μπορεί να αποδειχθεί σημαντικότερη για την επίλυση ενός προβλήματος από τη χρήση αποδοτικών αλγορίθμων. Για να έχουμε ένα πραγματικά «έξυπνο» σύστημα, η γνώση θα πρέπει να μπορεί να εντοπίζεται, να επεξεργάζεται και να επαναχρησιμοποιείται, χαρακτηριστικά τα οποία υποστηρίζονται από τις οντολογίες.

Η Οντολογία είναι μια ακριβής περιγραφή μιας εννοιολογικής θεώρησης ενός φαινομένου (explicit specification of conceptualization) [6]. Περιγράφει τη δομή ενός πεδίου εφαρμογής (application domain) και περιλαμβάνει ένα σύνολο από κλάσεις αντικειμένων και συσχετίσεις μεταξύ αυτών. Η οντολογία χρησιμοποιείται τόσο για την ύπαρξη ενός κοινά αποδεκτού λεξιλογίου του πεδίου εφαρμογής όσο και για την εξαγωγή συμπερασμάτων εκμεταλλευόμενοι τα στοιχεία μοντελοποίησης και σημασιολογίας της.

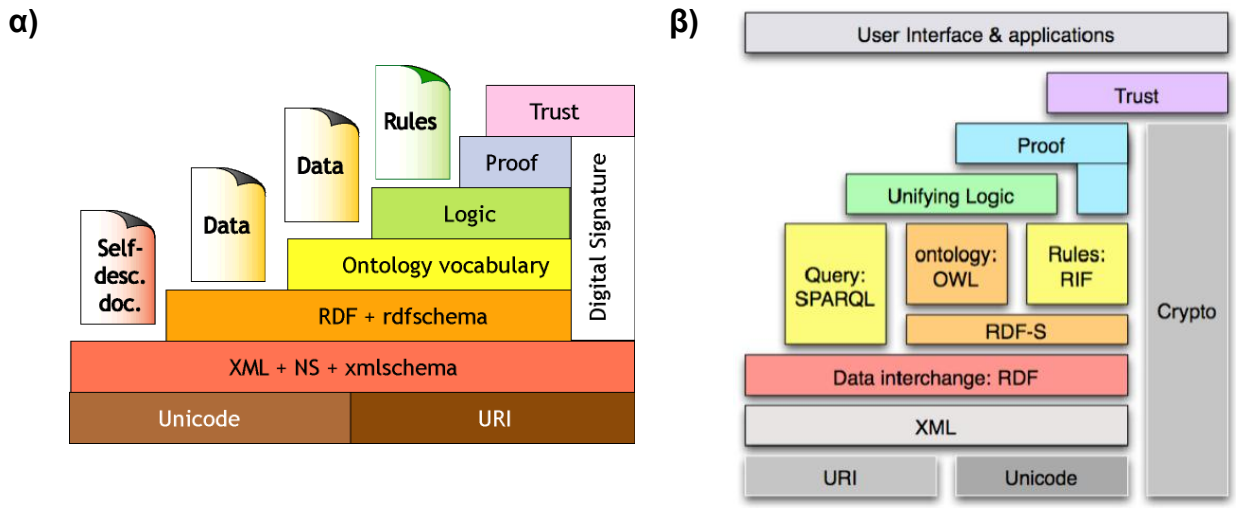
Οι οντολογίες αποτελούν βασικό τρόπο αναπαράστασης γνώσης στον Σημασιολογικό Ιστό, προέκταση της περιοχής της Τεχνητής Νοημοσύνης, εξαιτίας της προτυποποίησης που παρέχει σε γλώσσες και τεχνολογίες ανάπτυξης και χρήσης οντολογιών. Πιο συγκεκριμένα, σήμερα η πιο διαδεδομένη γλώσσα για δημιουργία οντολογιών είναι η Web Ontology Language (OWL) [6, 13] και ο συμπερασμός τους γίνεται με τεχνολογίες κανόνων (σε συνδυασμό με οντολογίες), με κυριότερη την Semantic Web Rule Language (SWRL) [15].

2.1 Semantic Web Layer Cake

Η ανάπτυξη του Σημασιολογικού Ιστού γίνεται σε επίπεδα (Semantic Web Layer Cake) [1, 7 - 11]. Τα διάφορα επίπεδα του Σημασιολογικού Ιστού, που έχουν συμφωνηθεί, π.χ. XML, RDF/RDF Schema, έχουν προτυτοποιηθεί, με σκοπό να ακολουθούνται από όλες τις ομάδες, που δημιουργούν εργαλεία, που διαχειρίζονται δεδομένα του Σημασιολογικού Ιστού.

Στην Εικόνα 2.1α παρουσιάζεται η αρχική εκδοχή της αρχιτεκτονικής του Σημασιολογικού Ιστού, που προτάθηκε από τον Tim Berners – Lee. Η Εικόνα 2.1β δείχνει μια πιο πρόσφατη απεικόνιση. Σημείο αναφοράς των επιμέρους προσπαθειών

αποτελεί ο βαθμός στον οποίο θα ενοποιηθούν τα επίπεδα οντολογιών και κανόνων του Semantic Web Layer Cake. Παρόλα αυτά, νέες βελτιώσεις ή/και τροποποιήσεις είναι πιθανό να προκύψουν στο μέλλον σχετικά με την αρχιτεκτονική του Σημασιολογικού Ιστού .



Εικόνα 2.1:α) Αρχική εκδοχή Semantic Web Layer Cake, β) Πρόσφατη εκδοχή [8]

2.2 Μεθοδολογίες Αναπαράστασης Γνώσης

Περιγραφικές Λογικές

Οι Περιγραφικές Λογικές (Description Logics, DL) [12] είναι η μέθοδος αναπαράστασης γνώσης ενός πεδίου εφαρμογής, δηλαδή την αναπαράσταση των εννοιών, των ιδιοτήτων (properties) και των στιγμιοτύπων (individuals) του πεδίου ορισμού. Οι DL προέρχονται από τα σημασιολογικά δίκτυα και δεν είναι μια γλώσσα, αλλά ένα σύνολο γλωσσών. Τα χαρακτηριστικά των DL είναι η τυπική λογική σημασιολογία και η δυνατότητα συμπερασμού (reasoning) γνώσης από την ήδη ορισμένη γνώση της βάσης. Τα βασικά στοιχεία τους είναι: οι έννοιες (concepts), οι ρόλοι (roles), τα στιγμιότυπα (individuals) και οι ισχυρισμοί (assertions) των στιγμιοτύπων.

Ένα σύστημα βάσης γνώσης (knowledge – based system), που βασίζεται σε DL, παρέχει δυνατότητες για δημιουργία βάσης γνώσης, συμπερασμού του περιεχομένου της, και επεξεργασία αυτού (βλ. Εικόνα 2.2). Η βάση γνώσης αποτελείται από δύο συστατικά, το TBox (Terminological Box) και το ABox (Assertional Box).

Το TBox περιλαμβάνει την ορολογία (terminology), π.χ. το λεξιλόγιο του πεδίου ορισμού της εφαρμογής, δηλαδή περιλαμβάνει έννοιες και ρόλους – δυαδικές σχέσεις – μεταξύ

των στιγμιοτύπων. Μπορεί να περιλαμβάνει, επίσης ορισμούς εννοιών από ήδη ορισμένες έννοιες. Για παράδειγμα:

$$\text{Woman} \equiv \text{Person} \sqcap \text{Female}$$

η παραπάνω δήλωση ερμηνεύεται σαν μια λογική ισοδυναμία, που ορίζει τις απαραίτητες συνθήκες για να χαρακτηριστεί ένα στιγμιότυπο ως «Woman». Για κάθε έννοια επιτρέπεται μόνο ένας ορισμός. Οι ορισμοί των εννοιών δεν είναι κυκλικοί, δηλαδή μια έννοια δεν περιλαμβάνεται στο δικό της ορισμό.

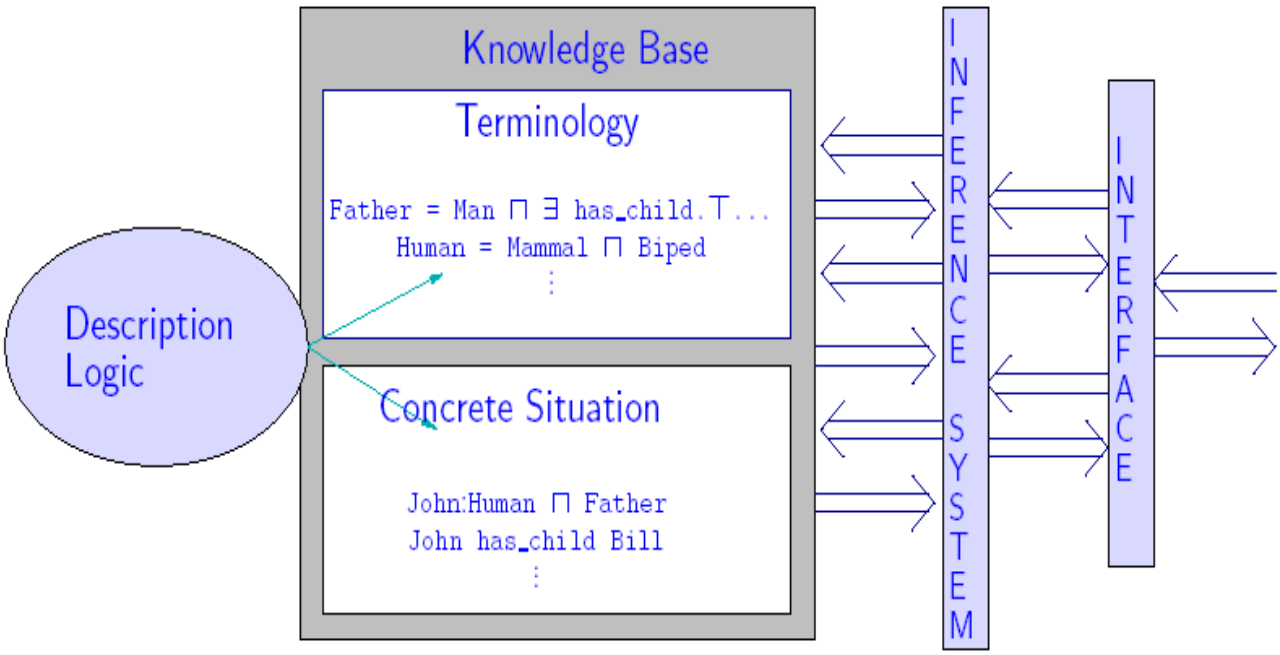
Το ABox περιλαμβάνει τους ορισμούς των στιγμιοτύπων του συστήματος, προσφέροντας επιπλέον γνώση για το πεδίο ενδιαφέροντος. Για παράδειγμα, ο ορισμός:

$$\text{Female} \sqcap \text{Person}(\text{ANNA}),$$

δηλώνει ότι το στιγμιότυπο «Άννα» ανήκει στις έννοιες «Female» και «Person» και σύμφωνα με τον παραπάνω ορισμό της έννοιας «Woman», μπορεί να προκύψει ότι η «Άννα» είναι στιγμιότυπο της «Woman».

Ένα DL σύστημα δεν αποθηκεύει μόνο το λεξιλόγιο και ορισμούς στιγμιοτύπων, αλλά προσφέρει και δυνατότητα συμπερασμού νέας γνώσης μέσω της ιεραρχίας των εννοιών. Επίσης, είναι εφικτό να προστεθούν κανόνες για το συμπερασμό γνώσης (inference rules). Αυτοί τοποθετούνται συνήθως στο TBox, καθώς συμπληρώνουν τους ορισμούς των εννοιών και των ρόλων. Για να επιτευχθεί ο συμπερασμός νέας γνώσης, δεν θα πρέπει να υπάρχουν συγκρούσεις στη περιγραφή του πεδίου εφαρμογής, ενώ στο ABox το σύνολο των ορισμών των στιγμιοτύπων πρέπει να είναι συνεπές. Οι έλεγχοι ικανοποιησιμότητας (satisfiability) και συνέπειας (consistency) των ορισμών των στιγμιοτύπων, είναι χρήσιμοι για να αποφασιστεί αν η βάση γνώσης είναι ικανοποιήσιμη και συνεπής, αντιστοίχως.

Οι Περιγραφικές Λογικές αποτελούν τη βάση για τη δημιουργία της γλώσσας αναπαράστασης γνώσης στο Διαδίκτυο, OWL.



Εικόνα 2.2: Αρχιτεκτονική συστήματος αναπαράστασης γνώσης βασισμένο σε DL

2.3 Η γλώσσα αναπαράστασης γνώσης – OWL

Η γλώσσα αναπαράστασης OWL [6, 13] είναι μια γλώσσα δημιουργίας οντολογιών, που προτάθηκε από το World Wide Web Consortium (W3C). Η γλώσσα OWL προέκυψε από τη γλώσσα DAML+OIL, το συντακτικό της βασίζεται στην XML και στο RDF/RDF Schema (Resource Description Framework) (βλ. Εικόνα 2.1), η εκφραστικότητα και η σημασιολογία της καθορίζεται από τις Περιγραφικές Λογικές (Description Logics).

Η γλώσσα OWL προσφέρει τρία διαφορετικά επίπεδα εκφραστικότητας, καθένα από τα οποία αποτελεί μια επέκταση του προηγούμενου επιπέδου:

- **OWL Lite:** Η OWL – Lite [13] υλοποιεί την Περιγραφική Λογική $SHIF(D_n)$. Υποστηρίζει μόνο τα βασικά χαρακτηριστικά της γλώσσας OWL, και προσφέρει τη δυνατότητα μόνο απλών ταξινομήσεων με απλούς περιορισμούς, π.χ. υποστηρίζει τον περιορισμό της πληθικότητας, όμως επιτρέπει σαν τιμές μόνο 0 ή 1.
- **OWL DL:** Η OWL – DL [13] επεκτείνει την εκφραστικότητα της OWL – Lite και υλοποιεί τη Περιγραφική Λογική $SHOIN(D_n)$. Η OWL – DL υποστηρίζει αποφασίσιμες διαδικασίες συμπερασμού. Είναι αρκετά εκφραστική για την αναπαράσταση και τη μοντελοποίηση των πεδίων ενδιαφέροντος, που περιλαμβάνουν πολύπλοκες συσχετίσεις. Περιλαμβάνει όλους τους βασικούς

κατασκευαστές (constructors) της γλώσσας OWL (Εικόνα 2.3), όμως προϋποθέτει την ύπαρξη κάποιων περιορισμών, π.χ. μία κλάση δεν μπορεί ταυτόχρονα να είναι και στιγμιότυπο ή ιδιότητα, το ίδιο ισχύει και για τα στιγμιότυπα όπως και για τις ιδιότητες.

- **OWL Full:** Η OWL – Full [13] προσφέρει τη μέγιστη εκφραστικότητα. Υποστηρίζει πλήρως τη γλώσσα OWL και το συντακτικό της δεν εξαρτάται από το RDF. Στην OWL – Full μια κλάση μπορεί να οριστεί σαν ένα σύνολο στιγμιοτύπων, όμως την ίδια στιγμή μπορεί να οριστεί και η ίδια ως στιγμιότυπο. Όμως, εξαιτίας της μεγάλης εκφραστικότητας που προσφέρει, είναι μια μη αποφασίσιμη γλώσσα,. Προς το παρόν δεν υπάρχουν αλγόριθμοι διαδικασιών συμπερασμού, που να υποστηρίζουν όλα τα χαρακτηριστικά της. Χρησιμοποιείται μόνο σε περιπτώσεις, που η υψηλή εκφραστικότητα είναι πιο σημαντική από την εγγύηση της αποφασισιμότητας της γλώσσας.

Constructor	DL syntax	Example
intersectionOf	$C_1 \sqcap \cdots \sqcap C_n$	Human ⊓ Male
unionOf	$C_1 \sqcup \cdots \sqcup C_n$	Doctor ⊔ Lawyer
complementOf	$\neg C$	¬Male
oneOf	$\{x_1 \ldots x_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	∀hasChild.Doctor
someValuesFrom	$\exists r.C$	∃hasChild.Lawyer
hasValue	$\exists r.\{x\}$	∃citizenOf.{USA}
minCardinality	$(\geqslant nr)$	$(\geqslant 2$ hasChild)
maxCardinality	$(\leqslant nr)$	$(\leqslant 1$ hasChild)
inverseOf	r^-	hasChild$^-$

Εικόνα 2.3: Οι constructors της γλώσσας OWL [14]

Τα βασικά στοιχεία της γλώσσας OWL είναι η κλάση (class), η οποία μπορεί να περιλαμβάνει ένα σύνολο στιγμιοτύπων (individuals) και ένα σύνολο συσχετίσεων (properties) μεταξύ των στιγμιοτύπων. Τέλος, επιτρέπει τον ορισμό κάποιων αξιωμάτων, που καθορίζουν, π.χ. την ιεραρχία ή την ισοδυναμία των κλάσεων ή των ιδιοτήτων, και επιτρέπουν τη διαδικασία συμπερασμού (reasoning) στη βάση γνώσης. Παρακάτω εμφανίζονται τα βασικά στοιχεία της γλώσσας (Πίνακας 2.1) και μερικά από τα αξιώματα της γλώσσας (Εικόνα 2.4).

Πίνακας 2.1: Βασικά στοιχεία της γλώσσας OWL

Στοιχείο	Περιγραφή
Class	Μία έννοια του πεδίου εφαρμογής. Μια κλάση μπορεί να οριστεί και μέσω άλλων κλάσεων χρησιμοποιώντας κάποιους από τους constructors της γλώσσας OWL.
Property	Η ιδιότητα είναι μια δυαδική σχέση. Οι ιδιότητες διακρίνονται σε ObjectProperties και DatatypeProperties
ObjectProperty	Είναι οι σχέσεις ανάμεσα σε στιγμιότυπα κλάσεων
DatatypeProperty	Είναι οι σχέσεις ανάμεσα σε στιγμιότυπο μιας κλάσης και κάποιο χαρακτηριστικό.
Individual	Στιγμιότυπο μιας κλάσης

Axiom	DL syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$Female
sameAs	$\{x_1\} \equiv \{x_2\}$	{Pres_Bush} $\equiv$ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
TransitiveProperty	P transitive role	hasAncestor is a transitive role
FunctionalProperty	$\top \sqsubseteq (\leqslant 1\ P)$	$\top \sqsubseteq (\leqslant 1$ hasMother)
InverseFunctionalProperty	$\top \sqsubseteq (\leqslant 1\ P^-)$	$\top \sqsubseteq (\leqslant 1$ isMotherOf$^-$)
SymmetricProperty	$P \equiv P^-$	isSiblingOf $\equiv$ isSiblingOf$^-$

Εικόνα 2.4: Αξιώματα της γλώσσας OWL [14]

Ένα παράδειγμα οντολογίας σε γλώσσα OWL φαίνεται παρακάτω:

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns="http://example.com/father#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://example.com/father">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="father"/>
  <owl:Class rdf:ID="female"/>
  <owl:Class rdf:ID="male">
    <owl:equivalentClass>
      <owl:Class>
        <owl:complementOf rdf:resource="#female"/>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="hasChild"/>
  <father rdf:ID="markus">
    <rdf:type rdf:resource="#male"/>
    <hasChild>
      <female rdf:ID="anna">
        <hasChild>
          <male rdf:ID="heinz"/>
        </hasChild>
      </female>
    </hasChild>
  </father>
  <father rdf:ID="stefan">
    <rdf:type rdf:resource="#male"/>
    <hasChild rdf:resource="#markus"/>
  </father>
  <female rdf:ID="michelle"/>
  <father rdf:ID="martin">
    <hasChild rdf:resource="#heinz"/>
    <rdf:type rdf:resource="#male"/>
  </father>
</rdf:RDF>
```

Η τελευταία έκδοση της γλώσσας OWL, OWL 2 [32], προσφέρει δύο διαφορετικά επίπεδα εκφραστικότητας: OWL 2 DL και OWL 2 Full. Η OWL 2 DL υλοποιεί την Περιγραφική Λογική $SROIQ(D)$. Ενώ η OWL 2 Full προσφέρει και πάλι τη μέγιστη εκφραστικότητα. Η OWL 2 Profile είναι μια περιορισμένη έκδοση της OWL 2, στην οποία περιορίζεται λίγο η εκφραστικότητα της OWL 2 με σκοπό μεγαλύτερη απόδοση στη

διαδικασία συμπερασμού. Η OWL 2 προσφέρει τρία διαφορετικά profiles, καθένα από τα οποία δεν είναι το ίδιο αποδοτικό και χρησιμοποιείται σε διαφορετικές εφαρμογές:

- **OWL 2 EL:** Η OWL 2 EL [32] βασίζεται στη λογική EL++ [33], που εγγυάται πολυωνυμικό χρόνο συμπερασμού ανάλογα με το μέγεθος της οντολογίας. Χρησιμοποιείται σε εφαρμογές, που οι οντολογίες έχουν μεγάλο αριθμό ιδιοτήτων (properties) και κλάσεων (classes) και απαιτείται εκφραστική μοντελοποίηση των ιδιοτήτων (expressive property modeling).
- **OWL 2 QL:** Η OWL 2 QL [32] χρησιμοποιείται σε εφαρμογές, που χρησιμοποιείται μεγάλος όγκος παραδειγμάτων και οι απαντήσεις ερωτήσεων είναι πιο σημαντικές από τη διαδικασία του συμπερασμού. Οι απαντήσεις των ερωτήσεων μπορούν να υλοποιηθούν σε ένα τυπικό σχεσιακό σύστημα βάσεων δεδομένων.
- **OWL 2 RL:** Η OWL 2 RL [32] χρησιμοποιείται σε επεκτάσιμες (scalable) εφαρμογές, στις οποίες δεν είναι επιθυμητός ο περιορισμός της εκφραστικότητας. Σ' αυτήν την περίπτωση χρησιμοποιούνται συστήματα συμπερασμού, που χρησιμοποιούν μηχανές συμπερασμού βασισμένες σε κανόνες (rule – based reasoning engines).

Κάθε ένα από τα profiles επιβάλλει, συνήθως συντακτικούς, περιορισμούς σε OWL, έτσι ώστε να είναι πιο αποτελεσματική η διαδικασία συμπερασμού. Τα profiles OWL 2 EL και OWL 2 QL είναι υποσύνολα της OWL 2 DL. Το OWL 2 RL προκύπτει σε δύο υποπεριπτώσεις, εκ των οποίων η μια είναι υποσύνολο της OWL 2 Full και η άλλη υποσύνολο της OWL 2 DL.

2.4 Η γλώσσα κανόνων του Σημασιολογικού Ιστού – SWRL

Στα πλαίσια του Σημασιολογικού Ιστού είναι χρήσιμη η παραγωγή κανόνων, οι οποίοι ταυτόχρονα θα είναι κατανοητοί από τους υπολογιστές. Η SWRL [15] είναι μια πρόταση του W3C για προσθήκη κανόνων στη γλώσσα OWL, δηλαδή προσθήκη αξιωμάτων κανόνων (rule axioms) στο σύνολο των αξιωμάτων της γλώσσας (OWL axioms). Η SWRL προκύπτει από το συνδυασμό των OWL DL και OWL Lite με τη γλώσσα Datalog RuleML (Rule Markup Language) [18].

Συντακτικό

Ένας SWRL κανόνας αποτελείται από το «σώμα» (Body ή Antecedent) και την «κεφαλή» (Head ή Consequent), καθένα από τα οποία μπορεί να περιλαμβάνει κανένα ή περισσότερα στοιχεία (atoms). Ένας κανόνας, όπου είτε το σώμα του είναι κενό

(γεγονότα που πάντα είναι αληθή) είτε η κεφαλή του (συνθήκες που δεν ισχύουν ποτέ) δεν έχει κάποιο ενδιαφέρον, γιατί δεν εισάγει νέα γνώση στη βάση γνώσης.

$$Antecedent \Rightarrow Consequent$$

Το σώμα ή/και η κεφαλή του κανόνα μπορεί να αποτελείται από πολλά στοιχεία (atoms), που συνδέονται με λογική σύζευξη μεταξύ τους. Το συντακτικό της γλώσσας δεν επιτρέπει τη λογική διάζευξη ή μη μονοτονικά χαρακτηριστικά, π.χ άρνηση. Ένας κανόνας, του οποίου η κεφαλή αποτελείται από πολλά στοιχεία, μπορεί να μετατραπεί σε ένα σύνολο κανόνων, όπου η κεφαλή του καθενός αποτελείται από μόνο ένα στοιχείο.

Τα στοιχεία (atoms) μπορεί να είναι:

- Έννοιες της οντολογίας
- Συσχετίσεις της οντολογίας
- SameAs συσχετίσεις
- DifferentFrom συσχετίσεις
- Built – Ins

Ένας τυπικός κανόνας SWRL είναι ο εξής:

$$hasParent(?x,?y) \wedge hasChild(?y,?z) \Rightarrow hasSibling(?x,?z),$$

ο οποίος διαβάζεται ως εξής: για κάθε (x) που έχει γονιό (y) και ο (y) πού έχει ένα παιδί (z), τότε τα (x) και (z) είναι αδέρφια.

Παρακάτω φαίνεται ο κανόνας:

$Person(y) \wedge hasParent(x,y) \wedge hasConsort(y,z) \rightarrow hasParent(x,z)$, όπως αυτός αναπαρίσταται σε μια οντολογία:

```
  <swrl:Imp rdf:ID="Def-hasParent">
    <swrla:isRuleEnabled
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</swrla:isRuleEnabled>
    <swrl:body>
      <swrl:AtomList>
        <rdf:first>
          <swrl:ClassAtom>
            <swrl:classPredicate rdf:resource="#Person"/>
            <swrl:argument1 rdf:resource="#y"/>
          </swrl:ClassAtom>
        </rdf:first>
        <rdf:rest>
          <swrl:AtomList>
            <rdf:rest>
              <swrl:AtomList>
                <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                <rdf:first>
                  <swrl:IndividualPropertyAtom>
                    <swrl:argument1 rdf:resource="#x"/>
                    <swrl:propertyPredicate rdf:resource="#hasParent"/>
                    <swrl:argument2 rdf:resource="#y"/>
                  </swrl:IndividualPropertyAtom>
                </rdf:first>
              </swrl:AtomList>
            </rdf:rest>
            <rdf:first>
              <swrl:IndividualPropertyAtom>
                <swrl:propertyPredicate rdf:resource="#hasConsort"/>
                <swrl:argument1 rdf:resource="#y"/>
                <swrl:argument2 rdf:resource="#z"/>
              </swrl:IndividualPropertyAtom>
            </rdf:first>
          </swrl:AtomList>
        </rdf:rest>
      </swrl:AtomList>
    </swrl:body>
    <swrl:head>
      <swrl:AtomList>
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        <rdf:first>
          <swrl:IndividualPropertyAtom>
            <swrl:argument1 rdf:resource="#x"/>
            <swrl:argument2 rdf:resource="#z"/>
            <swrl:propertyPredicate rdf:resource="#hasParent"/>
          </swrl:IndividualPropertyAtom>
        </rdf:first>
      </swrl:AtomList>
    </swrl:head>
  </swrl:Imp>
```

SWRL Σημασιολογία (SWRL Semantics)

Ένας SWRL κανόνας αντιμετωπίζεται σαν μια λογική συνέπεια ανάμεσα στο σώμα και την κεφαλή του. Δηλαδή όποτε οι συνθήκες στο σώμα του κανόνα ισχύουν, τότε και οι συνθήκες στην κεφαλή του πρέπει, επίσης, να ισχύουν. Η σημασιολογία των κανόνων προκύπτει ως μια προέκταση της ερμηνείας της OWL, στην όποια ορίζονται οι δεσμεύσεις (bindings). Τα bindings αντιστοιχίζουν τις μεταβλητές του κανόνα σε στοιχεία (elements) του πεδίου ορισμού της οντολογίας.

Ένας κανόνας ικανοποιείται αν και μόνο αν κάθε binding που ικανοποιεί το σώμα του κανόνα, ικανοποιεί επίσης και την κεφαλή του. Η ικανοποίηση του σώματος ή/και της κεφαλής του κανόνα απαιτεί την ικανοποίηση της σύζευξης των στοιχείων, από τα οποία αποτελείται.

Αποφασισιμότητα (Decidability)

Η γλώσσα OWL DL και οι γλώσσες κανόνων, όπως η SWRL, είναι γλώσσες αποφασίσιμες, όμως ο συνδυασμός τους οδηγεί σε μη αποφασισιμότητα [16, 17]. Ένας από τους βασικούς λόγους της μη αποφασισιμότητας, είναι πως στις γλώσσες κανόνων επιτρέπεται η χρήση ανώνυμων στοιχείων (anonymous individuals) στη διαδικασία της απόδειξης του κανόνα. Για να αποφευχθεί η μη αποφασισιμότητα, τα στοιχεία του κανόνα πρέπει να είναι DL – στοιχεία (DL – atoms), δηλαδή μόνο μοναδιαία ή δυαδικά στοιχεία επιτρέπονται, π.χ. OWL κλάσεις (OWL classes) ή OWL ιδιότητες (OWL properties). Για να είναι ασφαλής (safe) ένας κανόνας θα πρέπει κάθε μεταβλητή που εμφανίζεται στο σώμα του κανόνα να εμφανίζεται επίσης και σε ένα μη DL στοιχείο στο σώμα του κανόνα. Επιπλέον, μόνο οι μεταβλητές, που εμφανίζονται στο σώμα, μπορούν να εμφανίζονται στην κεφαλή του. Για παράδειγμα:

$hasParent(?x,?y) \wedge hasChild(?y,?z) \Rightarrow hasSibling(?x,?z)$, αυτός ο κανόνας είναι DL– unsafe, αφού οι μεταβλητές x, y, z στο σώμα του κανόνα εμφανίζονται μόνο σε DL στοιχεία και όχι και σε μη DL στοιχεία. Αυτός ο κανόνας μετατρέπεται σε DL – safe ως εξής:

$hasParent(?x,?y) \wedge hasChild(?y,?z) \wedge O(?x) \wedge O(?y) \wedge O(?z) \Rightarrow hasSibling(?x,?z)$, με την εισαγωγή του μη DL στοιχείου «O» στον κανόνα.

ΚΕΦΑΛΑΙΟ 3

ΕΡΓΑΛΕΙΑ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΓΝΩΣΗΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΟΥ

3.1 Εργαλεία Ανάπτυξης Οντολογιών

Η δημιουργία μιας οντολογίας προϋποθέτει την αναζήτηση του κατάλληλου λογισμικού σύνταξης. Μερικά βασικά χαρακτηριστικά ενός τέτοιου λογισμικού είναι τα ακόλουθα:

- **Εφαρμογή**: Δυνατότητα σύνταξης οντολογιών οποιουδήποτε πεδίου γνώσης.
- **Μεθοδολογία**: Επειδή η διαδικασία δημιουργίας μιας οντολογίας είναι χρονοβόρα, η μεθοδολογία συμβάλλει στη δημιουργία, τμηματοποίηση και συνάφεια της έννοιας και αποτρέπει την υπερβολική επεξεργασία της (υποβαθμίζει το βαθμό αφαίρεσης). Μια οντολογία είναι χρήσιμη, αν μπορεί να επαναχρησιμοποιηθεί για τον ορισμό άλλων οντολογιών.
- **Διαλειτουργικότητα**: Υποστήριξη πολλαπλών αναπαραστάσεων οντολογιών. Μια οντολογία μπορεί να χρησιμοποιηθεί από πολλές άλλες οντολογίες σε διαφορετικές μορφές αναπαράστασης.
- **Δυνατότητα εξαγωγής συμπερασμάτων**: Δυνατότητα συμπερασμού νέας γνώσης.

Λογισμικά δημιουργίας οντολογιών έχουν δημιουργηθεί πολλά. Στη συνέχεια αναφέρονται μερικά από τα πιο γνωστά.

Protégé

Το Protégé [19] είναι το πλέον διαδεδομένο ανοιχτού κώδικα (open source) λογισμικό επεξεργασίας οντολογιών. Προσφέρει ένα αυξανόμενο αριθμό εργαλείων για τη δημιουργία μοντέλων πεδίου ορισμού και εφαρμογές αναπαράστασης γνώσης με οντολογίες. Υλοποιεί ένα σύνολο από δομές και λειτουργίες υλοποίησης γνώσης, που υποστηρίζουν τη δημιουργία, την παρουσίαση και την επεξεργασία των οντολογιών σε πολλές μορφές αναπαράστασης. Υποστηρίζει υψηλή εκφραστικότητα, με αποτέλεσμα να μπορούν να μοντελοποιηθούν πολύπλοκες έννοιες και δυνατότητα πολύπλοκων ερωτημάτων συγκριτικά με άλλα λογισμικά. Μπορεί να επεκταθεί μέσω plug – ins, π.χ. να προσφέρει γραφική αναπαράσταση οντολογιών για την επεξεργασία τους. Πολλές δυνατότητες είναι διαθέσιμες μέσω Java APIs.

SWOOP

Το SWOOP [20] είναι ένα λογισμικό επεξεργασίας OWL οντολογιών υλοποιημένο σε Java. Έχει σχεδιαστεί σύμφωνα με τις συστάσεις του W3C και προσφέρει δυνατότητα συμπερασμού νέας γνώσης (Pellet). Μια άλλη λειτουργία είναι η δυνατότητα σύγκρισης, επεξεργασίας και συγχώνευσης διαφορετικών οντολογιών. Ένα σημαντικό χαρακτηριστικό του είναι ο έλεγχος συμβατότητας έκδοσης λογισμικού και μορφής οντολογίας. Αν προκύψουν προβλήματα συμβατότητας της οντολογίας με την τρέχουσα έκδοση του, αναζητείται μια παλιότερη συμβατή έκδοση. Προσφέρει επίσης δυνατότητα τμηματοποίησης και ορισμού της ιεραρχίας κλάσεων/ιδιοτήτων, όπως επίσης και δυνατότητα έκφρασης μόνο απλών ερωτημάτων (SWOOP query tool). Η εκφραστικότητα, που προσφέρει, δεν είναι αρκετά μεγάλη, γι' αυτό μπορεί να υποστηρίξει κλάσεις μόνο μικρού ή μεσαίου μεγέθους. Οι οντολογίες μπορεί να είναι σε μορφές OWL, XML, RDF ή και απλό κείμενο. Επίσης, τα αποτελέσματα μπορεί να είναι διαθέσιμα και σε HTML μορφή.

OntoEdit

Το OntoEdit [21] αποτελεί τμήμα του OntoStudio και βασίζεται σε IBM Eclipse. Είναι ένα εργαλείο σχεδιασμού και συντήρησης μεγάλων οντολογιών με πολύπλοκη δομή. Υποστηρίζει πολλές γλώσσες ανάπτυξης. Βασίζεται σε μια ανοικτή plug-in δομή. Κάθε plug – in παρέχει διαφορετικά χαρακτηριστικά για την ικανοποίηση των απαιτήσεων της οντολογίας. Τα δεδομένα των κλάσεων, των σχέσεων και των στιγμιοτύπων μπορούν να είναι διαφορετικών μορφών, όπως OXML, F – Logic, RDF/RDFS, OWL. Επίσης, η εισαγωγή των δεδομένων μπορεί να προέρχεται και από βάσεις δεδομένων κάνοντας πιο εύκολη την μετατροπή τους σε οντολογίες. Η ολοκληρωμένη έκδοση του είναι διαθέσιμη μόνο ως εμπορικό προϊόν.

pOWL

Το pOWL [22] προσφέρει μια PHP και Web-based λύση για την επεξεργασία και διαχείριση των οντολογιών. Προσφέρει δυνατότητα επεξεργασίας, παρουσίασης RDFS/OWL οντολογιών οποιοδήποτε μεγέθους, δυνατότητα έκφρασης ερωτημάτων (RDQL query builder), όπως επίσης και δυνατότητα αναζήτησης στοιχείων (resources). Όλη η λειτουργικότητα είναι προσπελάσιμη μέσω μιας διεπαφής (API) και η πρόσβαση πρέπει να πιστοποιηθεί. Τα μοντέλα αποθηκεύονται σε πίνακες Βάσεων δεδομένων (MySQL, PostgreSQL, Oracle, κ.α) και φορτώνονται στη μνήμη μόνο τα τμήματα του μοντέλου, που χρειάζονται κάθε φορά, εξασφαλίζοντας μια γρήγορη απόκριση. Οι

μορφές της οντολογίας, που υποστηρίζει, είναι παραλλαγές RDF (XML, N3, N – triples). Επίσης, τα αποτελέσματα μπορεί να είναι διαθέσιμα και σε HTML μορφή. Είναι εύκολο στη χρήση. Το βασικό του πρόβλημα, είναι ότι δεν υποστηρίζει αρκετά τις προτυποποιήσεις του W3C.

OilEd

Το OilEd [23] είναι ένα εργαλείο σύνταξης οντολογιών σε γλώσσα DAML+OIL, μια γλώσσα που προέκυψε από την OWL. Η πρόσφατη έκδοση του λογισμικού δεν προσφέρει ένα πλήρες περιβάλλον ανάπτυξης οντολογιών, αλλά προσφέρει αρκετή λειτουργικότητα, που επιτρέπει τους χρήστες να δημιουργήσουν οντολογίες και να χρησιμοποιήσουν το εργαλείο συμπερασμού, FaCT, για τον έλεγχο της συνέπειας των οντολογιών. Δεν μπορεί να διαχειριστεί περιπτώσεις, που υπάρχουν «κύκλοι» στην ιεραρχία, π.χ. ύπαρξη ισοδύναμων εννοιών (equivalent concepts). Η μορφή των εισαγόμενων δεδομένων μπορεί να γίνεται σε μορφές DAML+OIL, OWL, RDF/XML ή/και OIL, όμως η μορφή των εξαγόμενων οντολογιών μπορεί να είναι μόνο της μορφής DAML+OIL. Δεν είναι ένα ευέλικτο λογισμικό λόγω του περιορισμού στις εισαγόμενες/εξαγόμενες μορφές της οντολογίας.

Παρακάτω εμφανίζονται συνοπτικά κάποια χαρακτηριστικά των λογισμικών που αναφέρθηκαν παραπάνω (Πίνακας 3.1).

Πίνακας 3.1: Συγκριτικός Πίνακας Λογισμικών Επεξεργασίας Οντολογιών

Λογισμικό/ Χαρακτηριστικό	Εφαρμογή	Εκφραστικότητα	Μορφές Δεδομένων	Εργαλεία Συμπερασμού
Protégé	Μεγάλο Πεδίο Εφαρμογών	Υψηλή	RDF(S), OWL, XML Schema	Pellet, Fact++, RacerPro, DIG, Hermit, κ.α
SWOOP	Μεγάλο Πεδίο Εφαρμογών	Χαμηλή	OWL, XML, RDF + HTML	Pellet
OntoEdit	Μεγάλο Πεδίο Εφαρμογών	Υψηλή	OXML, F – Logic, RDF/RDFS, OWL	Pellet
OilEd	Περιορισμένο Πεδίο Εφαρμογών	Χαμηλή	DAML+OIL	FaCT
pOWL	Περιορισμένο Πεδίο Εφαρμογών	Χαμηλή	Παραλλαγές RDF (XML, N3, N – triples) + HTML	

3.2 Μηχανές Συμπερασμού

Για την εξαγωγή συμπερασμάτων σε εφαρμογές του Σημασιολογικού Ιστού, υπάρχουν δύο βασικοί μηχανισμοί: οι μηχανές συμπερασμού (reasoning engines) και οι μηχανές κανόνων (rules engines). Η διαφοροποίησή τους έγκειται στον τρόπο λειτουργίας τους και στο είδος των νέων συμπερασμάτων που μπορούν να παράγουν.

3.2.1 Μηχανές κανόνων

Μια μηχανή κανόνων περιλαμβάνει μια Βάση Γνώσης (Knowledge Base, KB) η οποία περιέχει τόσο το μοντέλο του κόσμου που μας ενδιαφέρει όσο και τους κανόνες με βάση

τους οποίους γίνεται η συλλογιστική (reasoning). Ο κάθε κανόνας έχει τη μορφή, που περιγράφηκε παραπάνω (βλ. Κεφάλαιο 2). Πολλές φορές η κεφαλή ενός κανόνα αποτελεί το σώμα ενός άλλου, οπότε σε τέτοιες περιπτώσεις προκαλείται αλυσιδωτή επαλήθευση κανόνων.

Για εφαρμογές του Σημασιολογικού Ιστού έχουν αναπτυχθεί διάφορες μηχανές κανόνων. Οι πιο σημαντικές από αυτές είναι:

Jess

Η Jess [24] είναι μια μηχανή συμπερασμού με scripting περιβάλλον γραμμένο ολοκληρωτικά για την γλώσσα Java. Βασίζεται στο εργαλείο CLIPS και η αναπαράσταση των κανόνων στο Jess γίνονται με δύο τρόπους: Με σύνταξη LISP ή με σύνταξη JessML που στην ουσία είναι μια αναπαράσταση της σύνταξης LISP σε XML. Επίσης υπάρχει και η επέκταση OWLJessKB που είναι ικανή να χειρίζεται κανόνες με OWL βάσεις γνώσης.

SweetRules

Είναι μια πλατφόρμα μηχανών συμπερασμού που χρησιμοποιούν οντολογίες σημασιολογικού ιστού και βάσεις γνώσεων με κανόνες για την αναπαράσταση της γνώσης. Οι οντολογίες και οι κανόνες που χρησιμοποιούνται από τα βασικά συστατικά της πλατφόρμας αυτής βασίζονται σε διάφορες τεχνολογίες όπως η RuleML, η SWRL, η OWL και η RDF. Η πλατφόρμα προσφέρει δυνατότητες τόσο backwardchaining όσο και forward-chaining συμπερασμού όπως επίσης και δυνατότητες συγχώνευσης βάσεων κανόνων και οντολογιών.

Prova

Το όνομά της βγαίνει από τις λέξεις Prolog και Java αποτυπώνοντας πλήρως τα χαρακτηριστικά τους. Υποστηρίζει υψηλή εκφραστικότητα στην δήλωση κανόνων και με αυτό τον τρόπο, συνδυάζει τη φυσική σύνταξη κανόνων με περιβάλλοντα λογικού προγραμματισμού, όπως η Prolog. Τέλος, δίνει τη δυνατότητα χρήσης κατανεμημένων μεθόδων συμπερασμού βασιζόμενες στη τεχνολογία κινητών πρακτόρων.

3.2.2 Μηχανές Συμπερασμού (Reasoners)

Όσον αφορά τις μηχανές συμπερασμού (reasoners), που έγιναν πολύ δημοφιλείς και απαραίτητες κατά την υλοποίηση του Σημασιολογικού Ιστού, αυτές έχουν διαφορετικά χαρακτηριστικά και τρόπο λειτουργίας. Οι reasoners μπορούν να παρέχουν πολύ αποδοτικά κάποιες υπηρεσίες συμπερασμού, όπως ο έλεγχος συνέπειας της βάσης

γνώσης (consistency checking), η κατηγοριοποίηση των κλάσεων (classification) και ο υπολογισμός των κλάσεων στις οποίες ανήκει κάθε στιγμιότυπο (instance checking). Οι πιο δημοφιλείς μηχανές συμπερασμού είναι:

RacerPro

Ο RacerPro [27] αποτελεί την εμπορική έκδοση του λογισμικού Racer. Ο Racer (Renamed Abox and Concept Expression Reasoner) αποτέλεσε τον πρώτο reasoner για τη γλώσσα OWL που κυκλοφόρησε. Σήμερα είναι ο πιο διαδεδομένος reasoner και ένας από τους γρηγορότερους.

Η μηχανή Racer χρησιμοποιεί μια βελτιωμένη έκδοση του tableau calculus για πολύ εκφραστικές Description Logics, όπως είναι η OWL-DL. Οι γλώσσες που μπορεί να υποστηρίξει προέρχονται από το χώρο των Description Logics και περιλαμβάνουν, μεταξύ άλλων, ιεραρχίες εννοιών και σχέσεων, τελεστές αριθμητικών περιορισμών, συμμετρικές και μεταβατικές σχέσεις. Ο συμπερασμός νέας γνώσης προκύπτει λόγω της σαφώς ορισμένης σημασιολογίας της εκάστοτε DL γλώσσας.

Μερικές βασικές λειτουργίες που υποστηρίζει η μηχανή Racer είναι οι ακόλουθες:

- Έλεγχος συνέπειας μιας οντολογίας (Consistency checking)
- Εντοπισμός έμμεσων υποκλάσεων (Classification of taxonomy)
- Τοποθέτηση στιγμιότυπων σε άλλες κλάσεις (Individual Inference)
- Αλγεβρικός συμπερασμός (Algebraic Reasoning)

Τέλος, αξίζει να σημειωθεί πως η μηχανή Racer δεν υποθέτει πως τα στιγμιότυπα με διαφορετικά ονόματα είναι μεταξύ τους διαφορετικά. Αυτό είναι ένα βασικό συστατικό ενός αποτελεσματικού reasoner για οντολογίες, καθώς οι οντολογίες λειτουργούν, όπως έχουμε δει, με την υπόθεση του κλειστού κόσμου. Το χαρακτηριστικό αυτό της μηχανής Racer σημαίνει πως είναι δυνατό να επιτρέψει την ταύτιση δύο στιγμιότυπων με διαφορετικό όνομα, αν φυσικά αυτό προκύψει από τη διαδικασία συμπερασμού.

FaCT++

Ο FaCT++ [28] αποτελεί μετεξέλιξη του FaCT [25] ικανή να υποστηρίξει συμπερασμό για τη γλώσσα OWL-DL. Οι υπηρεσίες που προσφέρει είναι παρόμοιες με του RacerPro. Συγκεκριμένα, είναι βασισμένο σε tableau αλγορίθμους, υποστηρίζοντας ένα πλήθος από επιπρόσθετα χαρακτηριστικά (π.χ., υποστήριξη nominals). Ωστόσο, το μεγαλύτερο μειονέκτημα του Fact++ αποτελεί η αδυναμία του για πλήρη συμπερασμό πάνω από ABox.

Pellet

Η μηχανή συμπερασμού Pellet [29] είναι υλοποιημένη στη γλώσσα Java και αποτελεί λογισμικό ανοικτού κώδικα (open-source) ικανό να διαχειριστεί οντολογίες υψηλής εκφραστικότητας. Πιο συγκεκριμένα, είναι βασισμένη σε βελτιστοποιημένους tableau αλγορίθμους υπεύθυνους για το χειρισμό βάσεων γνώσης εκφρασμένες σε περιγραφικές λογικές. Παράλληλα, υποστηρίζει ένα σύνολο από πρόσθετα χαρακτηριστικά, όπως υποστήριξη UNA, συλλογιστική βασισμένη σε CWA, εκτέλεση SPARQL [26] ερωτημάτων κ.λ.π. Επίσης, η μηχανή Pellet περιέχει ένα μηχανισμό επεξήγησης που έχει σκοπό τη διευκόλυνση της διαδικασίας σχεδιασμού και ανάπτυξης οντολογιών, μέσω του αποτελεσματικού και γρήγορου εντοπισμού λαθών και ασυνεπειών της βάσης γνώσης. Σε αντίθεση, λοιπόν, με τις περισσότερες μηχανές συμπερασμού για περιγραφικές λογικές, η μηχανή Pellet όχι μόνο εντοπίζει σημεία ασυνέπειας αλλά την αίτια που οδήγησε τη βάση γνώσης σε μη-ικανοποιησιμότητα (unsatisfiability). Πιο αναλυτικά, ο Pellet παρέχει στο χρήστη πρόσθετη γνώση, όπως αξιώματα (axioms) και περιορισμούς (restrictions), ώστε να διευκολύνει την επίλυση του προκληθέντος προβλήματος. Τέλος, η συγκεκριμένη μηχανή συμπερασμού επιτρέπει τη χρήση τύπων δεδομένων (datatypes) προδιαγραμμένων από το συντακτικό της γλώσσας XML, αλλά και ορισμένων από το χρήστη (user-defined).

Jena2

Το Jena2 [30] αποτελεί ένα ολοκληρωμένο εργαλείο ανάπτυξης και διαχείρισης γνώσης εκφρασμένης σε γλώσσες αναπαράστασης του Σημασιολογικού Ιστού. Συγκεκριμένα, το εργαλείο αυτό προσφέρει μία προγραμματιστική διεπαφή για το χειρισμό της γνώσης και επιτρέπει την εκτέλεση διαδικασιών συμπερασμού. Ωστόσο, οι υποστηριζόμενες διαδικασίες συμπερασμού είναι αρκετά περιορισμένες.

HermiT

Η μηχανή συμπερασμού HermiT [31] είναι υλοποιημένη στη γλώσσα Java και αποτελεί λογισμικό ανοικτού κώδικα (open-source) ικανό να διαχειριστεί οντολογίες υψηλής εκφραστικότητας. Είναι βασισμένη σε «hypertableu calculus» αλγορίθμους. Παρέχει πολύ αποδοτικότερο μηχανισμό συμπερασμού, με αποτέλεσμα η ταξινόμηση των οντολογιών να γίνεται γρηγορότερα. Είναι η μοναδική μηχανή, που μπορεί να ταξινομήσει ένα σύνολο από οντολογίες, κάτι που φαινόταν πολύ πολύπλοκο για όλες τις προηγούμενες μηχανές συμπερασμού. Μπορεί να χειριστεί OWL οντολογίες, για τις οποίες μπορεί να αποφασίσει αν είναι ή όχι συνεπείς και να αποδείξει επιπλέον σχέσεις

ανάμεσα στις κλάσεις. Μπορεί να χειριστεί DL safe κανόνες, η εισαγωγή των οποίων μπορεί να γίνει είτε μέσω της οντολογίας εισόδου είτε μέσω κάποιου OWL API.

Παρακάτω εμφανίζονται συνοπτικά κάποια χαρακτηριστικά των μηχανών συμπερασμού που αναφέρθηκαν παραπάνω (Πίνακας 3.2).

Πίνακας 3.2: Συγκριτικός Πίνακας Μηχανών Συμπερασμού

Μηχανή/ Χαρ/στικό	OWL - DL	Εκφρ/τητα	Αλγόριθμος	Έλεγχος συνέπειας	Υποστή-ριξη κανόνων
SweetRules	**Όχι**		**Rule – based**	**Όχι**	**SWRL, RuleML, Jess**
RacerPro	√	**SHIQ(D-)**	**Tableau**	√	**SWRL – μη πλήρης**
FaCT++	√	**SROIQ(D)**	**Tableau**	√	
Pellet	√	**SROIQ(D)**	**Tableau**	√	**SWRL – DL safe κανόνες**
Jena2	**Όχι πλήρης**	**Ελλιπής για DL**	**Rule – based**	**Ελλιπής για DL**	**Δική της μορφή κανόνων**
HermiT	√	**SHOIQ+**	**Hypertableau**	√	**SWRL – DL safe κανόνες**

ΚΕΦΑΛΑΙΟ 4

ΑΝΑΚΑΛΥΨΗ ΓΝΩΣΗΣ ΑΠΟ ΤΟ ΣΗΜΑΣΙΟΛΟΓΙΚΟ ΙΣΤΟ

4.1 Ανακάλυψη Γνώσης

Πριν αναφερθούμε στην ανακάλυψη γνώσης από το Σημασιολογικό Ιστό παρακάτω παρουσιάζονται συνοπτικά οι κυριότερες μεθοδολογίες Ανακάλυψης Γνώσης και οι κατηγορίες προβλημάτων, που ανήκουν σε κάθε μία από αυτές.

4.1.1 Προτασιακή Ανακάλυψη Γνώσης

Στην Προτασιακή Ανακάλυψη Γνώσης (Propositional Data Mining) η είσοδος της διαδικασίας είναι ένα σύνολο ανεξάρτητων παραδειγμάτων της έννοιας που πρόκειται να μελετηθεί. Τα παραδείγματα χαρακτηρίζονται από ένα σύνολο γνωρισμάτων. Σ' αυτήν την κατηγορία έχουν προταθεί πολλοί αλγόριθμοι, που δημιουργούν διαφορετικά μοντέλα εξόρυξης δεδομένων και έχουν διαφορετικές αναπαραστάσεις γνώσης [34, 39, 40]. Οι αλγόριθμοι αυτοί μπορούν να χωριστούν στις παρακάτω βασικές κατηγορίες:

Μάθηση με επίβλεψη

Η Μάθηση με επίβλεψη (Supervised Learning) αποτελεί ίσως τη δημοφιλέστερη διαδικασία εκπαίδευσης. Ονομάζεται επίσης ταξινόμηση (classification) ή επαγωγική μάθηση (inductive learning) και περιλαμβάνει την εκπαίδευση του μοντέλου μέσω παραδειγμάτων. Όλα τα παραδείγματα, που συμμετέχουν στη διαδικασία, έχουν ένα σύνολο χαρακτηριστικών. Η κλάση στην οποία ανήκει κάθε παράδειγμα εκπαίδευσης είναι γνωστή εκ των προτέρων. Για κάθε κλάση παραδειγμάτων η μέθοδος δημιουργεί ένα γενικό κανόνα που τη χαρακτηρίζει και στη συνέχεια τοποθετεί σε μια από τις κλάσεις ένα παράδειγμα, που δεν είναι γνωστή η κλάση στην οποία ανήκει εκ των προτέρων.

Οι πιο γνωστοί εκπρόσωποι αυτής της κατηγορίας είναι τα δέντρα απόφασης (decision trees), οι μέθοδοι που βασίζονται στον κανόνα του Bayes (Naïve Bayes μέθοδος, Bayesian Belief Networks) και οι μέθοδος των Support Vector Machines (SVM) [39].

Μάθηση χωρίς επίβλεψη

Ο στόχος της Μάθησης χωρίς επίβλεψη (Unsupervised Learning) είναι η τμηματοποίηση (partitioning) ενός συνόλου παραδειγμάτων σε ομάδες (clusters), έτσι ώστε τα παραδείγματα που ανήκουν σε μια ομάδα να είναι περισσότερο όμοια μεταξύ τους από ότι είναι με τα παραδείγματα των άλλων ομάδων. Οι αλγόριθμοι αυτής της

κατηγορίας εντοπίζουν κρυφές δομές και κανονικότητες στα δεδομένα με σκοπό την ομαδοποίησή τους.

Σε αντίθεση με τη μάθηση με επίβλεψη, στη μάθηση χωρίς επίβλεψη οι ομάδες δεν είναι προκαθορισμένες, ούτε υπάρχει κάποια πληροφορία που να δηλώνει τις επιθυμητές – έγκυρες σχέσεις μεταξύ των δεδομένων. Τέλος, η διαδικασία μπορεί να οδηγήσει σε διαφορετικές ομαδοποιήσεις ενός συνόλου δεδομένων, ανάλογα με το κριτήριο που χρησιμοποιείται για την ομαδοποίηση.

Οι παραλλαγές των μεθόδων αυτής της κατηγορίας ποικίλουν στις μετρικές ομοιότητας που χρησιμοποιούν (εντός μιας ομάδας και ανάμεσα στις ομάδες), στο πλήθος των ομάδων που δημιουργούν και αν επιτρέπουν ένα παράδειγμα να ανήκει σε περισσότερες από μια ομάδες. Ο πιο γνωστός αλγόριθμος αυτής της κατηγορίας είναι ο k – Means [63].

Κανόνες Συσχέτισης

Η εξαγωγή των κανόνων συσχέτισης (Association Rules) θεωρείται από τις σημαντικότερες διεργασίες ανακάλυψης γνώσης λόγω της εφαρμογής της σε πραγματικές εφαρμογές. Έχει προσελκύσει ιδιαίτερο ενδιαφέρον καθώς οι κανόνες συσχέτισης παρέχουν έναν συνοπτικό τρόπο για να εκφραστούν οι ενδεχομένως χρήσιμες πληροφορίες ώστε να γίνονται εύκολα κατανοητές από τους χρήστες. Οι κανόνες συσχέτισης ανακαλύπτουν κρυμμένες «συσχετίσεις» μεταξύ των γνωρισμάτων ενός συνόλου δεδομένων.

Οι κανόνες συσχέτισης ορίζονται ως εξής [61]:

Έστω **I** = $\{i_1, i_2, ..., i_m\}$ ένα σύνολο διακριτών αντικειμένων (items), και D ένα σύνολο από δοσοληψίες (transactions), όπου κάθε δοσοληψία T είναι ένα σύνολο αντικειμένων (itemset), για το οποίο ισχύει $T \subseteq I$. Μια δοσοληψία T *περιέχει* το X, ένα σύνολο από κάποια αντικείμενα του I, αν ισχύει $X \subseteq T$.

Ο κανόνας συσχέτισης είναι μια συσχέτιση της μορφής $X \to Y$, όπου $X \subseteq I$, $Y \subseteq I$ και $X \cap Y = \varnothing$. Το πρώτο μέλος του κανόνα ονομάζεται *υπόθεση* και το δεύτερο *συμπέρασμα*.

Ο κανόνας $X \to Y$ ισχύει στο σύνολο των δοσοληψιών D με *εμπιστοσύνη (confidence)* c, αν c% των δοσοληψιών στο D που περιέχουν το X περιέχουν επίσης και το Y. Ο κανόνας $X \to Y$ έχει *υποστήριξη (support) s*, αν το s% των δοσοληψιών στο D περιέχουν το $X \cup Y$.

Ο ορισμός της υποστήριξης μπορεί να γενικευτεί και για ένα itemset. Επομένως, ένα itemset X έχει υποστήριξη s, sup(X) = s, αν το s% των δοσοληψιών στο D περιέχουν το X, δηλαδή η υποστήριξη του X είναι αντίστοιχη με την πιθανότητα εμφάνισης του σε μια οποιαδήποτε δοσοληψία. Αν P(X) η πιθανότητα εμφάνισης σε μια οποιαδήποτε δοσοληψία, τότε ισχύει $\sup(X) = P(X)$.

Σύμφωνα, με τον παραπάνω ορισμό της υποστήριξης ενός itemset, οι ισοδύναμοι ορισμοί για την υποστήριξη και την εμπιστοσύνη ενός κανόνα συσχέτισης είναι οι ακόλουθοι:

Ο κανόνας $X \rightarrow Y$ έχει υποστήριξη s, όταν:

$$\sup (X \rightarrow Y) = \sup (X \cup Y) = P (X \cup Y) = s,$$

και εμπιστοσύνη c, όταν:

$$conf (X \rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)} = P(X \mid Y) = c .$$

Από τους ορισμούς γίνεται αντιληπτό, ότι η υποστήριξη δηλώνει πόσο συχνά εμφανίζονται τα itemsets του κανόνα, ενώ η εμπιστοσύνη δείχνει την ισχύ της συνεπαγωγής του κανόνα. Είναι προφανές ότι οι κανόνες $X \rightarrow Y$ και $Y \rightarrow X$ έχουν την ίδια υποστήριξη, αλλά ο κανόνας με τη μεγαλύτερη εμπιστοσύνη είναι περισσότερο χρήσιμος, καθώς η σχέση αιτίας και αποτελέσματος είναι πιο ισχυρή.

Ένα παράδειγμα υπολογισμού της υποστήριξης και της εμπιστοσύνης κανόνων φαίνεται παρακάτω:

Πίνακας 4.1: Σύνολο δοσοληψιών

ID	Items
1	Bread, Jelly, Butter
2	Bread, Butter
3	Bread, Milk, Butter
4	Beer, Bread
5	Beer, Milk

Θα εξετάσουμε έναν κανόνα συσχέτισης μεταξύ των αντικειμένων (items) Bread και Butter. Δηλαδή Bread $\rightarrow$ Butter. Σύμφωνα με τον πίνακα 4.1 το itemset {Bread, Butter}

εμφανίζεται σε 3 δοσοληψίες, ενώ το itemset {Bread} εμφανίζεται σε 4, οπότε σύμφωνα με τους ορισμούς που δόθηκαν παραπάνω ο κανόνας συσχέτισης Bread → Butter, έχει υποστήριξη s = (3/5) = 0.6 και εμπιστοσύνη c = (3/4) = 0.75.

Το πρόβλημα της Εξαγωγής Κανόνων Συσχέτισης αναφέρεται στην εύρεση όλων των κανόνων συσχέτισης που ικανοποιούν κάποια κατώφλια σε σχέση με την υποστήριξη (support) και την εμπιστοσύνη (confidence). Η τιμή της υποστήριξης ενός κανόνα πρέπει να είναι μεγαλύτερη από την ελάχιστη υποστήριξη (minsup), και η τιμή της εμπιστοσύνης πρέπει να είναι μεγαλύτερη από την ελάχιστη εμπιστοσύνη (minconf). Επομένως, οι κανόνες που θα προκύψουν πρέπει να ικανοποιούν αυτούς τους δύο περιορισμούς. Τα itemsets που έχουν υποστήριξη μεγαλύτερη από την minsup λέγονται συχνά σύνολα (frequent itemsets).

Συνεπώς, η εξαγωγή κανόνων συσχέτισης αποτελείται από δύο βήματα:

1. Εύρεση όλων των frequent items, δηλαδή των συνόλων από αντικείμενα που ικανοποιούν την απαίτηση για μεγάλη υποστήριξη
2. Εξαγωγή των κανόνων συσχέτισης που έχουν minconf.

Ο πιο γνωστός αλγόριθμος αυτής της κατηγορίας είναι ο Apriori [41, 61].

Ανακάλυψη ακολουθιών

Σε πολλές εφαρμογές οι εγγραφές των δεδομένων δεν είναι σύνολα στοιχείων αλλά ακολουθίες γεγονότων (sequences of events). Η ανακάλυψη ακολουθιών [42] (sequence discovery) επεκτείνει τη μέθοδο των κανόνων συσχέτισης ώστε να εντοπίζει ακολουθίες γεγονότων, που επαναλαμβάνονται στα δεδομένα. Οι ακολουθίες των γεγονότων είναι διατεταγμένες. Ένας κανόνας, που προκύπτει για μια συχνή ακολουθία, εκφράζει την πιθανότητα το τελευταίο στοιχείο της ακολουθίας να εμφανιστεί μετά την αλληλουχία των στοιχείων που έχει προηγηθεί. Αυτοί οι κανόνες χρησιμοποιούνται για την πρόβλεψη γεγονότων μιας δεδομένης σειράς παρατηρήσεων.

Όμως, οι κανόνες αυτής της κατηγορίας δεν είναι οι πλέον κατάλληλοι για την πρόβλεψη ενός γεγονότος. Για παράδειγμα, έστω ότι οι ακολουθίες «A – B – C - D» και «A – B – C - E» είναι συχνές. Αυτό σημαίνει ότι τα γεγονότα D και E είναι πολύ πιθανόν να προκύψουν μετά την ακολουθία A – B – C. Αν, όμως, ο σκοπός είναι ο εντοπισμός της ακολουθίας, που οδηγεί στο γεγονός E, τότε η ακολουθία «A – B – C» δεν κάνει σωστή πρόβλεψη, γιατί η ίδια ακολουθία μπορεί να οδηγήσει και στο γεγονός D. Παρόλα αυτά, έχουν αναπτυχθεί μέθοδοι που βασίζονται στην ανακάλυψη ακολουθιών για την πρόβλεψη γεγονότων [43].

4.1.2 Σχεσιακή Ανακάλυψη Γνώσης

Οι απαραίτητες προϋποθέσεις της ανεξαρτησίας και της ομοιομορφίας των δεδομένων, που ισχύουν στην προτασιακή ανακάλυψη γνώσης, δεν είναι πάντα εφικτές. Οι πολύπλοκες συσχετίσεις των δεδομένων, που πρέπει να ληφθούν υπόψη για την επίλυση γενικότερων και περισσότερο πολύπλοκων προβλημάτων, δεν επιτρέπουν την εφαρμογή της προτασιακής ανακάλυψης γνώσης, με αποτέλεσμα να απαιτείται η εφαρμογή της σχεσιακής ανακάλυψης γνώσης (Relational Data Mining).

Στη σχεσιακή ανακάλυψη γνώσης αναιρούνται οι δύο προϋποθέσεις της παραδοσιακής διαδικασίας, τα παραδείγματα δεν είναι ομοιόμορφα, αλλά αποτελούν σύνολα ετερογενών εγγραφών, και επιτρέπεται η ύπαρξη εξαρτήσεων μεταξύ αυτών [35]. Οι μέθοδοι αυτής της κατηγορίας πρέπει να διαχειρίζονται την ανομοιομορφία και την εξάρτηση των δεδομένων. Οι περισσότεροι αλγόριθμοι σχεσιακών δεδομένων βασίζονται στον επαγωγικό λογικό προγραμματισμό (Inductive Logic Programming, ILP) [36], που δημιουργεί πρότυπα εκφρασμένα ως λογικά προγράμματα, υποσύνολο της λογικής πρώτης τάξης. Επειδή ο ορισμός λογικών προγραμμάτων είναι μια πολύπλοκη διαδικασία, η σχεσιακή ανακάλυψη γνώσης στρέφεται στην εισαγωγή πιθανοτικών αναπαραστάσεων (probabilistic representations) [38] στους αλγορίθμους.

Ο συνδυασμός της σχεσιακής ανακάλυψης γνώσης και της πιθανοτικής εκπαίδευσης αναφέρεται ως Στατιστική Σχεσιακή Μάθηση (Statistical Relational Learning, SRL) [37]. Η SRL σχετίζεται με την μοντελοποίηση πεδίων που παρουσιάζουν αβεβαιότητα (χειρίζονται με χρήση στατιστικών μεθόδων) και πολύπλοκες, σχεσιακές δομές. Τυπικά, οι φορμαλισμοί αναπαράστασης γνώσης της SRL χρησιμοποιούν Λογική Πρώτης Τάξης για τη γενική περιγραφή των σχεσιακών ιδιοτήτων του πεδίου και πιθανοτικά γραφικά μοντέλα (Bayesian networks, Markov networks) για τη μοντελοποίηση της αβεβαιότητας. Το πλεονέκτημα της SRL είναι η καλύτερη εφαρμογή σε πραγματικά προβλήματα ταξινόμησης, αφού συνδυάζει ποικίλους τρόπους αναπαράστασης γνώσης για τη μοντελοποίηση σχεσιακών, ανομοιογενών και ημιδομημένων δεδομένων.

Οι πιο γνωστοί αλγόριθμοι αυτής της κατηγορίας είναι ο Σχεσιακός Ταξινομητής Bayes (Relational Bayes Classifier, RBC), που επεκτείνει τον απλό Bayesian ταξινομητή και το Σχεσιακό Δέντρο Πιθανότητας (Relational Probability Tree, RPT), που επεκτείνει το τυπικό δέντρο αναγνώρισης.

4.2 Ανακάλυψη Γνώσης από το Σημασιολογικό Ιστό

Η ανακάλυψη γνώσης από το Σημασιολογικό Ιστό στοχεύει στη βελτίωση της ανακάλυψης γνώσης από τον Παγκόσμιο Ιστό (Web Mining) χρησιμοποιώντας επιπλέον σημασιολογικές δομές του Ιστού. Η πληροφορία του Σημασιολογικού Ιστού αποτελείται από μη ομοιογενή και, γενικά, αλληλοεξαρτώμενα δεδομένα. Ο Σημασιολογικός Ιστός προσφέρει επιπλέον υπονοούμενη (inferred) πληροφορία, που δεν υπάρχει στα σχεσιακά δεδομένα. Η εφαρμογή των μεθόδων της μηχανικής μάθησης στην πληροφορία του Σημασιολογικού Ιστού μπορεί να εντοπίσει χρήσιμες σχέσεις, που πιθανώς να βελτιώσουν το Σημασιολογικό Ιστό. Η πλειοψηφία των μεθόδων της μηχανικής μάθησης απαιτούν ως είσοδο τη μονοδιάστατη αναπαράσταση των χαρακτηριστικών (feature – vector representation), μια μορφή στην οποία δεν είναι εφικτή η μετατροπή των RDF δεδομένων. Από τις μεθόδους ανακάλυψης γνώσης, που αναφέρθηκαν στην προηγούμενη ενότητα, μόνο οι μέθοδοι της σχεσιακής ανακάλυψης γνώσης μπορούν να επιτύχουν την ανακάλυψη γνώσης από το περιεχόμενο και τη δομή του Σημασιολογικού Ιστού. Αυτές οι μέθοδοι μπορούν εύκολα να επεκταθούν ή/και να τροποποιηθούν, ώστε να διαχειριστούν δεδομένα, που περιγράφονται σε RDF ή σε οντολογίες. Μια αρχική προσπάθεια τροποποιήσεων περιλαμβάνει τη δημιουργία νέας μορφής αναπαράστασης γνώσης παρόμοια με τη Horn Λογική, μια μορφή πολύ συνηθισμένη στις τεχνικές ILP [44]. Επιπλέον, οι αλγόριθμοι, που βασίζονται σε πιθανοτικά μοντέλα, αποτελούν μια λύση για τα δεδομένα του Σημασιολογικού Ιστού εξαιτίας της ομοιότητάς τους με τα πιθανοτικά σχεσιακά μοντέλά.

Παρόλα αυτά, το μέγεθος των συνόλων των δεδομένων και η κατανεμημένη φύση των δεδομένων του Σημασιολογικού Ιστού είναι δύο παράγοντες, που επηρεάζουν την αποδοτικότητα των μεθόδων της σχεσιακής ανακάλυψης γνώσης. Τα μεγάλα σύνολα δεδομένων αποτελούν βασικό πρόβλημα για τους ILP αλγορίθμους. Με την αναμενόμενη ανάπτυξη του Σημασιολογικού Ιστού το πρόβλημα είναι αναμενόμενο να ενταθεί. Όμως, η απόδοση των αλγορίθμων μπορεί να βελτιωθεί με χρήση δειγματοληψίας [45].

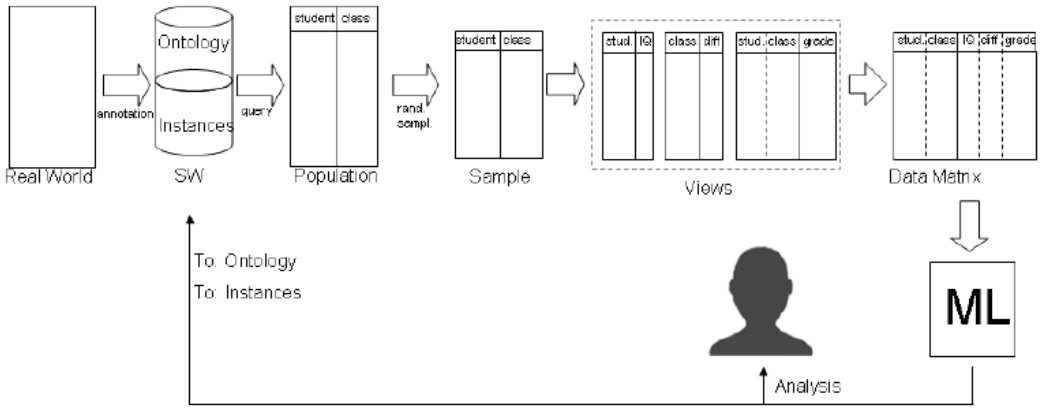
4.2.1 Μέθοδοι Ανακάλυψης Γνώσης από το Σημασιολογικό Ιστό

Παρακάτω παρουσιάζονται μερικοί βασικοί μέθοδοι της σχεσιακής ανακάλυψης γνώσης, που μπορούν να τροποποιηθούν ώστε να επιτευχθεί η ανακάλυψη γνώσης από το Σημασιολογικό Ιστό.

Στατιστική Σχεσιακή Μάθηση βασισμένη σε χαρακτηριστικά

Όπως αναφέρθηκε και προηγουμένως η μέθοδος SRL μπορεί να εφαρμοστεί σε δεδομένα του ΣΙ. Η στατιστική μάθηση βασισμένη σε χαρακτηριστικά (Feature – based Statistical Relational Learning) [46] στο Σημασιολογικό Ιστό εξαρτάται μόνο από το πλήθος των δειγμάτων στη δειγματοληψία και είναι ανεξάρτητη από το μέγεθος του. Πριν την εφαρμογή της δειγματοληψίας, είναι απαραίτητη η εύρεση των χαρακτηριστικών των δειγμάτων. Εξαιτίας της ιδιαιτερότητας των δεδομένων του Σημασιολογικού Ιστού, τα χαρακτηριστικά που θα επιλεγούν θα πρέπει να μπορούν να εκφράσουν τη σχεσιακή δομή της πληροφορίας. Η δειγματοληψία μπορεί να επιτευχθεί με χρήση μηχανών αναζήτησης ή crawlers. Στη συνέχεια γίνεται ο διαχωρισμός των χαρακτηριστικών σε διαφορετικούς πίνακες – όψεις (views), ένας πίνακας για κάθε χαρακτηριστικό. Τέλος, δημιουργείται ένας ενιαίος πίνακας για όλες τις όψεις (Data Matrix) πάνω στον οποίο θα εφαρμοστεί μια μέθοδος Μηχανικής Μάθησης για την ανακάλυψη γνώσης από τα δεδομένα του Σημασιολογικού Ιστού (Εικόνα 4.1). Η μέθοδος μετασχηματίζει τα δεδομένα του ΣΙ σε μονοδιάστατη μορφή, με αποτέλεσμα να είναι εφικτή η εφαρμογή κλασσικών μεθόδων ανακάλυψης γνώσης.

Το βασικό πρόβλημα της μεθόδου είναι, ότι πολλά δεδομένα μπορεί να είναι ελλιπή. Για αυτό το λόγο γίνεται η υπόθεση του κλειστού κόσμου (closed – world assumption) ορίζοντας ως κόσμο, μόνο ότι περιγράφεται από το δείγμα, με αποτέλεσμα μόνο οι ισχυρισμοί, που είναι γνωστοί ή μπορούν να παραχθούν, να είναι αληθείς και όλοι οι υπόλοιποι να θεωρούνται ψευδείς.



Εικόνα 4.1: Στατιστική Μάθηση στο Σημασιολογικό Ιστό

Επαγωγικός Λογικός Προγραμματισμός

Οι αλγόριθμοι του ILP μπορούν, επίσης, να εφαρμοστούν στα δεδομένα του Σημασιολογικού Ιστού. Ο ILP ταιριάζει πολύ καλά στο ντετερμινιστικό πλαίσιο του Σημασιολογικού Ιστού. Η στατιστική μάθηση που περιγράφηκε παραπάνω σχετίζεται με το ILP, η μόνη διαφορά είναι, ότι η προηγούμενη μέθοδος βασίζεται σε ένα δείγμα των δεδομένων. Ένα πλεονέκτημα του ILP είναι η παραγωγή προτάσεων (clauses), οι οποίες μπορούν να ολοκληρωθούν με κάποιους περιορισμούς σε SWRL (βλ. Κεφάλαιο 2). Το μειονέκτημα των ILP αλγορίθμων είναι η χρήση πολύπλοκων μεθόδων εντοπισμού του βέλτιστου σώματος του κανόνα, με αποτέλεσμα ο χρόνος της εκπαίδευσης να αυξάνεται. ILP περιπτώσεις στις οποίες εφαρμόζονται μεθοδολογίες μάθησης σε Περιγραφικές Λογικές περιγράφονται εδώ [47, 48].

Σχεσιακά Γραφικά Μοντέλα

Σε αντίθεση με τις δύο προηγούμενες μεθόδους που ανακαλύπτουν τις λογικές εξαρτήσεις μεταξύ των χαρακτηριστικών των δεδομένων του ΣΙ, τα σχεσιακά γραφικά μοντέλα (Relational Graphical Models, RGM) [46] προβλέπουν τις πραγματικές τιμές όλων των δηλώσεων (statements) (RDF – triples) στο Σημασιολογικό Ιστό. Τα RGM είναι πιθανοτικά μοντέλα και οι δηλώσεις αναπαρίστανται από τυχαίες μεταβλητές. Μπορούν να θεωρηθούν βελτιωμένη εκδοχή των κανονικών γραφικών μοντέλων, π.χ. Bayesian Networks, Markov Networks. Δημιουργήθηκαν στα πλαίσια των σχεσιακών μοντέλων δεδομένων και της λογικής πρώτης τάξης, αλλά η βασική ιδέα μπορεί εύκολα να προσαρμοστεί στο μοντέλο δεδομένων του Σημασιολογικού Ιστού. Ίσως είναι η καλύτερη μέθοδος στην οποία οι εξαρτήσεις ανάμεσα στις μεταβλητές των δηλώσεων είναι σωστά μοντελοποιημένες. Το μειονέκτημα της μεθόδου είναι ότι οι υπολογιστικές

απαιτήσεις είναι ανάλογες των δηλώσεων των οποίων η σωστή τιμή είναι ήδη γνωστή ή του συνολικού αριθμού των πιθανώς σωστών δηλώσεων. Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί με τη δειγματοληψία και την εισαγωγή της βάσης γνώσης της οντολογίας (ontology background knowledge) με τον τρόπο που περιγράφηκε στην στατιστική μάθηση. Για παράδειγμα, αν η στατιστική μονάδα είναι ο «μαθητής», τότε τα δεδομένα δεν θα αντιστοιχούν σε ένα σύνολο χαρακτηριστικών, αλλά σε έναν υπογράφο προσαρμοσμένο στην στατιστική μονάδα «μαθητής». Όπως και στη στατιστική μάθηση με τη δειγματοληψία, ο χρόνος εκπαίδευσης είναι ουσιαστικά ανεξάρτητος από το μέγεθος του Σημασιολογικού Ιστού. Τα RGM δημιουργούν τυπικά υποθέσεις ανοικτού κόσμου (open world assumption).

Η πιο γνωστή μορφή των RGM είναι τα πιθανοτικά σχεσιακά μοντέλα (Probabilistic Relational Models, PRM) [46]. Τα PRM συνδυάζουν ένα πλαίσιο λογικής αναπαράστασης (frame – based logical representation) με πιθανοτική σημασιολογία βασισμένη σε κατευθυνόμενα γραφικά μοντέλα. Οι κόμβοι των PRM μοντελοποιούν την πιθανότητα κατανομής των χαρακτηριστικών των αντικειμένων (object attributes) των οποίων οι σχέσεις μεταξύ των αντικειμένων θεωρούνται γνωστές. Προφανώς, αυτή η υπόθεση απλοποιεί πολύ τη μοντελοποίηση. Στα πλαίσια του ΣΙ τα χαρακτηριστικά των αντικειμένων πρέπει αρχικά να αντιστοιχούν σε αντικείμενο – στοιχείο δηλώσεις (object – to – literal statements). Σε εξελιγμένη μορφή των PRM συμπεριλαμβάνεται και η περίπτωση οι σχέσεις μεταξύ των αντικειμένων να θεωρούνται άγνωστες. Η περίπτωση αυτή ονομάζεται δομική αβεβαιότητα (structural uncertainty) στα PRM. Για κάποια PRM, η κανονικότητα στη δομή τους μπορεί να χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων. Τα μεγάλα PRM απαιτούν προσεγγιστική εξαγωγή συμπερασμάτων. Η διαδικασία της εκπαίδευσης στα PRM βασίζεται στην εμπειρική διαδικασία εκπαίδευσης κατά Bayes. Η δομική διαδικασία εκπαίδευσης τυπικά χρησιμοποιεί στρατηγική άπληστης αναζήτησης (greedy search strategy), όμως θα πρέπει να υπάρχει η εγγύηση ότι ο γράφος δεν περιλαμβάνει κατευθυνόμενους κύκλους.

4.3 Εφαρμογές Ανακάλυψης Γνώσης στο Σημασιολογικό Ιστό

Ο Σημασιολογικός Ιστός προσφέρει μια καλή βάση για τον εμπλουτισμό της διαδικασίας της ανακάλυψης γνώσης στον Ιστό (Web Mining) [54]. Ο τύπος των συνδέσμων (hyperlinks) περιγράφονται ακριβώς συμβάλλοντας στην ανακάλυψη γνώσης και μέσω της δομής του Ιστού (Web structure mining). Επίσης, η τυποποιημένη μορφή των δεδομένων του Σημασιολογικού Ιστού έχει δύο συνέπειες στη διαδικασία της

ανακάλυψης γνώσης. Η πρώτη είναι, ότι μεγάλο τμήμα της πληροφορίας έχει αποκτήσει δομή, με αποτέλεσμα η εφαρμογή των μεθόδων ανακάλυψης γνώσης να είναι εφικτή με ελάχιστες τροποποιήσεις, και επιπλέον η χρήση της τυποποιημένης πληροφορίας (ιεραρχία εννοιών σε RDF ή αναπαράσταση γνώσης σε OWL) σε συνδυασμό με τα δεδομένα του Ιστού μπορούν να χρησιμοποιηθούν στη διαδικασία της ανακάλυψης γνώσης.

Παρακάτω εμφανίζονται μερικές βασικές εφαρμογές της ανακάλυψης γνώσης στο Σημασιολογικό Ιστό.

- ***Κατηγοριοποίηση Εγγράφου***

  Η διαδικασία της ταξινόμησης εγγράφου (Document Classification / Text Categorization) περιλαμβάνει την κατηγοριοποίηση του σε ένα σύνολο προκαθορισμένων κατηγοριών ανάλογα με το περιεχόμενο του. Η διαδικασία ταξινόμησής του στο Σημασιολογικό Ιστό είναι παρόμοια με τη διαδικασία στον Παγκόσμιο Ιστό [64]. Η διαφορά είναι ότι εκτός από τα γενικά χαρακτηριστικά του κειμένου, μπορούν να χρησιμοποιηθούν και οι σχολιασμοί (annotations) ως επιπλέον χαρακτηριστικά ή ως χαρακτηριστικά δομής. Η αναπαράσταση του αρχείου σε μορφή οντολογίας μπορεί να δημιουργήσει επιπλέον πληροφορία για το αρχείο προσφέροντας καλύτερη κατηγοριοποίηση. Αυτού του είδος η ταξινόμηση χρησιμοποιεί διαδικασία εκπαίδευσης με γνώση (background knowledge) και δημιουργία χαρακτηριστικών [49]. Η ταξινόμηση μπορεί να εφαρμοστεί και μόνο σε προκαθορισμένα τμήματα του αρχείου.

- ***Ομαδοποίηση Εγγράφου***

  Όπως και στην κατηγοριοποίηση έτσι και στην ομαδοποίηση εγγράφου (Document Clustering) οι σχολιασμοί (annotations) του αρχείου μπορούν να χρησιμοποιηθούν για την δημιουργία επιπλέον πληροφορίας που αφορά το αρχείο.

  Οι ομάδες των αρχείων (document clusters) και οι περιγραφές αυτών μπορούν να παρουσιαστούν ως οντολογίες. Για αυτό το λόγο, ιεραρχικοί μέθοδοι ομαδοποίησης δημιουργούν τις οντολογίες από αρχεία και στη συνέχεια πραγματοποιείται η συντήρηση τους ομαδοποιώντας νέα αρχεία στην ιεραρχία [65].

- ***Ανακάλυψη Γνώσης για Εξαγωγή Πληροφορίας***

  Στα πλαίσια του Σημασιολογικού Ιστού, η διαδικασία της εξαγωγής πληροφορίας (Information Retrieval) από τα αρχεία χρησιμοποιεί τους σχολιασμούς (annotations)

για την παραγωγή κανόνων. Οι υπάρχουσες οντολογίες μπορούν να υποστηρίξουν την επίλυση προβλημάτων συμπεριλαμβάνοντας την εκπαίδευση άλλων οντολογιών και την ανάθεση οντολογικών εννοιών σε κείμενο (text annotation). Αρκετή έρευνα έχει γίνει για την προσπάθεια δημιουργίας κανόνων ενός σχολιασμένου κειμένου.

Αυτό το είδος της ανακάλυψης πληροφορίας στοχεύει στην ανάθεση μιας ετικέτας (label) σε τμήμα του κειμένου [50]. Καθώς δεν είναι καθόλου εύκολη η απόκτηση ενός ήδη σχολιασμένου κειμένου, έχουν ανακαλυφθεί άλλες τεχνικές, όπως η επεξεργασία φυσικής γλώσσας για την εύρεση μονάδων κειμένου (π.χ. ομάδες ουσιαστικών, ομάδες προτάσεων) και την αντιστοίχηση αυτών σε έννοιες μιας υπάρχουσας οντολογίας [51].

- ***Αντιστοίχηση Οντολογιών (Ontology matching/alignment)***

  Επειδή οι οντολογίες δημιουργούνται για έναν συγκεκριμένο σκοπό, είναι αναπόφευκτο οι παρόμοιες οντολογίες να ενοποιούνται συνδυάζοντας τις βάσεις γνώσης τους [52, 53]. Αυτή η διαδικασία απαιτεί τη δημιουργία αντιστοίχησης των εννοιών (concepts), των χαρακτηριστικών (attributes), των τιμών (values) και των σχέσεων (relations) ανάμεσα στις δύο οντολογίες με σκοπό την ενοποίηση τους. Αρχικά, η πληροφορία των εννοιών προκύπτει από τις οντολογίες και η επιπλέον πληροφορία από τις σελίδες του Ιστού, που είναι σχετικές με κάθε έννοια. Η πληροφορία των ιστοσελίδων μπορεί να χρησιμοποιηθεί για την εκπαίδευση ενός ταξινομητή παραδειγμάτων (instances) μιας κλάσης. Ο ταξινομητής εφαρμόζεται στα παραδείγματα των εννοιών της άλλης οντολογίας, με σκοπό να αποδειχτεί αν κάποια από αυτές τις έννοιες είναι σχετική με την αρχική έννοια.

- ***Μοντελοποίηση χρηστών, Συστάσεις, Εξατομίκευση***

  Ο Σημασιολογικός Ιστός δίνει τη δυνατότητα για χρήση της διαδικασίας μάθησης (Semantic Web Usage Mining), επειδή οι οντολογίες παρέχουν πληροφορία για τις ενέργειες των χρηστών και τις ιστοσελίδες σε μια τυποποιημένη μορφή, με αποτέλεσμα να επιτυγχάνεται η ανακάλυψη χρήσιμων προτύπων (patterns) [54]. Για παράδειγμα, οι σχολιασμοί των προϊόντων, που ενδιαφέρουν και αγοράζουν οι πελάτες, προσθέτουν πληροφορία με αποτέλεσμα να είναι πιθανή η ανακάλυψη επιπλέον γενικών προτύπων. Τέτοια πρότυπα μπορούν να χρησιμοποιηθούν για την πρόβλεψη των αντιδράσεων των πελατών σε νέα προϊόντα. Αυτό ίσως να μην ήταν εφικτό, αν μόνο το όνομα, η εικόνα και η τιμή του προϊόντος ήταν διαθέσιμα

και η ανακάλυψη γνώσης θα ήταν περισσότερο αποτελεσματική αν χρησιμοποιούνταν μια ενιαία οντολογία αντί για τα αρχεία που περιγράφουν τα προϊόντα.

Εφαρμογές της ανακάλυψης γνώσης, όπως σύσταση, εξατομίκευση και ανάλυση συνδέσμων έχουν όφελος από τη χρήση σχολιασμένων αρχείων (annotated documents). Μόνο μερικά τεχνικά προβλήματα πρέπει να λυθούν για την επέκταση των υπαρχόντων μεθόδων. Η μεγάλης κλίμακας εφαρμογές χρειάζονται μεγαλύτερες οντολογίες, οι οποίες να διατηρούνται και να εφαρμόζονται ημιαυτόματα.

Το μεγαλύτερο πρόβλημα των τωρινών εφαρμογών του Σημασιολογικού Ιστού είναι η έλλειψη μεγάλου αριθμού οντολογιών και annotations. Παρόλο, που η πρόσβαση σε οντολογίες και δεδομένα μέσω διαδικτύου είναι εφικτή, οι υπάρχουσες εφαρμογές βασίζονται αποκλειστικά σε τοπικούς υπολογισμούς. Οι οντολογίες και τα παραδείγματα ενσωματώνονται και διατηρούνται τοπικά για μεγαλύτερη ταχύτητα. Οι εφαρμογές θα αντιμετώπιζαν πρόβλημα κλιμάκωσης, αν ο Σημασιολογικός Ιστός χρησιμοποιούταν σε μεγάλη κλίμακα. Το πρόβλημα μπορεί να αντιμετωπιστεί με κατανεμημένο υπολογισμό, χρησιμοποιώντας κατανεμημένες οντολογίες, παραδείγματα και γνώση. Αυτή η λύση συνδυάζει το Σημασιολογικό Ιστό και τον υπολογισμό πλέγματος (Grid Computing), η οποία είναι γνωστή ως Σημασιολογικό Πλέγμα (Semantic Grid).

4.4 Ιδιαιτερότητες της Ανακάλυψης Γνώσης στο Σημασιολογικό Ιστό

Παρακάτω συνοψίζονται κάποιες ιδιαιτερότητες από την εφαρμογή των μεθόδων ανακάλυψης γνώσης στο Σημασιολογικό Ιστό. Οι απαιτήσεις αναφέρονται σύμφωνα με τα κριτήρια της εφαρμογής (applicability), της επεκτασιμότητα (scalability) και της δυνατότητας χρήσης (usability).

Εφαρμογή

Η εφαρμογή των μεθόδων ανακάλυψης γνώσης στο Σημασιολογικού Ιστού, διευρύνει τις δυνατότητες έρευνας και εφαρμογής του πεδίου της μηχανικής μάθησης, καθώς επεκτείνεται και στα δεδομένα του Σημασιολογικού Ιστού. Υπάρχουν μέθοδοι ανακάλυψης γνώσης, όπως οι τεχνικές ILP, που βασίζονται σε λογικά προγράμματα για να πετύχουν τη διαδικασία μάθησης. Όμως, οι οντολογίες OWL και τα λογικά προγράμματα δεν έχουν την ίδια εκφραστικότητα, π.χ. υπάρχουν OWL οντολογίες, που δεν μπορούν να εκφραστούν σε Horn Λογική και αντίστροφα. Αυτό σημαίνει, ότι ο ίδιος

αλγόριθμος δεν μπορεί να εφαρμοστεί σε όλες τις περιπτώσεις. Για παράδειγμα, ένας περιορισμός είναι τα κατηγορήματα (predicates) με αριθμό στοιχείων (arity) μεγαλύτερο των δύο. Οι έννοιες (concepts) σε DL αντιστοιχούν σε predicates με arity ίσο με 1 και οι ρόλοι (roles) σε predicates με arity ίσο με 2, επομένως δεν είναι εύκολη η έκφραση κατηγορημάτων μεγαλύτερων arity.

Η ανακάλυψη γνώσης προβλημάτων εκφρασμένα σε DL μπορεί να επιτευχθεί με χρήση εκφραστικών φορμαλισμών (expressive formalisms), όπως η λογική πρώτης τάξης, εφόσον οι γλώσσες DL είναι υποσύνολα της λογικής πρώτης τάξης. Παρόλα αυτά, αυτή η διαδικασία δεν είναι αποτελεσματική εξαιτίας της υψηλής πολυπλοκότητας συμπερασμού και της μη – αποφασισιμότητας της Λογικής Πρώτης Τάξης.

Επεκτασιμότητα

Όπως έχει αναφερθεί και προηγουμένως, τα μεγάλα σύνολα δεδομένων αποτελούν βασικό πρόβλημα για τους αλγορίθμους ανακάλυψης γνώσης. Επομένως, η ανακάλυψη γνώσης από οντολογίες με μεγάλες βάσεις γνώσης δεν είναι καθόλου αποδοτική, βέβαια εξαρτάται από την πολυπλοκότητα των αξιωμάτων (axioms), που περιλαμβάνουν. Αυτό το πρόβλημα σε μερικές εφαρμογές αντιμετωπίζεται με την επιλογή των σχετικών υποσυνόλων γνώσης της βάσης, που θα συμμετέχουν στη διαδικασία της ανακάλυψης γνώσης.

Δυνατότητα χρήσης

Καμία μέθοδος ανακάλυψης γνώσης δε μπορεί να εγγυηθεί τη λύση ενός προβλήματος σε μικρό χρονικό διάστημα. Ο χρόνος απόκρισης του συστήματος εξαρτάται από το μέγεθος της βάσης γνώσης και την πολυπλοκότητα των αξιωμάτων της. Αυτό έχει σαν αποτέλεσμα, οι μεγάλες οντολογίες να έχουν μεγάλες απαιτήσεις σε πόρους του συστήματος, όπως και σε χρόνο εκτέλεσης του αλγορίθμου μάθησης. Πολλά συστήματα ανακάλυψης γνώσης (π.χ. Aleph) περιορίζουν αρκετά το πεδίο ενδιαφέροντος, έτσι ώστε να είναι αποδοτικά τόσο στο χρόνο εκτέλεσης, όσο και σε κατανάλωση των πόρων του συστήματος. Από την άλλη πλευρά, συστήματα όπως το DL – Learner (βλ. Παράγραφο 4.5) για να πετύχουν απόκριση του συστήματος σε πραγματικό χρόνο (real - time), ορίζουν τον μέγιστο χρόνο εκτέλεσης των αλγορίθμων μάθησης, έτσι ώστε να δίνουν ένα τμήμα της εκπαίδευσης της οντολογίας μέσα στα επιτρεπτά χρονικά όρια της εφαρμογής.

4.5 Εργαλεία Ανακάλυψης Γνώσης στο Σημασιολογικό Ιστό

Τα εργαλεία για ανακάλυψη γνώσης από DL δεδομένα δεν είναι πολλά, και αυτό γιατί μόλις πρόσφατα έχει αποκτήσει ενδιαφέρον η συγκεκριμένη περιοχή εξαιτίας της ανάπτυξης του Σημασιολογικού Ιστού. Παρακάτω παρουσιάζονται κάποια εργαλεία, που πετυχαίνουν την εκπαίδευση των εννοιών (concepts) στο Σημασιολογικό Ιστό.

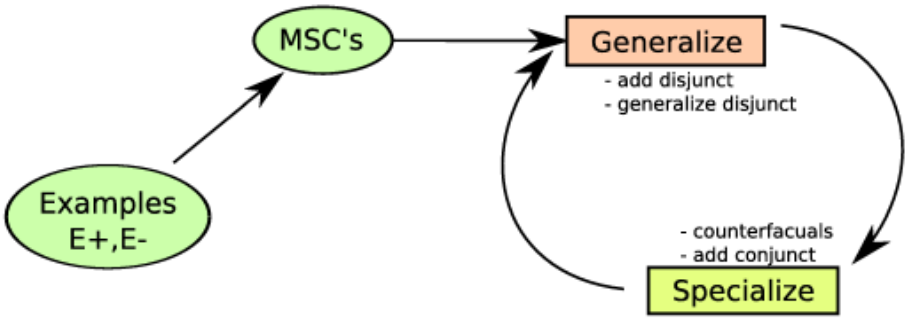
LCSLearn

Το LCSLearn [55] είναι το πρώτο εργαλείο για εκπαίδευση σε DL δεδομένα και βασίζεται στον CLASSIC DL, έναν αρχικό DL φορμαλισμό. Η προσέγγιση του είναι απλή και ο ορισμός του προβλήματος λίγο διαφορετικός. Υποθέτει την ύπαρξη των περισσότερο ειδικών εννοιών (the most specific concepts, MSC) ως στιγμιότυπα, τα οποία αποτελούν και την είσοδο του αλγορίθμου. Το MSC ενός στιγμιοτύπου είναι η πιο συγκεκριμένη έκφραση κλάσης (most specific class expression). Στη συνέχεια δημιουργείται η έννοια του αποτελέσματος (target concept), που προκύπτει από την ένωση όλων των MSC με διάζευξη (⊔). Αυτή η διαδικασία δημιουργεί υπερβολικά μεγάλους ορισμούς εννοιών. Το βασικό πλεονέκτημα της μεθόδου είναι, ότι μπορεί να δώσει λύση σε πολυωνυμικό χρόνο σε σχέση με το μήκος των MSCs. Στα μειονεκτήματα της μεθόδου ανήκουν οι μεγάλοι ορισμοί εννοιών, που δεν μπορούν να τροποποιηθούν και οδηγούν σε υπερβολική ταύτιση με τα παραδείγματα εκπαίδευσης (overfitting). Επομένως, οι ορισμοί των εννοιών, που προκύπτουν, δεν μπορούν να χρησιμοποιηθούν για κάποια πρόβλεψη, αφού λύση μπορούν να δώσουν μόνο για τα δεδομένα εκπαίδευσης. Το LCSLearn δεν είναι πλέον διαθέσιμο.

YinYang

Το YinYang [56] βασίζεται σε αλγορίθμους εκπαίδευσης για DL, πιο συγκεκριμένα για τη γλώσσα ALC, και χρησιμοποιεί τελεστές βελτίωσης (refinement operators). Η βασική ιδέα των αλγορίθμων είναι η εύρεση και η διαγραφή των τμημάτων της έννοιας, που είναι υπεύθυνα για τα λάθη ταξινόμησης. Στην πιο πρόσφατη έκδοση του [57], αντί για τη χρήση των κλασικών refinement operators, χρησιμοποιούνται τα MSCs για την επίλυση των προβλημάτων μάθησης. Όπως το LCSLearn, έτσι και το YinYang δημιουργεί υπερβολικά μεγάλους ορισμούς εννοιών. Ένας λόγος του μεγάλου μήκους των MSCs είναι, ότι MSC για γλώσσες υψηλής εκφραστικότητας δεν υπάρχουν, οπότε πρέπει να προσεγγιστούν. Επίσης, υπάρχει η περίπτωση σε μια μεγάλη βάση γνώσης δύο στιγμιότυπα να έχουν τις ίδιες ιδιότητες και να μοιράζονται το ίδιο MSC. Το πρόβλημα αντιμετωπίζεται, μόνο αν η λύση περιλαμβάνει τα MSCs όλων των θετικών

παραδειγμάτων και κανενός αρνητικού. Ο αλγόριθμος του YinYang περιλαμβάνει τη φάση της γενίκευσης (generalization) για τη γενίκευση των θετικών παραδειγμάτων και τη φάση της εξειδίκευσης (specialization) των γενικευμένων εννοιών, έτσι ώστε να αποκλείονται τα αρνητικά παραδείγματα (Εικόνα 4.2).



Εικόνα 4.2: Η διαδικασία εκπαίδευσης του YinYang

DL – FOIL

Το DL – FOIL [58] βασίζεται στο συνδυασμό βελτίωσης των εκφράσεων των κλάσεων από πάνω προς τα κάτω και αντίστροφα. Χρησιμοποιεί εναλλακτικές μετρικές για την αξιολόγηση του, δίνοντας έμφαση στη διαφορά μεταξύ συμπερασματικού (deductive) και επαγωγικού (inductive) συμπερασμού και λαμβάνοντας υπόψη τα ανοιχτού κόσμου (open world semantics) σημασιολογικά χαρακτηριστικά των DL. Επομένως, τρεις περιπτώσεις ελέγχου των στιγμιοτύπων μπορούν να προκύψουν: Ένα στιγμιότυπο να είναι ανήκει σε μια έννοια, το στιγμιότυπο να ανήκει σε μια άρνηση της έννοιας (negation of concept) ή κανένα από τα δυο να μη μπορεί να προκύψει. Αυτό οδηγεί στη χρήση διαφορετικών μετρικών για τα DL. Στους ελέγχους των στιγμιοτύπων, ένα «match» προκύπτει όταν τόσο ο deductive όσο και ο inductive ταξινομητής συμφωνούν. Ένα «omission» προκύπτει, όταν η inductive μέθοδος δεν μπορεί να αποφασίσει την ένταξη ή μη σε μια έννοια, ενώ η deductive μέθοδος μπορεί. Ενώ «Commission» προκύπτει όταν οι μέθοδοι, deductive και inductive, δεν συμφωνούν. Το DL – FOIL εργαλείο δεν είναι ελεύθερα διαθέσιμο προς το παρόν.

DL – Learner

Το DL – Learner [59] χρησιμοποιεί μια προσέγγιση τελείως διαφορετική. Ο αλγόριθμος εκπαίδευσης είναι ένας γενετικός αλγόριθμος (Genetic Algorithm) αναζήτησης και χρησιμοποιεί μια ευριστική (heuristic) αναζήτηση στο χώρο των πιθανών λύσεων συνδυασμένη με μια ασφαλή μέθοδο αποτυχίας, η οποία να εξασφαλίζει την ολοκλήρωση του αλγορίθμου και να εγγυάται ότι η λύση, αν υπάρχει, θα βρεθεί

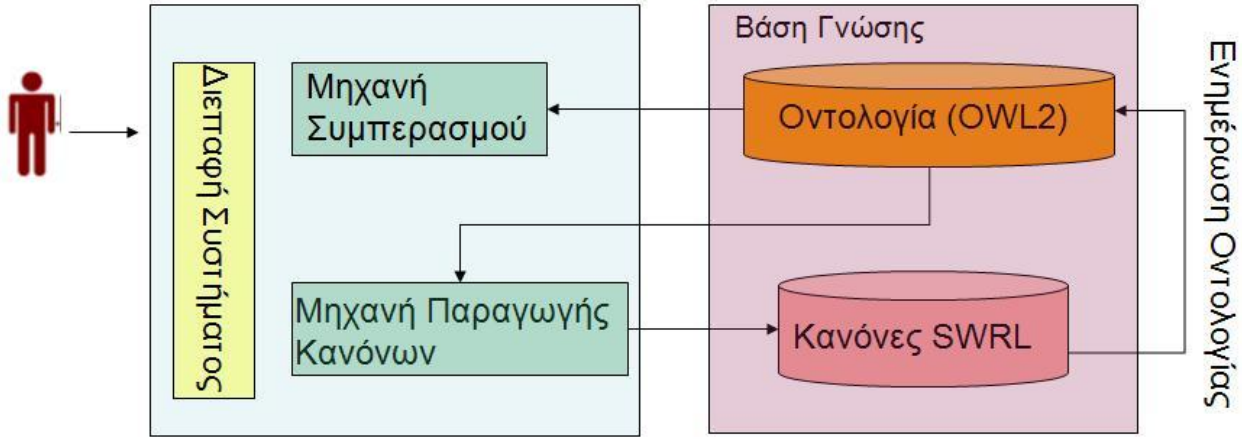
(failsafe). Είναι υλοποιημένο ως ένας αλγόριθμος από πάνω προς τα κάτω, δηλαδή ξεκινά από την πιο γενική λύση και την εξειδικεύει, αλλά και η αντίστροφη περίπτωση είναι πιθανή χρησιμοποιώντας κάποιο διαφορετικό heuristic. Οι αλγόριθμοι του DL – Learner δεν μπορούν να εγγυηθούν ότι βρίσκουν τη μικρότερη λύση του προβλήματος. Παρόλα αυτά, οι αλγόριθμοι είναι σχεδιασμένοι έτσι ώστε να παράγουν μικρές και αναγνώσιμες λύσεις. Οι αλγόριθμοι του βασίζονται στη λογική των δένδρων αναζήτησης χρησιμοποιώντας έναν operator που είναι υπεύθυνος για την επέκταση των κόμβων χρησιμοποιώντας κατάλληλα τη μερική διάταξη (subsumption) και δημιουργεί ένα δένδρο αναζήτησης με TOP, το μέγιστο του μερικώς διατεταγμένου συνόλου (supremum of the partial – ordered space) σαν τη ρίζα του. Τότε ξεκινά η αναζήτηση των κόμβων για την καλύτερη εκτίμηση του heuristic μέχρι ένα συγκεκριμένο όριο βάθους, το οποίο ονομάζεται οριζόντια επέκταση (horizontal expansion). Αν φτάσει στο μέγιστο βάθος, τότε εξετάζονται οι κόμβοι που δεν έχουν ακόμα επεκταθεί (failsafe). Η διαδικασία εξασφαλίζει, ότι αν υπάρχει η λύση θα βρεθεί. Επίσης, χρησιμοποιούνται τεχνικές για την περικοπή (pruning) των δένδρων αναζήτησης. Τέλος, οι αλγόριθμοι χρησιμοποιούν αρκετές μετρικές για την αξιολόγηση της λύσης τους.

ΚΕΦΑΛΑΙΟ 5

ΕΡΓΑΛΕΙΟ ΑΥΤΟΜΑΤΗΣ ΠΑΡΑΓΩΓΗΣ ΚΑΝΟΝΩΝ SWRL

5.1 Γενική Αρχιτεκτονική Συστήματος

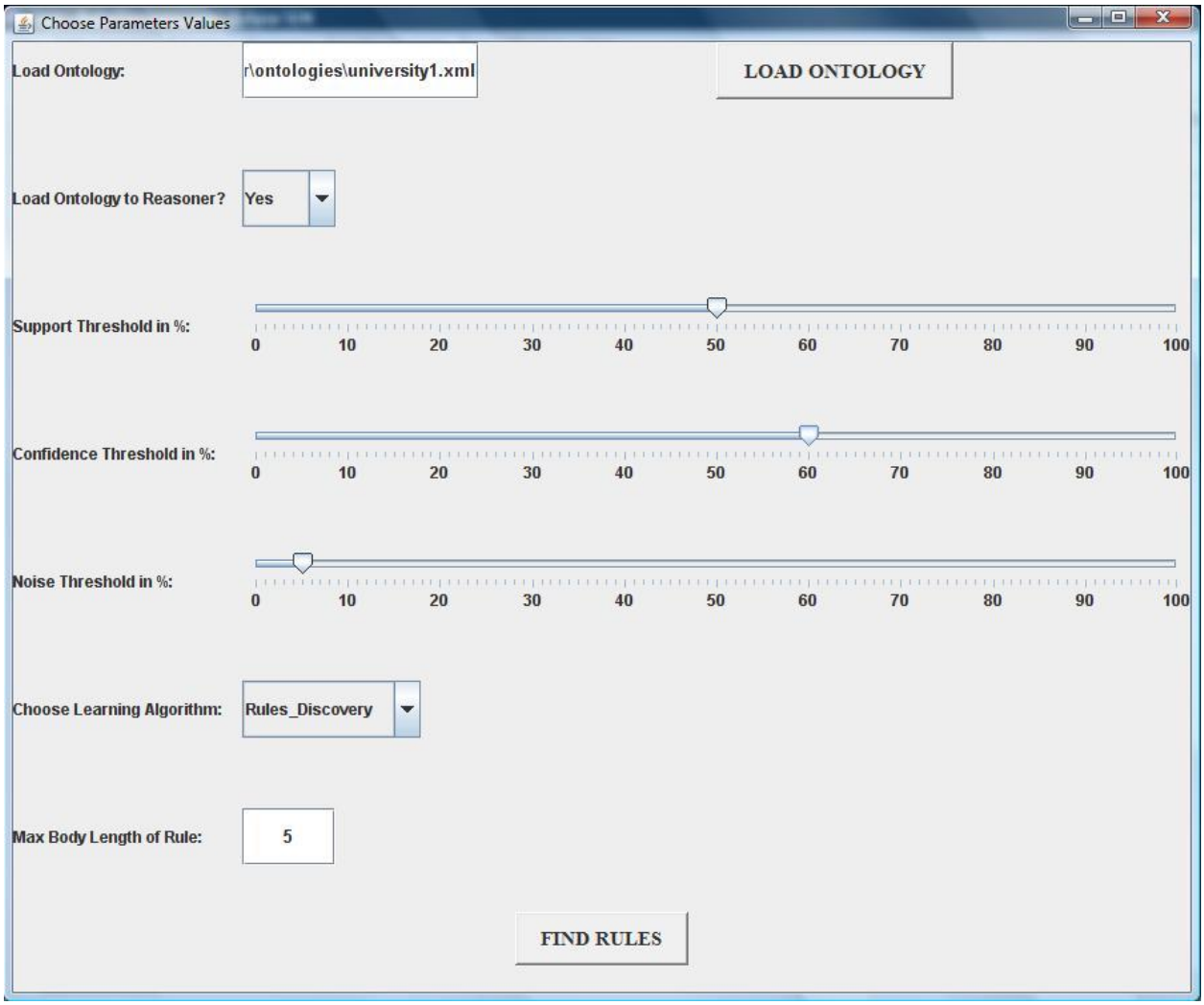
Τα συστατικά που συνθέτουν την αρχιτεκτονική του εργαλείου αυτόματης παραγωγής κανόνων SWRL απεικονίζονται στην Εικόνα 5.1. Στη συνέχεια περιγράφεται περιληπτικά η λειτουργικότητα όλων των συστατικών του και στις επόμενες ενότητες ο σχεδιασμός αυτών καθώς και οι εξαρτήσεις/συνδέσεις μεταξύ τους.



Εικόνα 5.1: Γενική Αρχιτεκτονική Συστήματος

<u>Διεπαφή Συστήματος</u>

Η διεπαφή αυτή είναι η αρχική διεπαφή ανάμεσα στον χρήστη και στο υπόλοιπο σύστημα. Ο χρήστης καθορίζει τα κριτήρια ικανοποίησης των κανόνων, αποφασίζει αν θα προηγηθεί η διαδικασία συμπερασμού πριν τη διαδικασία παραγωγής κανόνων και αποφασίζει το μέγιστο πλήθος των στοιχείων των σωμάτων των κανόνων που θα προκύψουν (Εικόνα 5.2). Οι παράμετροι που καθορίζει ο χρήστης είναι οι ελάχιστες τιμές υποστήριξης (SupportThreshold), εμπιστοσύνης (ConfidenceThreshold) και θορύβου (NoiseThreshold), που πρέπει να ικανοποιούν οι κανόνες, που παράγονται, με σκοπό να είναι μέρος της απόκρισης του συστήματος. Ο τρόπος υπολογισμού των κριτηρίων ικανοποίησης ενός κανόνα περιγράφονται στην παράγραφο 5.2.



Εικόνα 5.2: Διεπαφή Συστήματος

Μηχανή Συμπερασμού

Ανάλογα με την επιλογή του χρήστη, η οντολογία «περνάει» μέσω της διαδικασίας συμπερασμού. Η μηχανή συμπερασμού, που χρησιμοποιείται στην παρούσα εργασία, είναι ο Pellet (Κεφάλαιο 3).

Μηχανή Παραγωγής Κανόνων

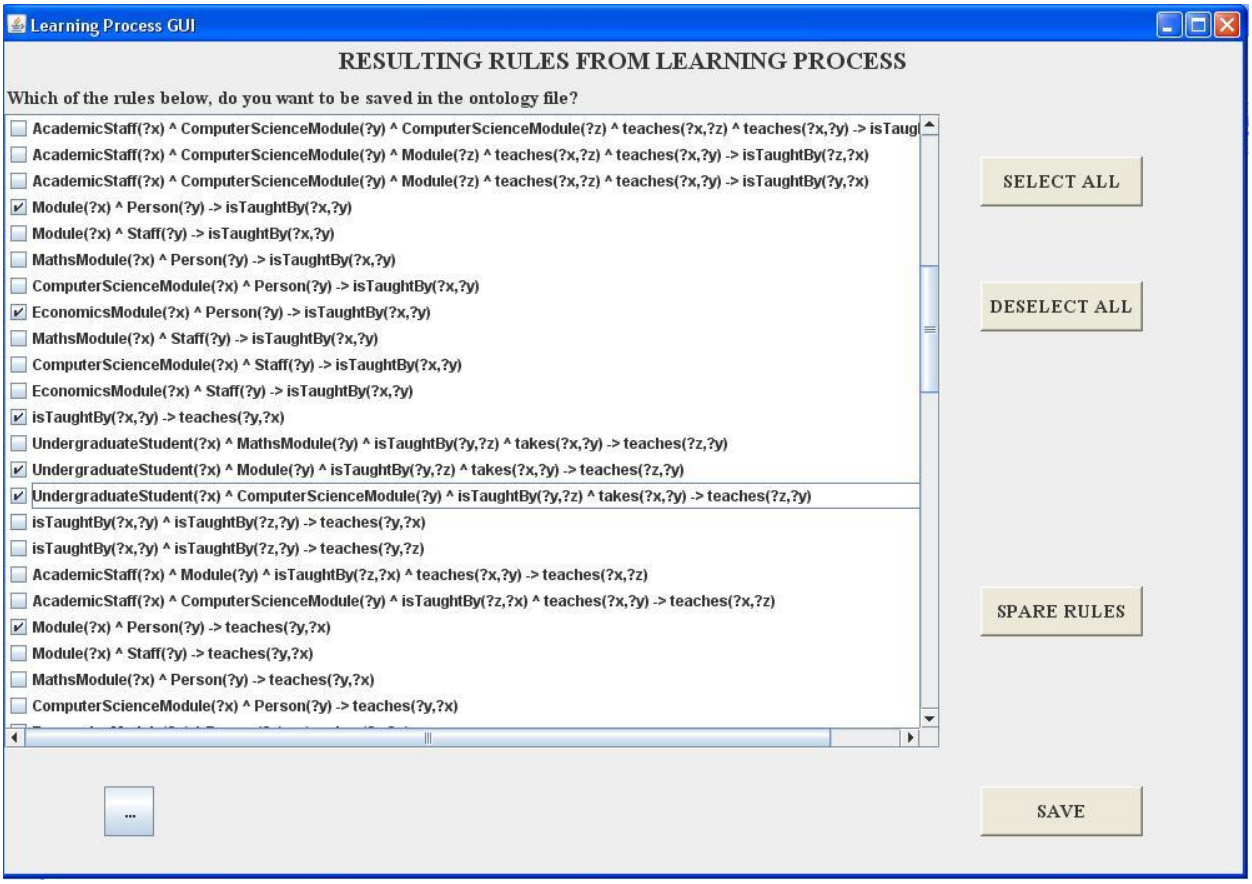
Η μηχανή παραγωγής κανόνων αποτελεί το βασικό συστατικό του συστήματος. Σ' αυτό το στάδιο εκτελείται η διαδικασία δημιουργίας των κανόνων, που ικανοποιούν τα κριτήρια, που όρισε ο χρήστης στην αρχική διεπαφή. Ο αλγόριθμος που χρησιμοποιείται για την παραγωγή των κανόνων παρουσιάζεται αναλυτικά στην παράγραφο 5.4.

Οντολογία

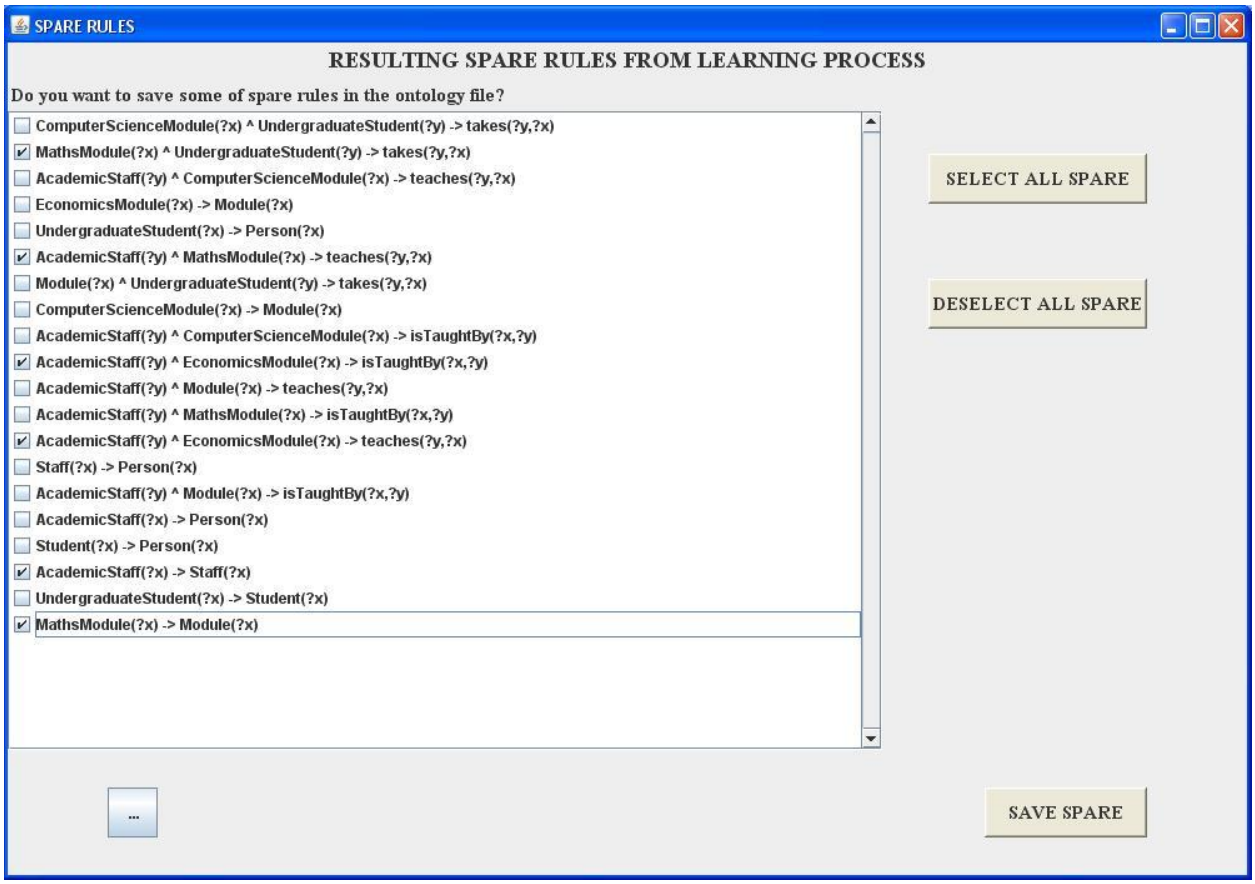
Η οντολογία παρέχει στο σύστημα το λεξιλόγιο (έννοιες και συσχετίσεις), τα δεδομένα (στιγμιότυπα κλάσεων και ιδιοτήτων) και την απαιτούμενη σημασιολογία για την αυτόματη παραγωγή κανόνων που ικανοποιούν τις παραμέτρους και τις απαιτήσεις που τέθηκαν από το χρήστη (βλ. Εικόνα 5.2).

Κανόνες SWRL

Η έξοδος του συστήματος είναι ένα σύνολο SWRL κανόνων, που ο χρήστης μέσω διεπαφών αποφασίζει, ποιοι από αυτούς, είναι οι πλέον κατάλληλοι για να αποθηκευτούν στην οντολογία. Το σύνολο SWRL κανόνων χωρίζεται σε δύο υποσύνολα. Το πρώτο υποσύνολο (Εικόνα 5.3) είναι το σύνολο των κανόνων που το ίδιο το σύστημα θεωρεί σημαντικούς για την ενημέρωση της βάσης γνώσης. Το δεύτερο υποσύνολο (Εικόνα 5.4) είναι το σύνολο των κανόνων που το σύστημα θεωρεί περιττούς, δηλαδή δεν προσφέρουν ουσιαστική πληροφορία για την οντολογία. Παρόλα αυτά, ο χρήστης έχει τη δυνατότητα να επιλέξει κανόνες και από τα δύο υποσύνολα για την εγγραφή τους στην οντολογία. Οι κανόνες που θεωρούνται περιττοί από το σύστημα, περιγράφονται στην παράγραφο 5.3.



Εικόνα 5.3: Διεπαφή Επιλογής Κανόνων



Εικόνα 5.4: Διεπαφή Επιλογής Περιττών Κανόνων

5.2 Υπολογισμός Κριτηρίων Κανόνων

5.2.1 Υπολογισμός Υποστήριξης/Εμπιστοσύνης

Δύο κριτήρια, που είναι σημαντικά για την αξιολόγηση ενός υποψήφιου κανόνα (candidate rule), είναι η υποστήριξη (support) και η εμπιστοσύνη (confidence) (Κεφάλαιο 4). Σ' αυτήν την παράγραφο η υποστήριξη και η εμπιστοσύνη προσαρμόζονται, ώστε να εφαρμοστούν για την αξιολόγηση των SWRL κανόνων.

Υποστήριξη

Η ***υποστήριξη*** (support) ενός κανόνα R ($Body \rightarrow Head$) ορίζεται ως ο αριθμός των διαφορετικών ποσοτικοποιήσεων (bindings) των μεταβλητών της κεφαλής του που ικανοποιούν τον κανόνα (σώμα και κεφαλή), διαιρεμένος με το συνολικό αριθμό των bindings των μεταβλητών της κεφαλής του κανόνα. Δηλαδή:

$$Support = \frac{|Total\_Head\_Bindings \cap Total\_Body\_Bindings|}{|Total\_Head\_Bindings|},$$

όπου $|Total\_Head\_Bindings|$ είναι το συνολικό πλήθος των διαφορετικών συνδυασμών των στιγμιοτύπων των μεταβλητών της κεφαλής (Head) και

$|Total\_Head\_Bindings \cap Total\_Body\_Bindings|$ είναι το συνολικό πλήθος των διαφορετικών συνδυασμών των στιγμιοτύπων των μεταβλητών της κεφαλής (Head), που ικανοποιούν ταυτόχρονα και το σώμα και την κεφαλή του κανόνα.

Επομένως, η υποστήριξη είναι ο λόγος των θετικών ποσοτικοποιήσεων που συμπεραίνονται από τον κανόνα προς το συνολικό πλήθος των ποσοτικοποιήσεων της κεφαλής. Η φυσική ερμηνεία της μετρικής είναι πως εκφράζει τη βαρύτητα του κανόνα και το πόσο σημαντικός είναι στο σύνολο των δεδομένων.

Εμπιστοσύνη

Η ***εμπιστοσύνη*** (confidence) ενός κανόνα R ($Body \rightarrow Head$) ορίζεται ως ο αριθμός των διαφορετικών bindings των μεταβλητών της κεφαλής του κανόνα που ικανοποιούν τον κανόνα (σώμα και κεφαλή), διαιρεμένος με τον αριθμό των διαφορετικών bindings των μεταβλητών της κεφαλής, που ικανοποιούν μόνο το σώμα του κανόνα. Δηλαδή:

$$Confidence = \frac{|Total\_Head\_Bindings \cap Total\_Body\_Bindings|}{|Total\_Body\_Bindings|},$$

όπου $|Total\_Body\_Bindings|$ είναι το συνολικό πλήθος των διαφορετικών συνδυασμών των στιγμιοτύπων των μεταβλητών της κεφαλής, Head, που ικανοποιούν μόνο το σώμα του κανόνα και $|Total\_Head\_Bindings \cap Total\_Body\_Bindings|$ είναι το συνολικό πλήθος των διαφορετικών συνδυασμών των στιγμιοτύπων των μεταβλητών της κεφαλής (Head), που ικανοποιούν ταυτόχρονα και το σώμα και την κεφαλή του κανόνα.

Επομένως, η εμπιστοσύνη είναι ο λόγος των θετικών ποσοτικοποιήσεων που συμπεραίνονται από τον κανόνα προς το συνολικό αριθμό των ποσοτικοποιήσεων που προκύπτουν από το σώμα του κανόνα. Η εμπιστοσύνη εκφράζει την πιθανότητα να ικανοποιείται η κεφαλή ενός κανόνα δεδομένου πως ικανοποιείται το σώμα του. Με άλλα λόγια δηλώνει την ισχύ της συνεπαγωγής, δηλαδή πόσο «ισχυρός» είναι ο κανόνας.

Παραδείγματα Υπολογισμού Υποστήριξης/Εμπιστοσύνης

Παρακάτω παρουσιάζονται δύο παραδείγματα, που αναλύουν τον τρόπο υπολογισμού της υποστήριξης και της εμπιστοσύνης των κανόνων.

Πίνακας 5.1: Βάση Γνώσης Παραδείγματος

TBox	Person, Female ⊑ Person
ABox	Person(mary), Person(ann), Person(tom), Person(eve), Female(mary), Female(ann), Female(eve), isDaughterOf(mary, ann), isDaughterOf(eve, tom), isParentOf(ann, mary), isParentOf(ann, tom), isParentOf(tom, eve)

Σύμφωνα με την βάση γνώσης του παραδείγματος (Πίνακας 5.1) θεωρούμε τους ακόλουθους κανόνες [60]:

$isParentOf(Y, tom), Person(X) \rightarrow isDaughterOf(X, Y)$. (s = 0.5, c = 0.25)

Σύμφωνα με τον ορισμό της υποστήριξης που δόθηκε παραπάνω, προκύπτει ως εξής:

Οι συνδυασμοί των στιγμιοτύπων, που ικανοποιούν γενικά την κεφαλή του κανόνα σύμφωνα με τη βάση γνώσης (Πίνακας 5.1) είναι οι ακόλουθοι δύο: ((mary, ann), (eve, tom)). Από αυτούς τους δύο συνδυασμούς, ο μόνος που επαληθεύει τον κανόνα στο σύνολο του (σώμα και κεφαλή) είναι ο (mary, ann). Οπότε η υποστήριξη του κανόνα είναι ½ = 0.5.

Ο υπολογισμός της εμπιστοσύνης ακολουθώντας και πάλι τον ορισμό γίνεται ως εξής:

Οι συνδυασμοί των στιγμιοτύπων, που ικανοποιούν γενικά το σώμα του κανόνα σύμφωνα με τη βάση γνώσης είναι οι ακόλουθοι: ((mary, ann), (ann, ann), (tom, ann) και (eve, ann)). Από αυτούς τους συνδυασμούς, ο μόνος που επαληθεύει και την κεφαλή του κανόνα είναι ο (mary, ann). Οπότε η εμπιστοσύνη του κανόνα είναι ¼ = 0.25.

Ομοίως, γίνεται ο υπολογισμός και για το παρακάτω παράδειγμα:

$Female(X), Person(Y) \rightarrow isDaughterOf(X, Y)$. (s = 1.0, c = 0.17)

Οι συνδυασμοί των στιγμιοτύπων, που ικανοποιούν γενικά την κεφαλή του κανόνα σύμφωνα με τη βάση γνώσης (Πίνακας 5.1) είναι οι ακόλουθοι: ((mary, ann), (eve,

tom)). Αυτοί οι συνδυασμοί ικανοποιούν και οι δύο τον κανόνα (σώμα και κεφαλή), οπότε η υποστήριξη του κανόνα είναι 2/2 = 1.

Ομοίως για την εμπιστοσύνη, οι συνδυασμοί που ικανοποιούν γενικά το σώμα είναι οι εξής: ((ann, ann), (ann, mary), (ann, tom), (ann, eve), (mary, ann), (mary, mary), (mary, tom), (mary, eve), (eve, ann), (eve, mary), (eve, tom), (eve, eve)). Από τους οποίους οι μόνοι που επαληθεύουν και την κεφαλή του κανόνα είναι οι εξής: ((mary, ann), (eve, tom)). Δηλαδή από τις 12 πιθανές ποσοτικοποιήσεις της κεφαλής μόνο οι 2 είναι οι σωστές. Επομένως, η εμπιστοσύνη του κανόνα είναι πολύ χαμηλή, δηλαδή 2/12 = 0.17.

Προϋπόθεση Σωστού Υπολογισμού Εμπιστοσύνης

Για να είναι σωστός ο υπολογισμός της εμπιστοσύνης, πρέπει ο κανόνας να είναι safe (βλ. Κεφάλαιο 2). Αν ο κανόνας δεν είναι safe, τότε η τιμή της εμπιστοσύνης, που προκύπτει, είναι μεγαλύτερη από την πραγματική τιμή της, με αποτέλεσμα ο κανόνας να φαίνεται περισσότερο «ισχυρός» από ότι πραγματικά είναι.

Για παράδειγμα, ας θεωρήσουμε τους κανόνες για τους οποίους υπολογίσαμε τις σωστές τιμές της υποστήριξης και της εμπιστοσύνης προηγουμένως, με τη μόνη διαφορά ότι τώρα οι κανόνες δεν είναι safe. Επομένως, έχουμε τους κανόνες:

$isParentOf(Y, tom) \rightarrow isDaughterOf(X, Y)$ (s = 0.5, c = 1.0) (1)

$Female(X) \rightarrow isDaughterOf(X, Y)$ (s = 1.0, c = 0.67) (2)

Σύμφωνα με τη βάση γνώσης (Πίνακας 5.1) και με τον τρόπο υπολογισμού της υποστήριξης, όπως περιγράφηκε προηγουμένως, οι δύο κανόνες έχουν και πάλι την ίδια τιμή υποστήριξης, δηλαδή s = 0.5 και s = 1.0, αντίστοιχα.

Δε συμβαίνει όμως το ίδιο και με την τιμή της εμπιστοσύνης. Για τον πρώτο κανόνα, ο συνδυασμός που ικανοποιεί γενικά το σώμα είναι ο εξής: (ann, tom), ο οποίος επαληθεύει και την κεφαλή του κανόνα, αφού υπάρχει ο συνδυασμός (mary, ann) στη βάση. Επομένως η τιμή της εμπιστοσύνης του κανόνα είναι 1/1 = 1, δηλαδή ο κανόνας φαίνεται ότι είναι πολύ ισχυρός κάτι που όπως αποδείχτηκε και προηγουμένως δεν ισχύει.

Το ίδιο ισχύει και για το δεύτερο κανόνα. Τα στιγμιότυπα που ικανοποιούν το σώμα του κανόνα είναι οι: ((ann), (mary), (eve)), οπότε οι συνδυασμοί που επαληθεύουν την κεφαλή του κανόνα είναι οι ακόλουθοι: ((mary, ann), (eve, tom)). Επομένως, η τιμή της εμπιστοσύνης είναι 2/3 = 0.67, μια τιμή που απέχει πολύ από την πραγματική τιμή της εμπιστοσύνης του κανόνα.

Επομένως, αν ο κανόνας δεν είναι safe, πρέπει πρώτα να γίνει safe για το σωστό υπολογισμό της εμπιστοσύνης του [60].

Παρατήρηση: Στην παρούσα εργασία, όλοι οι κανόνες που προκύπτουν είναι safe οπότε δεν είναι απαραίτητη κάποια επιπλέον μετατροπή τους.

Αποδεκτοί Κανόνες

Για να θεωρηθεί ένας κανόνας μέρος του συνόλου των κανόνων, που προκύπτουν από την οντολογία, θα πρέπει να ικανοποιεί τα κριτήρια των SupportThreshold και ConfidenceThreshold, που ορίστηκαν μέσω της διεπαφής του συστήματος. Δηλαδή, η τιμή της υποστήριξης του κανόνα πρέπει να είναι μεγαλύτερη ή ίση του SupportThreshold, όπως επίσης και η τιμή της εμπιστοσύνης πρέπει να είναι μεγαλύτερη ή ίση του ConfidenceThreshold.

5.2.2 Θόρυβος

Μια άλλη παράμετρος, που χρησιμοποιείται στον αλγόριθμο παραγωγής κανόνων, είναι ο θόρυβος (noise). Αυτή η παράμετρος καθορίζει το μέγιστο επιτρεπτό πλήθος στιγμιοτύπων της κεφαλής, που μπορούν να μην καλύπτονται από τον κανόνα, δηλαδή $\lceil (1-noise)*Total\_Individuals\_Of\_HeadPart \rceil$, όπου Total_Individuals_Of_HeadPart είναι το πλήθος των στιγμιοτύπων που ανήκουν στην κεφαλή (HeadPart), αν η κεφαλή είναι μια κλάση της οντολογίας ή το συνολικό πλήθος των συνδυασμών των στιγμιοτύπων της κεφαλής, αν η κεφαλή του κανόνα είναι μια ιδιότητα (property) της οντολογίας.

Τα στιγμιότυπα, που δεν καλύπτονται από τον κανόνα, είναι: $\left|Total\_Individuals\_Of\_HeadPart - Covered\_Individuals\_Of\_HeadPart\right|$, όπου Covered_Individuals_Of_HeadPart είναι το πλήθος των στιγμιοτύπων ή των συνδυασμών των στιγμιοτύπων της κεφαλής (HeadPart), που καλύπτονται επίσης και από το σώμα του κανόνα.

Επομένως, ένας κανόνας για μη θεωρηθεί «αδύναμος» (weak), θα πρέπει να ισχύει η συνθήκη:

$$\left|Total\_Individuals\_Of\_HeadPart - Covered\_Individuals\_Of\_HeadPart\right| \leq \lceil (1-noise)*Total\_Individuals\_Of\_HeadPart \rceil.$$

Αν ένας κανόνας ικανοποιεί τα κριτήρια της υποστήριξης και της εμπιστοσύνης αλλά όχι του θορύβου, τότε αυτός ο κανόνας δεν αποτελεί μέρος του συνόλου των κανόνων της οντολογίας.

Η τιμή του θορύβου εξαρτάται από την ορθότητα της βάσης γνώσης. Αν η βάση γνώσης είναι σωστή και τα λάθη είναι περιορισμένα, τότε η παράμετρος του θορύβου μπορεί να οριστεί σε μια τιμή πολύ κοντά στο 0. Αν όμως η βάση γνώσης περιλαμβάνει λάθη, π.χ. δημιουργήθηκε αυτόματα από κείμενο (*ontology learning*), τότε η τιμή του θορύβου πρέπει να έχει υψηλότερη τιμή.

Η προκαθορισμένη τιμή του θορύβου είναι 0%, δηλαδή θεωρείται πως η βάση γνώσης είναι σωστή. Όμως, η πολύ χαμηλή τιμή του θορύβου μπορεί να οδηγήσει στην παραγωγή μη απαραίτητων κανόνων.

5.3 Περιττοί SWRL Κανόνες

Όπως αναφέρθηκε και στην παράγραφο 5.1, το σύστημα διακρίνει ένα σύνολο κανόνων ως περιττούς. Ως περιττοί θεωρούνται οι κανόνες που δεν προσφέρουν επιπλέον πληροφορία για την οντολογία. Η πρώτη κατηγορία περιττών κανόνων είναι οι τετριμμένες περιπτώσεις, δηλαδή το σώμα του κανόνα να περιέχει μόνο ένα στοιχείο το οποίο να είναι ακριβώς το ίδιο με το στοιχείο της κεφαλής του κανόνα ($A \rightarrow A$). Οι τετριμμένες περιπτώσεις αποκλείονται τελείως και από το σύνολο των περιττών κανόνων.

Ακόμα, αν η οντολογία έχει περάσει μέσω διαδικασίας συμπερασμού πριν τη διαδικασία παραγωγής κανόνων, τότε εξαιτίας της ιεραρχίας των κλάσεων και των ιδιοτήτων που έχει προκύψει, τα στιγμιότυπα των υποκλάσεων έχουν γίνει πλέον και στιγμιότυπα των υπερκλάσεων (inferred), όπως το ίδιο συμβαίνει και με τους συνδυασμούς των στιγμιοτύπων των δυαδικών ιδιοτήτων, που έχουν μεταφερθεί παραπάνω στην ιεραρχία των ιδιοτήτων, αν υπάρχει. Επίσης, και πάλι λόγω συμπερασμού, αν μια ιδιότητα Α είναι inverse μιας ιδιότητας Β, τότε αυτές οι δύο ιδιότητες θα έχουν ακριβώς τους ίδιους συνδυασμούς με αντεστραμμένα τα υποκείμενα και τα αντικείμενα τους. Σύμφωνα με τα παραπάνω, το σύστημα διαχωρίζει κάποιους κανόνες ως περιττούς είτε γιατί είναι προφανείς λόγω της ιεραρχίας, είτε γιατί δίνουν ακριβώς την ίδια πληροφορία με κάποιον άλλον κανόνα, που έχει ήδη προκύψει. Πιο συγκεκριμένα, οι περιπτώσεις των περιττών κανόνων, που διακρίνονται από το σύστημα είναι οι παρακάτω:

- **SubClass** $\rightarrow$ **SuperClass**: Αν η κεφαλή του κανόνα είναι μια κλάση και στο σώμα του κανόνα υπάρχει μια υποκλάση αυτής, τότε ο κανόνας θεωρείται περιττός. Αυτό συμβαίνει, γιατί ο κανόνας είναι προφανής λόγω της ιεραρχίας των κλάσεων και ταυτόχρονα σίγουρα πολύ «ισχυρός», αφού η εμπιστοσύνη του θα είναι πάντα 100%.
- **SubProperty** $\rightarrow$ **SuperProperty**: Είναι ακριβώς η ίδια περίπτωση με την προηγούμενη.
- **BodyParts** $\rightarrow$ **Property** και **BodyParts** $\rightarrow$ **InverseProperty**: Σ' αυτήν την περίπτωση ο ένας από τους δυο κανόνες θα εμφανιστεί στο σύνολο των κανόνων και ο άλλος κανόνας στο σύνολο των περιττών κανόνων. Αυτό συμβαίνει, γιατί αυτοί οι δύο κανόνες είναι ίδιοι, με τη μόνη διαφορά την αντιστροφή των υποκειμένων και των αντικειμένων στις κεφαλές των κανόνων. Επίσης, λόγω του συμπερασμού αυτοί οι δύο κανόνες θα έχουν και αντεστραμμένες τις τιμές της υποστήριξης και της εμπιστοσύνης. Επομένως, αυτοί οι δύο κανόνες προσφέρουν ακριβώς την ίδια πληροφορία, οπότε η μηχανή παραγωγής κανόνων διακρίνει ότι δεν είναι απαραίτητοι και οι δύο.
- **Αν μια SubClass έχει ακριβώς τα ίδια στιγμιότυπα με κάποια SuperClass**:

  Έστω για παράδειγμα έχουμε τους εξής κανόνες:

  $Staff(x) \cap Module(y) \rightarrow teaches(x, y)$ (1)

  $AcademicStaff(x) \cap Module(y) \rightarrow teaches(x, y)$ (2)

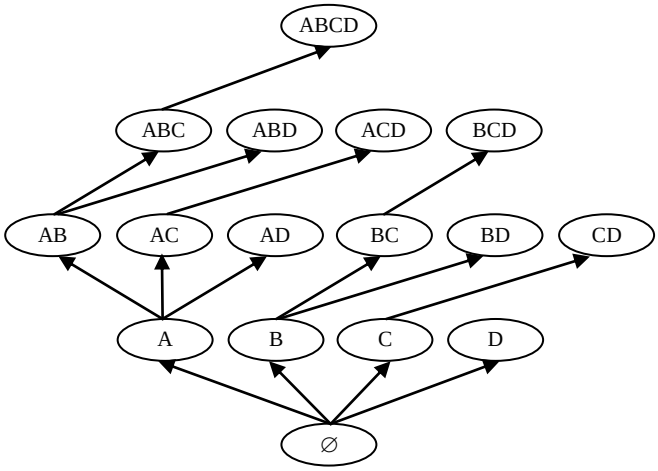
  Η μόνη διαφορά των κανόνων είναι οι κλάσεις «Staff» και «AcademicStaff». Αν όμως μέσω της διαδικασίας του συμπερασμού η κλάση «Staff» θεωρείται υπερκλάση (superclass) της «AcademicStaff» και αν έχουν ακριβώς τα ίδια στιγμιότυπα, τότε οι δύο κανόνες θα έχουν ακριβώς την ίδια τιμή σε υποστήριξη, εμπιστοσύνη και θόρυβο και ταυτόχρονα προσφέρουν παρόμοια πληροφορία, με τη μόνη διαφορά τους ότι η πληροφορία του (1) είναι πιο γενική. Επειδή στα πλαίσια της Μηχανικής Μάθησης, οι κανόνες πρέπει να είναι όσο το δυνατόν γενικότεροι για να καλύπτουν περισσότερες περιπτώσεις, ο κανόνας, που θα εμφανιστεί στο σύνολο των κανόνων θα είναι ο (1), καθώς περιλαμβάνει την υπερκλάση στο σώμα του, ενώ ο (2) θα θεωρηθεί περιττός και θα εμφανιστεί στο σύνολο των περιττών κανόνων.

Οι περιπτώσεις κανόνων, που θεωρούνται περιττοί, όπως αναφέρθηκαν παραπάνω εμφανίζονται στο σύνολο των περιττών κανόνων (Spare Rules). Παρόλο, που είναι περιπτώσεις κανόνων που δεν προσφέρουν επιπλέον πληροφορία για τη βάση γνώσης, υπάρχει και γι' αυτούς η δυνατότητα εγγραφής τους στην οντολογία.

5.4 Αλγόριθμος Παραγωγής Κανόνων

Όπως περιγράφηκε και στην παράγραφο 5.1 το βασικό τμήμα του συστήματος είναι η διαδικασία παραγωγής κανόνων για την οντολογία.

Ο αλγόριθμος παραγωγής κανόνων πετυχαίνει την παραγωγή κανόνων συσχέτισης σε Περιγραφικές Λογικές. Ο αλγόριθμος εντοπίζει όλους τους δυνατούς κανόνες συσχέτισης, που ικανοποιούν τα κριτήρια της υποστήριξης, της εμπιστοσύνης και του θορύβου. Επίσης, ο αλγόριθμος συνδυάζει την εφαρμογή κατά πλάτος και κατά βάθος αναζήτηση με σκοπό να πετύχει καλύτερη διαχείριση των πόρων του συστήματος. Παρακάτω παρουσιάζεται ένα πλάνο εκτέλεσης του αλγορίθμου, δηλαδή πώς ο αλγόριθμος εντοπίζει τους κανόνες μιας οντολογίας (Εικόνα 5.5).



Εικόνα 5.5: Πλάνο εκτέλεσης αλγορίθμου Rules Discovery

Όπου A, B, C, D μπορεί να είναι κάποια κλάση ή ιδιότητα της οντολογίας. Για να συμμετέχει κάποια κλάση ή ιδιότητα στη δημιουργία κανόνων θα πρέπει η κλάση να έχει ένα σύνολο στιγμιοτύπων (individuals) και η ιδιότητα να συσχετίζει κάποια στιγμιότυπα. Κάθε ένα από τα A, B, C, D μπορούν να αποτελούν και την κεφαλή του κανόνα. Όπως αναφέρθηκε και στην παράγραφο 5.3, κάθε φορά που ένα από αυτά τα στοιχεία

βρίσκεται στην κεφαλή του κανόνα, αποκλείεται ταυτόχρονα από το σώμα του κανόνα, αν το σώμα αποτελείται μόνο από ένα στοιχείο. Για παράδειγμα, αν η κεφαλή του κανόνα είναι το στοιχείο Α τότε οι κανόνες που μπορούν να προκύψουν είναι οι εξής:

$B \to A$, $C \to A$, $D \to A$, $AB \to A$, $AC \to A$, $AD \to A$, $BC \to A$, $BD \to A$, $CD \to A$, $ABC \to A$, $ABD \to A$, $ACD \to A$, $BCD \to A$ και $ABCD \to A$. Για να προκύψουν αυτοί οι κανόνες πρέπει να ικανοποιούνται τα κριτήρια, που περιγράφηκαν στην παράγραφο 5.2.

Στο παράδειγμα εύρεσης κανόνων για την κεφαλή Α η σειρά εξέτασης των κανόνων είναι η εξής:

Στο πρώτο επίπεδο γίνεται μια κατά πλάτος αναζήτηση των κανόνων, δηλαδή αρχικά προκύπτουν οι κανόνες: $B \to A$, $C \to A$ και $D \to A$. Αυτό συμβαίνει, έτσι ώστε αν κάποιος συνδυασμός που δεν ικανοποιεί το κριτήριο της υποστήριξης, αυτό σημαίνει ότι και οι συνδυασμοί των υψηλότερων επιπέδων, που τον περιλαμβάνουν, δεν θα ικανοποιούν τα κριτήρια, οπότε δεν είναι απαραίτητη η εξέταση τους.

Στα υψηλότερα επίπεδα η αναζήτηση των συνδυασμών των σωμάτων του κανόνα γίνεται κατά βάθος. Η κατά πλάτος αναζήτηση, που είναι καλύτερη στο πρώτο επίπεδο, δεν είναι όμως κατάλληλη για τα υψηλότερα επίπεδα, αφού ως διαδικασία έχει μεγαλύτερες απαιτήσεις σε μνήμη και μη γνωρίζοντας το πλάτος και το βάθος αναζήτησης των συνδυασμών, η διαδικασία εύρεσης των κανόνων μπορεί να γίνει απαγορευτική ως προς τους πόρους του συστήματος. Οπότε, σύμφωνα με τα παραπάνω, μετά εξετάζονται οι συνδυασμοί: $AB \to A$, $ABC \to A$, $ABCD \to A$, $ABD \to A$, $AC \to A$, $ACD \to A$ και $AD \to A$. Στη συνέχεια οι: $BC \to A$, $BCD \to A$, $BD \to A$ και τέλος ο $CD \to A$. Αυτό είναι το σύνολο των κανόνων για τους κανόνες με κεφαλή το στοιχείο Α και θα εμφανίζονταν όλοι με τη συγκεκριμένη σειρά, που αναφέρθηκαν, αν δεν υπήρχαν τα κριτήρια ικανοποίησης, αν η διαδικασία δε διέκρινε κάποιον κανόνα, να ανήκει στις περιπτώσεις της παραγράφου 5.3 και αν η παράμετρος του μέγιστου πλήθους δεν απέκλειε τη δημιουργία κάποιων από αυτών. Η ίδια διαδικασία επαναλαμβάνεται λαμβάνοντας ως κεφαλή και τα υπόλοιπα στοιχεία, δηλαδή Β, C και D.

Στη συνέχεια φαίνονται συνοπτικά τα βήματα του αλγορίθμου (Εικόνα 5.6):

Διαδικασία Rules_Discovery (Οντολογία, SupportThreshold, ConfidenceThreshold, MaxNoise, MaxBodyLength)

Είσοδος: Οντολογία

Έξοδος: Rules: Σύνολο κανόνων συσχέτισης οντολογίας,

SpareRules: Σύνολο περιττών κανόνων οντολογίας

Σταθερές: SupportThreshold, ConfidenceThreshold, MaxNoise //Κριτήρια ικανοποίησης κανόνων

MaxBodyLength // Μέγιστο πλήθος στοιχείων στο σώμα του κανόνα

1. Υπολογισμός Head Vector // Περιέχει κλάσεις, ιδιότητες (properties) και πληροφορίες αυτών, που θα εξεταστούν ως κεφαλές των κανόνων.
2. Υπολογισμός Body Vector // Περιέχει κλάσεις, ιδιότητες (properties) και πληροφορίες αυτών, που θα εξεταστούν ως τμήμα των σωμάτων των κανόνων.

Για κάθε class ή property headPart $\in$ Head Vector

Αρχικοποίηση NextLevelCombinations Vector // Περιλαμβάνει τα bodyPart που είτε δε θα εξεταστούν στο πρώτο επίπεδο, αλλά στα επόμενα, γιατί ο κανόνας, που προκύπτει είναι unsafe ή τετριμμένη περίπτωση, είτε τα bodyPart, που ικανοποιούν το κριτήριο της υποστήριξης, με σκοπό να συσχετιστούν με άλλα bodyParts, ώστε να προκύψει πιο συγκεκριμένος κανόνας.

Για κάθε class ή property bodyPart $\in$ Body Vector

Αν (BodyPart, HeadPart) είναι Unsafe Περίπτωση

NextLevelCombinations = NextLevelCombinations $\cup$ (bodyPart, HeadPart)

Αλλιώς

Εύρεση όλων των περιπτώσεων του κανόνα

Για κάθε περίπτωση κανόνα $Rule_i$

Υπολογισμός Support, Confidence, Noise

Αν ((Noise $\leq$ MaxNoise) && (Support $\geq$ SupportThreshold))

NextLevelCombinations = NextLevelCombinations $\cup$ (bodyPart, HeadPart)

Τέλος_Αν

Αν ((Noise $\leq$ MaxNoise) && (Support $\geq$ SupportThreshold) && (Confidence $\geq$ Confidence Threshold))

Rules = Rules $\cup$ $Rule_i$

SpareRules = SpareRules $\cup$ FindSpareRules($Rule_i$)

Τέλος_Αν

Τέλος_Για

Τέλος_Αν

Τέλος_Για

Για κάθε bodyPart $\in$ Heads

Specialization (NextLevelCombinations.bodyPart, HeadPart, 2)

Τέλος_Για

Τέλος_Για

Διαδικασία Specialization(BodyInformation, HeadInformation, Level)

Είσοδος: HeadInformation, BodyInformation, Level

Αν Level $\leq$ MaxBodyLength

UNION: Εύρεση όλων των πιθανών συνδυασμών του BodyInformation στο Level επίπεδο

Για κάθε συνδυασμό (NewBodyCombination, HeadInformation)

Εύρεση όλων των περιπτώσεων του κανόνα

Frequent $\leftarrow$ False

Για κάθε περίπτωση κανόνα $Rule_i$

Αν ικανοποιούνται τα κριτήρια (SupportThreshold, ConfidenceThreshold, NoiseThreshold)

Rules = Rules $\cup$ $Rule_i$

SpareRules = SpareRules $\cup$ FindSpareRules($Rule_i$)

Τέλος_Αν

Αν ((Noise $\leq$ MaxNoise) && (Support $\geq$ SupportThreshold))

Frequent $\leftarrow$ True

Τέλος_Αν

Τέλος_Για

Αν (Frequent == True)

Specialization(NewBodyCombination, HeadInformation, Level+1**)**

Τέλος_Αν

Τέλος_Για

Τέλος_Αν

Εικόνα 5.6: Αλγόριθμος Παραγωγής Κανόνων

Παρακάτω εμφανίζεται μια τυπική εκτέλεση του αλγορίθμου για την οντολογία «University.xml» [69]. Τα αποτελέσματα, που παρουσιάζονται, προκύπτουν ορίζοντας ως SupportThreshold = 30%, ConfidenceThreshold = 40%, NoiseThreshold = 5% και Μέγιστο πλήθος στοιχείων σώματος κανόνα = 5. Επίσης, έχει επιλεγεί η οντολογία να περάσει και μέσω διαδικασίας συμπερασμού. Ένα τμήμα των αποτελεσμάτων της διαδικασίας παρουσιάζονται στους Πίνακες 5.2 και 5.3.

Πίνακας 5.2: Τμήμα κανόνων οντολογίας

	Support (%)	Confidence (%)	Κανόνας
1.	**60**	**100**	**AcademicStaff(?x) ^ Module(?y) ^ Module(?z) ^ teaches(?x,?z) ^ teaches(?x,?y) -> ComputerScienceModule(?z)**
2.	**60**	**100**	**AcademicStaff(?x) ^ Module(?y) ^ Module(?z) ^ teaches(?x,?z) ^ teaches(?x,?y) -> ComputerScienceModule(?y)**
3.	**60**	**100**	**AcademicStaff(?x) ^ Module(?y) ^ AcademicStaff(?z) ^ teaches(?z,?y) ^ teaches(?x,?y) -> ComputerScienceModule(?y)**
4.	**100**	**100**	**Student(?x) -> UndergraduateStudent(?x)**
5.	**100**	**100**	**Staff(?x) -> AcademicStaff(?x)**
6.	**100**	**57**	**AcademicStaff(?x) ^ Module(?y) ^ teaches(?x,?y) -> ComputerScienceModule(?y)**
7.	**100**	**50**	**Person(?x) -> Staff(?x)**
8.	**100**	**40**	**UndergraduateStudent(?x) ^ Module(?y) ^ takes(?x,?y) -> MathsModule(?y)**
9.	**40**	**60**	**UndergraduateStudent(?x) ^ Module(?y) ^ takes(?x,?y) -> ComputerScienceModule(?y)**
10.	**40**	**75**	**AcademicStaff(?x) ^ Module(?y) ^ UndergraduateStudent(?z) ^ takes(?z,?y) ^ teaches(?x,?y) -> ComputerScienceModule(?y)**
11.	**100**	**45**	**Module(?x) -> ComputerScienceModule(?x)**
12.	**40**	**86**	**UndergraduateStudent(?x) ^ Module(?y) ^ Module(?z) ^ takes(?x,?z) ^ takes(?x,?y) -> ComputerScienceModule(?y)**

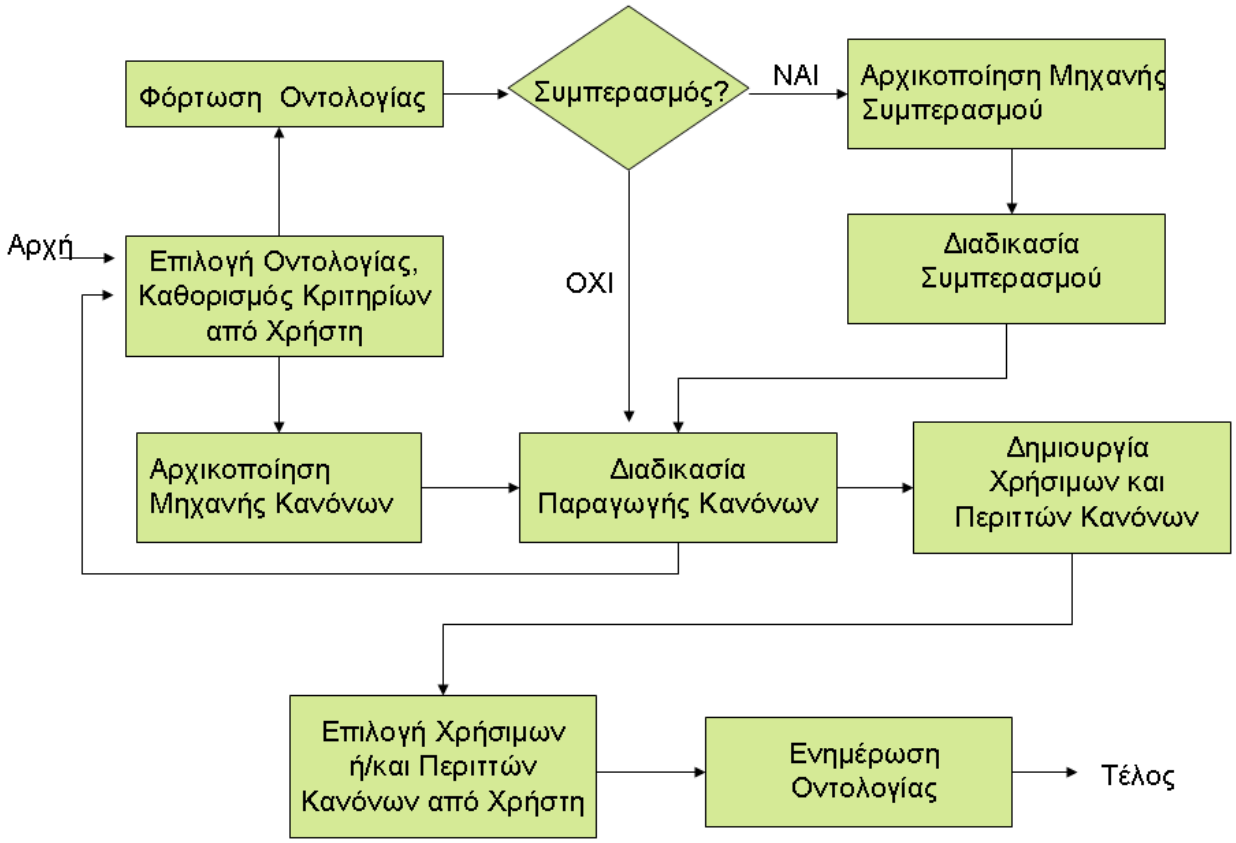
Πίνακας 5.3: Τμήμα περιττών κανόνων οντολογίας

	Support(%)	Confidence(%)	Κανόνας
1.	**60**	**100**	**AcademicStaff(?x) ^ Module(?y) ^ Module(?z) ^ isTaughtBy(?y,?x) ^ isTaughtBy(?z,?x) -> ComputerScienceModule(?y)**
2.	**60**	**100**	**AcademicStaff(?x) ^ Module(?y) ^ Module(?z) ^ isTaughtBy(?y,?x) ^ isTaughtBy(?z,?x) -> ComputerScienceModule(?z)**
3.	**60**	**100**	**AcademicStaff(?x) ^ AcademicStaff(?z) ^ Module(?y) ^ isTaughtBy(?y,?x) ^ isTaughtBy(?y,?z) -> ComputerScienceModule(?y)**
4.	**100**	**100**	**UndergraduateStudent(?x) -> Student(?x)**
5.	**100**	**100**	**AcademicStaff(?x) -> Staff(?x)**
6.	**50**	**100**	**AcademicStaff(?x) -> Person(?x)**
7.	**45**	**100**	**ComputerScienceModule(?x) -> Module(?x)**
8.	**50**	**100**	**UndergraduateStudent(?x) -> Person(?x)**
9.	**50**	**100**	**Staff(?x) -> Person(?x)**
10.	**40**	**75**	**AcademicStaff(?x) ^ Module(?y) ^ UndergraduateStudent(?z) ^ isTaughtBy(?y,?x) ^ takes(?z,?y) -> ComputerScienceModule(?y)**

5.5 Συνολική Λειτουργικότητα Συστήματος

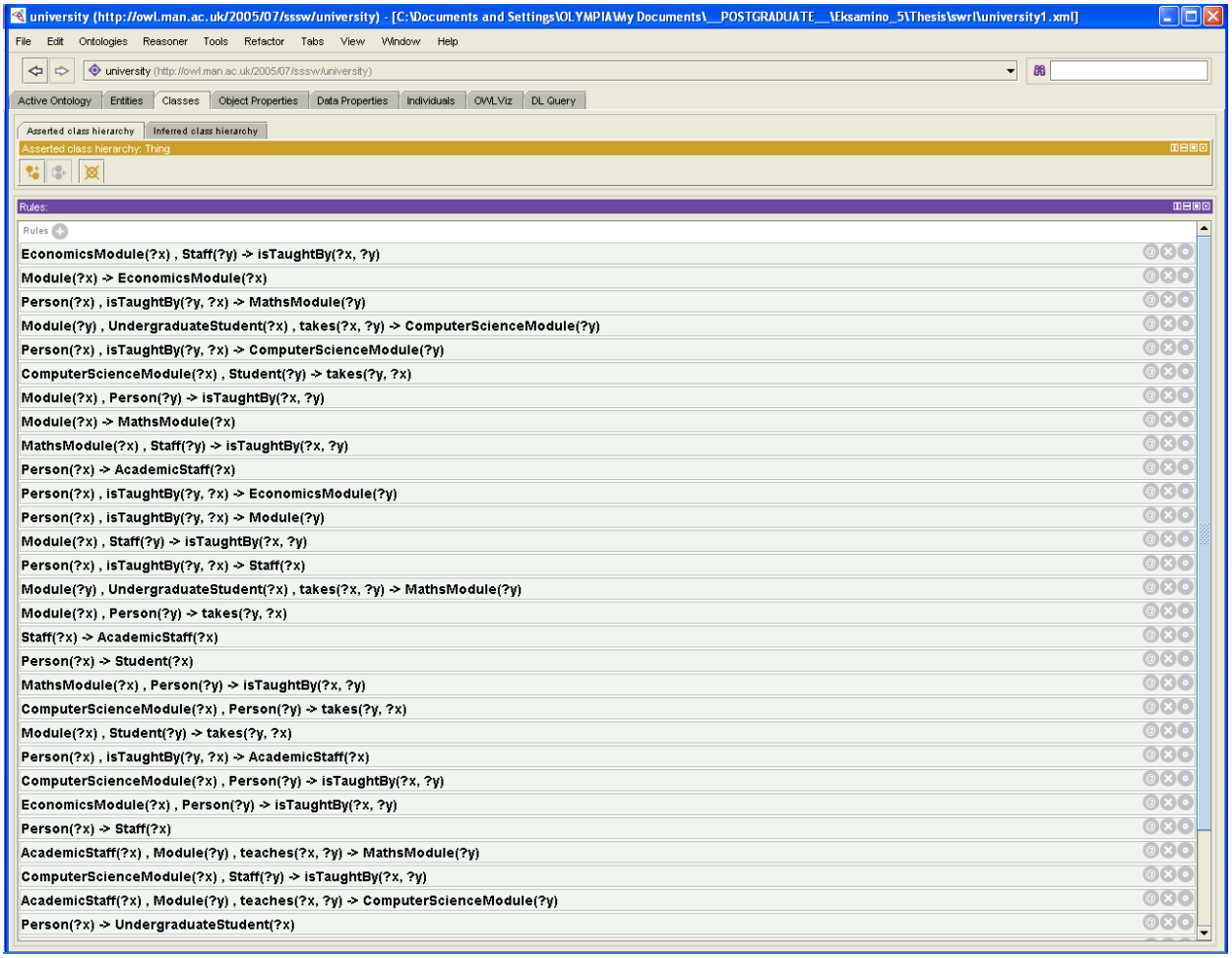
Στην εικόνα 5.7 παρουσιάζεται η ροή πληροφορίας στο σύστημα. Αρχικά, ο χρήστης επιλέγει την οντολογία που θα χρησιμοποιηθεί για την εκπαίδευση των κανόνων, καθορίζει τα κριτήρια που πρέπει να ικανοποιούν οι κανόνες, και αποφασίζει, αν η οντολογία θα περάσει μέσω της διαδικασίας συμπερασμού. Στη συνέχεια φορτώνεται η οντολογία στη μνήμη και γίνεται η αρχικοποίηση της μηχανής παραγωγής κανόνων του συστήματος. Αν έχει επιλεγεί να περάσει η οντολογία μέσω της μηχανής συμπερασμού, ακολουθεί η αρχικοποίησή της (Pellet), και πραγματοποιείται η διαδικασία του συμπερασμού. Στη συνέχεια καλείται η μηχανή παραγωγής κανόνων και δημιουργεί το σύνολο των κανόνων και το σύνολο των περιττών κανόνων της οντολογίας. Μόλις

προκύψουν οι κανόνες, ο χρήστης έχει τη δυνατότητα να επιλέξει ποιους από τους κανόνες (χρήσιμοι ή/και περιττοί) θέλει να αποθηκευτούν στην οντολογία σε μορφή SWRL. Τέλος, ο χρήστης μπορεί να επαναλάβει την ίδια διαδικασία με διαφορετική οντολογία, διαφορετική επιλογή κριτηρίων κανόνων και δυνατότητας χρήσης της μηχανής συμπερασμού.



Εικόνα 5.7: Βασική Λειτουργικότητα Συστήματος

Στην εικόνα 5.8 παρουσιάζεται ένα τμήμα SWRL κανόνων, που έχουν αποθηκευτεί στην οντολογία. Στην οντολογία αποθηκεύονται μόνο οι κανόνες που δε δημιουργούν ασάφειες και δεν επηρεάζουν τη συνέπεια της οντολογίας.



Εικόνα 5.8: Ενημέρωση Οντολογίας

5.6 Τεχνολογίες Υλοποίησης

Για την ανάπτυξη του συγκεκριμένου συστήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java.

Παράλληλα χρησιμοποιήθηκε η βιβλιοθήκη OWL – API για το χειρισμό των οντολογικών μοντέλων. Επίσης, η βιβλιοθήκη επιτρέπει τη σύνδεση με μηχανές συμπερασμού για την εκτέλεση των σχετικών διαδικασιών. Στην παρούσα εργασία, μέσω αυτής της βιβλιοθήκης, έγινε η σύνδεση με τη μηχανή συμπερασμού Pellet και εκτελέστηκαν οι διαδικασίες της κατηγοριοποίησης της ιεραρχίας εννοιών των μοντέλων και της ανεύρεσης των κλάσεων στις οποίες ανήκει το κάθε στιγμιότυπο. Η βιβλιοθήκη OWL – API δεν είναι απόλυτα συμβατή με τη μηχανή συμπερασμού Pellet, με αποτέλεσμα σε

πολλές περιπτώσεις, όταν η οντολογία περιέχει data properties να αποτυγχάνει να ολοκληρώσει τη διαδικασία κατηγοριοποίησης της ιεραρχίας των εννοιών.

Επιπλέον, μέσω της βιβλιοθήκης OWL – API πραγματοποιήθηκε και η δημιουργία, ο έλεγχος συνέπειας των SWRL κανόνων και η αποθήκευση αυτών στις οντολογίες.

Η έκδοση της βιβλιοθήκης OWL – API που χρησιμοποιήθηκε στην εργασία είναι η «owl– api 2.2.0» και η έκδοση της μηχανής συμπερασμού Pellet είναι η «pellet 2.2.2».

ΚΕΦΑΛΑΙΟ 6

ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΟΣ

Το προτεινόμενο σύστημα αυτόματης παραγωγής νέας γνώσης (Rules Discovery) στα πλαίσια του Σημασιολογικού Ιστού αξιολογήθηκε ποιοτικά και ποσοτικά ώστε να αποτιμηθεί η αξία του και το ποσοστό εκπλήρωσης των αρχικών απαιτήσεών του. Στην παρούσα ενότητα θα περιγραφεί το σενάριο με βάση το οποίο αξιολογήθηκε το σύστημα, τα αποτελέσματα των πειραματικών δοκιμών και κάποια συμπεράσματα. Η παρούσα αξιολόγηση έχει στόχο τόσο την μέτρηση των επιδόσεων του συστήματος όσο και την επικύρωση της ορθής λειτουργίας του.

Η αξιολόγηση του συστήματος χωρίζεται σε δύο μέρη: την ποιοτική αξιολόγηση και την αξιολόγηση των επιδόσεων. Η ποιοτική αξιολόγηση αφορά τον έλεγχο της ορθής λειτουργίας του συστήματος, ενώ η αξιολόγηση των επιδόσεων αναφέρεται στο χρόνο που απαιτείται για την εκτέλεση της εφαρμογής. Σαν χρόνος απόκρισης ορίζεται η χρονική διάρκεια της διαδικασίας παραγωγής κανόνων.

6.1 Σενάριο Αξιολόγησης

6.1.1 Μετρικές

Για την αξιολόγηση του συστήματος χρησιμοποιήθηκε το σενάριο, στο οποίο έχουν οριστεί οι παράμετροι του συστήματος ως εξής:

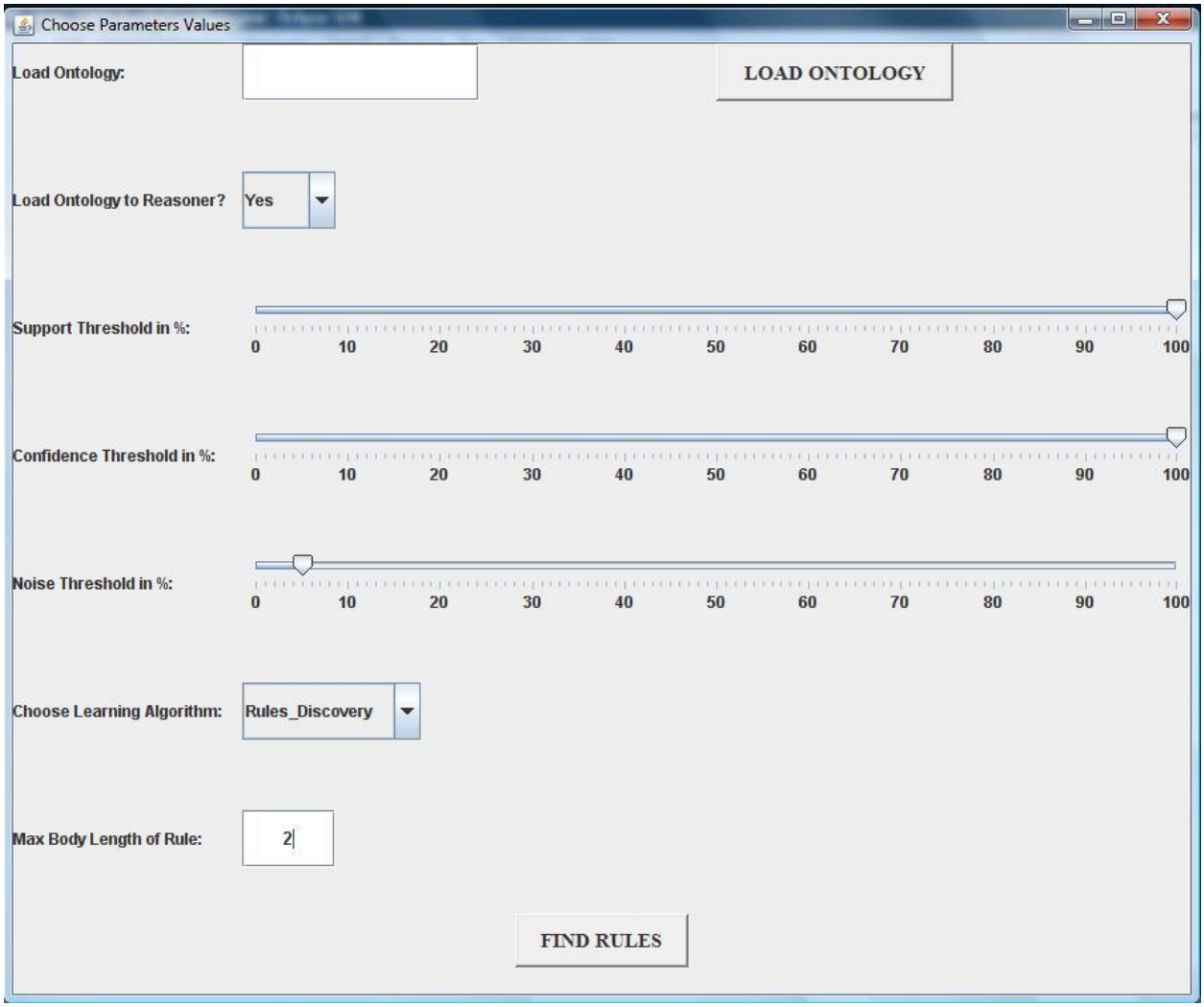
Ελάχιστη τιμή υποστήριξης κανόνων = 100%

Ελάχιστη τιμή εμπιστοσύνης κανόνων = 100%

Μέγιστη τιμή θορύβου που είναι επιτρεπτός = 5%

Μέγιστο πλήθος στοιχείων σώματος κανόνα = 2

Ο καθορισμός των αρχικών παραμέτρων του συστήματος φαίνονται στην εικόνα 6.1.



Εικόνα 6.1: Αρχικοί παράμετροι συστήματος

6.1.2 Σύνολο Οντολογιών

Οι οντολογίες που χρησιμοποιήθηκαν για τον έλεγχο των επιδόσεων του συστήματος είναι: η οντολογία μοντελοποίησης μαθημάτων (University.xml) [67] και η οντολογία περιγραφής συγγενικών σχέσεων (Family.xml) [66], η οποία χρησιμοποιήθηκε και για την ποιοτική αξιολόγησή του. Τα γενικά χαρακτηριστικά των οντολογιών φαίνονται στον Πίνακα 6.1.

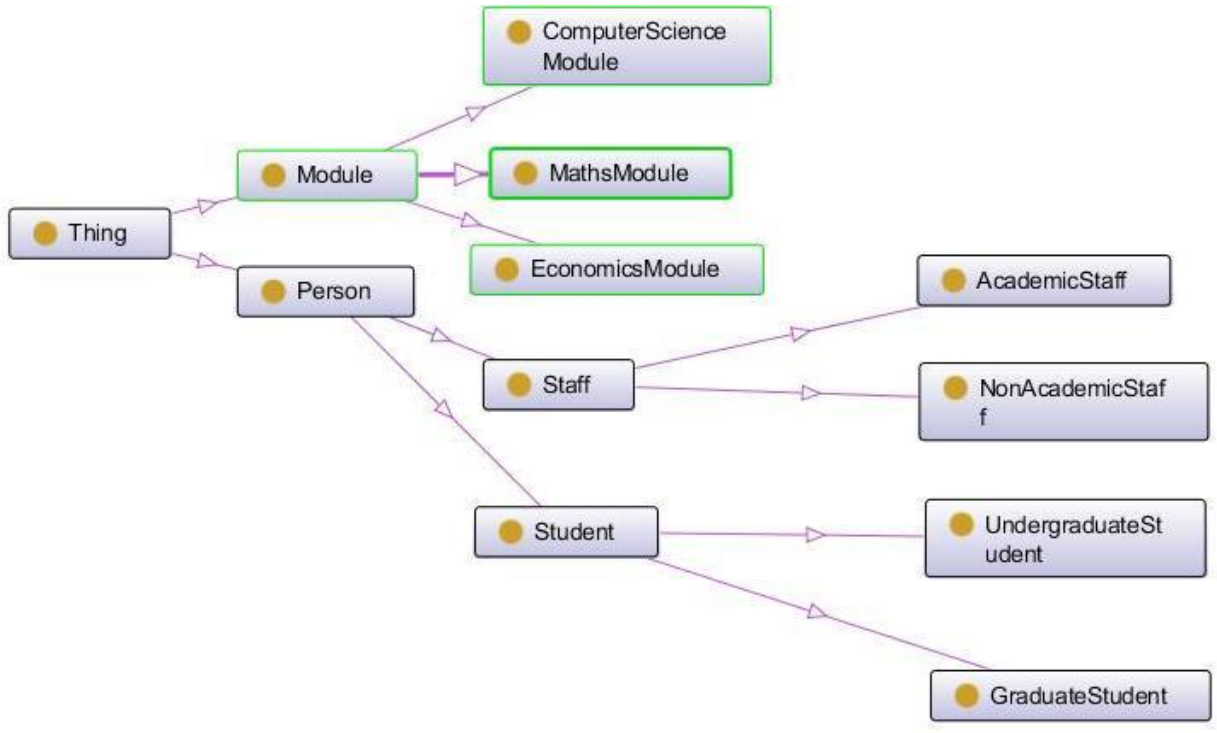
Πίνακας 6.1: Γενικά χαρακτηριστικά οντολογιών

Οντολογία / Χαρακτηριστικά	Πλήθος Κλάσεων	Πλήθος δυαδικών συσχετίσεων	Πλήθος συσχετίσεων δεδομένων (Data Properties)	Πλήθος Στιγμιοτύπων
University	12	4		29
Family	11	16		24

Ένα τμήμα της περιγραφής των οντολογιών φαίνεται παρακάτω:

Οντολογία μοντελοποίησης μαθημάτων (University.xml)

Η οντολογία (University) [67] αποτελεί ένα σημασιολογικό μοντέλο περιγραφής και ορισμού ενός μέρους των βασικών και δομικών εννοιών των μαθημάτων, όπως επίσης και των μεταξύ τους συσχετίσεων. Η βασική εννοιολογική ιεραρχία (concept taxonomy) της οντολογίας απεικονίζεται στην Εικόνα 6.2.



Εικόνα 6.2: Ιεραρχία εννοιών της University.xml

Η University εκτός από έννοιες περιέχει και ρόλους (δυαδικές σχέσεις μεταξύ εννοιών), αξιώματα και περιορισμούς. Στη συνέχεια παρουσιάζεται ένα τμήμα της προδιαγραφής

της University. Ο Πίνακας 6.2 περιέχει κάποιες από τις προδιαγραφές των πρωταρχικών (primitive) και των ορισμένων (defined) εννοιών. Ο Πίνακας 6.3 περιέχει τις αντίστοιχες προδιαγραφές για τις δυαδικές σχέσεις της οντολογίας και ο Πίνακας 6.4 επεξηγεί με περισσότερη λεπτομέρεια τις πιο βασικές σχέσεις της οντολογίας.

Πίνακας 6.2: Λεξικό εννοιών (Concept Dictionary) της οντολογίας University

Όνομα έννοιας	Χαρακτηριστικά	Σχέσεις	Περιγραφή
Module		assistsWith, takes, teaches	Η έννοια του μαθήματος
ComputerScienceModule			Μάθημα Επιστήμης Υπολογιστών
MathsModule			Μαθηματικά
EconomicsModule			Οικονομικά
Person			Η έννοια του ατόμου
Staff			Προσωπικό
AcademicStaff		teaches	Ακαδημαϊκό Προσωπικό
NonAcademicStaff			Μη Ακαδημαϊκό Προσωπικό
Student			Φοιτητής
GraduateStudent			Μεταπτυχιακός Φοιτητής
UndergraduateStudent		takes	Προπτυχιακός Φοιτητής

Πίνακας 6.3: Πίνακας Δυαδικών Σχέσεων της οντολογίας University

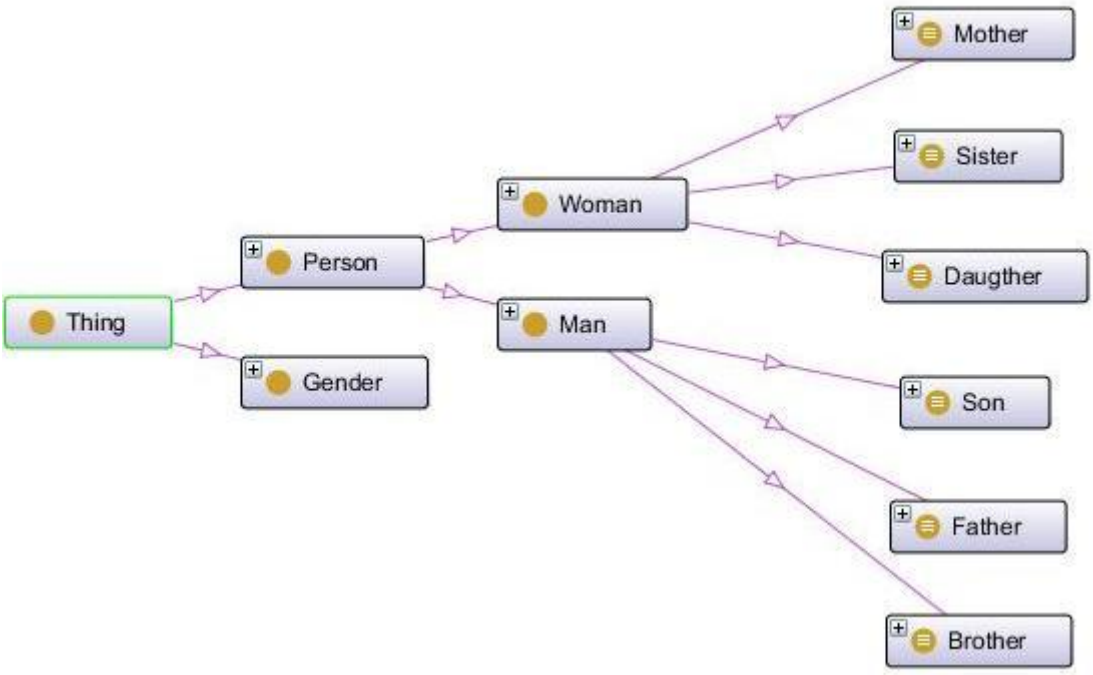
Όνομα σχέσης	Υποκείμενο	Πληθικό-τητα	Αντικείμενο/ Τύπος δεδομένων	Μαθηματικές ιδιότητες
assistsWith	-	=1	Module	-
isTaughtBy	Module	>=1	-	-
takes	UndergraduateStudent	>=1	Module	-
teaches	AcademicStaff	>=1	Module	-

Πίνακας 6.4: Περιγραφή βασικών σχέσεων της οντολογίας University

Σχέση	Περιγραφή
assistsWith	Η σχέση αυτή συσχετίζει ένα Person με κάποιο Module.
isTaughtBy	Δηλώνει ποιο μάθημα διδάσκεται από ποιο μέλος του ακαδημαϊκού προσωπικού.
takes	Δηλώνει ποια μαθήματα παρακολουθεί κάθε φοιτητής.
teaches	Δηλώνει ποια μαθήματα διδάσκει κάθε μέλος του ακαδημαϊκού προσωπικού.

Οντολογία μοντελοποίησης συγγενικών σχέσεων (Family.xml)

Η οντολογία (Family) [66] αποτελεί ένα σημασιολογικό μοντέλο περιγραφής και ορισμού ενός μέρους των βασικών και δομικών εννοιών των συγγενικών σχέσεων, όπως επίσης και των μεταξύ τους συσχετίσεων. Η βασική εννοιολογική ιεραρχία (taxonomy) της οντολογίας απεικονίζεται στην Εικόνα 6.3.



Εικόνα 6.3: Ιεραρχία εννοιών της οντολογίας Family

Η Family εκτός από έννοιες περιέχει και ρόλους (δυαδικές σχέσεις μεταξύ εννοιών), αξιώματα και περιορισμούς. Στη συνέχεια παρουσιάζεται ένα τμήμα της προδιαγραφής της Family. Ο Πίνακας 6.5 περιέχει κάποιες από τις προδιαγραφές των πρωταρχικών (primitive) και των ορισμένων (defined) εννοιών. Ο Πίνακας 6.6 περιέχει τις αντίστοιχες προδιαγραφές για τις δυαδικές σχέσεις της οντολογίας και ο Πίνακας 6.7 επεξηγεί με περισσότερη λεπτομέρεια τις πιο βασικές σχέσεις της οντολογίας.

Πίνακας 6.5: Λεξικό εννοιών (Concept Dictionary) της οντολογίας Family

Όνομα έννοιας	Χαρακτηριστικά	Σχέσεις	Περιγραφή
Gender		hasSex	Η έννοια του φύλου
Person		has Aunt, hasChild, hasDaughter, hasSon, hasConsort, hasNephew, hasNiece, hasParent, hasFather, hasMother, hasSex, hasSibling, hasBrother, hasSister, hasUncle	Η έννοια του ατόμου
Daughter			Η κόρη
Son			Ο γιος
Man		hasSon, hasNephew, hasFather, hasBrother, hasUncle	Ο άνδρας
Brother			Ο αδερφός
Father			Ο πατέρας
Mother			Η μητέρα
Woman		has Aunt, hasDaughter, hasNiece, hasMother, hasSister	Η γυναίκα
Sister			Η αδερφή

Πίνακας 6.6: Πίνακας Δυαδικών Σχέσεων της οντολογίας Family

Όνομα σχέσης	Υποκείμενο	Πληθικό-τητα	Αντικείμενο/ Τύπος δεδομένων	Μαθηματικές ιδιότητες
has Aunt	Person	>=1	Woman	-
hasChild	Person	>=1	Person	-
hasDaughter	Person	-	Woman	-
hasSon	Person	-	Son	-
hasConsort	Person	=1	Person	Συμμετρική
hasNephew	Person	>=1	Man	
hasNiece	Person	>=1	Woman	
hasParent	Person	<=2	Person	
hasFather	Person	=1	Man	
hasMother	Person	=1	Woman	
hasSex	Person	=1	Gender	
hasSibling	Person	>=1	Person	Συμμετρική
hasBrother	Person	>=1	Man	
hasSister	Person	>=1	Sister	
hasUncle	Person	>=1	Man	

Πίνακας 6.7: Περιγραφή βασικών σχέσεων της οντολογίας Family

Σχέση	Περιγραφή
has Aunt	Δηλώνει ποιο Person έχει κάποια θεία
hasChild	Δηλώνει ποιο Person έχει κάποιο παιδί
hasDaughter	Δηλώνει ποιο Person έχει κόρη
hasSon	Δηλώνει ποιο Person έχει γιο
hasConsort	Δηλώνει ποιο Person έχει σύζυγο κάποιο άλλο Person
hasNephew	Δηλώνει ποιο Person έχει ανιψιό
hasNiece	Δηλώνει ποιο Person έχει ανιψιά
hasParent	Δηλώνει ποιο Person έχει κάποιο γονιό
hasFather	Δηλώνει ποιο Person έχει κάποιο Man ως πατέρα
hasMother	Δηλώνει ποιο Person έχει κάποια Woman ως μητέρα
hasSex	Δηλώνει ποιο είναι το φύλο του Person
hasSibling	Δηλώνει ποιο Person έχει αδερφό ή αδερφή κάποιο άλλο Person
hasBrother	Δηλώνει ποιο Person έχει αδερφό
hasSister	Δηλώνει ποιο Person έχει αδερφή
hasUncle	Δηλώνει ποιο Person έχει θείο

6.1.3 Ρυθμίσεις Συστήματος

Τέλος, οι τεχνολογίες υλοποίησης που χρησιμοποιήθηκαν είναι αυτές που αναφέρθηκαν στην παράγραφο 5.6, ενώ το υπολογιστικό σύστημα στο οποίο εκτελέστηκαν τα πειράματα έχει τα ακόλουθα χαρακτηριστικά:

- Επεξεργαστής (CPU): Intel Core Duo 2.50 GHz
- Μνήμη RAM: 3 GigaBytes
- Λειτουργικό Σύστημα: Windows Vista Home Edition
- Ειδικές ρυθμίσεις: Καμία

6.2 Αποτελέσματα Αξιολόγησης

6.2.1 Αξιολόγηση Επιδόσεων

Στη συνέχεια παρουσιάζονται τα αποτελέσματα της εκτέλεσης της διαδικασίας παραγωγής κανόνων για τις οντολογίες της παραγράφου 6.1.2 και με τις μετρικές που επίσης αναφέρθηκαν στην παράγραφο 6.1.1. Από κάθε οντολογία δημιουργήθηκαν οι κανόνες είτε προηγήθηκε το στάδιο του συμπερασμού είτε όχι.

Οντολογία University.xml

Οι προκύπτοντες κανόνες για την οντολογία University χωρίς τη διαδικασία συμπερασμού παρουσιάζονται στον Πίνακα 6.8.

Πίνακας 6.8: Κανόνες οντολογίας University.xml χωρίς διαδικασία συμπερασμού

	Support(%)	Confidence(%)	Κανόνας
1.	100	100	AcademicStaff(?x) ^ Module(?y) ^ teaches(?x,?y) -> isTaughtBy(?y,?x)
2	100	100	isTaughtBy(?x,?y) -> teaches(?y,?x)
3.	100	100	isTaughtBy(?x,?y) -> AcademicStaff(?y)

Σύνολο κανόνων = 3

Χρόνος Απόκρισης Διαδικασίας = 172 ms.

Οι προκύπτοντες κανόνες για την οντολογία University μετά τη διαδικασία συμπερασμού παρουσιάζονται στον Πίνακα 6.9.

Πίνακας 6.9: Κανόνες οντολογίας University.xml μετά τη διαδικασία συμπερασμού

	Support(%)	Confidence(%)	Κανόνας
1.	100	100	Person(?x) ^ Module(?y) ^ teaches(?x,?y) -> isTaughtBy(?y,?x)
2.	100	100	isTaughtBy(?x,?y) -> teaches(?y,?x)
3.	100	100	isTaughtBy(?x,?y) -> Module(?x)
4.	100	100	isTaughtBy(?x,?y) -> Staff(?y)
5.	100	100	Student(?x) -> UndergraduateStudent(?x)
6.	100	100	isTaughtBy(?x,?y) -> AcademicStaff(?y)
7.	100	100	Staff(?x) -> AcademicStaff(?x)

Οι προκύπτοντες περιττοί κανόνες για την οντολογία University μετά τη διαδικασία συμπερασμού παρουσιάζονται στον Πίνακα 6.10.

Πίνακας 6.10: Περιττοί κανόνες οντολογίας University.xml μετά τη διαδικασία συμπερασμού

	Support(%)	Confidence(%)	Κανόνας
1.	**100**	**100**	**UndergraduateStudent(?x) -> Student(?x)**
2.	**100**	**100**	**AcademicStaff(?x) -> Staff(?x)**

Σύνολο κανόνων = 7

Σύνολο περιττών κανόνων = 2

Χρόνος Απόκρισης Διαδικασίας = 766 ms.

Οντολογία Family.xml

Ένα τμήμα των προκυπτόντων κανόνων για την οντολογία Family χωρίς τη διαδικασία συμπερασμού παρουσιάζονται στον Πίνακα 6.11.

Πίνακας 6.11: Τμήμα κανόνων οντολογίας Family.xml χωρίς διαδικασία συμπερασμού

	Support(%)	Confidence(%)	Κανόνας
1.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ Person(?z) ^ hasParent(?z,?x) ^ hasSister(?x,?y) -> hasAunt(?z,?y)**
2.	**100**	**100**	**Person(?x) ^ Person(?y) ^ hasChild(?x,?y) -> hasParent(?y,?x)**
3.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ Person(?z) ^ hasChild(?z,?x) ^ hasMother(?x,?y) -> hasConsort(?z,?y)**
4.	**100**	**100**	**Person(?x) ^ Person(?y) ^ hasParent(?x,?y) -> hasChild(?y,?x)**
5.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Man(?z) ^ hasBrother(?y,?z) ^ hasParent(?x,?y) -> hasUncle(?x,?z)**
6.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Woman(?z) ^ hasDaughter(?y,?z) ^ hasParent(?x,?y) -> hasSister(?x,?z)**
7.	**100**	**100**	**Person(?x) ^ Man(?y) ^ Person(?y) ^ Person(?z) ^ hasConsort(?y,?z) ^ hasFather(?x,?y) ->**

			hasMother(?x,?z)
8.	**100**	**100**	**Person(?x) ^ Man(?y) ^ Person(?z) ^ hasParent(?z,?x) ^ hasSon(?x,?y) -> hasBrother(?z,?y)**
9.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ Person(?z) ^ hasChild(?x,?z) ^ hasSister(?x,?y) -> hasAunt(?z,?y)**
10.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Man(?z) ^ hasBrother(?x,?z) ^ hasChild(?x,?y) -> hasUncle(?y,?z)**
11.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Person(?z) ^ hasConsort(?y,?z) ^ hasParent(?x,?y) -> hasFather(?x,?y)**
12.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Person(?z) ^ hasChild(?x,?z) ^ hasConsort(?x,?y) -> hasFather(?z,?x)**
13.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ Person(?z) ^ hasChild(?z,?x) ^ hasMother(?x,?y) -> hasFather(?x,?z)**
14.	**100**	**100**	**Person(?x) ^ Man(?y) ^ Person(?y) ^ Woman(?z) ^ hasDaughter(?y,?z) ^ hasFather(?x,?y) -> hasSister(?x,?z)**

Σύνολο κανόνων = 20

Χρόνος Απόκρισης Διαδικασίας = 5.5 sec.

Ένα τμήμα των προκυπτόντων κανόνων για την οντολογία Family μετά τη διαδικασία συμπερασμού παρουσιάζονται στον Πίνακα 6.12.

Πίνακας 6.12: Τμήμα κανόνων οντολογίας Family.xml μετά τη διαδικασία συμπερασμού

	Support(%)	**Confidence(%)**	**Κανόνας**
1.	**100**	**100**	**Woman(?x) ^ Person(?y) ^ Person(?z) ^ hasParent(?z,?y) ^ hasSibling(?x,?y) -> hasAunt(?z,?x)**
2.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ Person(?z) ^ hasParent(?z,?x) ^ hasSister(?x,?y) -> hasAunt(?z,?y)**
3.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Woman(?z) ^**

			hasChild(?x,?z) ^ hasSibling(?x,?y) -> hasNiece(?y,?z)
4.	**100**	**100**	**Person(?x) ^ Brother(?y) ^ Person(?z) ^ hasParent(?z,?x) ^ hasSibling(?x,?y) -> hasUncle(?z,?y)**
5.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Man(?z) ^ hasBrother(?y,?z) ^ hasParent(?x,?y) -> hasUncle(?x,?z)**
6.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Man(?z) ^ hasChild(?x,?z) ^ hasSibling(?x,?y) -> hasNephew(?y,?z)**
7.	**100**	**100**	**Woman(?x) ^ Person(?y) ^ hasParent(?x,?y) -> hasDaughter(?y,?x)**
8.	**100**	**100**	**Woman(?x) ^ Person(?y) ^ Person(?z) ^ hasConsort(?y,?z) ^ hasParent(?x,?y) -> hasDaughter(?z,?x)**
9.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ hasSibling(?x,?y) -> hasSister(?x,?y)**
10.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ Man(?z) ^ hasConsort(?y,?z) ^ hasMother(?x,?y) -> hasFather(?x,?z)**
11.	**100**	**100**	**Man(?x) ^ Person(?y) ^ hasSibling(?x,?y) -> hasBrother(?y,?x)**
12.	**100**	**100**	**Man(?x) ^ Woman(?y) ^ hasSibling(?x,?y) -> Sister(?y)**
13.	**100**	**100**	**Man(?x) ^ Woman(?y) ^ hasConsort(?x,?y) -> Mother(?y)**
14.	**100**	**100**	**Man(?x) ^ Son(?y) ^ Person(?z) ^ hasSibling(?y,?z) ^ hasSon(?x,?y) -> hasBrother(?z,?y)**

Ένα τμήμα των προκυπτόντων περιττών κανόνων για την οντολογία Family μετά τη διαδικασία συμπερασμού παρουσιάζονται στον Πίνακα 6.13.

Πίνακας 6.13: Τμήμα περιττών κανόνων οντολογίας Family.xml μετά τη διαδικασία συμπερασμού

	Support(%)	**Confidence(%)**	**Κανόνας**
1.	**100**	**100**	**Person(?x) ^ Brother(?y) ^ hasBrother(?x,?y) -> hasSibling(?x,?y)**
2.	**100**	**100**	**Person(?x) ^ Daugther(?y) ^ hasSister(?x,?y) -> hasSibling(?x,?y)**
3.	**100**	**100**	**Person(?x) ^ Father(?y) ^ hasFather(?x,?y) -> hasParent(?x,?y)**

4.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ hasDaughter(?x,?y) -> hasChild(?x,?y)**
5.	**100**	**100**	**Woman(?x) ^ Person(?y) ^ hasSister(?y,?x) -> hasSibling(?x,?y)**
6.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ hasMother(?x,?y) -> hasParent(?x,?y)**
7.	**100**	**100**	**Person(?x) ^ Son(?y) ^ hasBrother(?x,?y) -> hasSibling(?x,?y)**
8.	**100**	**100**	**Person(?x) ^ Daugther(?y) ^ hasDaughter(?x,?y) -> hasChild(?x,?y)**
9.	**100**	**100**	**Person(?x) ^ Man(?y) ^ hasFather(?x,?y) -> hasParent(?x,?y)**
10.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ hasSister(?x,?y) -> hasSibling(?x,?y)**
11.	**100**	**100**	**Person(?x) ^ Sister(?y) ^ hasSister(?x,?y) -> hasSibling(?x,?y)**
12.	**100**	**100**	**Brother(?x) ^ Person(?y) ^ hasBrother(?y,?x) -> hasSibling(?x,?y)**

Σύνολο κανόνων = 12244

Σύνολο περιττών κανόνων = 17

Χρόνος Απόκρισης Διαδικασίας = 4h & 7 min

Από τα πειράματα φαίνεται ότι ο χρόνος απόκρισης της διαδικασίας μετά το στάδιο του συμπερασμού είναι κατά πολύ μεγαλύτερος συγκριτικά με το χρόνο απόκρισης του συστήματος χωρίς αυτό. Η διαφορά αυτή οφείλεται στην ταξινόμηση των κλάσεων και των συσχετίσεων που προέκυψε μετά τη διαδικασία του συμπερασμού, με αποτέλεσμα πολλά στιγμιότυπα κάποιων κλάσεων να ταξινομηθούν και σε όλες τις κλάσεις, που βρίσκονται υψηλότερα στην ιεραρχία. Το ίδιο συμβαίνει και με την ιεραρχία των συσχετίσεων και την ταξινόμηση των συνδυασμών των στιγμιοτύπων. Αυτή η διαδικασία έχει ως αποτέλεσμα να αυξηθούν κατά πολύ οι συνδυασμοί που πρέπει να ελεγχθούν ως πιθανά σώματα (bodies) των κανόνων.

6.2.2 Ποιοτική Αξιολόγηση

Η ορθότητα της μεθόδου ελέγχθηκε με δύο τρόπους. Αρχικά, αξιολογήθηκε κατά πόσο είναι ικανή να παράγει τους κανόνες μιας οντολογίας που είναι γνωστοί εκ των προτέρων και επίσης αξιολογήθηκε η δυνατότητα να παράγει κανόνες, που προκύπτουν επίσης από τα ήδη γνωστά ILP συστήματα. Ο έλεγχος πραγματοποιήθηκε με την οντολογία Family.xml, για την οποία υπάρχουν SWRL κανόνες στην περιγραφή της. Οι κανόνες της Family.xml παρουσιάζονται στον πίνακα 6.14.

Πίνακας 6.14: Γνωστοί κανόνες οντολογίας Family.xml

	Κανόνας
1.	Man(?y) , Person(?x) , hasChild(?x, ?y) -> hasSon(?x, ?y)
2.	Man(?y) , Person(?x) , hasSibling(?x, ?y) -> hasBrother(?x, ?y)
3.	Person(?x) , Woman(?y) , hasChild(?x, ?y) -> hasDaughter(?x, ?y)
4.	Person(?x) , hasDaughter(?y, ?z) , hasSibling(?x, ?y) -> hasNiece(?x, ?z)
5.	Person(?x) , hasBrother(?y, ?z) , hasParent(?x, ?y) -> hasUncle(?x, ?z)
6.	Person(?x) , Woman(?y) , hasSibling(?x, ?y) -> hasSister(?x, ?y)
7.	Man(?y) , Person(?x) , hasParent(?x, ?y) -> hasFather(?x, ?y)
8.	Person(?x) , Woman(?y) , hasParent(?x, ?y) -> hasMother(?x, ?y)
9.	Person(?y) , hasConsort(?y, ?z) , hasParent(?x, ?y) -> hasParent(?x, ?z)
10.	Person(?y) , hasChild(?y, ?x) , hasChild(?y, ?z) , differentFrom(?x, ?z) -> hasSibling(?x, ?z)
11.	Person(?x) , hasParent(?x, ?y) , hasSister(?y, ?z) -> hasAunt(?x, ?z)
12.	Person(?x) , hasSibling(?x, ?y) , hasSon(?y, ?z) -> hasNephew(?x, ?z)

Οι κανόνες που προκύπτουν από τη διαδικασία παραγωγής κανόνων του συστήματος φαίνονται στον Πίνακα 6.15.

Πίνακας 6.15: Προκύπτοντες κανόνες αυτόματης διαδικασίας παραγωγής κανόνων

	S(%)	C(%)	Κανόνας
1.	100	100	Person(?x) ^ Man(?y) ^ hasChild(?x,?y) -> hasSon(?x,?y)
2.	100	100	Person(?x) ^ Man(?y) ^ hasSibling(?x,?y) -> hasBrother(?x,?y)
3.	100	100	Person(?x) ^ Woman(?y) ^ hasChild(?x,?y) -> hasDaughter(?x,?y)
4.	100	100	Person(?x) ^ Woman(?y) ^Person(?z) ^ hasDaughter(?x,?y) ^ hasSibling(?x,?z) -> hasNiece(?z,?y)
5.	100	100	Person(?y) ^ Person(?x) ^ Man(?z) ^ hasBrother(?y, ?z) ^

			hasParent(?x, ?y) -> hasUncle(?x, ?z)
6.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ hasSibling(?x,?y) -> hasSister(?x,?y)**
7.	**100**	**100**	**Person(?x) ^ Man(?y) ^ hasParent(?x,?y) -> hasFather(?x,?y)**
8.	**100**	**100**	**Person(?x) ^ Woman(?y) ^ hasParent(?x,?y) -> hasMother(?x,?y)**
9.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Person(?z) ^ hasConsort(?x, ?y) ^ hasParent(?z, ?y) -> hasParent(?z, ?x)**
10.	**50** **50**	**100** **100**	**Person(?x) ^ Person(?y) ^ Person(?z) ^ hasChild(?x, ?y) ^ hasChild(?x, ?z) ^ differentFrom(?y,?z) -> hasSibling(?y, ?z)** **Person(?x) ^ Person(?y) ^ Person(?z) ^ hasChild(?x, ?y) ^ hasChild(?x, ?z) ^ differentFrom(?y,?z) -> hasSibling(?z, ?y)**
11.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Woman(?z) ^ hasParent(?x, ?y) ^ hasSister(?y, ?z) -> hasAunt(?x, ?z)**
12.	**100**	**100**	**Person(?x) ^ Person(?y) ^ Man(?z) ^ hasSibling(?x, ?y) ^ hasSon(?y, ?z) -> hasNephew(?x, ?z)**

Από τη σύγκριση των κανόνων φαίνεται ότι οι κανόνες που προκύπτουν αυτόματα μέσω της διαδικασίας παραγωγής κανόνων για την οντολογία είναι ίδιοι με τους κανόνες της οντολογίας. Η μόνη διαφορά παρατηρείται στον κανόνα 10, όπου η μηχανή παραγωγής κανόνων δημιούργησε δύο υποπεριπτώσεις του ίδιου κανόνα, οι οποίοι έχουν ακριβώς το ίδιο σώμα (body), αλλά διαφορετικές κεφαλές (head). Η δημιουργία των δύο κανόνων οφείλεται στο γεγονός ότι ως κεφαλή υπάρχει μια ιδιότητα (hasSibling) η οποία είναι συμμετρική (symmetric), δηλαδή μπορεί να αντιστρέφεται το υποκείμενο με το αντικείμενο της. Αυτή η ιδιότητα οδηγεί στη δημιουργία δύο διαφορετικών μορφών κανόνων με το ίδιο νόημα. Επίσης, παρατηρείται ότι και η τιμή της υποστήριξης των δύο κανόνων είναι 50%, αυτό συμβαίνει γιατί οι συνδυασμοί που ικανοποιούν την κεφαλή των κανόνων μοιράζονται στις δυο υποπεριπτώσεις κανόνων.

Η διαφορετική σειρά των μεταβλητών στους κανόνες οφείλεται στη διαδικασία παραμετροποίησης των κανόνων που ακολουθείται.

6.2.2.1 Συγκριτική Ποιοτική Αξιολόγηση

Στα πλαίσια της συγκριτικής αξιολόγησης του συστήματος αυτόματης παραγωγής κανόνων χρησιμοποιήθηκαν τα εξής ILP συστήματα:

- **FOIL:** Το FOIL [68] είναι ένα «από πάνω προς τα κάτω» (top-down) σχεσιακό ILP σύστημα. Εφαρμόζει μια ευριστική στρατηγική αναζήτησης των λύσεων, που περιορίζει κατά πολύ το χώρο αναζήτησης. Σαν γενική στρατηγική αναζήτησης χρησιμοποιεί μια στρατηγική κάλυψης (covering approach) των δοθέντων παραδειγμάτων. Ξεκινά την αναζήτηση με κενό σώμα και αναζητά στοιχεία στο χώρο αναζήτησης. Σταματά την εισαγωγή στοιχείων στο σώμα των κανόνων, αν η δημιουργημένη πρόταση έχει την ελάχιστη επιτρεπτή τιμή ακρίβειας ή αν το πλήθος των στοιχείων στο σώμα έχει φτάσει στο μέγιστο δυνατό. Η εισαγωγή αρνητικών παραδειγμάτων δεν είναι απαραίτητη καθώς το FOIL μπορεί να δημιουργήσει αρνητικά παραδείγματα βασισμένο στην υπόθεση κλειστού κόσμου (Closed World Assumption).
- **GOLEM:** Το GOLEM [69] μπορεί να διαχειριστεί αποδοτικά μεγάλα σύνολα δεδομένων. Αυτό συμβαίνει γιατί αποφεύγει την εξέταση μεγάλου τμήματος του χώρου αναζήτησης. Δημιουργεί μια μοναδική πρόταση που καλύπτει ένα σύνολο θετικών παραδειγμάτων της βάσης γνώσης. Βασίζεται στη χρήση των ελάχιστων σχετικών γενικών γενικεύσεων (relative least general generalizations, rlgg) και προσαρμόζει τη χρήση τους σε μια καλυπτική προσέγγιση. Για τη δημιουργία της μοναδικής πρότασης επιλέγει τυχαία ένα σύνολο θετικών παραδειγμάτων και υπολογίζει το rlgg τους. Από αυτά, επιλέγει αυτό που καλύπτει τα περισσότερα θετικά παραδείγματα και είναι συνεπές με τα αρνητικά παραδείγματα. Αυτή η πρόταση γενικεύεται επιπλέον. Η διαδικασία της γενίκευσης σταματάει όταν η κάλυψη της καλύτερης πρότασης σταματήσει να αυξάνεται. Το GOLEM περιλαμβάνει ένα στάδιο μετα-επεξεργασίας που περιλαμβάνει την αφαίρεση άσχετων στοιχείων από τις προτάσεις.
- **PROGOL:** Το PROGOL [70] είναι ένα «από πάνω προς τα κάτω» (top-down) σχεσιακό ILP σύστημα, που βασίζεται στην αντίστροφη συνεπαγωγή (inverse entailment). Περιορίζει το χώρο αναζήτησης χρησιμοποιώντας ένα σύνολο δηλώσεων (mode declarations), που ορίζει ο χρήστης και την πιο ειδική πρόταση (bottom clause), που ορίζεται ως το τελικό όριο αναζήτησης κανόνων. Ο ειδικός κανόνας είναι ο περισσότερο ειδικός κανόνας και καλύπτει ένα θετικό παράδειγμα, που προκύπτει από την αντίστροφη συνεπαγωγή. Το PROGOL ξεκινά την

αναζήτηση με κενό σώμα και αναζητά στοιχεία στο χώρο αναζήτησης, που υπάρχουν ταυτόχρονα και στην bottom clause. Το PROGOL κρατάει μόνο το κανόνα, που έχει τη μέγιστη τιμή της μετρικής αξιολόγησης.

- **ALEPH:** Το ALEPH [71] είναι ένα «από πάνω προς τα κάτω» (top-down) σχεσιακό ILP σύστημα, που βασίζεται στην αντίστροφη συνεπαγωγή (inverse entailment) όπως και το PROGOL. Ο βασικός αλγόριθμος του είναι ίδιος με τον αλγόριθμο του PROGOL, ενώ μπορεί να εφαρμόσει διαφορετικές στρατηγικές αναζήτησης, μετρικές αξιολόγησης και τελεστές βελτιστοποίησης (refinement operators). Επίσης, μπορούν να οριστούν ρυθμίσεις ελάχιστης υποστήριξης και εμπιστοσύνης.

Για την αξιολόγηση ήταν απαραίτητη η μετατροπή της οντολογίας Family σε κατάλληλες μορφές για το κάθε σύστημα, μιας και κανένα από τα παραπάνω συστήματα δεν μπορούν να δεχθούν ως είσοδο τους μια οντολογία. Αυτή η μετατροπή δεν ήταν εφικτή χωρίς την αφαίρεση κάποιων αξιωμάτων από την οντολογία, όπως αντίστροφων ιδιοτήτων (inverse properties), ξένες κλάσεις (disjoint classes) κ.α.

Παρακάτω εμφανίζονται οι κανόνες για την οντολογία Family που προέκυψαν από τα παραπάνω συστήματα:

Πίνακας 6.16: Προκύπτοντες κανόνες από το σύστημα FOIL

	Κανόνας
1.	man(A) :- son(A).
2.	man(A) :- father(A).
3.	woman(A) :- hasmother(B,A).
4.	woman(A) :- daughter(A).
5.	son(A) :- hasson(B,A).
6.	father(A) :- hasfather(B,A).
7.	daughter(A) :- hasdaughter(B,A).
8.	mother(A) :- hasmother(B,A), hasparent(B,A).
9.	brother(A) :- hasbrother(B,A).
10.	sister(A) :- hasaunt(B,A).
11.	hasniece(A,B) :- woman(B), hasfather(B,C), hassibling(A,C).
12	hasniece(A,B) :- hasmother(B,C), hasconsort(D,B), hassibling(A,C).
13.	hasaunt(A,B) :- hasmother(A,C), hassister(C,B).
14.	hasaunt(A,B) :- hasfather(A,C), hassister(C,B).
15.	hasmother(A,B) :- hasfather(A,C), hasconsort(C,B).
16.	hasfather(A,B) :- hasparent(A,B), man(B).
17.	hassibling(A,B) :- hasbrother(A,B).
18.	hassibling(A,B) :- hassister(A,B).

19.	**haschild(A,B) :- hasparent(B,A).**
20.	**hasuncle(A,B) :- hasfather(A,C), hasmother(A,D), hasmother(C,E), hasbrother(D,B).**
21.	**hasuncle(A,B) :- hasfather(A,C), hasbrother(C,B).**
22.	**hasdaughter(A,B) :- hasparent(B,A), woman(B).**
23.	**hasconsort(A,B) :- father(A), hasmother(C,B), hasparent(C,A).**
24.	**hassister(A,B) :- hassibling(A,B), woman(B).**
25.	**hasbrother(A,B) :- hassibling(A,B), man(B).**

Πίνακας 6.17: Προκύπτοντες κανόνες από το σύστημα GOLEM

	Κανόνας
1.	**hasaunt(A,B).**
2.	**hassister(A,B) :- hassibling(A,B).**
3.	**hasparent(f05,m04).**
4.	**hasparent(A,B) :- haschild(B,A).**
5.	**haschild(f04,f05).**
6.	**haschild(A,B) :- hasparent(B,A).**
7.	**hasuncle(A,B).**
8.	**hasbrother(A,B) :- hassibling(A,B).**
9.	**man(A).**
10.	**son(A) :- hasfather(A,B).**
11.	**brother(A) :- hasmother(A,B), mother(B).**
12	**father(A) :- hasparent(B,A).**
13.	**woman(A).**
14.	**daughter(A) :- hasfather(A,B).**
15.	**mother(A) :- hasparent(B,A).**
16.	**sister(A) :- daughter(A).**
17.	**hassex(A,female) :- woman(A).**
18.	**hassex(A,male) :- man(A).**
19.	**hasnephew(m02,m06).**
20.	**hasnephew(A,B) :- hasaunt(B,A).**
21.	**hasniece(A,B) :- daughter(B).**
22.	**hasconsort(A,B) :- hasparent(C,A), hasmother(C,B).**
23.	**hasdaughter(m01,f02).**
24.	**hasdaughter(m01,f03).**
25.	**hasdaughter(f01,f03).**
26.	**hasdaughter(A,B) :- hasfather(B,C), hasparent(C,D).**
27.	**hassibling(A,B) :- hasfather(A,C), hasfather(B,C).**
28.	**hasfather(A,B) :- hasparent(A,B).**

29.	**hasson(A,B) :- hasparent(B,A).**
30.	**hasmother(A,B) :- hasconsort(C,B), hasfather(A,C).**

Πίνακας 6.18: Τμήμα προκυπτόντων κανόνων από το σύστημα PROGOL

	Κανόνας
1.	**hasaunt(A,B) :- hasparent(A,C), hassister(C,B).**
2.	**hasparent(A,B) :- hasfather(A,B).**
3.	**hasparent(A,B) :- hasmother(A,B).**
4.	**hasparent(A,B) :- haschild(B,A).**
5.	**hasconsort(A,B) :- hasparent(C,A), hasparent(C,B).**
6.	**haschild(A,B) :- hasdaughter(A,B).**
7.	**haschild(A,B) :- hasson(A,B).**
8.	**haschild(A,B) :- hasparent(B,A).**
9.	**hassibling(A,B) :- hasbrother(B,A).**
10.	**hassibling(A,B) :- hasbrother(A,B).**
11.	**hassibling(A,B) :- hassister(A,B).**
12	**hassibling(A,B) :- hassister(B,A).**
13.	**hasuncle(A,B) :- hasparent(A,C), hassibling(B,C).**
18.	**hasdaughter(A,B) :- hasparent(B,A).**
19.	**woman(f07).**
20.	**woman(A) :- daugther(A).**
21.	**woman(A) :- mother(A).**
22.	**hassister(f03,f02).**
23.	**hassister(A,B) :- hassibling(A,B).**
24.	**hasmother(A,B) :- hasparent(A,B).**
25.	**hasfather(A,B) :- hasparent(A,B).**
26.	**hasbrother(A,B) :- hasparent(A,C), hassibling(A,B).**
27.	**hasson(m06,m09).**
28.	**hasson(A,B) :- hasparent(B,A).**
29.	**hasniece(A,B) :- hasaunt(B,A).**
30.	**mother(A) :- hasparent(B,A), woman(A).**

Πίνακας 6.19: Τμήμα προκυπτόντων κανόνων συστήματος ALEPH

	Κανόνας
1.	**hassex(A,B) :- hasparent(C,A).**
2.	**hassex(f02,female).**
3.	**hassex(A,B) :- hasaunt(A,C).**
4.	**hassex(f07,female).**
5.	**hassex(m10,male).**

6.	**hasnephew(A,B) :- hasson(C,B), hassibling(C,A).**
7.	**hasniece(A,B) :- hasdaughter(C,B), hassibling(C,A).**
8.	**hasparent(A,B) :- haschild(B,A).**
9.	**hasaunt(A,B) :- hasparent(A,C), hassister(C,B).**
10.	**hasconsort(A,B) :- hasparent(C,A), hasparent(C,B).**
11.	**hasconsort(m08,f06).**
12	**hasconsort(m07,f07).**
13.	**haschild(A,B) :- hasparent(B,A).**
14.	**hasuncle(A,B) :- hasparent(A,C), hasbrother(C,B).**
15.	**hasdaughter(f08,f09).**
16.	**…**
17.	**hasdaughter(f01,f03).**
18.	**hassister(A,B) :- hasdaughter(C,B), hasparent(A,C).**
19.	**hasbrother(A,B) :- hasparent(A,C), hasson(C,B).**
20.	**hassibling(A,B) :- hasparent(A,C), hasparent(B,C).**
21.	**hasfather(m10,m08).**
22.	**…**
23.	**hasfather(f09,m03).**
24.	**hasson(m06,m09).**
25.	**…**
26.	**hasson(f03,m06).**
27.	**hasmother(m10,f06).**
28.	**…**
29.	**hasmother(m06,f03).**
30.	**son(A).**
31.	**brother(A).**
32.	**father(A).**
33.	**daughter(A).**
34.	**mother(A).**
35.	**sister(A).**

Από τους κανόνες που προέκυψαν από τα συστήματα, παρατηρούμε ότι υπάρχει περίπτωση να παραχθούν κανόνες που δεν είναι safe (Πίνακας 6.19), ενώ όλα τα συστήματα εκτός από το FOIL δημιουργούν κανόνες που θεωρούνται πάντα αληθείς, δηλαδή να λείπει τελείως το σώμα του κανόνα, όπως επίσης να παρουσιάζουν ως κανόνες και κάποια από τα παραδείγματα της βάσης γνώσης.

Αντίθετα, στον προτεινόμενο αλγόριθμο (Rules Discovery) δε συμβαίνει το ίδιο. Όπως, αναφέρθηκε και στο Κεφάλαιο 5 δεν επιτρέπεται η δημιουργία unsafe κανόνων, η δημιουργία κανόνων χωρίς καθόλου σώμα και η εμφάνιση των απλών παραδειγμάτων

της βάσης γνώσης ως κανόνες. Όλοι οι υπόλοιποι κανόνες προκύπτουν και από το προτεινόμενο σύστημα και επιπλέον δημιουργούνται κανόνες, που τα συστήματα δεν μπόρεσαν να δημιουργήσουν.

Παρακάτω εμφανίζονται συνοπτικά το πλήθος των κανόνων που παράγει κάθε σύστημα, χωρίς να υπολογίζονται οι unsafe κανόνες, οι πάντα αληθείς και τα απλά παραδείγματα συγκριτικά με τους κανόνες του Rules Discovery για διαφορετικές τιμές της υποστήριξης (support) και της εμπιστοσύνης (confidence) (Πίνακας 6.20).

Πίνακας 6.20: Συγκριτικός Πίνακας Συστημάτων

Σύστημα		**Πλήθος Κανόνων**
FOIL		**25**
GOLEM		**19**
PROGOL		**34**
ALEPH		**10**
Rules Discovery	**S = 0%, C = 0%**	**2426**
	S = 50%, C = 60%	**283**
	S = 100%, C = 100%	**20**

ΚΕΦΑΛΑΙΟ 7

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΝΟΙΚΤΑ ΘΕΜΑΤΑ

7.1 Συμπεράσματα

Στην παρούσα εργασία παρουσιάστηκε η αρχιτεκτονική και οι λεπτομέρειες υλοποίησης ενός συστήματος αυτόματης παραγωγής νέας γνώσης από το Σημασιολογικό Ιστό. Αρχικά, παρουσιάστηκαν τα γενικά χαρακτηριστικά της αναπαράστασης γνώσης στο Σημασιολογικό Ιστό και περιγράφηκαν περιληπτικά οι τρόποι ανακάλυψης γνώσης. Στη συνέχεια, παρουσιάστηκαν συνοπτικά κάποια σχετικά συστήματα που πετυχαίνουν ανακάλυψη γνώσης από το Σημασιολογικό Ιστό. Τέλος, ακολούθησε η περιγραφή της αρχιτεκτονικής του συστήματος και αναλύθηκαν τα επιμέρους συστατικά της.

Η συνεισφορά της εργασίας συνοψίζεται στα ακόλουθα:

- Προσαρμογή διαδικασίας εξόρυξης δεδομένων από τα δεδομένα του Σημασιολογικού Ιστού. Η εφαρμογή μεθοδολογιών Μηχανικής Μάθησης στα δεδομένα του Σημασιολογικού Ιστού δεν έχει μελετηθεί αρκετά. Στην παρούσα εργασία παρουσιάζεται η εφαρμογή δημιουργίας κανόνων συσχέτισης από μια οντολογία.
- Αυτόματη παραγωγή νέας γνώσης από τα δεδομένα του Σημασιολογικού Ιστού. Τα περισσότερα συστήματα που εφαρμόζουν μεθοδολογίες ανακάλυψης γνώσης από το Σημασιολογικό Ιστό περιορίζονται στην σωστή ομαδοποίηση και κατηγοριοποίηση των δεδομένων του, σ' αυτήν την εργασία παρουσιάζεται η δυνατότητα παραγωγής νέας γνώσης σε μορφή SWRL κανόνων.
- Δυνατότητα επιλογής κανόνων και ενημέρωση των δεδομένων του Σημασιολογικού Ιστού. Ο σχεδιασμός της αρχιτεκτονικής της εργασίας προσφέρει τη δυνατότητα εμπλουτισμού της βάσης γνώσης με τη νέα γνώση που προέκυψε μέσω της διαδικασίας παραγωγής κανόνων. Η αρχιτεκτονική προσφέρει τη δυνατότητα επιλογής των πλέον κατάλληλων κανόνων για την ενημέρωση της βάσης γνώσης.
- Δυνατότητα ικανοποίησης κριτηρίων από τους κανόνες. Οι κανόνες που προκύπτουν από τη διαδικασία πρέπει να καλύπτουν τα κριτήρια της υποστήριξης και της εμπιστοσύνης, όπως και στην απλή διαδικασία παραγωγής κανόνων συσχέτισης. Σ' αυτήν την εργασία έγινε μια προσαρμογή του

υπολογισμού των κριτηρίων, ώστε να είναι εφικτή η εφαρμογή τους στο είδος των δεδομένων του ΣΙ.

7.2 Ανοικτά Θέματα

Στα πλαίσια της εργασίας παρουσιάστηκαν κάποια ανοικτά θέματα. Αρχικά, η μέθοδος που παρουσιάστηκε δημιουργεί το σύνολο όλων των κανόνων που ικανοποιούν τα κριτήρια, που έχουν οριστεί από το χρήστη. Επειδή όμως η διαδικασία ανακάλυψης γνώσης είναι μια χρονοβόρα διαδικασία, για να είναι εφικτή η εφαρμογή της μεθόδου σε πραγματικές εφαρμογές, είναι απαραίτητη η απόκριση του συστήματος σε επιθυμητούς χρόνους. Για να επιτευχθεί αυτό μπορεί να οριστεί ένας μέγιστος χρόνος εκτέλεσης της διαδικασίας και με το πέρας αυτού του χρονικού διαστήματος να εμφανίζεται το υποσύνολο των κανόνων, που έχουν προκύψει ως εκείνη τη στιγμή.

Επιπλέον, ο αλγόριθμος εύρεσης κανόνων απαιτεί τη «φόρτωση» ολόκληρης της οντολογίας στη μνήμη για την επεξεργασία της και την εξαγωγή των κανόνων. Αυτό έχει ως αποτέλεσμα η ολοκλήρωση της διαδικασίας για πολύ μεγάλες οντολογίες να μην είναι πάντα εφικτή. Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί με την τμηματοποιημένη μεταφορά της οντολογίας στη μνήμη και την κατάλληλη επεξεργασία της.

Τέλος, εξαιτίας της μη απόλυτης συμβατότητας του συνδυασμού της βιβλιοθήκης OWL – API και της μηχανής συμπερασμού Pellet δεν ήταν πάντα εφικτή η εύρεση των κανόνων μετά το στάδιο του συμπερασμού. Αυτό μπορεί να διορθωθεί με χρήση διαφορετικού συνδυασμού βιβλιοθήκης ή/και μηχανής συμπερασμού.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος Όρος	**Ελληνικός Όρος**
World Wide Web	Παγκόσμιος Ιστός
Semantic Web	Σημασιολογικός Ιστός
Knowledge Discovery	Ανακάλυψη Γνώσης
Ontology	Οντολογία
Inductive Logic Programming	Επαγωγικός Λογικός Προγραμματισμός
Description Logics	Περιγραφικές Λογικές
Property	Ιδιότητα, συσχέτιση
Individual	Στιγμιότυπο
Reasoning	Διαδικασία συμπερασμού
Concept	Έννοια
Role	Ρόλος
Assertion	Ισχυρισμός
Class	Κλάση
Axiom	Αξίωμα
Knowledge – base system	Σύστημα βάσης γνώσης
Satisfiability	Ικανοποιησιμότητα
Consistency	Συνέπεια
Propositional Data Mining	Προτασιακή Ανακάλυψη Γνώσης
Supervised Learning	Μάθηση με επίβλεψη
Unsupervised Learning	Μάθηση χωρίς επίβλεψη
Cluster	Ομάδα
Association Rule	Κανόνας Συσχέτισης
Confidence	Εμπιστοσύνη
Support	Υποστήριξη
Sequence of events	Ακολουθία γεγονότων
Relational Data Mining	Σχεσιακή Ανακάλυψη Γνώσης
Statistical Relational Learning	Στατιστική Ανακάλυψη Γνώσης
Relational Graphical Models	Σχεσιακά Γραφικά Μοντέλα
Probalistic Graphical Models	Πιθανοτικά Γραφικά Μοντέλα
Web Mining	Ανακάλυψη Γνώσης στον Παγκόσμιο Ιστό
Feature – vector representation	Μονοδιάστατη αναπαράσταση χαρακτηριστικών
Feature – based Statistical Learning	Στατιστική Μάθηση βασισμένη σε χαρακτηριστικά
Document Classification	Κατηγοριοποίηση εγγράφου
Text Annotation	Ανάθεση οντολογικών εννοιών σε κείμενο
Generalization	Γενίκευση
Specialization	Εξειδίκευση

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

WWW	World Wide Web
SW	Semantic Web
ILP	Inductive Logic Programming
OWL	Web Ontology Language
SWRL	Semantic Web Rule Language
DL	Description Logics
TBox	Terminological Box
ABox	Assertional Box
W3C	World Wide Web Consortium
RDF	Resource Description Framework
RuleML	Rule Markup Language
KB	Knowledge Base
SVM	Support Vector Machines
SRL	Statistical Relational Learning
RGM	Relational Graphical Models
PRM	Probalistic Relational Models
RBC	Relational Bayes Classifier
RPT	Relational Probability Tree
MSC	Most Specific Concepts
Rlgg	Relative Least General Generalization

ΑΝΑΦΟΡΕΣ

[1] G. Antoniou and F. Harmelen. *The Semantic Primer*, MIT Press, 2008, p. 287.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5), 2001, pp. 1 – 12.

[3] M. R. Koivunen and E. Miller. W3C Semantic Web Activity. World Wide Web Consortium (W3C), 2001; http://www.w3.org/2001/12/semweb-fin/w3csw

[4] U. Fayyad, G. Paitetsky – Shapiro and P. Smyth. Knowledge Discovery and Data Mining: Towards a Unifying framework. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, 1996.

[5] K. Anyanwu, and A. Sheth, P-Queries: Enabling Querying for Semantic Associations on the Semantic Web, In *Proceedings of WWW '03*, 2003, pp. 690-699.

[6] M. Obitko. Ontologies – Introduction to ontologies and Semantic Web, 2007; http://www.obitko.com/tutorials/ontologies-semantic-web/introduction.html

[7] A. Gerber, A. Merwe and A. Barnard. Towards a Semantic Web Layered Architecture. In *Proceedings of IASTED International Conference on Software Engineering, (SE2007)*, Austria 2007, pp. 353–362.

[8] A. Gerber, A. Merwe and A. Barnard. A Functional Semantic Web Architecture. In *Proceedings of European Semantic Web Conference (ESWC 2008)*, Spain 2008, pp. 273 – 287.

[9] M. Kifer, J. de Bruijn, H. Boley and D. Fensel. A Realistic Architecture for the Semantic Web. In *Proc. RuleML 2005*, Ireland 2005, pp. 17–29.

[10] T. Berners-Lee. The Semantic Web and Challenges. World Wide Web Consortium (W3C), 2003; http://www.w3.org/2003/Talks/01-sweb-tbl/Overview.html.

[11] I. Horrocks, B. Parsia, P. F. Patel-Schneider and J. Hendler (2005). Semantic web architecture: Stack or two towers? In *Proc. PPSWR 2005*, Germany 2005, pp. 37–41.

[12] F. Baader, D. L. McGuiness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge: Cambridge University Press, 2003, p. 510.

[13] M. Dean, G. Schreiber, S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference. World Wide Web Consortium (W3C), 2004; http://www.w3.org/TR/owl-ref/.

[14] F. Baader, I. Horrocks, and U. Sattler. Description Logics. Handbook of Knowledge Representation. Elsavier, 2008, pp. 135 - 179.

[15] I. Horrocks, P. F. Patel-Schneider, H. Boley. SWRL: A semantic web rule language combining OWL and RuleML. World Wide Web Consortium (W3C), 2004; http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.

[16] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. In *International Semantic Web Conference*, 2004, pp. 549–563.

[17] Boris Motik and Riccardo Rosati. A faithful integration of description logics with logic programming. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (IJCAI-07), AAAI Press, 2007, pp. 477–482.

[18] H. Boley. The Rule Markup Initiative. 2011; http://ruleml.org/

[19] T. Tudorache, J. Vendetti, and N. F. Noy. Web – Protégé: A Lightweight OWL Ontology Editor for the Web. In *Proceedings of Fifth International OWL: Experiences and Directions Workshop* (OWLED 2008), 2008.

[20] A. Kalyapur, B. Parsia, E. Sirin, B. C. Grau, and J. Hendler. Swoop: A 'Web' Ontology Editing Browser. In *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4(2), 2006, pp. 144 – 153.

[21] Y. Sure, M. Erdmann, J. Angele, S. Staah, R. Studer, and D. Wenke. OntoEdit: Collaborative Ontology Development for the Semantic Web. In *Proceedings of First International Semantic Web Conference* (ISWC 2002), Springer – Verlag Berlin Heidelberg, 2002, pp. 221 – 235.

[22] S. Auer. Powl – A Web Based Platform for Collaborative Semantic web Development. In *Proceedings of First Workshop on Scripting for the Semantic Web* (SFSW '05), 2005.

[23] S. Bechofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: a Reason – able Ontology Editor for the Semantic Web. In *Proceedings of the Ninth Austrian Conference on Artificial Intelligence* (KI 2001), vol. 2174, 2001, pp. 396 – 405.

[24] Jess, The Rule Engine For the Java Platform, 2008; http://www.jessrules.com/jess/index.shtml.

[25] I. Horrocks. Using an expressive Description Logic: FaCT or fiction? In: *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning* (KR'98), Italy, 1998, pp. 636–647.

[26] E. Prud'hommeaux, and A. Seaborne. SPARQL Query Language for RDF, 2005; http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/.

[27] RacerPro, 2008; http://www.racersystems. com/products/racerpro/index.phtml.
[28] D. Tsarkov, and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proceedings of the International Joint Conference on Automated Reasoning* (IJCAR 2006), 2006.
[29] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner, *Journal of Web Semantics*, 5(2), 2007.
[30] B. McBride. Jena: A Semantic Web Toolkit, *IEEE Internet Computing*, vol. 6(6), 2002, pp. 55-59.
[31] R. Shearer, B. Motik, and I. Horrocks. HermiT: A Highly – Efficient OWL Reasoner. In *Proceedings of Fifth International OWL: Experiences and Directions Workshop* (OWLED 2008), 2008.
[32] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Focoue, and C. Lutz. OWL 2 Web Ontology Language. World Wide Web Consortium (W3C), 2009; http://www.w3.org/TR/owl2-profiles/.
[33] F. Baader, S. Brandt, and C. Lutz. Pushing the EL Envelope Further. In *Proceedings of Washington DC Workshop on OWL: Experiences and Directions* (OWLEL08DC), 2008.
[34] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Datacentric Systems and Applications. Springer, 2007.
[35] J. Neville, D. Jensen, and B. Gallagher. Simple Estimators for Relational Bayesian Classifiers. *ICDM IEEE Computer Society*, 2003, pp. 609 – 612.
[36] S. Dzeroski. Multi-Relational Data Mining: An Introduction. *SIGKDD Explorations*, 5(1), 2003, pp. 1–16.
[37] J. Neville, M. Rattigan, and D. Jensen. Statistical Relational Learning: Four Claims and a Survey. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data, Eighteenth International Joint Conference on Artificial Intelligence*, 2003.
[38] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning Probabilistic Relational Models. In *IJCAI*, 1999, pp. 1300–1309.
[39] T. Mitchell. *Machine Learning*, McGraw Hill, 1997, p. 419.
[40] D. Hand, H. Mannila, P. Smyth. *Principles of Data Mining*, MIT Press, 2001, p. 322.
[41] J. M. Adamo. *Data Mining and Association Rules for Sequential Patterns: Sequential and Parallel Algorithms*. Springer, New York, 2001.
[42] J. Roddick, M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. of Knowledge and Data Engineering*, 14(4), 2002, pp. 750 - 767.
[43] M. Zaki, N. Lesh, M. Ogihara, M. Mining features for sequence classification. *In KDD '99: Proceedings of 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 342–346.
[44] B. Grosof, I. Horrocks, R. Volz, and S. Decker. Description Logic Programs: Combining Logic Programs with Description Logics. In *Proc. of WWW-2003*, Budapest, 2003.
[45] T. Scheffer and S. Wrobel. A sequential sampling algorithm for a general class of utility criteria. In *Knowledge Discovery and Data Mining*, 2000, pp. 330–334.
[46] V. Tresp, M. Bundschus, A. Rettinger, and Y. Huang. *Towards Machine Learning on the Semantic Web*. Springer, 2008.
[47] C. Rouveirol, and V. Ventos. Towards learning in CARIN-ALN. In: *International Workshop on Inductive Logic Programming*, 2000
[48] F. A. Lisi. Principles of inductive reasoning on the semantic web: A framework for learning in AL-Log. Principles and Practice of Semantic Web Reasoning, 2005.
[49] S. Bloehdorn, and A. Hotho. Boosting for text classification with semantic features. In *KDD '04: Proceedings. of the Mining for and from the Semantic Web Workshop*, 2004.
[50] S. Handschuh, and S. Staab. Authoring and annotation of web page in CREAM. In *Proceedings of WWW Conference 2002*, 2002.
[51] A. Hotho, S. Staab, and G. Stumme. Explaining text clustering results using semantic structures. In *Proceedings of ECML/PKDD*, 2003, pp. 217–228.
[52] E. Hovy. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proceedings 1st International Conference on Language Resources and Evaluation (LREC), Granada*, 1998.
[53] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology matching: A machine learning approach. In: *Handbook on Ontologies. Springer, Berlin*, 2004, pp. 385–404.
[54] B. Berendt, A. Holto, D. Mladenic, M. Someren, M. Spiliopoulou, and G. Stumme. A RoadMap for Web Mining: From Web to Semantic Web. In *European Web Mining Forum (EWMF)*, 2003, pp. 1 – 22.
[55] W. W. Cohen, and H. Hirsh. Learning the Classic Description Logic: Theoretical and experimental results. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, 1994, pp. 121–133.
[56] L. Iannone, and I. Palmisano. *An algorithm based on counterfactuals for concept learning in the Semantic Web*. Springer, 2005, pp. 370–379.
[57] L. Iannone, I. Palmisano, and Nicola Fanizzi. An algorithm based on counterfactuals for concept learning in the Semantic Web. *Applied Intelligence*, 26(2), 2007, pp. 139–159.

[58] N. Fanizzi, C. d'Amato, and F. Esposito. DL-FOIL concept learning in description logics. In *Proceedings of the 18th International Conference on Inductive Logic Programming*, vol. 5194, 2008, pp. 107 – 121.
[59] J. Lehmann. DL-Learner: Learning Concepts in Description Logics, *Journal of Machine Learning*, 2009, pp. 2639-2642.
[60] Y. Kavurucu, P. Senkul, and I. H. Toroslu. Confidence – based Concept Discovery in Multi – Relational Data Mining, In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (IMECS 2008), vol. 1, 2008.
[61] R. Agrawal, and R. Srikant. Fast Algorithms for Mining Association Rules, In *Proceedings of 20th International Conference on Very Large Data Bases* (VLDB '94), Chile, 1994, pp. 487 – 499.
[62] J. Lehmann, *Learning OWL Class Expression*, Dept. Mathematik und Informatik, Univ. Leipzig, 2010.
[63] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K – means Clustering with Background Knowledge. In *Proceedings of the 18th International Conference on Machine Learning*, 2001, pp. 577 – 584.
[64] C. Haruechaiyasak, M. – L. Shyu, S. – C. Chen, and X. Li. Web Document Classification Based on Fuzzy Association. In *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment* (COMPSAC '02), IEEE Computer Society Washington, USA, 2002.
[65] A. Hotho, S. Staab, and G. Stumme. Explaining text clustering results using semantic structures. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases* (PKDD '03), 2003, pp. 217 – 228.
[66] Family.xml, http://swrl.stanford.edu/ontologies/examples/family.swrl.owl
[67] University.xml, http://owl.man.ac.uk/2005/07/sssw/university1.owl
[68] J.R. Quilan, and R.M. Cameron-Jones. FOIL: A midterm report. In *Proceedings of 6th European Conference on Machine Learning*, vol. 667, Springer – Verlag, 1993, pp. 3 - 20.
[69] S. Muggleton, and C. Feng. Efficient induction in logic programs. Inductive Logic Programming, Academic Press, 1992, pp. 281 – 298.
[70] S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4), 1995, pp. 245 – 286.
[71] A. Srinivasan, The Aleph Manual, 1999; http://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html