Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΣΗΜΑΣΙΟΛΟΓΙΚΗΣ ΣΥΣΧΕΤΙΣΗΣ ΣΧΗΜΑΤΩΝ

ΜΑΡΙΑ ΚΑΛΑΪΤΖΑΚΗ

Μεταπτυχιακή Εργασία

Ηράκλειο, Σεπτέμβριος 2009

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΣΗΜΑΣΙΟΛΟΓΙΚΗΣ ΣΥΣΧΕΤΙΣΗΣ ΣΧΗΜΑΤΩΝ

ΜΑΡΙΑ ΚΑΛΑΪΤΖΑΚΗ

Εργασία που υποβλήθηκε ως μερική εκπλήρωση
των απαιτήσεων για την απόκτηση

Μεταπτυχιακού Διπλώματος Ειδίκευσης

Συγγραφέας:

_____

Μαρία Καλαϊτζάκη, Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

_____

Δημήτρης Πλεξουσάκης, Καθηγητής, Επόπτης

_____

Γρηγόρης Αντωνίου, Καθηγητής, Μέλος

_____

Ιωάννης Τσαμαρδίνος, Επίκουρος Καθηγητής, Μέλος

Δεκτή:

_____

Πάνος Τραχανιάς, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Σεπτέμβριος 2009

Περίληψη

Η συσχέτιση σχημάτων εφαρμόζεται σε ένα ευρύ πεδίο εφαρμογών και αποτελεί θεμέλιο λίθο της ενοποίησης της πληροφορίας (data integration). Λέγοντας συσχέτιση σχημάτων εννοείται η λειτουργία της επεξεργασίας δύο ή περισσοτέρων σχημάτων πληροφορίας και ο εντοπισμός των σημείων στα οποία σχετίζονται σημασιολογικά τα σχήματα αυτά. Η πληθώρα των εφαρμογών που παρουσιάζει το συγκεκριμένο ερευνητικό θέμα αποτέλεσε το εφαλτήριο της εργασίας που περιγράφεται στη συνέχεια. Παρόλο που το πρόβλημα της συσχέτισης σχημάτων μετράει χρόνια έρευνας και έχουν γίνει πολλές αξιόλογες προσπάθειες στα πλαίσια αντιμετώπισης του, εξακολουθεί να είναι αδήριτη η ανάγκη της δημιουργίας ενός συστήματος που μπορεί να ενσωματώσει κάθε αλγόριθμο συσχέτισης και, ταυτόχρονα, είναι φιλικό προς το χρήστη.

Ο πυρήνας της προσέγγισης αυτής, περιγράφεται από το συνδυασμό της λεξικογραφικής, σημασιολογικής και συντακτικής πληροφορίας των δύο, υπό συσχέτιση, σχημάτων. Ο προτεινόμενος αλγόριθμος, αξιοποιεί τα αποτελέσματα της λεξικογραφικής ομοιότητας που προκύπτει από το εξωτερικό λεξικό WordNet και εξάγει την ομοιότητα με βάση την περιγραφή των στοιχείων. Στη συνέχεια, υπολογίζεται η συντακτική ομοιότητα με γνώμονα τις σχέσεις κληρονομικότητας. Τέλος, τα παραπάνω αποτελέσματα, συνυπολογίζονται για την εύρεση της συνολικής ομοιότητας των στοιχείων.

Απόρροια της παρούσας ερευνητικής εργασίας, είναι ένα ολοκληρωμένο σύστημα συσχέτισης σχημάτων, με δομικά στοιχεία τον υβριδικό αλγόριθμο σημασιολογικής συσχέτισης και τη διαδικτυακή διεπαφή συσχέτισης. Η γενικότητα της προτεινόμενης λύσης βασίζεται τόσο στο εύρος των σχημάτων που δύναται να συσχετίσει (σχήματα σχεσιακών βάσεων, οντολογίες, XML σχήματα), όσο και στην ποικιλία των παραγόντων που λαμβάνει υπόψη της. Η σημαντική συμβολή της πρότασης αυτής καταδεικνύεται από τα σημαντικά αποτελέσματα της αξιολόγησης του αλγορίθμου.

Επόπτης: Δημήτρης Πλεξουσάκης, Καθηγητής
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Abstract

The impact of schema matching in the domain of data integration is considered to be a fundamental importance, due to the extensive variety of its applications. Schema matching is the process of developing semantic matches between two or more schemas, and the detection of semantic correspondences between them. Despite the fact that the problem of schema matching is recurring and many solutions have been proposed, there is still the need of a unified system of a schema matching algorithm and a human-centric user interface.

The core of this approach is described by the combination of the schemas' lexical, semantic and syntactic information. The proposed solution utilizes the results of lexical matching, through the external dictionary WordNet, and the attributes of each concept, in order to produce the semantic similarity. Afterwards, the syntactic similarity is computed based on the relations of inheritance. Finally, these three factors decide the overall similarity between the concepts of the two schemas.

The principal outcome of this work is an integrated system based on a hybrid semantic schema matching algorithm and a user-friendly web interface. The universality of this approach depends not only on the variety of the different types of schemas (relational schemas, ontologies and XML schemas) but also on the many factors that are considered in order to produce the similarity between two concepts. The experiments that have been carried out demonstrate the effectiveness and importance of the results.

Supervisor: Dimitris Plexousakis, Professor
Computer Science Department
University of Crete

Στους γονείς μου Γιώργο και Βαγγελιώ

Στον αδερφό μου Γιάννη

Ευχαριστίες

Απόρροια της ολοκλήρωσης ενός μεγάλου στόχου είναι η συνειδητοποίηση της συνολικής πορείας του ανθρώπου και των επιρροών που δέχτηκε από τον κόσμο γύρω του. Κατ' αρχάς, οφείλω να ευχαριστήσω τον επόπτη μου κ. Δ. Πλεξουσάκη, για την ευκαιρία που μου έδωσε να συμμετάσχω στο Μεταπτυχιακό Πρόγραμμα του Τμήματος Επιστήμης Υπολογιστών. Ήταν τιμή και χαρά η συνεργασία αυτή, γιατί όχι μόνο αποκόμισα γνώση αλλά και η καθοδήγηση του ήταν βαρύνουσας σημασίας για την πορεία της μεταπτυχιακής εργασίας.

Ιδιαίτερη μνεία πρέπει να γίνει στον ερευνητή του Ινστιτούτου Πληροφορικής (Ι.Π. - Ι.Τ.Ε.) κ. Martin Doerr εξαιτίας της καθοριστικής συμβολής του στην εξέλιξη της εργασίας.

Η αμέριστη υποστήριξη των φίλων μου σε όλα τα επίπεδα ήταν καθοριστική για την διεκπεραίωση της εργασίας αυτής. Στάθηκα τυχερή, γιατί στην πορεία των σπουδών μου γνώρισα την Πόπη Μπουράκη, τον Γιάννη Αυγουλέα, τη Φλώρα Γιαμάκη, τη Γεωργία Σερδενέ και τη Μαρία Κουλεντάκη. Στο σημείο αυτό οφείλω να ευχαριστήσω τη Μαρία Μίχου, που με συμβούλεψε, με εμψύχωσε και με βοήθησε έμπρακτα. Η συμβολή της ήταν, πράγματι, καταλυτική. Επίσης, ευχαριστώ πολύ τον Γιώργο Μπαργιάννη, τον Νίκο Παππά και τον Γιάννη Ανδρουλάκη για τη συμπαράσταση, τις συμβουλές και την απεριόριστη διάθεση τους να βοηθήσουν. Στους φίλους μου που υπήρξαν δίπλα μου σε ιδιαίτερα κρίσιμες στιγμές και όταν προέκυπταν εμπόδια που έδειχναν να είναι ανυπέρβλητα, συμπεριλαμβάνεται η Λίτσα Κουζούλογλου και ο Γιάννης Παναγιωτάκης.

Το μεγαλύτερο όμως ευχαριστώ οφείλω στους γονείς μου, Γιώργο και Βαγγελιώ, και στον αδερφό μου, Γιάννη. Η συμπαράσταση, η κατανόηση και η απύθμενη υπομονή που έδειξαν υπήρξαν ο καθοριστικός παράγοντας που συντέλεσε στο να ξεπεράσω τις δύσκολες στιγμές που αντιμετώπισα κατά τη διάρκεια των σπουδών. Θα ήθελα να ευχαριστήσω τους γονείς μου για όσα μου έχουν προσφέρει όλα αυτά τα χρόνια γιατί σε αυτούς χρωστάω τα πάντα. Σίγουρα με διαφορετικούς γονείς δε θα είχα καταφέρει τίποτα από αυτά. Εξίσου σημαντική είναι η αρωγή του αδερφού μου, που μέσω της ψυχολογικής του στήριξης αλλά και της ωριμότητας που τον διακατέχει, μπόρεσα να αντιμετωπίσω πολλές δυσκολίες με το βέλτιστο τρόπο.

Όμως η συνεχής στήριξη της οικογένειας, των φίλων και των καθηγητών μου, καθώς και η δύναμη που αντλώ από αυτούς, δεν μπορούν να περιγραφούν σε λίγες γραμμές, στην αναζήτηση για τη δική μου Ιθάκη...

"Κι αν πτωχική τη βρεις, η Ιθάκη δε σε γέλασε.
Έτσι σοφός που έγινες, με τόση πείρα,
ήδη θα το κατάλαβες οι Ιθάκες τι σημαίνουν."
Κ.Π.Καβάφης

Μαρία Καλαϊτζάκη, Σεπτέμβριος 2009

Περιεχόμενα

Λίστα Εικόνων

Λίστα Πινάκων

Κεφάλαιο 1. Εισαγωγή

Επιβάλλεται πριν την ανάλυση του αλγορίθμου σημασιολογικής συσχέτισης, καθώς και του συστήματος που τον υλοποιεί, να προηγηθεί μια περιγραφή της έννοιας της «συσχέτισης σχημάτων» (schema matching). Λέγοντας συσχέτιση σχημάτων, εννοείται η λειτουργία της επεξεργασίας δύο σχημάτων πληροφορίας και ο εντοπισμός των σημείων στα οποία σχετίζονται σημασιολογικά τα σχήματα αυτά.

Αρχικά, αξίζει να γίνει μια αναφορά στα κίνητρα που ώθησαν στο σχεδιασμό και την υλοποίηση ενός νέου σημασιολογικού αλγορίθμου συσχέτισης σχημάτων. Στη συνέχεια, γίνεται λόγος για τη συνεισφορά της πρότασης αυτής τόσο στον ερευνητικό τομέα της συσχέτισης σχημάτων όσο και σε εφαρμογές που βασίζονται σε αυτή. Τέλος, περιγράφεται επιγραμματικά ο βασικός άξονας στον οποίο στηρίζεται η ανάπτυξη της εργασίας.

1.1 Κίνητρο

Η συσχέτιση σχημάτων εφαρμόζεται σε ένα ευρύ πεδίο εφαρμογών. Πρόκειται για εφαρμογές που χαρακτηρίζονται από τη χρήση δομημένων δεδομένων, όπως οντολογίες και XML, και η χρήση ενός αλγορίθμου συσχέτισης προαπαιτείται για την εκκίνηση του εκάστοτε συστήματος. Με άλλα λόγια, η πλειοψηφία των συστημάτων διαχειρίζεται πληροφορία που προέρχεται από ετερογενείς βάσεις δεδομένων και αυτό έχει ως αποτέλεσμα την ανάγκη συσχέτισης ή ακόμα και ενοποίησης τους. Για παράδειγμα, μερικές από τις εφαρμογές αυτές είναι ο σημασιολογικός ιστός, οι αποθήκες δεδομένων, το ηλεκτρονικό επιχειρείν και οι ηλεκτρονικές υπηρεσίες. Ειδικότερα, η συσχέτιση σχημάτων αποτελεί θεμέλιο λίθο της ενοποίησης της πληροφορίας (data integration). Πρόκειται για λειτουργία που έχει τόσο επιστημονικές όσο και εμπορικές εφαρμογές, ενώ η εφαρμογή της κρίνεται επιτακτική, όλο και περισσότερο, καθώς ο όγκος της πληροφορίας αυξάνεται δραματικά. Στο χώρο της τεχνητής νοημοσύνης, αυτό είναι το πρόβλημα της ενοποίησης ανεξάρτητα ανεπτυγμένων οντολογιών [1], με σκοπό την παραγωγή μιας ενοποιημένης οντολογίας. Η πληθώρα των εφαρμογών που

παρουσιάζει το συγκεκριμένο ερευνητικό θέμα αποτέλεσε το εφαλτήριο της εργασίας που περιγράφεται στη συνέχεια.

Παρόλο, που το πρόβλημα της συσχέτισης σχημάτων μετράει χρόνια έρευνας, και έχουν γίνει πολλές αξιόλογες προσπάθειες στα πλαίσια αντιμετώπισης του, εξακολουθεί να είναι αδήριτη η ανάγκη της δημιουργίας ενός συστήματος που μπορεί να ενσωματώσει κάθε αλγόριθμο συσχέτισης και, ταυτόχρονα, είναι φιλικό προς το χρήστη. Τόσο τα εργαλεία όσο και οι αλγόριθμοι συσχέτισης που έχουν προταθεί μέχρι τώρα δεν καλύπτουν πλήρως το πρόβλημα της ενοποίησης των σχημάτων, αφού ακόμη και περιπτώσεις ισοδύναμων εννοιών δεν εντοπίζονται λόγω δομικών ή σημασιολογικών διαφορών. Σύμφωνα με τις απαιτήσεις των χρηστών συστημάτων συσχέτισης, είναι σημαντικό για κάθε σύστημα συσχέτισης να είναι επεκτάσιμο (modular), δηλαδή να μπορεί να συμπεριλάβει νέους αλγορίθμους συσχέτισης και να είναι διαθέσιμο στον Ιστό. Επιπροσθέτως, ένα σύστημα συσχέτισης σχημάτων πρέπει να επιτρέπει στο χρήστη του να προτείνει χειροκίνητα τις συσχετίσεις και να τις καταχωρεί. Εκτός από τα παραπάνω, θεμιτό είναι το σύστημα να είναι ανθρωποκεντρικό και να επιτρέπει στο χρήστη να το χειριστεί ανεξάρτητα από τον ερευνητικό τομέα από τον οποίο προέρχεται. Εδώ αξίζει να σημειωθεί ότι αρκετά από τα συστήματα που έχουν προταθεί ως τώρα, παρουσιάζουν έλλειψη γραφικής αναπαράστασης των σχημάτων ή είναι δύσχρηστα λόγω της χαμηλής διαλειτουργικότητας τους.

Εν κατακλείδι, όλα τα παραπάνω συνετέλεσαν στην πρόταση ενός νέου αλγορίθμου σημασιολογικής συσχέτισης σχημάτων με σκοπό την επίλυση των βασικότερων προβλημάτων που αντιμετωπίζει σήμερα το θέμα αυτό. Τέλος, υλοποιήθηκε και ένα ευέλικτο σύστημα για την αναπαράσταση και συσχέτιση σχημάτων που λειτουργεί επικουρικά με τον αλγόριθμο που προτείνεται.

1.2 Συμβολή

Η παρούσα εργασία καλείται να καλύψει την πλειοψηφία των αναγκών του χρήστη που έχει ως στόχο τη συσχέτιση σχημάτων. Παρακάτω, περιγράφεται η συμβολή του σημασιολογικού αλγορίθμου συσχέτισης, καθώς επίσης και του συστήματος που τον υποστηρίζει.

Πρώτα απ' όλα, ο σημασιολογικός αλγόριθμος παρέχει πιο ακριβή αποτελέσματα σε σχέση με ανταγωνιστικούς του αλγορίθμους. Η ακρίβεια των αποτελεσμάτων

έγκειται στο συνδυασμό των τεχνικών που χρησιμοποιήθηκαν τόσο στο σχεδιασμό όσο και στην υλοποίηση του. Πρόκειται για υβριδικό αλγόριθμο που αξιοποιεί την πληροφορία των, υπό συσχέτιση, σχημάτων στο έπακρο. Επιπροσθέτως, βασικό χαρακτηριστικό του σημασιολογικού αλγορίθμου συσχέτισης είναι η επαναχρησιμοποίηση συσχετίσεων. Αυτό καθιστά τα αποτελέσματα περισσότερο αξιόπιστα και πιο κοντά στην πραγματικότητα. Μια επιπλέον ιδιότητα του αλγορίθμου είναι ότι μπορεί να χειριστεί και να συσχετίσει διαφορετικούς τύπους σχημάτων, δηλαδή σχήματα σχεσιακών βάσεων, οντολογίες (owl και rdf) και XML σχήματα (xsd και xdr).

Επιπλέον, η διεπαφή του συστήματος χαρακτηρίζεται από διαλειτουργικότητα, γεγονός που έχει ως αποτέλεσμα την εύκολη και γρήγορη εκμάθηση του από το χρήστη, καθώς επίσης και τη μείωση του συνολικού χρόνου, που απαιτείται για την διεκπεραίωση της χειροκίνητης συσχέτισης. Σε αυτό το σημείο, αξίζει να σημειωθεί ότι το σύστημα δεν απευθύνεται αποκλειστικά σε ειδικούς της Πληροφορικής ή κάποιου άλλου ερευνητικού τομέα. Μέσω της διεπαφής, ο χρήστης του συστήματος δύναται να πλοηγηθεί στις γραφικές αναπαραστάσεις των σχημάτων, να επιλέξει την πυροδότηση ενός αλγορίθμου συσχέτισης ή να προτείνει συσχετίσεις όρων, χειροκίνητα. Εκτός από τα παραπάνω, η επεκτασιμότητα του συστήματος το καθιστά ευέλικτο, αφού μπορεί εύκολα να ενσωματώσει εναλλακτικούς αλγορίθμους συσχέτισης.

Τέλος, η εφαρμογή του συστήματος δεν περιορίζεται σε κάποιο συγκεκριμένο ερευνητικό τομέα, αφού μπορεί να αξιοποιηθεί για κάθε είδος σχήματος και να δώσει αποδεκτά αποτελέσματα για διαφορετικά πεδία εφαρμογών (πχ. Βιοπληροφορική, Ενοποίηση οντολογιών).

1.3 Οργάνωση

Όσο αφορά την οργάνωση της εργασίας που ακολουθεί, περιγράφεται παρακάτω.

Το δεύτερο κεφάλαιο, αποτελεί μια σύντομη ανασκόπηση του προβλήματος της ενοποίησης σχημάτων. Επίσης, περιγράφονται μερικές από τις περιοχές εφαρμογών που αντιμετωπίζουν αυτό το πρόβλημα και αναλύονται με παραδείγματα.

Το τρίτο κεφάλαιο, εστιάζει σε μια απαρίθμηση των σημαντικότερων προσεγγίσεων που έχουν προταθεί μέχρι σήμερα, για την επίλυση της συσχέτισης

σχημάτων. Εκτός της αναλυτικής περιγραφής των συστημάτων, γίνεται και μια κατηγοριοποίηση ως προς τα βασικά χαρακτηριστικά που τα διακρίνουν.

Το τέταρτο κεφάλαιο, παρουσιάζει, εκτενώς, το σημασιολογικό αλγόριθμο συσχέτισης, τόσο ως προς τη δομή του, όσο και ως προς τα κριτήρια αξιολόγησης που ικανοποιεί. Το κεφάλαιο ολοκληρώνεται με την αναφορά στα συγκριτικά αποτελέσματα μετρικών αξιολόγησης του αλγορίθμου σε σχέση με άλλες προσεγγίσεις.

Το πέμπτο κεφάλαιο, περιλαμβάνει την περιγραφή του συστήματος που υλοποιήθηκε, καθώς και τον τρόπο με τον οποίο αυτό συνεργάζεται με τον αλγόριθμο σημασιολογικής συσχέτισης. Το κεφάλαιο αυτό περιέχει μια εκτενή περιγραφή της διεπαφής του συστήματος, ενώ για την πληρέστερη κατανόηση της λειτουργίας του παρουσιάζεται ένα πλήρες σενάριο χρήσης του.

Το έκτο κεφάλαιο, παραθέτει τα συμπεράσματα που προέκυψαν κατά τη διάρκεια της έρευνας καθώς επίσης και τις δυνατότητες βελτιστοποίησης του συστήματος.

Στο παράρτημα της εργασίας περιέχονται τα σημαντικότερα τμήματα κώδικα που συναπαρτίζουν το σημασιολογικό αλγόριθμο συσχέτισης καθώς επίσης και τα σχήματα που χρησιμοποιήθηκαν στα πλαίσια της αξιολόγησης του αλγορίθμου με τα αποτελέσματα της αξιολόγησης αναλυτικά.

Κεφάλαιο 2. Γνωστικό Υπόβαθρο

Συσχέτιση σχημάτων, είναι η διαδικασία ανάπτυξης σημασιολογικών συσχετίσεων ή αντιστοιχιών ανάμεσα σε δύο ή περισσότερα σχήματα. Η συσχέτιση σχημάτων είναι το πρώτο και πιο κρίσιμο βήμα της ενοποίησης σχημάτων [2, 3]. Γενικότερος σκοπός της συσχέτισης σχημάτων, αφ' ενός, είναι η συνένωση δύο ή περισσότερων βάσεων δεδομένων και αφ' ετέρου, η δυνατότητα εκτέλεσης επερωτήσεων σε πολλαπλές ετερογενείς βάσεις δεδομένων χρησιμοποιώντας ένα μοναδικό σχήμα [2]. Ενοποίηση σχήματος, ορίζεται η ενοποίηση σχημάτων σε ένα και μοναδικό σχήμα [4]. Το σχήμα που προκύπτει μπορεί να έχει τη μορφή όψης (view) ή το σχήμα μιας ενοποιημένης βάσης.

Το κεφάλαιο αυτό εστιάζει στην έρευνα που έχει διεξαχθεί στο πρώτο στάδιο της ενοποίησης δεδομένων, τη συσχέτιση σχημάτων, που έχει χαρακτηριστεί ως την καρδιά της διαδικασίας ενοποίησης δεδομένων [3] και χρησιμοποιείται συχνά στην ανάπτυξη ενδιάμεσων σχημάτων ανάμεσα σε δύο ή περισσότερες πηγές δεδομένων. Μια πλήρης λύση στο πρόβλημα της ενοποίησης δεδομένων, αρχίζει με τη συσχέτιση σχημάτων και είναι απαραίτητο να λαμβάνει υπ' όψη της την αρχιτεκτονική ενοποίησης και, επιπροσθέτως, την επεξεργασία επερωτήσεων.

Η χειροκίνητη συσχέτιση σχημάτων είναι μια χρονοβόρα και επιρρεπής σε λάθη διαδικασία. Για παράδειγμα, ο χρόνος που απαιτείται για τη χειροκίνητη συσχέτιση 27,000 στοιχείων από 40 βάσεις δεδομένων εκτιμάται ότι είναι περισσότερο από 12 χρόνια [7]. Προς το παρόν, η συσχέτιση σχημάτων εκτελείται χειροκίνητα [3], ή στην καλύτερη των περιπτώσεων, ημι-αυτόματα με κάποιους αλγορίθμους, που προτείνουν πιθανές συσχετίσεις και κάποιος χρήστης καλείται να λάβει την τελική απόφαση και να δεχτεί ή να απορρίψει κάποια πρόταση.

Καμία από τις μεθόδους συσχέτισης σχημάτων που αναλύονται παρακάτω δεν έχει καταφέρει ακόμη να λειτουργεί εντελώς αυτόματα και η συμβολή του ανθρώπου παραμένει απαραίτητη. Στην πραγματικότητα, κάποιοι ερευνητές (για παράδειγμα, [8, 9, 10, 6]), αποκλείουν την πιθανότητα της ύπαρξης αλγορίθμου που συσχετίζει σχήματα αυτόματα και προσανατολίζουν την έρευνα τους στην υποβοηθούμενη από τον άνθρωπο συσχέτιση σχημάτων. Η συσχέτιση σχημάτων

ανήκει στα "AI πλήρη" προβλήματα [11]. Στον τομέα της τεχνητής νοημοσύνης, τα δυσκολότερα προβλήματα είναι γνωστά ως AI-πλήρη (AI complete ή AI-hard), υποδηλώνοντας ότι η δυσκολία των υπολογιστικών αυτών προβλημάτων είναι ισοδύναμη με την επίλυση του βασικού προβλήματος της τεχνητής νοημοσύνης, δηλαδή να γίνουν οι υπολογιστές τόσο έξυπνοι όσο οι άνθρωποι. Ο χαρακτηρισμός ενός προβλήματος ως AI-πλήρες αντικατοπτρίζει ένα πρόβλημα που δεν μπορεί να λυθεί με έναν απλό αλγόριθμο. Τέτοια προβλήματα συνήθως περιλαμβάνουν θέματα όπως υπολογιστική όραση, κατανόηση φυσικής γλώσσας και αντιμετώπιση απροσδόκητων καταστάσεων κατά τη διάρκεια επίλυσης[1].

Στο υπόλοιπο αυτού του κεφαλαίου, αρχικά, περιγράφεται το πρόβλημα της συσχέτισης σχημάτων εκτενέστερα και στη συνέχεια, ακολουθεί ένα υποσύνολο των εφαρμογών του θέματος αυτού.

2.1 Το πρόβλημα της συσχέτισης

Σε ένα κατανεμημένο σύστημα, όπως είναι ο σημασιολογικός ιστός και πολλές άλλες εφαρμογές που παρουσιάζονται παρακάτω, η ετερογένεια δεν μπορεί να αποφευχθεί. Διαφορετικοί χρήστες έχουν διαφορετικά ενδιαφέροντα και συνήθειες, χρησιμοποιούν διαφορετικά εργαλεία και γνώση και συχνά, σε διαφορετικά επίπεδα λεπτομέρειας. Όλα τα προηγούμενα έχουν ως αποτέλεσμα διαφορετικές μορφές ετερογένειας και έτσι θα πρέπει να εξεταστούν εξονυχιστικά.

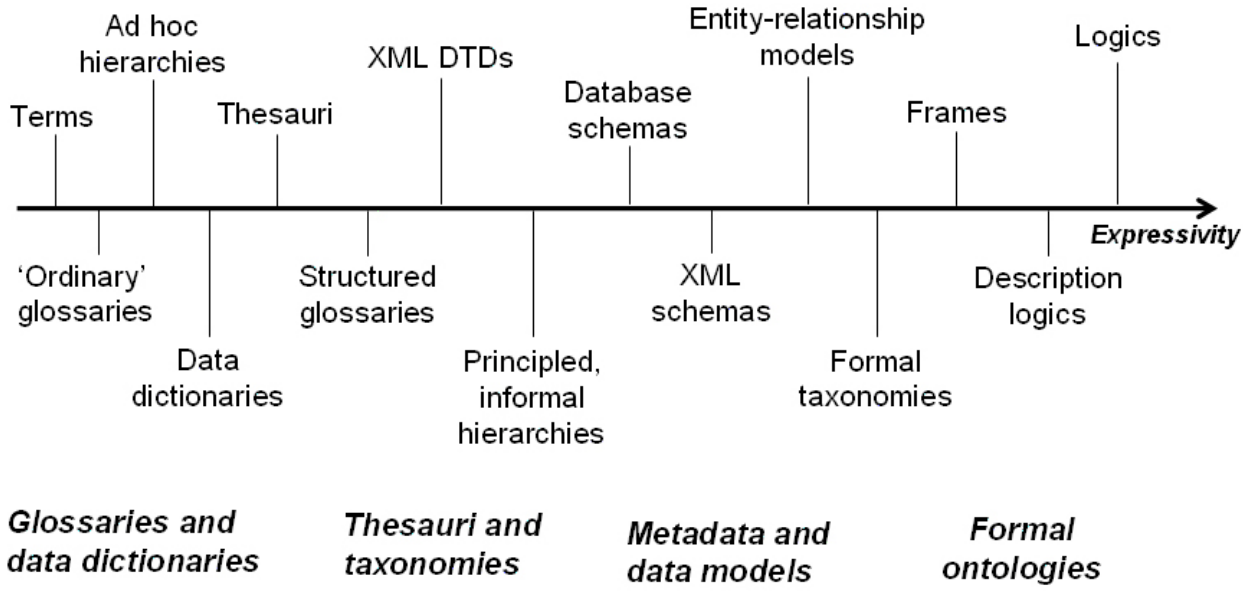
Στο κεφάλαιο αυτό, παρουσιάζονται εναλλακτικοί τρόποι αναπαράστασης της γνώσης που χρησιμοποιούνται σε διάφορες εφαρμογές. Στη συνέχεια, παρουσιάζονται διαφορετικές αιτίες ετερογένειας. Η επίγνωση τους θα μπορούσε να βοηθήσει κατά τη διάρκεια της σχεδίασης μιας στρατηγικής συσχέτισης, με βάση τη μορφή ετερογένειας που πρόκειται να αντιμετωπιστεί. Παρακάτω, γίνεται μια σύντομη περιγραφή της ορολογίας που περιγράφει έννοιες της συσχέτισης. Τέλος, γίνεται μια συνολική περιγραφή του προβλήματος της συσχέτισης.

Σκοπός του κεφαλαίου αυτού δεν είναι να συνοψίσει το θέμα της συσχέτισης αλλά να δώσει κάποιους ορισμούς που θα βοηθήσουν τον αναγνώστη στην κατανόηση των λύσεων που έχουν προταθεί και παρουσιάζονται στο επόμενο κεφάλαιο.

1 http://en.wikipedia.org/wiki/AI-complete

2.1.1 Λεξιλόγια, Σχήματα και Οντολογίες

Μέχρι τώρα έχουμε εξετάσει οντολογίες που δεν είναι ακριβείς. Μια οντολογία μπορεί να θεωρηθεί ως ένα σύνολο από ισχυρισμούς που μοντελοποιούν ένα συγκεκριμένο τομέα. Συνήθως, η οντολογία ορίζει ένα λεξιλόγιο που χρησιμοποιείται από μια συγκεκριμένη εφαρμογή. Σε διάφορες περιοχές της επιστήμης υπολογιστών υπάρχουν διαφορετικά δεδομένα και εννοιολογικά μοντέλα που μπορούν να αντιμετωπιστούν ως οντολογίες. Υπάρχουν, για παράδειγμα, κατηγοριοποιήσεις περιεχομένου από χρήστες με τη χρήση ετικετών (folksonomies), σχήματα βάσεων δεδομένων, UML μοντέλα, ευρετήρια, θησαυροί, XML σχήματα και formal οντολογίες. Αυτά και άλλα παραδείγματα περιγράφονται σε φθίνουσα σειρά στην Εικόνα 2.1. Έτσι, μια οντολογία υποτίθεται ότι έχει σαφώς ορισμένη σημασιολογία, ενώ η προσέγγιση των ευρετηρίων σαν σύστημα αρχείων συνήθως υπονοείται. Στην πραγματικότητα, αυτό εξαρτάται αποκλειστικά από τον ίδιο το δημιουργό, δηλαδή, την έννοια των ετικετών, το γνωστικό υπόβαθρο και το πλαίσιο στο οποίο τοποθετούνται οι ετικέτες, έτσι δεν αποτελούν τμήμα των προδιαγραφών του ευρετηρίου.

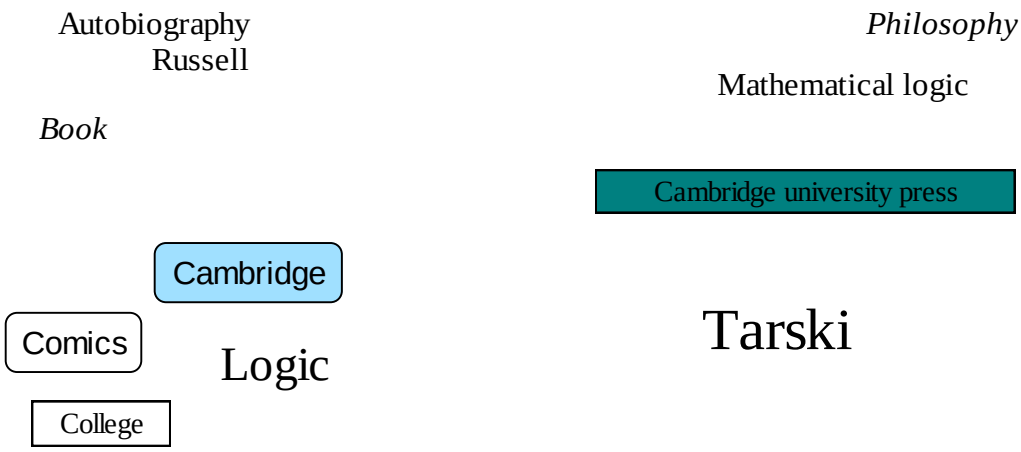


Εικόνα 2.1 Διαφορετικές μορφές σχημάτων ταξινομημένες με βάση την εκφραστικότητα τους (με βάση το [12])

Παρακάτω δίνονται παραδείγματα ποικίλων σχημάτων της εικόνας και σκιαγραφούν κάποια προβλήματα ετερογένειας που μπορεί να δημιουργηθούν.

Ετικέτες και κατηγοριοποιήσεις περιεχομένων από χρήστες με τη χρήση ετικετών

Οι ετικέτες και οι κατηγοριοποιήσεις περιεχομένων μέσω ετικετών αποτελούν απλούς τρόπους περιγραφής ενός συνόλου γνώσης μέσω των ονομάτων. Η προσέγγιση αυτή χρησιμοποιείται σε δημοφιλής ιστοσελίδες, όπως το del.icio.us[2] για το σχολιασμό της ιστοσελίδας ή το Flickr[3] για το σχολιασμό των εικόνων. Ένα παράδειγμα ετικετών για βιβλία και συλλογές βιβλίων δίνεται στην Εικόνα 2.2.



Εικόνα 2.2 Τμήματα από δύο κατηγοριοποιήσεις περιεχομένων από χρήστες μέσω ετικετών

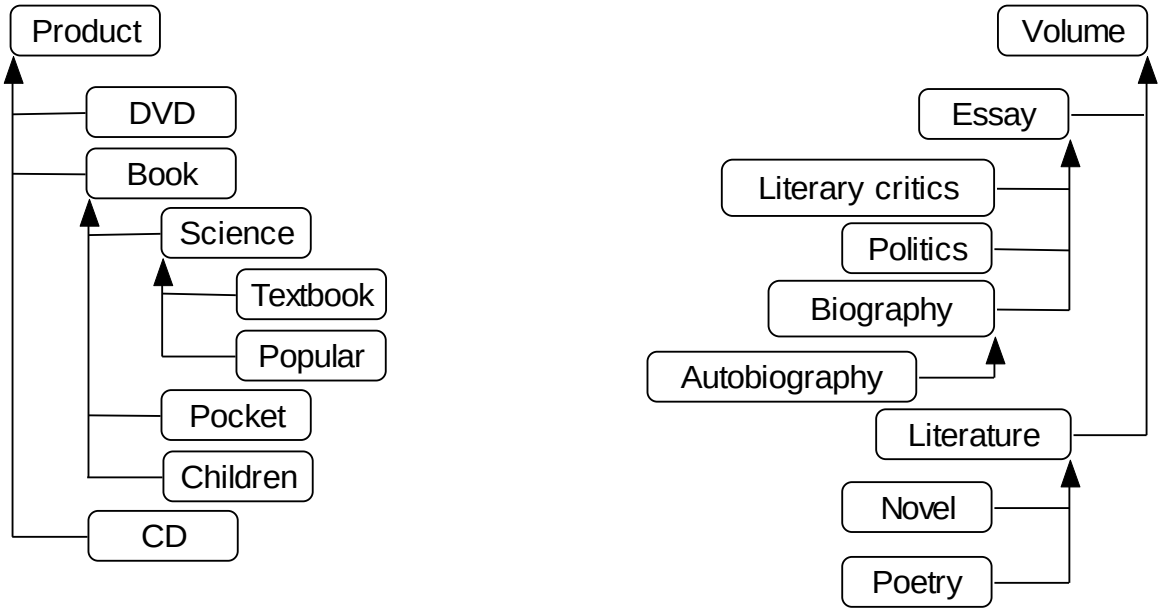
Προφανώς, διαφορετικοί χρήστες χρησιμοποιούν διαφορετικές ετικέτες. Ακόμη και αν αυτές οι ετικέτες παραμένουν εσωτερικά συναφής για το χρήστη που τις δημιούργησε, η εσωτερική δομή τους δεν είναι ακριβής για τον υπολογιστή. Ο εντοπισμός σχέσεων μεταξύ των ετικετών δύο κατηγοριοποιήσεις περιεχομένων από χρήστες μέσω ετικετών αποτελεί δύσκολη υπόθεση. Επιπλέον, το γεγονός ότι οι ετικέτες αυτές δεν έχουν άμεση σχέση μεταξύ τους (σε μια κατηγοριοποίηση περιεχομένου από χρήστες μέσω ετικετών) κάνει το πρόβλημα δυσκολότερο. Όμως, έχει γίνει αρκετή δουλειά για τη δημιουργία μιας δομής μεταξύ των ετικετών, πχ. οι συστάδες του Flickr, που βασίζονται, κυρίως, σε ένα σύνολο αντικειμένων, όπως εικόνες και ιστοσελίδες και ευρετηριοποιούνται με βάση τις αντίστοιχες ετικέτες.

Ευρετήρια

Μια ταξονομία είναι ένα μερικώς ταξινομημένο σύνολο ταξινομικών βαθμίδων (taxons) ή κλάσεων, στο οποίο μια ταξινομική βαθμίδα είναι μεγαλύτερη μιας

2 http://del.icio.us
3 http://www.flickr.com

άλλης όταν αυτό που περιγράφει, περιγράφεται και από την άλλη. Τα ευρετήρια είναι ταξινομίες που χρησιμοποιούνται από εταιρείες για την αναπαράσταση αγαθών προς πώληση, από βιβλιοθήκες για την οργάνωση βιβλίων ή από ιδιώτες για την ταξινόμηση αρχείων στον προσωπικό τους υπολογιστή. Μερικά γνωστά παραδείγματα ευρετηρίων είναι αυτά της Google[4], της Yahoo[5] και το Open Directory Project[6]. Τα ευρετήρια αυτά είναι ιεραρχίες φακέλων που εντοπίζονται μέσω των ετικετών τους και των περιεχομένων τους. Η σημασιολογία των φακέλων αυτών δίνεται από τα αντικείμενα που περιέχουν [13]. Φυσικά, κάθε ανεξάρτητη οντότητα τείνει να αναπτύσσει το δικό της ευρετήριο βασισμένο στις ανάγκες της, όπως στην Εικόνα 2.3.



Εικόνα 2.3 Τμήματα δύο ευρετηρίων

Στην Εικόνα 2.3, το ευρετήριο που βρίσκεται αριστερά αναπαριστά ένα σύνολο από αντικείμενα ενός βιβλιοπωλείου, ενώ το δεύτερο είναι ένα ευρετήριο ενός ατόμου που σκιαγραφεί το περιεχόμενο της βιβλιοθήκης του. Όπως μπορούμε να δούμε, τα ευρετήρια αυτά κωδικοποιούν έναν τομέα σε διαφορετικά επίπεδα λεπτομέρειας, δεδομένου ότι έχουν σχεδιαστεί ανεξάρτητα και για διαφορετικό σκοπό, δηλαδή, πώληση σε αντίθεση με κατηγοριοποίηση.

4 http://www.google.com/dirhp
5 http://www.yahoo.com
6 http://dmoz.org

Τέλος, υπάρχουν μερικές συναινετικές κατηγοριοποιήσεις. Στη βιβλιοθηκονομία, έχει χρησιμοποιηθεί η κατηγοριοποίηση Dewey, για περισσότερο από έναν αιώνα, για την κατηγοριοποίηση βιβλίων με βάση το θέμα [14].

Σχήματα σχεσιακών βάσεων δεδομένων

Οι σχεσιακές βάσεις δεδομένων προϋποθέτουν ότι τα δεδομένα είναι οργανωμένα με έναν προκαθορισμένο τρόπο όπως οι πίνακες ή οι σχέσεις. Ένα σχεσιακό σχήμα καθορίζει τα ονόματα των πινάκων καθώς επίσης και τον τύπο τους, τα ονόματα και τους τύπους του κάθε πεδίου του πίνακα. Επίσης, το σχεσιακό μοντέλο περιλαμβάνει την έννοια του κλειδιού για κάθε πίνακα: ένα υποσύνολο από πεδία που καθορίζουν μοναδικά κάθε γραμμή του πίνακα, όπως στην Εικόνα 2.4. Εν τέλει, ένα πεδίο ενός πίνακα μπορεί να οριστεί ως ξένο κλειδί όταν παραπέμπει σε ένα πεδίο ενός άλλου πίνακα. Αυτό χρησιμοποιείται για τη δημιουργία περιορισμών μεταξύ διαφορετικών οντοτήτων.

item (key: id):
id -> varchar(30)
type -> varchar(10)
price -> int(11) NULL
name -> varchar(100)

creator (key: id, author):
id -> varchar(30)
author -> varchar(100)

book (key: isbn):
isbn -> int(11) auto_incr
type -> varchar(10) [Volume]
year -> int(11)
title -> varchar(100)

author (key: firstname, lastname):
firstname -> varchar(30)
middlename -> varchar(30)
lastname -> varchar(30)

writer (key: isbn, firstname, lastname):
isbn -> int(11)
firstname -> varchar(30)
lastname -> varchar(30)

id	type	price	name
89	Pocket	9.95	La chute
134	Popular	60	My life
77	Textbook		Introduction to logic
58	Science		Principia mathematica

id	name
89	Albert Camus
58	Alfred N. Whitehead
77	Alfred Tarski
134	Bertrand Russell
58	Bertrand Russell

isbn	type	year	title
2070360105	Novel	1956	La chute
0415189853	Autobiogr	1969	My life
048628462X	Essay	1941	Introduction to logic

firstname	middlename	lastname
Albert		Camus
Alfred	North	Whitehead
Alfred		Tarski
Bertrand		Russell

isbn	firstname	lastname
2070360105	Albert	Camus
2070394387	Albert	Camus
0521626064	Bertrand	Russell
0521626064	Alfred	Whitehead
0415189853	Bertrand	Russell
048628462X	Alfred	Tarski

Εικόνα 2.4 Τμήματα δύο σχημάτων βάσεων δεδομένων

Τα σχήματα της Εικόνας 2.4 παρουσιάζονται με κάποια στιγμιότυπα στους πίνακες. Αναπαριστούν όμοιες συλλογές πληροφορίας για βιβλία και συγγραφείς, με διαφορετικούς, όμως, τρόπους.

Κατά μια έννοια, οι σχεσιακές βάσεις δεδομένων είναι σχετικά περιορισμένες: τα κελιά του πίνακα μπορεί να περιέχουν μόνο πρωταρχικούς τύπους δεδομένων, όπως αλφαριθμητικά και ακεραίους. Για παράδειγμα, το σχήμα που υπάρχει δεξιά στην Εικόνα 2.4, για να αναπαραστήσει τη σχέση μεταξύ ενός βιβλίου και των συγγραφέων του, απαιτείται ένας επιπλέον πίνακας με τη συνένωση (join) των κλειδιών από τους πίνακες *βιβλίο* και *συγγραφέας*. Επιπλέον, το σχεσιακό μοντέλο υστερεί στην οργάνωση των δεδομένων όπως σε μια ταξονομία. Και στα δύο σχήματα της Εικόνας 2.4, οι πίνακες που αντιστοιχούν στην έννοια *βιβλίο* έχουν ένα πεδίο *τύπος* για την ανάθεση του ονόματος της κλάσης στα αντικείμενα. Έχουν

προταθεί διάφορες προσεγγίσεις για την επίλυση του προβλήματος της εκφραστικότητας. Για παράδειγμα, η χρήση ενός πιο εκφραστικού μοντέλου, όπως ένα μοντέλο οντοτήτων-σχέσεων κατά τη διάρκεια της σχεδίασης και στη συνέχεια η δημιουργία της βάσης δεδομένων από αυτό, ή η χρήση ενός πιο περίπλοκου μοντέλου, όπως είναι το μοντέλο αντικειμενοστραφούς βάσης δεδομένων.

Τελικά, αξίζει να αναφερθούν οι ευρέως διαδεδομένες γλώσσες για τον προσδιορισμό σχεσιακών σχημάτων, όπως είναι η Structured Query Language (SQL), που παρέχει δυνατότητες μοντελοποίησης, πχ. τύποι καθορισμένοι από το χρήστη, συνάθροιση, γενίκευση, κλπ.

XML σχήματα

Οι Document Type Definitions (DTDs) και τα XML σχήματα δημιουργήθηκαν για τον προσδιορισμό της δομής των XML αρχείων. Τα βασικά συστατικά των XML σχημάτων είναι στοιχεία, χαρακτηριστικά και τύποι. Τα στοιχεία μπορεί να είναι σύνθετα για τον καθορισμό εμφωλευμένων υποστοιχείων, ή απλά για τον καθορισμό ενσωματωμένων τύπων, όπως αλφαριθμητικά, για ένα στοιχείο ή ένα χαρακτηριστικό. Τα XML σχήματα λειτουργούν μάλλον συμπληρωματικά ως προς τα ευρετήρια, αντί να περιγράφουν πως κατηγοριοποιούνται τα αντικείμενα, περιγράφουν τα αντικείμενα εκ των έσω. Για παράδειγμα, το σχήμα στην αρχή της Εικόνας 2.5 περιγράφει το στοιχείο *Product* που περιλαμβάνει ένα στοιχείο *name* που είναι αλφαριθμητικό, ένα *id* που είναι ένα URI, ένα στοιχείο *price* που είναι μη αρνητικός ακέραιος, και τα πεδία *topics* που είναι αλφαριθμητικά. Επίσης, περιγράφει το στοιχείο *Book* που είναι ένα στοιχείο τύπου *Product* και επιπροσθέτως, έχει μια ακολουθία από στοιχεία *Author*, που με τη σειρά τους είναι τύπου *Person*, και ένα ακριβώς στοιχείο τύπου *Publisher*. Ακόμα και αν οι ορισμοί των στοιχείων επεκταθούν ή περιοριστούν ως υποκατηγορίες μιας κατηγοριοποίησης, η έμφαση δίνεται στη δομή τους: η επέκταση ενός στοιχείου γίνεται παρέχοντας τα στοιχεία που αλλάζουν στη δομή. Η ακολουθιακή πτυχή που χαρακτηρίζει τα XML αρχεία είναι τμήμα της προδιαγραφής του στοιχείου, παρά το γεγονός ότι μπορεί να ανατραπεί.

Στην πραγματικότητα, τα σχήματα αυτά παρέχουν τη μορφή σύμφωνα με την οποία τα μελλοντικά αρχεία θα δημιουργηθούν, σε αντίθεση με μια οντολογία, που είναι η περιγραφή εξωτερικών αντικειμένων που υπάρχουν. Η ιεραρχία εξειδίκευσης των XML σχημάτων είναι ένας τύπος ιεράρχισης που ορίζει ποιο είδος

στοιχείων μπορεί να καταλάβει τη θέση κάποιου άλλου, πχ. αν ένα ράφι περιέχει βιβλία, τότε επιτρέπεται να τοποθετηθεί στο ράφι μια βιογραφία. Κατ' αρχήν, αυτή η δομή κατηγοριοποίησης δεν είναι απαραίτητο να ικανοποιεί κάποια φυσική κατηγοριοποίηση των αντικειμένων.

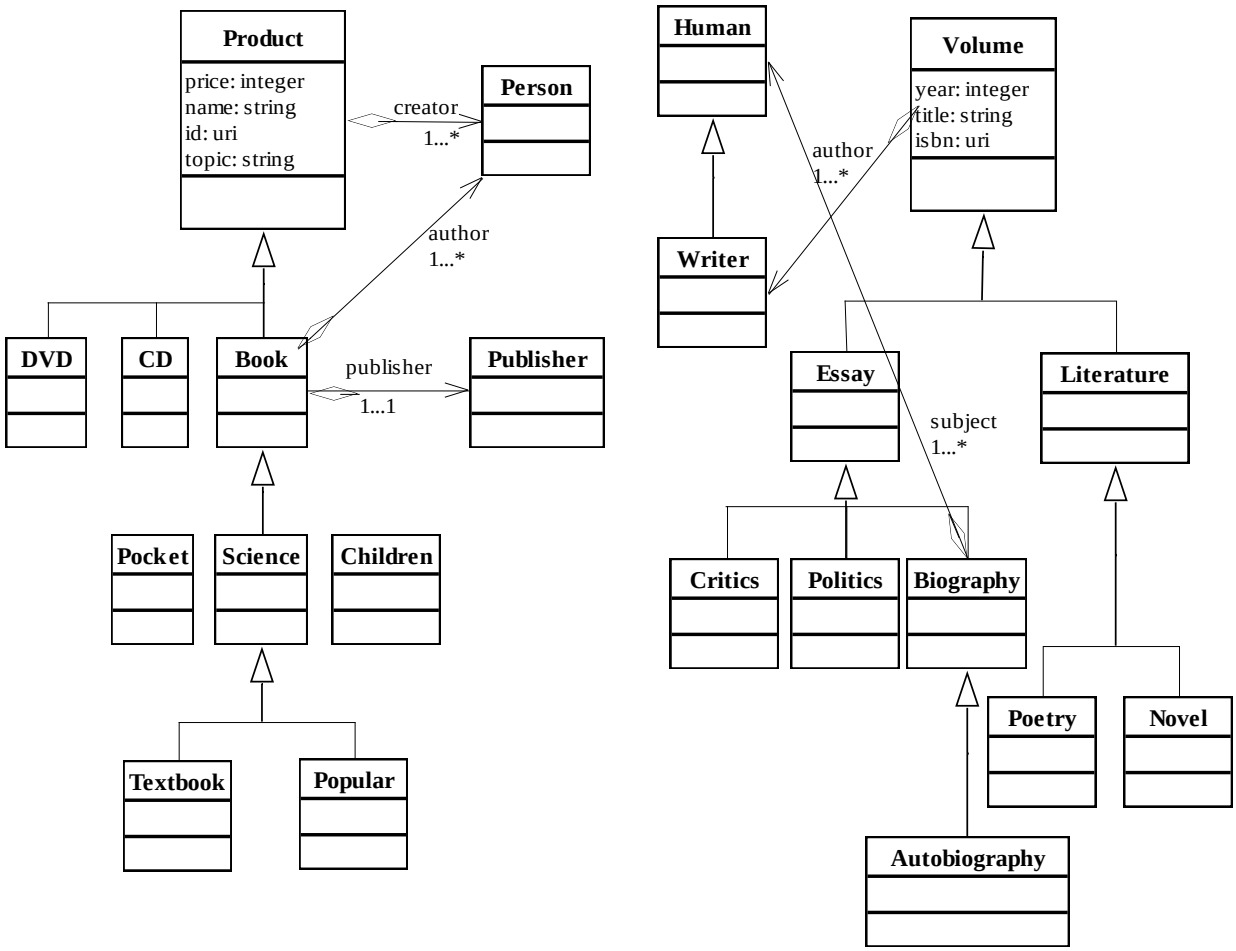
```
<schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.w3.org/2001/XMLSchema">
    <complexType name="Person">
        <sequence><element name="name" type="xsd:string"/></sequence>
    </complexType>

    <simpleType name="creator"><restriction base="Person"/></simpleType>
    <simpleType name="author"><restriction base="creator"/></simpleType>

    <complexType name="Product">
        <sequence>
            <element ref="creator" minOccurs="1"/>
            <element name="name" type="xsd:string" minOccurs="1"/>
            <element name="id" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
            <element name="price" type="xsd:nonNegativeInteger" minOccurs="1"/>
            <element name="topic" type="xsd:string"/>
        </sequence>
    </complexType>

    <complexType name="Book">
        <complexContent>
            <extension base="Product">
                <sequence>
                    <element ref="author" type="xsd:any"/>
                    <element name="publisher" type="Publisher" minOccurs="1" maxOccurs="1"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

```
<schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.w3.org/2001/XMLSchema">
    <complexType name="Volume">
        <sequence>
            <element name="author" type="Writer" minOccurs="1"/>
            <element name="title" type="xsd:string" minOccurs="1"/>
            <element name="year" type="xsd:decimal"/>
        </sequence>
        <attribute name="isbn" type="xsd:anyURI"/>
    </complexType>

    <complexType name="Essay">
        <complexContent>
            <extension>
                <sequence><element name="subject" type="xsd:any"/></sequence>
            </extension>
        </complexContent>
    </complexType>

    <complexType name="Human">
        <sequence>
            <element name="firstname" type="xsd:string"/>
            <element name="middlename" type="xsd:string"/>
            <element name="lastname" type="xsd:string"/>
        </sequence>
    </complexType>

    <complexType name="Writer">
        <complexContent><extension base="Human"/></complexContent>
    </complexType>
</schema>
```

Εικόνα 2.5 Τμήματα δύο XML σχημάτων

Εννοιολογικά μοντέλα

Συχνά οι ερευνητές των βάσεων δεδομένων δε λαμβάνουν υπόψη το ίδιο το σχεσιακό σχήμα, αλλά το υποκείμενο μοντέλο οντοτήτων-σχέσεων [15]. Τα εννοιολογικά μοντέλα καλύπτουν αυτά που περιγράφονται στο [16], καθώς επίσης και τα μοντέλα οντοτήτων-σχέσεων [17], που αποσκοπούν στην αφαιρετική προσέγγιση των αντικειμενοστραφών προγραμμάτων.



Εικόνα 2.6 Τμήματα δύο εννοιολογικών μοντέλων στη μορφή UML διαγραμμάτων κλάσεων. Τα κουτιά περιγράφουν τις οντότητες και την εσωτερική τους δομή. Η εξειδίκευση εκφράζεται με κάθετα τριγωνικά βέλη.

Τα μοντέλα αυτά προσφέρουν έναν εμπλουτισμένο τρόπο αναπαράστασης των οντοτήτων, που σε αυτήν την περίπτωση μπορούν να θεωρηθούν ως οντότητες ενός μοντελοποιημένου τομέα, όπως άνθρωποι σε μια βάση δεδομένων ή προδιαγραφές οντοτήτων που πρόκειται να δημιουργηθούν, όπως προγράμματα. Τα εννοιολογικά μοντέλα προσφέρουν μεθόδους-δημιουργούς (constructors) για την

οργάνωση κλάσεων σε μια ιεραρχία καθώς επίσης και μεθόδους-δημιουργούς για την περιγραφή της εσωτερικής δομής των αντικειμένων. Έτσι έχουμε στη διάθεση μας τον καλύτερο συνδυασμό: ευρετήρια και βάσεις δεδομένων. Για παράδειγμα, η Εικόνα 2.6 περιγράφει δύο UML διαγράμματα κλάσεων που αντιστοιχούν στο ίδιο είδος μοντέλων που παρουσιάστηκαν παραπάνω: μια ταξονομία κλάσεων από μια ιστοσελίδα ηλεκτρονικού εμπορίου με sθέμα την πώληση πολιτισμικών αγαθών αριστερά και μια βιβλιοθήκη δεξιά. Και τα δύο αποτελούν μια πλήρη περιγραφή των αντικειμένων τους μέσω των προδιαγραφών των ιδοτήτων τους και μέσω μιας ταξονομίας κλάσεων. Επιπλέον, μπορούν να εκφράσουν σχέσεις μεταξύ των κλάσεων, πχ. ότι ο *author* ενός *Book* είναι ένα *Person* στο μοντέλο αριστερά. Τα δύο μοντέλα της Εικόνας 2.6 εκφράζουν συγκρίσιμους τομείς, πχ. ένα *Volume* μπορεί να αντιστοιχεί σε ένα *Book*, ακόμη και αν πρόκειται για εντελώς διαφορετικούς τομείς, πχ. δεν υπάρχει υπερκλάση *Product* στο μοντέλο που βρίσκεται δεξιά.

Οντολογίες

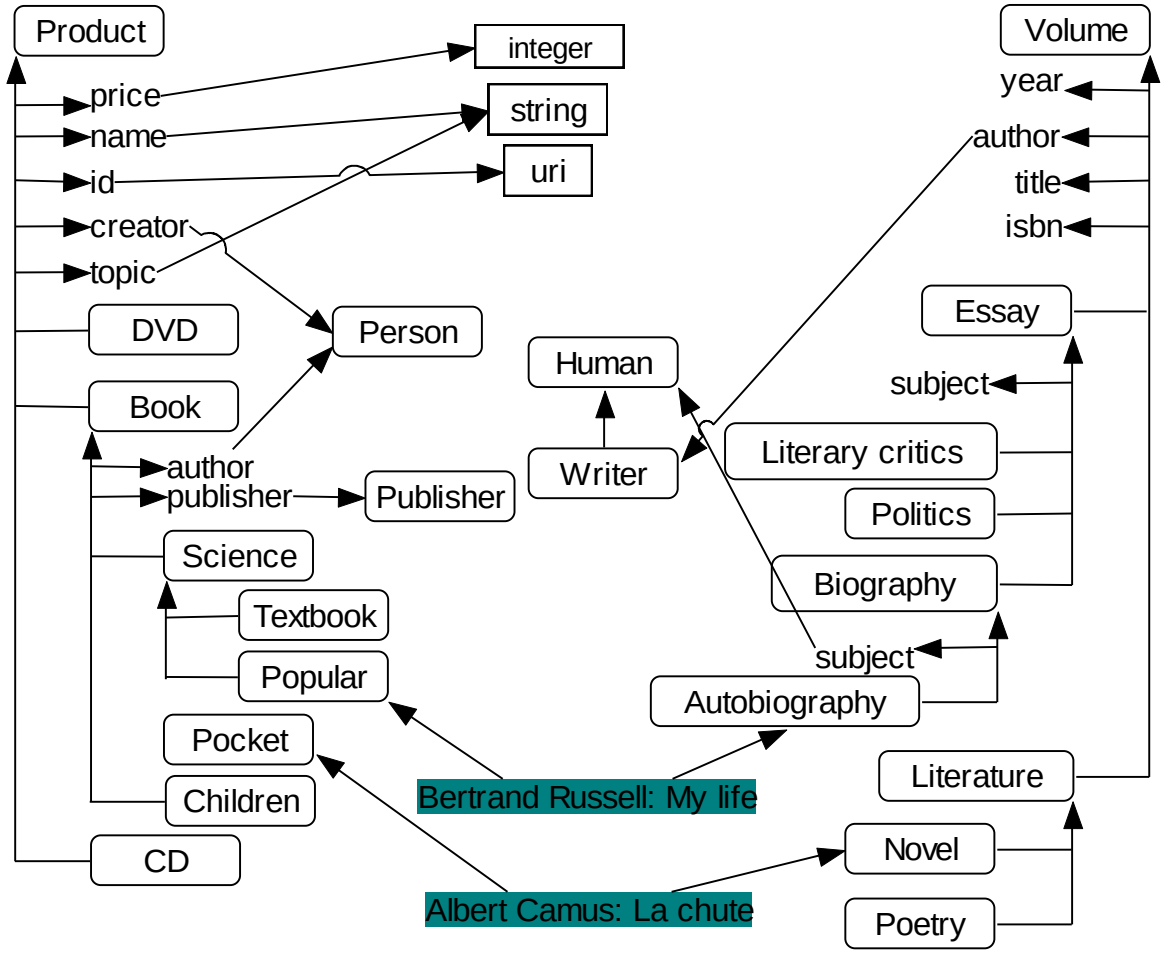
Είναι συνηθισμένο στις μέρες μας, ευρετήρια ή εννοιολογικά μοντέλα να προωθούνται ως οντολογίες. Οι οντολογίες διαθέτουν τα περισσότερα από τα χαρακτηριστικά των μοντέλων οντοτήτων-σχέσεων, κι έτσι διαθέτουν και πιο πολλά είδη σχημάτων που περιγράφηκαν παραπάνω. Οι οντολογίες της Εικόνας 2.7 αντιστοιχούν συντακτικά στα μοντέλα της Εικόνας 2.6.

Το ξεχωριστό χαρακτηριστικό των οντολογιών είναι η ύπαρξη ενός θεωρητικού μοντέλου σημασιολογίας: οι οντολογίες είναι θεωρίες λογικής. Η ερμηνεία των οντολογιών δε γίνεται από τους χρήστες που διαβάζουν τα διαγράμματα, ή από τα συστήματα διαχείρισης της γνώσης που τις υλοποιούν, αλλά είναι ορισμένη σαφώς. Η σημασιολογία παρέχει τους κανόνες για την ερμηνεία του συντακτικού, το οποίο δεν παρέχει την ερμηνεία άμεσα, αλλά περιορίζει τις πιθανές ερμηνείες αυτών που ορίζονται.

Είναι κοινότυπο, στη θεωρητική έρευνα των βάσεων δεδομένων να αντιμετωπίζονται οι σχεσιακές βάσεις με σημασιολογία πρώτης τάξης (first order semantics). Όμως, αυτό δεν αποτελεί τμήμα του προτύπου της SQL [18]. Επιπλέον, η σχεσιακή άλγεβρα που χρησιμοποιείται στα σχήματα των βάσεων δεν είναι αρκετά εκφραστική: η εκφραστικότητα υπόκειται στη γλώσσα επερώτησης.

Για αυτούς τους λόγους, της πλούσιας εκφραστικότητας και της παρουσίας ενός θεωρητικού μοντέλου σημασιολογίας, εστιάζουμε ειδικά στις οντολογίες. Παραδοσιακά, οι οντολογίες θεωρούνται διαφορετικές από τις βάσεις γνώσης, όπως ένα σχήμα βάσης είναι διαφορετικό από τη βάση που το χρησιμοποιεί.

Η σημασιολογία των οντολογιών μπορεί να περιοριστεί από επιπρόσθετα αξιώματα. Αυτό θα μπορούσε να είναι, σε κάποιες γλώσσες, η δυνατότητα της προσθήκης αξιωμάτων, όπως ότι μια αυτοβιογραφία είναι μια βιογραφία με θέμα το συγγραφέα της.



Εικόνα 2.7 Τμήματα δύο οντολογιών

2.1.2 Είδη Ετερογένειας

Σκοπός της συσχέτισης σχημάτων είναι η μείωση της ετερογένειας μεταξύ τους. Η ετερογένεια δεν υπάρχει αποκλειστικά στις διαφορές μεταξύ των στόχων της εκάστοτε εφαρμογής και το σκοπό για τον οποίο σχεδιάστηκε, ή στην έκφραση των

αντικειμένων και πως αυτά κωδικοποιήθηκαν. Έως τώρα, έχουν γίνει πολλές διαφορετικές κατηγοριοποιήσεις των τύπων ετερογένειας [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]. Κάποιες από αυτές εστιάζουν στις αναντιστοιχίες [28], ενώ άλλες αναφέρονται σε επίπεδα διαλειτουργικότητας [29]. Παρακάτω παρουσιάζονται οι πιο προφανείς τύποι ετερογένειας.

Η *συντακτική ετερογένεια* εμφανίζεται όταν δύο σχήματα δεν εκφράζονται με την ίδια γλώσσα. Αυτό, προφανώς, συμβαίνει όταν, για παράδειγμα, συγκρίνεται ένα ευρετήριο με ένα εννοιολογικό μοντέλο. Επίσης, συμβαίνει όταν δύο σχήματα μοντελοποιούνται χρησιμοποιώντας διαφορετικούς φορμαλισμούς αναπαράστασης γνώσης, όπως OWL και F-logic. Αυτό το είδος αναντιστοιχίας, γενικά, αντιμετωπίζεται σε θεωρητικό επίπεδο όταν ορίσει κάποιος ισοδυναμίες μεταξύ των δομικών στοιχείων των διαφορετικών γλωσσών. Έτσι, μερικές φορές είναι πιθανό να μεταφράζονται οντολογίες μεταξύ διαφορετικών γλωσσών οντολογιών, ενώ το νόημα παραμένει ίδιο [34].

Η *ετερογένεια ορολογίας* εμφανίζεται εξαιτίας των παραλλαγών των ονομάτων όταν αναφέρονται στις ίδιες οντότητες σε διαφορετικά σχήματα. Αυτό μπορεί να προκληθεί από τη χρήση διαφορετικών φυσικών γλωσσών, πχ. Paper αντί Articulo, διαφορετικών τεχνικών υπογλωσσών, πχ. Paper αντί Memo, ή από τη χρήση συνονύμων, πχ. Paper αντί Article.

Η *εννοιολογική ετερογένεια*, που ονομάζεται και *σημασιολογική ετερογένεια* στο [29] και *λογική αναντιστοιχία* στο [28], περιγράφει τις διαφορές στη μοντελοποίηση του ίδιου τομέα. Αυτό μπορεί να συμβεί εξαιτίας της χρήσης διαφορετικών (και καμιά φορά και ισοδύναμων) αξιωμάτων, για τον ορισμό εννοιών ή εξαιτίας της χρήσης εντελώς διαφορετικών εννοιών, πχ. μοντελοποίηση της γεωμετρίας χρησιμοποιώντας σημεία ως θεμελιακά στοιχεία και μοντελοποίηση της γεωμετρίας χρησιμοποιώντας σφαίρες ως θεμελιακά στοιχεία. Σύμφωνα με το [28] και το [35], υπάρχει μια διαφορά ανάμεσα στην εννοιολογική αναντιστοιχία, που βασίζεται στις διαφορές μεταξύ των μοντελοποιημένων εννοιών, και στη ρητή αναντιστοιχία, που βασίζεται στον τρόπο με τον οποίο εκφράζονται αυτές οι έννοιες.

Τελικά, στο πλαίσιο των εννοιολογικών διαφορών, το [36] προτείνει τρεις σημαντικές αιτίες τους. Παρακάτω τις περιγράφουμε και δίνονται παραδείγματα με βασική ιδέα ένα γεωγραφικό χάρτη:

- *Διαφορά κάλυψης (Difference in coverage)*, συμβαίνει όταν δύο οντολογίες περιγράφουν διαφορετικές, ίσως επικαλυπτόμενες, περιοχές, στο ίδιο επίπεδο λεπτομέρειας και υπό το ίδιο πρίσμα προσέγγισης. Χαρακτηριστικό παράδειγμα της περίπτωσης αυτής είναι δύο μερικά επικαλυπτόμενοι γεωγραφικοί χάρτες.
- *Διαφορά (Difference in granularity),* συμβαίνει όταν δύο οντολογίες περιγράφουν τον ίδιο τομέα ενδιαφέροντος, υπό την ίδια σκοπιά, αλλά σε διαφορετικό επίπεδο λεπτομέρειας. Αυτό μπορεί να εντοπιστεί σε γεωγραφικούς χάρτες διαφορετικής κλίμακας, πχ. ο πρώτος αναπαριστά κτίρια ενώ ο δεύτερος πόλεις.
- *Διαφορά οπτικής/σκοπιάς (Difference in perspective),*γνωστή και ως διαφορά εύρους (scope) [37], συμβαίνει όταν δύο οντολογίες περιγράφουν τον ίδιο τομέα ενδιαφέροντος, στο ίδιο επίπεδο λεπτομέρειας, υπό διαφορετικό πρίσμα. Παράδειγμα αποτελούν χάρτες διαφορετικού σκοπού: ένας πολιτικός και ένας γεωγραφικός χάρτης δεν αναπαριστούν τα ίδια αντικείμενα.

Η σημειωτική (semiotic) ετερογένεια, γνωστή και ως πρακτική (pragmatic) ετερογένεια [33], εστιάζει στον τρόπο με τον οποίο οι οντότητες ερμηνεύονται από τους ανθρώπους. Πράγματι, οντότητες που έχουν ακριβώς την ίδια σημασιολογική ερμηνεία ερμηνεύονται από τους ανθρώπους με βάση το πλαίσιο, για παράδειγμα πως χρησιμοποιούνται. Το είδος αυτό της ετερογένειας είναι δύσκολο να εντοπιστεί από τον υπολογιστή, και ακόμη πιο δύσκολο να αντιμετωπιστεί επειδή είναι ανέφικτο για τον ίδιο τον υπολογιστή. Η προτεινόμενη χρήση των οντοτήτων έχει σημαντική επίδραση στην ερμηνεία τους, έτσι, η συσχέτιση οντοτήτων που δεν έχουν όμοια εφαρμογή στο ίδιο πλαίσιο είναι επιρρεπής σε λάθη. Δεδομένου των περιορισμένων δυνατοτήτων που διαθέτει ο υπολογιστής σε τέτοια ζητήματα, δεν μπορούμε να χειριστούμε το είδος αυτό της ετερογένειας στην τρέχουσα εργασία.

Συνήθως, διαφορετικοί τύποι ετερογένειας συνυπάρχουν. Στο τρίτο κεφάλαιο της παρούσας εργασίας παρουσιάζονται τεχνικές αντιμετώπισης των μορφών ετερογένειας, είτε ξεχωριστά είτε σε συνδυασμό.

2.1.3 Το Πρόβλημα της Συσχέτισης

Στην απλούστερη της μορφή η συσχέτιση σχημάτων αποτελείται από τον εντοπισμό δύο στοιχείων από δύο διαφορετικά σχήματα που είναι σημασιολογικά ισοδύναμα ή συσχετισμένα. Στην Εικόνα 2.8, παρακάτω, για παράδειγμα, οι περισσότερες μέθοδοι θα αντιμετώπιζαν μικρή δυσκολία για το αν το πεδίο Zip του σχήματος Α συσχετίζεται με το πεδίο ZipCode του σχήματος Β. Είναι ένα παράδειγμα συσχέτισης που θα μπορούσε να προκύψει με την εξέταση των ονομάτων των στοιχείων. Είναι, επίσης, και ένα παράδειγμα άμεσης συσχέτισης, μερικές φορές εντοπίζεται στη βιβλιογραφία και ως συσχέτιση πληθικότητας 1:1, που σημαίνει ότι ένα μοναδικό στοιχείο από ένα σχήμα συσχετίζεται με ένα μοναδικό στοιχείο από ένα άλλο σχήμα.

Σχήμα Α

First	Last	Address	Zip	Phone
Carol	Frenditte	1102 Washington St, Dover, MA	02051	6174431685
Allen	LeBlanc	42 Union St, Medfield, MA	02053	5089291094
Thomas	Gutierrez	74 Chestnut St, Franklin, MA	02041	5088864218

Σχήμα Β

Name	Address	City	State	ZipCode
William J Lyttle III	441 Elm St	Easton	MA	02356
Mr. Robert Sheridan	65 Georgetown Dr	Framingham	Ma	01701
Nancy Langford	891 Dudley St	Providence	RI	02919

Εικόνα 2.8 Συσχέτιση σχημάτων βασισμένη στο όνομα του στοιχείου

Μεγάλο τμήμα της έρευνας έχει αναπτυχθεί με γνώμονα τις άμεσες συσχετίσεις. Τι συμβαίνει όμως με τα υπόλοιπα στοιχεία του σχήματος; Στο προηγούμενο παράδειγμα, η διεύθυνση μπορεί να απαιτεί μια έμμεση συσχέτιση, μερικές φορές εντοπίζεται στη βιβλιογραφία ως συσχέτιση πληθικότητας 1:n, έτσι ώστε να συσχετιστεί το πεδίο Address του σχήματος Α με το τα πεδία Address, City και State του σχήματος Β. Επίσης, τα στοιχεία First και Last από το σχήμα Α θα πρέπει να συνενωθούν ώστε να συσχετιστούν με το πεδίο Name του σχήματος Β. Επίσης, ένα σχήμα μπορεί να περιέχει επιπλέον πληροφορία που δεν περιέχεται στο άλλο, όπως για παράδειγμα στο σχήμα Β που περιλαμβάνει το πρόθεμα και το επίθεμα

σαν τμήμα του στοιχείου του ονόματος. Σε αυτήν την περίπτωση, μπορεί να εφαρμοστεί μόνο μια μερική συσχέτιση ανάμεσα στα δύο σχήματα. Το φαινόμενο αυτό εντοπίζεται και στο στοιχείο Phone του σχήματος Α που δεν αντιστοιχεί με κανένα στοιχείο του σχήματος Β.

Επιπροσθέτως, μερικές συσχετίσεις αναφέρονται ως σύνθετες συσχετίσεις, και συναντώνται στη βιβλιογραφία ως συσχετίσεις πληθικότητας m:n, και πρόκειται για πολλαπλά στοιχεία του ενός σχήματος που συσχετίζονται με πολλαπλά στοιχεία του άλλου. Σε σχέση με το παραπάνω παράδειγμα, αν το σχήμα Β αναπαριστούσε το πεδίο Name με τα πεδία Prefix, First, Middle Initial, Last και Suffix τότε θα ήταν απαραίτητη μια σύνθετη συσχέτιση.

Η θεώρηση συσχετίσεων πληθικότητας m:n αυξάνει την πολυπλοκότητα της συσχέτισης σχημάτων εκθετικά. Επιπλέον, επειδή υπάρχει η πιθανότητα τα δεδομένα να πρέπει να μετασχηματιστούν πριν εκτελεστεί η διαδικασία συσχέτισης τους, η πολυπλοκότητα της συσχέτισης σχημάτων, γενικά, θεωρείται απεριόριστη [2]. Θεωρείστε το παράδειγμα της Εικόνας 2.9, στο οποίο κάθε σχήμα έχει ένα στοιχείο με όνομα Price αλλά σημασιολογικά δεν είναι ισοδύναμα.

Σχήμα Γ

Item	Qty	Price	Taxes
1405	5	$110.00	$6.00
1982	3	$45.00	$2.25
2023	1	$18.00	$.90

Σχήμα Δ

Item	Quantity	Price	Total
A110C	2	$11.00	$22.00
AV99x	4	$9.00	$36.00
AL129	5	$18.00	$18.00

Εικόνα 2.9 Παράδειγμα σύνθετης συσχέτισης σχημάτων

Στην περίπτωση που πρέπει να συσχετιστούν τα δύο αυτά σχήματα, ο συσχετιστής (matcher) θα πρέπει όχι μόνο να ανακαλύψει τη σημασιολογική διαφορά αλλά και

να αποτιμήσει τα πεδία Price και Price+Taxes του σχήματος Γ ως συσχετίσεις με το Total του σχήματος Δ.

Το παράδειγμα αυτό αναδεικνύει, επίσης, ότι στιγμιότυπα δεδομένων προσθέτουν πληροφορία στη διαδικασία συσχέτισης σχημάτων και ότι οι προσεγγίσεις που βασίζονται αμιγώς στην εξέταση του σχήματος και μόνο μπορεί να μην είναι αρκετή. Εκτός από τα παραπάνω, γίνεται κατανοητό ότι οι τύποι δεδομένων μπορεί να χρησιμοποιηθούν ώστε να βελτιωθεί η ακρίβεια των συσχετίσεων. Σε αυτήν την περίπτωση και τα δύο πεδία Qty και Quantity έχουν τον ίδιο τύπο δεδομένων (ακέραιος) και αυτό το τμήμα της πληροφορίας του σχήματος αυξάνει την πληροφορία για άλλες, υπό εξακρίβωση, συσχετίσεις.

Τα παραδείγματα που προαναφέρθηκαν, ανήκουν στην κατηγορία της συσχέτισης σχημάτων βασισμένη στα στοιχεία, δηλαδή οι συσχετίσεις καθορίζονται χωρίς να λαμβάνεται υπ' όψη η γνώση της δομής της βάσης δεδομένων. Όμως σε πολλές περιπτώσεις η πληροφορία της δομής της βάσης δεδομένων δύναται να βελτιώσει τη συσχέτιση των σχημάτων.

Σχήμα Ε

Badge	Name	DeptID	HomePhone
41723	Katherine Baker	172	5082307682
56784	Mark Bharati	189	6179641242
66010	Edward Waters	189	9788917692

DeptID	Name
172	Purchasing
189	Marketing

Σχήμα ΣΤ

EmployeeID	Name	Department
13572	Kevin Li	Accounting
20473	Julie McCormack	HR
33717	Fran Liebowitz	HR

EmployeeID	PhoneType	PhoneNumber
13572	Home	4015629982
13572	Cell	4018849010
13572	Emergency	4012399497
20473	Home	5083431884
20473	Cell	5087898386
33717	Home	6172847702

Εικόνα 2.10 Παράδειγμα συσχέτισης σχημάτων χρησιμοποιώντας τη δομή

Το παράδειγμα της Εικόνας 2.10, περιγράφει τον τρόπο με τον οποίο, η συσχέτιση στοιχείων καθορίζεται μέσω της εξέτασης της δομής του κάθε σχήματος. Για να συσχετιστούν τα ονόματα των τμημάτων χρειάζεται να εξεταστεί τόσο η οντότητα του τμήματος από το σχήμα Ε όσο και η οντότητα του υπαλλήλου από το σχήμα ΣΤ. Λαμβάνοντας υπ' όψη τη δομή της βάσης δεδομένων, γίνεται εφικτή η συσχέτιση του DeptName αντί του DeptID από το σχήμα Ε με το Department του σχήματος ΣΤ, αφού έχουν, προφανώς, τον ίδιο τύπο δεδομένων.

Η συσχέτιση των αριθμών τηλεφώνου ανάμεσα στα δύο σχήματα αναδεικνύει πόσο σημαντική και απαραίτητη είναι η γνώση της δομής της βάσης δεδομένων. Ο αριθμός τηλεφώνου στο σχήμα Ε υπάρχει στον πίνακα που αντιστοιχεί στον υπάλληλο, αλλά στο σχήμα ΣΤ διατηρείται ξεχωριστός πίνακας με τους αριθμούς τηλεφώνων και συνδέεται με τον πίνακα των υπαλλήλων. Η συσχέτιση των σχημάτων προϋποθέτει αυτήν την πληροφορία. Επιπροσθέτως, το στοιχείο HomePhone από το σχήμα Ε πρέπει να συσχετιστεί μόνο με συγκεκριμένες γραμμές του πεδίου PhoneNumber του σχήματος ΣΤ. Αυτό αποτελεί απόδειξη της σημαντικής αρωγής που προσφέρει η γνώση της δομής της βάσης δεδομένων στην επεξεργασία συσχέτισης σχήματος.

Τα παραπάνω παραδείγματα περιγράφουν μερικές από τις σκοπιές των προβλημάτων που αντιμετωπίζονται στην προσπάθεια συσχέτισης σχημάτων. Οι Ram και Park [38] υποστηρίζουν ότι τέτοιου είδους προβλήματα αποτελούν αντιφάσεις ανάμεσα στα σχήματα και τα διαχωρίζουν σε αντιφάσεις επιπέδου σχήματος ή επιπέδου δεδομένων. Οι αντιφάσεις επιπέδου σχήματος, περιλαμβάνουν τις περιπτώσεις που τα σχήματα χρησιμοποιούν διαφορετικά ονόματα για όμοιες οντότητες ή χαρακτηριστικά ή διαφορετικές δομές (γενίκευση, συνάθροιση) για την αναπαράσταση όμοιων εννοιών. Ενώ, οι αντιφάσεις επιπέδου δεδομένων, περιλαμβάνουν τις περιπτώσεις που χρησιμοποιούνται διαφορετικοί τύποι δεδομένων ή μονάδες μέτρησης για την αναπαράσταση όμοιων δεδομένων.

Η δυσκολία της συσχέτισης σχημάτων οφείλεται, εν μέρει, στο γεγονός ότι τα σχήματα σχεδιάζονται από πολλούς και διαφορετικούς ανθρώπους. Ενώ, δυο διαφορετικοί άνθρωποι αντιμετωπίζουν ταυτόσημα προβλήματα, μπορεί να δημιουργήσουν πολύ διαφορετικά σχήματα [6]. Επιπλέον, ακόμη και αν χρησιμοποιούνται προτυποποιημένα σχήματα, η βελτίωση της σημασιολογικής συσχέτισης είναι ήσσονος σημασίας και περιορισμένη [6]. Εκτός από αυτό, τα μεταδεδομένα ενός σχήματος, όπως το όνομα ενός στοιχείου ή ο τύπος δεδομένων, δεν επιτρέπουν μια πλήρη αναπαράσταση της σημασιολογίας. Ακόμα και σε περιπτώσεις που τα ονόματα των στοιχείων είναι περιγραφικά, μπορεί να μην είναι εφικτός ο εντοπισμός των, απαραίτητων για τη συσχέτιση, λεπτομερειών. Από τη στιγμή που θα σχεδιαστεί κάποιο σχήμα, τμήμα της σημασιολογίας έχει χαθεί και αυτό προσθέτει πιθανή δυσκολία στις επακόλουθες προσπάθειες συσχέτισης.

Σύμφωνα με τα παραπάνω, οι προσεγγίσεις επίλυσης του προβλήματος της συσχέτισης σχημάτων, έχουν χρησιμοποιήσει πολλές διαφορετικές πηγές εισόδου, με σκοπό την απόκτηση απαραίτητης πληροφορίας για την εκτέλεση της συσχέτισης. Οι προσεγγίσεις που έχουν αναπτυχθεί συνεκτιμούν τη δομή, τους τύπους δεδομένων, τους περιορισμούς, τις προεπιλεγμένες και επιτρεπόμενες τιμές, τα πρωτεύοντα και δευτερεύοντα κλειδιά εκτός από τα ονόματα των στοιχείων και τα δεδομένα στιγμιοτύπων. Γενικότερα, οι προσεγγίσεις που αξιοποιούν την περισσότερη πληροφορία, έχουν τα καλύτερα αποτελέσματα [2]. Το επόμενο τμήμα περιγράφει μια επισκόπηση παλαιότερων ερευνών και αναλύει μια νέα κατηγοριοποίηση προσεγγίσεων συσχέτισης σχημάτων με βάση τον τύπο των δεδομένων που χρησιμοποιούν και πως γίνεται αυτό.

Η ενοποίηση σχήματος είναι ένα βήμα πιο κοντά για την ενοποίηση της πληροφορίας [5]. Η ανάγκη της ενοποίησης των δεδομένων προέκυψε από την πρόσφατη έκρηξη στην αποθήκευση δεδομένων και τις δικτυακές δυνατότητες [6], και έχει πολλαπλές και σημαντικές εφαρμογές. Στις εφαρμογές αυτές συμπεριλαμβάνονται και οι παρακάτω, χωρίς όμως να περιορίζονται σε αυτές.

2.2 Εφαρμογές

Η συσχέτιση σχημάτων αποτελεί σημαντική λειτουργία που χρησιμοποιείται σε παραδοσιακές εφαρμογές, όπως την ενοποίηση οντολογιών, την ενοποίηση σχημάτων ή τις αποθήκες δεδομένων. Συνήθως, αυτές οι εφαρμογές χαρακτηρίζονται από μοντέλα που παρουσιάζουν ετερογένεια ως προς τη δομής τους και υπόκεινται σε αυτόματη ή ημι-αυτόματη συσχέτιση. Σε τέτοιες εφαρμογές, η συσχέτιση λειτουργεί ως προαπαιτούμενο βήμα για την εκτέλεση της εκάστοτε εφαρμογής.

Εκτός από τις παραπάνω εφαρμογές, μια σειρά από άλλες αναδύεται με την πάροδο του χρόνου, όπως πράκτορες (agents), διομότιμα (peer-to-peer) συστήματα και ηλεκτρονικές υπηρεσίες. Σε αντίθεση με τις παραδοσιακές εφαρμογές, απαιτούν τη διαδικασία συσχέτισης κατά τη διάρκεια της εκτέλεσης της ίδιας της εφαρμογής και αξιοποιούν περισσότερο ακριβή εννοιολογικά μοντέλα.

Στη συνέχεια του κεφαλαίου αυτού παρουσιάζεται ένα υποσύνολο δημοφιλών εφαρμογών που αντιμετωπίζουν τη συσχέτιση ως την πιο ευλογοφανή και διαχρονική λύση. Οι λεγόμενες παραδοσιακές εφαρμογές είναι η μηχανική

οντολογιών (ontology engineering), η ενοποίηση πληροφορίας και σχημάτων, η ενοποίηση καταλόγων, οι αποθήκες δεδομένων και η ενοποίηση δεδομένων. Έπειτα, περιγράφονται νέες εφαρμογές, όπως ο διαμοιρασμός πληροφορίας σε διομότιμα συστήματα, η σύνθεση ηλεκτρονικών υπηρεσιών, αυτόνομα συστήματα επικοινωνίας (πράκτορες και κινητές συσκευές επικοινωνίας) και πλοήγηση και απάντηση επερωτήσεων στον Ιστό.

2.2.1 Μηχανική Οντολογιών

Ένα πλαίσιο στο οποίο οι χρήστες αντιμετωπίζουν ετερογενείς οντολογίες είναι η μηχανική οντολογιών και γενικότερα, η σχεδίαση, η υλοποίηση και η συντήρηση εφαρμογών βασισμένων σε οντολογίες. Η λειτουργία αυτή στηρίζεται στη συσχέτιση οντολογιών επειδή η μηχανική οντολογιών χειρίζεται πολλαπλές, κατανεμημένες και εξελισσόμενες οντολογίες.

Επεξεργασία και εισαγωγή οντολογιών

Συνήθως η ετερογένεια των οντολογιών εμφανίζεται αρχικά κατά τη διάρκεια της σχεδίασης μιας οντολογίας για ένα συγκεκριμένο τομέα. Ο σχεδιαστές συστημάτων προσανατολισμένων στη χρήση οντολογιών συχνά καλούνται να υλοποιήσουν διαφορετικές οντολογίες, είτε προς χάρη της επαναχρησιμοποίησης έτσι ώστε να αποφευχθεί η ύπαρξη πολλαπλών οντολογιών για το ίδιο θέμα, είτε επειδή είναι απαραίτητη η διασύνδεση διαφορετικών σχετικών πηγών.

Είναι συνηθισμένη η περίπτωση μια εφαρμογή να απαιτεί την ταυτόχρονη χρήση διαφορετικών εξωτερικών οντολογιών. Για παράδειγμα, η υλοποίηση μιας οντολογίας που περιγράφει την καταλογογράφηση μιας βιβλιοθήκης μπορεί να περιλαμβάνει τη συναρμολόγηση οντολογιών για ανθρώπους, βιβλία, θέματα καθώς επίσης και μονάδες μέτρησης, γεωγραφικές συντεταγμένες, αναγνωριστικούς αριθμούς βιβλίων κ.ο.κ. Αυτές οι οντολογίες μοιράζονται σχετιζόμενες έννοιες, όπως η friend-of-a-friend (FOAF[7]) οντολογία (μπορεί να χρησιμοποιηθεί κατά την έναρξη της μοντελοποίησης της έννοιας άνθρωπος) που περιγράφει την έννοια του αρχείου που σχετίζεται με τις κλάσεις της οντολογίας που περιγράφει του αναγνωριστικούς αριθμούς των βιβλίων.

Οι μηχανικοί οντολογιών χρειάζονται υποβοήθηση τόσο στον εντοπισμό σχετιζόμενων οντολογιών, όσο και στη συσχέτιση και την καταγραφή των σχέσεων

7 http://www.foaf-project.org

μεταξύ των οντοτήτων των οντολογιών αυτών. Επιπροσθέτως, υπάρχει η απαίτηση της εισαγωγής οντολογιών και της συνένωσης τους (σε αυτή την περίπτωση, μπορεί να χρησιμοποιήσουν κάποια αξιώματα που παράγονται μετά τη φάση της συσχέτισης) ή της χρήσης δεδομένων που εκφράζονται από μια τρίτη οντολογία (στην περίπτωση που αποσκοπούν στην παραγωγή ενός διαμεσολαβητή –mediator- από τα αποτελέσματα της συσχέτισης).

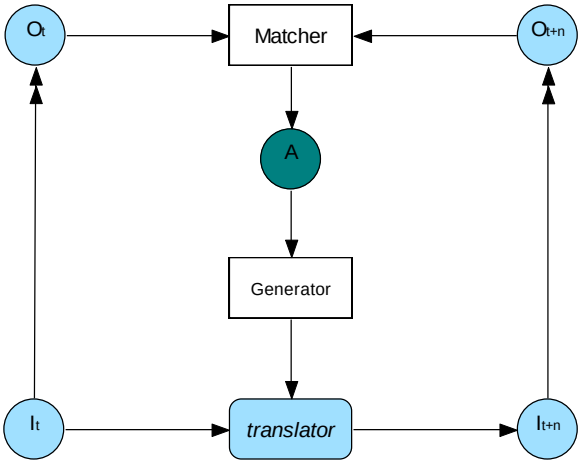
Στην πραγματικότητα οι οντολογίες εμπλέκονται κατά τη διάρκεια της σχεδίασης και οι διαμεσολαβητές δημιουργούνται εκείνη τη στιγμή. Έτσι ο σχεδιαστής της εφαρμογής μπορεί να εντοπίσει τις ομοιότητες και να σχεδιάσει τους απαραίτητους μετασχηματισμούς χειρωνακτικά. Μερικά εργαλεία παρέχουν υποστήριξη για τον εντοπισμό των συσχετίσεων, όπως η πλατφόρμα του Protégé μέσω του Prompt [39]. Τα σύγχρονα περιβάλλοντα ανάπτυξης οντολογιών πρέπει, εξ αρχής, να λαμβάνουν υπόψη την ύπαρξη πολλαπλών οντολογιών και την ανάγκη χρήσης διαμεσολαβητών μεταξύ τους.

Εξέλιξη και εκδόσεις οντολογιών

Είναι φυσικό οι τομείς, οι απαιτήσεις και ο τρόπος που οι μηχανικοί μοντελοποιούν τη γνώση μέσω τον οντολογιών αλλάζουν και εξελίσσονται με την πάροδο του χρόνου. Επιπλέον, η ανάπτυξη οντολογιών, όπως και η ανάπτυξη λογισμικού, συχνά υλοποιείται με έναν κατανεμημένο και συνεργατικό τρόπο. Συνεπώς, συχνά υπάρχουν πολλαπλές εκδόσεις της ίδιας οντολογίας, όπως η οντολογία Gene[8]. Κάποιες εφαρμογές ενημερώνουν τις οντολογίες τους, ενώ κάποιες άλλες εξακολουθούν να χρησιμοποιούν παλαιότερες εκδόσεις. Αυτές οι περιπτώσεις προκύπτουν επειδή οι μηχανικοί και οι σχεδιαστές δεν έχουν σφαιρική εικόνα της εξέλιξης των οντολογιών. Στην πραγματικότητα, οι αλλαγές στα αρχεία καταγραφών (logs) δεν είναι πάντα διαθέσιμες εξαιτίας της κατανεμημένης ανάπτυξης των οντολογιών. Έτσι οι σχεδιαστές καλούνται να χειρίζονται και να συντηρούν διαφορετικές εκδόσεις των οντολογιών.

Στο σημείο αυτό μπορεί να λειτουργήσει επικουρικά η λειτουργία της συσχέτισης, Εικόνα 2.11. Η λειτουργία αυτή εστιάζει κυρίως στον εντοπισμό των διαφορών, πχ. οι οντότητες που έχουν προστεθεί, διαγραφεί ή μετονομαστεί ανάμεσα σε δύο εκδόσεις της ίδιας οντολογίας.

8 http://www.geneontology.org



Εικόνα 2.11 Σενάριο εξέλιξης οντολογίας. Στο σενάριο αυτό: (1) συσχετίζεται (Matcher) η παλαιότερη έκδοση O_t με τη νεότερη της οντολογίας και έτσι προκύπτει ένα σύνολο αντιστοιχίσεων (A) μεταξύ των εκδόσεων αυτών, (2) παράγεται (Generator) ένας μετασχηματισμός χρησιμοποιώντας τις αντιστοιχίσεις αυτές και (3) μεταφράζονται (translator) τα δεδομένα στιγμιοτύπων από I_t σε I_{t+n}.

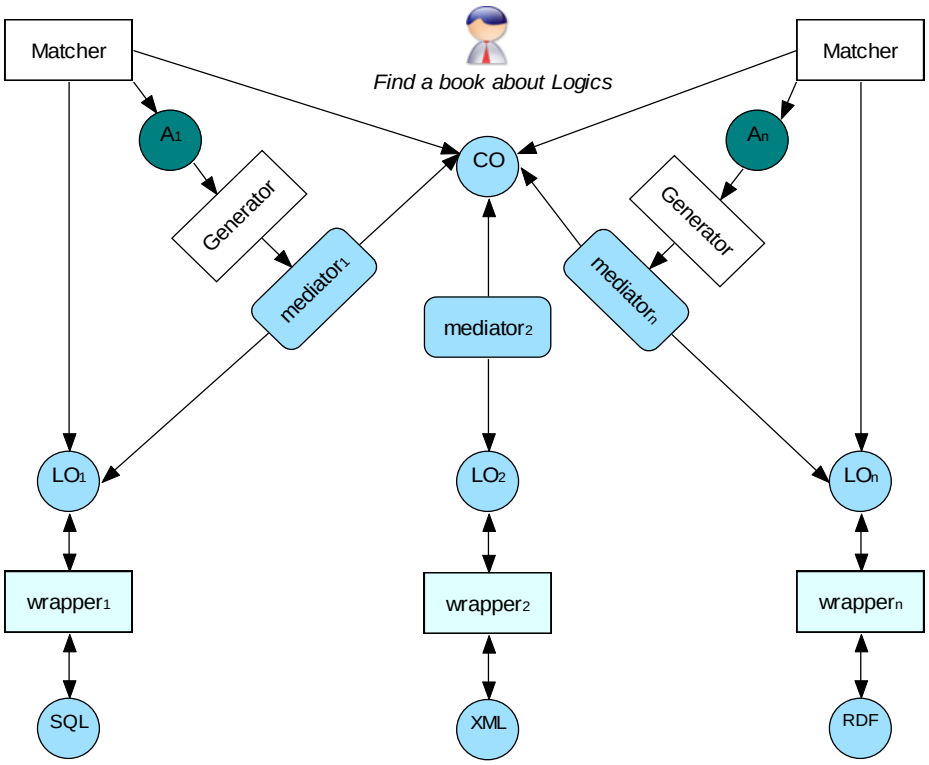
2.2.2 Ενοποίηση Πληροφορίας

Η ενοποίηση της πληροφορίας αποτελεί ένα από τα παλαιότερα σύνολα εφαρμογών που αντιμετωπίζουν τη συσχέτισης ως την πιο εύλογη λύση. Στα πλαίσια της ενοποίησης της πληροφορίας αντιμετωπίζονται προβλήματα όπως η ενοποίηση σχημάτων, οι αποθήκες δεδομένων, η ενοποίηση δεδομένων ή ενοποίηση εμπορικής πληροφορίας και η ενοποίηση καταλόγων.

Ένα γενικού σκοπού σενάριο ενοποίησης πληροφορίας παρουσιάζεται στην Εικόνα 2.12. Δεδομένου ενός συνόλου τοπικών πηγών πληροφορίας (τοπικές οντολογίες LO_1, … LO_n), που πιθανόν αποθηκεύουν τα δεδομένα τους σε διαφορετική μορφή, πχ. SQL DDL, XML ή RDF, παρέχεται στους χρήστες μια ενοποιημένη διεπαφή μέσω μιας ενδιάμεσης οντολογίας CO πάνω από όλες της τοπικές πηγές πληροφορίας. Αυτό επιτρέπει στους χρήστες να αποφεύγουν τις διαδοχικές επερωτήσεις στις τοπικές πηγές πληροφορίας μια προς μια και να εξασφαλίζουν ένα συνολικό αποτέλεσμα επερωτώντας μια κοινή οντολογία.

Για παράδειγμα, αν οι χρήστες θέτουν επερωτήσεις όπως «βρες ένα βιβλίο σχετικό με τη Λογική» σε μια κοινή οντολογία, τότε ένα σύστημα ενοποίησης της πληροφορίας επικοινωνεί με πηγές πληροφορίας, πχ. Amazon, Barnes & Noble, και

επιστρέφει ένα συνολικό αποτέλεσμα βασισμένο στην είσοδο που παρέχεται από αυτές τις πηγές.



Εικόνα 2.12 Γενικό σενάριο ενοποίησης πληροφορίας. Οι πηγές δεδομένων (SQL, RDF, κλπ.) μετασχηματίζονται (wrapperi) σε οντολογίες (LOi), οι οποίες συσχετίζονται με βάση μια κοινή οντολογία (CO). Οι ευθυγραμμίσεις (Ai) μεταξύ αυτών βοηθούν στην παραγωγή (Generator) μεσολαβητών (mediatori), οι οποίοι με τη σειρά τους μετασχηματίζουν τις επερωτήσεις που τίθενται στην κοινή οντολογία, σε επερωτήσεις στην πηγή πληροφορίας και μεταφράζουν τις απαντήσεις με την αντίθετη κατεύθυνση.

Γενικά, το σύστημα ενοποίησης της πληροφορίας εκτελεί τα παρακάτω βήματα:

- Μετάφραση της επερώτησης σε όρους της κοινής οντολογίας
- Εντοπισμός των σημασιολογικών ομοιοτήτων ανάμεσα στις οντότητες των τοπικών πηγών και στην κοινή οντολογία
- Μετάφραση των σχετιζόμενων στιγμιοτύπων των τοπικών πηγών σε μια αναπαράσταση της γνώσης του συστήματος ενοποίησης της πληροφορίας
- Συνυπολογισμός των αποτελεσμάτων που προκύπτουν από τις πολλαπλές τοπικές πηγές, μέσω λειτουργιών όπως εντοπισμός και απαλοιφή πλεονασμών και διπλοτύπων και επιστροφή του τελικού αποτελέσματος

Ο εντοπισμός των σημασιολογικών ομοιοτήτων ανάμεσα στις οντότητες των τοπικών πηγών και στην κοινή οντολογία, αποτελεί το βήμα συσχέτισης. Στη συνέχεια περιορίζουμε την προσέγγιση της συσχέτισης στην περιγραφή που δόθηκε παραπάνω.

Στον αντίποδα του, γενικού σκοπού, σενάριο που δόθηκε προηγουμένως, βρίσκονται συγκεκριμένα σενάρια που η κοινή οντολογία μπορεί είτε να υφίσταται είτε να είναι εικονική. Παρακάτω, εξετάζονται τέτοιου είδους σενάρια, λεπτομερώς.

Ενοποίηση σχημάτων

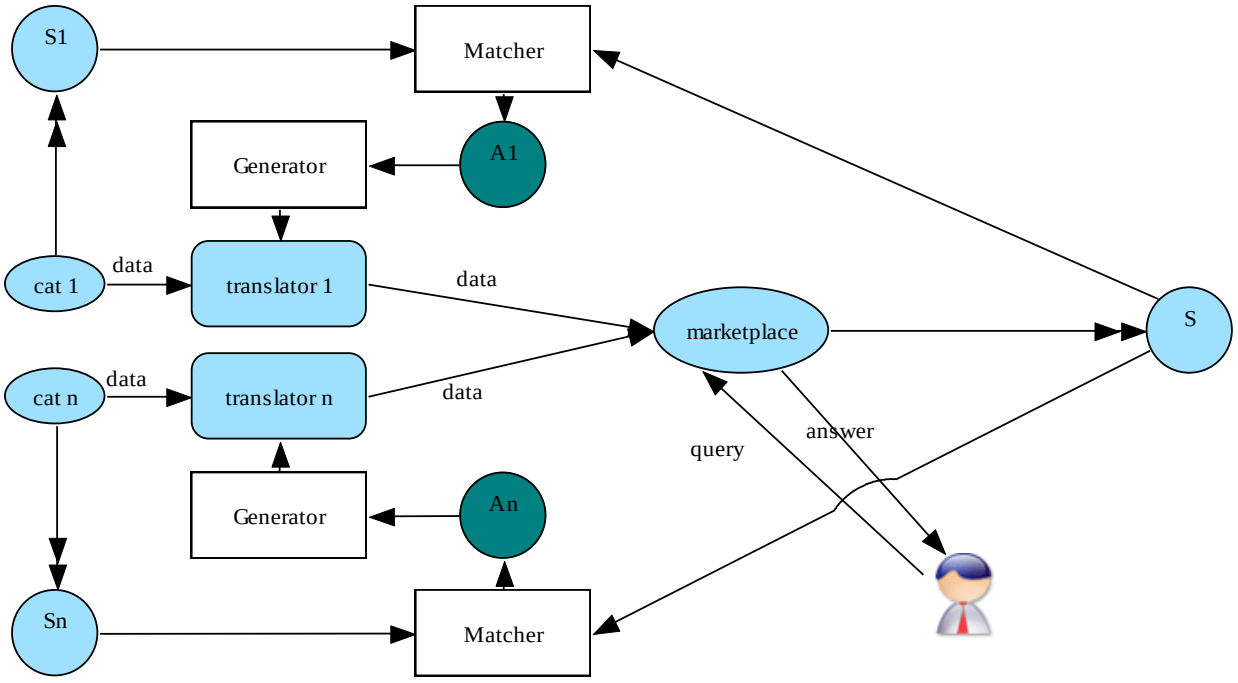
Η ενοποίηση σχημάτων αποτελεί το παλαιότερο σενάριο. Ας υποθέσουμε ότι δύο ή περισσότερες επιχειρήσεις επιθυμούν είτε να συνενωθούν είτε να εξαγοράσει η πρώτη τη δεύτερη. Τελικά, οι επιχειρήσεις αυτές θα πρέπει να ενοποιήσουν τις βάσεις δεδομένων τους σε μια και μοναδική. Συνήθως, το πρώτο βήμα είναι ο εντοπισμός των σημασιολογικών ομοιοτήτων ανάμεσα στις οντότητες των σχημάτων πριν τη συνένωση των βάσεων δεδομένων. Αυτό το βήμα, που είναι γνωστό ως συσχέτιση, απαιτείται ακόμη και αν οι βάσεις δεδομένων, που πρόκειται να ενοποιηθούν, προέρχονται από τον ίδιο τομέα, πχ. πωλήσεις βιβλίων, ενοικιάσεις αυτοκινήτων. Αυτό συμβαίνει επειδή τα σχήματα σχεδιάζονται και αναπτύσσονται ανεξάρτητα. Στην πραγματικότητα, οι άνθρωποι ακολουθούν ποικίλες αρχές μοντελοποίησης, ακόμα και όταν πρόκειται να κωδικοποιήσουν ένα και μοναδικό αντικείμενο του πραγματικού κόσμου. Επιπλέον, τα σχήματα που πρόκειται να ενοποιηθούν μπορεί να έχουν αναπτυχθεί βάσει διαφορετικών επιχειρηματικών στόχων, γεγονός που κάνει το πρόβλημα της συσχέτισης δυσκολότερο.

Υπό το πρίσμα της ενοποίησης σχημάτων, μπορεί να προταθούν εναλλακτικές κατηγοριοποιήσεις σεναρίων. Ένα παράδειγμα είναι οι ομόσπονδες βάσεις δεδομένων (federated databases). Οι ομόσπονδες βάσεις συνήθως χαρακτηρίζονται από ένα καθολικό σχήμα που παρέχει ενοποιημένη πρόσβαση στις επιμέρους βάσεις δεδομένων της ομοσπονδίας. Οι επιμέρους βάσεις δεδομένων είναι αυτόνομες. Έτσι, για παράδειγμα, στη συγκεκριμένη εφαρμογή, όταν ένα επιμέρους σχήμα της ομόσπονδης βάσης αλλάζει, τότε το καθολικό σχήμα θα πρέπει να αναδιαρθρωθεί. Η συσχέτιση μπορεί να υποστηρίξει τον εντοπισμό των αλλαγών αυτών.

Επίσης, αξίζει να αναφερθούν εφαρμογές που δεν εξετάζονται περαιτέρω, όπως είναι τα συστήματα κατανεμημένων βάσεων δεδομένων. Τα συστήματα κατανεμημένων βάσεων δεδομένων, συνήθως σχεδιάζονται με έναν κεντρικό τρόπο, πχ. από τον διαχειριστή της βάσης δεδομένων και έτσι δεν υπάρχει σημασιολογική ετερογένεια στη φάση της δημιουργίας.

Ενοποίηση καταλόγων

Σε εφαρμογές μεταξύ επιχειρήσεων (Business-to-Business, B2B), οι εμπορικοί εταίροι αποθηκεύουν πληροφορία των προϊόντων σε ηλεκτρονικούς καταλόγους. Κλασικά παραδείγματα τέτοιων καταλόγων αποτελούν οι κατάλογοι ηλεκτρονικών πωλήσεων ιστοχώρων όπως το Amazon ή το eBay. Όταν ένας έμπορος θέλει να συμμετάσχει στην αγορά, πχ. το eBay, θα πρέπει να καθορίσει τις ομοιότητες ανάμεσα στις καταχωρήσεις των καταλόγων του και σε αυτές των καταλόγων της αγοράς (Εικόνα 2.13). Η διαδικασία του εντοπισμού των ομοιοτήτων ανάμεσα στις καταχωρήσεις των καταλόγων αναφέρεται ως πρόβλημα συσχέτισης καταλόγων. Αν αντιμετωπίσουμε το πρόβλημα από τη σκοπιά του εμπόρου, η συσχέτιση θα πρέπει να εφαρμοστεί σε κάθε αγορά στην οποία θέλει να συμμετάσχει. Έχοντας εντοπίσει τις ομοιότητες μεταξύ των καταχωρήσεων των καταλόγων, μπορούν να εξεταστούν περαιτέρω ώστε να παράγουν επερωτήσεις που μεταφράζουν αυτόματα τα στιγμιότυπα από κατάλογο σε κατάλογο. Στη συνέχεια, έχοντας συσχετίσει τους καταλόγους, οι χρήστες της αγοράς έχουν μια ενοποιημένη πρόσβαση στα προς πώληση προϊόντα. Το σενάριο που περιγράφηκε παραπάνω, συμπεριλαμβανομένων των εμπόρων και των αγορών, μπορεί να θεωρηθεί ως χαρακτηριστικό παράδειγμα ενοποίησης τοπικών πηγών πληροφορίας σε μια αποθήκη δεδομένων [40].



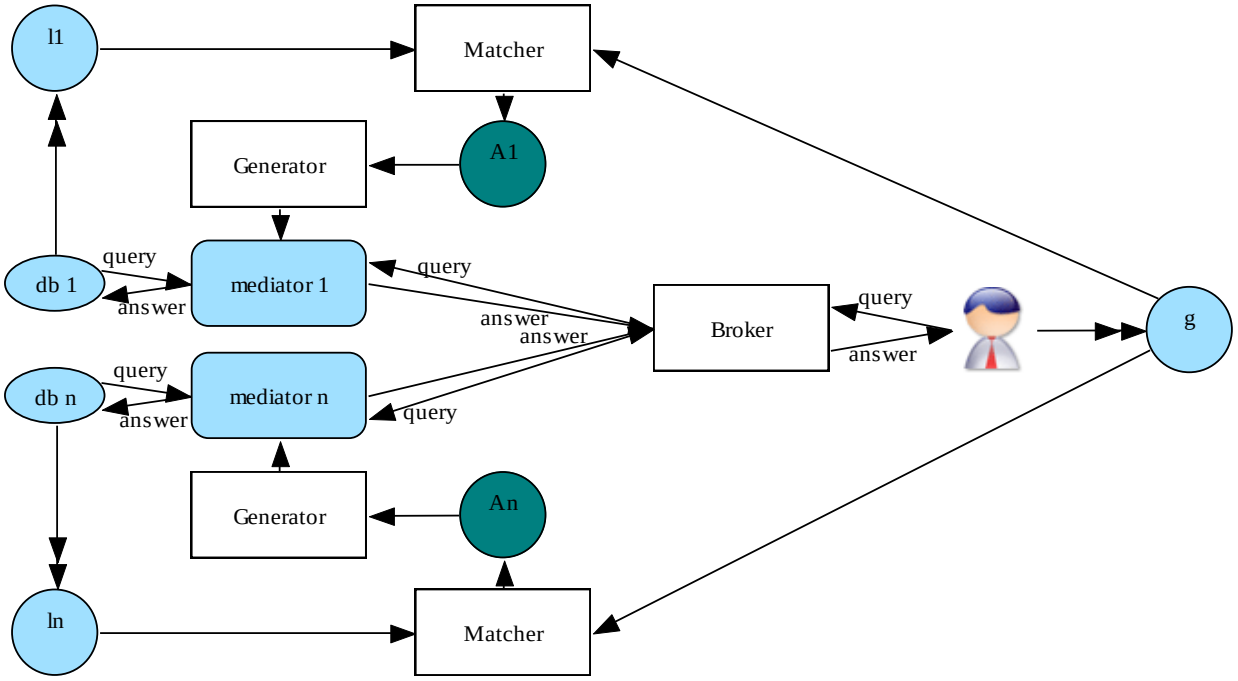
Εικόνα 2.13 Σενάριο ενοποίησης καταλόγων με συσχέτιση. Κάθε έμπορος συσχετίζει τον κατάλογο του (s_i) με έναν από τους καταλόγους του marketplace (s). Από το αποτέλεσμα που προκύπτει (A_i) παράγεται ένα πρόγραμμα μετάφρασης δεδομένων ($translator_i$) το οποίο χρησιμοποιείται για να φορτωθεί ο κατάλογος (cat_i) στο marketplace. Οι χρήστες μπορούν να κάνουν επερωτήσεις στο marketplace και να λαμβάνουν απαντήσεις με βάση τον ενοποιημένο κατάλογο.

Ενοποίηση δεδομένων

Η ενοποίηση των δεδομένων αποτελεί μια προσέγγιση της ενοποίησης της πληροφορίας που προέρχεται από πολλαπλές τοπικές πηγές και πραγματοποιείται χωρίς να προαπαιτείται η ύπαρξη των δεδομένων σε μια κεντρική αποθήκη. Αυτό επιτρέπει τη διαλειτουργικότητα ανάμεσα σε πολλαπλές τοπικές πηγές, έχοντας πρόσβαση στα ενημερωμένα δεδομένα. Στο σενάριο που παρουσιάστηκε παραπάνω και περιέγραφε την ενοποίηση καταλόγων, οι έμποροι καλούνται να ενημερώνουν την κεντρική αποθήκη της αγοράς. Στο τρέχον σενάριο, το σύστημα ενοποίησης δεδομένων παρέχει αυτή τη λειτουργία.

Το σενάριο παρουσιάζεται στην Εικόνα 2.14 και έχει ως εξής: Αρχικά, εντοπίζονται οι τοπικές πηγές πληροφορίας που πρόκειται να συμμετάσχουν στην ενοποίηση, πχ. βιβλιοπωλεία, βιβλιοθήκες, μουσεία. Μετά, δημιουργείται μια εικονική κοινή οντολογία. Τίθενται επερωτήσεις πάνω στην εικονική κοινή οντολογία που στη

συνέχεια μετατρέπονται σε επερωτήσεις πάνω στις τοπικές πηγές πληροφορίας, πχ. σε εφαρμογές πολιτισμικής κληρονομιάς, θα μπορούσε να είναι μουσεία, όπως το Iconclass[9] και το Rijksmuseum[10]. Οι ομοιότητες ανάμεσα στις σημασιολογικά σχετιζόμενες οντότητες των τοπικών πηγών και στην εικονική κοινή οντολογία θα πρέπει να οριστούν, έτσι ώστε να είναι εφικτή η απάντηση των επερωτήσεων. Ο ορισμός αυτός των ομοιοτήτων είναι γνωστός ως συσχέτιση.



Εικόνα 2.14 Σενάριο ενοποίησης δεδομένων με συσχέτιση. Ανάλογα με το αν το καθολικό σχήμα (g) είναι συσχετισμένο με τα υπάρχοντα τοπικά σχήματα (l_i) ή ανάποδα, πρόκειται για την GAV προσέγγιση ή την LAV αντίστοιχα. Συνήθως, η φάση της συσχέτισης έχει ως αποτέλεσμα τις ευθυγραμμίσεις (A_i) και παράγουν τους διαμεσολαβητές ($mediator_i$) για κάθε τοπική βάση δεδομένων. Η επερώτηση αποστέλεται στο Broker που καλεί τους κατάλληλους διαμεσολαβητές. Αυτοί μεταφράζουν την επερώτηση, την αποτιμούν σε κάθε βάση και μεταφράζουν την απάντηση πριν την επιστρέψουν.

Η απάντηση των επερωτήσεων γίνεται χρησιμοποιώντας τις συσχετίσεις μέσω των τεχνικών Local-as-View (LAV), Global-as-View (GAV), ή Global-Local-as-View (GLAV) [41]. Σύμφωνα με τη LAV προσέγγιση, τα τοπικά σχήματα ορίζονται με

9 http://www.unspsc.org
10 http://www.eclass.de

βάση ένα καθολικό σχήμα, δηλαδή, η συσχέτιση καθορίζεται από τη δημιουργία μιας εικόνας του εκάστοτε τοπικού σχήματος πάνω στο καθολικό σχήμα. Η επεξεργασία των επερωτήσεων γίνεται μέσω ενός μηχανισμού εξαγωγής συμπερασμάτων που μεταφράζει τα δομικά στοιχεία του καθολικού σχήματος στα ισοδύναμα των τοπικών σχημάτων. Στην GAV προσέγγιση, ένα καθολικό σχήμα ορίζεται με βάση τα τοπικά σχήματα, δηλαδή, η συσχέτιση καθορίζεται δίνοντας έναν ορισμό για κάθε καθολικό σχήμα που λειτουργεί ως εικόνα πάνω στο τοπικό σχήμα. Η επεξεργασία των επερωτήσεων γίνεται σταδιακά, δηλαδή, επεκτείνοντας τα δομικά στοιχεία με βάση τον ορισμό τους (έτσι ώστε να καταρτιστούν οι σχέσεις με τα τοπικά σχήματα). Η GLAV προσέγγιση αποτελεί μια μικτή προσέγγιση. Είναι μια παραλλαγή της LAV προσέγγισης που επιτρέπει οποιαδήποτε επερώτηση πάνω στα τοπικά σχήματα.

Τέλος, σύμφωνα με το [41], ο πυρήνας των εφαρμογών αυτών είναι ο ορισμός των συσχετίσεων, δηλαδή, η διαδικασία συσχέτισης. Τα αποτελέσματα της συσχέτισης μπορεί να χρησιμοποιηθούν εκτός από τη δημιουργία των καθολικών (ή αντίστοιχα των τοπικών) εικόνων, αλλά και για τη συντήρηση τους καθώς τα σχήματα εξελίσσονται.

2.2.3 Διαμοιρασμός Πληροφορίας Διομότιμων Συστημάτων

Τα Διομότιμα Συστήματα (Peer-to-Peer, P2P) είναι ένα κατανεμημένο μοντέλο επικοινωνίας που τα ομότιμα τμήματα τους (peers) έχουν ισοδύναμη λειτουργικότητα και παρέχουν μεταξύ τους δεδομένα και υπηρεσίες [42]. Τα Διομότιμα δίκτυα έγιναν δημοφιλή μέσω της εφαρμογής τους για το διαμοιρασμό αρχείων, πχ. εικόνων, μουσικής, βίντεο, βιβλίων. Υπάρχουν αρκετά ευρέως διαδεδομένα P2P συστήματα διαμοιρασμού αρχείων, όπως Kazaa, Edonkey και BitTorrent. Αυτές οι εφαρμογές περιγράφουν το περιεχόμενο των αρχείων μέσω ενός απλού σχήματος (σύνολο από χαρακτηριστικά και ιδιότητες, όπως *τίτλος τραγουδιού, συγγραφέας*, κλπ.) στο οποίο όλα τα μέλη του δικτύου θα πρέπει να είναι συνδρομητές. Τα σχήματα αυτά δεν είναι δυνατό να υποστούν επεξεργασία τοπικά από ένα μέλος του δικτύου. Συνεπώς, στα συστήματα που περιγράφηκαν παραπάνω, το πρόβλημα της σημασιολογικής ετερογένειας (στο επίπεδο του σχήματος) δεν υπάρχει στη φάση της δημιουργίας. Η χρήση ενός μοναδικού σχήματος του συστήματος, παραβιάζει την αυτονομία των ομότιμων τμημάτων του δικτύου. Παρά το γεγονός ότι αρκετά αυτόνομα διομότιμα συστήματα επιτρέπουν

στα ομότιμα τμήματα τους να συνδέονται και να αποσυνδέονται από το δίκτυο ανά πάσα χρονική στιγμή, κι έτσι σέβονται κατά κάποιο τρόπο την αυτονομία τους, εξακολουθούν να περιορίζουν τη σχεδιαστική αυτονομία τους, σε θέματα όπως η περιγραφή των δεδομένων και τι περιορισμοί τίθενται σε αυτά [42].

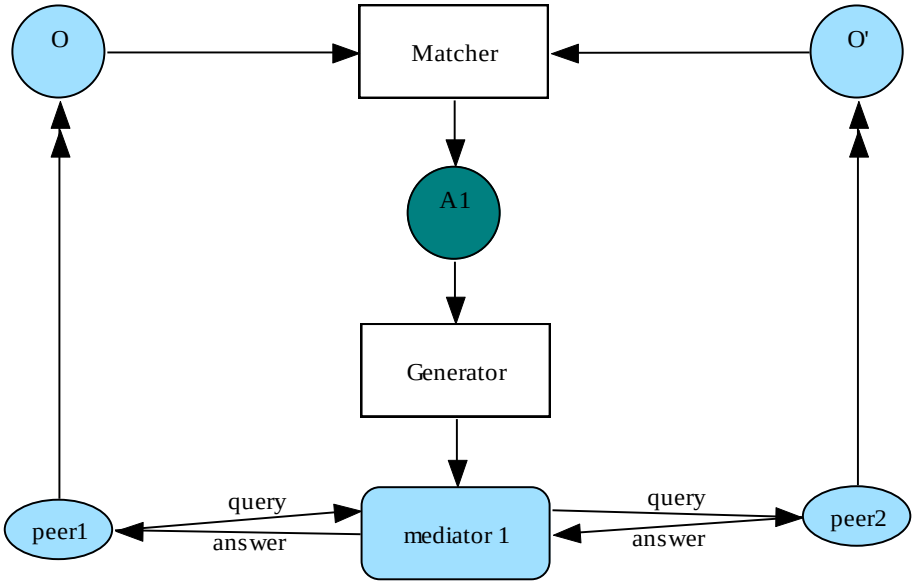
Αν τα ομότιμα τμήματα λειτουργούν εντελώς αυτόνομα, τότε είναι πιθανό να χρησιμοποιούνται διαφορετικές ορολογίες και μοντέλα για την αναπαράσταση των δεδομένων τους, ακόμα και αν αναφέρονται στον ίδιο τομέα. Έτσι, ένα από τα βήματα που πρέπει να εκτελεστεί με σκοπό την πραγματοποίηση την ανταλλαγή πληροφορίας μεταξύ των ομότιμων τμημάτων, είναι ο εντοπισμός σχέσεων μεταξύ των οντολογιών τους. Πρόκειται για τη διαδικασία συσχέτισης. Έχοντας εντοπίσει τις σχέσεις μεταξύ των οντολογιών, μπορούν να αξιοποιηθούν για την απάντηση των επερωτήσεων, πχ. χρησιμοποιώντας τεχνικές που εφαρμόζονται σε συστήματα ενοποίησης δεδομένων.

Σημασιολογικά διομότιμα συστήματα

Τα σημασιολογικά διομότιμα συστήματα [43], χρησιμοποιούν περισσότερο σύνθετες προδιαγραφές για τα περιεχόμενα τους, όπως σχήματα βάσεων δεδομένων [44], ή οντολογίες [45], σε αντίθεση με τα κλασικά διομότιμα συστήματα που περιγράφηκαν παραπάνω. Ο βασικός πυρήνας πίσω από αυτό είναι η βελτίωση της ακρίβειας της αναζήτησης, παρέχοντας μια λεπτομερή περιγραφή των αντικειμένων. Για παράδειγμα, οι χρήστες που επιθυμούν να μοιραστούν κάποια βιβλία τους με τους φίλους τους, μπορούν να τα ευρετηριοποιήσουν με βάση το συγγραφέα, το θέμα και το έτος δημοσίευσης. Η κατηγοριοποίηση αυτή μπορεί να αξιοποιήσει περιγραφές των οντολογιών, πχ. για την ανάκτηση βιβλίων μαθηματικών συγγραφέων από το Cambridge πριν το 1920, σε αντίθεση με βιβλία του Bertrand Russell το τομέα της λογικής του 1908. Για παράδειγμα, το σύστημα BibSter [46], χρησιμοποιεί μια βιβλιογραφική οντολογία εκφρασμένη σε RDF. Συστήματα όπως το BibSter, ακολουθούν την προσέγγιση της μοναδικής οντολογίας, με αποτέλεσμα να περιορίζεται η αυτονομία των ομότιμων τμημάτων και έτσι το πρόβλημα της σημασιολογικής ετερογένειας στο επίπεδο σχήματος δεν υπάρχει στη φάση της δημιουργίας.

Τα περισσότερο λεπτομερή σημασιολογικά διομότιμα συστήματα μειώνουν τις απαιτήσεις ομογένειας των κλασικών διομότιμων συστημάτων, επιτρέποντας στα

ομότιμα τμήματα να χρησιμοποιούν ανεξάρτητα σχήματα και οντολογίες, Εικόνα 2.15.



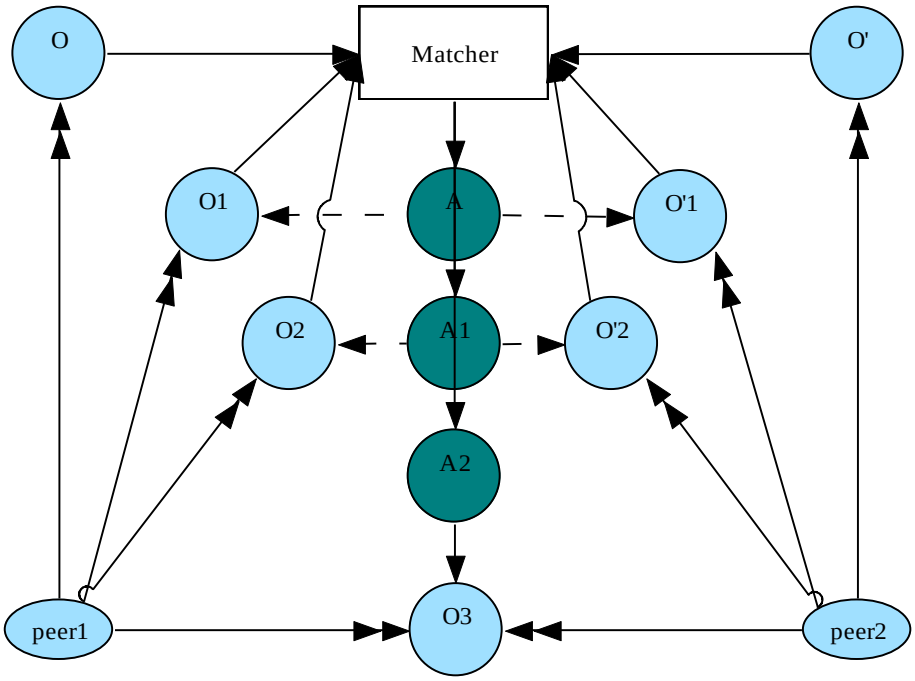
Εικόνα 2.15 Απάντηση P2P επερωτήσεων. Στο σενάριο αυτό είναι χρήσιμο να: (1) συσχετίζονται σχετικά τμήματα των οντολογιών ο και ο', έτσι ώστε να προκύπτει η ευθυγράμμιση A, (2) παράγεται ένας διαμεσολαβητής (mediator) μεταξύ των $peer_1$ και $peer_2$ για τη μετάφραση των επερωτήσεων και μερικές φορές για τη μετάφραση των απαντήσεων.

Τέτοιες εφαρμογές ενέχουν επιπρόσθετες απαιτήσεις από τις προτάσεις συσχέτισης. Στις ρυθμίσεις των διομότιμων συστημάτων που σέβονται τη συνολική αυτονομία των ομότιμων τμημάτων, δεν μπορεί να διατυπωθεί μια υπόθεση ότι όλα τα ομότιμα τμήματα ικανοποιούν ένα καθολικό σχήμα, όπως στην ενοποίηση δεδομένων, επειδή το καθολικό σχήμα θα πρέπει να ενημερώνεται κάθε φορά που το σύστημα εξελίσσεται [47]. Ενώ στην περίπτωση της ενοποίησης δεδομένων η συσχέτιση σχημάτων μπορεί να πραγματοποιηθεί κατά τη διάρκεια της σχεδίασης, στις εφαρμογές διομότιμων συστημάτων τα ομότιμα τμήματα πρέπει να συντονίζονται με τις βάσεις δεδομένων δυναμικά, έτσι η συσχέτιση σχημάτων επιβάλλεται κατά το χρόνο εκτέλεσης. Επίσης, εάν η εφαρμογή αποδέχεται ελλιπείς ή προσεγγιστικές απαντήσεις, τότε οι απαντήσεις αυτές ικανοποιούν και τις παραπάνω ρυθμίσεις. Αυτό συμβαίνει επειδή κάποιες συσχετίσεις που λαμβάνονται υπόψη για την απάντηση επερωτήσεων, μπορεί να μην είναι διαθέσιμες προσωρινά ή να μην είναι έγκυρες [48].

Μερικά παραδείγματα σεναρίων χρήσης διομότιμων συστημάτων που βασίζονται σε διαφορετικά μοντέλα αναπαράστασης δεδομένων, όπως σχήματα σχεσιακών βάσεων, XML σχήματα, RDF σχήματα ή OWL οντολογίες, περιγράφονται στα [44, 42, 49, 50, 45]. Για παράδειγμα, εφαρμογές όπως το SomeWhere [45] ενοποιεί ομότιμες βάσεις δεδομένων και τις συνδέει μέσω συσχετίσεων εκφρασμένων σε προτάσεις Horn από τη μια βάση δεδομένων στην άλλη. Όταν ένα ομότιμο τμήμα θέτει μια επερώτηση, το σύστημα υπολογίζει όλες τις πιθανές επεκτάσεις της επερώτησεις όσο αφορά τις συσχετίσεις, δηλαδή ακολουθείται η LAV προσέγγιση [41]. Στη συνέχεια, το σύστημα στέλνει σε κάθε σχετιζόμενο ομότιμο τμήμα τις επερωτήσεις που μπορεί να βοηθήσουν στην απάντηση της αρχικής επερώτησης και συγκεντρώνονται ομαδοποιούνται οι απαντήσεις. Η προσέγγιση αυτή προϋποθέτει ότι το σχήμα της βάσης δεδομένων των peers έχουν συσχετιστεί εκ των προτέρων και εκτός δικτύου. Έτσι, μόνο η απάντηση της επερώτησης αξιοποιεί τη δυναμική του διομότιμου περιβάλλοντος.

Αναδυόμενη σημασιολογία ανάμεσα σε ομότιμα συστήματα

Αναδυόμενη σημασιολογία [51, 52], είναι η διαδικασία κατά την οποία ένα σύνολο από ομότιμα συστήματα σταδιακά συγκλίνει σε μια συναινετική οντολογία μέσω συνεχούς αλληλεπίδρασης και διαπραγμάτευσης της ερμηνείας των όρων. Η διαδικασία αυτή μιμείται σε κάποιο βαθμό την ανθρώπινη κοινωνία που μπορεί μεν να μην ολοκληρώνεται ποτέ αλλά σταδιακά βελτιώνει την κατανόηση των πραγμάτων. Δεδομένου ότι η ομοφωνία αυτή δημιουργείται σταδιακά, και επειδή προκύπτει από διαφορετικές από σημείο σε σημείο συμβάσεις μεταξύ των ομότιμων συστημάτων, μια ευθυγράμμιση (alignment) ανάμεσα στις οντολογίες των ομότιμων συστημάτων θεωρείται ως ένας πρακτικός τρόπος για την καθιέρωση των συμβάσεων αυτών. Έτσι τα ομότιμα συστήματα θα πρέπει συνεχώς να ενημερώνουν τις σχέσεις ανάμεσα στις οντολογίες τους. Οι ενημερώσεις αυτές επιτυγχάνονται μέσω της διαδικασίας συσχέτισης. Η διαδικασία της αναδυόμενης σημασιολογίας ανάμεσα σε δύο ομότιμα συστήματα περιγράφεται στην Εικόνα 2.16.



Εικόνα 2.16 Διομότιμα συστήματα και αναδυόμενη σημασιολογία: μετά την πρώτη συσχέτιση μεταξύ των οντολογιών O και O′, η ευθυγράμμιση A που προκύπτει έχει ως αποτέλεσμα τα peers (διακεκομμένες γραμμές) να αναπτύσσουν τις οντολογίες τους στις O_1 και O'_1 αντίστοιχα. Με τη σειρά τους, αυτές οι οντολογίες μπορεί να συσχετιστούν ξανά και να προκύψει η ευθυγράμμιση A_1 κ.ο.κ. Τελικά, τα peers μπορεί να συγκλίνουν σε μια κοινή οντολογία (O_3).

Η συνεχόμενη διαδικασία συσχέτισης των οντολογιών έχει ως αποτέλεσμα την αντιπαράθεση και την αναθεώρηση των ίδιων των οντολογιών. Στην πραγματικότητα, οι χρήστες μπορούν να δημιουργήσουν περισσότερο ομόφωνες/συναινετικές οντολογίες μέσω της αντιπαράθεσης αυτής [53]. Υπάρχουν εναλλακτικοί τρόποι με τους οποίους η ευθυγράμμιση μπορεί να λειτουργήσει εποικοδομητικά:

- Οι ευθυγραμμίσεις παρέχουν τη βάση από την οποία μπορεί να ξεκινήσει η διαπραγμάτευση μεταξύ των ομότιμων συστημάτων (όπως στα πρωτόκολλα πρακτόρων για την τεκμηρίωση των αντιστοιχιών, που θα εξετάσουμε παρακάτω)
- Οι αλγόριθμοι συσχέτισης συχνά μπορούν να υπολογίζουν την απόσταση μεταξύ των οντολογιών. Αυτή η πληροφορία είναι χρήσιμη όταν, για παράδειγμα, ένα ομότιμο σύστημα αναζητά την «πλησιέστερη» οντολογία.

- Δημιουργώντας ένα δίκτυο οντολογιών μαζί με τις ευθυγραμμίσεις μεταξύ τους και αξιοποιώντας την απόσταση, με τη βοήθεια τεχνικών ανάλυσης κοινωνικών δικτύων (social networks), είναι δυνατό να καθοριστεί ο βαθμός εγγύτητας ανάμεσα σε χρήστες ή πράκτορες. Αυτό έχει ως αποτέλεσμα τη διευκόλυνση της διαδικασίας απάντησης επερωτήσεων.

Τέτοιου είδους αποτελέσματα, λειτουργούν επικουρικά για χρήστες και κοινότητες για την καθιέρωση των οντολογιών τους, μέσω της σταδιακής πρότασης συσχετίσεων μεταξύ διαφορετικών αναπαραστάσεων του ίδιου τομέα και μέσω του ορισμού της περισσότερο κεντρικής οντολογίας (σύμφωνα με την ορολογία των κοινωνικών δικτύων) για τον εκάστοτε τομέα ενδιαφέροντος.

2.2.4 Σύνθεση Ηλεκτρονικών Υπηρεσιών

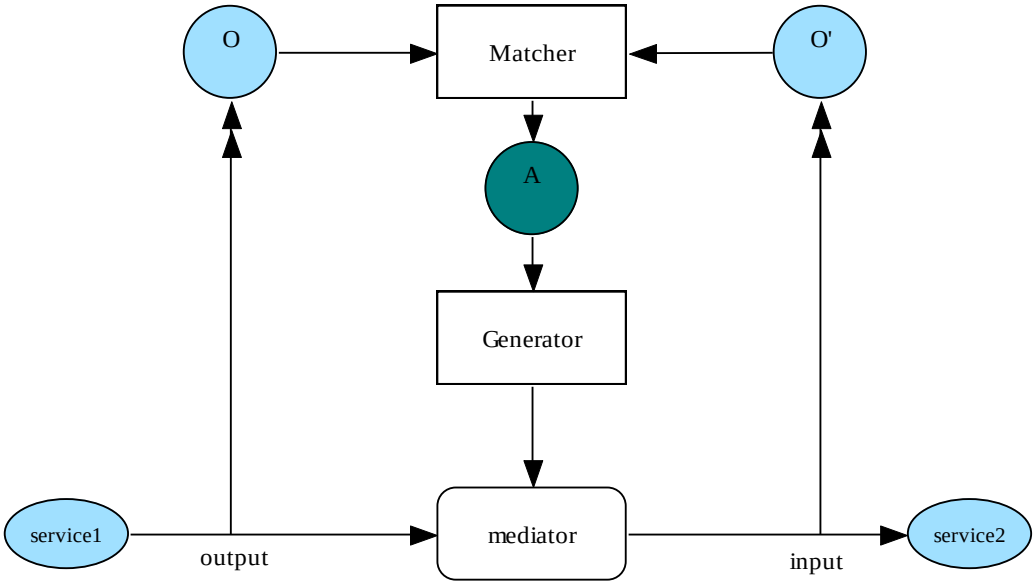
Οι ηλεκτρονικές υπηρεσίες είναι διεργασίες που εκθέτουν τις διεπαφές τους στο Διαδίκτυο έτσι ώστε οι χρήστες να μπορούν να τις συμπεριλάβουν για ιδία χρήση. Οι σημασιολογικές ηλεκτρονικές υπηρεσίες παρέχουν έναν πλουσιότερο και περισσότερο ακριβή τρόπο για την περιγραφή των υπηρεσιών αυτών μέσω των γλωσσών αναπαράστασης γνώσης και των οντολογιών [54]. Η ανακάλυψη και η ενοποίηση ηλεκτρονικών υπηρεσιών είναι η διαδικασία της εύρεσης μιας ηλεκτρονικής υπηρεσίας ικανής να παρέχει μια συγκεκριμένη υπηρεσία και να συνθέτει διαφορετικές υπηρεσίες με σκοπό να επιτυγχάνεται ένας συγκεκριμένος στόχος [55, 56, 57].

Οι ηλεκτρονικές υπηρεσίες έχουν σχεδιαστεί έτσι ώστε να είναι ανεξάρτητες και να αντικαθίστανται. Έτσι, οι επεξεργαστές ηλεκτρονικών υπηρεσιών είναι ικανοί να ενσωματώνουν νέες υπηρεσίες στη ροή εργασίας τους και οι πελάτες μπορούν, δυναμικά, να επιλέξουν νέες και πολλά υποσχόμενες υπηρεσίες. Για αυτό το λόγο, πρέπει να είναι ικανοί να συγκρίνουν τις περιγραφές των υπηρεσιών (με σκοπό να γνωρίζουν αν είναι πράγματι σχετικές) και τη διαδρομή της γνώσης που επεξεργάζονται, με σκοπό τη σύνθεση διαφορετικών υπηρεσιών καθοδηγώντας την έξοδο μιας υπηρεσίας στην είσοδο μιας άλλης.

Όμως, στη περίπτωση των σημασιολογικών ηλεκτρονικών υπηρεσιών, που μπορούν να περιγραφούν με βάση τις οντολογίες, η επιβολή μιας κοινής κεντρικής οντολογίας (όπως στα διομότιμα συστήματα που χρησιμοποιούν μια μοναδική οντολογία), δεν έχει ρεαλιστική εφαρμογή και θα μπορούσε να περιορίσει την

εξέλιξη τέτοιων υπηρεσιών. Εφεξής, ένας διαμεσολαβητής δεδομένων (data mediator), λειτουργεί ως τη γέφυρα ανάμεσα στα διαφορετικά λεξιλόγια τόσο για την εύρεση της κατάλληλης υπηρεσίας όσο και για τη διεπαφή των υπηρεσιών [58, 59]. Οι διαμεσολαβητές πρέπει να είναι ικανοί να μεταφράζουν την έξοδο της πρώτης υπηρεσίας σε κατάλληλη είσοδο για τη δεύτερη υπηρεσία, με βάση τις αντιστοιχίες μεταξύ των όρων των περιγραφών, όπως στην Εικόνα 2.17.

Έτσι, ο πυρήνας ενός διαμεσολαβητή είναι η ευθυγράμμιση μεταξύ δύο οντολογιών. Αυτό μπορεί να παρέχεται μέσω της συσχέτισης των οντολογιών, είτε εκτός δικτύου, όταν κάποιος σχεδιάζει μια προκαταρκτική σύνθεση υπηρεσιών, είτε δυναμικά (εντός δικτύου) [60, 61], όταν εντοπίζονται νέες υπηρεσίες που ικανοποιούν ένα αίτημα.



Εικόνα 2.17 Σύνθεση ηλεκτρονικών υπηρεσιών. Στο σενάριο αυτό είναι χρήσιμο να: (1) συσχετίζονται σχετικά τμήματα των οντολογιών O και O', έτσι ώστε να προκύπτει η ευθυγράμμιση A, (2) παράγεται ένας διαμεσολαβητής μεταξύ των $service_1$ και $service_2$ με σκοπό να είναι εφικτή η μετατροπή των πραγματικών δεδομένων.

Για παράδειγμα, ας υποθέσουμε ότι μια υπηρεσία διαδικτυακής βιβλιοθήκης, παρέχει την περιγραφή της εξόδου της με κάποια οντολογία και μια υπηρεσία πώλησης οικοπέδων χρησιμοποιεί μια δεύτερη οντολογία για την περιγραφή της εισόδου της. Η συσχέτιση αυτών των οντολογιών είναι λειτουργική για τους εξής λόγους:

- Έλεγχος εάν ό,τι προκύπτει από την πρώτη υπηρεσία, πχ. ένα Βιβλίο, συσχετίζεται με αυτό που περιμένει ως είσοδο η δεύτερη υπηρεσία, πχ. ένα Αντικείμενο
- Επαλήθευση των προϋποθέσεων της δεύτερης υπηρεσίας, πχ. το size είναι σε εκατοστά ενώ το dimensions σε ίντσες, και
- Δημιουργία ενός διαμεσολαβητή ικανού να μετατρέψει την έξοδο της πρώτης υπηρεσίας με σκοπό να λειτουργεί ως είσοδος της δεύτερης.

2.2.5 Αυτόνομα Συστήματα Επικοινωνίας

Άλλα είδη εφαρμογών περιλαμβάνουν αυτόνομες οντότητες που μπορούν να συνυπάρχουν σε ένα δίκτυο και έχουν σχεδιαστεί ανεξάρτητα. Όταν αυτές οι οντότητες είναι προγράμματα λογισμικού, τότε μπορούν να θεωρηθούν ως πράκτορες για ένα μεγάλο χρονικό διάστημα. Όμως, αν είναι ένας συνδυασμός υλικού και λογισμικού, πρόκειται για θέμα προγραμματισμού περιβάλλοντος (ambient computing). Προφανώς, όπως έχει περιγραφεί και παραπάνω, τέτοιες οντότητες δεν είναι εφικτό να διαμοιράζονται μια κοινή οντολογία. Έτσι, στην περίπτωση που οι πράκτορες θέλουν να επικοινωνήσουν, είναι χρήσιμο να συσχετίζονται οι οντολογίες τους.

Multi-agent communication

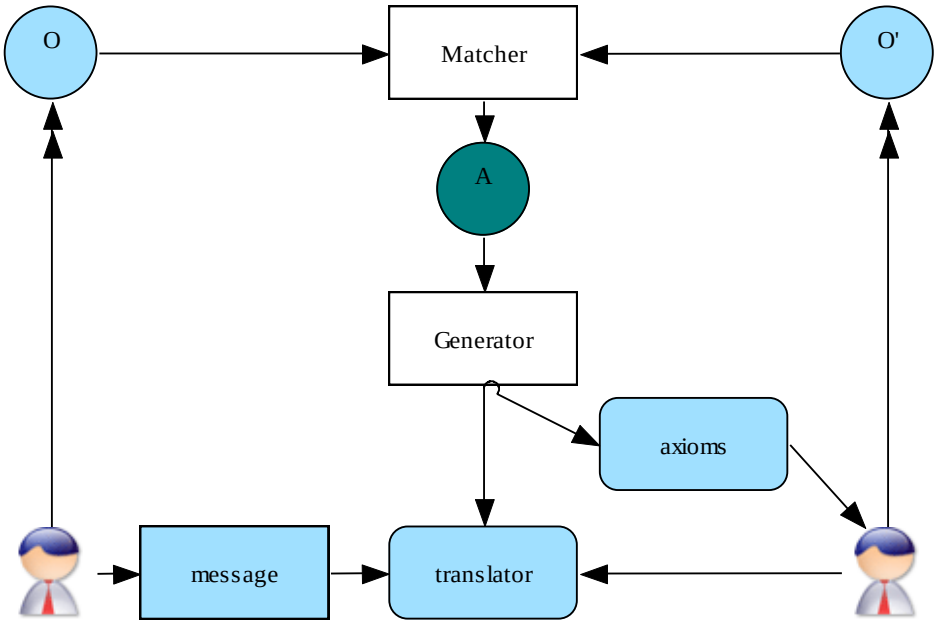
Οι πράκτορες είναι οντότητες λογισμικού που χαρακτηρίζονται από την αυτονομία τους και την ικανότητα για αλληλεπίδραση. Συνήθως διαιρούνται σε γνωστικούς (cognitive) και αντιδραστικούς (reactive) πράκτορες. Οι αντιδραστικοί πράκτορες υλοποιούν μια απλή συμπεριφορά και η ισχύς των πρακτόρων αυτών έγκειται στην ικανότητα να προκύπτει μια καθολική συμπεριφορά από την ξεχωριστή συμπεριφορά του κάθε πράκτορα. Οι γνωστικοί πράκτορες έχουν μια κάπως περισσότερο λεπτομερή συμπεριφορά που συχνά χαρακτηρίζεται ως η ικανότητα να επιδιώκει σκοπούς, να σχεδιάζει τις ενέργειες και να διαπραγματεύεται με τους άλλους πράκτορες με σκοπό την επίτευξη των πρωταρχικών στόχων.

Οι πράκτορες επικοινωνούν ανταλλάσσοντας μηνύματα διατυπωμένα σε γλώσσες επικοινωνίας πρακτόρων, όπως η FIPA Agent Communication Language [62, 63]. Οι γλώσσες αυτές καθορίζουν τη μορφή του «φακέλου» των μηνυμάτων και επιτρέπουν στους πράκτορες να συμμετέχουν σε κάποιο συγκεκριμένο πλαίσιο αλληλεπίδρασης. Όμως, δεν προσδιορίζουν το πραγματικό περιεχόμενο του μηνύματος, το οποίο συχνά εκφράζεται μα βάση μια οντολογία, προσβάσιμη από

τον πράκτορα. Τα σημερινά πρότυπα για την έκφραση των μηνυμάτων παρέχουν τη δυνατότητα επιλογής και ορισμού της γλώσσας περιεχομένου και της οντολογίας που πρόκειται να χρησιμοποιηθεί.

Συνεπώς, όταν δύο αυτόνομοι και ανεξάρτητα σχεδιασμένοι πράκτορες πρόκειται να συνεργαστούν, έχουν την ευκαιρία να ανταλλάξουν μηνύματα αλλά είναι μικρή η πιθανότητα να καταλάβουν ο ένας τον άλλο αν δε χρησιμοποιούν την ίδια γλώσσα περιεχομένου και την ίδια οντολογία. Έτσι είναι χρήσιμο να υποβοηθηθούν οι πράκτορες με τη συσχέτιση των οντολογιών τους με σκοπό τη μετάφραση των μηνυμάτων τους ή τη δημιουργία αξιωμάτων γεφύρωσης (bridge axioms, BA) στα μοντέλα τους. Έχουν γίνει αρκετές προσεγγίσεις με σκοπό την αξιολόγηση των αντιστοιχιών ανάμεσα στους όρους των οντολογιών [64, 65, 66, 67, 68].

Οι πράκτορες αντιμετωπίζουν ετερογενείς οντολογίες και πρέπει να εντοπίσουν τις αντιστοιχίες μεταξύ των οντολογιών αυτών με σκοπό την κατανόηση των μηνυμάτων που ανταλλάσσονται. Η διαδικασία της συσχέτισης των οντολογιών μπορεί να γίνει από τους ίδιους τους πράκτορες ή αξιοποιώντας βιβλιοθήκες ευθυγραμμίσεων ή υπηρεσίες συσχέτισης. Μόλις οριστεί μια ευθυγράμμιση, οι πράκτορες μπορούν να προχωρήσουν στη φάση διαπραγμάτευσης [69], στην οποία ανταλλάσσουν επιχειρήματα υπέρ ή κατά των συσχετίσεων. Όταν συναινέσουν σε μια αντιστοίχηση, μπορούν να τη μετατρέψουν σε ένα πρόγραμμα που μεταφράζει τα μηνύματα που ανταλλάσσονται ή σε αξιώματα που, αφού εξελιχθούν σε γνώση του πράκτορα, επιτρέπουν τη μετάφραση μηνυμάτων, όπως στην Εικόνα 2.18.



Εικόνα 2.18 Επικοινωνία πρακτόρων. Στο σενάριο αυτό είναι χρήσιμο να: (1) συσχετίζονται σχετικά τμήματα των οντολογιών O και O′ που χρησιμοποιούνται από κάθε πράκτορα και προκύπτει η ευθυγράμμιση A, (2) παράγονται αξιώματα γεφύρωσης μεταξύ των δύο οντολογιών και (3) ενσωματώνονται τα αξιώματα στην O′. Εναλλακτικά, η διαδικασία (2) μπορεί να εκτελεστεί και ως εξής, πρώτα παραγωγή ενός μηνύματος για μετάφραση από την οντολογία O στην οντολογία O′ και εφαρμογή του translator στο μήνυμα αυτό.

Πλαίσιο Συσχέτισης στον προγραμματισμό περιβάλλοντος

Στον προγραμματισμό περιβάλλοντος, οι εφαρμογές τρέχουν σε κινητές συσκευές εκμεταλλευόμενες το περιβάλλον που παρέχει υπηρεσίες στους χρήστες. Φυσιολογικά, το περιβάλλον μπορεί να υποστεί μεταβολές, πχ. όσο αφορά τη θέση του χρήστη, και οι εφαρμογές θα πρέπει πάντα να παρακολουθούν τις αλλαγές αυτές, συμπεριλαμβανομένων και των νέων συσκευών που εμφανίζονται και των αισθητήρων. Ο χαρακτηρισμός του πλαισίου του προγραμματισμού περιβάλλοντος, προκύπτει μέσω της πληροφορίας για την τρέχουσα κατάσταση στο περιβάλλον χρησιμοποιώντας διάφορες συσκευές που είναι διαθέσιμες στο περιβάλλον, πχ. αισθητήρες. Με αυτόν τον τρόπο, οι εφαρμογές παρέχουν λύσεις που λαμβάνουν υπόψη το πλαίσιο. Εάν κάποιος επιθυμεί να σχεδιάσει ευέλικτες και έξυπνες εφαρμογές, είναι χρήσιμο να αξιοποιήσει τις οντολογίες των συσκευών των οποίων οι αισθητήρες και οι δυνατότητες τους είναι διαθέσιμες στο περιβάλλον [70]. Όπως συμβαίνει και με τις περιγραφές των ηλεκτρονικών υπηρεσιών, οι οντολογίες αυτές

παρέχουν περιγραφές των συσκευών, ακόμη και αφηρημένων συσκευών, όπως μια υπηρεσία θερμοκρασίας καθώς επίσης και περιγραφών του τρόπου αλληλεπίδρασης τους.

Για άλλη μια φορά, είναι αναμενόμενο ότι οι πάροχοι των συσκευών θα αναπτύξουν διαφορετικές οντολογίες, προσαρμοσμένες στα προϊόντα τους ή επεκτάσεις κάποιων πρότυπων οντολογιών. Επιπλέον, δεδομένου ότι η εφαρμογές εξελίσσονται σε μεταβαλλόμενα περιβάλλοντα, στα οποία συσκευές μπορεί να αποτύχουν να συνεργαστούν και να εμφανιστούν καινούριες, δεν υπάρχει τρόπος όλες οι οντολογίες, που είναι σχετικές, να είναι διαθέσιμες μια συγκεκριμένη χρονική στιγμή.

Συνεπώς, οι εφαρμογές θα πρέπει να είναι εκφρασμένες στο πλαίσιο γενικών χαρακτηριστικών που συνοδεύουν το πραγματικό περιβάλλον, με σκοπό την κανονική λειτουργία σε περιπτώσεις προγραμματισμού περιβάλλοντος. Η διαδικασία αυτή συσχέτισης μπορεί να αξιοποιήσει τη συσχέτιση οντολογιών, αφού είναι πιθανό όμοιες συσκευές να χρησιμοποιούνται από όμοιες εφαρμογές. Έτσι, παρέχοντας μια υπηρεσία συνδυασμού διαφορετικών οντολογιών και αποθηκεύοντας τα αποτελέσματα που προκύπτουν από τις προηγούμενες αλληλεπιδράσεις, υποβοηθούνται αυτές οι εφαρμογές στο διαμοιρασμό και στην επαναχρησιμοποίηση των έγκυρων ευθυγραμμίσεων.

2.2.6 Πλοήγηση και Επερωτήσεις στον Ιστό

Το τμήμα αυτό της εργασίας παρουσιάζει διάφορες εφαρμογές οι οποίες επεκτείνουν τον ιστό στο σημασιολογικό ιστό χρησιμοποιώντας πηγές, όπως οντολογίες. Δεδομένου ότι αυτές οι εφαρμογές λειτουργούν σε ανοιχτά περιβάλλοντα, συχνά απαιτείται συσχέτιση. Συγκεκριμένα, οι εφαρμογές που εξετάζονται παρακάτω περιλαμβάνουν: πλοήγηση στο σημασιολογικό ιστό, απάντηση επερωτήσεων στον ιστό και απάντηση επερωτήσεων στον αόρατο (deep) ιστό.

Πλοήγηση στο σημασιολογικό ιστό

Φυλλομετρητές όπως ο Magpie [71, 72] είναι σχεδιασμένοι έτσι ώστε να αξιοποιούν σημασιολογικές επισημάνσεις (annotations) που σχετίζονται με ιστοσελίδες. Για παράδειγμα, ο Magpie μπορεί να εντοπίσει στιγμιότυπα μιας οντολογίας σε μια ιστοσελίδα, να αναπαραστήσει τα στιγμιότυπα αυτά (χρησιμοποιώντας

διαφορετικά χρώματα για διαφορετικές κλάσεις) και να προσθέσει υπηρεσίες που να συνδέονται με τα στιγμιότυπα της ιστοσελίδας.

Στην ανοιχτή περιήγηση του ιστού (web browsing), το κλειδί είναι να είναι εφικτή η επιλογή των κατάλληλων οντολογιών για το δεδομένο πλαίσιο περιήγησης, σε χρόνο εκτέλεσης. Πράγματι, οι ιστοσελίδες είναι συνδεδεμένες με άλλες ιστοσελίδες που το περιεχόμενο τους διαφέρει σημαντικά από αυτό της πηγαίας σελίδας. Για τη βελτίωση της εξοικείωσης του χρήστη, είναι απαραίτητο να ληφθούν υπόψη νέες οντολογίες δυναμικά και να είναι δυνατή η σύνδεση τους με τις παλαιότερες. Έτσι, η συσχέτιση των οντολογιών είναι απαραίτητη για τη συσχέτιση μεταξύ των όρων που περιγράφουν το θέμα της ιστοσελίδας και των σχετικών οντολογιών.

Παρακάτω περιγράφεται ένα παράδειγμα [73] σχετικά με ταξίδια σε εξωτικά μέρη και ομιλίες.

"Τον Απρίλιο και Μάιο του 2005, ο τυχοδιώκτης Lorenzo Gariano συμμετείχε σε μια δεκαμελή αποστολή μεταξύ του 7summits.com και του 7summits club από τη Ρωσία, υπό την εποπτεία των Alex Abramov και Harry Kikstra για το North Face του Έβερεστ. Απόψε θα παρουσιάσει μια ομιλία για τις εμπειρίες του καθώς επίσης και μερικές από τις εντυπωσιακές φωτογραφίες που τράβηξε."

Θα πρέπει να επιλεγεί μια οντολογία που να περιγράφει έννοιες όπως τυχοδιώκτης, αποστολή, ομιλία, φωτογραφία. Αυτό προϋποθέτει ότι οι έννοιες που αναφέρθηκαν προηγουμένως συσχετίζονται με τις έννοιες που υπάρχουν στις διαθέσιμες οντολογίες. Επιπροσθέτως, στην περίπτωση που κάποια από τις έννοιες δεν υπάρχει στην οντολογία, αντιστοιχίσεις με περισσότερο ή λιγότερο γενικές έννοιες είναι αποδεκτές. Εν τέλει, δεν είναι απαραίτητο να συσχετιστούν όλες οι οντότητες της οντολογίας. Αρκεί να ληφθούν υπόψη μόνο οι οντότητες που παρουσιάζουν ομοιότητα με τους όρους που υπάρχουν στην ιστοσελίδα.

Απάντηση επερωτήσεων στον ιστό

Σε αντίθεση με το σενάριο που παρουσιάστηκε παραπάνω, στο τμήμα της ενοποίησης της πληροφορίας, η πληροφορία στον ιστό δεν περιγράφεται με ένα καθολικό σχήμα πάνω στο οποίο διατυπώνονται οι επερωτήσεις. Επιπλέον, οι χρήστες είναι συνηθισμένοι να διατυπώνουν επερωτήσεις στον ιστό χρησιμοποιώντας τη δική τους ορολογία. Έτσι, ένα σύστημα απάντησης

σημασιολογικών επερωτήσεων στον ιστό, θα πρέπει να μετατρέψει την επερώτηση με βάση τις διαθέσιμες οντολογίες για να τις αξιοποιήσει στην παραγωγή των απαντήσεων.

Για παράδειγμα, ένα σύστημα απάντησης επερωτήσεων όπως το AquaLog [74] διαθέτει μια οντολογία για την ακαδημαϊκή κοινότητα, που δημοσιοποιήθηκε για την περιγραφή της γνώσης που σχετίζεται με κάποιο πανεπιστήμιο [73]. Για την απάντηση μιας επερώτησης όπως: «Ποιες εργασίες σχετίζονται με ερευνητές που ασχολούνται με οντολογίες;», το AquaLog μεταφράζει την επερώτηση σε όρους των οντοτήτων που είναι διαθέσιμες στην οντολογία του συστήματος. Αρχικά, μετατρέπεται η επερώτηση στις εξής τριπλέτες <εργασία, σχετίζεται_με, ερευνητές> και <ερευνητές, ασχολούνται_με, οντολογίες>. Στη συνέχεια, το σύστημα επιχειρεί να συσχετίσει αυτές τις τριπλέτες με τους όρους της υποκείμενης οντολογίας. Για παράδειγμα, ο όρος *εργασίες* μπορεί να θεωρηθεί ισοδύναμος με τον όρο *Εργασία* της οντολογίας και ο όρος *οντολογίες* ισοδύναμος με το στιγμιότυπο *οντολογίες* της έννοιας *Περιοχή_Έρευνας*. Αν η *Λειτουργία* είναι υποκλάση της *Εργασίας*, το σύστημα θα μπορεί να λάβει υπόψη του λειτουργίες για την απάντηση επερωτήσεων.

Προς το παρόν, το πεδίο δράσης του AquaLog είναι περιορισμένο από το μέγεθος της γνώσης που έχει κωδικοποιηθεί στην οντολογία του συστήματος. Μια νέα έκδοση του AquaLog, γνωστή ως Power-Aqua [75], επεκτείνει τον πρόγονο του καθώς επίσης και άλλα συστήματα με παρόμοια λειτουργία όπως το Observer [76], με στόχο την ανοιχτή απάντηση επερωτήσεων. Το Power-Aqua είναι προσανατολισμένο στην επιλογή και στη συνάθροιση πληροφορίας που προκύπτει από πολλαπλές ετερογενείς οντολογίες του ιστού. Η συσχέτιση αποτελεί τον πυρήνα της επιλογής αυτής. Σε αντίθεση με το AquaLog, η συσχέτιση εφαρμόζεται μεταξύ των τριπλετών και των οντολογιών (όχι μόνο με τη μοναδική οντολογία του συστήματος). Δεν είναι απαραίτητο να συσχετιστούν όλες οι τριπλέτες της επερώτησης με μια οντολογία. Όταν δεν υπάρχει αντιστοίχηση ενός στοιχείου της τριπλέτας με όρο της οντολογίας, η χρήση γενικότερων όρων είναι αποδεκτή. Επιπλέον, δεν είναι απαραίτητη η συσχέτιση ολόκληρης της οντολογίας με την επερώτηση, αρκεί αυτή των σχετιζόμενων τμημάτων.

Απάντηση επερωτήσεων στον αόρατο (deep) ιστό

Ο λεγόμενος αόρατος ιστός, βασίζεται σε ιστοσελίδες εξερευνήσιμες μέσω διεπαφών επερωτήσεων (HTML φόρμες) δίνοντας πρόσβαση σε μια ή περισσότερες βάσεις δεδομένων. Θεωρείται ότι περιέχει πολύ περισσότερη πληροφορία [77] από τα δισεκατομμύρια των στατικών HTML σελίδων. Προς το παρόν, οι μηχανές αναζήτησης δεν είναι τόσο αποτελεσματικές στο διάβασμα ιστοσελίδων (crawling) και τον ευρετηριασμό (indexing) του αόρατου ιστού, δεδομένου ότι δεν μπορούν να χειριστούν ουσιαστικά τις διεπαφές επερωτήσεων. Για παράδειγμα, σύμφωνα με το [77], η Google και η Yahoo έχουν καταφέρει να ευρετηριοποήσουν το 32% των υπαρχόντων αντικειμένων του αόρατου ιστού. Άρα, ο αόρατος ιστός παραμένει ευρέως ανεξερεύνητος, παρά το γεγονός ότι περιέχει μεγάλο αριθμό διαδικτυακών βάσεων που θα μπορούσε να αξιοποιήσει.

Έτσι, οι χρήστες αντιμετωπίζουν δυσκολίες, αρχικά στον εντοπισμό των σχετικών πηγών και στη συνέχεια, στην επερώτηση τους. Μια κλασική περίπτωση χρήσης περιλαμβάνει την αγορά ενός βιβλίου στη χαμηλότερη τιμή μεταξύ της πληθώρας των διαθέσιμων διαδικτυακών βιβλιοπωλείων. Οι διεπαφές επερώτησης μπορεί να θεωρηθούν ως απλά σχήματα (σύνολα από όρους). Για παράδειγμα, στον τομέα πώλησης βιβλίων, η διεπαφή επερώτησης ενός διαδικτυακού βιβλιοπωλείου μπορεί να θεωρηθεί ως ένα σχήμα που αναπαριστά ένα σύνολο από όρους και χαρακτηριστικά, όπως *Συγγραφέας, Τίτλος, Θέμα, ISBN, Εκδότης*. Έτσι, για την απάντηση επερωτήσεων από πολλαπλές πηγές του αόρατου ιστού, είναι απαραίτητο να εντοπιστούν οι σημασιολογικές αντιστοιχίες μεταξύ των γνωρισμάτων των διεπαφών επερώτησης των ιστοσελίδων. Ο εντοπισμός αυτός των αντιστοιχιών είναι η διαδικασία συσχέτισης. Τελικά, οι αντιστοιχίες αυτές χρησιμοποιούνται για τη δυναμική μετάφραση μιας επερώτησης μεταξύ των διεπαφών των διαδικτυακών βάσεων δεδομένων.

Κεφάλαιο 3. Ανασκόπηση Βιβλιογραφίας

Το κεφάλαιο αυτό αποτελεί μια επισκόπηση των συστημάτων συσχέτισης που έχουν προταθεί κατά τη διάρκεια της τελευταίας δεκαετίας. Υπάρχουν ήδη διαθέσιμες συγκρίσεις των συστημάτων συσχέτισης και συγκεκριμένα οι εξής: [78, 3, 79, 80, 81, 82, 83]. Στόχος του συγκεκριμένου κεφαλαίου δεν είναι η λεπτομερής τους σύγκριση, παρά το γεγονός ότι συγκρίνονται, αλλά κυρίως η ανάδειξη της ποικιλίας τους, με απώτερο σκοπό την προβολή των διαφορετικών μεθόδων που χρησιμοποιούνται.

3.1 Επισκόπηση Συστημάτων Συσχέτισης

Η δομή του κεφαλαίου αναλύεται παρακάτω. Αρχικά, περιγράφονται συστήματα που εστιάζουν, κυρίως, σε πληροφορία επιπέδου σχήματος. Στη συνέχεια, γίνεται αναφορά σε συστήματα που επικεντρώνονται σε πληροφορία επιπέδου στιγμιοτύπου. Ακολούθως, παρουσιάζονται συστήματα που εκμεταλλεύονται τόσο πληροφορία επιπέδου σχήματος, όσο και πληροφορία επιπέδου στιγμιοτύπου. Τέλος, δίνεται μια ανασκόπηση συστημάτων μετα-συσχέτισης.

3.1.1 Συστήματα Βασισμένα στην Πληροφορία του Σχήματος

Τα συστήματα βασισμένα στο σχήμα, είναι τα συστήματα εκείνα που βασίζονται κυρίως στην πληροφορία των σχημάτων που δίνονται ως είσοδος για την εκτέλεση της συσχέτισης τους.

DELTA (The MITRE Corporation)

Το DELTA (Data Element Tool-based Analysis) είναι ένα σύστημα που εντοπίζει της αντιστοιχίες των γνωρισμάτων μεταξύ σχημάτων βάσεων δεδομένων, με ημιαυτόματο τρόπο [84]. Χειρίζεται σχεσιακά και εκτεταμένα σχήματα οντοτήτων-σχέσεων (extended entity-relationship, EER). Ο πυρήνας της προσέγγισης αυτής είναι η χρήση της κειμενικής ομοιότητας μεταξύ των ορισμών των στοιχείων με σκοπό την πρόταση υποψήφιων συσχετίσεων. Το σύστημα μετατρέπει τη διαθέσιμη πληροφορία ενός γνωρίσματος, πχ. το όνομα, τον τύπο δεδομένων, την περιγραφή, σε ένα απλό αλφαριθμητικό, που καλείται έγγραφο. Τα έγγραφα που περιγράφουν τα γνωρίσματα της κάθε βάσης, συνιστούν μια βάση εγγράφων. Μετά, το DELTA τροφοδοτεί ένα εργαλείο ανάκτησης κειμενικής πληροφορίας με τη βάση

εγγράφων του πρώτου σχήματος. Η συσχέτιση υλοποιείται με τη μορφή επερωτήσεων στο εργαλείο, βασισμένων στην πληροφορία του δεύτερου σχήματος. Η επερώτηση μπορεί να είναι ένα αλφαριθμητικό ασύνδετων φράσεων, μια λογική επερώτηση, μερικές σχετικές λέξεις, ή ένα ολόκληρο έγγραφο. Το εργαλείο υπολογίζει μια εκτίμηση της ομοιότητας (χρησιμοποιώντας ευριστικές φυσικής γλώσσας, όπως το γεγονός ότι σπάνιες ή επαναλαμβανόμενες λέξεις παρουσιάζουν αυξημένη σημαντικότητα) μεταξύ του υποδείγματος αναζήτησης και των περιεχομένων της βάσης εγγράφων. Έτσι, η συσχέτιση βασίζεται αποκλειστικά σε τεχνικές βασισμένες σε αλφαριθμητικά. Το σύστημα επιστρέφει μια κατάταξη των εγγράφων που παρουσιάζουν ομοιότητα. Η επιλογή των τελικών αποτελεσμάτων γίνεται από τους χρήστες του συστήματος.

DIKE (Universita di Reggio Calabria and Universita di Calabria)

Το DIKE (Database International Knowledge Extractor) είναι ένα σύστημα που υποστηρίζει την ημιαυτόματη δημιουργία συνεργαζόμενων πληροφοριακών συστημάτων (*cooperative information systems, CISs*) από ετερογενείς βάσεις δεδομένων [85, 86, 87, 88]. Δέχεται ως είσοδο ένα σύνολο βάσεων δεδομένων που ανήκουν στο CIS. Δημιουργεί ένα είδος ενδιάμεσου σχήματος (που καλείται data repository ή καθολικό δομημένο λεξικό) με σκοπό να παρέχει μια φιλική προς το χρήστη πρόσβαση στις διαθέσιμες πηγές δεδομένων. Το DIKE εστιάζει στα σχήματα οντοτήτων-σχέσεων. Η συσχέτιση ορίζεται στο στάδιο της εξαγωγής γνώσης από το σχήμα και εκτελείται με ημιαυτόματο τρόπο. Μερικά παραδείγματα των ιδιοτήτων του σχήματος, που το DIKE μπορεί να εντοπίσει, είναι ιδιότητες ορολογίας, όπως συνώνυμα, ομώνυμα, ή ασυμφωνίες τύπων, πχ. ομοιότητες μεταξύ διαφορετικών τύπων αντικειμένων, όπως οντότητες, γνωρίσματα, σχέσεις, δομικές ιδιότητες, όπως έγκλιση αντικειμένων (object inclusion), ομοιότητες υποσχημάτων, όπως ομοιότητες μεταξύ τμημάτων των σχημάτων. Το κάθε είδος ιδιότητας σχετίζεται με ένα συντελεστή αληθοφάνειας στο εύρος [0, 1]. Οι ιδιότητες με ένα συντελεστή αληθοφάνειας μικρότερο από ένα κατώφλι απορρίπτονται, ενώ οι υπόλοιπες γίνονται αποδεκτές. Το DIKE λειτουργεί υπολογίζοντας ακολουθιακά τις ιδιότητες που περιγράφηκαν παραπάνω. Για παράδειγμα, συνώνυμα και ομώνυμα εξάγονται με βάση την πληροφορία από εξωτερικές πηγές, όπως το WordNet[11]. Επίσης, κάποια βάρη μπορεί να χρησιμοποιηθούν για την παραγωγή του τελικού

11 http://wordnet.princeton.edu/

συντελεστή. Έτσι, οι ασυμφωνίες τύπων αναλύονται λαμβάνοντας υπόψη τα αποτελέσματα της ανάλυσης των συνωνύμων και των ομωνύμων.

Artemis (Universita di Milano and Universita di Modena e Reggio Emilia)

Το Artemis (Analysis of Requirements: Tool Environment for Multiple Information Systems) [89] σχεδιάστηκε ως ένα τμήμα του ενδιάμεσου συστήματος MOMIS [90, 91] για τη δημιουργία καθολικών όψεων (views). Εκτελεί μια ανάλυση βασισμένη στη συνάφεια και συσταδοποιεί ιεραρχικά τα στοιχεία του σχήματος της βάσης. Η ανάλυση βασισμένη στη συνάφεια αποτελεί το στάδιο της συσχέτισης: με ακολουθιακό τρόπο υπολογίζει τον ονοματικό, δομικό και καθολικό συντελεστή συνάφειας αξιοποιώντας έναν κοινό θησαυρό. Ο κοινός θησαυρός δημιουργείται χρησιμοποιώντας το ODB-Tools [92], το WordNet ή χειρωνακτική είσοδο. Αναπαριστά ένα αρχικό σύνολο και ένα επεκτάσιμο σύνολο σχέσεων που απεικονίζουν τη γνώση των σχημάτων και μεταξύ τους, για τις κλάσεις και τα γνωρίσματα τους. Βασισμένη στον καθολικό συντελεστή συνάφειας, μια τεχνική ιεραρχικής συσταδοποίησης κατηγοριοποιεί τις κλάσεις σε ομάδες διαφορετικού επιπέδου συνάφειας. Για κάθε συστάδα δημιουργεί ένα σύνολο καθολικών γνωρισμάτων και την καθολική κλάση. Οι λογικές αντιστοιχίες μεταξύ των γνωρισμάτων της καθολικής κλάσης και των γνωρισμάτων των πηγαίων σχημάτων καθορίζεται με τη χρήση ενός πίνακα αντιστοίχισης.

Anchor-Prompt (Stanford Medical Informatics)

Το Anchor-Prompt [93] είναι μια επέκταση του Prompt, που ήταν επίσης γνωστό ως SMART. Είναι ένα εργαλείο συνένωσης και ευθυγράμμισης οντολογιών με ένα εξελιγμένο γρήγορο μηχανισμό παραγωγής πιθανών συσχετίσεων όρων [94]. Το Prompt χειρίζεται οντολογίες, διατυπωμένες σε OWL Και RDF Schema. Το Anchor-Prompt είναι ένας αλγόριθμος ακολουθιακής συσχέτισης που παίρνει ως είσοδο δύο οντολογίες, τις αναπαριστά εσωτερικά ως γράφους και ένα σύνολο από ζεύγη σχετιζόμενων όρων, που εντοπίζονται με τη βοήθεια τεχνικών βασισμένων σε αλφαριθμητικά. Μετά, ο αλγόριθμος εκκαθαρίζει τα ζεύγη, αναλύοντας τα μονοπάτια των οντολογιών εισόδου, με σκοπό να καθορίσει τους όρους που εμφανίζονται συχνά σε όμοιες θέσεις, όμοιων μονοπατιών. Τελικά, ο αλγόριθμος, βασισμένος στις συχνότητες και στην ανάδραση του χρήστη, καθορίζει τις υποψήφιες συσχετίσεις.

Το Prompt και το Anchor-Prompt έχουν συνεισφέρει στη σχεδίαση άλλων αλγορίθμων, όπως στο PromptDiff, που εντοπίζει διαφορές μεταξύ δύο οντολογιών και παρέχει τη δυνατότητα επεξεργασίας για τη μετατροπή της μιας οντολογίας στην άλλη [95, 96].

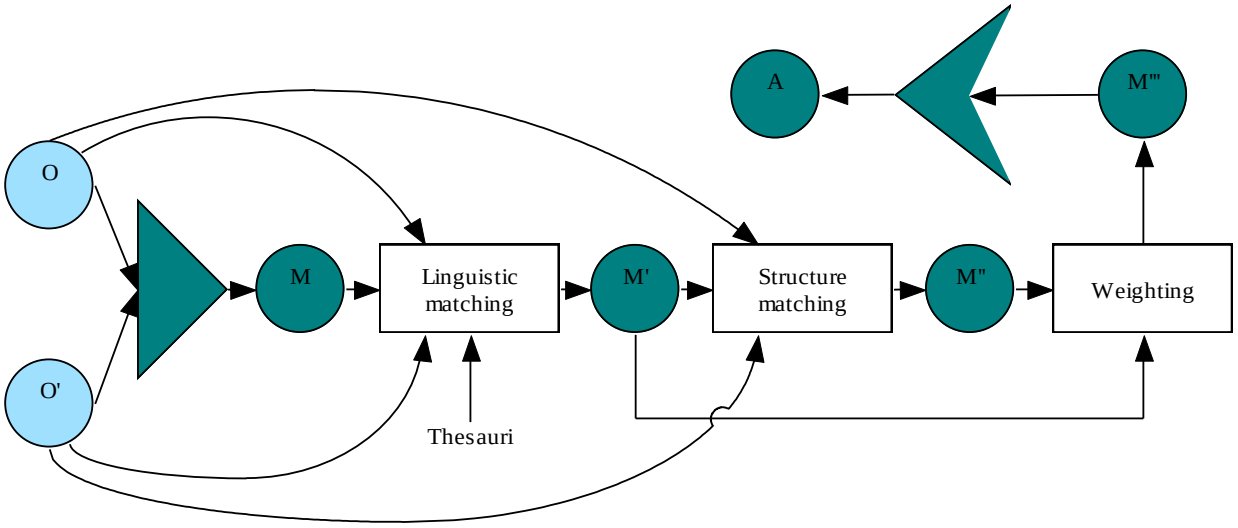
OntoBuilder (Technion Israel Institute of Technology)

Το OntoBuilder είναι ένα σύστημα αναζήτησης πληροφορίας στον ιστό [97]. Μια κλασική περίπτωση που το σύστημα χειρίζεται είναι όταν ένας χρήστης αναζητά να ενοικιάσει ένα αυτοκίνητο. Προφανώς, θα ήθελε να συγκρίνει τις τιμές από τους διαφορετικούς παρόχους, με σκοπό να ενημερωθεί πριν αποφασίσει. Το OntoBuilder λειτουργεί σε δύο φάσεις: (i) τη δημιουργία οντολογίας (φάση εκπαίδευσης) και (ii) την προσαρμογή οντολογίας (φάση προσαρμογής). Κατά τη διάρκεια της φάσης της εκπαίδευσης, δημιουργείται μια πρωταρχική οντολογία βασισμένη σε έναν σχετικό ιστότοπο, πχ. μια εταιρεία ενοικίασης αυτοκινήτων. Η φάση προσαρμογής περιλαμβάνει τη δυναμική συσχέτιση και τη διαδραστική συνένωση των σχετιζόμενων οντολογιών με την αρχική οντολογία. Στη συνέχεια, περιγράφεται εκτενώς η φάση της προσαρμογής. Κατά τη διάρκεια της φάσης της προσαρμογής, οι χρήστες προτείνουν ιστοσελίδες που επιθυμούν να εξεταστούν περαιτέρω. Κάθε τέτοια ιστοσελίδα υπόκειται στη διαδικασία μετατροπής σε οντολογία. Έτσι προκύπτει μια υποψήφια οντολογία, που συνενώνεται με την αρχική. Επιλέγεται η βέλτιστη συσχέτιση για κάθε όρο της αρχικής οντολογίας με τους όρους από την υποψήφια οντολογία. Η στρατηγική επιλογής επιστρατεύει ελάχιστα όρια (thresholds). Ο αλγόριθμος συσχέτισης λειτουργεί με ένα από όρο-σε-όρο τρόπο, ακολουθιακά αξιοποιώντας διαφορετικούς συσχετιστές, πχ. συσχέτιση υποαλφαριθμητικών. Αν οι συσχετιστές αυτοί αποτύχουν, τότε εκτελείται αναζήτηση σε θησαυρό. Τέλος, εμφανίζονται στο χρήστη οι ασυσχέτιστοι όροι για χειρωνακτική συσχέτιση.

Cupid (University of Washington, Microsoft Corporation and University of Leipzig)

Το Cupid [98] υλοποιεί έναν αλγόριθμο που περιλαμβάνει γλωσσολογικές και δομικές τεχνικές συσχέτισης σχημάτων και υπολογίζει τους συντελεστές ομοιότητας με τη βοήθεια εξειδικευμένου θησαυρού. Τα σχήματα που δίνονται ως είσοδοι, κωδικοποιούνται στη μορφή γράφων. Οι κόμβοι αναπαριστούν τα στοιχεία του σχήματος και η διάσχιση τους γίνεται με ένα συνδυαστικό από κάτω προς τα πάνω και από πάνω προς τα κάτω τρόπο. Ο αλγόριθμος συσχέτισης αποτελείται

από τρεις φάσεις (Εικόνα 3.1) και χειρίζεται μόνο δενδρικές δομές, στις οποίες ανάγονται και οι δομές που δεν είναι δέντρα.



Εικόνα 3.1. Η αρχιτεκτονική του Cupid: Πρόκειται για μια συνηθισμένη αρχιτεκτονική που συνδυάζει την παράλληλη και ακολουθιακή σύνθεση. Η δομική συσχέτιση αξιοποιεί τα αποτελέσματα της λεξικογραφικής, αλλά τα αποτελέσματα και των δύο συσχετίσεων συνυπολογίζονται με την απόδοση βαρών.

Στην πρώτη φάση (γλωσσολογική συσχέτιση) υπολογίζονται οι συντελεστές γλωσσικής ομοιότητας μεταξύ των ονομάτων (ετικέτες) των στοιχείων των σχημάτων, επιστρατεύοντας τη μορφολογική κανονικοποίηση, κατηγοριοποίηση, τεχνικές βασισμένες σε αλφαριθμητικά (πχ. πρόθεμα) και αναζήτηση σε θησαυρό. Η δεύτερη φάση (δομική συσχέτιση) συνίσταται στον υπολογισμό συντελεστών δομικής ομοιότητας σταθμισμένων από τα φύλλα, που μετρούν την ομοιότητα των στοιχείων των σχημάτων με βάση τα συμφραζόμενα. Η τρίτη φάση (παραγωγή συσχέτισης των στοιχείων), συναθροίζει τα αποτελέσματα της γλωσσολογικής και δομικής συσχέτισης, μέσω ενός σταθμισμένου αθροίσματος και παράγει μια τελική ευθυγράμμιση, επιλέγοντας ζεύγη στοιχείων των σχημάτων με σταθμισμένους συντελεστές ομοιότητας που ξεπερνούν ένα κατώφλι.

COMA and COMA++ (University of Leipzig)

Το COMA (Combination of Matching algorithms) [99] είναι ένα εργαλείο συσχέτισης σχημάτων βασισμένο στην παράλληλη σύνθεση συσχετιστών. Παρέχει μια επεκτάσιμη βιβλιοθήκη αλγορίθμων συσχέτισης, τη δυνατότητα διαχείρισης

προηγούμενων αποτελεσμάτων και μια πλατφόρμα για την αξιολόγηση της αποτελεσματικότητας των διαφορετικών συσχετιστών. Σύμφωνα με τους [99], το COMA περιέχει έξι στοιχειώδεις συσχετιστές, πέντε υβριδικούς και έναν προσανατολισμένο στην επαναχρησιμοποίηση συσχετίσεων. Οι περισσότεροι από αυτούς υλοποιούν τεχνικές βασισμένες σε αλφαριθμητικά, όπως προθέματα, επιθέματα, αποστάσεις, ενώ άλλοι χρησιμοποιούν τεχνικές όπως το Cupid, πχ. αναζήτηση σε θησαυρό. Ένα πρωτότυπο συστατικό, ο συσχετιστής προσανατολισμένος στην επαναχρησιμοποίηση, προσπαθεί να αξιοποιήσει προηγούμενα αποτελέσματα για νέα σχήματα. Τα σχήματα αναπαριστώνται εσωτερικά στη μορφή άκυκλων γράφων, όπου τα στοιχεία είναι τα μονοπάτια. Αυτό αποσκοπεί στην καταγραφή των συμφραζομένων στα οποία εμφανίζονται τα στοιχεία. Τα ξεχωριστά χαρακτηριστικά του COMA, σε σχέση με το Cupid, είναι η περισσότερο ευέλικτη αρχιτεκτονική και η δυνατότητα των επαναλήψεων κατά τη διάρκεια της διαδικασίας της συσχέτισης. Αυτό προϋποθέτει αλληλεπίδραση με τους χρήστες που εγκρίνουν προτεινόμενες συσχετίσεις και αναντιστοιχίες και βαθμιαία βελτιώνεται και τελειοποιείται η ακρίβεια της συσχέτισης. Το COMA++ στηρίζεται στο COMA, εστιάζοντας με περισσότερη λεπτομέρεια στην ευθυγράμμιση με βάση την επαναχρησιμοποίηση. Επίσης, παρέχει μια περισσότερο αποδοτική υλοποίηση των αλγορίθμων του COMA και μια γραφική διεπαφή.

Similarity flooding (Stanford University and University of Leipzig)

Η προσέγγιση του Similarity flooding [100] βασίζεται στην ιδέα της διάδοσης της ομοιότητας. Τα σχήματα αναπαριστώνται ως κατευθυνόμενοι επισημασμένοι γράφοι κάτω από τις προδιαγραφές του OMI [101]. Ο αλγόριθμος χειρίζεται τους γράφους με μια επαναληπτική διαδικασία για να παράγει την ευθυγράμμιση μεταξύ των κόμβων τους. Η τεχνική αυτή ξεκινά με μια σύγκριση, βασισμένη σε αλφαριθμητικά όπως κοινό πρόθεμα ή επίθεμα, των ετικετών για να προτείνει μια αρχική ευθυγράμμιση που βελτιστοποιείται μέσω του επαναληπτικού υπολογισμού. Η βασική ιδέα πίσω από τον αλγόριθμο του Similarity flooding είναι ότι η ομοιότητα εξαπλώνεται από όμοιους κόμβους στους παρακείμενα γειτονικούς τους μέσω συντελεστών διάδοσης. Από τη μια επανάληψη έως την επόμενη, η μετρική ομοιότητας εξαπλώνεται στο γράφο μέχρι ένα προκαθορισμένο σημείο ή μέχρι να ολοκληρωθεί η διαδικασία. Το αποτέλεσμα είναι μια ακριβής ευθυγράμμιση που μπορεί να φιλτραριστεί περαιτέρω για την παραγωγή της τελικής ευθυγράμμισης.

MapOnto (University of Toronto and Rutgers University)

Το MapOnto είναι ένα σύστημα για τη δημιουργία σύνθετων συσχετίσεων μεταξύ οντολογιών, σχεσιακών και XML σχημάτων [102, 103, 104]. Το εργαλείο αυτό λειτουργεί όμοια με το Clio. Κατά μια έννοια, το σύστημα αυτό μπορεί να θεωρηθεί ως επέκταση του Clio όταν το σχήμα στόχου είναι μια οντολογία που αντιμετωπίζεται ως σχεσιακό σχήμα που περιέχει μοναδιαίους και δυαδικούς πίνακες. Το MapOnto δέχεται ως είσοδο τρία ορίσματα: (i) μια οντολογία σε μια προκαθορισμένη γλώσσα, πχ. OWL, (ii) σχεσιακό ή XML σχήμα και (iii) απλές αντιστοιχίες, πχ. μεταξύ XML γνωρισμάτων και ιδιοτήτων των τύπων δεδομένων της οντολογίας. Τα σχήματα εισόδου και η οντολογία κωδικοποιούνται εσωτερικά με τη μορφή επισημασμένων γράφων. Μετά, η προσέγγιση αυτή αναζητά «λογικές» συνδέσεις μεταξύ των γράφων. Το σύστημα παρέχει με ένα ημιαυτόματο τρόπο ένα σύνολο σύνθετων τύπων συσχέτισης, εκφρασμένους σε ένα υποσύνολο λογισμού πρώτης τάξης (προτάσεις Horn). Η λίστα των τύπων αυτών ταξινομείται από το εργαλείο, έτσι ώστε να προτείνει τις περισσότερο λογικές συσχετίσεις. Τέλος, οι χρήστες μπορούν να ελέγξουν τη λίστα αυτή και να επιλέξουν τα βέλτιστα αποτελέσματα.

OntoMerge (Yale University and University of Oregon)

Το OntoMerge [105] είναι ένα εργαλείο μετάφρασης οντολογιών στο σημασιολογικό ιστό. Η μετάφραση οντολογιών αναφέρεται σε λειτουργίες όπως: (i) μετάφραση συνόλου δεδομένων, που αποτελεί τη μετάφραση ενός συνόλου γεγονότων εκφρασμένων από τη μια οντολογία στην άλλη, (ii) παραγωγή επεκτάσεων οντολογίας, δεδομένων δύο οντολογιών και μιας επέκτασης της πρώτης, δημιουργείται η αντίστοιχη επέκταση της δεύτερης, (iii) απάντηση επερωτήσεων για πολλαπλές οντολογίες. Η βασική ιδέα της προσέγγισης αυτής είναι η μετάφραση οντολογιών μέσω της συνένωσης οντολογιών και του αυτόματου συλλογισμού (reasoning). Οι οντολογίες εισόδου μεταφράζονται από τη γλώσσα αναπαράστασης γνώσης, πχ. OWL, σε μια εσωτερική αναπαράσταση, την Web-PDDL [106]. Η συνένωση των δύο οντολογιών γίνεται δημιουργώντας την ένωση των αξιωμάτων που τις ορίζουν. Τα αξιώματα ή οι κανόνες γεφύρωσης προστίθενται, στη συνέχεια, για να συσχετίσουν τους όρους της μιας οντολογίας με τους όρους της άλλης. Αφού γίνει η συνένωση των οντολογιών, μπορεί να υλοποιηθεί η λειτουργία της μετάφρασης της οντολογίας που προέκυψε, με πλήρως αυτόματο τρόπο. Θεωρείται ότι οι κανόνες γεφύρωσης ορίζονται από τους ειδικούς

του ερευνητικού τομέα, ή από άλλους αλγορίθμους συσχέτισης ικανούς να τους εντοπίζουν και να μεταφράζουν τη σημασιολογία τους. Τέλος, αξίζει να σημειωθεί ότι το OntoMerge υποστηρίζει κανόνες γεφύρωσης, που μπορεί να εκφραστούν χρησιμοποιώντας όλες τις δυνατότητες του λογισμού κατηγορημάτων.

S-Match (University of Trento)

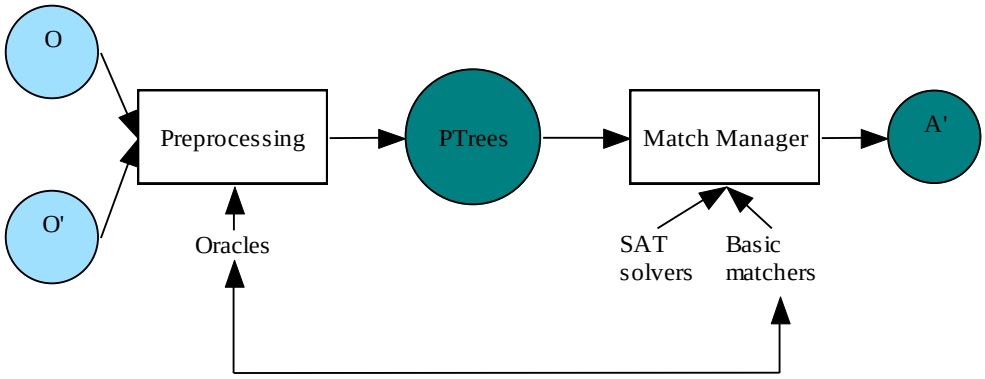
Το S-Match υλοποιεί την ιδέα της σημασιολογικής συσχέτισης όπως περιγράφηκε αρχικά στο [107]. Το S-Match περιορίζεται σε δενδρικές δομές και δε λαμβάνει υπόψη ιδιότητες ή ρόλους.

Το S-Match δέχεται ως είσοδο δύο δομές στη μορφή γράφου, πχ. κατηγοριοποιήσεις, XML σχήματα, οντολογίες και επιστρέφει ως έξοδο λογικές σχέσεις, πχ. ισότητες, οι οποίες υποτίθεται ότι συνδέουν τους κόμβους των γράφων. Οι σχέσεις καθορίζονται από (i) την έκφραση των οντοτήτων των οντολογιών ως λογικοί τύποι, και (ii)τον περιορισμό του προβλήματος της συσχέτισης σε πρόβλημα εγκυρότητας των προτεινόμενων. Συγκεκριμένα, οι οντότητες μεταφράζονται σε προτασιακούς τύπους, οι οποίες περιγράφουν τις έννοιες όπως κωδικοποιούνται στη δομή της οντολογίας αλλά και σε εξωτερικές πηγές, όπως το WordNet. Αυτό επιτρέπει τη μετάφραση του προβλήματος της συσχέτισης σε πρόβλημα εγκυρότητας των προτεινόμενων, το οποίο μπορεί να αντιμετωπιστεί αποδοτικά (ορθό και πλήρες) χρησιμοποιώντας προηγμένης τεχνολογίας προτασιακής ικανοποιησιμότητας λύσεις.

Το S-Match σχεδιάστηκε και αναπτύχθηκε ως πλατφόρμα για σημασιολογική συσχέτιση, δηλαδή ένα επεκτάσιμο σύστημα με πυρήνα τον υπολογισμό τις σημασιολογικές σχέσεις, όπου κάθε συστατικό μπορεί να είναι ενσωματωμένο, μη ενσωματωμένο ή κατάλληλα προσαρμοσμένο. Είναι ένα ακολουθιακό σύστημα με μια παράλληλη σύνθεση στο επίπεδο στοιχείου (Εικόνα 3.2). Οι οντολογίες εισόδου (δενδρική δομή) κωδικοποιούνται σε μια πρότυπη XML μορφή. Το τμήμα που δέχεται ως είσοδο τις οντολογίες εκτελεί κάποιες προεπεξεργασίες με τη βοήθεια προβλέψεων (oracles), που παρέχουν την απαραίτητη εκ των προτέρων λεξικογραφική και εξειδικευμένη σε τομέα γνώση. Παραδείγματα τέτοιων προβλέψεων περιλαμβάνουν το WordNet και το UMLS[12]. Η έξοδος είναι ένα εμπλουτισμένο δέντρο. Τα εμπλουτισμένα δέντρα αποθηκεύονται σε μια εσωτερική βάση (PTrees) και είναι εφικτό να προβληθούν και να επεξεργαστούν. Το Match

12 http://www.nlm.nih.gov/research/umls/

manager συντονίζει τη διαδικασία συσχέτισης. Οι βιβλιοθήκες του S-Match περιέχουν περίπου είκοσι βασικούς συσχετιστές επιπέδου στοιχείου από τρεις κατηγορίες, δηλαδή βασισμένοι σε αλφαριθμητικά, όπως επεξεργασία απόστασης, με βάση την έννοια και τα σχόλια στο WordNet. Οι συσχετιστές επιπέδου δομής περιλαμβάνουν SAT λύσεις και ειδικές μεθόδους λογικής (ad hoc reasoning)[60].



Εικόνα 3.2. Η αρχιτεκτονική του S-Match: Οι οντότητες των οντολογιών μετατρέπονται σε λογικούς τύπους χρησιμοποιώντας τον προεπεξεργαστή και προβλέψεις. Το τμήμα του Match Manager χρησιμοποιεί διάφορους βασικούς συσχετιστές επιπέδου στοιχείου και λογικές αποδείξεις για την εύρεση σχέσεων μεταξύ των τύπων αυτών, οι οποίες με τη σειρά τους αντιστοιχούν σε σχέσεις μεταξύ των οντοτήτων.

3.1.2 Συστήματα Βασισμένα στην Πληροφορία των Στιγμιοτύπων

Τα συστήματα βασισμένα στα στιγμιότυπα είναι αυτά που αξιοποιούν κυρίως τα στιγμιότυπα, δηλαδή τα δεδομένα που εκφράζει ένα σχήμα ή τα δεδομένα που ευρετηριοποιεί ένα σχήμα.

LSD (University of Washington)

Το LSD (Learning Source Description) είναι ένα σύστημα ημιαυτόματου εντοπισμού των ευθυγραμμίσεων μεταξύ των στοιχείων των σχημάτων πηγής και ένα ενδιάμεσο (καθολικό) σχήμα στην ενοποίηση δεδομένων [108]. Η βασική ιδέα της προσέγγισης αυτής είναι η εκμάθηση από τις χειρωνακτικές συσχετίσεις μεταξύ του ενδιάμεσου σχήματος και μερικών σχημάτων πηγής, με σκοπό να προτείνει με αυτόματο τρόπο τις συσχετίσεις για τα επόμενα σχήματα πηγής. Το LSD χειρίζεται XML σχήματα. Ένα σχήμα μοντελοποιείται στη μορφή δέντρου, του οποίου οι κόμβοι είναι XML ετικέτες. Η προσέγγιση αυτή λειτουργεί σε δύο φάσεις. Κατά τη διάρκεια της πρώτης φάσης (εκπαίδευση), χρήσιμα αντικείμενα, όπως ονόματα

αντικειμένων και τύποι δεδομένων, συλλέγονται από τα σχήματα εισόδου. Στη συνέχεια, δημιουργούνται ευθυγραμμίσεις από αυτά τα αντικείμενα, το σύστημα εκπαιδεύει πολλαπλούς βασικούς συσχετιστές (αντιμετωπίζοντας διαφορετικά χαρακτηριστικά των αντικειμένων, όπως η μορφή, συχνότητες λέξεων, κατανομές των τιμών) και ένα μετα-συσχετιστή (meta-matcher). Μερικά παραδείγματα τέτοιων συσχετιστών είναι οι εξής: WHIRL learner και naïve Bayesian learner. Ο Meta-matcher συνδυάζει τις προβλέψεις των βασικών συσχετιστών. Εκπαιδεύεται χρησιμοποιώντας μια τεχνική εκμάθησης, τη συσσωρευμένη γενίκευση (stacked generalization). Κατά τη διάρκεια της δεύτερης φάσης (συσχέτιση), το LSD αποσπά τα απαραίτητα αντικείμενα από τα νέα σχήματα πηγής. Μετά, εφαρμόζοντας τους εκπαιδευμένους βασικούς συσχετιστές και τον μετα-συσχετιστή στα νέα αντικείμενα (η διαδικασία της κατηγοριοποίησης), το LSD δημιουργεί μια λίστα προβλέψεων με τις υποψήφιες συσχετίσεις. Τέλος, λαμβάνοντας υπόψη περιορισμούς ακεραιότητας και εφαρμόζοντας κατώφλια, προκύπτει το αποτέλεσμα της τελικής ευθυγράμμισης.

GLUE (University of Washington)

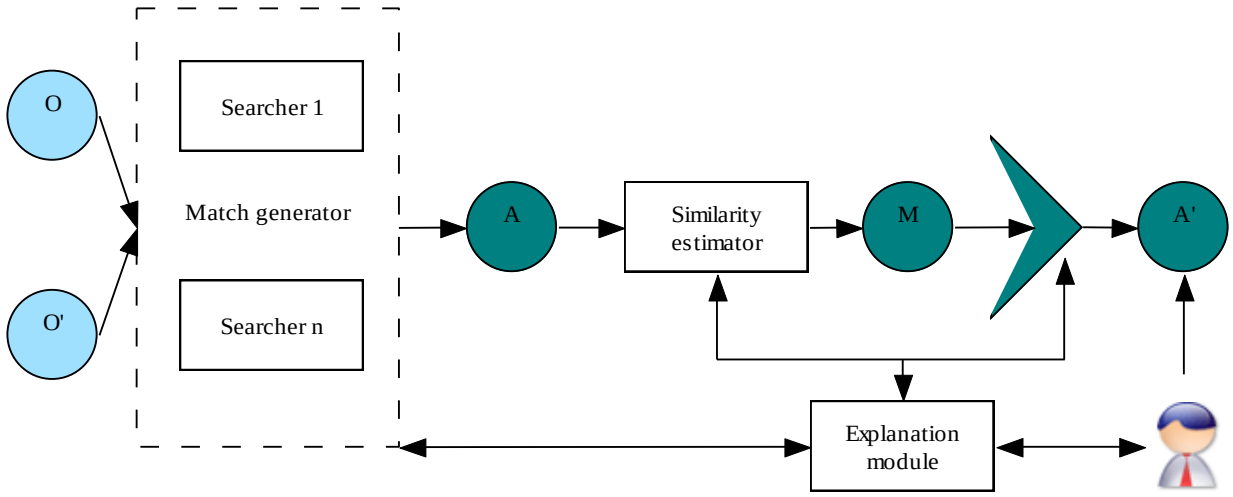
Το GLUE [109], ένας διάδοχος του LSD, είναι ένα σύστημα που επιστρατεύει πολλαπλές τεχνικές μηχανικής μάθησης για τον ημιαυτόματο εντοπισμό σημασιολογικών συσχετίσεων (που μερικές φορές καλούνται “glue” για τη διαλειτουργικότητα) μεταξύ δύο ταξονομιών. Η ιδέα της προσέγγισης αυτής είναι ο υπολογισμός της από κοινού κατανομής (joint distribution) των κατανομών των κλάσεων, αντί της πρότασης μιας συγκεκριμένης μετρικής ομοιότητας. Έτσι, κάθε μετρική ομοιότητας μπορεί να υπολογιστεί συναρτήσει των από κοινού κατανομών. Όπως και ο προκάτοχος LSD, το GLUE ακολουθεί μια πολυστρατηγική εκμάθησης, που περιλαμβάνει διάφορους βασικούς συσχετιστές και ένα μετα-συσχετιστή. Το σύστημα λειτουργεί σε τρία βήματα. Πρώτον, μαθαίνει τις από κοινού κατανομές πιθανότητας (joint probability distributions) των κλάσεων των δύο ταξονομιών. Συγκεκριμένα, εκμεταλλεύεται δύο βασικούς συσχετιστές, το μαθητευόμενο περιεχομένου (content learner, naïve Bayes τεχνική) και το μαθητευόμενο ονόματος (name learner, μια παραλλαγή του προηγούμενου). Ο μετα-συσχετιστής, με τη σειρά του εκτελεί ένα γραμμικό συνδυασμό των βασικών συσχετιστών. Τα βάρη για τους συσχετιστές αυτούς καθορίζονται χειρωνακτικά. Κατά τη διάρκεια του δεύτερου βήματος, το σύστημα προτείνει μια εκτίμηση της ομοιότητας μεταξύ των δύο κλάσεων χρησιμοποιώντας μια συνάρτηση που

καθορίζει ο χρήστης και τις *από* κοινού κατανομές πιθανότητας. Αυτό έχει ως αποτέλεσμα έναν πίνακα ομοιότητας μεταξύ των όρων των δύο ταξονομιών. Τελικά, μερικοί εξαρτημένοι από τον τομέα (πχ. περίληψη -subsumption-) και ανεξάρτητοι από αυτόν (πχ. αν όλα τα παιδιά ενός κόμβου *x* συσχετίζονται με έναν κόμβο *y*, τότε και ο κόμβος *x* συσχετίζεται με τον κόμβο *y*), περιορισμοί ή ευριστικές εφαρμόζονται χρησιμοποιώντας μια τεχνική χαλαρής ετικετοποίησης (relaxation labeling*)*. Οι περιορισμοί αυτοί χρησιμοποιούνται με σκοπό το φιλτράρισμα των συσχετίσεων στον πίνακα ομοιότητας για να μείνουν οι βέλτιστες.

iMAP (University of Illinois and University of Washington)

Το iMAP [110] είναι ένα σύστημα που εντοπίζει ημιαυτόματα μια-προς-μια (πχ. amount - quantity) και, κυρίως, σύνθετες (πχ. address – concat(city,street)) συσχετίσεις μεταξύ σχημάτων σχεσιακών βάσεων δεδομένων. Το πρόβλημα συσχέτισης σχημάτων ανάγεται σε αναζήτηση στο χώρο συσχέτισης, ο οποίος είναι συνήθως πολύ μεγάλος ή τείνει στο άπειρο. Για την αποδοτική υλοποίηση της αναζήτησης, το iMAP χρησιμοποιεί πολλαπλούς βασικούς συσχετιστές, πχ. κειμενικούς, αριθμητικούς, εξέταση κατηγορίας, μετατροπή μονάδων μέτρησης, κάθε ένας από τους οποίους εξετάζει ένα υποσύνολο του χώρου συσχέτισης. Για παράδειγμα, η αναζήτηση με βάση το κείμενο λαμβάνει υπόψη της τη συνένωση των κειμενικών γνωρισμάτων, ενώ η αριθμητική αναζήτηση λαμβάνει υπόψη της τους συνδυασμούς γνωρισμάτων με αριθμητικές εκφράσεις. Το σύστημα λειτουργεί σε τρία βήματα (Εικόνα 3.3). Αρχικά, παράγονται οι υποψήφιες συσχετίσεις, εφαρμόζοντας τους βασικούς συσχετιστές (τμήμα του match generator). Ακόμα κι αν ένας βασικός συσχετιστής, όπως ο κειμενικός, εξετάζει μόνο το χώρο των συνενώσεων των γνωρισμάτων, ο χώρος αυτός μπορεί να είναι πολύ μεγάλος. Για το σκοπό αυτό η στρατηγική αναζήτησης ελέγχεται χρησιμοποιώντας μια τεχνική αναζήτησης δέσμης (beam search) [111]. Κατά τη διάρκεια του δεύτερου βήματος, για κάθε γνώρισμα του σχήματος στόχου, αξιολογούνται οι υποψήφιες συσχετίσεις με το σχήμα πηγής, αξιοποιώντας πρόσθετους τύπους πληροφορίας, πχ. χρησιμοποιώντας τον naïve Bayes αξιολογητή που είναι υπολογιστικά ακριβός για να εφαρμοστεί στο πρώτο βήμα. Έτσι προκύπτουν επιπρόσθετα αποτελέσματα. Στη συνέχεια, τα αποτελέσματα αυτά συνδυάζονται σε ένα τελικό (το τμήμα του similarity estimator). Το αποτέλεσμα του βήματος αυτού είναι ένας πίνακας ομοιότητας με τα ζεύγη <*target attribute, match candidate*>. Τελικά, χρησιμοποιώντας ένα σύνολο εξειδικευμένων στον τομέα περιορισμών και συσχετίσεις από

προηγούμενες εκτελέσεις του αλγορίθμου (αν είναι εφαρμόσιμες και διαθέσιμες), γίνεται μια εκκαθάριση του πίνακα ομοιοτήτων, τέτοια ώστε να επιστραφούν οι βέλτιστες συσχετίσεις για τα γνωρίσματα του σχήματος στόχου (το τμήμα του match selector). Το σύστημα αυτό είναι επίσης ικανό να εξηγεί τα αποτελέσματα που παράγει με τη βοήθεια του τμήματος explanation.



Εικόνα 3.3.Η αρχιτεκτονική του iMAP: Διαφορετικοί συσχετιστές (εδώ searchers), λειτουργούν παράλληλα. Παρέχουν υποψήφιες συσχετίσεις που μπορεί να είναι σύνθετες. Οι συσχετίσεις αυτές υπόκεινται, στη συνέχεια, σε μια διαδικασία επιλογής χρησιμοποιώντας το Similarity estimator και, μετά, παράγεται η τελική ευθυγράμμιση. Επιπροσθέτως, το Explanation module βοηθάει τους χρήστες να κατανοήσουν τα αποτελέσματα και να ελέγχουν τη διαδικασία.

Automatch (George Mason University)

Το Automatch [112] είναι ένα σύστημα αυτόματου εντοπισμού συσχετίσεων μεταξύ των γνωρισμάτων σχημάτων βάσεων δεδομένων. Η προσέγγιση αυτή προϋποθέτει ότι αρκετά σχήματα του ίδιου τομέα λαμβάνονται υπόψη και έχουν ήδη συσχετιστεί χειρωνακτικά από ειδικούς του τομέα. Η προϋπόθεση αυτή είναι ρεαλιστική για ένα σενάριο ενοποίησης δεδομένων. Στη συνέχεια, χρησιμοποιώντας Bayesian εκμάθηση, το Automatch αποκτά πιθανολογική γνώση από τα χειρωνακτικά συσχετισμένα σχήματα και δημιουργεί ένα λεξικό γνωρισμάτων, το οποίο συσσωρεύει τη γνώση για κάθε γνώρισμα μέσω των πιθανών τιμών του και τις εκτιμήσεις πιθανότητας των τιμών αυτών. Για να αποφευχθεί η γρήγορη ανάπτυξη του λεξικού, το σύστημα επιστρατεύει τεχνικές στατιστικής επιλογής γνωρισμάτων, όπως αμοιβαία πληροφορία (mutual Information, MI), κέρδος

πληροφορίας (information gain) και απόκλιση (likelihood ratio) για την αποδοτική εκμάθηση, δηλαδή μόνο από τις περισσότερο πληροφοριακές τιμές, όπως το 10% των πραγματικών διαθέσιμων δεδομένων εκπαίδευσης. Ένα νέο ζεύγος σχημάτων συσχετίζεται αυτόματα μέσω του λεξικού γνωρισμάτων. Το σύστημα πρώτα συσχετίζει κάθε γνώρισμα των σχημάτων εισόδου με το λεξικό, έτσι παράγονται ξεχωριστά αποτελέσματα συσχέτισης. Στη συνέχεια, τα ξεχωριστά αυτά αποτελέσματα συνδυάζονται περαιτέρω , υπολογίζοντας το άθροισμα τους για να παραχθεί το αποτέλεσμα μεταξύ των γνωρισμάτων των σχημάτων εισόδου. Τέλος, τα αποτελέσματα μεταξύ τω σχημάτων εισόδου, συνδυάζονται ξανά χρησιμοποιώντας έναν ελάχιστου κόστους και μέγιστης ροής αλγόριθμο γράφων και κάποια κατώφλια για την εύρεση της συνολικής βέλτιστης συσχέτισης μεταξύ των σχημάτων εισόδου, σε σχέση με το άθροισμα των ξεχωριστών αποτελεσμάτων συσχέτισης.

3.1.3 Συστήματα Βασισμένα στο Συνδυασμό Πληροφορίας Σχημάτων και Στιγμιοτύπων

Τα παρακάτω συστήματα εκμεταλλεύονται τόσο την πληροφορία επιπέδου σχήματος όσο και την πληροφορία επιπέδου στιγμιοτύπου, στις περιπτώσεις που είναι διαθέσιμες.

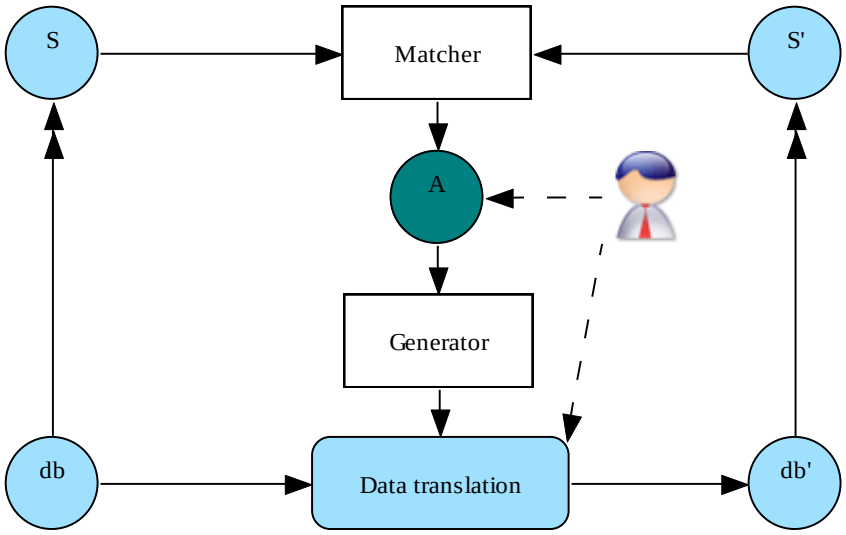
SEMINT (Northwestern University, NEC and The MITRE Corporation)

Το SEMINT (SEMantic INTegrator) είναι ένα εργαλείο βασισμένο σε νευρωνικά δίκτυα για την υποβοήθηση στον εντοπισμό αντιστοιχήσεων γνωρισμάτων σε ετερογενείς βάσεις δεδομένων [113, 114]. Το σύστημα αυτό υποστηρίζει την πρόσβαση σε μια ποικιλία συστημάτων βάσεων δεδομένων και αξιοποιεί τόσο πληροφορία σχήματος όσο και στιγμιοτύπων για να παράγει κανόνες για την αυτόματη συσχέτιση γνωρισμάτων. Η προσέγγιση αυτή λειτουργεί ως εξής. Αρχικά, εξάγει από τις δύο βάσεις όλη την απαραίτητη πληροφορία (γνωρίσματα ή διευκρινίσεις), η οποία είναι διαθέσιμη και χρήσιμη για τη συσχέτιση. Η πληροφορία αυτή περιλαμβάνει κανονικοποιημένη πληροφορία σχήματος, πχ. προδιαγραφές πεδίων όπως τύποι δεδομένων, μήκος, περιορισμοί, και στατιστικά για τις τιμές των δεδομένων, πχ. υποδείγματα χαρακτήρων, όπως αναλογία αριθμητικών χαρακτήρων, αναλογία κενών χαρακτήρων, και υποδείγματα χαρακτήρων, όπως μέση τιμή, διαφορά, τυπική απόκλιση. Στη συνέχεια, χρησιμοποιώντας ένα νευρωνικό δίκτυο για την κατηγοριοποίηση με τον αλγόριθμο συσχέτισης, ομαδοποιούνται τα γνωρίσματα με βάση την ομοιότητα

τους ως προς τη μια βάση δεδομένων. Μετά, χρησιμοποιείται ένα νευρωνικό δίκτυο οπίσθιας διάδοσης (back-propagation) για την εκμάθηση και την αναγνώριση. Έτσι εκτελείται η λειτουργία της εκμάθησης, με βάση τις συστάδες που διαμορφώθηκαν. Τελικά, χρησιμοποιώντας ένα εκπαιδευμένο νευρωνικό δίκτυο στα γνωρίσματα και στις συστάδες της πρώτης βάσης δεδομένων, το σύστημα αναγνωρίζει και υπολογίζει τις ομοιότητες μεταξύ των κατηγοριών των γνωρισμάτων της πρώτης και της δεύτερης βάσης. Έτσι, παράγεται μια λίστα με τις υποψήφιες συσχετίσεις που εξετάζονται και γίνονται αποδεκτές ή απορρίπτονται από τους χρήστες.

Clio (IBM Almaden and University of Toronto)

Το Clio είναι ένα σύστημα διαχείρισης και υποβοήθησης λειτουργιών μετατροπής δεδομένων και ενοποίησης μεταξύ ετερογενών περιβαλλόντων [115, 116, 117, 118] (Εικόνα 3.4), και χειρίζεται σχεσιακά και XML σχήματα. Πρώτα απ' όλα, το σύστημα μετατρέπει τα σχήματα εισόδου σε μια εσωτερική αναπαράσταση εμφωλευμένου σχεσιακού μοντέλου. Η προσέγγιση του Clio εστιάζει στη λειτουργική υλοποίηση της ευθυγράμμισης. Το στάδιο της συσχέτισης, ή ο εντοπισμός των αντιστοιχιών των τιμών, εκτελείται με τη βοήθεια ενός τμήματος κώδικα συσχέτισης ή χειρωνακτικά. Ο ενσωματωμένος αλγόριθμος συσχέτισης του Clio συνδυάζει με ακολουθιακό τρόπο την ταξινόμηση των γνωρισμάτων με βάση τα στιγμιότυπα, μέσω μιας παραλλαγής του naïve Bayes classifier και τη συσχέτιση αλφαριθμητικών μεταξύ των ονομάτων των στοιχείων, πχ. χρησιμοποιώντας την απόσταση. Στη συνέχεια, λαμβάνοντας υπόψη τις n-m αντιστοιχίες σε συνδυασμό με τους περιορισμούς που προκύπτουν από τα σχήματα εισόδου, το Clio τα συνδυάζει έτσι ώστε να προκύψει μια εσωτερική αναπαράσταση γράφου επερωτήσεων. Συγκεκριμένα, προκύπτει μια μετάφραση των αντιστοιχιών που δόθηκαν ως είσοδος. Έτσι παράγεται ένα σύνολο λογικών συσχετίσεων με τυπική σημασιολογία. Για το σκοπό αυτό, ο γράφος επερωτήσεων μπορεί να μετατραπεί σε διαφορετικές γλώσσες επερωτήσεων, πχ. SQL, XSLT, XQuery, και είναι εφικτό τα πραγματικά δεδομένα να αλλάζουν μορφή από ένα σχήμα πηγής σε σχήμα στόχου ή να απαντούν επερωτήσεις. Το σύστημα, εκτός των τετριμμένων μετασχηματισμών που εκτελεί, στοχεύει στον εντοπισμό σύνθετων, όπως την παραγωγή κλειδιών, και αναφορών.



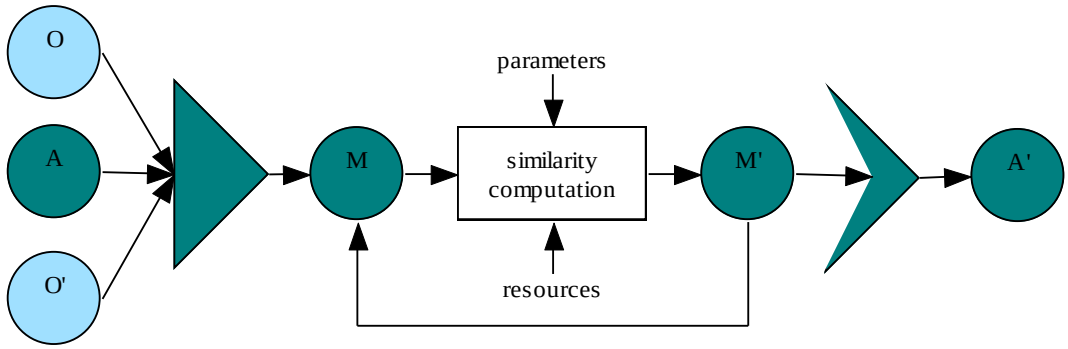
Εικόνα 3.4.Η αρχιτεκτονική του Clio: Το Clio διεκπεραιώνει τη διαδικασία της συσχέτισης ανάγοντας τη στη μετάφραση των δεδομένων από τη μια βάση στην άλλη. Βασίζεται σε ένα κλασικό συσχετιστή αλλά εμπλέκει και τους χρήστες σε κάθε βήμα: είσοδος, έλεγχος συσχέτισης και μετάφραση.

OLA (INRIA Rhone-Alpes and Universite de Montreal)

Το OLA (OWL Lite Aligner) [119] είναι ένα σύστημα συσχέτισης οντολογιών, σχεδιασμένο με βάση την ιδέα της εξισορρόπησης της συνεισφοράς του κάθε συστατικού που συνθέτει μια οντολογία, πχ. κλάσεις, περιορισμούς, στιγμιότυπα δεδομένων. Το OLA επεξεργάζεται οντολογίες σε OWL. Αρχικά, μετατρέπει τις οντολογίες εισόδου σε δομές γράφων, αποκαλύπτοντας όλες τις σχέσεις μεταξύ των οντοτήτων. Αυτές οι δομές γράφων παράγουν τους περιορισμούς για την έκφραση της ομοιότητας μεταξύ των στοιχείων των οντολογιών. Η ομοιότητα μεταξύ των κόμβων των γράφων ακολουθεί τους εξής κανόνες: (i) εξαρτάται από την κατηγορία στην οποία ανήκει ο κόμβος, πχ. κλάση, ιδιότητα, και (ii) λαμβάνει υπόψη όλα τα γνωρίσματα της κατηγορίας αυτής, πχ. υπερκλάσεις, ιδιότητες.

Η απόσταση μεταξύ των κόμβων του γράφου εκφράζεται ως ένα σύστημα ισοτήτων βασισμένων σε αλφαριθμητικές, γλωσσικές και δομικές ομοιότητες (ενώ λαμβάνονται υπόψη και τα στιγμιότυπα όποτε κρίνεται απαραίτητο). Οι αποστάσεις αυτές συναθροίζονται, συνήθως, γραμμικά (συναθροίζονται γραμμικά και υπολογίζεται το υπόλοιπο της ακέραιας διαίρεσης με το πλήθος των τοπικών συσχετίσεων των οντοτήτων). Για τον υπολογισμό των αποστάσεων, ο αλγόριθμος υπολογίζει, αρχικά, βασικές μετρικές απόστασης χρησιμοποιώντας τις ετικέτες και

διακριτούς τύπους δεδομένων. Στη συνέχεια, επαναλαμβάνει έναν αλγόριθμο σταθερού σημείου (fixed point algorithm) έως ότου να μην παράγεται κάποιο διαφορετικό αποτέλεσμα από το προηγούμενο. Η αρχιτεκτονική του συστήματος περιγράφεται στην Εικόνα 3.5.



Εικόνα 3.5.Η αρχιτεκτονική του OLA: Ο επαναληπτικός υπολογισμός του σταθερού σημείου μιας ομοιότητας ή συνάρτησης απόστασης.

Corpus-based matching (University of Washington, Microsoft Research and University of Illinois)

Το [120] προτείνει μια προσέγγιση της συσχέτισης σχημάτων, η οποία εκτός την αξιοποίηση της πληροφορίας των σχημάτων, εκμεταλλεύεται, επίσης, εξειδικευμένη σε τομέα γνώση μέσω μιας εξωτερικής συλλογής σχημάτων και συσχετίσεων. Η προσέγγιση αυτή είναι εμπνευσμένη από τη χρήση μιας συλλογής από την ανάκτηση πληροφορίας, όπου η ομοιότητα μεταξύ των επερωτήσεων και των όρων καθορίζεται με βάση την ανάλυση μεγάλων σωμάτων κειμένου (corpora). Στη συσχέτιση σχημάτων μια τέτοια συλλογή μπορεί να αρχικοποιηθεί με ένα μικρό αριθμό σχημάτων, για παράδειγμα χρησιμοποιώντας πρότυπα διαθέσιμα σχήματα από τον τομέα ενδιαφέροντος και είναι δυνατό να εξελιχθούν σταδιακά.

Δεδομένου ότι η συλλογή προορίζεται να περιέχει διαφορετικές αναπαραστάσεις για κάθε έννοια του τομέα, θα μπορούσε να διευκολύνει την εκμάθηση των παραλλαγών αυτών στα στοιχεία και στις ιδιότητες τους. Η συλλογή αυτή μπορεί να επεκταθεί με δύο τρόπους. Πρώτον, μπορεί να καταγράψει για κάθε προς συσχέτιση στοιχείο τις ομοιότητες του με τα υπόλοιπα στοιχεία της συλλογής. Δεύτερον, στη συλλογή αυτή, όμοια στοιχεία συσταδοποιούνται και υπολογίζονται στατιστικά για τις συστάδες αυτές, όπως γειτονιές και ταξινομήσεις στοιχείων. Τα

στατιστικά αυτά χρησιμοποιούνται για τη δημιουργία περιορισμών που διευκολύνουν στην επιλογή των αντιστοιχίσεων της τελικής ευθυγράμμισης.

Η προσέγγιση αυτή χειρίζεται ηλεκτρονικές φόρμες και σχεσιακά σχήματα και εστιάζει στις μια-προς-μια ευθυγραμμίσεις. Λειτουργεί σε δύο φάσεις. Αρχικά, τα υπό εξέταση σχήματα συσχετίζονται με τη συλλογή, έτσι αυξάνεται το μέγεθος της συλλογής μα τις πιθανές παραλλαγές των στοιχείων με βάση τη διαθέσιμη γνώση της συλλογής. Στη συνέχεια, τα σχήματα συσχετίζονται μεταξύ τους. Και στις δύο φάσεις χρησιμοποιείται το ίδιο σύνολο συσχετιστών. Συγκεκριμένα, οι βασικοί συσχετιστές, περιλαμβάνουν (i)ένα μαθητευόμενο ονόματος, (ii) ένα μαθητευόμενο κειμένου, (iii) ένα μαθητευόμενο στιγμιοτύπων δεδομένων, και (iv) ένα μαθητευόμενο περιεχομένου. Οι συσχετιστές αυτοί ακολουθούν, κυρίως, την πολιτική των τεχνικών του LSD και του Cupid. Για παράδειγμα, ο μαθητευόμενος ονόματος αξιοποιεί τα ονόματα των στοιχείων. Εφαρμόζει κατάτμηση και την τεχνική των n-grams στα ονόματα με σκοπό να δημιουργήσει παραδείγματα εκπαίδευσης. Ο συσχετιστής είναι ένας ταξινομητής κειμένου, όπως ο naïve Bayes. Επιπροσθέτως, ο μαθητευόμενος ονόματος, χρησιμοποιεί την επεξεργασία αποστάσεων για να καθορίσει την ομοιότητα μεταξύ των ονομάτων των στοιχείων. Ο μαθητευόμενος στιγμιοτύπων δεδομένων καθορίζει τις περιπτώσεις που οι τιμές των στιγμιοτύπων χρησιμοποιούν κοινά πρότυπα, ίδιες λέξεις, κλπ. Ένας συσχετιστής, γνωστός ως μετα-συσχετιστής, συνδυάζει τα αποτελέσματα που παράγονται από τους βασικούς συσχετιστές. Χρησιμοποιεί λογιστική παλινδρόμηση (logistic regression) με σκοπό την εκμάθηση των παραμέτρων του. Τελικά, χρησιμοποιώντας περιορισμούς βασισμένους στα στατιστικά που εξάγονται από τη συλλογή, φιλτράρονται οι υποψήφιες συσχετίσεις με σκοπό την παραγωγή της τελικής ευθυγράμμισης.

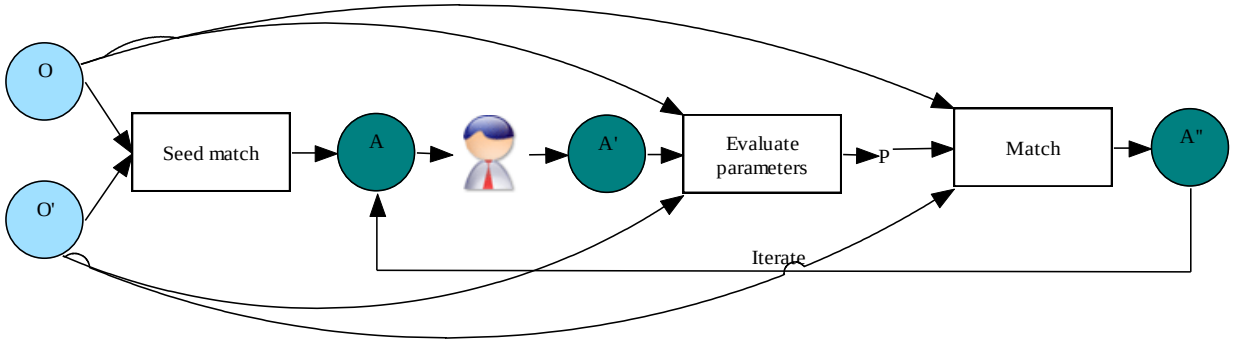
3.1.4 Συστήματα Μετα-συσχέτισης

Τα meta-matching συστήματα είναι συστήματα που η πρωτοτυπία τους έγκειται στον τρόπο που χρησιμοποιούν και συνδυάζουν άλλα συστήματα συσχέτισης αντί να εκτελούν τα ίδια συσχέτιση.

APFEL (University of Karlsruhe and University of Koblenz-Landau)

Το APFEL (Alignment Process Feature Estimation and Learning) είναι μια προσέγγιση μηχανικής μάθησης που διερευνά την αποδοχή η απόρριψη από το χρήστη των αρχικών ευθυγραμμίσεων, για την αυτόματη βελτιστοποίηση των παραμέτρων των

στρατηγικών μηχανικής μάθησης του συστήματος, πχ. βάρη, κατώφλια [121]. Είναι ένα συστατικό του FOAM. Η συνολική αρχιτεκτονική του APFEL περιγράφεται στην Εικόνα 3.6.



Εικόνα 3.6.Η αρχιτεκτονική του APFEL: παράγει ευθυγραμμίσεις και ζητάει ανάδραση από τους χρήστες. Στη συνέχεια, προσαρμόζει τις μεθόδους και τις παραμέτρους συνάθροισης με σκοπό να ελαχιστοποιήσει το σφάλμα και επαναλαμβάνει, αν είναι απαραίτητο.

Το APFEL παραμετροποιεί το FOAM χρησιμοποιώντας δηλωτικές αναπαραστάσεις των (i) χαρακτηριστικών μηχανικής, (ii) εκτιμήσεων ομοιότητας, (iii) σχημάτων βαρών, πχ. για συνάθροιση ομοιότητας, και (iv) κατωφλίων. Για το σκοπό αυτό, οι διεπαφές των συστημάτων μηχανικής μάθησης ενοποιούνται ως Παραμετροποιήσιμες Μέθοδοι Ευθυγράμμισης (Parameterisable Alignment Methods, PAM), και αποδέχονται αυτές τις παραμέτρους. Αρχικά, δεδομένου ενός συστήματος συσχέτισης, πχ. το Prompt, ένα PAM αρχικοποιείται με αυτό. Στη συνέχεια, αφού προκύψει μια αρχική συσχέτιση, η συσχέτιση αυτή γίνεται αποδεκτή ή απορρίπτεται από τους χρήστες. Τελικά, αναλύοντας την επικυρωμένη συσχέτιση και τις προηγούμενες παραμέτρους, με τη βοήθεια τεχνικών μηχανικής μάθησης όπως decision tree learner, νευρωνικά δίκτυα, support vector machines του WEKA περιβάλλοντος μηχανικής μάθησης[13], ένα σύστημα στάθμισης και κατωφλίων παράγεται για τη λειτουργία της συσχέτισης. Η διαδικασία αυτή μπορεί να είναι επαναλαμβανόμενη.

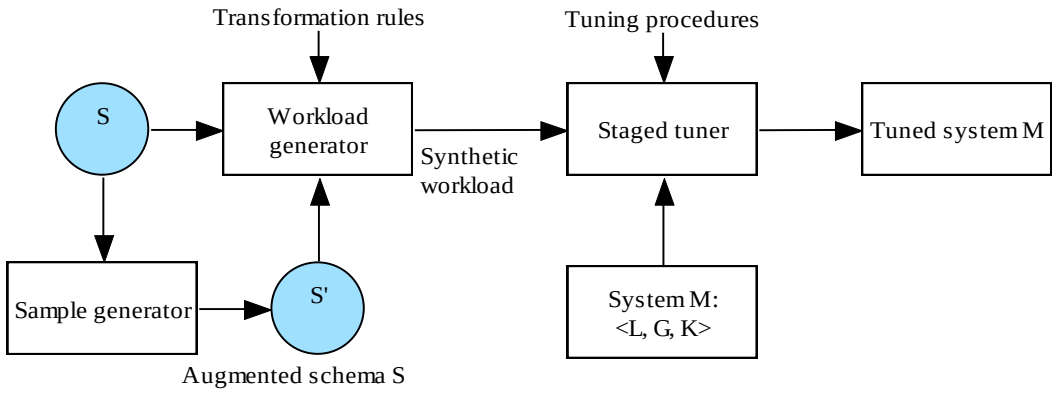
eTuner (University of Illinois and The MITRE Corporation)

Το eTuner [122] είναι ένα σύστημα, το οποίο δεδομένης μιας συγκεκριμένης λειτουργίας συσχέτισης, συντονίζει αυτόματα το σύστημα συσχέτισης σχημάτων

13 http://www.cs.waikato.ac.nz/ml/weka/

(για τον υπολογισμό μια-προς-μια ευθυγραμμίσεων). Για το σκοπό αυτό, επιλέγει τους περισσότερο αποτελεσματικούς βασικούς συσχετιστές και τις βέλτιστες παραμέτρους, πχ. κατώφλια. Το eTuner μοντελοποιεί ένα σύστημα συσχέτισης στη μορφή της τριπλέτας <L, G, K>, όπου:

- L είναι μια βιβλιοθήκη συστατικών συσχέτισης, που περιλαμβάνει τους βασικούς συσχετιστές, πχ. επεξεργασία απόσταση, n-grams, συνδυαστές (πχ. υπολογισμός μέσου όρου, ελάχιστα και μέγιστα αποτελέσματα των βασικών συσχετιστών), εφαρμογές περιορισμών (πχ. προκαθορισμένοι περιορισμοί τομέα ή ευριστικές που η εφαρμογή τους είναι ακριβή ώστε να χρησιμοποιηθούν ως βασικοί συσχετιστές) και επιλογές συσχετιστών (πχ. τμήματα που εφαρμόζουν κατώφλια) για τον καθορισμό της τελικής ευθυγράμμισης.
- G είναι ένας κατευθυνόμενος γράφος που κωδικοποιεί τη ροή εκτέλεσης μεταξύ των συστατικών του δεδομένου συστήματος συσχέτισης.
- K είναι ένα σύνολο από κόμβους που πρέπει να τεθούν (σχηματισμός κόμβων, knob configuration). Τα συστατικά συσχέτισης θεωρούνται ως μαύρα κουτιά που εκθέτουν ένα σύνολο από κόμβους, όπως κατώφλια, βάρη ή συντελεστές.



Εικόνα 3.7.Η αρχιτεκτονική του eTuner: Το eTuner παράγει ένα σύνολο σχημάτων για να συσχετιστούν με ένα αρχικό σχήμα. Στη συνέχεια, παράγει ένα πλάνο για τις παραμέτρους εκμάθησης. Τέλος, ρυθμίζει τις παραμέτρους της μεθόδου και τις παραμέτρους συνάθροισης.

Το σύστημα λειτουργεί σε δύο φάσεις (Εικόνα 3.7). Κατά τη διάρκεια της πρώτης φάσης, δεδομένου ενός σχήματος S, το σύστημα συνθέτει διαφορετικά σχήματα (S_1, S_2, …, S_n) από το S αλλάζοντας το (για παράδειγμα, αλλάζοντας τα ονόματα των

γνωρισμάτων). Έτσι, εξετάζοντας ένα σύνολο από ζευγάρια {<S, S_1>, <S, S_2>, …, <S, S_n>} με τις διαθέσιμες συσχετίσεις τους, μπορεί να υπολογιστεί το F-measure πάνω από κάθε σχηματισμό κόμβων. Η δεύτερη φάση στηρίζεται στην αναζήτηση των βέλτιστων παραμέτρων. Δεδομένου ότι ο χώρος των σχηματισμών κόμβων μπορεί να είναι πολύ μεγάλος, το σύστημα χρησιμοποιεί μια ακολουθιακή, άπληστη μέθοδο, γνωστή ως σταδιακό συντονισμό (staged tuning). Συγκεκριμένα, συντονίζει αρχικά κάθε έναν από τους βασικούς συσχετιστές ξεχωριστά. Μετά, συντονίζει το συνδυασμό των βασικών συσχετιστών και τον συνδυαστή τους κ.ο.κ. Αφού συντονιστεί ολόκληρο το σύστημα, μπορεί να εφαρμοστεί στη συσχέτιση του σχήματος S με κάθε επόμενο σχήμα.

3.2 Ανακεφαλαιωτικά Σχόλια

Συνολικά, μπορούμε να παρατηρήσουμε τα ακόλουθα από τα συστήματα που παρουσιάστηκαν στο κεφάλαιο αυτό:

- Με βάση το πλήθος των συστημάτων που παρουσιάστηκαν στο τρέχον κεφάλαιο, μπορούμε να συμπεράνουμε ότι οι βασισμένες στο σχήμα προσεγγίσεις συσχέτισης έχουν εξεταστεί περισσότερο σε σχέση με αυτές που είναι βασισμένες στα στιγμιότυπα. Πρόκειται για αντικειμενική τάση, δεδομένου ότι προσπαθήσαμε να καλύψουμε τα περισσότερα συστήματα χωρίς να λάβουμε υπόψη μας συγκεκριμένες λύσεις.
- Τα περισσότερα από τα συστήματα που παρουσιάστηκαν, εστιάζουν σε εξειδικευμένες περιοχές ενδιαφέροντος, όπως βιβλία και μουσική, καθώς επίσης χειρίζονται συγκεκριμένα είδη σχημάτων, όπως DTDs, σχεσιακά σχήματα και OWL οντολογίες. Λίγα μόνο συστήματα αποσκοπούν στη γενική εφαρμογή της λύσης, δηλαδή, να εφαρμόζονται σε διαφορετικούς τομείς και στη γενικότητα της εισόδου, δηλαδή να χειρίζονται πολλαπλούς τύπους σχημάτων. Μερικά παραδείγματα συστημάτων που χειρίζονται διαφορετικά είδη σχημάτων είναι τα Cupid, COMA, COMA++ και S-Match.
- Οι περισσότερες των προσεγγίσεων δέχονται ως είσοδο ένα ζεύγος σχημάτων, ενώ λίγα συστήματα δέχονται πολλαπλά σχήματα.
- Οι περισσότερες των προσεγγίσεων χειρίζονται μόνο δενδρικές δομές, ενώ λίγα συστήματα γράφους. Μερικά συστήματα που χειρίζονται γράφους είναι τα Cupid, COMA, COMA++ και OLA.

- Η πλειοψηφία των συστημάτων εστιάζει στον εντοπισμό μια-προς-μια συσχετίσεων, ενώ λίγα συστήματα επιχειρούν να προτείνουν περισσότερο σύνθετες συσχετίσεις, όπως 1-n ή n-m, όπως το iMAP.
- Τα περισσότερα συστήματα εστιάζουν στον υπολογισμό μετρικών ομοιότητας στο εύρος [0, 1], που αντιπροσωπεύουν τη σχέση ισότητας μεταξύ των οντοτήτων των σχημάτων. Λίγα συστήματα υπολογίζουν λογικές σχέσεις μεταξύ των οντοτήτων των σχημάτων, όπως ισότητα. Παράδειγμα των συστημάτων της δεύτερης περίπτωσης αποτελεί το S-Match.

Ο Πίνακας 3.1 συνοψίζει τους βασικούς συσχετιστές που χρησιμοποιήθηκαν σε διάφορα συστήματα. Για παράδειγμα, το S-Match αξιοποιεί βασισμένους σε αλφαριθμητικά συσχεστιστές σε επίπεδο σχήματος, εξωτερικούς συσχετιστές βασισμένους στο WordNet, τεχνικές ελέγχου ικανοποιησιμότητας, κ.ο.κ., το OLA, με τη σειρά του, αξιοποιεί εκτός από βασισμένους σε αλφαριθμητικά συσχεστιστές σε επίπεδο σχήματος, και ένα συσχετιστή βασισμένο στο WordNet, επαναληπτικό υπολογισμό σταθερού σημείου κτλ. Ο Πίνακας 3.1, επίσης, αποδεικνύει ότι η έρευνα στη συσχέτιση σχημάτων εστιάζει κυρίως σε συντακτικές και εξωτερικές τεχνικές. Στην πραγματικότητα, πολλά συστήματα βασίζονται στις ίδιες τεχνικές σύγκρισης αλφαριθμητικών. Η ίδια παρατήρηση ισχύει και για τη χρήση του WordNet ως την εξωτερική πηγή γνώσης. Επίσης, έχουν αξιοποιηθεί και σημασιολογικές τεχνικές, όπως στο S-Match. Όσο αφορά τα συστήματα που βασίζονται στην πληροφορία των στιγμιοτύπων, οι πιο πολλά υποσχόμενες τεχνικές είναι ο naïve Bayes classifier και η χρήση κοινών προτύπων τιμών.

	Τεχνικές Επιπέδου Στοιχείου	**Εξωτερικές Πηγές**	**Τεχνικές Επιπέδου Δομής**	**Σημασιολογία**
DELTA	Επεξεργασία αλφαριθμητικών	-	-	-
DIKE	Επεξεργασία αλφαριθμητικών, έλεγχος συμβατότητας τομέα	Λεξικό WordNet	Συσχέτιση γειτονικών κόμβων	-
Artemis	Έλεγχος συμβατότητας	Θησαυρός	Συσχέτιση γειτονικών	-

	τομέα, Επεξεργασία γλώσσας		κόμβων μέσω θησαυρού, συσταδοποίηση	
Anchor-Prompt	Επεξεργασία αλφαριθμητικών, έλεγχος τομέα και εύρους τιμής	-	Συσχέτιση οριοθετημένων μονοπατιών: (αυθαίρετοι σύνδεσμοι), Δομή Ταξονομίας	-
OntoBuilder	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας	Αναζήτηση σε θησαυρό	-	-
Cupid	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας, τύποι δεδομένων, ιδιότητες κλειδιών	Βοηθητικός θησαυρός	Συσχέτιση δέντρων σταθμισμένη μέσω φύλλων	-
COMA and COMA++	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας, τύποι δεδομένων	Βοηθητικός θησαυρός, επαναχρησιμοποίηση συσχετίσεων, αποθήκευση δομών	Συσχέτιση κατευθυνόμενων άκυκλων γράφων	-
Similarity flooding	Επεξεργασία αλφαριθμητικών, τύποι δεδομένων, ιδιότητες κλειδιών	-	Επαναληπτικός υπολογισμός σταθερού σημείου	-
MapOnto	-	Εξωτερικές συσχετίσεις	Σύγκριση δομής	-

OntoMerge	-	Εξωτερικές συσχετίσεις	-	-
S-Match	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας	WordNet	-	Έλεγχος ικανοποιησιμότητας (SAT)
LSD / GLUE / iMAP	WHIRL, Naïve Bayes	Περιορισμοί τομέα	Ιεραρχική δομή	-
Automatch	Naïve Bayes	-	Εσωτερική δομή, στατιστικά	-
SEMINT	Νευρωνικά δίκτυα, τύποι δεδομένων, πρότυπα τιμών	-	-	-
Clio	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας, Naïve Bayes	-	Σύγκριση δομής	-
OLA	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας, τύποι δεδομένων	WordNet	Επαναληπτικός υπολογισμός σταθερού σημείου, συσχέτιση γειτονικών κόμβων, δομή ταξονομίας	-
Corpus-based matching	Επεξεργασία αλφαριθμητικών, επεξεργασία γλώσσας, Naïve Bayes, πρότυπα τιμών	Συλλογή σχημάτων, περιορισμοί τομέα	-	-

Πίνακας 3.1 Βασικοί συσχετιστές που χρησιμοποιούνται από διαφορετικά συστήματα

Ο Πίνακας 3.1 συνοψίζει τα παραπάνω συστήματα ως προς τις προδιαγραφές που πρέπει να πληρούν. Η στήλη *Είσοδος* αντιπροσωπεύει την είσοδο που δέχονται τα συστήματα. Συγκεκριμένα, περιέχει τις γλώσσες των σχημάτων που δέχονται τα συστήματα (αν η πληροφορία αυτή δεν είναι διαθέσιμη τότε χρησιμοποιείται η γενική ορολογία της βάσης δεδομένων και της οντολογίας). Η πληροφορία αυτή είναι απαραίτητη για κάποιον που επιθυμεί να συσχετίσει ένα συγκεκριμένο είδος σχήματος και αναζητά το κατάλληλο εργαλείο. Η στήλη *Απαιτήσεις* περιγράφει τις πηγές που είναι απαραίτητες για τη λειτουργία του εκάστοτε συστήματος. Στην περίπτωση που αναγράφεται *Χρήστης*, υποδηλώνεται ότι είναι απαραίτητη η ανάδραση του χρήστη, *Ημιαυτόματα* όταν το σύστημα αξιοποιεί την ανάδραση του χρήστη αλλά μπορεί να λειτουργήσει και χωρίς αυτήν, *Αυτόματα* όταν το σύστημα λειτουργεί χωρίς την παρέμβαση του χρήστη (φυσικά, οι χρήστες μπορούν να επηρεάσουν το σύστημα παρέχοντας την αρχική είσοδο ή αξιολογώντας τα τελικά αποτελέσματα, αλλά αυτό δεν εξετάζεται σε αυτό το σημείο). Ομοίως, η τιμή *Στιγμιότυπα* προσδιορίζει ότι το σύστημα απαιτεί στιγμιότυπα δεδομένων για να λειτουργήσει. Επιπροσθέτως, κάποια συστήματα προϋποθέτουν *εκπαίδευση* για τη λειτουργία τους καθώς επίσης και *ευθυγράμμιση* για τη βελτίωση τους. Η στήλη της *Εξόδου* υποδηλώνει τη μορφή των αποτελεσμάτων που επιστρέφει το σύστημα: *Ευθυγράμμιση* σημαίνει ότι το σύστημα επιστρέφει ένα σύνολο από αντιστοιχίες, *συνένωση* ότι συνενώνει τα σχήματα που δόθηκαν στην είσοδο, *αξιώματα* ή *κανόνες* ότι παρέχει κανόνες για την επερώτηση η την ολοκλήρωση των οντολογιών, κλπ.

Σύστημα	Είσοδος	Απαιτήσεις	Έξοδος	Λειτουργία
DELTA	Σχεσιακό σχήμα, EER	Χρήστης	Συσχέτιση	-
DIKE	ER	Ημιαυτόματα	Συνένωση	Μετάφραση δεδομένων
Artemis	Σχεσιακό σχήμα, OO, ER	Αυτόματα	Όψεις	Διαμεσολάβηση επερωτήσεων
Anchor-Prompt	OWL, RDF	Χρήστης	Αξιώματα (OWL/RDF)	Συνένωση οντολογιών
OntoBuilder	Φόρμες ιστού, XML σχήμα	Χρήστης	Διαμεσολαβητής	Διαμεσολάβηση επερωτήσεων
Cupid	XML σχήμα, σχεσιακό σχήμα	Αυτόματα	Συσχέτιση	-
COMA & COMA++	Σχεσιακό σχήμα, XML σχήμα, OWL	Χρήστης	Συσχέτιση	Μετάφραση δεδομένων
Similarity flooding	XML σχήμα, σχεσιακό σχήμα	Χρήστης	Συσχέτιση	-
MapOnto	Σχεσιακό σχήμα, XML σχήμα, OWL	Συσχέτιση	Κανόνες	Μετάφραση δεδομένων
OntoMerge	OWL	Συσχέτιση	Οντολογία	Συνένωση οντολογιών
S-Match	Κατηγοριοποίηση, XML σχήμα, OWL	Αυτόματα	Συσχέτιση	-
LSD / GLUE	Σχεσιακό σχήμα, XML σχήμα, ταξονομία	Αυτόματα, στιγμιότυπα, εκπαίδευση	Συσχέτιση	-
iMAP	Σχεσιακό σχήμα	Αυτόματα, στιγμιότυπα, εκπαίδευση	Συσχέτιση	-
Automatch	Σχεσιακό σχήμα	Αυτόματα, στιγμιότυπα, εκπαίδευση	Συσχέτιση	-
SEMINT	Σχεσιακό σχήμα	Αυτόματα,	Συσχέτιση	-

		στιγμιότυπα (προαιρετικά), εκπαίδευση		
Clio	Σχεσιακό σχήμα, XML σχήμα	Ημιαυτόματα, στιγμιότυπα (προαιρετικά)	Μετασχηματισμός επερωτήσεων	Μετάφραση δεδομένων
OLA	RDF, OWL	Αυτόματα, στιγμιότυπα (προαιρετικά)	Συσχέτιση	-
Corpus-based matching	Σχεσιακό σχήμα, φόρμες ιστού	Σώματα κειμένου, στιγμιότυπα, εκπαίδευση	Συσχέτιση	-

Πίνακας 3.2 Περιγραφή των συστημάτων ως προς τις απαιτήσεις που πρέπει να πληρούν

Η *Έξοδος* που προκύπτει από ένα σύστημα παρουσιάζει αυξημένη σημαντικότητα επειδή αποδεικνύει την ικανότητα του να χρησιμοποιηθεί από άλλες εφαρμογές, πχ. ένα σύστημα που εξάγει όψεις και μεταφραστές δεδομένων δεν μπορεί να χρησιμοποιηθεί για τη συνένωση οντολογιών χωρίς καμία παραλλαγή. Είναι αξιοσημείωτο ότι πολλά συστήματα εξάγουν ευθυγραμμίσεις. Έτσι, τα συστήματα αυτά δεν επιστρατεύονται για ένα συγκεκριμένο είδος λειτουργίας αλλά μπορεί να χρησιμοποιηθούν από ένα εύρος εφαρμογών. Όμως, εξαιτίας της έλλειψης μιας κοινής μορφοποίησης της ευθυγράμμισης, κάθε σύστημα χρησιμοποιεί το δικό του τρόπο για την εξαγωγή ευθυγραμμίσεων (πχ. λίστες από URIs, πίνακες, κλπ.). Τελικά, η στήλη *Λειτουργία* περιγράφει τους τρόπους με τους οποίους ένα σύστημα μπορεί να επεξεργαστεί τις ευθυγραμμίσεις.

Στον Πίνακα 3.2 δεν περιγράφονται όλες οι απαιτήσεις. Κανένα σύστημα δεν μπορεί να αποδειχτεί ως πλήρες, ορθό ή το ταχύτερο δυνατό για τη συσχέτιση σχημάτων. Συνεπώς, ο βαθμός πληρότητας των απαιτήσεων αυτών αξιολογείται και συγκρίνεται με τα υπόλοιπα συστήματα. Επιπλέον, διαφορετικές εφαρμογές έχουν διαφορετικές προτεραιότητες ως προς τις απαιτήσεις αυτές, άρα, μπορεί να

χρησιμοποιούν και διαφορετικά συστήματα. Έτσι, η αξιολόγηση αυτή εξαρτάται από την εφαρμογή στην οποία χρησιμοποιείται ένα σύστημα.

Η περιοχή της συσχέτισης σχημάτων αποτελεί πηγή έρευνας για περισσότερο από 20 χρόνια. Η αύξηση του όγκου της διαθέσιμης πληροφορίας, οι ετερογενείς βάσεις δεδομένων και ο δραματικός πολλαπλασιασμός των δομημένων και αδόμητων δεδομένων στον Ιστό, έχουν το καθένα χωριστά αυξημένη σημασία στην ανάπτυξη αποτελεσματικών λύσεων συσχέτισης σχημάτων.

Έχει πραγματοποιηθεί αξιοσημείωτη επιτυχία στην ανάπτυξη μεθόδων συσχέτισης σχημάτων και κάποιο τμήμα της έρευνας αυτής παρουσιάστηκε παραπάνω προβάλλοντας την ικανότητα της σωστής συσχέτισης στοιχείων μεταξύ δύο σχημάτων, πηγής και προορισμού, της τάξης του 70-90%. Οι περισσότερο επιτυχημένες μέθοδοι χρησιμοποιούν το μεγαλύτερο τμήμα της διαθέσιμης πληροφορίας, συμπεριλαμβανομένης της δομής των δεδομένων, τα στοιχεία, τα δεδομένα στιγμιοτύπου και προηγούμενες συσχετίσεις.

Όμως, αρκετές σημαντικές προκλήσεις παραμένουν και ταλανίζουν τον ερευνητή. Οι πιο αποτελεσματικές συσχετίσεις σχημάτων πραγματοποιούνται όταν συμμετέχουν σχήματα από ένα σαφώς συγκεκριμένο και εξειδικευμένο τομέα, ή από εφαρμογές που δε χρησιμοποιούνται για μεγάλου μεγέθους σχήματα, που υπάρχουν στον πραγματικό κόσμο. Αυτό έχει ως αποτέλεσμα, την αύξηση της έμφασης που πρέπει να δοθεί στην προτυποποίηση και αξιολόγηση των μεθόδων συσχέτισης σχημάτων και στη βελτίωση της συσχέτισης μεγαλύτερων και δυναμικών σχημάτων. Συνολικά, λοιπόν, απαιτείται περαιτέρω έρευνα έτσι ώστε να πραγματοποιηθεί η πλήρως αυτόματη συσχέτιση σχημάτων.

Κεφάλαιο 4. Παρουσίαση Σημασιολογικού Αλγορίθμου Συσχέτισης

Στο σημείο αυτό της μεταπτυχιακής εργασίας περιγράφεται εκτενώς ο αλγόριθμος που εκπονήθηκε. Προσεγγίζοντας τον επιδερμικά, πρόκειται για ακόμη μια πρόταση από την πληθώρα αυτών που έχουν κληθεί να αντιμετωπίσουν το πρόβλημα της συσχέτισης δεδομένων. Επί της ουσίας, όμως, ο σημασιολογικός αυτός αλγόριθμος, που συσχετίζει δύο σχήματα ίδιου ή διαφορετικού τύπου, είναι ο πυρήνας ενός συστήματος που προσφέρει γρήγορα και ακριβή αποτελέσματα στο χρήστη του καθώς επίσης και διαισθητικότητα και ευκολία στο χειρισμό τους. Η διεπαφή του συστήματος αυτού αναλύεται περαιτέρω στο επόμενο κεφάλαιο.

Στη συνέχεια αυτού του κεφαλαίου περιγράφεται ο αλγόριθμος στα πλαίσια της εισόδου που δέχεται, των λειτουργιών που εκτελεί και των αποτελεσμάτων που επιστρέφει. Στο δεύτερο τμήμα του κεφαλαίου αξιολογείται ο αλγόριθμος υπό το πρίσμα των κριτηρίων ποιότητας συσχετίσεων, της προσπάθειας που δαπανήθηκε και της χρονικής διάρκειας. Το τρίτο κεφάλαιο ολοκληρώνεται με τα ανακεφαλαιωτικά σχόλια που συνοψίζουν την παρουσίαση του σημασιολογικού αλγορίθμου συσχέτισης.

```
Είσοδος: 2 Σχήματα S1, S2 (Σχεσιακές Βάσεις, XDR, XSD, OWL, RDF)
σε μορφή κατευθυνόμενου άκυκλου γράφου
Έξοδος: Πίνακας Συχσετίσεων μεταξύ 2 σχημάτων
Λειτουργία: (Ψευδοκώδικας)
(1): Πίνακας_τιμών_ομοιότητας := Λεξικογραφική αναζήτηση(S1, S2)
(2):Για κάθε S1.στοιχείο{
(3):      πάρε τα attributes του, attr1, μεγέθους size1
(4):      Αν size1 != 0{
(5):            Για κάθε S2.στοιχείο{
(6):                  πάρε τα attributes του, attr2, μεγέθους size2
(7):                  Αν size2 != 0{
(8):                        diff := size2 – size1
(9):                        Αν diff ≤ limit1{
(10):                             Για κάθε attr1{
(11):                                   Όσο (υπάρχει attr2 && flag = false){
(12):                                         Αν (Πίνακας_τιμών_ομοιότητας[][] >
thres){
(13):                                               commonAttrs ++
```

```
(14):                                                   flag = true
(15):                                               }
(16):                                           }
(17):                                   }
(18):                                   Πίνακας_σημασιολογικής_ομοιότητας[][] :=
(19):                                       2*commonAttrs / (size1 + size2)
(20):                                   commonAttrs := 0
(21):                           }
(22):                           Αλλιώς      Πίνακας_σημασιολογικής_ομοιότητας[][] := 0
(23):                   }
(24):                   Αλλιώς      Πίνακας_σημασιολογικής_ομοιότητας[][] := 0
(25):           }
(26):   }
(27):   Αλλιώς
(28):           Για κάθε S2.στοιχείο
(29):                   Πίνακας_σημασιολογικής_ομοιότητας[][] := 0
(30):}
(31): Για κάθε S1.στοιχείο{
(32):   πάρε τα παιδιά του, children1, μεγέθους size1
(33):   Αν size1 != 0{
(34):           Για κάθε S2.στοιχείο{
(35):                   Αν (Πίνακας_τιμών_ομοιότητας[][] > 0.1 ||
(36):                       Πίνακας_σημασιολογικής_ομοιότητας[][] > 0.1) &&
(37):                       (Πίνακας_τιμών_ομοιότητας[][] < 0.9 ||
(38):                       Πίνακας_σημασιολογικής_ομοιότητας[][] < 0.9){
(39):                       πάρε τα παιδιά του, children2, μεγέθους size2
(40):                       Αν size2 != 0{
(41):                           diff := size2 – size1
(42):                           Αν diff ≤ limit2{
(43):                               Για κάθε children 1{
(44):                                   Όσο (υπάρχει children 2 && flag = false){
(45):                                       Αν
                                        ((Πίνακας_τιμών_ομοιότητας[][] +
(46):                                   Πίνακας_σημασιολογικής_ομοιότητας[][]) / 2 >
                                        thres){
(47):                                           commonChildren ++
(48):                                           flag = true
(49):                                           }
(50):                                   }
(51):                               }
(52):                               Πίνακας_συντακτικής_ομοιότητας[][] :=
(53):                                   2*commonChildren / (size1 + size2)
(54):                               commonChildren := 0
```

```
(55):                                   }
(56):                                   Αλλιώς          Πίνακας_ συντακτικής _ομοιότητας[][] :=
0
(57):                               }
(58):                               Αλλιώς          Πίνακας_ συντακτικής _ομοιότητας[][] := 0
(59):                       }
(60):                       Αλλιώς
(61):                       Πίνακας_ συντακτικής _ομοιότητας[][] :=
(Πίνακας_τιμών_ομοιότητας[][] +
(62):                                   Πίνακας_σημασιολογικής_ομοιότητας[][]) / 2
(63):               }
(64):    }
(65):    Αλλιώς
(66):               Για κάθε S2.στοιχείο
(67):                       Πίνακας_ συντακτικής _ομοιότητας[][] := 0
(68):}
(69):Επιστροφή Πίνακας_συντακτικής_ομοιότητας
```

Εικόνα 4.19Σημασιολογικός αλγόριθμος συσχέτισης σχημάτων

4.1 Περιγραφή – Δομή

Στην ενότητα αυτή αναλύεται διεξοδικά ο αλγόριθμος, αρχικά, από την σκοπιά της εισόδου, των λειτουργιών και της εξόδου. Σε αυτό το σημείο της εργασίας θα πρέπει να διευκρινιστεί ότι πρόκειται για υβριδικό αλγόριθμο συσχέτισης σχημάτων αφού συνδυάζει όλες τις στρατηγικές που έχουν προταθεί ως τώρα. Δηλαδή, βασίζεται στη λεξικογραφική ομοιότητα των όρων και επεκτείνει τη σύγκριση με βάση σημασιολογικά και συντακτικά χαρακτηριστικά των σχημάτων. Δεδομένου, λοιπόν, ότι δεν υλοποιεί μια αμιγή προσέγγιση επίλυσης του προβλήματος, χαρακτηρίζεται ως υβριδικός. Πριν την αναλυτική όμως περιγραφή του αλγορίθμου προηγείται μια συντομότερη και προσανατολισμένη στη μορφή ψευδοκώδικα. Στο παράρτημα, που βρίσκεται στο τέλος της εργασίας, υπάρχουν τμήματα του κώδικα που συνετέλεσε στην υλοποίηση του αλγορίθμου.

4.1.1 Περιγραφή Εισόδου

Όπως προαναφέρθηκε ο σημασιολογικός αλγόριθμος συσχέτισης δύναται να συγκρίνει και να συσχετίσει σχήματα διαφορετικού τύπου. Τα σχήματα που δίνονται ως είσοδος στον αλγόριθμο μπορεί να είναι XML (xdr και xsd), οντολογίες (owl και rdf) και σχήματα σχεσιακών βάσεων.

Αξίζει να σημειωθεί ότι πριν την εκτέλεση του αλγορίθμου θα πρέπει να γίνει μια προεπεξεργασία όσο αφορά στην είσοδο που δέχεται. Αφού καθορίσει ο χρήστης του συστήματος την είσοδο για την πυροδότηση του αλγορίθμου, τα σχήματα ανακτώνται από τη βάση του συστήματος ως άκυκλοι γράφοι και στην περίπτωση που δεν υπάρχουν τότε μετατρέπονται σε άκυκλους γράφους και αποθηκεύονται στη βάση. Στην περίπτωση που το προς συσχέτιση σχήμα παρουσιάζει κύκλο, τότε ο κύκλος αυτός σπάει και αποθηκεύεται στη βάση ως άκυκλο. Αφού ολοκληρωθεί αυτή η προεργασία και έχουν φορτωθεί τα σχήματα, μπορεί να εκκινήσει ο αλγόριθμος για την πρόταση των συσχετίσεων.

4.1.2 Περιγραφή Λειτουργίας

Εφαλτήριο του αλγορίθμου σημασιολογικής συσχέτισης υπήρξε η μεταπτυχιακή εργασία του Δ. Μανακανάτα που, επί της ουσίας, εκτελεί λεξικογραφική συσχέτιση μεταξύ δύο σχημάτων. Εν συντομία, ο αλγόριθμος αυτός δέχεται ως είσοδο δύο σχήματα και αφού τα μετατρέψει σε άκυκλους γράφους τα συγκρίνει στοιχείο προς στοιχείο. Για τις ανάγκες του γράφου χρησιμοποιεί μια βάση και εξετάζει αν υπάρχει κάποια ήδη καταχωρημένη συσχέτιση. Αν υπάρχει και ικανοποιεί τα κριτήρια του χρήστη τότε επιστρέφεται αυτή η καταχώρηση, διαφορετικά το αποτέλεσμα προκύπτει με βάση το λεξικό Wordnet. Αφού ολοκληρωθεί ο αλγόριθμος, επιστρέφει έναν πίνακα τιμών ομοιότητας που περιέχει όλους τους δυνατούς συνδυασμούς όρων των δύο σχημάτων και την τιμή ομοιότητας τους.

Παρακάτω αναλύεται ο σημασιολογικός αλγόριθμος και μπορεί να διαιρεθεί σε δύο τμήματα, τη σημασιολογική συσχέτιση που αξιοποιεί την πληροφορία των χαρακτηριστικών κάθε όρου και τη δομική συσχέτιση που εκμεταλλεύεται τις σχέσεις κληρονομικότητας μεταξύ όρων. Σε αυτό το σημείο πρέπει να διευκρινιστεί ότι δεν πρόκειται για εναλλακτικές προτάσεις στην επίλυση του προβλήματος αλλά για διαδοχικές διαδικασίες που η επόμενη αξιοποιεί τα αποτελέσματα της προηγούμενης.

Κατά την έναρξη του αλγορίθμου εκτελείται η λεξικογραφική συσχέτιση μεταξύ των σχημάτων, όπως προαναφέρθηκε. Στο σημείο αυτό πυροδοτείται η σημασιολογική συσχέτιση. Ο αλγόριθμος σαρώνει όλα τα στοιχεία του πρώτου σχήματος, σχήμα-πηγή S1, και τα συγκρίνει με όλα τα στοιχεία του δεύτερου, σχήμα-στόχος S2. Για κάθε κόμβο του σχήματος πηγή, ανακτά τα χαρακτηριστικά του και στην περίπτωση που αυτά υπάρχουν τότε μπορεί να συνεχιστεί η

διαδικασία, πλήθους Ν. Ακολουθείται η ίδια διαδικασία και για τον εκάστοτε κόμβο του σχήματος-στόχου και συλλέγονται τα χαρακτηριστικά του, πλήθους Μ. Στη συνέχεια υπολογίζεται η διαφορά μεταξύ των συνόλων, αφού η σύγκριση μεταξύ όρων που έχουν σημαντική διαφορά ως προς την περιγραφή τους δεν μπορεί να εγγυηθεί ακριβές αποτέλεσμα. Για παράδειγμα, αν ο πρώτος κόμβος χαρακτηρίζεται από δύο γνωρίσματα ενώ ο δεύτερος από 20, τότε η σημασιολογική συσχέτιση δεν μπορεί να αξιοποιηθεί. Αφού υπολογιστεί η διαφορά των χαρακτηριστικών και ικανοποιεί κάποιο περιορισμό (limit1) υπολογίζεται το πλήθος των κοινών χαρακτηριστικών μεταξύ των δύο όρων. Στον πίνακα σημασιολογικής ομοιότητας αποθηκεύεται η Dice coefficient που ορίζεται ως εξής:

$$Πίνακας\_σημασιολογικής\_ομοιότητας(S1.στοιχείο, S2.στοιχείο) = \frac{Κοινά\ χαρακτηριστικά}{N + M}$$

Στις περιπτώσεις που δεν εκτελέστηκε ο σημασιολογικός αλγόριθμος εξ ολοκλήρου, για παράδειγμα όταν οι όροι δεν είχαν περιγραφή, το αποτέλεσμα της ομοιότητας καταχωρείται ως μηδενικό. Μετά το πέρας της παραπάνω διαδικασίας, επιστρέφεται ο πίνακας σημασιολογικής συσχέτισης και ο αλγόριθμος προχωράει στο τμήμα της δομικής συσχέτισης.

Στα πλαίσια της δομικής συσχέτισης, σαρώνονται οι κόμβοι της πηγής και συγκρίνονται ως προς την κληρονομικότητα με τους κόμβους του στόχου. Αναλυτικότερα, για κάθε όρο του σχήματος πηγής συγκεντρώνονται οι απόγονοι του, στην περίπτωση που υπάρχουν, και στη συνέχεια σαρώνονται όλοι οι όροι του σχήματος στόχου. Στο σημείο αυτό εξετάζουμε τα δεδομένα που έχουμε στη διάθεση μας από τις προηγούμενες συσχετίσεις και διακρίνουμε τις παρακάτω περιπτώσεις. Όταν τα αποτελέσματα της λεξικογραφικής και της σημασιολογικής συσχέτισης συγκλίνουν στα άκρα του συνόλου τιμών, δηλαδή 0 ή 1, τότε ο περαιτέρω έλεγχος ομοιότητας θεωρείται περιττός φόρτος στην πολυπλοκότητα του αλγορίθμου. Σε όλες τις άλλες περιπτώσεις επιβάλλεται να προχωρήσουμε σε συντακτική συσχέτιση για την εξαγωγή ασφαλούς συμπεράσματος ως προς την ομοιότητα. Σε αυτές τις περιπτώσεις λοιπόν, για κάθε όρο του σχήματος στόχου επιλέγονται οι απόγονοι του και αν τα δύο σύνολα είναι συγκρίσιμα (υπολογίζεται το diff όπως παραπάνω), τότε γίνεται η καταμέτρηση των κοινών απογόνων. Σε αυτό το σημείο της σύγκρισης, αξιοποιούμε τα δεδομένα και των δύο πινάκων, άρα

την ομοιότητα την αποφασίζει ο μέσος όρος της λεξικογραφικής και της σημασιολογικής σύγκρισης. Στις περιπτώσεις που δεν εκτελέστηκε ο συντακτικός αλγόριθμος εξ ολοκλήρου. Μετά το πέρας της παραπάνω διαδικασίας, επιστρέφεται ο πίνακας συντακτικής συσχέτισης και ο αλγόριθμος τερματίζεται μετά το φιλτράρισμα των αποτελεσμάτων σύμφωνα με την επιλογή του χρήστη.

4.1.3 Περιγραφή Εξόδου

Τα αποτελέσματα του αλγορίθμου συνοψίζονται σε έναν πίνακα συσχετίσεων μεταξύ των όρων των σχημάτων που δόθηκαν ως είσοδος. Αρχικά ο πίνακας αυτός περιέχει τις συσχετίσεις που προέκυψαν από τις μετρήσεις για όλους τους δυνατούς συνδυασμούς των όρων των σχημάτων και στη συνέχεια επεξεργάζεται με βάση κάποιο φίλτρο. Το φίλτρο αυτό πρακτικά λειτουργεί σαν μια στρατηγική απαλοιφής αποτελεσμάτων και με αυτόν τον τρόπο παρουσιάζονται τα τελικά αποτελέσματα στο χρήστη. Τα φίλτρα επιλογής περιγράφονται αναλυτικότερα στο επόμενο κεφάλαιο.

4.1.4 Παράδειγμα Χρήσης Αλγορίθμου

Για την καλύτερη κατανόηση του αλγορίθμου, περιγράφεται στη συνέχεια ένα παράδειγμα χρήσης του. Ας υποθέσουμε ότι έχουμε στη διάθεση μας δύο στοιχεία, το *Arts* του σχήματος S1 και το *ArtsHumanities* του σχήματος S2. Το στοιχείο *Arts* περιγράφεται από τα στοιχεία: *Literature, Music, ArtHistory, VisualArts* και το στοιχείο *ArtsHumanities* από τα στοιχεία: *Humanities, DesignArt, ArtHistory*. Επίσης, θεωρούμε ότι το κατώφλι της αποδεκτής ομοιότητας για τα δύο στοιχεία έχει οριστεί στην τιμή 0.7. Αρχικά, η λεξικογραφική σύγκριση μεταξύ των δύο στοιχείων θα επιστρέψει την τιμή 0.65. Στη συνέχεια, ο αλγόριθμος προχωράει υπολογίζοντας τη σημασιολογική ομοιότητα. Έτσι εντοπίζει τα εξής ζεύγη αντιστοιχιών: *Arts.Literature-ArtsHumanities.Humanities, Arts.ArtHistory-ArtsHumanities.ArtHistory* και *Arts.VisualArts-ArtsHumanities.DesignArt*. Η Dice coefficient μεταξύ των στοιχείων Arts και ArtsHumanities αποτιμάται στην τιμή 0,86. Υπολογίζοντας το μέσο όρο των δύο αποτελεσμάτων, προκύπτει ότι η ομοιότητα των στοιχείων είναι 0.76, τιμή μεγαλύτερη από το κατώφλι, άρα προτείνεται η συσχέτιση μεταξύ των στοιχείων *Arts* και *ArtsHumanities*.

4.2 Αξιολόγηση Αλγορίθμου

Στα πλαίσια της αξιολόγησης του αλγορίθμου, λήφθηκε υπόψη η σκοπιά της εισόδου που δέχεται, της εξόδου που παράγεται, κάποια κριτήρια ποιότητας που έχουν καθιερωθεί στον τομέα της συσχέτισης, η προσπάθεια που δαπανήθηκε καθώς επίσης και ο χρόνος που διήρκεσαν οι πειραματικές μετρήσεις. Παρακάτω,

αναπτύσσεται η αξιολόγηση του αλγορίθμου με βάση τα χαρακτηριστικά που προαναφέρθηκαν.

4.2.1 Είσοδος Αλγορίθμου

Ο προτεινόμενος αλγόριθμος, προτείνει συσχετίσεις σχημάτων διαφορετικών μορφών. Μπορεί να συσχετίσει τόσο ομογενή όσο και ετερογενή σχήματα που ανήκουν στο εύρος των σχημάτων σχεσιακών βάσεων, των οντολογιών και XML σχημάτων. Οι οντολογίες μπορεί να είναι σε μορφή .owl και .rdf, ενώ τα XML σχήματα σε .xsd και .xdr.

Σημαντικός παράγοντας που επηρεάζει άμεσα την πολυπλοκότητα του αλγορίθμου της συσχέτισης είναι το μέγεθος των σχημάτων. Όσο μεγαλύτερα είναι τα προς συσχέτιση σχήματα, τόσο αυξάνεται το εύρος των συσχετίσεων που πρέπει να εξεταστούν, επομένως, αυξάνεται δραματικά και ο χρόνος εκτέλεσης του αλγορίθμου.

Επίσης, εκτός των σχημάτων, χρησιμοποιήθηκε ως είσοδος του αλγορίθμου και το λεξικό WordNet, υποβοήθησε τη διαδικασία της λεξικογραφικής συσχέτισης.

4.2.2 Έξοδος Αλγορίθμου

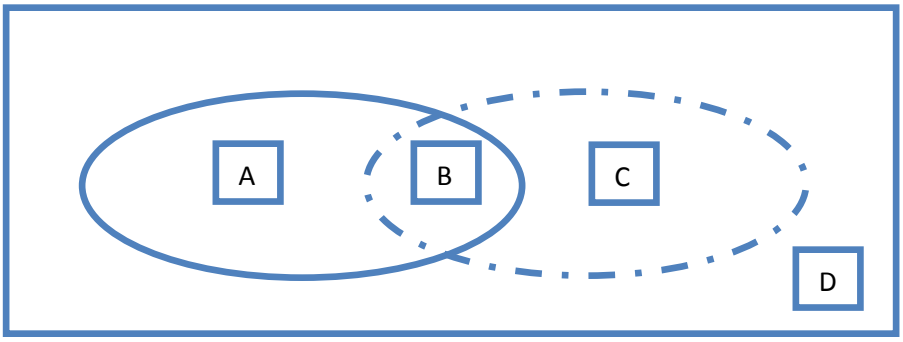
Για την αναπαράσταση των αποτελεσμάτων αντίστοιχων προσεγγίσεων έχουν χρησιμοποιηθεί διαφορετικές τεχνικές. Η πιο συνηθισμένη από αυτές είναι η βαθμολόγηση της συσχέτισης δύο όρων με μια αριθμητική τιμή μεταξύ 0 και 1. Αυτή η μέθοδος έχει ακολουθηθεί και στην προτεινόμενη λύση του προβλήματος. Όμως, αυτή η μετρική δεν είναι αρκετή για την αξιολόγηση των αποτελεσμάτων του αλγορίθμου σημασιολογικής συσχέτισης. Εκτός, λοιπόν από την αριθμητική αυτή σύγκριση για την ποιότητα των αποτελεσμάτων, λήφθηκαν υπόψη και η εσωτερική αναπαράσταση των στοιχείων σχήματος καθώς και η πληθικότητα.

Όσο αφορά την εσωτερική αναπαράσταση ή παρουσίαση των στοιχείων του εκάστοτε σχήματος, χρησιμοποιήθηκαν άκυκλοι γράφοι. Έτσι, κάθε σχήμα, πηγή ή στόχος, μετατρέπεται και αποθηκεύεται ως άκυκλο γράφο. Με τη χρήση του άκυκλου γράφου, τα αποτελέσματα που προκύπτουν μπορεί να είναι συσχετίσεις μεταξύ στοιχείων ή συσχετίσεις μεταξύ μονοπατιών. Η συγκεκριμένη προσέγγιση επιστρέφει συσχετίσεις μεταξύ στοιχείων για λόγους εποπτικότητας.

Σε σχέση με τη δεύτερη παράμετρο, δηλαδή την πληθικότητα, είναι προφανές ότι ένα στοιχείο του σχήματος πηγής μπορεί να συμμετέχει σε μηδέν, μια ή περισσότερες συσχετίσεις με στοιχεία του σχήματος στόχου και γίνεται λόγος για ολική πληθικότητα 1:1, 1:n/n:1 ή n:m. Επίσης, μέσα σε μία συσχέτιση ένα ή περισσότερα στοιχεία από το πρώτο σχήμα μπορεί να συσχετιστούν με ένα ή περισσότερα στοιχεία από το δεύτερο σχήμα και τότε αναφέρεται ως τοπική πληθικότητα 1:1, 1:n/n:1 ή n:m. Η πλειοψηφία των συστημάτων που έχουν προταθεί μέχρι σήμερα περιορίζονται στις 1:1 τοπικές πληθικότητες προτείνοντας τη συσχέτιση με το μεγαλύτερο βαθμό ομοιότητας. Το σύστημα που προτείνεται στην παρούσα ερευνητική εργασία, υποστηρίζει όλων των ειδών τις σχέσεις μεταξύ των στοιχείων, τοπικής και ολικής πληθικότητας, δίνοντας τη δυνατότητα στο χρήστη να επιλέξει την κατάλληλη στρατηγική από τη διεπαφή.

4.2.3 Κριτήρια Ποιότητας Συσχετίσεων

Παρά το γεγονός ότι η χειροκίνητη συσχέτιση αποτελεί την πιο χρονοβόρα, επίπονη και επιρρεπή σε λάθη προσέγγιση, ταυτόχρονα είναι και η πιο ακριβής και ποιοτική λύση, όταν, φυσικά, δε συνοδεύεται από λάθη. Για αυτό το λόγο, η χειροκίνητη εύρεση συσχετίσεων θα λειτουργήσει ως πρότυπο κατά τη διάρκεια της αξιολόγησης των αποτελεσμάτων του αλγορίθμου.



Εικόνα 4.20 Χειροκίνητη – Αυτόματη Συσχέτιση Σχημάτων

Πριν προχωρήσουμε στην αξιολόγηση των ποιοτικών χαρακτηριστικών του αλγορίθμου, θα πρέπει να δοθεί μια επεξήγηση των συνόλων που παρουσιάζονται στην Εικόνα 7. Πρώτα από όλα, το σύνολο A περιέχει τις συσχετίσεις που θα έπρεπε να βρει ο αλγόριθμος αλλά δε βρήκε. Το σύνολο B περιέχει τις σωστές συσχετίσεις που πρότεινε ο αλγόριθμος, ενώ το σύνολο C τις λάθος συσχετίσεις που πρότεινε ο αλγόριθμος. Τέλος το σύνολο D αντιπροσωπεύει τις λάθος

συσχετίσεις που ορθά δεν πρότεινε ο αλγόριθμος. Διαισθητικά, μπορεί να γίνει κατανοητό ότι, τόσο τα Ψεύτικα Σωστά (False Positives) όσο και τα Ψεύτικα Λάθος (False Negatives) μειώνουν την ποιότητα του αποτελέσματος.

Παρακάτω, παρουσιάζονται δύο μετρικές που η εξέταση τους έχει καθιερωθεί στα πλαίσια της αξιολόγησης αντίστοιχων συστημάτων και είναι η Ακρίβεια (Precision) και η Ανάκληση (Recall). Πρόκειται για μετρικές που προέρχονται από τον τομέα της ανάκτησης πληροφοριών και υπολογίζονται ως εξής:

$$Ακρίβεια = \frac{|B|}{|B| + |C|}$$

Η ακρίβεια αντιπροσωπεύει το τμήμα των πραγματικών συσχετίσεων μεταξύ όλων όσων προτάθηκαν από τον αλγόριθμο.

$$Ανάκληση = \frac{|B|}{|A| + |B|}$$

Η ανάκληση προσδιορίζει το τμήμα των πραγματικών συσχετίσεων που βρέθηκαν από τον αλγόριθμο.

Ιδανικά, δηλαδή όταν δεν προτείνονται Ψεύτικα Σωστές και Ψεύτικα Λάθος Συσχετίσεις, τότε $Ακρίβεια = Ανάκληση = 1$. Βέβαια, αυτές οι δύο μετρικές δεν αρκούν για την πλήρη αξιολόγηση των ποιοτικών αποτελεσμάτων του αλγορίθμου. Αυτό συμβαίνει γιατί είναι πολύ εύκολο να μεγιστοποιηθεί η ανάκληση, σε βάρος της ακρίβειας, πχ. όταν επιστραφούν όλες οι πιθανές συσχετίσεις από τον αλγόριθμο. Επίσης, στην περίπτωση που επιστραφούν μόνο λίγες σωστές συσχετίσεις από τον αλγόριθμο, τότε προκαλείται αυξημένη ακρίβεια εις βάρος της χαμηλής ανάκλησης. Όλα τα παραπάνω οδήγησαν στην πρόταση μιας νέας μετρικής, που πρακτικά αποτελεί συνδυασμό της ακρίβειας και της ανάκλησης και είναι η ακόλουθη:

$$FMeasure(a) = \frac{|B|}{(1-a) * |A| + |B| + a * |C|} = \frac{Ακρίβεια * Ανάκληση}{(1-\alpha) * Ακρίβεια + \alpha * Ανάκληση}$$

Αξίζει να σημειωθεί ότι και το παραπάνω μέτρο ανήκει στον τομέα της ανάκτησης πληροφοριών. Διαισθητικά, η παράμετρος a ($0 \leq a \leq 1$), δίνει τη δυνατότητα διαφορετική συσχετιστική αξία να αντιστοιχίζεται με την ακρίβεια και την ανάκληση. Πιο αναλυτικά, όταν το a τείνει στο 1, τότε το $FMeasure(a)$ → Ακρίβεια

και δε συνυπολογίζεται η ανάκληση. Αντίστοιχα, όταν το a τείνει στο 1, τότε το $FMeasure(a)$ → Ανάκληση και δε συνυπολογίζεται η ακρίβεια. Στην περίπτωση που η συμμετοχή της ακρίβειας και της ανάκλησης είναι ισοβαρής, ισχύει το παρακάτω συνδυασμένο μέτρο σύγκρισης:

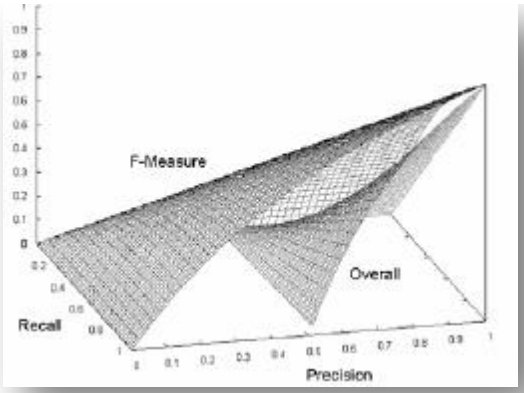
$$FMeasure(a) = \frac{2 * |B|}{(|A| + |B|) + (|B| + |C|)} = 2 * \frac{Ακρίβεια * Ανάκληση}{Ακρίβεια + Ανάκληση}$$

Η παραπάνω μετρική εκφράζει την αρμονική συνύπαρξη της ακρίβειας και της ανάκλησης είναι η πιο κοινή έκφραση του $FMeasure(a)$ στον τομέα της ανάκτησης πληροφορίας.

Η τελευταία μετρική που παρουσιάζεται είναι αυτή του Γενικού-δείκτη:

$$Γενικός - δείκτης = 1 - \frac{|A| + |C|}{|A| + |B|} = \frac{|B| - |C|}{|A| + |B|} = Ανάκληση * (2 - \frac{1}{Ακρίβεια})$$

Για να συγκρίνουμε την συμπεριφορά των δύο μέτρων (FMeasure και Γενικός-δείκτης) η Εικόνα 8 δείχνει αυτά τα μέτρα σαν συνάρτηση της Ακρίβειας και της Ανάκλησης.



Εικόνα 4.3 FMeasure και Γενικός-δείκτης σε συνάρτηση Ακρίβειας και Ανάκλησης

Είναι φανερό ότι το FMeasure είναι πολύ πιο αισιόδοξο σε σχέση με το Γενικό-δείκτη. Για την ίδια τιμή Ακρίβειας και Ανάκλησης, το FMeasure παραμένει πολύ υψηλότερο από το Γενικό-δείκτη. Αντίθετα με τα άλλα μέτρα, ο Γενικός-δείκτης μπορεί να πάρει και αρνητικές τιμές, όταν ο αριθμός των ψεύτικα σωστών

ξεπεράσει τον αριθμό των θετικά σωστών π.χ. Ακρίβεια < 0.5. Και τα δύο συνδυασμένα μέτρα παίρνουν τη μέγιστη τιμή (1.0) όταν Ακρίβεια = Ανάκληση = 1.0. Σε όλες τις άλλες περιπτώσεις, καθώς η τιμή του FMeasure είναι μέσα στα όρια που καθορίζονται από την Ακρίβεια και την Ανάκληση, ο Γενικός-δείκτης είναι μικρότερος από την Ακρίβεια και την Ανάκληση μαζί.

4.2.4 Προσπάθεια που Δαπανήθηκε

Η προσπάθεια που δαπανήθηκε από τον χρήστη κατά τη διάρκεια της χρήσης του συστήματος, αποτελεί τον τέταρτο παράγοντα που πρέπει να εξεταστεί και να συμπεριληφθεί στην αξιολόγηση του. Γενικότερα, αυτό το κριτήριο αξιολόγησης επηρεάζεται και από υποκειμενικούς παράγοντες και γι' αυτό το λόγο, η μέτρηση της χειρωνακτικής εργασίας που απαιτείται δυσχεραίνει τη διαδικασία αξιολόγησης. Δεδομένου ότι η συσχέτιση σχημάτων δεν μπορεί να διεκπεραιωθεί ως πλήρως αυτόματη διαδικασία, η συμμετοχή του ανθρώπου κρίνεται επιτακτική. Όμως, η συμμετοχή του ανθρώπινου παράγοντα, δημιουργεί την ανάγκη της ταχύτητας από την πλευρά του χρήστη, που μεταφράζεται σε απλότητα, αμεσότητα και ευκολία χρήσης του συστήματος.

Η προσπάθεια που πρέπει να καταβληθεί συνοψίζεται στη ρύθμιση των παραμέτρων του αλγορίθμου. Έτσι, ο χρήστης καλείται να επιλέξει τις αριθμητικές παραμέτρους καθώς επίσης και τη στρατηγική επιλογής των αποτελεσμάτων. Όμως, η ρύθμιση των παραμέτρων πριν την εκκίνηση του αλγορίθμου συσχέτισης, δεν είναι το μοναδικό σημείο που επεμβαίνει ο χρήστης. Παρά το γεγονός ότι οι περισσότεροι αλγόριθμοι δε λαμβάνουν υπόψη τους την προσπάθεια που πρέπει να καταβάλει ο χρήστης μετά την εκτέλεση του αλγορίθμου, πρόκειται για καθοριστικό παράγοντα στον υπολογισμό της προσπάθειας. Στα πλαίσια της προσπάθειας αυτής, ο χρήστης εξετάζει και απορρίπτει ή αποδέχεται τα προτεινόμενα, από τον αλγόριθμο, αποτελέσματα.

Εκτός από τα παραπάνω, άλλοι παράγοντες που επηρεάζουν την αποτελεσματικότητα της προσπάθειας του χρήστη, και κατ' επέκταση την αποτελεσματικότητα του ίδιου του συστήματος, είναι το γνωστικό επίπεδο του ατόμου που χειρίζεται το σύστημα, η συνάφεια του με το γνωστικό πεδίο των υπό συσχέτιση σχημάτων καθώς επίσης και η εξοικείωση του με το σύστημα και τις λειτουργίες του.

Τελικά οι απαιτήσεις του τελικού αποτελέσματος του αλγορίθμου εξαρτώνται από την αντίληψη του χρήστη σχετικά με το τι είναι σωστή και τι λάθος συσχέτιση. Έτσι η ποιότητα των αποτελεσμάτων μπορεί να διαφέρει από χρήστη σε χρήστη. Αυτή η ασυνέπεια μπορεί σε κάποιο βαθμό να περιοριστεί μετά από κατάλληλη αξιολόγηση.

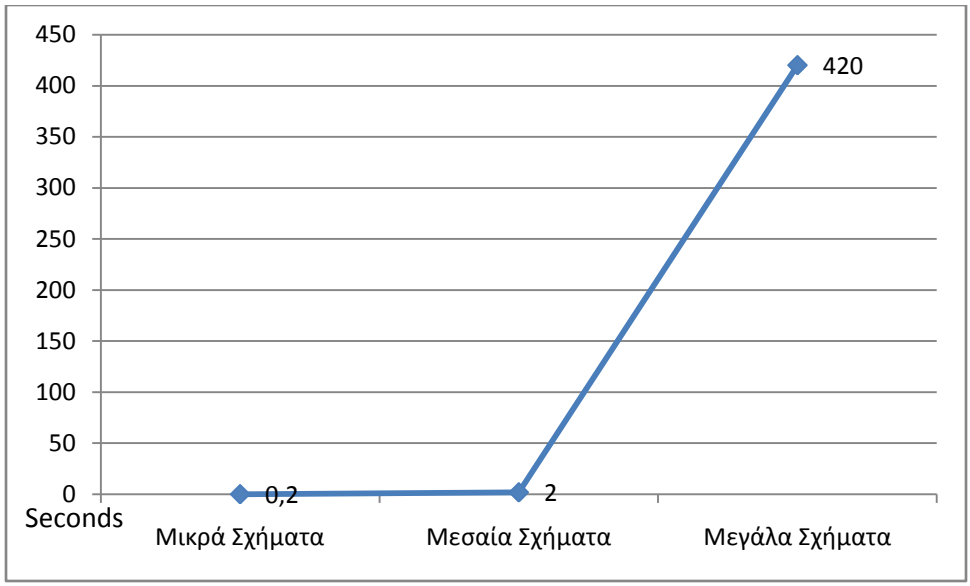
4.2.5 Μέτρηση Χρόνου

Συνήθως, η χρονική διάρκεια της εκτέλεσης ενός αλγορίθμου έχει βαρύνουσα σημασία όταν δεν παρεμβαίνει στη διαδικασία ο ανθρώπινος παράγοντας. Επίσης, σε αυτές τις περιπτώσεις εφαρμογών είναι σημαντική όχι μόνο η ταχύτητα του αλγορίθμου αλλά και η ποιότητα των αποτελεσμάτων. Μια τέτοια εφαρμογή είναι η χρήση του αλγορίθμου σε ομότιμα δίκτυα (peer-to-peer).

Όμως, επειδή δεν είναι σκοπός αυτής της εργασίας η εφαρμογή της σε ανάλογα συστήματα όπως το παραπάνω, δεν μας απασχόλησε άμεσα η ταχύτητα. Προφανώς ελήφθησαν υπόψη οι περιπτώσεις συσχετίσεων που δε χρήζουν εξέτασης, ώστε να μην επιβαρύνεται η πολυπλοκότητα του αλγορίθμου. Επίσης, παρέχεται η δυνατότητα στο χρήστη να θυσιάσει την ποιότητα των αποτελεσμάτων στο βωμό της ταχύτητας. Με άλλα λόγια ο χρήστης του συστήματος δύναται να έχει στη διάθεση του λιγότερο σωστά αποτελέσματα σε συντομότερο χρονικό διάστημα.

Στην Εικόνα 4.4 παρουσιάζεται η χρονική διάρκεια της εκτέλεσης του αλγορίθμου σε σχέση με το μέγεθος των υπό συσχέτιση σχημάτων.

Η μετρήσεις έγιναν σε ένα μηχάνημα με λειτουργικό σύστημα Windows XP, επεξεργαστή Intel Core2 (1,83 GHZ) και μνήμη 1GB. Στο μηχάνημα δεν έτρεχε τίποτα άλλο εκτός από το σημασιολογικό αλγόριθμο που περιγράφηκε πιο πάνω.



Εικόνα 4.21 Χρόνος εκτέλεσης αλγορίθμου

4.3 Πειράματα – Μετρήσεις – Συγκριτικά αποτελέσματα

Στα πλαίσια της αξιολόγησης του αλγορίθμου, επιλέχθηκαν σχήματα που ικανοποιούν όλες τις πιθανές περιπτώσεις. Τα σχήματα αυτά διαφέρουν ως προς το είδος και ως προς το πλήθος των όρων που περιέχουν. Το μέγεθος τον σχημάτων που συμμετείχαν στην αξιολόγηση είναι μεταξύ 2 και 1360 περίπου στοιχείων και περιγράφονται αναλυτικά στον πίνακα, ενώ το είδος τους είναι σχεσιακή βάση δεδομένων (SQL), οντολογία (OWL, RDF) και XML σχήμα (XSD, XDR).

Σχήμα	Κόμβοι	Γνωρίσματα	Σύνδεσμοι/ISA
Images	2	3	1
Europe	3	4	2
Purchase Order 1	3	12	2
Purchase Order 2	5	10	4
Google web directory (mini)	9	13	8
Yahoo web directory (mini)	6	10	5
CIDX Purchase Order	10	32	8
Excel Purchase Order	12	40	10
RDB Schema	20	70	17
Warehouse Star Schema	6	35	7
Cornell University (mini)	6	33	5
Washington University (mini)	8	34	7
Yahoo Finance (mini)	3	9	2
Standard Taxonomy (mini)	3	15	2
Yahoo Finance	~1200	~1450	~500
Standard Taxonomy	~1250	~1350	~400
Company	13	34	12
Company-er	42	99	41
Conference	19	31	10
Ka	~1360	~1600	~661
Dbgroup	24	62	21
Factbook-out	315	402	89
Mondial	27	119	26
Bibliographic-Data	126	230	74
DBLP	12	68	11
targetDBLP	8	19	7
sigmodRecord	6	10	4
Amalgam1	16	116	15
Amalgam2	28	80	27

Πίνακας 4.1 Σχήματα που χρησιμοποιήθηκαν στα πλαίσια της αξιολόγησης του αλγορίθμου

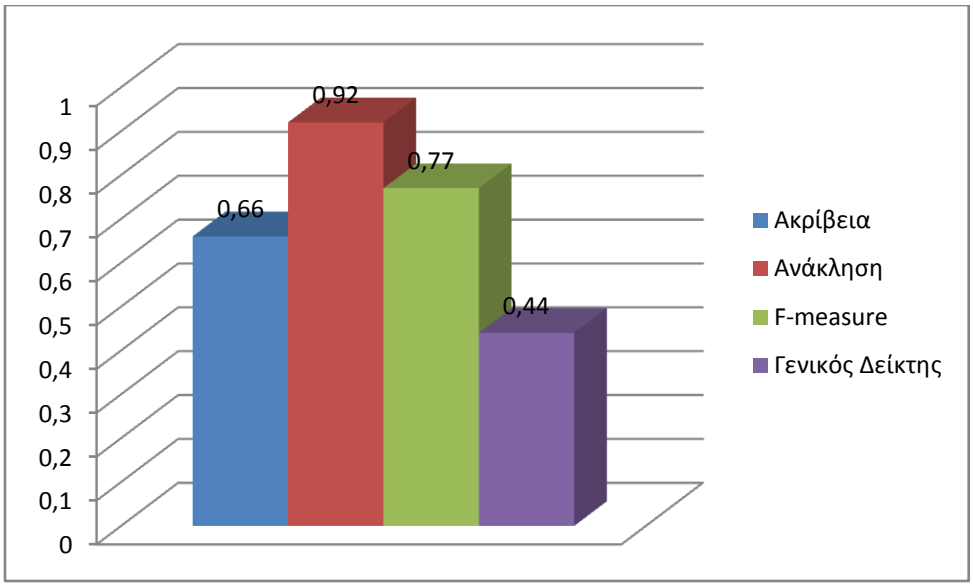
Οι αξιολογήσεις που έχουν γίνει έως τώρα σε αντίστοιχες λύσεις του προβλήματος είναι ελλιπείς, είτε όσο αφορά τη μη εξέταση σύγκρισης σχημάτων διαφορετικού τύπου, είτε εξετάζουν αποκλειστικά συγκρίσεις σχημάτων διαφορετικού τύπου. Στην αξιολόγηση που ακολουθεί, διακρίνονται και οι δύο περιπτώσεις. Έτσι,

αναλύεται αρχικά η συσχέτιση ομοειδών σχημάτων (για παράδειγμα, οντολογία με οντολογία) και στη συνέχεια περιγράφονται τα αποτελέσματα της συσχέτισης ετεροειδών σχημάτων (πχ. οντολογία με σχεσιακή βάση). Επίσης, λαμβάνονται υπόψη και άλλες παράμετροι για την αξιολόγηση του αλγορίθμου, όπως το μέγεθος των σχημάτων και ο επιστημονικός τομέας στον οποίο ανήκουν τα σχήματα.

Σύγκριση ομοειδών σχημάτων

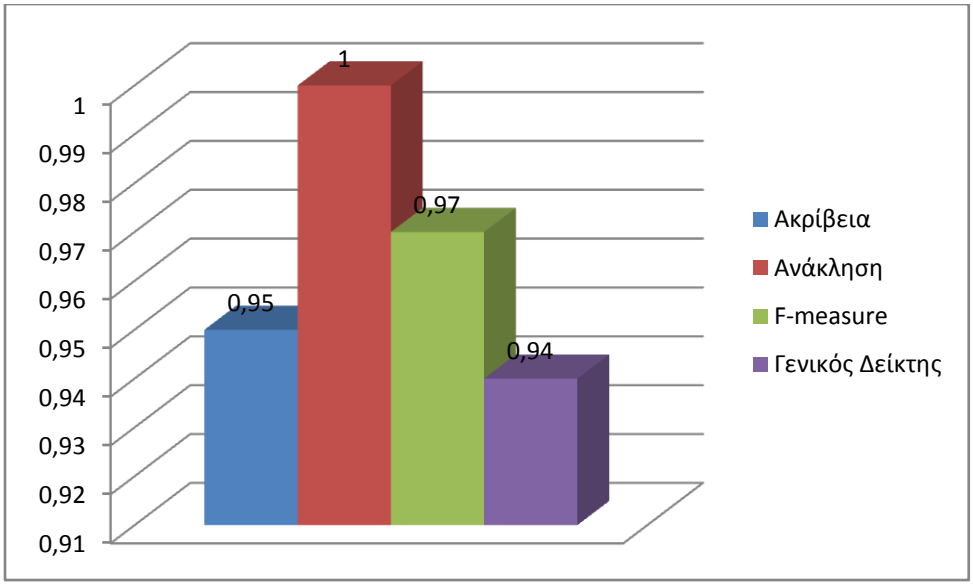
Στο τμήμα αυτό της αξιολόγησης, παρουσιάζονται τα αποτελέσματα της συσχέτισης ομοειδών σχημάτων. Τα σχήματα που χρησιμοποιήθηκαν προέρχονται από πραγματικές εφαρμογές, περιγράφουν παραγγελίες, εταιρείες και δημοσιεύσεις και καλύπτουν όλο το εύρος μεγεθών. Για κάθε ζεύγος σχημάτων που συσχετίστηκαν, περιγράφονται οι τιμές των ποιοτικών κριτηρίων (Ακρίβεια, Ανάκληση, FMeasure, Γενικός-Δείκτης) που περιγράφηκαν παραπάνω.

Πρώτα απ' όλα, παρουσιάζεται η σύγκριση των σχημάτων δύο σχεσιακών βάσεων δεδομένων. Τα σχήματα που συμμετείχαν στη σύγκριση, προέρχονται από τον τομέα της διαχείρισης παραγγελιών και αγορών. Τα αποτελέσματα συνοψίζονται στα παρακάτω: Ακρίβεια, Ανάκληση, FMeasure και Γενικός-Δείκτης στην Εικόνα 4.4. Είναι προφανές ότι εκτός των αξιόλογων αποτελεσμάτων στις παραπάνω μετρικές που παρουσιάστηκαν, ήταν αισθητή και η μείωση του χρόνου που απαιτήθηκε για τις ανάγκες της συσχέτισης, κυρίως σε σύγκριση με τη χειροκίνητη προσέγγιση.



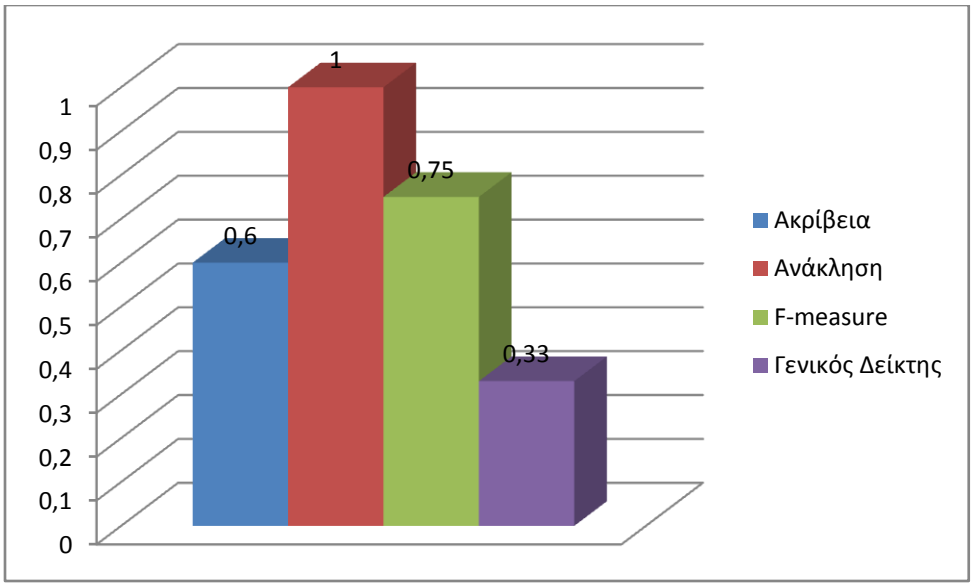
Εικόνα 4.22 Αποτελέσματα σύγκρισης δύο σχεσιακών βάσεων δεδομένων

Στη συνέχεια εξετάστηκε ο αλγόριθμος με είσοδο δύο οντολογίες που περιγράφουν την οργάνωση πανεπιστημιακών ιδρυμάτων. Τα αποτελέσματα (Εικόνα 4.5) ήταν και σε αυτήν την περίπτωση ενθαρρυντικά.



Εικόνα 4.6 Αποτελέσματα σύγκρισης δύο οντολογιών

Στην Εικόνα 4.6 φαίνονται τα αποτελέσματα από τη σύγκριση δύο XML σχημάτων.

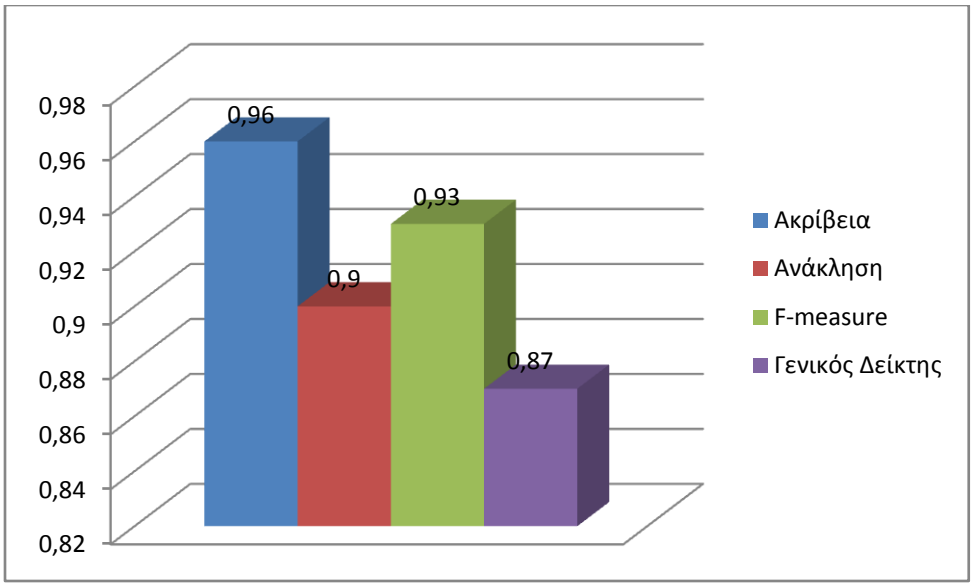


Εικόνα 4.7 Αποτελέσματα σύγκρισης δύο XML σχημάτων

Σύγκριση ετεροειδών σχημάτων

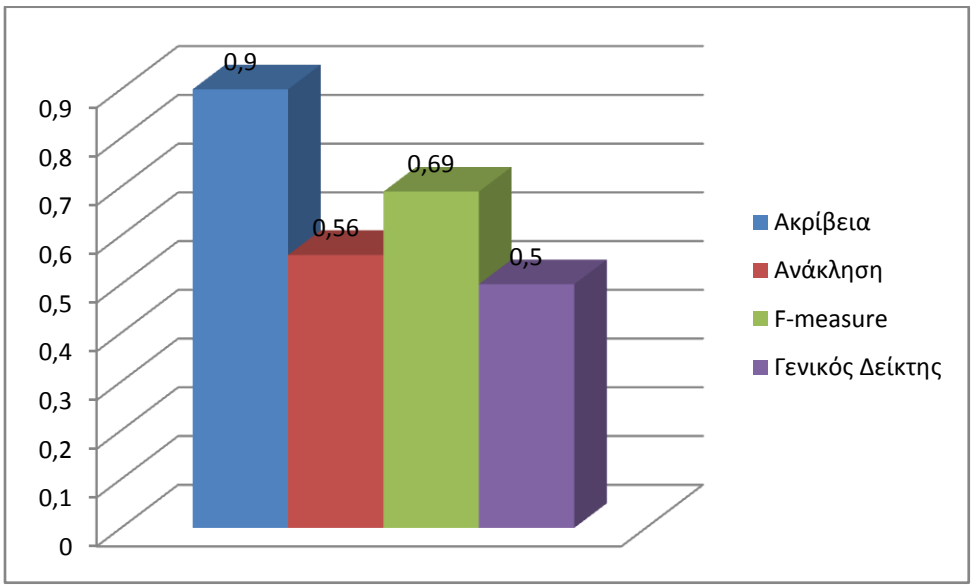
Όπως και παραπάνω, έτσι και στην αξιολόγηση ετεροειδών σχημάτων χρησιμοποιήθηκαν οι τέσσερις ποιοτικές μετρικές της Ακρίβειας, της Ανάκλησης, του FMeasure και του Γενικού-Δείκτη.

Πρώτα, εξετάσαμε την περίπτωση συσχέτισης μιας οντολογίας με το σχήμα μιας σχεσιακής βάσης και τα αποτελέσματα των ποιοτικών μετρικών, είναι άριστα και περιγράφονται στην Εικόνα 4.7. Και τα δύο σχήματα που χρησιμοποιήθηκαν περιέγραφαν παραγγελίες.



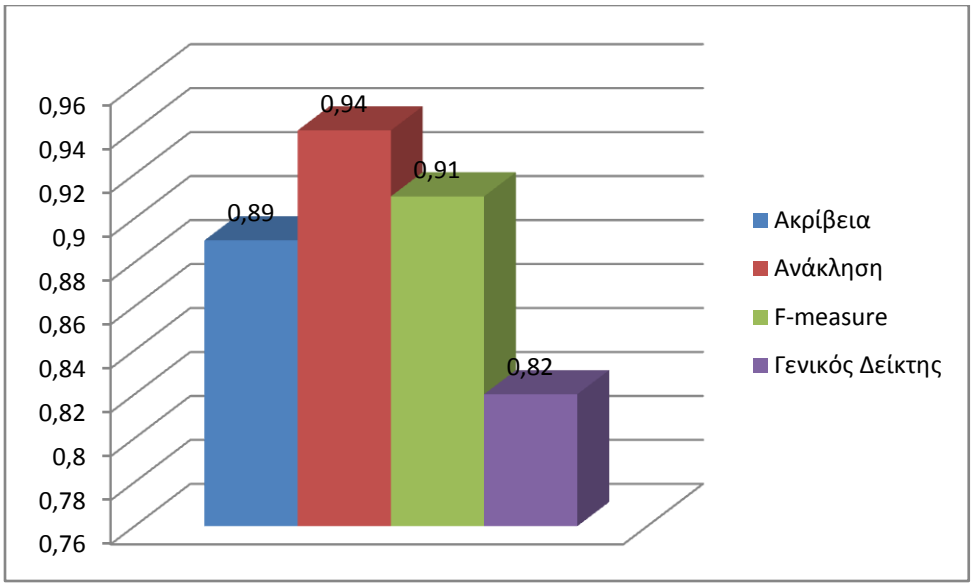
Εικόνα 4.8 Αποτελέσματα σύγκρισης οντολογίας με σχήμα σχεσιακής βάσης

Στη συνέχεια, εξετάστηκε η περίπτωση της σύγκρισης σχήματος σχεσιακής βάσης με XML σχήμα. Τα δύο σχήματα περιγράφουν τον τομέα των αγορών και πωλήσεων και τα ικανοποιητικά αποτελέσματα της μελέτης για την περίπτωση αυτή φαίνονται στην Εικόνα 4.8.



Εικόνα 4.9 Αποτελέσματα σύγκρισης XML σχήματος με σχήμα σχεσιακής βάσης

Στην Εικόνα 4.9 μπορεί να δει κάποιος τα αποτελέσματα από τη σύγκριση ενός XML σχήματος και μιας οντολογίας που περιγράφουν βιβλιογραφίες και τα συγκεκριμένα αποτελέσματα είναι αρκετά καλά.

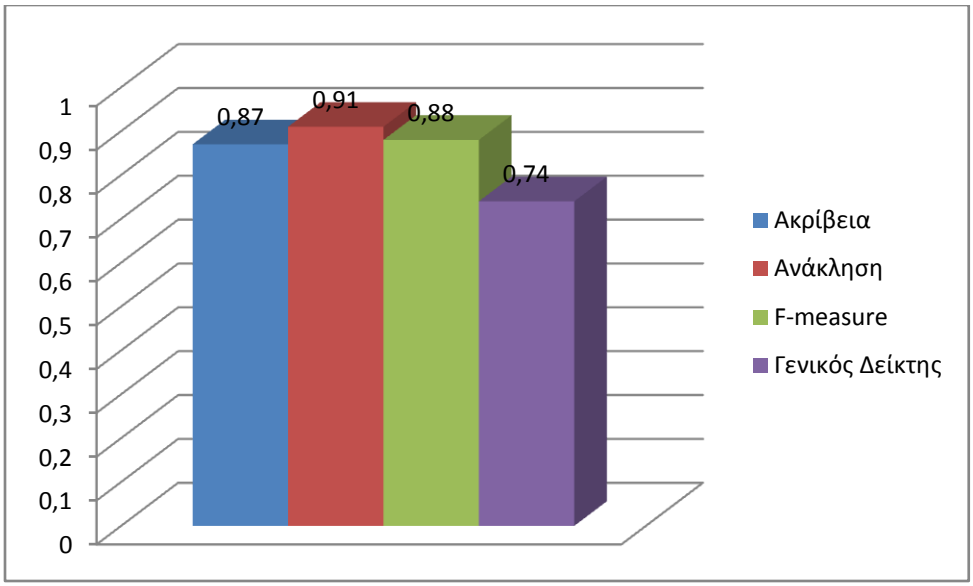


Εικόνα 4.10 Αποτελέσματα σύγκρισης XML σχήματος με οντολογία

Σύγκριση με άλλα συστήματα

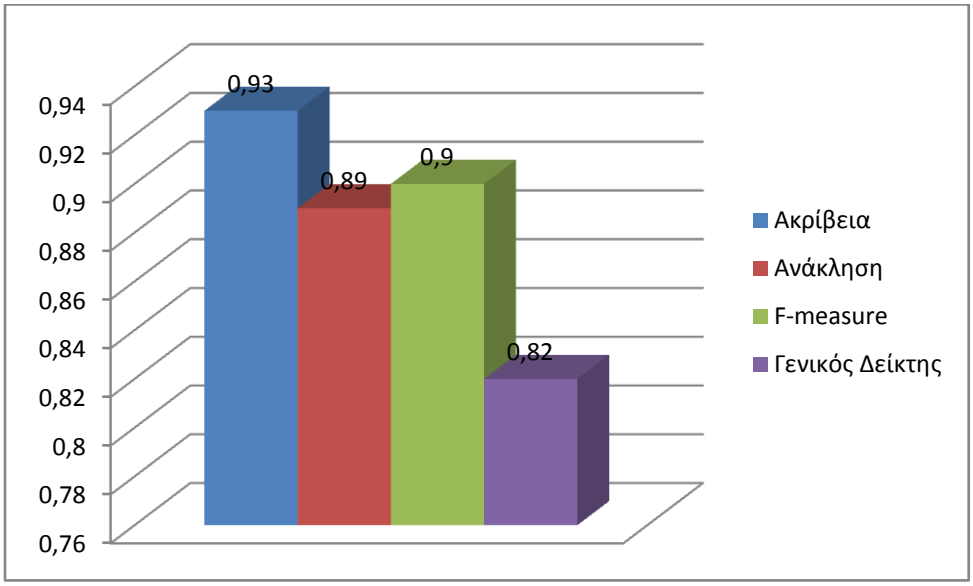
Στην ενότητα αυτή, παρουσιάζεται μια σύγκριση του προτεινόμενου αλγορίθμου με τις πιο διαδεδομένες προσεγγίσεις. Οι προσεγγίσεις αυτές είναι το COMA/COMA++, το Cupid, το SemInt και το LSD. Εκτός των ποιοτικών μετρικών, χρησιμοποιήθηκαν και τα κριτήρια της εισόδου/εξόδου και η πληθικότητα των συσχετίσεων που προκύπτουν.

Πιο αναλυτικά, αρχικά παρατίθενται οι μέσες τιμές στις μετρικές που χρησιμοποιήθηκαν και παραπάνω, για κάθε σύστημα χωριστά, στη μορφή ραβδογράμματος. Επίσης, θα πρέπει να σημειωθεί ότι χρησιμοποιήθηκε το ίδιο σύνολο σχημάτων για την αξιολόγηση των διαφορετικών συστημάτων. Έτσι, ο προτεινόμενος αλγόριθμος παρουσιάζει τα αποτελέσματα της Εικόνας 4.10.



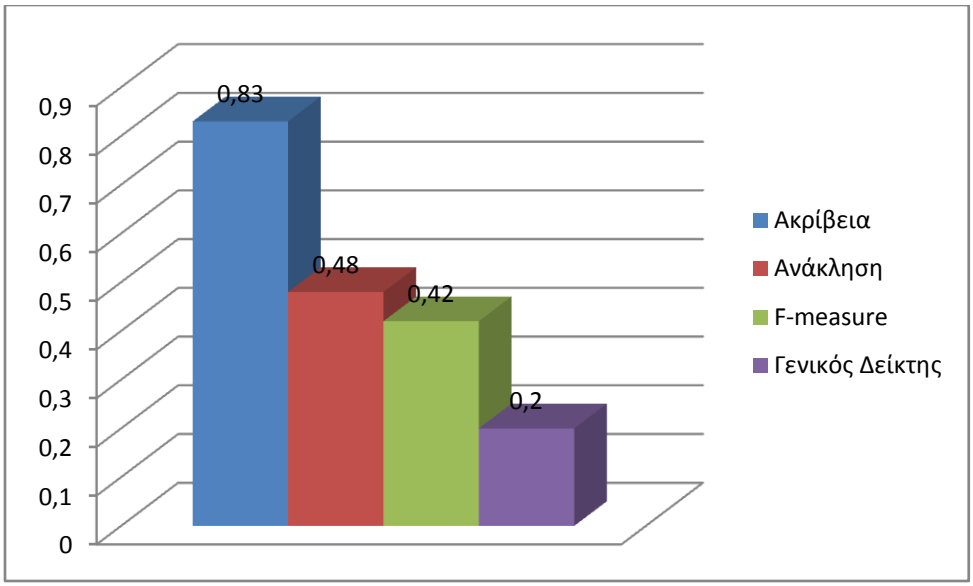
Εικόνα 4.11 Συνολικά αποτελέσματα αξιολόγησης προτεινόμενου αλγορίθμου

Τα αποτελέσματα της αξιολόγησης του COMA/COMA++ περιγράφονται στην Εικόνα 4.11.



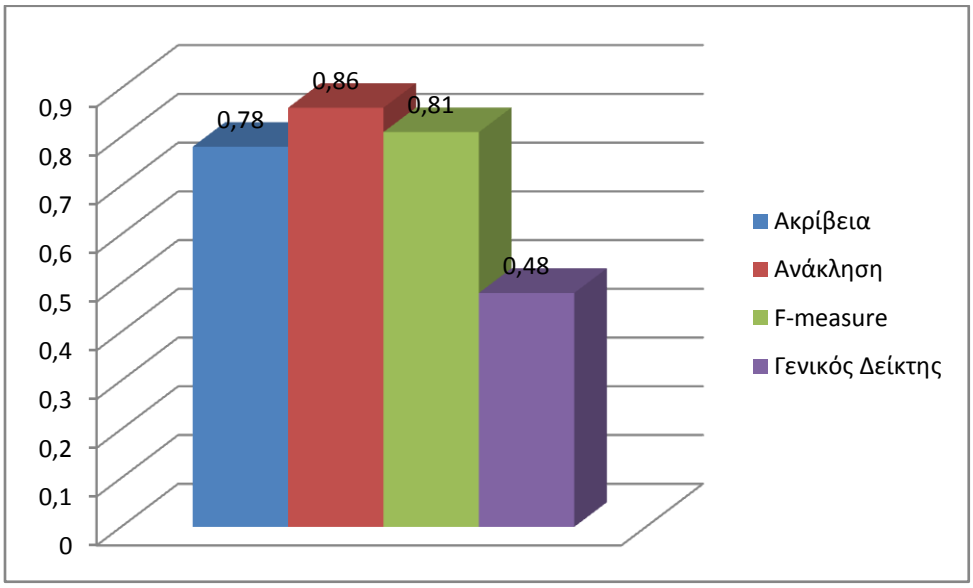
Εικόνα 4.12 Συνολικά αποτελέσματα αξιολόγησης COMA/COMA++

Τα αποτελέσματα της αξιολόγησης του Cupid περιγράφονται στην Εικόνα 4.12.



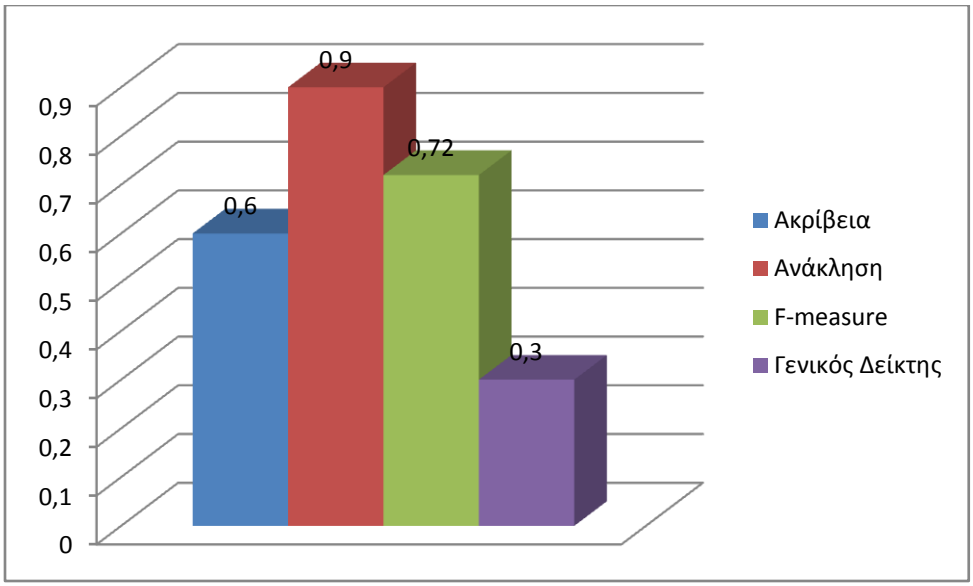
Εικόνα 4.13 Συνολικά αποτελέσματα αξιολόγησης Cupid

Τα αποτελέσματα της αξιολόγησης του SemInt περιγράφονται στην Εικόνα 4.13.



Εικόνα 4.14 Συνολικά αποτελέσματα αξιολόγησης SemInt

Τα αποτελέσματα της αξιολόγησης του LSD περιγράφονται στην Εικόνα 4.14.



Εικόνα 4.15 Συνολικά αποτελέσματα αξιολόγησης LSD

	Προτεινόμενος Αλγόριθμος	COMA/ COMA++	Cupid	SemInt	LSD
Τύπος Σχημάτων Εισόδου	Σχεσιακά, Οντολογίες, XML, Συνδυασμό αυτών	Σχεσιακά, Οντολογίες, XML, Συνδυασμό αυτών	XML	Σχεσιακά	XML
Πληθικότητα Συσχετίσεων	1:1, 1:n, n:1, n:m	1:1, n:m	1:1, n:1	n:m	1:1, n:1
Μέση Ακρίβεια	0.87	0.93	0.83	0.78	0.6
Μέση Ανάκληση	0.91	0.89	0.48	0.86	0.9
FMeasure	0.88	0.90	0.42	0.81	0.72
Μέσος Γενικός Δείκτης	0.74	0.82	~0.2	0.48	0.3

Πίνακας 4.2 Συνολικά αποτελέσματα σύγκρισης συστημάτων συσχέτισης σχημάτων

Από την παραπάνω συνοπτική περιγραφή των επιδόσεων των συστημάτων, μπορούμε να παρατηρήσουμε την πολύ καλή απόδοση του προτεινόμενου αλγορίθμου σε σχέση με το Cupid, το SemInt και το LSD. Όσο αφορά τη σύγκριση του προτεινόμενου αλγορίθμου με το COMA/COMA++, παρά τη διαφορά στα αποτελέσματα των μετρικών, ο αλγόριθμος εξακολουθεί να έχει ικανοποιητικά αποτελέσματα λόγω της γενικότητας των αποτελεσμάτων του.

4.4 Ανακεφαλαιωτικά Σχόλια

Στο τέταρτο κεφάλαιο της μεταπτυχιακής εργασίας, περιγράφηκε αναλυτικά ο σημασιολογικός αλγόριθμος συσχέτισης καθώς επίσης και η αξιολόγηση του.

Ο αλγόριθμος έχει πολλά ισχυρά χαρακτηριστικά και μερικά από αυτά είναι η ποικιλία των σχημάτων που μπορεί να χειριστεί, η επαναχρησιμοποίηση προηγούμενων αποτελεσμάτων σύγκρισης καθώς και τα άκρως ικανοποιητικά αποτελέσματα της αξιολόγησης του. Ο Πίνακας 4.4 συνοψίζει τους μέσους όρους των μετρικών της αξιολόγησης και καταδεικνύει την ποιοτική αξία του αλγορίθμου για σχήματα διαφόρων μεγεθών. Οι τιμές αυτές είναι δύσκολο να επιτευχθούν, ιδιαίτερα όταν πρόκειται για σχήματα διαφορετικού τύπου.

Επίσης, η λιτή και διαισθητική σχεδίαση του συστήματος, συντελεί στη μείωση του χρόνου που απαιτείται για την ολοκλήρωση της διαδικασίας της συσχέτισης. Συνολικά, ο στόχος της εργασίας αυτής έχει ολοκληρωθεί, γεγονός που αποδεικνύεται από τα υψηλά αποτελέσματα των μετρικών αλλά και τη γενικότερη θετική εικόνα της αξιολόγησης.

Κεφάλαιο 5. Περιγραφή Διεπαφής Συστήματος

Στα πλαίσια της μεταπτυχιακής εργασίας, εκτός από τον αλγόριθμο που αναπτύχθηκε και παρουσιάστηκε στο προηγούμενο κεφάλαιο, υλοποιήθηκε και μια διαδικτυακή διεπαφή για την αναπαράσταση του. Μέσω της διεπαφής, ο χρήστης δύναται να παραμετροποιήσει και να καλέσει τον αλγόριθμο καθώς επίσης και να πλοηγηθεί στα προς συσχέτιση σχήματα. Πρόκειται για ένα αυτόνομο σύστημα που επιτρέπει στο χρήστη την απομακρυσμένη πρόσβαση και δεν απαιτείται η προεγκατάσταση λογισμικού στον υπολογιστή εργασίας. Στο κεφάλαιο αυτό αναλύεται περαιτέρω η διεπαφή του συστήματος, ο τρόπος σύνδεσης της με το σημασιολογικό αλγόριθμο συσχέτισης και στο τέλος του κεφαλαίου περιγράφεται ένα ενδεικτικό σενάριο χρήσης.

5.1 Περιγραφή Διεπαφής Συστήματος

Η διεπαφή του συστήματος επικεντρώνεται σε μια διαδικτυακή πλατφόρμα, άμεσα προσβάσιμη στον κάθε χρήστη. Στη συνέχεια περιγράφεται αναλυτικά η διάταξη της διαδικτυακής διεπαφής καθώς και ο τρόπος με τον οποίο συμμετέχει στη διαδικασία το λεξικό WordNet.

5.1.1 Διεπαφή Χρήσης

Η διεπαφή του συστήματος είναι αρκετά λιτή και περιεκτική έτσι ώστε ακόμα και ένας αρχάριος χρήστης να μπορεί να παραμετροποιεί και να πυροδοτεί εύκολα τον αλγόριθμο. Εκτός της διαισθητικότητας του συστήματος δόθηκε έμφαση, κατά τη σχεδίαση του συστήματος, να μην επιβαρυνθεί από άσκοπες και περιττές πληροφορίες ή δυσνόητα σχέδια. Επίσης, η διεπαφή ακολουθεί όλους τους κανόνες ευχρηστίας και αποτελείται από δύο βασικά τμήματα.

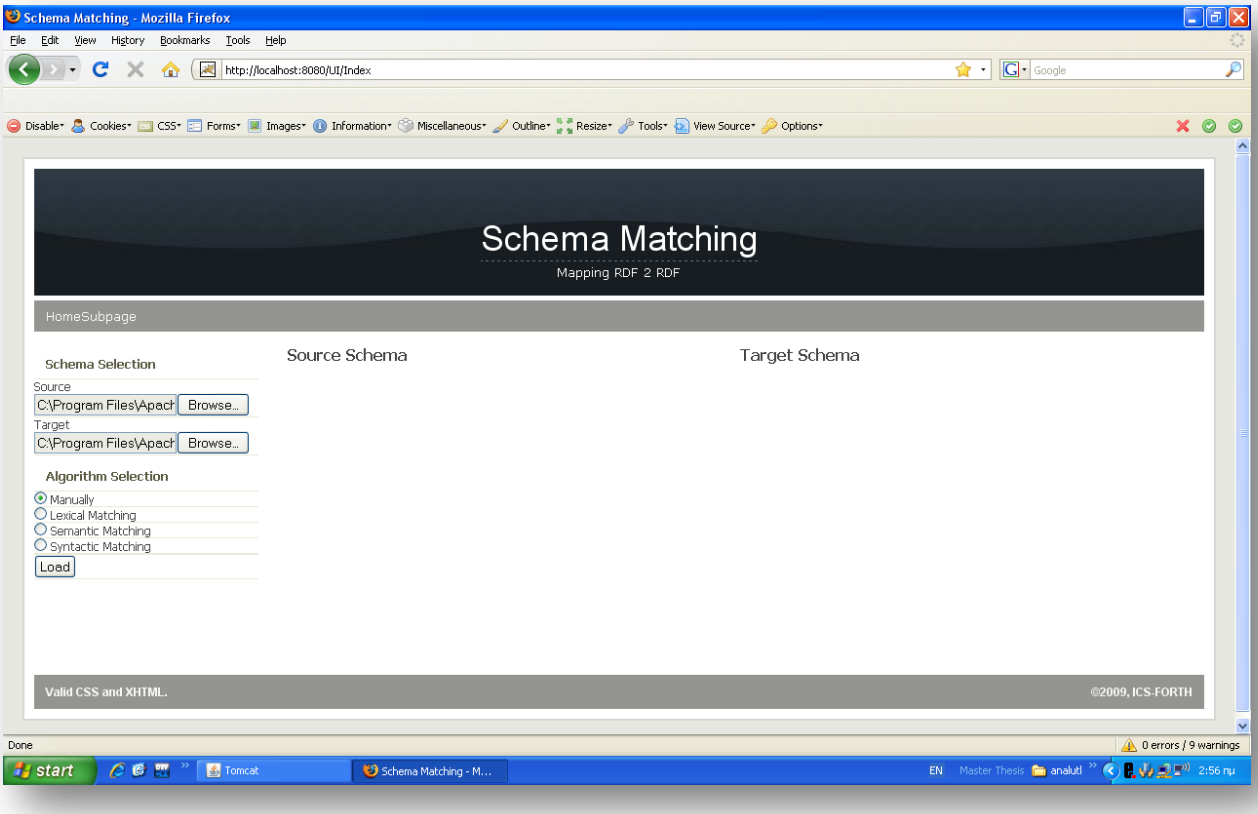
Το πρώτο τμήμα, που βρίσκεται και στην αριστερή πλευρά της ιστοσελίδας, και φαίνεται στην Εικόνα 5.1 είναι αυτό που παρέχει στο χρήστη τη δυνατότητα να επιλέξει τα προς συσχέτιση σχήματα καθώς και τον τρόπο συσχέτισης τους. Έτσι, τα δύο πτυσσόμενα μενού επιτρέπουν στο χρήστη να καθορίσει το σχήμα-πηγή και το σχήμα-στόχος, που θα αποθηκευτούν στη βάση δεδομένων που διατηρεί στο σύστημα, αν δεν έχουν καταχωρηθεί ήδη.

Στη συνέχεια, ο χρήστης μπορεί να επιλέξει την προεπιλεγμένη επιλογή με σκοπό να εκτελέσει χειροκίνητη συσχέτιση. Οπότε σε αυτήν την περίπτωση, απλά γίνεται έλεγχος αν υπάρχουν τα σχήματα στη βάση, αν όχι τότε καταχωρούνται και στη συνέχεια, φορτώνονται τα σχήματα, στο δεύτερο τμήμα του συστήματος. Εκτός της προεπιλεγμένης επιλογής, ο χρήστης μπορεί να επιλέξει μια από τις τρεις παραλλαγές του αλγορίθμου, λεξικογραφική, σημασιολογική και συντακτική. Εδώ αξίζει να σημειωθεί ότι η συντακτική αποτελεί την πλήρη έκδοση της προσέγγισης που προτείνεται από αυτή τη μεταπτυχιακή εργασία. Μετά την επιλογή του αλγορίθμου, ο χρήστης δύναται να καθορίσει ποια από τα στοιχεία του σχήματος πηγής θα συσχετιστούν με τα στοιχεία του σχήματος στόχου, προσδιορίζοντας το όριο που καθορίζει την ομοιότητα. Το χαμηλότερο επιτρεπτό όριο ομοιότητας είναι το 0, ενώ η μέγιστη τιμή ομοιότητας είναι το 1. Έτσι, στην περίπτωση που ο χρήστης δεν αλλάξει τις τιμές αυτές ή δώσει τιμές εκτός ορίων τότε τα πεδία αυτά παίρνουν τις προεπιλεγμένες τιμές. Αυτό σημαίνει ότι αν υπάρχει οποιαδήποτε τιμή ομοιότητας στη βάση, τότε ο αλγόριθμος δεν προχωράει σε επανεξέταση της ομοιότητας των δύο όρων και εκτελείται μόνο για τα στοιχεία της πηγής για τα οποία δεν έχει βρεθεί καμιά συσχέτιση με κανένα στοιχείο του σχήματος στόχου, από προηγούμενη συσχέτιση σχημάτων.

Αφού συμπληρωθούν τα όρια που περιγράφηκαν παραπάνω, ο χρήστης καλείται να καθορίσει ποιες από τις συσχετίσεις που θα επιστρέψει ο αλγόριθμος, επιθυμεί να καραχωρηθούν στη βάση. Εδώ πρακτικά, εκτελείται το φιλτράρισμα των αποτελεσμάτων με βάση τα παρακάτω φίλτρα. Ο χρήστης επιλέγει μέσω ενός πτυσσόμενου μενού, μια από τις τέσσερις στρατηγικές επιλογής συσχετίσεων, που είναι: Μέγιστη συσχέτιση, Συσχετίσεις πάνω από όριο ομοιότητας, Απόσταση από μέγιστη ομοιότητα και Όλες οι συσχετίσεις. Πιο αναλυτικά, επιλέγοντας ο χρήστης τη «Μέγιστη Συσχέτιση» το σύστημα κρατάει για κάθε στοιχείο του σχήματος πηγής τη μέγιστη τιμή ομοιότητας. Δηλαδή αν για ένα στοιχείο, του σχήματος πηγής, επιστραφούν συσχετίσεις με πολλά στοιχεία του σχήματος στόχου κρατιέται στη βάση η μέγιστη από αυτές. Εναλλακτικά, ο χρήστης μπορεί να ορίσει ως στρατηγική επιλογής συσχετίσεων τη «Συσχετίσεις πάνω από όριο ομοιότητας». Παραμετροποιώντας έτσι τον αλγόριθμο, ο χρήστης επιτυγχάνει να αποθηκεύσει στη βάση του συστήματος τις συσχετίσεις που ξεπερνούν το όριο που έδωσε. Η τρίτη εναλλακτική που έχει στη διάθεση του ο χρήστης της διεπαφής, είναι η «Απόσταση από μέγιστη ομοιότητα». Όταν επιλεχθεί αυτή η στρατηγική ο

αλγόριθμος επιστρέφει τις συσχετίσεις μεταξύ των σχημάτων εισόδου των οποίων η ομοιότητα έχει τιμή πάνω από τη διαφορά της τιμής που έδωσε ο χρήστης μέσω της διεπαφής από τη μέγιστη ομοιότητα. Τέλος, υπάρχει και η στρατηγική επιλογής όλων των συσχετίσεων που βρήκε ο αλγόριθμος. Στην περίπτωση αυτή αποθηκεύονται στη βάση όλες οι συσχετίσεις που βρέθηκαν. Αφού τελειώσει κάποιος με την επιλογή στρατηγικής (αν δεν επιλεχθεί στρατηγική εκτελείται η στρατηγική μέγιστης συσχέτισης) είναι έτοιμος να πυροδοτήσει τον αλγόριθμο.

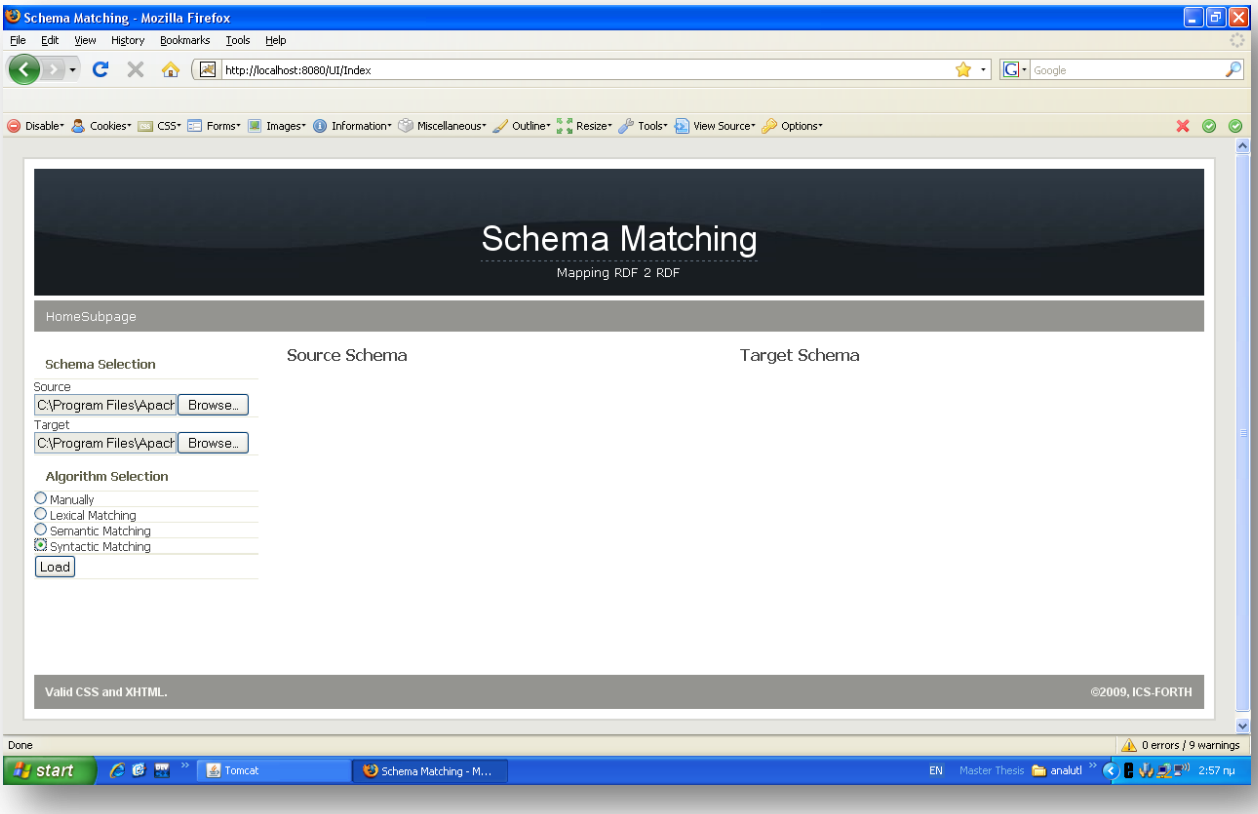
Όσο αφορά το δεύτερο τμήμα της διεπαφής, αυτό περιλαμβάνει τη γραφική αναπαράσταση των σχημάτων. Στο τμήμα αυτό, που καταλαμβάνει και το μεγαλύτερο χώρο στη διάταξη της εφαρμογής, αναπαριστώνται γραφικά τα δύο προς συσχέτιση σχήματα. Ένα αντιπροσωπευτικό παράδειγμα αποτελεί η Εικόνα 5.1. Η ευελιξία της γραφικής αναπαράστασης των σχημάτων, έγκειται στην ιδιότητα του συστήματος να συρρικνώνει και να επεκτείνει τα σχήματα. Ο χρήστης μπορεί να πλοηγηθεί στα δύο σχήματα, να επιλέξει όρους από τα δύο σχήματα πηγής και στόχου και να καταχωρήσει το βαθμό ομοιότητας τους στη βάση του συστήματος, με την αρωγή της φόρμας που υπάρχει στο τέλος της γραφικής αναπαράστασης. Σε αυτό το σημείο αξίζει να σημειωθεί ότι τα στοιχεία της φόρμας μπορούν να συμπληρωθούν με την απλή επιλογή του εκάστοτε όρου κατά την πλοήγηση του χρήστη στα σχήματα. Το χαρακτηριστικό αυτό γνώρισμα ενθαρρύνει τη χρήση του συστήματος, αφού πρόκειται για επιπλέον ευκολία της χειροκίνητης συσχέτισης.



Εικόνα 5.1 Διαδικτυακή διεπαφή συστήματος

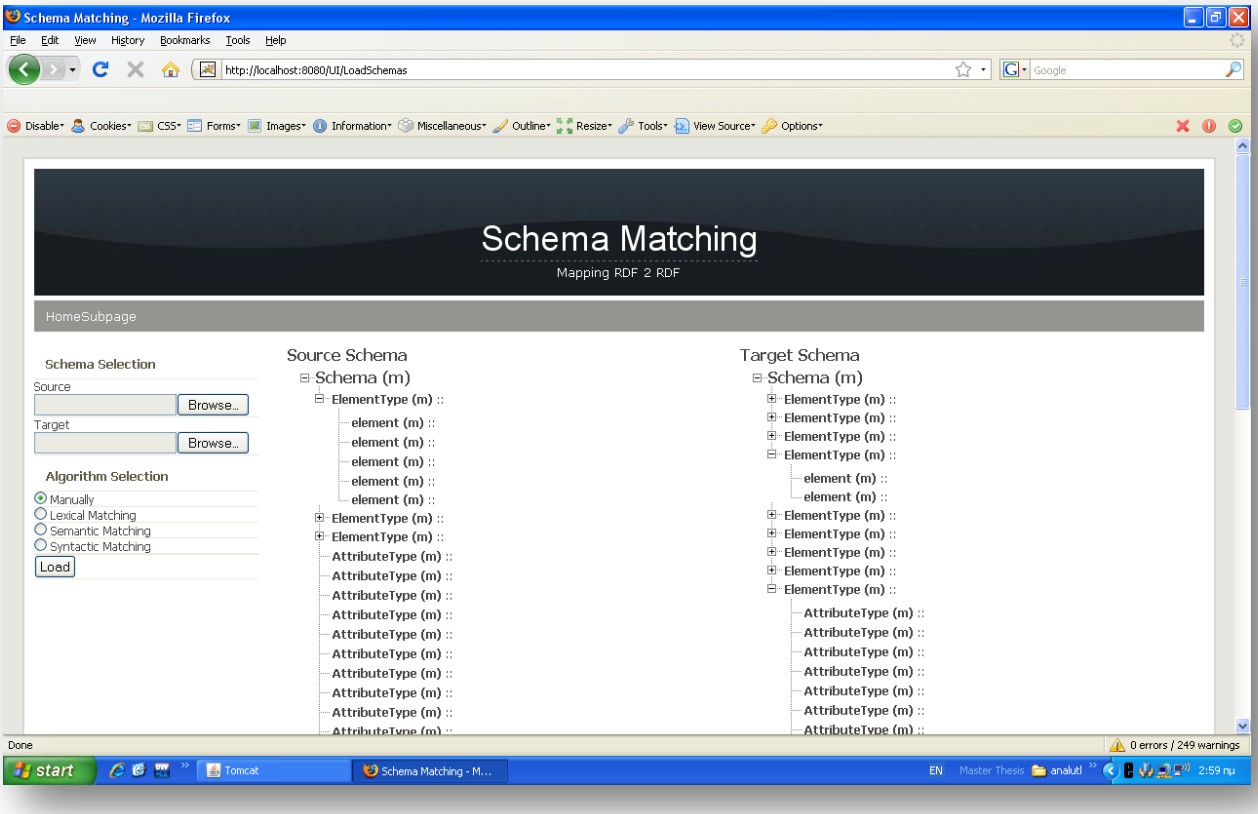
5.2 Σενάριο Χρήσης

Στο τμήμα αυτό του κεφαλαίου, περιγράφεται ένα αντιπροσωπευτικό σενάριο χρήσης του συστήματος. Υποθέτουμε ότι, για τις ανάγκες του σεναρίου, θα συσχετιστούν δύο XML σχήματα από τον τομέα των αγοραπωλησιών. Ο χρήστης, επιλέγει τα δύο προς συσχέτιση σχήματα, όπως φαίνεται στην Εικόνα 5.2. Θεωρούμε ότι τα σχήματα δεν έχουν καταχωρηθεί στη βάση δεδομένων του συστήματος, έτσι μετατρέπονται σε άκυκλους γράφους και στη συνέχεια αποθηκεύονται στη βάση. Αφού καθορίσει ο χρήστης τα σχήματα που θα συμμετάσχουν στη διαδικασία, έχει δύο επιλογές. Είτε να παραμετροποιήσει και να πυροδοτήσει το σημασιολογικό αλγόριθμο, είτε να εκτελέσει χειροκίνητη συσχέτιση. Στα πλαίσια του σεναρίου, ο χρήστης θα καλέσει, αρχικά, το σημασιολογικό αλγόριθμο και στη συνέχεια θα προχωρήσει σε χειροκίνητη συσχέτιση. Στο σημείο, λοιπόν, αυτό ο χρήστης παραμετροποιεί τον αλγόριθμο, όπως φαίνεται στην Εικόνα 5.2 και τον πυροδοτεί.



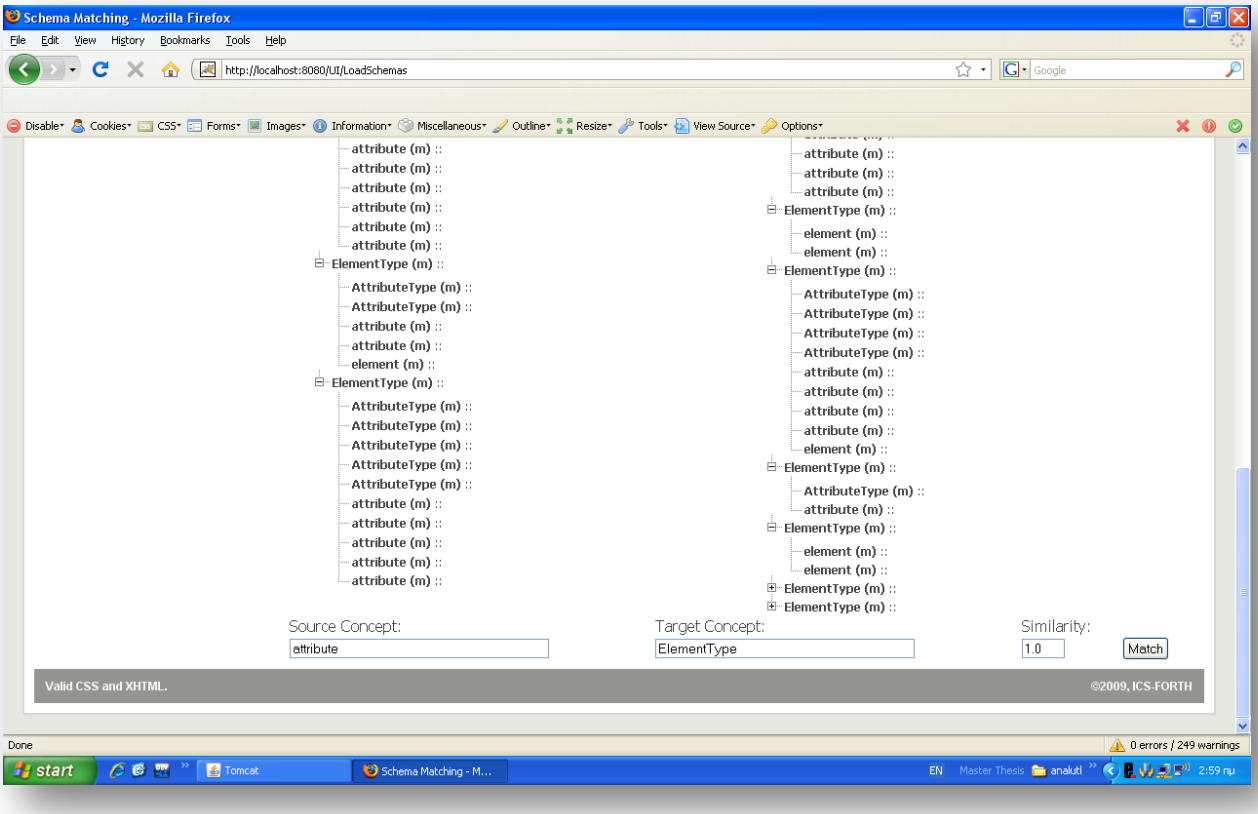
Εικόνα 5.2 Επιλογή σχημάτων, παραμετροποίηση και πυροδότηση αλγορίθμου

Αφού ολοκληρωθεί η εκτέλεση του αλγορίθμου, φορτώνεται η γραφική αναπαράσταση των σχημάτων, Εικόνα 5.3.



Εικόνα 5.3 Γραφική αναπαράσταση των σχημάτων

Τέλος, ο χρήστης μπορεί να επιλέξει και να συσχετίσει όρους από τα δύο σχήματα και η συσχέτιση αυτή να καταχωρηθεί στο σύστημα, Εικόνα 5.4.



Εικόνα 5.4 Χειροκίνητη συσχέτιση σχημάτων

5.3 Χρήση Λεξικού WORDNET

Στην ενότητα αυτή θα αναλυθεί το λεξικό που χρησιμοποιήθηκε για τη σύγκριση των σχημάτων και για τον καθορισμό των συσχετίσεων μεταξύ τους. Το σύστημα χρησιμοποιεί το αγγλικό λεξικό όρων WordNet, το οποίο είναι ευρέως διαδεδομένο τόσο στον τομέα της σύγκρισης σχημάτων, όσο και στον τομέα της ανάκτησης πληροφορίας.

Στο σύστημα χρησιμοποιείται μια πλατφόρμα διαχείρισης του συγκεκριμένου λεξικού υλοποιημένη στην java (jwnl). Μέσω της πλατφόρμας αυτής μπορεί κάποιος να προσθέσει νέους όρους στο λεξικό και να δημιουργήσει το δικό του αλγόριθμο αναζήτησης μέσα στο λεξικό. Πριν αναλύσουμε τις παρεμβάσεις που έγιναν στην συγκεκριμένη πλατφόρμα για να επιτύχουμε το ποθητό αποτελέσματα, θα αναλυθεί σε γενικές γραμμές η πολιτική λειτουργίας του συγκεκριμένου λεξικού.

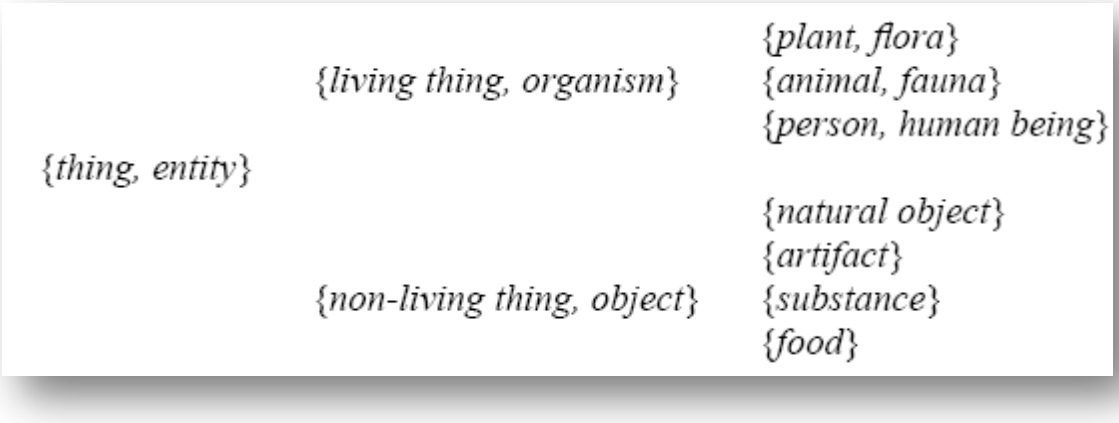
Για να είναι δυνατή η χρήση του λεξικού πρέπει κάποιος να το κατεβάσει από το δίκτυο και να το εγκαταστήσει στο μηχάνημά του (Αναλυτικές οδηγίες υπάρχουν στο παράρτημα της αναφοράς). Τόσο το λεξικό όσο και η πλατφόρμα διαχείρισης βρίσκονται σε κατάλληλο φάκελο στο επισυναπτόμενο οπτικό δίσκο. Το WordNet αποτελείται από αρχεία με όρους και είναι προγραμματισμένο με τέτοιο τρόπο, ώστε να μπορεί να μετατρέψει τα αρχεία αυτά σε μια βάση δεδομένων. Επίσης περιλαμβάνει και έτοιμες ρουτίνες αναζήτησης και κατάλληλη διεπαφή χρήσης, την οποία μπορεί να χρησιμοποιήσει κάποιος για να δει πληροφορίες από τη βάση δεδομένων του λεξικού. Τα αρχεία όρων οργανώνουν ουσιαστικά, ρήματα, επιρρήματα, αντωνυμίες και επίθετα σε ομάδες συνωνύμων. Κατάλληλος κώδικας μετατρέπει τα αρχεία αυτά σε μια βάση δεδομένων, η οποία κρατάει κωδικοποιημένες τις σχέσεις μεταξύ των ομάδων συνωνύμων. Οι σχέσεις μεταξύ των ομάδων γίνονται ορατές με κατάλληλες ρουτίνες που προσφέρονται έτοιμες από το λεξικό.

Οι πληροφορίες στο WordNet είναι οργανωμένες σε λογικές ομάδες, οι οποίες λέγονται «synsets». Κάθε «synset» αποτελείται από μία λίστα από συνώνυμες λέξεις ή συγκριτικά όμοιες, καθώς επίσης και από δείκτες οι οποίοι περιγράφουν τις σχέσεις μεταξύ του συγκεκριμένου «synset» και των άλλων «synsets». Μια λέξη μπορεί να εμφανιστεί σε περισσότερα από ένα «synset», και σε περισσότερα από ένα μέρη του λόγου ανάλογα με το νόημα της (sense). Οι λέξεις στο «synset» είναι ομαδοποιημένες με τέτοιο τρόπο ώστε να είναι ανταλλάξιμες με κάποια συναφή έκφραση.

Δύο ειδών σχέσεις εκπροσωπούν οι δείκτες: λεκτικές και σημασιολογικές. Λεκτικές σχέσεις κρατιούνται μεταξύ σημασιολογικά σχετικών μορφών λέξεων. Σημασιολογικές σχέσεις κρατιούνται μεταξύ λεκτικά όμοιων σημασιών. Αυτές οι σχέσεις περιλαμβάνουν (αλλά δεν περιορίζονται σε αυτά) υπερώνυμα-hypernymy (η λέξη είναι είδος – ..is kind of), υπώνυμα-hyponymy (είναι είδος λέξης – is kind of..), αντώνυμα, μερώνυμα-meronyms (μέρος της λέξης – parts of ...), ολώνυμα-olonyms (λέξη μέρος από – ..is part of), συνεπαγωγές, λογικά υποδεέστερες και λογικά σημαντικότερες (Πίνακας 5.1). Τα ουσιαστικά και τα ρήματα οργανώνονται σε ιεραρχίες βασισμένες στις σχέσεις υπερωνύμων και υπωνύμων μεταξύ των «synsets». Επιπρόσθετοι δείκτες χρησιμοποιούνται για να υποδείξουν άλλες σχέσεις μεταξύ των «synsets».

Σημασιολογική Σχέση	Συντακτική Κατηγορία	Παραδείγματα
Συνώνυμα	N, V, Aj, Av	Pipe, tube Rise, ascend Sad, unhappy Rapidly, speedily
Αντώνυμα	Aj, Av, (N, V)	Wet, dry Powerful, powerless Friendly, unfriendly Rapidly, slowly
Υπώνυμα (ειδικότερα)	N, V	Sugar maple, maple Maple, tree Tree, plant
Μερώνυμα (μέρος λέξης)	N	Brim, hat Gin, martini Ship, fleet
Τρόπος	V	March, walk Whisper, speak
Συνεπαγωγή	V	Drive, ride Divorce, marry
Σημείωση: N=**Ουσιαστικά** Aj=**Επίθετα** V=**Ρήματα** Av=**Επιρρήματα**		

Πίνακας 5.1 Σημασιολογικές σχέσεις λεξικού WordNet



Εικόνα 5.5 Παράδειγμα ιεραρχικής δομής ουσιαστικών και ρημάτων στο WordNet

Η Εικόνα 5.5 αποτελεί ένα παράδειγμα, όπου δέκα σύνολα συνωνύμων (synsets) ουσιαστικών δομούνται ιεραρχικά με κόμβο ρίζα το σύνολο {thing,entity}, το οποίο έχει παιδιά το σύνολο{living thing, organism} και το σύνολο {non-living thing, object}.

Αυτά αντίστοιχα τα σύνολα έχουν παιδιά τα σύνολα που φαίνονται στην Εικόνα 5.5 και παίζουν το ρόλο των φύλλων της δομής.

Τα επίθετα οργανώνονται σε ομάδες, οι οποίες περιέχουν ένα επικεφαλή «synset» και διάφορα περιφερειακά «synsets». Κάθε τέτοια ομάδα οργανώνεται γύρω από ζευγάρια αντιθέτων (και περιστασιακά από τριάδες αντιθέτων). Τα ζευγάρια αντιθέτων (ή οι τριάδες) έχουν οριστεί στο επικεφαλής «synset» της ομάδας. Τα περισσότερα επικεφαλής «synsets» έχουν ένα ή περισσότερα περιφερειακά «synsets», κάθε ένα από τα οποία αναπαριστά μια σκέψη, η οποία είναι παρόμοια σε σημασία της σκέψης που αναπαριστά το επικεφαλής «synset». Ένας τρόπος να φανταστεί κάποιος την οργάνωση των ομάδων των επιθέτων είναι να φανταστεί έναν τροχό, με το επικεφαλής «synset», το κέντρο του και τα περιφερειακά «synsets» σαν τις ακτίνες του. Δύο ή περισσότεροι τροχοί είναι λογικά συνδεδεμένοι μέσω ενός αντιθέτου, το οποίο μπορεί να νοηθεί σαν ο άξονας ανάμεσα στους τροχούς π.χ. στο παράδειγμα που ακολουθεί έχουμε δύο ομάδες επιθέτων αποτελούμενες από δύο «synsets» η κάθε μία. Τα ζευγάρια αντιθέτων στα γύρω από τα οποία οργανώνονται αυτές οι ομάδες είναι το HOT και το COLD. Οι δύο αυτές ομάδες συνδέονται μέσω του COLD.

{HOT, COLD, (hot to the touch)}
{warm} –
{COLD, HOT, frigid, (cold to the touch)}
{freezing}

Τέλος, τα επιρρήματα συχνά προκύπτουν από τα επίθετα, και μερικές φορές έχουν αντίθετα. Επομένως, το «synset» για ένα επίρρημα συνήθως περιέχει ένα λεκτικό δείκτη στο επίθετο από το οποίο προήλθε. Η οργάνωση αυτή φαίνεται και αναλυτικότερα στο επόμενο παράδειγμα: {badly, δείκτης σε επίθετο: bad, well, ill, (“He was badly prepared”).

Το σύστημά μας, όπως προαναφέρθηκε, για να χειριστεί το συγκεκριμένο λεξικό δεν χρησιμοποιεί τη διεπαφή που προσφέρεται από αυτό, αλλά μια πλατφόρμα διαχείρισης αυτού υλοποιημένη στην java (jwnl-open source). Η πλατφόρμα αυτή, μαζί με τις τροποποιήσεις που υπέστη, βρίσκεται στον επισυναπτόμενο οπτικό δίσκο. Για να μπορέσει ο αλγόριθμος να εντοπίσει σημασιολογικές ομοιότητες με τη χρήση του λεξικού, οι λέξεις οργανώθηκαν σε ιεραρχίες υπερωνύμων. Για τη σύγκριση δύο στοιχείων και για τον εντοπισμό πιθανής συσχέτισης γίνεται κάθε

φορά η ακόλουθη διαδικασία. Πρώτα, τα ονόματα των στοιχείων του σχήματος πηγής και του σχήματος στόχου χωρίζονται σε τμήματα (tokens). Για κάθε τμήμα του στοιχείου του σχήματος πηγή βρίσκει το μέγιστο βαθμό ομοιότητας με κάθε τμήμα του στοιχείου του σχήματος στόχου. Η τιμή του βαθμού καθορίζεται με βάση το βάθος που βρίσκονται τα τμήματα του στοιχείου του σχήματος πηγής και στόχου στην ιεραρχία του λεξικού. Αν, παραδείγματος χάριν, προσπαθεί να συσχετίσει τα στοιχεία «animal» και «person» με βάση την ιεραρχική δομή της Εικόνας 5.5, οι συναρτήσεις υπολογισμού του βαθμού ομοιότητας (Εικόνα 5.6) δίνουν: **depth**=1 **depth1**=2 **depth2**=2

sim= 2*1 / 2+2 => sim= 2/4 => **sim=0,5**
distance= ((2-1) / 2 + (2-1) / 2) / 2 => distance= 1/2 και sim=1 – 0,250 => **sim= 0,75**
distance= ((2-1) / 2 + (2-1) / 2) / 2 => distance= 1/2 και sim=1 – 0,5 => **sim= 0,5**

Έτσι, σχηματίζεται ένας πίνακας με τις τιμές ομοιότητας των τμημάτων του στοιχείου από το σχήμα πηγή με τα τμήματα από το στοιχείο του σχήματος στόχου. Αυτό γίνεται για κάθε στοιχείο του σχήματος πηγής που δεν ικανοποιεί τα όρια ομοιότητας που δίνει ο χρήστης μέσω της διεπαφής χρήσης. Το βασικό κομμάτι κώδικα βρίσκεται στο παράρτημα της εργασίας αυτής.

//πρώτος τρόπος επιλογής βασισμένος στον αριθμό των ISA συνδέσεων μέχρι το σημείο εντοπισμού.
//όπου **depth** το βάθος κοινού γονιού, **depth1** το βάθος του πρώτου λήμματος και **depth2** το βάθος του //δεύτερου λήμματος
sim = (float)2*(depth) / (depth1 + depth2);
break;
case 2:
//αισιόδοξη επιλογή βαθμού ομοιότητας βασισμένης στην ιεραρχική οργάνωση
distance = ((float)(depth1-depth) / depth1 + (float)(depth2-depth) /depth2) /2;
sim = 1 - (float)java.lang.Math.pow(distance, 2);
break;
case 3:
// απαισιόδοξη επιλογή βαθμού ομοιότητας βασισμένης στην ιεραρχική οργάνωση
distance = ((float)(depth1-depth) / depth1 + (float)(depth2-depth) /depth2) /2;
sim = 1 - distance;
Εικόνα 5.6 Υπολογισμός βαθμού Ομοιότητας με χρήση Λεξικού WordNet

Το συγκεκριμένο τμήμα κώδικα αυξάνει αρκετά την πολυπλοκότητα και το χρόνο εκτέλεσης του αλγορίθμου. Αλλά ο χρόνος είναι κάτι που έρχεται σε δεύτερη μοίρα για χάρη των σημασιολογικών αποτελεσμάτων που θέλουμε να επιτύχουμε.

5.4 Ανακεφαλαιωτικά Σχόλια

Στην ενότητα αυτή, έγινε μια αναλυτική παρουσίαση του συστήματος που υλοποιήθηκε. Πρώτα αναλύθηκε η διεπαφή χρήσης, μέσω της οποίας ο χρήστης δύναται να καλέσει το σημασιολογικό αλγόριθμο ή να προτείνει χειροκίνητα συσχετίσεις. Στη συνέχεια, περιγράφηκε το ενδεδειγμένο και αντιπροσωπευτικό σενάριο χρήσης του συστήματος για την εύρεση στο μικρότερο χρόνο των καλύτερων αποτελεσμάτων (ποιοτικές και σωστές συσχετίσεις). Τέλος, έγινε μια σύντομη περιγραφή του λεξικού WordNet καθώς επίσης και της χρήσης του.

Κεφάλαιο 6. Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Πλεονεκτήματα – Μειονεκτήματα

Ο σημασιολογικός αλγόριθμος αλλά και το σύστημα που τον υλοποιεί και έχουν περιγραφεί αναλυτικά σε προηγούμενα κεφάλαια, αποτελούν μια ολοκληρωμένη λύση στο πρόβλημα της συσχέτισης σχημάτων. Πρόκειται για ολοκληρωμένη πρόταση αφού καλείται να συσχετίσει σχήματα μεγαλύτερα ή μικρότερα, διαφορετικού τύπου και τα αποτελέσματα της μεθόδου αυτής είναι σημαντικά. Από την άλλη πλευρά, όμως, χαρακτηρίζεται και από σημεία που χρήζουν αντιμετώπισης ώστε να βελτιωθεί τόσο η αποδοτικότητα του αλγορίθμου, όσο και η διεπαφή του συστήματος. Στη συνέχεια του κεφαλαίου αυτού περιγράφονται τόσο τα δυνατά όσο και τα τρωτά σημεία του αλγορίθμου και του συστήματος υλοποίησης του.

6.1.1 Σημασιολογικός Αλγόριθμος

Το σημαντικότερο όλων των πλεονεκτημάτων του αλγορίθμου είναι η ακρίβεια των αποτελεσμάτων του. Η ακρίβεια και η ορθότητα όμως λειτουργούν εις βάρος της πολυπλοκότητας του. Αυτό σημαίνει ότι ο χρόνος που απαιτείται για την εκτέλεση του σημασιολογικού αλγορίθμου αυξάνεται ελάχιστα σε σχέση με ανταγωνιστικούς του αλγορίθμους.

Εκτός από τη σχετικά αυξημένη πολυπλοκότητα του αλγορίθμου, άλλο ένα σημαντικό μειονέκτημα είναι ότι στα πλαίσια της λεξικογραφικής συσχέτισης στηρίζεται στην ύπαρξη ενός εξωτερικού λεξικού, του Wordnet. Η επιβάρυνση εδώ είναι διττή, λόγω της επιπλέον πολυπλοκότητας της πρόσβασης σε εξωτερικό εργαλείο αλλά και λόγω της εξάρτησης αυτής.

Επίσης, το γεγονός ότι το σύστημα λειτουργεί με ημιαυτόματο τρόπο, το καθιστά ως ανταγωνιστική πρόταση στην αντιμετώπιση της συσχέτισης σχημάτων. Η συσχέτιση σχημάτων γίνεται κατά κανόνα χειροκίνητα, πράγμα που δημιουργεί σημαντικούς περιορισμούς. Η χειροκίνητη συσχέτιση σχημάτων αποτελεί χρονοβόρα και κουραστική διαδικασία, η οποία όταν εφαρμόζεται σε σχήματα μεγάλου μεγέθους ή σε μεγάλο αριθμό σχημάτων, η εφαρμογή της είναι ανέφικτη. Η προσπάθεια που πρέπει να καταβληθεί, κατά τη διάρκεια της χειρωνακτικής συσχέτισης, είναι συνήθως ανάλογη με το πλήθος των σχημάτων που πρέπει να

συσχετιστούν. Απόρροια της χρονικής διάρκειας και της κόπωσης είναι η επιρρέπεια σε λάθη. Ο προτεινόμενος αλγόριθμος, υλοποιώντας μια ημιαυτόματη προσέγγιση της συσχέτισης σχημάτων, καταφέρνει να περιορίσει στο ελάχιστο τη συμμετοχή του χρήστη αλλά παράλληλα επιτρέπει να επέμβει για να αλλάξει τη ροή εκτέλεσης του ή να αξιολογήσει τα αποτελέσματα.

Επίσης, όσον αφορά τη χρονική πολυπλοκότητα της εκτέλεσης του σημασιολογικού αλγορίθμου, αυτός αξιοποιεί είναι ιδιαίτερο χαρακτηριστικό. Ο έλεγχος ήδη παλαιότερων συσχετίσεων που προηγείται της συσχέτισης ζεύγους όρων, έχει ως αποτέλεσμα τη σημαντική εξοικονόμηση χρόνου. Ελέγχοντας ο αλγόριθμος για προϋπάρχουσα συσχέτιση στη βάση δεδομένων του συστήματος που να ικανοποιεί τα κριτήρια συσχέτισης (πχ. το κατώφλι - threshold), επιτυγχάνει να μειώσει σημαντικά το χρόνο εκτέλεσης αφού αποφεύγονται προσβάσεις στο εξωτερικό λεξικό.

Ένα, επιπλέον, δομικό χαρακτηριστικό του αλγορίθμου είναι η υβριδικότητα του. Τα αποτελέσματα της σημασιολογικής συσχέτισης δεν περιορίζονται αποκλειστικά στη λεξικογραφική ή στη σημασιολογική σκοπιά ομοιότητας. Πρόκειται για προσέγγιση του προβλήματος της συσχέτισης σχημάτων τόσο λεξικογραφικά όσο και σημασιολογικά και συντακτικά. Η εξέταση των παραπάνω παραμέτρων καθιστά την πρόταση αυτή περισσότερο σφαιρική, γεγονός που εγγυάται πιο αξιόπιστα αποτελέσματα. Η αξιοπιστία των αποτελεσμάτων αποδεικνύεται, τόσο από τις τιμές των μετρικών που προέκυψαν κατά την αξιολόγηση του αλγορίθμου, όσο και από το γεγονός ότι ο αλγόριθμος είναι αποδοτικός στην περίπτωση που τίθενται προς συσχέτιση σχήματα που μοντελοποιούν διαφορετικές έννοιες. Πιο αναλυτικά, όταν ο αλγόριθμος καλείται να εντοπίσει συσχετίσεις μεταξύ σχημάτων διαφορετικού ερευνητικού τομέα ή περιεχομένου, λειτουργεί αποτελεσματικά εξαιτίας της υβριδικότητας του.

6.1.2 Εργαλείο Διαχείρισης Συσχετίσεων

Η διεπαφή που δημιουργήθηκε στα πλαίσια της ερευνητικής διαδικασίας αποτελεί σημαντική αρωγή για τον χρήστη του συστήματος, αφού μέσω αυτής του δίνεται η δυνατότητα να προβάλει τα προς συσχέτιση σχήματα και να εκτελέσει τον αλγόριθμο συσχέτισης. Με άλλα λόγια, η γραφική αναπαράσταση των σχημάτων υποστηρίζει την εποπτικότητα της διαδικασίας και προσδίδει στη λειτουργία της συσχέτισης διαισθητικότητα.

Επίσης, μέσω της διεπαφής του συστήματος εκτός της εποπτικότητας, ο χρήστης έχει την επιλογή της πρότασης συσχετίσεων χειροκίνητα. Έτσι, επιλέγοντας του κατάλληλους όρους, ο εκάστοτε χρήστης της διεπαφής μπορεί να αποτιμήσει το βαθμό συσχέτισης τους και να τον αποθηκεύσει στη βάση του συστήματος. Επιπλέον, ο χρήστης μπορεί να επικυρώσει ή να απορρίψει τα προτεινόμενα αποτελέσματα των αλγορίθμων. Έτσι ο χρήστης συμμετέχει στη διαδικασία συσχέτισης και παράλληλα, αντιλαμβάνεται τις ενέργειες που εκτελούνται.

Επιπροσθέτως, η επεκτασιμότητα του συστήματος το καθιστά μια από τις πιο αποτελεσματικές λύσεις στο πρόβλημα της ενοποίησης της πληροφορίας, αφού εκτός από τον αλγόριθμο συσχέτισης που υποστηρίζει, μπορεί να ενσωματωθεί κάθε νέος αλγόριθμος σε αυτό με απώτερο σκοπό τη δημιουργία και διατήρηση μιας βιβλιοθήκης αλγορίθμων συσχέτισης. Πιο απλά, το σύστημα παρέχει τη δυνατότητα στους χρήστες του να αναπτύξουν τον αλγόριθμο τους και να τον εντάξουν στη διεπαφή του συστήματος. Με αυτόν τον τρόπο ενθαρρύνεται και η επαναχρησιμοποίηση αποθηκευμένων συσχετίσεων.

Τέλος, ένα από τα σημαντικότερα πλεονεκτήματα της διεπαφής του συστήματος είναι η ιδιότητα της ως διαδικτυακή. Δεδομένου, λοιπόν, ότι πρόκειται για διαδικτυακή διεπαφή αυτό μεταφράζεται ως εύκολα προσβάσιμη από οπουδήποτε χωρίς να προαπαιτείται εγκατάσταση συγκεκριμένου συστήματος στον υπολογιστή εργασίας. Δίνεται με λίγα λόγια η δυνατότητα απομακρυσμένης πρόσβασης και εργασίας.

6.2 Μελλοντικές Επεκτάσεις

Μακροπρόθεσμα, μια παράμετρος που αξίζει να εξεταστεί, λόγω της συμβολής της στον τομέα της συσχέτισης σχημάτων, είναι η εμπλοκή των τεχνικών επιπέδου στιγμιότυπου. Αυτό που προτείνεται, δηλαδή, είναι να διερευνηθεί ο συνδυασμός του προτεινόμενου αλγορίθμου με μια τεχνική που βασίζεται στην πληροφορία που παρέχουν τα δεδομένα στιγμιότυπων. Επίσης, σύμφωνα με τις απαιτήσεις της πλειοψηφίας των χρηστών ανάλογων συστημάτων, μια πρόκληση για τον ερευνητή είναι η απάντηση στο ερώτημα της ολοκλήρωσης της συσχέτισης. Πιο απλά, ο ερευνητής του τομέα της ενοποίησης της πληροφορίας καλείται να απαντήσει πότε έχει ολοκληρωθεί η διαδικασία της συσχέτισης. Η σημαντικότητα της πληροφορίας αυτής είναι προφανής, αφού, ειδικά σε μεγάλα σχήματα, ο χρήστης πρέπει να γνωρίζει αν μπορεί να συνεχίσει την αναζήτηση σχέσεων ανάμεσα σε όρους. Εκτός

από τα παραπάνω, ιδιαίτερη σημασία πρέπει να δοθεί και στα σχήματα που παρουσιάζουν κύκλο στην υλοποίηση τους. Στην τρέχουσα εργασία, δεν αξιοποιείται η πληροφορία του κύκλου σε ένα σχήμα παρά τη σημαντικότητα της και θα ήταν θεμιτό να ληφθεί υπόψη σε μια πιθανή μελλοντική επέκταση.

Όσο αφορά τη διαδικτυακή διεπαφή, μια επέκταση που πρόκειται να υλοποιηθεί άμεσα είναι η γραφική αναπαράσταση των συσχετίσεων ανάμεσα στους όρους. Επίσης, θεμιτό είναι να παρέχεται η δυνατότητα, στο χρήστη του συστήματος, να αποδέχεται ή να απορρίπτει μια προτεινόμενη συσχέτιση από τον εκάστοτε αλγόριθμο που χρησιμοποιήθηκε. Τέλος, καλό θα ήταν να παρέχεται η πληροφορία της εξέλιξης της διαδικασίας, δηλαδή μια μετρική που να αναπαριστά πόσοι όροι, από τους συνολικούς, έχουν συσχετιστεί.

Βιβλιογραφία

[1]. Antoniou, G. and F. van Harmelen, "A Semantic Web Primer", ISBN 0-262-01210-3, April 2004

[2]. Doan, A. and Halevy, A. (2005) "Semantic-Integration Research in the Database Community", *AI Magazine,* Spring 2005, pp. 83-94.

[3]. Rahm, E. and Bernstein, P. (2001) "A survey of approaches to automatic schema matching," *The VLDB Journal*, 10.

[4]. Batini, C., Lenzerini, M. and Navathe, S.B. (1986) "A Comparative Analysis of Methodologies for Database Schema Integration," ACM Computing Surveys, 18(4), pp. 323-364.

[5]. Noy, N., Doan, A., and Halevy, A. (2005) "Semantic Integration," *AI Magazine* ,Spring 2005, pp. 7-9.

[6]. Halevy, A. (2005) "Why Your Data Won't Mix." *ACM Queue,* October 2005, pp. 50-58.

[7]. Li, W. and Clifton, C. (2000), "Semint: A Tool for Identifying Attribute Correspondence in Heterogeneous Databases Using Neural Networks," *Data and Knowledge Engineering*, 33(1), pp. 49-84.

[8]. Yan, L., Miller, R., Haas, L, and Fagin, R. (2001) "Data-Driven Understanding and Refinement of Schema Mappings," *SIGMOD Record*,30(2), pp. 485-496.

[9]. Rahm, E., Do, H., and Massmann, S. (2004) "Matching Large XML Schemas," *ACM SIGMOD Record*, 33(4), pp.26-31.

[10]. Aumueller, D. Do, H., Massmann, S., and Rahm, E. (2005) "Schema and Ontology Matching with COMA++." *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data,* pp. 906-908.

[11]. Bernstein, P., Melnik, S., Petropoulos, M. and Quic, C. (2004) "Industrial-Strength Schema Matching." *SIGMOD Record*, 33(4), pp. 38-43.

[12]. Mike Uschold and Michael Gruninger (2004) "Ontologies and semantics for seamless connectivity" ACM SIGMOD Record, 33(4): 58-64

[13]. Fausto Giunchiglia, Maurizio Marchese, and Ilya Zaihrayeu (2006), "Encoding classifications into lightweight ontologies" in Proc. 3rd European Semantic Web Conference (ESWC), volume 4011 of Lecture notes in computer science, pages 80-94

[14]. Lois Mai Chan, John Comaromi, Joan Mitchell, and Mohinder Satija (1996), "Dewey decimalclassification: a practical guide" OCLC Forest Press, Dublin (OH US)

[15]. Jayant Madhavan, Philip Bernstein, Pedro Domingos, and Alon Halevy (2002), "Representing and reasoning about mappings between domain models" in Proc. 18th National Conference on Artificial Intelligence (AAAI), pages 122-133, Edmonton (CA)

[16]. Michael Brodie, John Mylopoulos, and Joachim Schimdt (1984), "On conceptual modeling" Springer, New York (NY US)

[17]. Peter Chen (1976), "The entity-relationship model – toward a unified view of data" ACM Transactions on Database Systems, 1(1): 9-36

[18]. Jim Melton (ed.) (2003), "Information technology – database languages – SQL" ISO standard ISO/CEI 9075:2003, ISO

[19]. Carlo Batini, Maurizio Lenzerini, and Shamkant Navathe (1986), "A comparative analysis of methodologies for database schema integration" ACM Computing Surveys, 18(4):323-364

[20]. Amit Sheth and James Larson (1990), "Federated database systems for managing distributed, heterogeneous, and autonomous databases" ACM Computing Surveys, 22(3):183-236

[21]. Yuri Breitbart (1990), "Multidatabase interoperability" ACM SIGMOD Record, 19(3):53-60

[22]. Won Kim and Jungyun Seo (1991), "Classifying schematic and data heterogeneity in multidatabase systems" IEEE Computer, 24(12):12-18

[23]. Cheng-Hian Goh (1997), "Representing and reasoning about semantic conflicts in heterogeneous information sources" PhD thesis, MIT, Cambridge (MA US)

[24]. Richard Hull (1997), "Managing semantic heterogeneity in databases: a theoretical perspective" in Proc. 16th Symposium on Principles of Database Systems (PODS), pages 51-61, Tucson (AZ US)

[25]. Vipul Kashyap and Amit Sheth (1998), "Semantic heterogeneity in global information systems: The role of metadata, context and ontologies" in Michael Papazoglou and Gunter Schlageter, editors, Cooperative information systems, pages 139-178, Academic Press, New York (NY US)

[26]. Massimo Benerecetti, Paolo Bouquet, and Chiara Ghidini (2000), "Contextual reasoning distilled" Journal of Experimental and Theoretical Artificial Intelligence, 12(3):279-305

[27]. Holger Wache, Thomas Voegele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hubner (2001), "Ontology-based integration of information – a survey of existing approaches" in Proc. IJCAI Workshop on Ontologies and Information Sharing, pages 108-117, Seattle (WA US)

[28]. Michel Klein (2001), "Combining and relating ontologies: an analysis of problems and solutions" in Proc. IJCAI Workshop on Ontologies and Information Sharing, Seattle (WA US)

[29]. Jerome Euzenat (2001), "Towards a principled approach to semantic interoperability" in Proc. IJCAI Workshop on Ontologies and Information Sharing, Seattle (WA US)

[30]. Oscar Corcho (2004), "A declarative approach to ontology translation with knowledge preservation" PhD thesis, Universidad Politecnica de Madrid, Madrid (ES)

[31]. Adil Hameed, Alun Preece, and Derek Sleeman (2004), "Ontology reconciliation" in Steffen Staab and Rudi Studer, editors, Handbook on ontologies, chapter 12, pages 231-250, Springer Verlag, Berlin (DE)

[32]. Chiara Ghidini and Fausto Giunchiglia (2004), "A semantics for abstraction" in Proc 15th European Conference on Artificial Intelligence (ECAI), pages 343-347, Valencia (ES)

[33]. Paolo Bouquet, Marc Ehrig, Jerome Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krotzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris (2004), "Specification of a common framework for characterizing alignment" Deliverable D2.2.1, Knowledge web NoE

[34]. Jerome Euzenat and Heiner Stuckenschmidt (2003), "The 'family of languages' approach to semantic interoperability" in Borys Omelayenko and Michel Klein, editors, Knowledge transformation for the semantic web, pages 49-63, IOS press Amsterdam (NL)

[35]. Pepijn Visser, Dean Jones, Trevor Bench-Capon, and Michael Shave (1998), "Assessing heterogeneity by classifying ontology mismatches" in Proc. 1st International Conference on Formal Ontology in Information Systems (FOIS), pages 148-162, Trento (IT)

[36]. Massimo Benerecetti, Paolo Bouquet, and Chiara Ghidini (2001), "On the dimensions of context dependence: partiality, approximation, and perspective" in Proc. 3rd International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT), volume 2116 of Lecture notes in computer science, pages 59-72, Dundee (UK)

[37]. Hans Chalupski (2000), "OntoMorph: a translation system for symbolic knowledge" in Proc. 7th International Conference on the Principles of Knowledge Representation and Reasoning (KR), pages 471-482, Breckenridge (CO US)

[38]. Ram, S. and Park, J. (2004) "Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts," *IEEE Transactions on Knowledge and Data Engineering*, 16(2), pp. 189-202.

[39]. Natalya Noy and Mark Musen (2000), "PROMPT: Algorithm and tool for automated ontology merging and alignment" in Proc. 17th National Conference of Artificial Intelligence (AAAI), pages 450-455, Austin (TX US)

[40]. Philip Bernstein and Erhard Rahm (2000), "Data warehouse scenarios for model management" in Proc 19 International Conference on Conceptual Modeling (ER), volume 1920 of Lecture notes in computer science, pages 1-15, Salt Lake City (UT US)

[41]. Maurizio Lenzerini (2002), "Data integration: A theoretical perspective" in Proc. 21st Symposium on Principles of Database Systems (PODS), pages 233-246, Madison (WI US)

[42]. Ilya Zaihrayeu (2006), "Towards Peer-to-Peer Information Management Systems" PhD thesis, International Doctorate School in Information and Communication Technology, University of Trento, Trento (IT)

[43]. Steffen Staab and Heiner Stuckenschmidt (2006), editors, "Semantic web and peer-to-peer", Springer, Heidelberg (DE)

[44]. Philip Bernstein, Fausto Giunchiglia, A. Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu (2002), "Data management for peer-to-peer computing: A vision" in Proc. 5th International Workshop on the Web and Databases (WebDB), Madison (WI US)

[45]. Marie-Christine Rousset, Philippe Adjiman, Philippe Chatalic, Francois Goasdoue, and Laurent Simon (2006), "Somewhere in the semantic web" in Proc. 32nd International Conference on Current Trends in Theory and Practice of Computer Science (SofSem), volume 3831 of Lecture notes in computer science, pages 84-99, Merin (CZ)

[46]. Peter Haase, Bjorn Schnizler, Jeen Broekstra, Marc Ehrig, Frank van Harmelen, Maarteen Menken, Peter Mika, Michal Plechawski, Pawel Pyszlak, Ronny Siebes, Steffen Staab, and Crhistoph Tempich (2004), "Bibster – a semantics-based bibliographic peer-to-peer system" Journal of Web Semantics, 2(1):99-103

[47]. Fausto Giunchiglia and Ilya Zaihrayeu (2002), "Making peer databases interact – a vision for an architecture supporting data coordination" in Proc. 6th International Workshop on Cooperative Information Agents (CIA), pages 18-35, Madrid (ES)

[48]. Pavel Shvaiko, Fausto Giunchiglia, Marco Schorlemmer, Fiona McNeil, Alan Bundy, Maurizio Marchese, Mikalai Yatskevich, Ilya Zaihrayeu, Bo Ho, Vanessa Lopez, Marta Sabou, Joaqin Abian, Ronny Siebes, and Spyros Kotoulas (2006), "Dynamic ontology matching: a survey" Deliverable 3.1, OpenKnowledge STREP

[49]. Zachary Ives, Alon Halevy, Peter Mork, and Igor Tatarinov (2004), "Piazza: mediation and integration infrastructure for semantic web data" Journal of Web Semantics, 1(2):155-175

[50]. Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjorn Naeve, Mikael Nilsson, Matthias Palmer, and Tore Risch (2002), "Edutella: A P2P networking infrastructure based on RDF" in Proc. 11th International World Wide Web Conference (WWW), pages 604-615, Honolulu (HA US)

[51]. Karl Aberer, Philippe Cudre-Mauroux, Aris Ouksel, Tiziana Catarci, Mohand-Said Hacid, Arantza Illarramendi, Vipul Kasyap, Massimo Mecella, Eduardo Mena, Erich Neuhold, Olga De Tryer, Thomas Risse, Monica Scannapieco, Felix Saltor, Luca de Santis, Stefano Spaccapietra, Steffen Staab, and Rudi Studer (2004), "Emergent semantics principles and issues" in Proc. 9th International Conference on Database Systems for Advanced Applications (DASFAA), volume 2973 of Lecture notes in computer science, pages 25-38, Jeju Island (KR)

[52]. Karl Aberer, Tiziana Catarci, Philippe Cudre-Mauroux, Tharam Dillon, Stephan Grimm, Mohand-Said Hacid, Arantza Illarramendi, Mustafa Jarrar, Vipul Kasyap, Massimo Mecella, Eduardo Mena, Erich Neuhold, Aris Ouksel, Thomas Risse, Monica Scannapieco, Felix Saltor, Luca de Santis, Stefano Spaccapietra, Steffen Staab, Rudi Studer, and Olga De Tryer (2004), "Emergent semantic systems" in Proc. 1st International Conference on Semantics of a Networked World (ICSNW), volume 3236 of Lecture notes in computer science, pages 14-43, Paris (FR)

[53]. Anna Zhdanova, Reto Krummenacher, Jan Henke, and Dieter Fensel (2005), "Community-driven ontology management: DERI case study" in Proc. 4th International Conference on Web Intelligence (WI), pages 73-79, Compiegne (FR)

[54]. Dieter Fensel, Holger Lausen, Axel Polleres, Jos de Brujin, Michael Stollberg, Dumitru Roman, and John Domingue (2007), "Enabling semantic web services: the web service modeling ontology" Springer, Heidelberg (DE)

[55]. Massimo Paolucci, Takahiro Kawamura, Terry Payne, and Katia Sycara (2002), "Semantic matching of web services capabilities" in Proc. 1st International Semantic Web Conference (ISWC), volume 2342 of Lecture notes in computer science, pages 333-347, Chia Laguna (IT)

[56]. Brahim Medjahed and Athman Bouguettaya (2005), "A multilevel composability model for semantic web services" IEEE Transactions on Knowledge and Data Engineering, 17(7):954-968

[57]. Swapna Oundhakar, Kunal Verma, Kaarthik Sivashanugam, Amit Sheth, and John Miller (2005), "Dscovery of web services in a multi-ontology and federated registry environment" International Journal of Web Services Research, 2(3):1-32

[58]. Christoph Bussler, Bieter Fensel, and Alexander Madche (2002), "A conceptual architecture for semantic web enabled web services" ACM SIGMOD Record, 31(4):24-29

[59]. Dumitru Roman, Holger Lausen, and Uwe Keller (2004), "Web service modeling ontology standard (WSMO-standard)" Working Draft D2v0.2, WSMO

[60]. Fausto Giunchiglia, Mikalai Yatskevich, and Enrico Giunchiglia (2005), "Efficient semantic matching" in Proc. 2nd European Semantic Web Conference (ESWC), volume 3532 of Lecture notes in computer science, pages 272-289, Hersonisous (GR)

[61]. Dave Robertson, Fausto Guinchiglia, Frank van Harmelen, Maurizio Marchese, Marta Sabou, Marco Schorlemmer, Nigel Shadbolt, Ronnie Siebes, Carles Sierra, Chris Walton, Srinandan Dasmahapatra, Dave Dupplaw, Paul Lewis, Mikalai Yatskevich, Spyros Kotoulas, Adrian Perreu de Pinnick, and Antonis Loizou (2006), "Open knowledge semantic webs through peer-to-peer interaction" Technical Report DIT-06-034, University of Trento

[62]. FIPA0061, FIPA ACL message structure specification (2002), http://www.fipa.org/specs/fipa00061

[63]. FIPA0037, FIPA ACL communicative act library specification, Technical Report (2002), http://www.fipa.org/specs/fipa00037

[64]. Rogier van Eijk, Frank de Boer, Wiebe van de Hoek, and John-Jules Meyer (2001), "On dynamically generated ontology translators in agent communication" International Journal of Intelligent Systems, 16(5):587-607

[65]. Floris Wiesman, Nico Roos, and Paul Volt (2002), "Automatic ontology mappings for agent communication" in Proc. 1st International joint Conference on Autonomous agents and multiagent systems (AAMAS), pages 563-564, Bologna (IT)

[66]. Sidney Bailin and Walt Truszkowski (2002) , "Ontology negotiation: How agents can really get to know each other" in Proc. 1st International Workshop on Radical Agent Concepts (WRAC), volume 2564 of Lecture notes in computer science, pages 320-334, McLean (VA US)

[67]. Jun Wang and Les Gasser (2002), "Mutual online ontology alignment" in Proc. AAMAS Workshop on Ontologies in Agent Systems (OAS), Bologna (IT)

[68]. Jerome Euzenat, Loredana Laera, Valentina Tamma, and Alexandre Viollet (2005), "Negotiation / argumentation techniques among agents complying to different ontologies" Deliverable 2.3.7, Knowledge web NoE

[69]. Loredana Laera, Valentina Tamma, Jerome Euzenat, Trevor Bench-Capon, and Terry Payne (2006), "Reaching agreement over ontology alignments" in Proc. 5th International semantic web Conference (ISWC), volume 4273 of Lecture notes in computer science, pages 371-384, Athens (GA US)

[70]. Joelle Coutaz, James Crowley, Simon Dobson, and David Garlan (2005), "Context is key" Communications of the ACM, 48(3):49-53

[71]. Martin Dzbor, John Domingue, and Enrico Motta (2003), "Magpie – towards a semantic web browser" in Proc. 2nd International Semantic Web Conference (ISWC), volume 2870 of Lecture notes in computer science, pages 690-705, Sanibel Island (FL US)

[72]. Martin Dzbor, Enrico Motta, and John Domingue (2004), "Opening up Magpie via semantic services" in Proc. 3rd International Semantic Web Conference (ISWC), volume 3298 of Lecture notes in computer science, pages 635-649, Hirosima (JP)

[73]. Marta Sabou, Vanessa Lopez, and Enrico Motta (2006), "Ontology selection for the real semantic web: How to cover the Queen birthday dinner?" in Proc. 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW), volume 4248 of Lecture notes in computer science, pages 96-111, Praha (CZ)

[74]. Vanessa Lopez, Michele Pasin, and Enrico Motta (2005), "AquaLog: An ontology-portable question answering system for the semantic web" in Proc. 2nd European Semantic Web Conference (ESWC), volume 3532 of Lecture notes in computer science, pages 546-562, Hersonisous (GR)

[75]. Vanessa Lopez, Enrico Motta, and Victoria Uren (2006), "PowerAqua: Fishing the semantic web" in York Sure and John Domingue, editors, Proc. 3rd European Semantic Web Conference (ESWC), volume 4011 of Lecture notes in computer science, pages 393-410, Budva (ME)

[76]. Eduardo Mena, Vipul Kasyap, Amit Sheth, and Arantza Illarramendi (1996), "Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies" in Proc. 4th International Conference on Cooperative Information Systems (CoopIS), pages 14-25, Brussels (BE)

[77]. Kevin Chang, Bin He, Chengkai Li, Mitesh Patel, and Zhen Zhang (2004), "Structured databases on the web: observavtions and implications" SIGMOD Record, 33(3):61-70

[78]. Christine Parent and Stefano Spaccapietra (2000), "Database integration: the key to data interoperability" in Mike Papazoglou, Stefano Spaccapietra, and Zahir Tari, editors, Object-oriented data modeling, chapter 9, pages 221-253, The MIT Press, Cambridge (MA US)

[79]. Hong-Hai Do, Sergei Melnik, and Erhard Rahm (2002), "Comparison of schema matching evaluations" in Proc. Workshop on Web, Web-Services, and Database Systems, volume 2593 of Lecture notes in computer science, pages 221-237, Erfurt (DE)

[80]. Yannis Kalfoglou and Marco Schorlemmer (2003), "Ontology mapping: the state of the art" The Knowledge Engineering Review, 18(1):1-31

[81]. Natalya Noy (2004), "Semantic integration: A survey of ontology-based approaches" ACM SIGMOD Record, 33(4):65-70

[82]. An-Hai Doan and Alon Halevy (2005), "Semantic integration research in database community: A brief survey" AI Magazine, 26(1):83-94, Special issue on Semantic integration

[83]. Shvaiko, P. and Euzenat, J. (2005) "A Survey of Schema-based Matching Approaches", *Journal on Data Semantic,* 4, pp. 146-171.

[84]. Chris Clifton, Ed Hausman, and Arnon Rosenthal (1997), "Experience with a combined approach to attribute matching across heterogeneous databases" in Proc. 7th IFIP Conference on Database Semantics, pages 428-453, Leysin (CH)

[85]. Luigi Palopoli, Giorgio Terracina, and Domenico Ursino (2003), "DIKE: a system supporting the semi-automatic construction of cooperative information systems from heterogeneous databases" Software-Practice and Experience, 33(9):847-884

[86]. Luigi Palopoli, Domenico Sacca, Giorgio Terracina, and Domenico Ursino (2003), "Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases" IEEE Transactions on Knowledge and Data Engineering, 15(2):271-294

[87]. Luigi Palopoli, Domenico Sacca, and Domenico Ursino (1998), "An automatic technique for detecting type conflicts in database schemes" in Proc. 7th International Conference on Information and Knowledge Management (CIKM), pages 306-313, Bethesda (ML US)

[88]. Luigi Palopoli, Luigi Pontieri, Giorgio Terracina, and Domenico Ursino (2000), "Intensional and extensional integration and abstraction of heterogeneous databases" Data and Knowledge Engineering, 35()3:201-237

[89]. Silvana Castano, Valeria De Antonellis, and Sabrina De Capitani di Vimercati (2000), "Global viewing of heterogeneous data sources" IEEE Transactions on Knowledge and Data Engineering, 13(2):277-297

[90]. Sonia Bergamaschi, Silvana Castano, and Maurizio Vincini (1999), "Semantic integration of semistructured and structured data sources" ACM SIGMOD Record, 28(1): 54-59

[91]. Sonia Bergamaschi, Domenico Beneventano, Silvana Castano, and Maurizio Vincini (1998), "MOMIS: An intelligent system for the integration of semistructured and structured data" Technical Report T3R07, Universita di Modena e Reggio Emilia, Modena (IT)

[92]. Domenico Beneventano, Sonia Bergamaschi, Stefano Lodi, and Claudio Sartori (1998), "Consistency checking in complex object database schemata with integrity constraints" IEEE Transactions on Knowledge and Data Engineering, 10(4):576-598

[93]. Natalya Noy and Mark Musen (2001), "Anchor-PROMPT: Using non-local context for semantic matching" in Proc. IJCAI Workshop on Ontologies and Information Sharing, pages 63-70, Seattle (WA US)

[94]. Natalya Noy and Mark Musen (1999), "SMART: Automated support for ontology merging and alignment" in Proc. 12th Wotkshop on Knowledge Acquisition, Modeling and Management (KAW), Banff (CA)

[95]. Natalya Noy and Mark Musen (2002), "PromptDiff: A fixed-point algorithm for comparing ontology versions" in Proc. 18th National Conference on Artificial Intelligence (AAAI), pages 744-750, Edmonton (CA)

[96]. Natalya Noy and Mark Musen (2003), "The PROMPT suite: interactive tools for ontology merging and mapping" International Journal of Human-Computer Studies, 59(6):983-1024

[97]. Giovanni Modica, Avigdor Gal, and Hasan Jamil (2001), "The use of machine-generated ontologies in dynamic information seeking" in Proc. 9th International Conference on Cooperative Information Systems (CooIS), volume 2172 of Lecture notes in computer science, pages 433-448, Trento (IT)

[98]. Jayant Madhavan, Philip Bernstein, and Erhard Rahm (2001), "Generic schema matching with Cupid" in Proc. 27th International Conference on Very Large Data Bases (VLDB), pages 48-58, Roma (IT)

[99]. Hong-Hai Do and Erhard Rahm (2002), "COMA – a system for flexible combination of schema mathing approaches" in Proc. 28th International Conference on Very Large Data Bases (VLDB), pages 610-621, Hong Kong (CN)

[100]. Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm (2002), "Similarity flooding: a versatile graph matching algorithm" in Proc. 18th International Conference on Data Engineering (ICDE), pages 117-128, San Jose (CA US)

[101]. MDC, Open information model, version 1.0, (1999) http://mdcinfo/oim/oim10.htm

[102]. Yuan An, Alexander Borgida, and John Mylopoulos (2005), "Constructing complex semantic mappings between XML data and ontologies" in Proc. 4th International Semantic Web Conference (ISWC), volume 3729 of Lecture notes in computer science, pages 6-20, Galway (IE)

[103]. Yuan An, Alexander Borgida, and John Mylopoulos (2005), "Inferring complex semantic mappings between relational tables and ontologies from simple correspondences" in Proc. 4th International Conference on Ontologies, Databases and Applications of Semantics (ODBASE), volume 3761 of Lecture notes in computer science, pages 1152-1169, Agia Napa (CY)

[104]. Yuan An, Alexander Borgida, and John Mylopoulos (2006), "Discovering the semantics of relational tables through mappings" Journal on Data Semantics, VII:1-32

[105]. Dejing Dou, Drew McDermott, and Peishen Qi (2005), "Ontology translation on the semantic web" Journal on Data Semantics, II:35-57

[106]. Drew McDermott and Dejing Dou (2002), "Representing disjunction and quantifiers in RDF" in Proc. 1st International Semantic Web Conference (ISWC), volume 2342 of Lecture notes in computer science, pages 250-263, Chia Laguna (IT)

[107]. Fausto Giunchiglia and Pavel Shvaiko (2003), "Semantic matching" The Knowledge Engineering Review, 18(3):265-280

[108]. An-Hai Doan, Pedro Domingos, and Alon Halevy (2001), "Reconciling schemas of disparate data sources: A machine-learning approach" in Proc. 20th International Conference on Management of Data (SIGMOD), pages 509-520, Santa Barbara (CA US)

[109]. An-Hai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy (2004), "Ontology matching: a machine-learning approach" in Steffen and Rudi Studer, editors, Handbook on ontologies, chapter 18, pages 385-404, Springer Verlag, Berlin (DE)

[110]. Robin Dhamankar, Yoonkyong Lee, An-Hai Doan, Alon Halevy, and Pedro Domingos (2004), "iMAP: Discovering complex semantic matches between database schemas" in Proc. 23rd International Conference on Management of Data (SIGMOD), pages 383-394, Paris (FR)

[111]. Stuart Russell and Peter Norving (1995), "Artificial intelligence: a modern approach" Prentice Hall, Englewood Cliffs (NJ US)

[112]. Jacob Berlin and Amihai Motro (2002), "Database schema matching using machine learning with feature selection" in Proc. 14th International Conference on Advanced Information Systems Engineering (CAiSE), volume 2348 of Lecture notes in computer science, pages 452-466, Toronto (CA)

[113]. Wen-Syan Li and Chris Clifton (1994), "Semantic integration in heterogeneous databases using neural networks" in Proc. 10th International Conference on Very Large Data Bases (VLDB), pages 1-12, Santiago (CL)

[114]. Wen-Syan Li and Chris Clifton (2000), "SEMINT: a tool for identifying attribute correspondences in heterogeneous databases using neural networks" Data and Knowlwdge Engineering, 33(1):49-84

[115]. Renee Miller, Laura Haas, and Mauricio Hernandez (2000), "Schema mapping as query discovery" in Proc. 26th International Conference on Very Large Data Bases (VLDB), pages 77-88, Cairo (EG)

[116]. Renee Miller, Mauricio Hernandez , Laura Haas, Lingling Yan, Howard Ho, Ronald Fagin, and Lucian Popa (2001), "The Clio project: managing heterogeneity" ACM SIGMOD Record, 30(1):78-83

[117]. Felix Naumann, Ching-Tien Ho, Xuqing Tian, Laura Haas, and Nimrod Megiddo (2002), "Attribute classification using feature analysis" in Proc. 18th International Conference on Data Engineering (ICDE), page 271, San Jose (CA US)

[118]. Laura Haas, Mauricio Hernandez, Howard Ho, Lucian Popa, and Mary Roth (2005), "Clio grows up: from research prototype to industrial tool" in Proc. 24th International Conference on Management of Data (SIGMOD), pages 805-810, Baltimore (MD US)

[119]. Jerome Euzenat and Petko Valtchev (2004), "Similarity-based ontology alignment in OWL-Lite" in Proc. 15th European Conference on Artificial Intelligence (ECAI), pages 333-337, Valencia (ES)

[120]. Madhavan, J., Bernstein, P., Doan, A., and Halevy, A. (2005) "Corpus-based Schema Matching," *Proceedings of the twenty-first International Conference on Data Engineering*.

[121]. Marc Ehrig, Steffen Staab, and York Sure (2005), "Bootstrapping ontology alignment methods with APFEL" in Proc. 4th International Semantic Web Conference (ISWC), volume 3729 of Lecture notes in computer science, pages 186-200, Galway (IE)

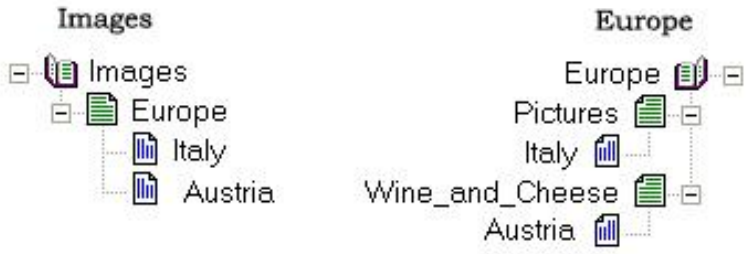
[122]. Mayssam Sayyadian, Yoonkyong Lee, An-Hai Doan, and Arnon Rosenthal (2005), "Tuning schema matching software using synthetic scenarios" in Proc. 31st International Conference on Very Large Data Bases (VLDB), pages 994-1005, Trondheim (NO)

[123]. Jerome Euzenat, Pavel Shvaiko (2007), "Ontology Matching", Springer-Verlag, Berlin Heidelberg, ISBN: 3-540-49611-4

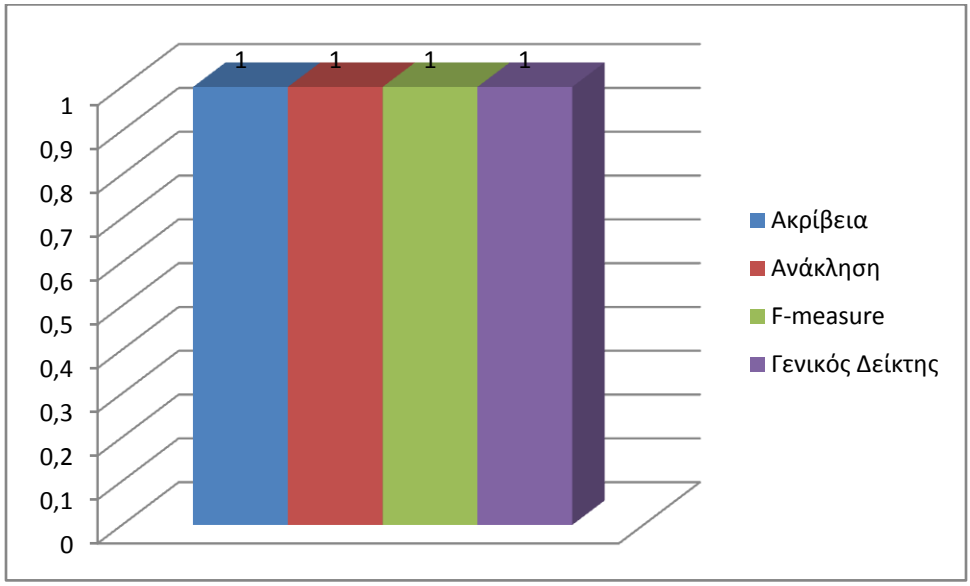
Παράρτημα

Α. Πειράματα

Πείραμα 1ο



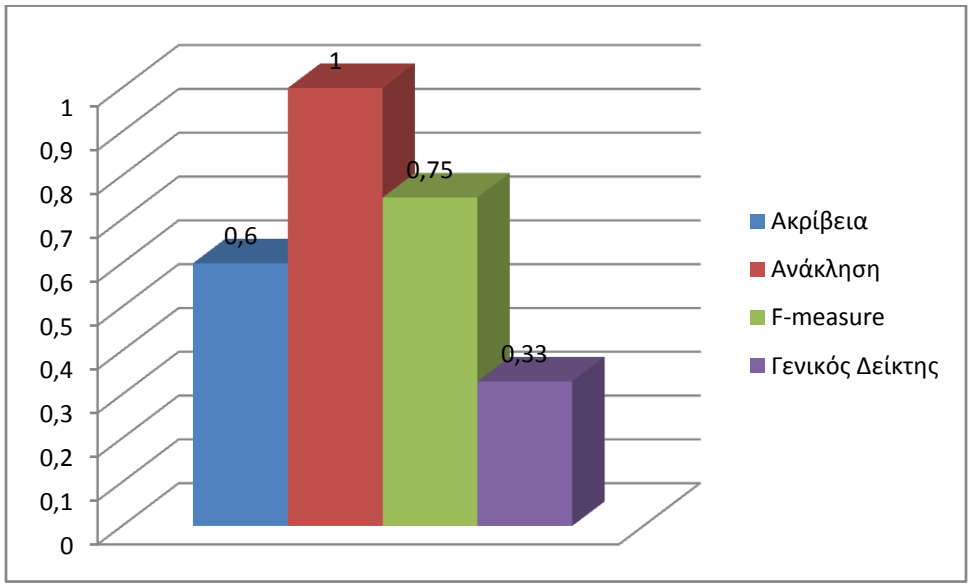
Εικόνα 1. Σχήματα προς συσχέτιση, Images – Europe



Εικόνα 2. Αποτελέσματα σύγκρισης, Images – Europe

Πείραμα 2ο

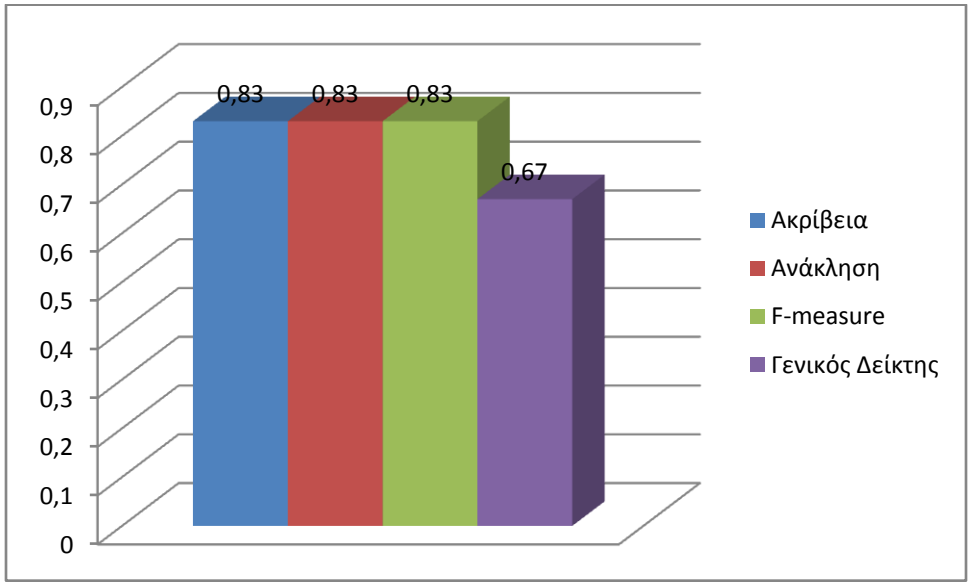
Εικόνα 3. Σχήματα προς συσχέτιση, Purchase Order 1 – Purchase Order 2



Εικόνα 4. Αποτελέσματα σύγκρισης, Purchase Order 1 – Purchase Order 2

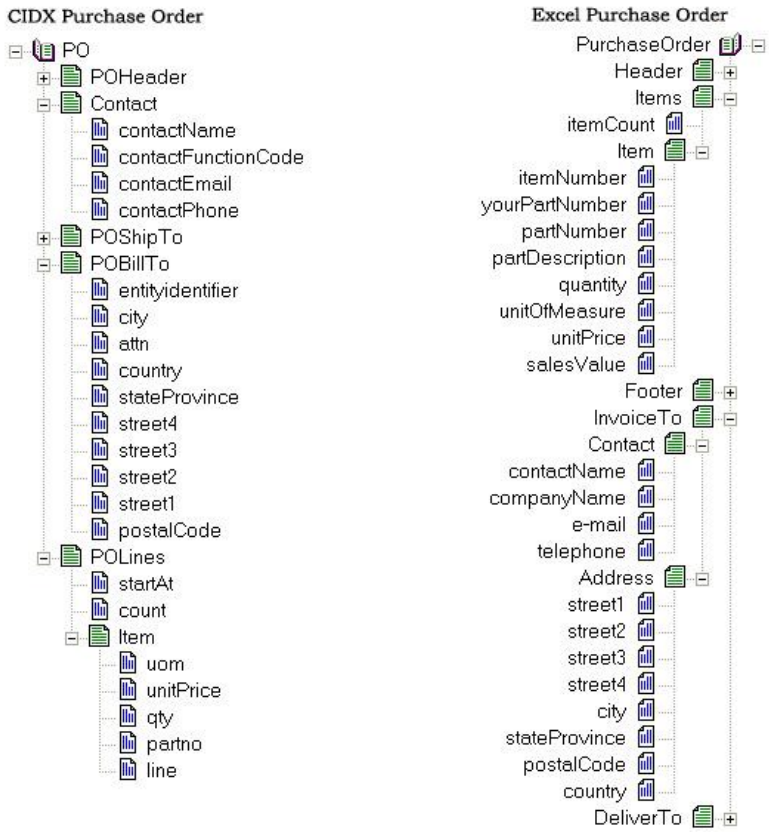
Πείραμα 3ο

Εικόνα 5. Σχήματα προς συσχέτιση, Google web directory (mini) – Yahoo web directory (mini)

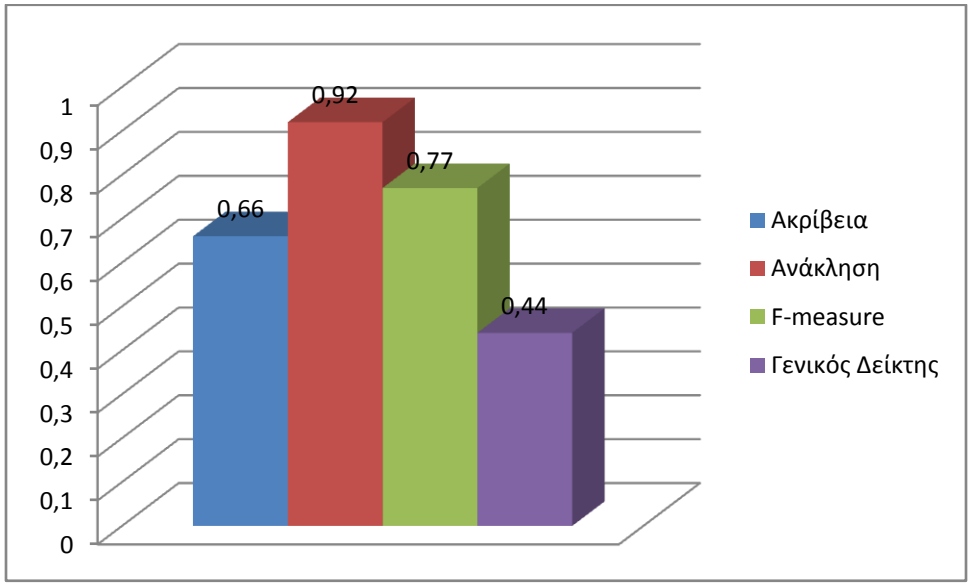


Εικόνα 6. Αποτελέσματα σύγκρισης, Google web directory (mini) – Yahoo web directory (mini)

Πείραμα 4ο

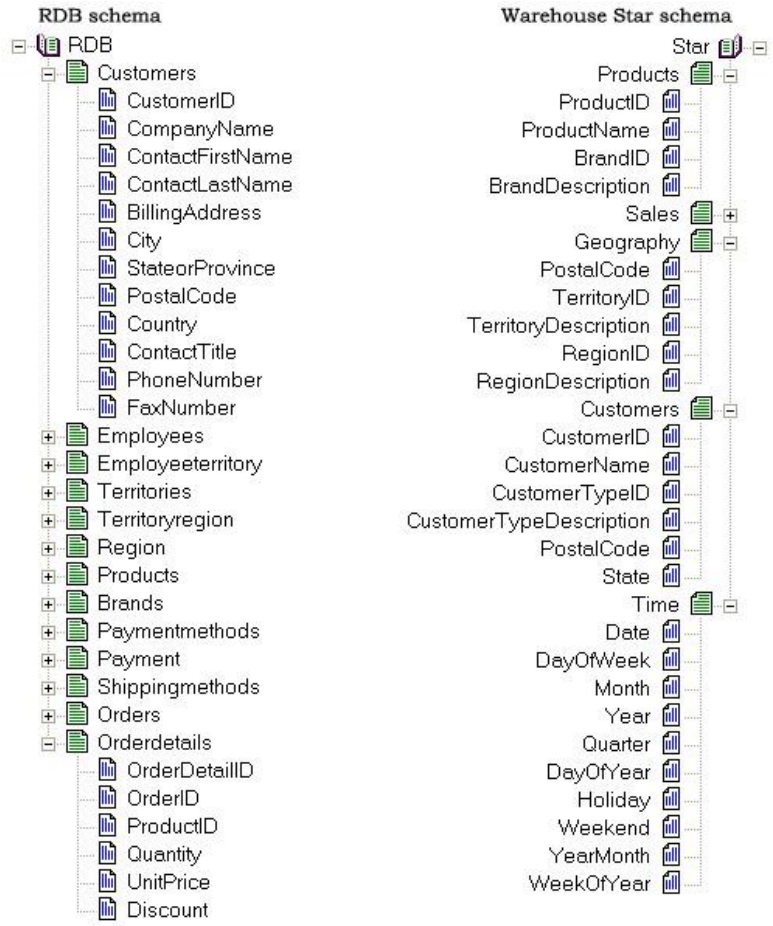


Εικόνα 7. Σχήματα προς συσχέτιση, CIDX Purchase Order – Excel Purchase Order

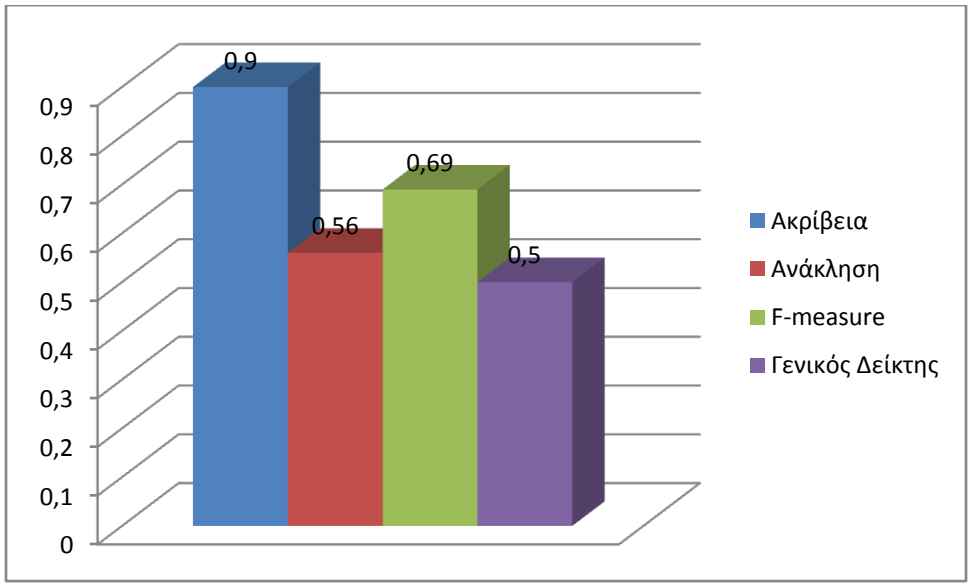


Εικόνα 8. Αποτελέσματα σύγκρισης, CIDX Purchase Order – Excel Purchase Order

Πείραμα 5ο

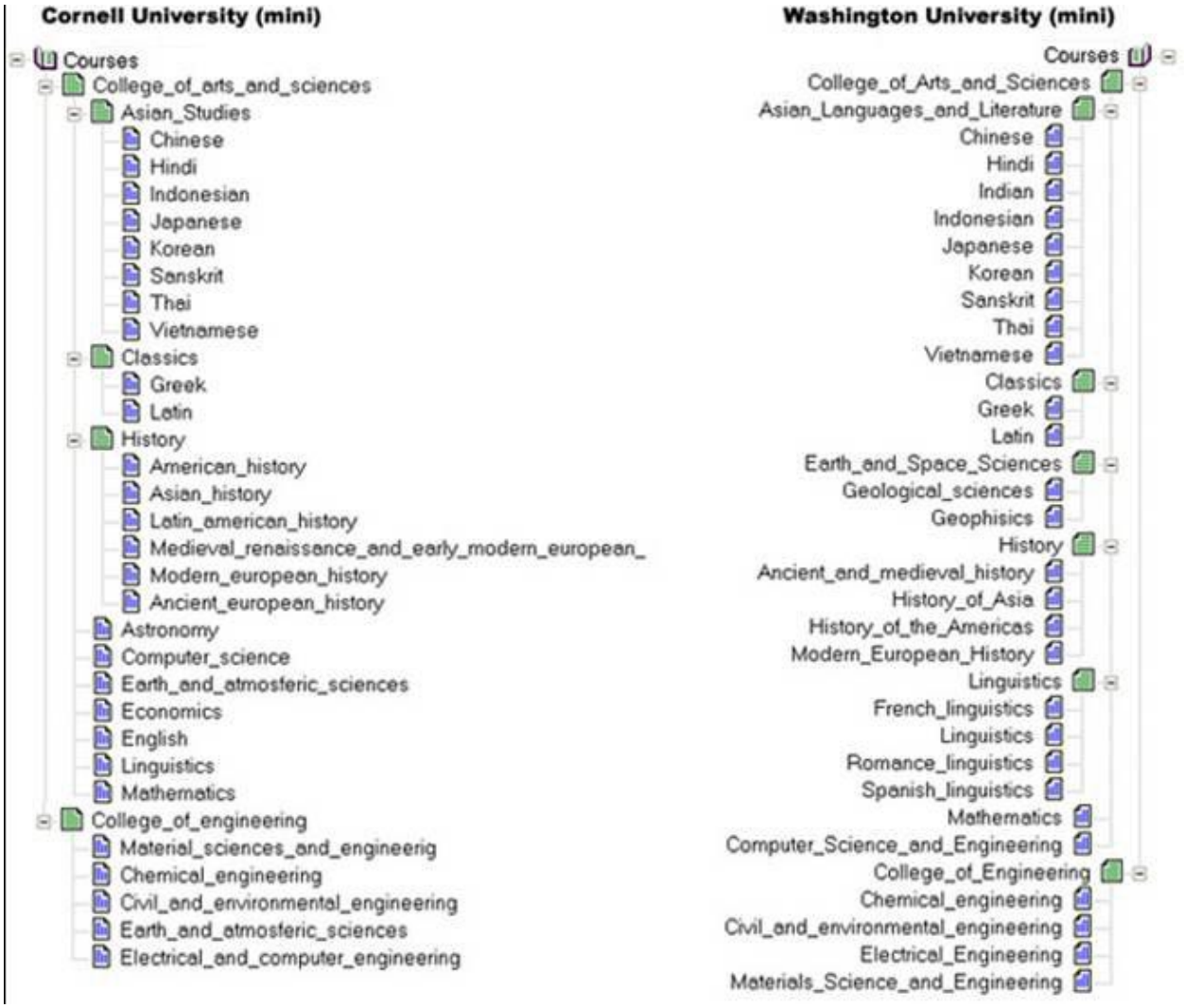


Εικόνα 9. Σχήματα προς συσχέτιση, RDB schema – Warehouse Star schema

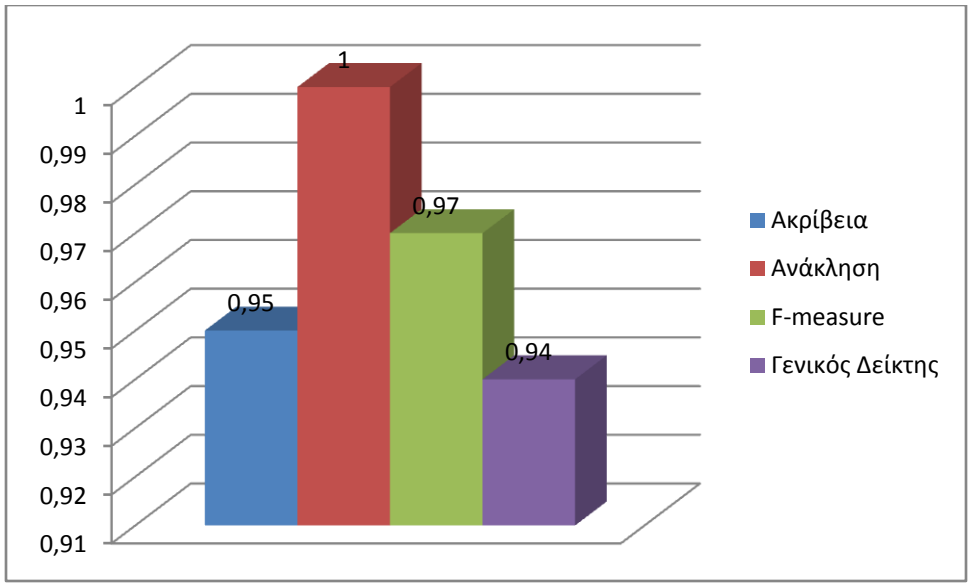


Εικόνα 10. Αποτελέσματα σύγκρισης, RDB schema – Warehouse Star schema

Πείραμα 6ο

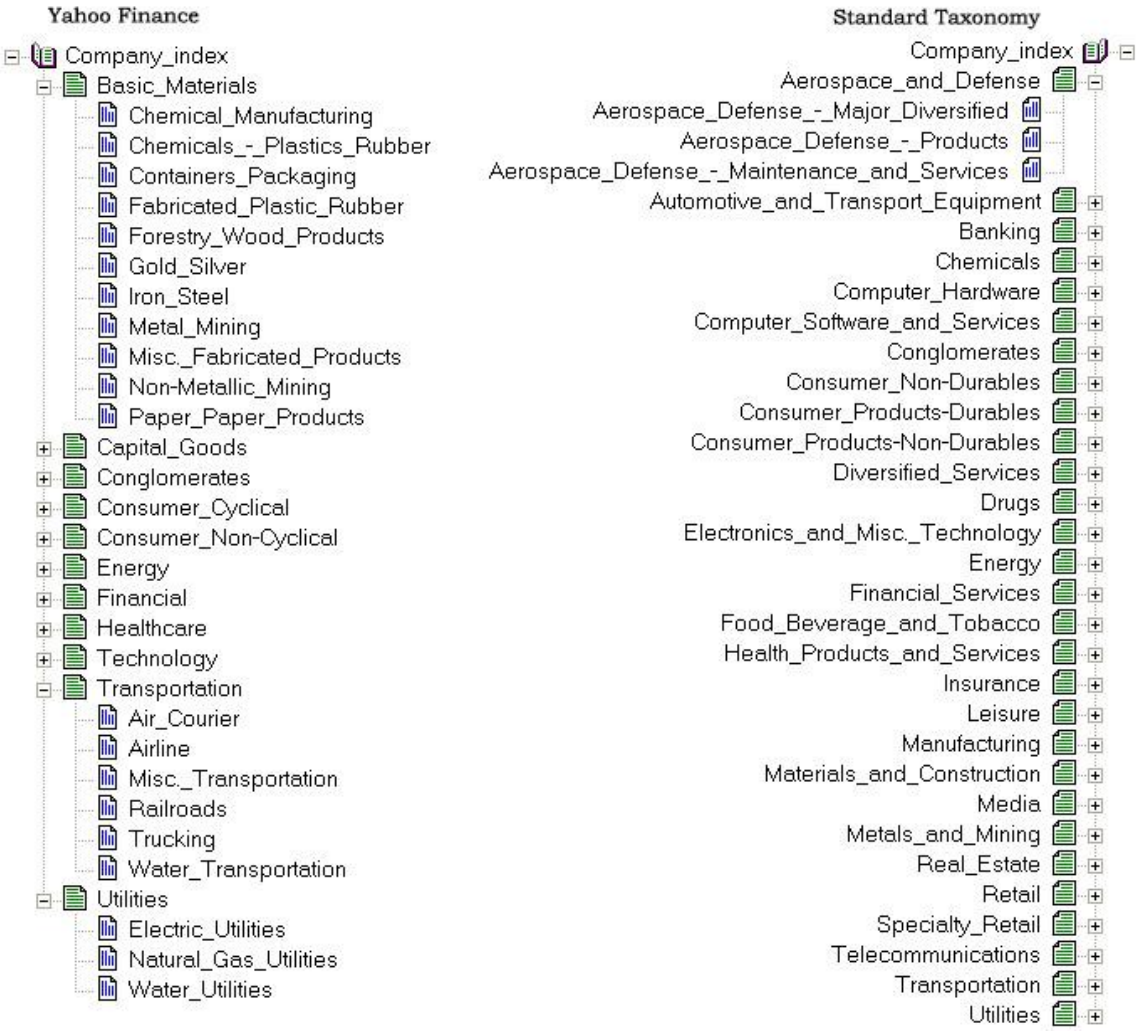


Εικόνα 11. Σχήματα προς συσχέτιση, Cornell University (mini) – Washington University (mini)

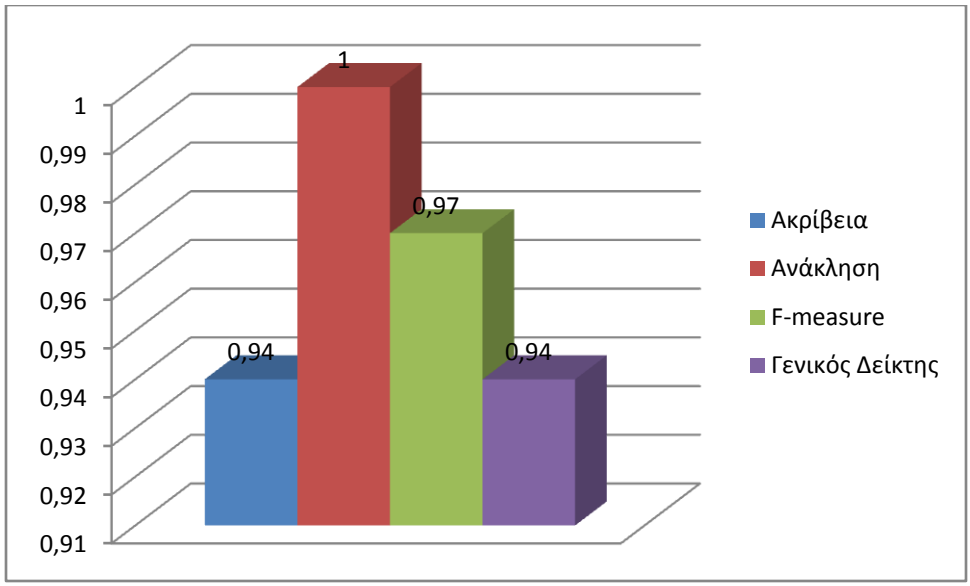


Εικόνα 12. Αποτελέσματα σύγκρισης, Cornell University (mini) – Washington University (mini)

Πείραμα 7ο

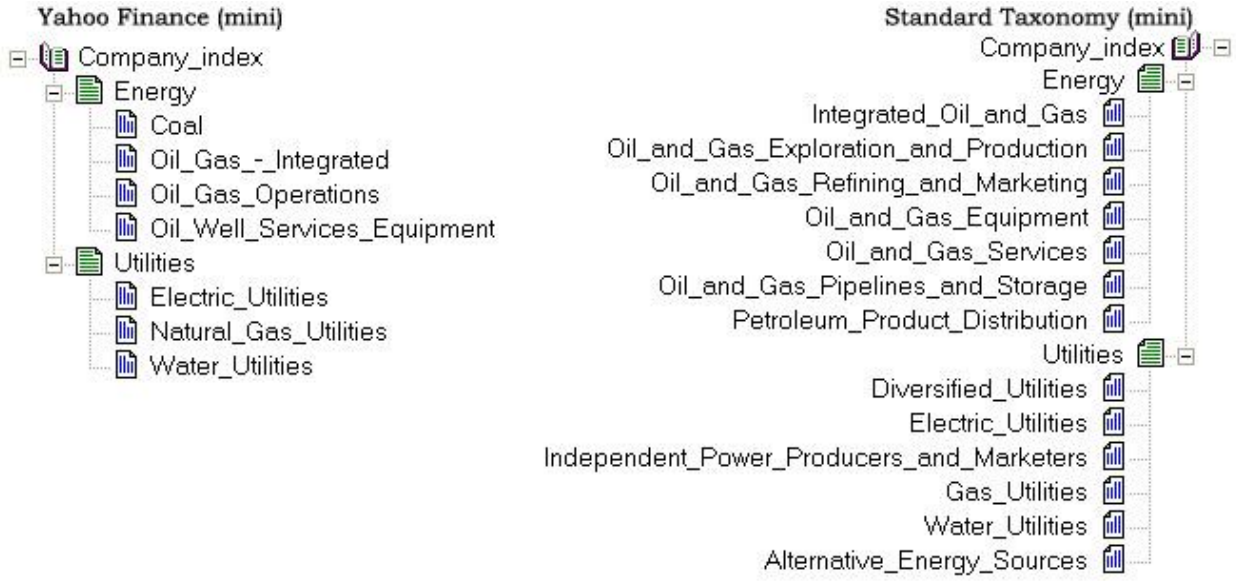


Εικόνα 13. Σχήματα προς συσχέτιση, Yahoo Finance – Standard Taxonomy

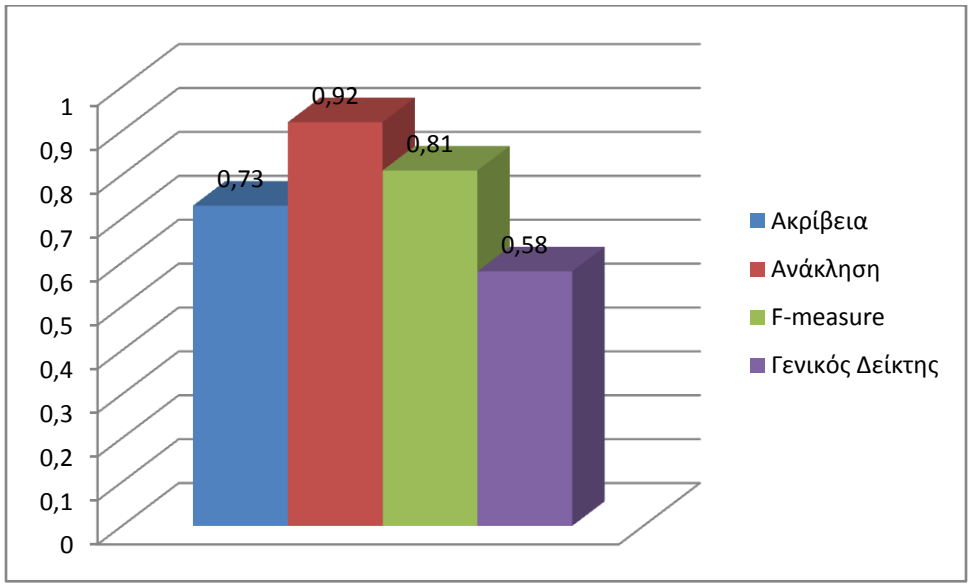


Εικόνα 14. Αποτελέσματα σύγκρισης, Yahoo Finance – Standard Taxonomy

Πείραμα 8ο

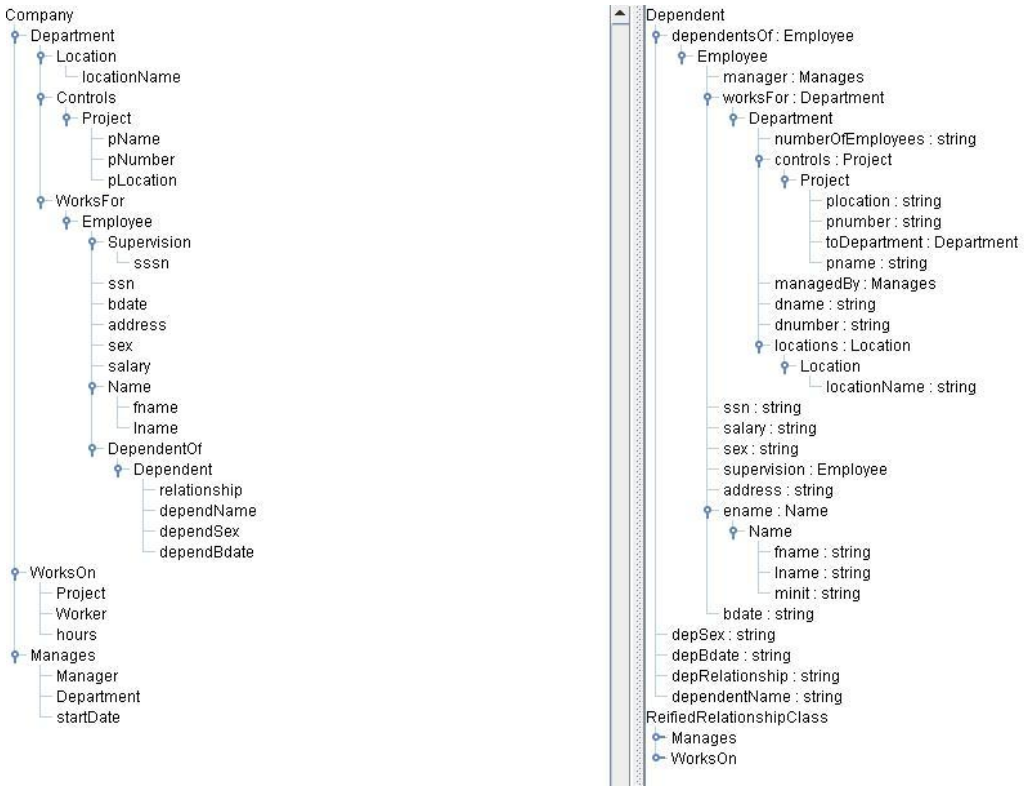


Εικόνα 15. Σχήματα προς συσχέτιση, Yahoo Finance (mini) – Standard Taxonomy (mini)

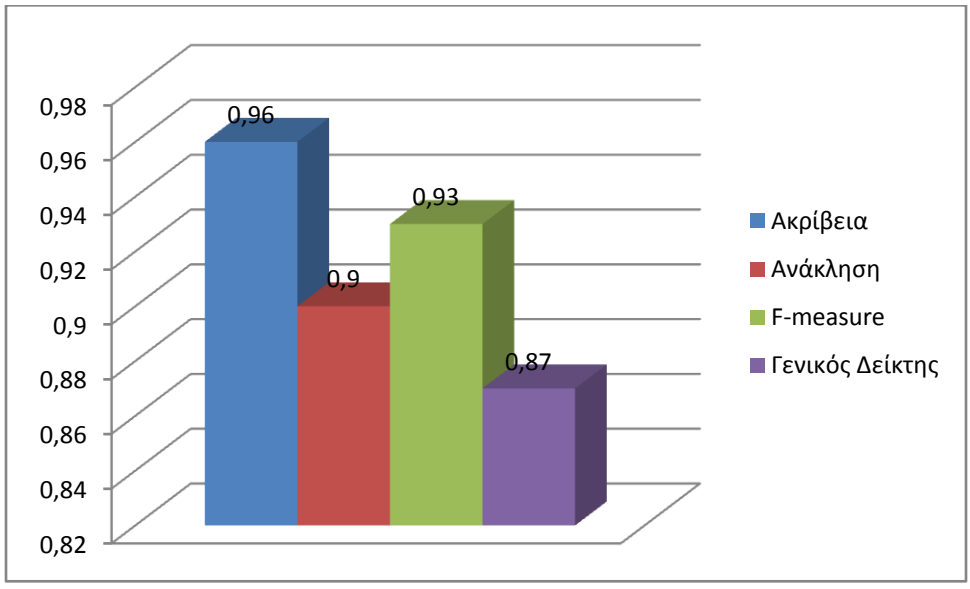


Εικόνα 16. Αποτελέσματα σύγκρισης, Yahoo Finance (mini) – Standard Taxonomy (mini)

Πείραμα 9ο

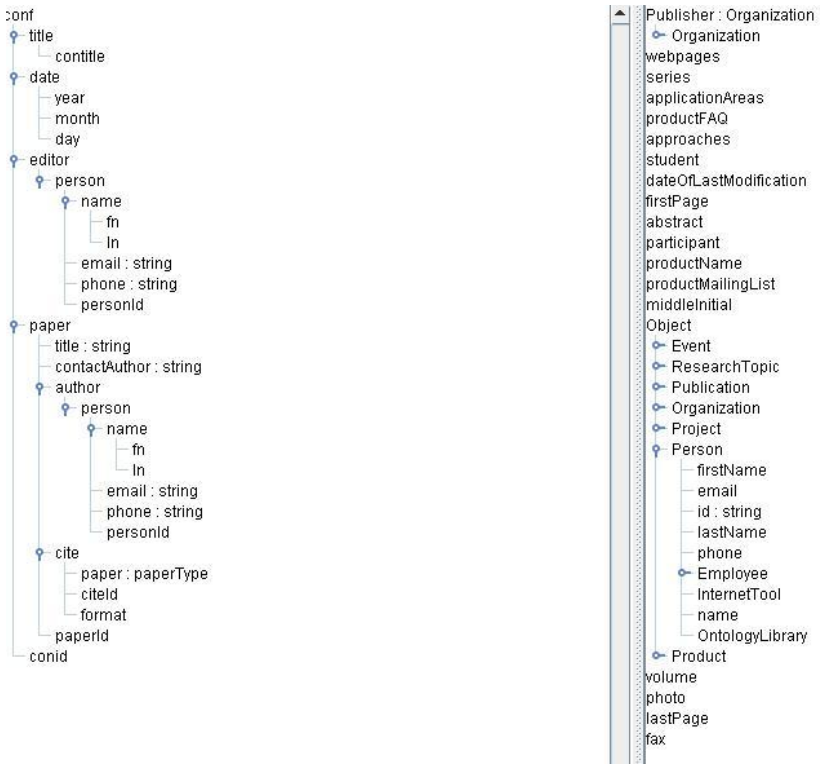


Εικόνα 17. Σχήματα προς συσχέτιση, Company – Company-er

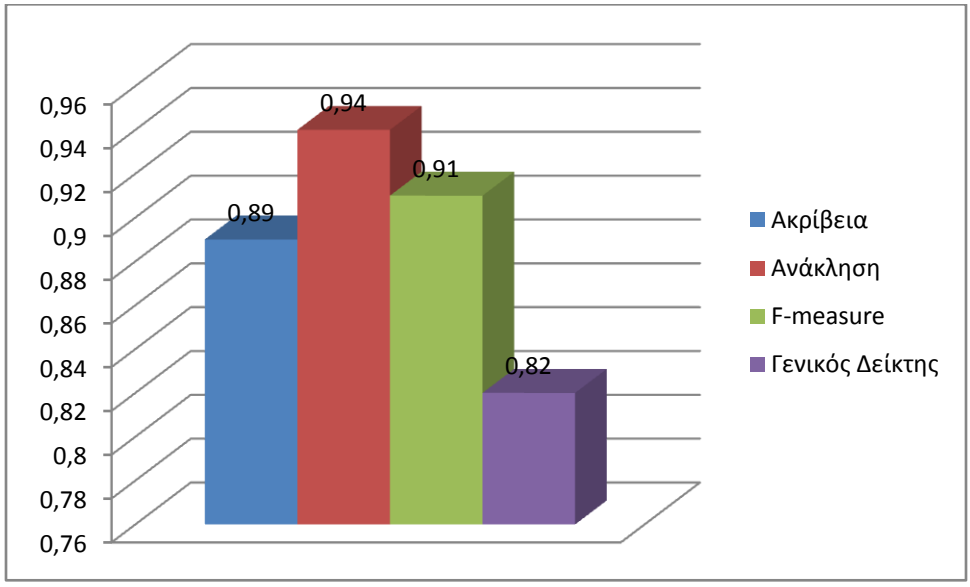


Εικόνα 18. Αποτελέσματα σύγκρισης, Company – Company-er

Πείραμα 10ο

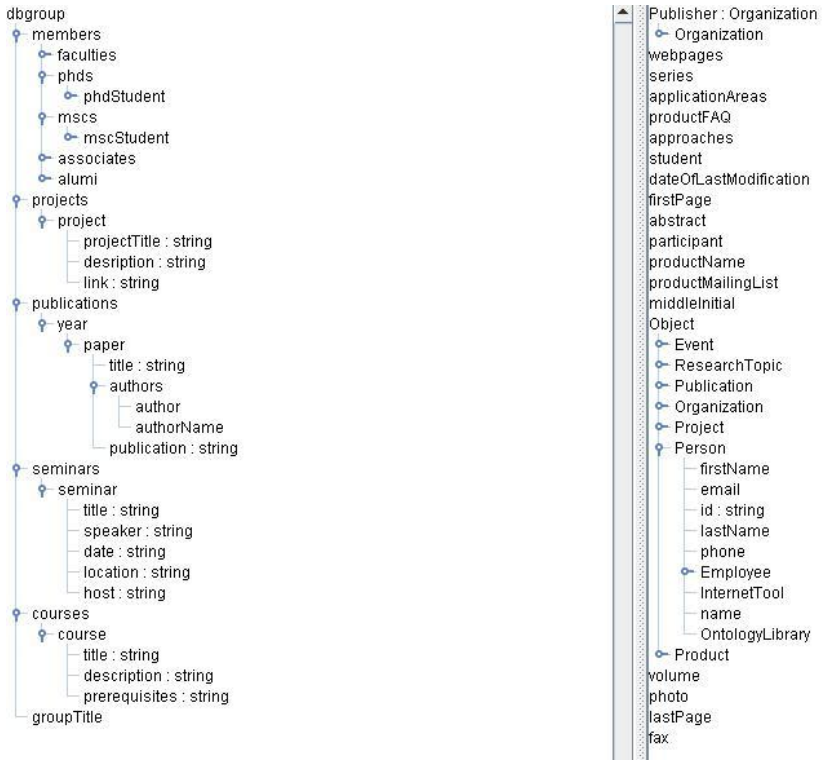


Εικόνα 19. Σχήματα προς συσχέτιση, Conference – Ka

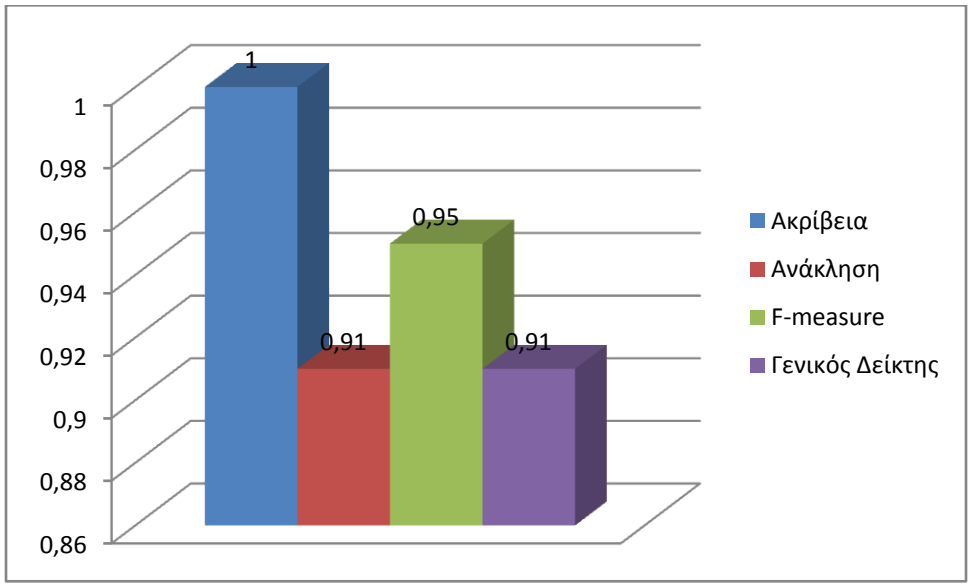


Εικόνα 20. Αποτελέσματα σύγκρισης, Conference – Ka

Πείραμα 11ο

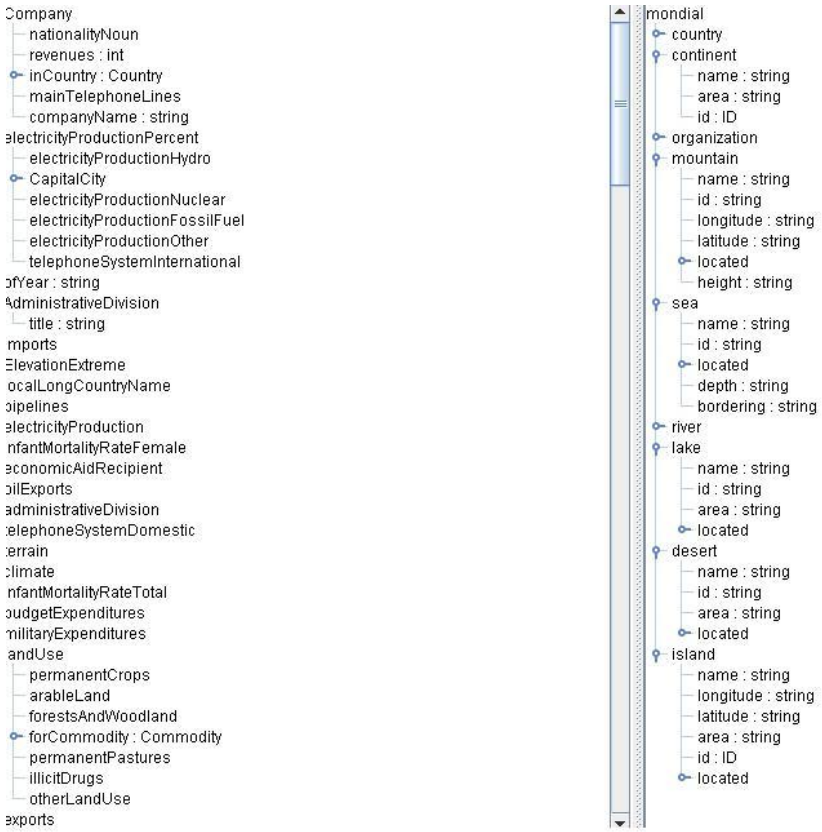


Εικόνα 21. Σχήματα προς συσχέτιση, dbgroup – Ka

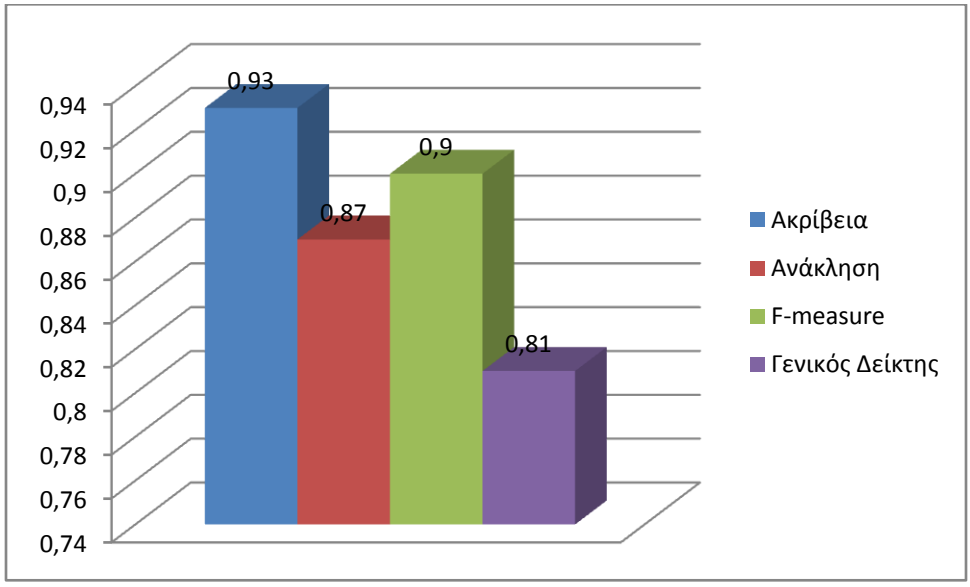


Εικόνα 22. Αποτελέσματα σύγκρισης, dbgroup – Ka

Πείραμα 12ο

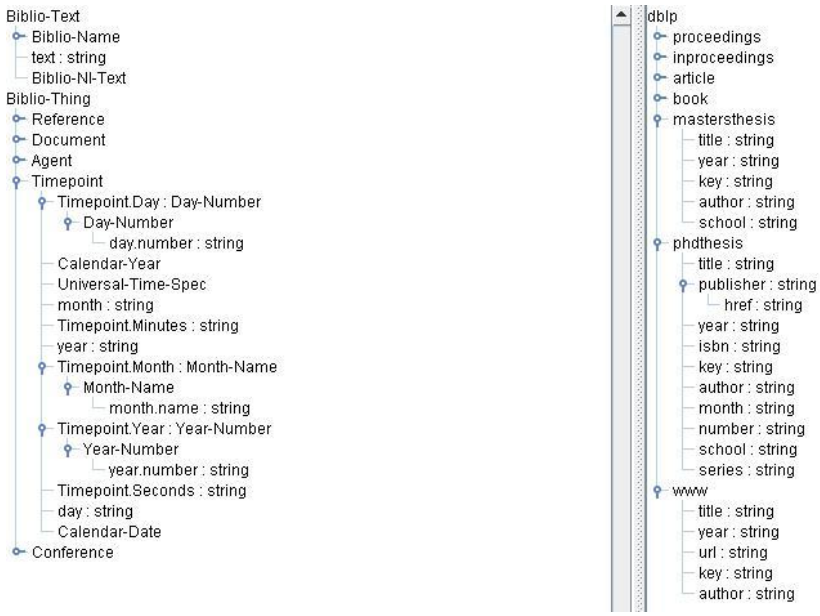


Εικόνα 23. Σχήματα προς συσχέτιση, factbook-owl – mondial

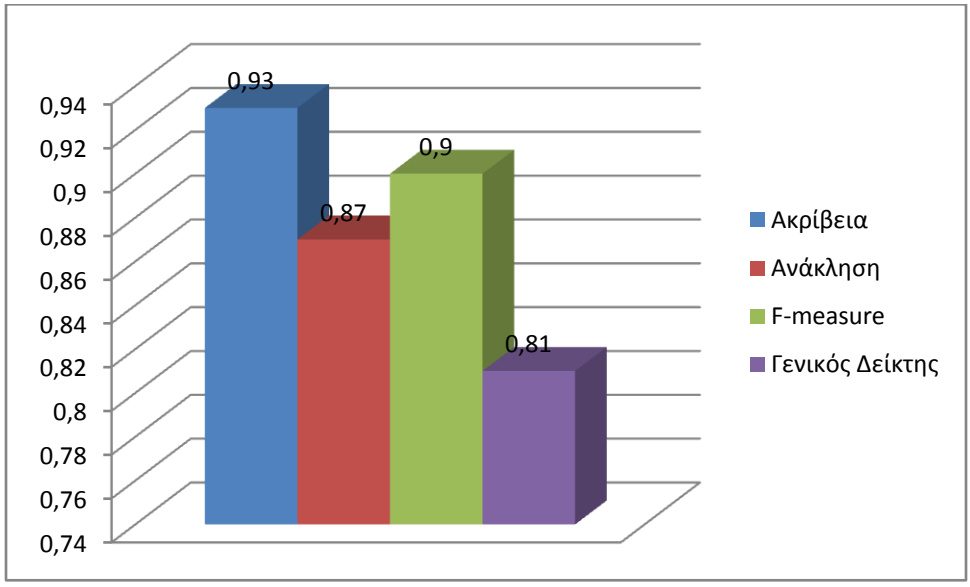


Εικόνα 24. Αποτελέσματα σύγκρισης,factbook-owl – mondial

Πείραμα 13ο



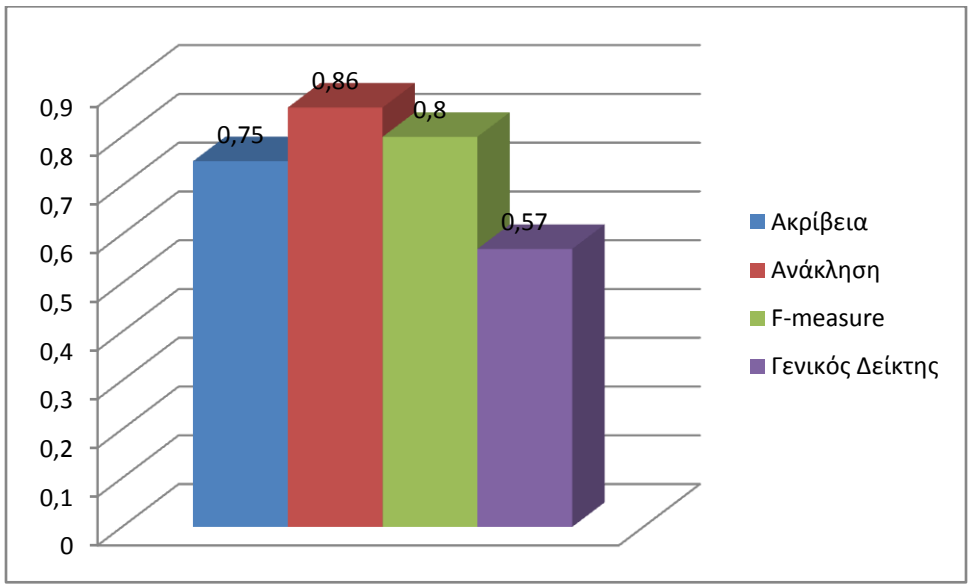
Εικόνα 25. Σχήματα προς συσχέτιση, Bibliographic-Data - DBLP



Εικόνα 26. Αποτελέσματα σύγκρισης, Bibliographic-Data – DBLP

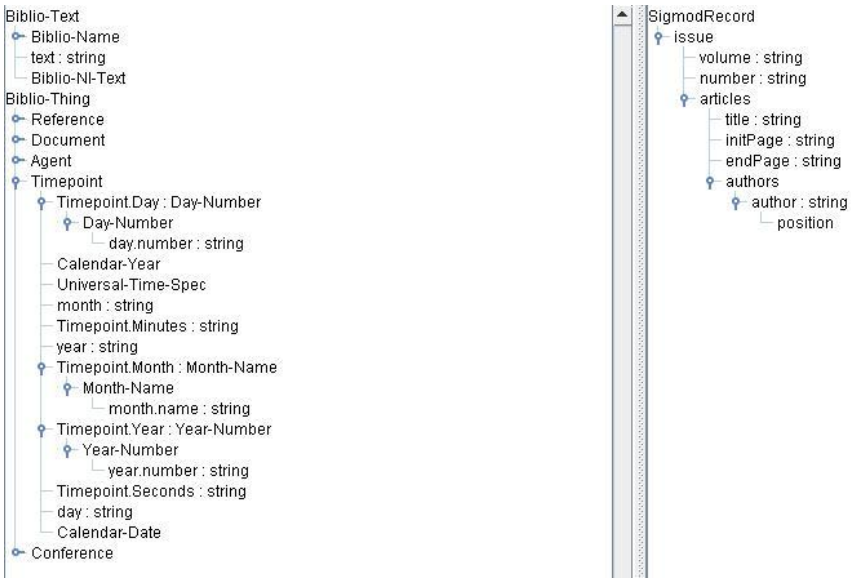
Πείραμα 14ο

Εικόνα 27. Σχήματα προς συσχέτιση, Bibliographic-Data – targetDBLP

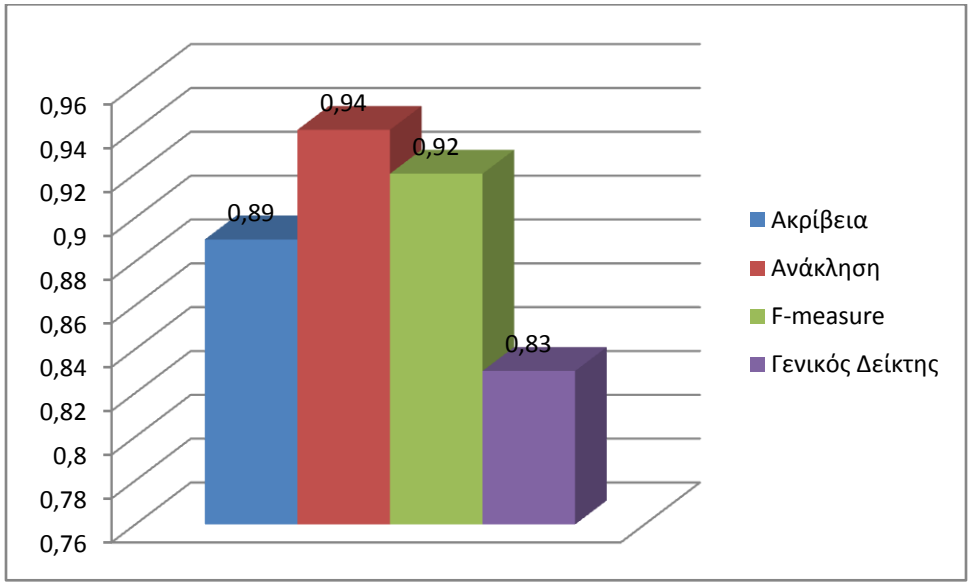


Εικόνα 28. Αποτελέσματα σύγκρισης, Bibliographic-Data – targetDBLP

Πείραμα 15ο

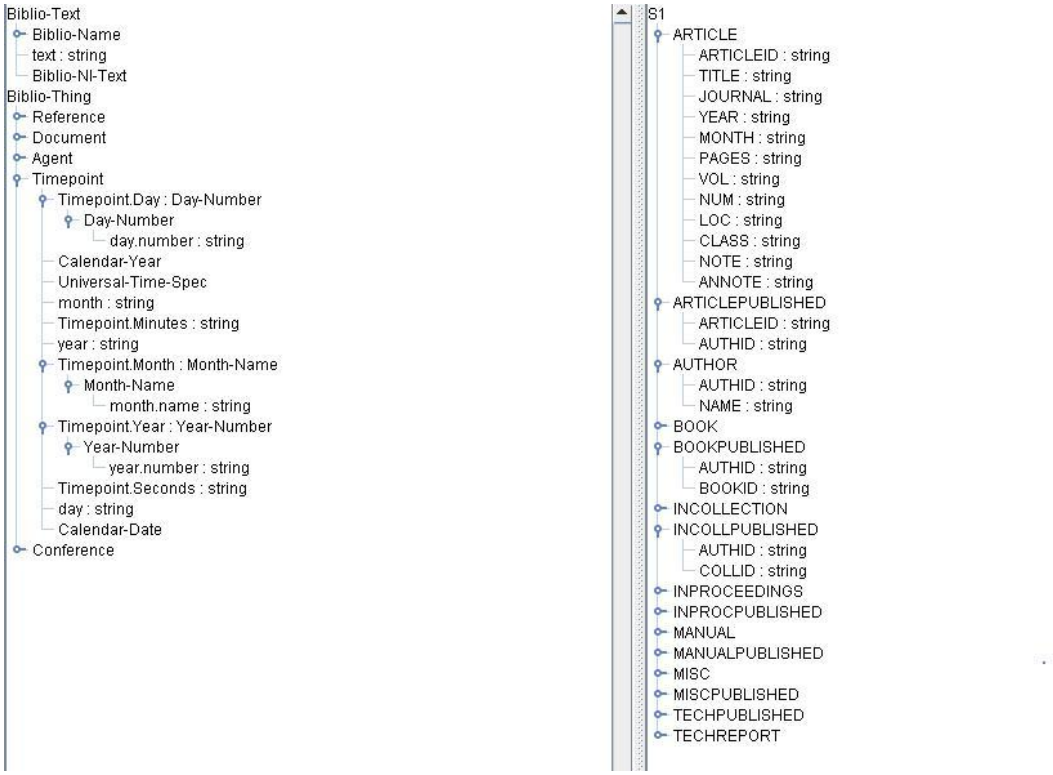


Εικόνα 29. Σχήματα προς συσχέτιση, Bibliographic-Data – sigmodRecord

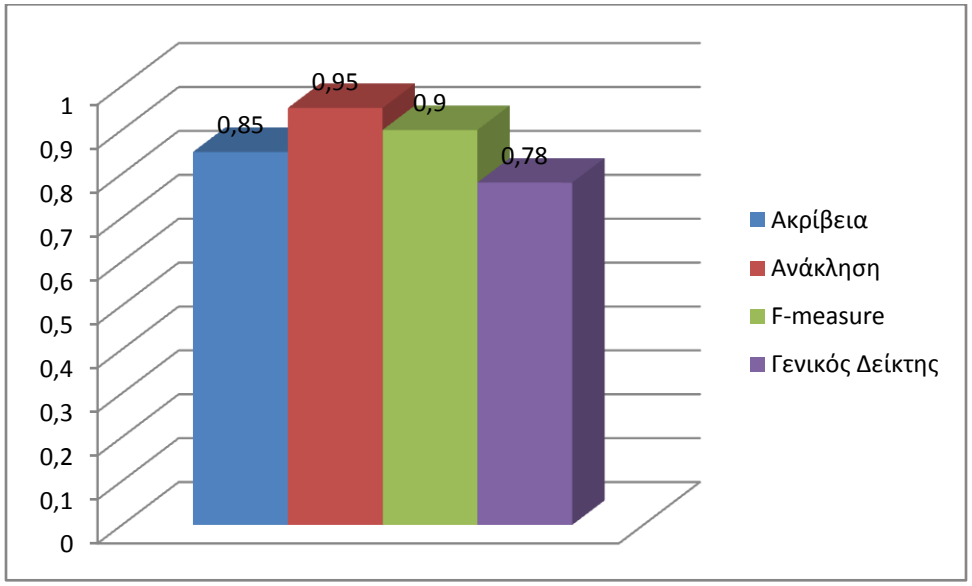


Εικόνα 30. Αποτελέσματα σύγκρισης, Bibliographic-Data – sigmodRecord

Πείραμα 16ο



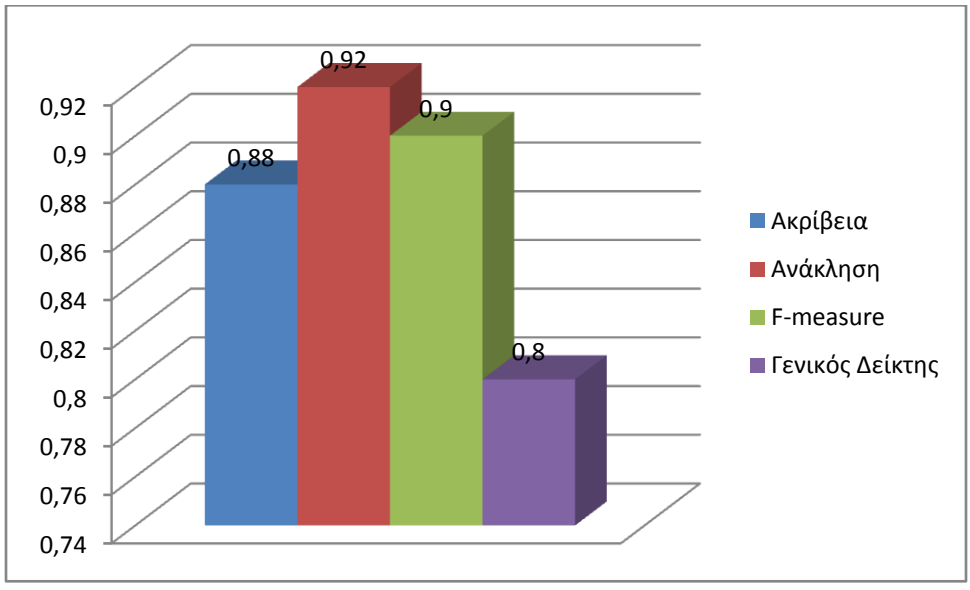
Εικόνα 31. Σχήματα προς συσχέτιση, Bibliographic-Data – amalgam1



Εικόνα 32. Αποτελέσματα σύγκρισης, Bibliographic-Data – amalgam1

Πείραμα 17ο

Εικόνα 33. Σχήματα προς συσχέτιση, Bibliographic-Data – amalgam2



Εικόνα 34. Αποτελέσματα σύγκρισης, Bibliographic-Data – amalgam2

Β. Κώδικας Σημασιολογικού Αλγορίθμου

```
public static float[][] syntacticMatching(Model sourceModel, Model targetModel,double min,double
max, String strategy_mod, Mapping approvedMapping){

	if (sourceModel==null || targetModel==null)
		return null;

	ArrayList sourceTraversal = sourceModel.pathDepthFirstTraversal();
	ArrayList targetTraversal = targetModel.pathDepthFirstTraversal();

	if (sourceTraversal==null || sourceTraversal.isEmpty() || targetTraversal==null ||
	targetTraversal.isEmpty())
		return null;

	int m = sourceTraversal.size();
	int n = targetTraversal.size();
	float[][] simMatrix = new float[m][n];
	float[][] simMatrixSemantic = new float[m][n];
	float[][] simMatrixSyntactic = new float[m][n];

	BufferedWriter out = null;

	Calendar cal = Calendar.getInstance();
	long sec1 = cal.getTimeInMillis();
	try{
		FileWriter fstream = new FileWriter("syntactic_matching.txt");
		out = new BufferedWriter(fstream);
		out.write("S Y N T A C T I C  M A T C H I N G: ");
		out.write(sourceModel.getName() + " VS." + targetModel.getName() + "\n\n");
		simMatrix = wordnetMatcherByManakan(sourceModel, targetModel, min, max,
		strategy_mod, approvedMapping);

		ArrayList rows = new ArrayList();
		for(int i=0; i < m; i++){
			ArrayList sourcePath = (ArrayList)sourceTraversal.get(i);
			Node sourceNode = (Node)sourcePath.get(sourcePath.size()-1);
			ModelNodeObject sourceObj = (ModelNodeObject)sourceNode.getObject();
			String source = sourceObj.getName();
			rows.add(source);
		}
		ArrayList columns = new ArrayList();
		for(int i=0; i < n; i++){
			ArrayList targetPath = (ArrayList)targetTraversal.get(i);
			Node targetNode = (Node)targetPath.get(targetPath.size()-1);
			ModelNodeObject targetObj = (ModelNodeObject)targetNode.getObject();
			String target = targetObj.getName();
			columns.add(target);
```

```
                }

                for(int i = 0; i < m; i++){
                        ArrayList sourcePath = (ArrayList)sourceTraversal.get(i);
                        Node sourceNode = (Node)sourcePath.get(sourcePath.size()-1);
                        ModelNodeObject sourceObj = (ModelNodeObject)sourceNode.getObject();
                        String source = sourceObj.getName();

                        ArrayList attr1 = new ArrayList();
                        ArrayList attr2 = new ArrayList();

                        //get source's attributes
                        Node[] children1 = sourceModel.getChildren(sourceNode);
                        if(children1 != null){
                                attr1 = new ArrayList();
                                for(int j = 0; j< children1.length; j++)
                attr1.add(((ModelNodeObject)children1[j].getObject()).getName());
                        }

                        //1st constraint: attr1.size() != 0
                        if(attr1.size() != 0){
                                //get target's attributes
                                for(int k=0; k < n; k++){
                                        ArrayList targetPath = (ArrayList)targetTraversal.get(k);
                                        Node targetNode = (Node)targetPath.get(targetPath.size() - 1);
                                        ModelNodeObject targetObj =
                                        (ModelNodeObject)targetNode.getObject();
                                        String target = targetObj.getName();
                                        Node[] children2 = targetModel.getChildren(targetNode);
                                        if(children2 != null){
                                                attr2 = new ArrayList();
                                                for(int l=0;l<children2.length;l++){
                attr2.add(((ModelNodeObject)children2[l].getObject()).getName());
                                                }
                                                //2nd constraint: attr2.size() != 0
                                                if(attr2.size() != 0){
                                                        int diff = (attr1.size()-attr2.size()>=0) ?
(attr1.size()-attr2.size()) : (attr2.size()-attr1.size());
                                                        //same size of attr1 & attr2
                                                        if(diff <= 5){
                                                        float[][] temp = new
                                                        float[attr1.size()][attr2.size()];
                                                        int commonAttrs = 0;
                                                        for(int a=0;a<attr1.size();a++){
                                                        boolean fl = false;
                                                        int b=0;
                                                        while(b < attr2.size() &&
                                                        fl == false){
```

```
temp[a][b] = simMatrix[rows.indexOf(attr1.get(a))][columns.indexOf(attr2.get(b))];
if(temp[a][b] > 0.8){
		commonAttrs++;
		fl = true;
}
b++;
}
fl = false;
}
float res = 2*((float)commonAttrs/(attr1.size()+attr2.size()));
if(res > 1.0)		res = (float)1.0;
		commonAttrs = 0;
		simMatrixSemantic[i][k] = res;
}
else
		simMatrixSemantic[i][k] = 0;
}
else
		simMatrixSemantic[i][k] = 0;
}
else
		simMatrixSemantic[i][k] = 0;
}//for
}//if(attr1.size() != 0)
else
for(int k=0; k < n; k++)	simMatrixSemantic[i][k] = 0;
}//for

ArrayList child1, child2;

for(int i=0; i < m; i++){
		ArrayList sourcePath = (ArrayList)sourceTraversal.get(i);
		Node sourceNode = (Node)sourcePath.get(sourcePath.size()-1);
		ModelNodeObject sourceObj = (ModelNodeObject)sourceNode.getObject();
		String source = sourceObj.getName();

		Node[] children1 = sourceModel.getChildren(sourceNode, Link.IS_A);
		if(children1 != null){
			child1 = new ArrayList();
			for(int k = 0; k< children1.length; k++){
		child1.add(((ModelNodeObject)children1[k].getObject()).getName());
			}
			for(int j=0; j < n; j++){
				ArrayList targetPath = (ArrayList)targetTraversal.get(j);
				Node targetNode = (Node)targetPath.get(targetPath.size()-1);
				ModelNodeObject targetObj = (ModelNodeObject)targetNode.getObject();
				String target = targetObj.getName();
```

```
                        //an den mporoume na bgaloume asfales sumperasma
                        if((simMatrix[i][j] > 0.1 || simMatrixSemantic[i][j] > 0.1) && (simMatrix[i][j] < 0.9
|| simMatrixSemantic[i][j] < 0.9)){
                                Node[] children2 = targetModel.getChildren(targetNode, Link.IS_A);

                                //kai an exei paidia o target node
                                if(children2 != null){
                                        child2 = new ArrayList();
                                        for(int k = 0; k < children2.length; k++){
                                child2.add(((ModelNodeObject)children2[k].getObject()).getName());
                                        }
                                        int diff = (child1.size()>child2.size()) ? child1.size()-child2.size() :
child2.size()-child1.size();

                                //an to plh8os twn paidiwn einai sugkrisimo
                                        if(diff <= 5){
                                                float[][] temp = new float[child1.size()][child2.size()];
                                                //float tmp;
                                                int commonChildren = 0;

                                                for(int a=0;a<child1.size();a++){
                                                        boolean fl = false;
                                                        int b=0;
/*************************************************************************/
        while((b < child2.size()) && (fl == false)){            //lexically einai or8o, semantically omws mporei
na brisketai se //diaforetikh 8esh tou pinaka
        //kai logw diaforetikhs perigrafhs h omoiothta na diaforopoieitai
                                                int r = children1[a].getId();
                                                int c = children2[b].getId();
out.write(child1.get(a) + " VS. " + child2.get(b) + " semantic similarity = " + simMatrixSemantic[r][c] +"\n");

/*
        out.write(children1[a].toString() + " VS. " + children2[b].toString() + " semantic similarity = "
+simMatrixSemantic[r][c] +"\n");
*/
temp[a][b] = (simMatrixSemantic[r][c] +
simMatrix[rows.indexOf(child1.get(a))][columns.indexOf(child2.get(b))]) / 2;
        if(temp[a][b] > 0.8){
                commonChildren++;
                fl = true;
        }
        b++;
}
fl = false;
}
float res = 2*((float)commonChildren/(child1.size()+child2.size()));
if(res > 1.0)       res = (float)1.0;
```

```
out.write(source + " VS. " + target + " -> " + res + "\n");
//out.write(" (dice coefficient): " + res + "\n");
commonChildren = 0;
simMatrixSyntactic[i][j] = res;
}//if(diff <= 5)
else
        simMatrixSyntactic[i][j] = 0;
}//if(children2 != null)
else
simMatrixSyntactic[i][j] = 0;
}//if: den mporoume na bgaloume asfales sumperasma
//                                                      simMatrixSyntactic[i][j] = 0;
simMatrixSyntactic[i][j] = (simMatrixSemantic[i][j] + simMatrix[i][j]) / 2;
}
}//if(children1 != null)
else
        for(int k=0; k < n; k++)    simMatrixSyntactic[i][k] = 0;
}//for

cal = Calendar.getInstance();
long sec2 = cal.getTimeInMillis();
long t = sec2-sec1;
out.write("\nStarted on: " + sec1 + "\nStopped on: " + sec2 + "\nTotal execution time: " + t +
"(millisecs)\n");
out.close();
}
catch(Exception e){
        System.out.println("E X C E P T I O N        I N        S Y N T A C T I C
        M A T C H I N G");
}//try-catch for writing in a file
return simMatrixSyntactic;
}
```

Γ. Κώδικας Στρατηγικών Επιλογής Συσχετίσεων

```
public static float[][] select_wordnet(float[][] simMatrix,String epilogi,double threshold,double diafora)
{
        String one="Maximum Similarity Matching";
    String two="Distance from Maximum Similarity";
    String three="Matchings above Similarity Threshold";
    String four="All Matchings";
    String five = "Maximum Attributes Similarity Matching";

        //prwth stratigikh -epilogh me bash orio-threshhold
        if(epilogi.equals(three)){
          //epilogh olwn twn upopshfiwn me omoiothta panw apo to orio poy dinei o xrhsths
          //me xrhsh threshold
          float[][] m1 = select(simMatrix, Combination.DIR_BOTH, SEL_SIMTHRESHOLD,0, (float)0.0,
(float)threshold);

          return m1;
        }
        //deuterh stratigikh- epilogh me bash to megisto kathe stoixeiou tou montelou
        else if(epilogi.equals(one)){
          System.out.println("Epilogh: " + epilogi);
          //mia sigourh politikh epiloghs einai -- epilogh megisths sysxetishs konmbou
          //pou einai h akolouthi me cand number 1 stragety SEL_SIMMAXN
          float[][] m2 = select(simMatrix, Combination.DIR_BOTH,SEL_SIMMAXN,1,(float)0.0,(float)0.1);
          return m2;}
        //triti stratigikh epiloges konta sto megisto kai to megisto me orio kontinothtas
        //poy orizei o xrhsths
        else if(epilogi.equals(two)){
          float[][] m2 = select(simMatrix, Combination.DIR_BOTH, SEL_SIMMAXDELTA,0,(float)diafora,
(float)0.0);
          return m2;
        }
        else if(epilogi.equals(five)){
          System.out.println("Epilogh: " + epilogi);
          System.out.println("***UNDER CONSTRUCTION***");
          return simMatrix;
        }
        else
         //teleutaia 4h stratigikh epistrefei ola ta apotelesmata oti brhke
         return simMatrix;
   }
```