



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΑΠΟΦΑΣΕΩΝ

**ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ
ΑΣΑΦΟΥΣ ΓΝΩΣΗΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΖΗΣΗ ΙΩΑΝΝΗ

Επιβλέπων : Ασκούνης Δημήτριος
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2011



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάπτυξη Συστήματος Διαχείρισης Ασαφούς Γνώσης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΖΗΣΗ ΙΩΑΝΝΗ

Επιβλέπων : Ασκούνης Δημήτριος
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20^η Ιουλίου 2011.

(Υπογραφή)

.....
Ασκούνης Δημήτριος
Επίκουρος Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Ψαρράς Ιωάννης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Μέντζας Γρηγόριος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2011

(Υπογραφή)

.....

ΖΗΣΗΣ ΙΩΑΝΝΗΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2011 – All rights reserved

Περίληψη

Η Διαχείριση Γνώσης αποσκοπεί στο να βοηθήσει τις εταιρείες και τους οργανισμούς να αυξήσουν την αποδοτικότητα των επιχειρηματικών λειτουργιών τους αλλά και να προσδώσουν σε αυτές καινοτόμα χαρακτηριστικά μέσω της αποτελεσματικής εκμετάλλευσης της γνώσης τους. Ως εκ τούτου, για μια σύγχρονη επιχείρηση είναι βασικό να αναπτύξει και να εφαρμόσει συστηματικές πρακτικές και μηχανισμούς για την διοίκηση και ανάπτυξη της γνώσης ώστε να εκμεταλλευτεί κατά το βέλτιστο δυνατό τα πλεονεκτήματα που αυτή προσφέρει.

Ένα φαινόμενο όμως που παρατηρείται συχνά στην επιχειρηματική γνώση, αλλά δεν αντιμετωπίζεται συνήθως, είναι αυτό της ασάφειας, της εγγενούς δηλαδή αοριστίας και ανακρίβειας με την οποία συνήθως εκφράζεται και αναπαρίσταται η γνώση της επιχείρησης. Η ασάφεια αυτή επηρεάζει αναμφισβήτητα τον τρόπο με τον οποίο χρησιμοποιείται η γνώση σε μια επιχείρηση και κατ' επέκταση υπάρχει η ανάγκη για μεθόδους και συστήματα διαχείρισης γνώσης που να μπορούν τη διαχειριστούν και να την εκμεταλλευτούν. Προς την κατεύθυνση αυτήν έχει προταθεί από την ερευνητική κοινότητα το πλαίσιο IKARUS-CBR, το οποίο συνδυάζει διάφορες τεχνικές από τους χώρους των Οντολογιών, της Βασισμένης σε Περιπτώσεις Συλλογιστικής και της Ασαφούς Λογικής στοχεύοντας στην αναπαράσταση και ανάκτηση ασαφούς επιχειρηματικής γνώσης.

Σκοπός της παρούσας διπλωματικής αποτελεί ο σχεδιασμός και η υλοποίηση μιας γραφικής διεπιφάνειας χρήστη η οποία θα παρέχει τη δυνατότητα στο χρήστη με απλό και εύχρηστο τρόπο να χρησιμοποιεί το πλαίσιο IKARUS-CBR για την υλοποίηση πρακτικών εφαρμογών διαχείρισης ασαφούς γνώσης. Η διεπιφάνεια αυτή σχεδιάστηκε ως εφαρμογή βασισμένη στο διαδίκτυο και επιτρέπει τη μοντελοποίηση ασαφούς γνώσης με τη μορφή οντολογιών καθώς και την παραμετροποίηση του μηχανισμού ανάκτησης γνώσης του πλαισίου IKARUS-CBR.

Λέξεις Κλειδιά: Διαχείριση Γνώσης, Ασάφεια, Οντολογίες, Συλλογιστική Βασισμένη σε Περιπτώσεις, Γραφική Διεπιφάνεια Χρήστη.

Abstract

Knowledge Management aims in helping companies and organizations not only to increase the efficiency of business operations, but also to offer them innovative features through exploiting knowledge with effective way. Therefore, it is essential for modern companies to develop and use systematic practices and mechanisms for the management and development of knowledge in order to gain the most of the advantages it offers.

But a phenomenon often observed in business knowledge, which is not usually solved, is that of fuzziness. Fuzziness refers to the inherent vagueness and imprecision that exists in the representation of business knowledge. This fuzziness certainly affects the way knowledge is manipulated in companies and therefore there is need for methods and knowledge managing systems that can manage and exploit it. Towards this direction has been proposed by the research community the IKARUS – CBR Framework, which combines various techniques from the fields of Ontologies, Case – Based Reasoning and Fuzzy Logics aiming at the representation and retrieval of vague business knowledge.

The aim of this diploma thesis is to design and implement a graphical user interface (GUI) which will provide user the ability to manipulate IKARUS – CBR with simple and practical way in order to cope with scenarios referring to managing fuzzy knowledge. This interface was designed as a web-based application and allows user to model fuzzy knowledge in the form of Fuzzy Ontologies and to parameterize the knowledge retrieval mechanism of IKARUS – CBR framework.

Keywords: Knowledge Management, Vagueness, Ontologies, Case Based Reasoning, Graphical User Interface.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Διαχείριση και Αναπαράσταση Γνώσης.....	1
1.2	Ασάφεια.....	3
1.3	Αντικείμενο Διπλωματικής.....	5
1.4	Δομή Διπλωματικής.....	6
2	Θεωρητικό Υπόβαθρο.....	7
2.1	Συλλογιστική Βασισμένη σε Περιπτώσεις(CBR).....	7
2.1.1	<i>Επίλυση προβλημάτων βασισμένα σε περιπτώσεις.....</i>	<i>7</i>
2.1.2	<i>Μοντέλο Αναπαράστασης CBR.....</i>	<i>9</i>
2.2	Οντολογίες.....	11
2.2.1	<i>Η Έννοια της Οντολογίας.....</i>	<i>11</i>
2.2.2	<i>Οντολογική Μηχανική.....</i>	<i>13</i>
2.2.3	<i>Αναπαράσταση Γνώσης και Γλώσσες Οντολογιών.....</i>	<i>13</i>
2.2.4	<i>Συστατικά Οντολογίας.....</i>	<i>14</i>
2.3	Ασαφής Λογική.....	15
2.3.1	<i>Ασαφή Σύνολα.....</i>	<i>15</i>
2.3.2	<i>Ιδιότητες Ασαφών Συνόλων.....</i>	<i>17</i>
2.3.3	<i>Ασαφείς Σχέσεις.....</i>	<i>18</i>
2.3.4	<i>Ασαφείς Αριθμοί και Γλωσσικές Μεταβλητές.....</i>	<i>18</i>
3	Υβριδικό Πλαίσιο IKARUS-CBR.....	21
3.1	Αναπαράσταση Ασαφούς Γνώσης.....	21
3.1.1	<i>Οντολογίες και Ασάφεια.....</i>	<i>21</i>
3.1.2	<i>Μοντέλο Αναπαράστασης Ασαφών Οντολογιών.....</i>	<i>23</i>
3.1.3	<i>Οντολογικό Μοντέλο Αναπαράστασης Ασαφών Περιπτώσεων.....</i>	<i>26</i>
3.2	Ανάκτηση Ασαφούς Γνώσης.....	27
3.2.1	<i>Ομοιότητα για Ασαφείς Ιδιότητες-Γνωρίσματα (FPA).....</i>	<i>27</i>
3.2.2	<i>Ομοιότητα για Ασαφώς Αποτιμώμενες Ιδιότητες-Γνωρίσματα (FVPA).....</i>	<i>28</i>

3.2.3	Ομοιότητα για Ασαφείς Σχέσεις-Γνωρίσματα (FRA).....	28
3.2.4	Σχήμα Ομοιότητας πλαισίου.....	30
4	Σχεδίαση Συστήματος	33
4.1	Ανάλυση Απαιτήσεων.....	33
4.1.1	Χρήστες	34
4.1.2	Περιπτώσεις Χρήσης	35
4.2	Αρχιτεκτονική.....	41
4.2.1	Πλευρά Model.....	44
4.2.2	Πλευρά View	44
4.2.3	Πλευρά Controller	51
5	Υλοποίηση	53
5.1	Τεχνικές λεπτομέρειες	53
5.1.1	Δημιουργία Όψεων.....	53
5.1.2	Δημιουργία Κλάσεων.....	56
5.2	Πλατφόρμες και προγραμματιστικά εργαλεία	61
5.2.1	ZK framework.....	61
5.2.2	Apache Tomcat 6	62
5.2.3	Subversion.....	63
5.2.4	StarUML.....	64
5.2.5	Πλατφόρμα ανάπτυξης	64
6	Σενάριο Χρήσης IKARUS-GUI.....	65
6.1	Περιγραφή Σεναρίου.....	65
6.1.1	Ορισμός Περιπτώσεων.....	66
6.1.2	Ορισμός Νοηματικού Πλαισίου Ανάκτησης.....	67
6.2	Ενέργειες στο GUI.....	68
6.2.1	Δημιουργία Ασαφούς Οντολογίας Αγοράς Ενέργειας.....	68
6.2.2	Δημιουργία Ασαφούς Οντολογίας Περιπτώσεων	71
6.2.3	Ορισμός Νοηματικού Πλαισίου Ανάκτησης.....	75
7	Επίλογος	86
7.1	Σύνοψη και συμπεράσματα.....	86

7.2	Μελλοντικές επεκτάσεις	87
8	Βιβλιογραφία.....	89

1

Εισαγωγή

Στο παρόν εισαγωγικό κεφάλαιο παρουσιάζονται ορισμένες βασικές έννοιες με τις οποίες θα ασχοληθεί η παρούσα διατριβή, όπως η διαχείριση γνώσης και η ασάφεια που ενδέχεται να περιέχει σε κάποια μορφή η γνώση. Στο τέλος του κεφαλαίου περιγράφονται ο σκοπός της διπλωματικής και η δομή που θα συναντήσει ο αναγνώστης στην παρούσα διατριβή.

1.1 Διαχείριση και Αναπαράσταση Γνώσης

Στη σημερινή εποχή η δημιουργία, η κατανομή και ο χειρισμός της πληροφορίας είναι μία από τις πιο σημαντικές οικονομικές δραστηριότητες των επιχειρήσεων τη στιγμή που παρέχεται άπλετα η δυνατότητα στους ανθρώπους να ανταλλάσσουν και να μεταφέρουν πληροφορίες ελεύθερα και να έχουν άμεση πρόσβαση σε πηγές γνώσης που θα ήταν δύσκολο ή αδύνατο να βρεθούν στο παρελθόν. Σε αυτό το βομβαρδισμένο από πληροφορίες περιβάλλον ο πλούτος και η απόκτηση στρατηγικού πλεονεκτήματος έναντι άλλων ανταγωνιστικών επιχειρήσεων παράγεται μέσα από την οικονομική εξερεύνηση, εκμετάλλευση και απόδοση του σημασιολογικού περιεχομένου της γνώσης.

Όλα αυτά τα παραπάνω στοιχεία μας οδηγούν στην Εποχή της Πληροφορίας και στην Κοινωνία της Πληροφορίας [8], η οποία διαφαίνεται ότι είναι ο διάδοχος της Βιομηχανικής Κοινωνίας. Στην Κοινωνία της Πληροφορίας, η πληροφορία αποτελεί κυριολεκτικά την

πρώτη ύλη και την κινητήρια δύναμη των επιστημονικών, οικονομικών, διοικητικών και άλλων κοινωνικών δραστηριοτήτων, το πιο σημαντικό οικονομικό αγαθό.

Όλη αυτή η παραγωγή και η έκρηξη των πληροφοριών ένεκα της επιστημονικής και τεχνολογικής έρευνας και ειδικά η ραγδαία και ταχύτατη εξάπλωσή τους μέσω του Διαδικτύου καταλήγει στον ανηλεή βομβαρδισμό των επιχειρήσεων από πληθώρα πληροφοριών. Βασική επιδίωξη της κάθε επιχείρησης είναι η συλλογή, ο χειρισμός και εν τέλει η αποτελεσματική εκμετάλλευση και ερμηνεία των πληροφοριών με απώτερο στόχο την παραγωγή νέας γνώσης. Κεντρική ιδέα επομένως της εποχής μας είναι η Διαχείριση Γνώσης.

Πώς καταλήγουμε όμως στην παραγωγή νέας γνώσης; Αρχικά, διακρίνουμε τα δεδομένα που είναι μη οργανωμένα και μη επεξεργασμένα γεγονότα σχετικά με αντικείμενα ή συμβάντα του πραγματικού κόσμου [3]. Συνήθως τα δεδομένα αποτελούν κάποιες μετρήσιμες ή υπολογίσιμες τιμές των ιδιοτήτων των αντικειμένων όπως η τιμή πώλησης ενός προϊόντος ή το ποσοστό της ανόδου του δείκτη του χρηματιστηρίου για κάποια συγκεκριμένη ημερομηνία. Από μόνα τους δεν αποτελούν πληροφορία, ενώ είναι στατικά, αφού από τη στιγμή της καταγραφής τους δεν μπορούν να αλλάξουν.

Στη συνέχεια, με κατάλληλη μορφοποίηση και αποτελεσματικό φιλτράρισμα των δεδομένων καταλήγουμε στην παραγωγή πληροφορίας, η οποία χρειάζεται ένα εννοιολογικό πεδίο που να επιτρέπει την ερμηνεία της. Η πληροφορία σε σχέση με τα δεδομένα διακρίνεται από νόημα, σκοπό και συνάφεια, ώστε να μπορεί να ερμηνεύεται και να διευκολύνει στη λήψη αποφάσεων. Για παράδειγμα, το ετήσιο άθροισμα των πωλήσεων ενός υποκαταστήματος κάποιας επιχείρησης αποτελεί τόσο επεξεργασμένη πληροφορία, λόγω της άθροισης, όσο και φιλτραρισμένη, λόγω του συγκεκριμένου υποκαταστήματος.

Για να φτάσουμε όμως στην παραγωγή γνώσης θα πρέπει η πληροφορία να υποστεί μια σειρά ειδικών ελέγχων για την πιστοποίησή της, όπως γίνεται για παράδειγμα στην επιστημονική γνώση ή στη γνώση που προέρχεται από τη μακρόχρονη επιβεβαίωση των καθημερινών εμπειριών. Γνώση θα αποτελούσε σε συνέχεια του προηγούμενου παραδείγματος η επαναλαμβανόμενη παρατήρηση ότι οι πωλήσεις από ένα συγκεκριμένο υποκατάστημα αυξάνονται 20% κατά τους θερινούς μήνες. Ουσιαστικά η γνώση αναδεικνύει τη σημαντικότητα της πληροφορίας συσχετίζοντάς τη με χρήσιμα συμπεράσματα ή αναγκαίες ενέργειες. Γενικότερα πάντως, γνώση θεωρείται η κατανόηση που αποκτάται μέσω εμπειρίας ή μελέτης και συμπεριλαμβάνει όλες τις πληροφορίες, τις εμπειρίες, τις ικανότητες, τις

δεξιότητες και την κοινή λογική που κατέχει ένας άνθρωπος ή ακόμα και μια επιχείρηση (με την έννοια ότι αποτελείται από ένα σύνολο ανθρώπων με κοινούς στόχους και οράματα).

Μετά την παραγωγή γνώσης οι επιχειρήσεις συγκεντρώνονται στην όσο κατά το δυνατόν βέλτιστη εκμετάλλευσή της. Για το σκοπό αυτό πολλοί ερευνητές έχουν προτείνει διοικητικά και τεχνολογικά πλαίσια διαχείρισης γνώσης (Knowledge Management Frameworks) που αφορούν τον τρόπο με τον οποίο η γνώση μιας επιχείρησης γίνεται αντικείμενο διαχείρισης. Τα πλαίσια αυτά στο σύνολό τους περιλαμβάνουν μεθόδους, τεχνικές και εργαλεία βάσει των οποίων η γνώση συλλέγεται, αποθηκεύεται, μεταδίδεται, οργανώνεται, εφαρμόζεται και εξελίσσεται εντός της επιχείρησης. Με τις τεχνικές αυτές ασχολείται ο χώρος της Τεχνητής Νοημοσύνης προτείνοντας ποικίλες προσεγγίσεις με γνώμονα πάντα τη Διαχείριση Γνώσης.

Με στόχο την οργάνωση και συντήρηση γνώσης, προτείνεται από τον τομέα της Τεχνητής Νοημοσύνης (TN) η Μοντελοποίηση και Αναπαράσταση Γνώσης. Πιο συγκεκριμένα, για τα συστήματα TN απαιτείται ένας μονοσήμαντος και τυποποιημένος συμβολισμός, ο οποίος εκτός της δυνατότητας που θα δίνει για ακριβή αναπαράσταση της γνώσης, θα πρέπει να μπορεί να συνδυαστεί κατάλληλα με μία ή περισσότερες συλλογιστικές για την εξαγωγή συμπερασμάτων. Για την αναπαράσταση γνώσης χρειάζεται ένα σύνολο συντακτικών και σημασιολογικών παραδοχών, οι οποίες καθιστούν δυνατή την περιγραφή ενός κόσμου. Συνεπώς, τα βασικά δομικά στοιχεία που έχει κάθε μέθοδος αναπαράστασης είναι:

- το συντακτικό (syntax), δηλαδή ο ορισμός των συμβόλων που χρησιμοποιεί και οι κανόνες με τους οποίους τα σύμβολα αυτά μπορεί να συνδυαστούν, και
- τη σημασιολογία (semantics), δηλαδή ο καθορισμός των εννοιών που αποδίδονται στα σύμβολα και συνδυασμούς συμβόλων που επιτρέπει το συντακτικό.

Μια μέθοδος αναπαράστασης γνώσης θα πρέπει να είναι ικανή να αναπαραστήσει τα αντικείμενα ενός κόσμου, της ευρύτερης κατηγορίας ή κλάσης αντικειμένων στην οποία ανήκουν, καθώς και τις σχέσεις μεταξύ αυτών των κλάσεων. Αυτό το είδος γνώσης αναφέρεται και ως σημασιολογική γνώση (semantic knowledge), η οποία συνήθως είναι ιεραρχικά δομημένη. Βασικό εργαλείο της TN που ικανοποιεί τις παραπάνω προδιαγραφές είναι οι Οντολογίες.

1.2 Ασάφεια

Αν μελετηθεί τώρα η επιχειρηματική γνώση, η οποία συλλέγεται αρχικά από τους ανθρώπους, θα διαπιστωθεί ότι μεταφέρει κάποια χαρακτηριστικά που τα προσδίδουν όχι

μόνο οι ίδιοι οι άνθρωποι αλλά και η φυσική γλώσσα που χρησιμοποιείται για τη μετάδοση γνώσης και πληροφορίας. Αξίζει ιδιαίτερη προσοχή ως προς αυτά τα χαρακτηριστικά η αβεβαιότητα με την έννοια της έλλειψης ακριβούς πληροφορίας, που οφείλεται κυρίως [3]:

- στα ανακριβή δεδομένα (imprecise data), που μπορεί να προέρχονται από ένα όργανο περιορισμένης ακρίβειας,
- στα ελλιπή δεδομένα (incomplete data), όπως για παράδειγμα όταν κάποια δυσλειτουργία ενός μηχανήματος το θέσει εκτός λειτουργίας για ορισμένο χρονικό διάστημα,
- στους οικονομικούς και χρονικούς περιορισμούς που κάνουν ασύμφορη και αδύνατη αντίστοιχα την πραγματοποίηση κάποιων μετρήσεων,
- στην υποκειμενικότητα που προσδίδουν οι άνθρωποι στις πληροφορίες που μεταφέρουν, καθώς η απόκτηση και αποθήκευση γνώσης στους ανθρώπους βασίζονται στην έννοια της αντίληψης, που είναι εγγενώς ανακριβής, καθώς εξαρτάται άμεσα από την περιορισμένη δυνατότητα των αισθητήριων οργάνων και του εγκεφάλου του ανθρώπου να αντιλαμβάνονται και να αποθηκεύουν πληροφορία σε μεγάλο βαθμό λεπτομέρειας.

Επίσης, σημαντικά χαρακτηριστικά της γνώσης είναι και η ανακρίβεια και αοριστία που προκαλείται από την πολυσημαντικότητα ή αμφισημία (ambiguity) της φυσικής γλώσσας, δηλαδή οι πολλαπλές ερμηνείες που επιδέχεται μια φράση, καθώς επίσης και από την ερμηνεία με βάση τα συμφραζόμενα (context-based interpretation).

Η ασάφεια (vagueness) στη γνώση όμως δε θα πρέπει να συγχέεται με την αβεβαιότητα, ανακρίβεια και αμφισημία. Έχει να κάνει ως σημασιολογικό φαινόμενο με εκείνα τα κατηγορήματα γνώσης που αφορούν οριακές καταστάσεις [1], δηλαδή καταστάσεις στις οποίες υπάρχει αδυναμία στο να αποφασιστεί εάν μια πρόταση ισχύει ή δεν ισχύει. Για παράδειγμα η φράση "ο Νίκος είναι οριακά ψηλός", η οποία δεν προσδιορίζει με ακρίβεια το ύψος του Νίκου, από τη μια πλευρά επιτρέπει να βγουν κάποια συμπεράσματα που εξαρτιούνται ωστόσο από την προσωπική αντίληψη του καθενός του όρου "ψηλός", από την άλλη όμως δε γίνεται να αποσαφηνιστεί το αν ο Νίκος είναι ψηλός ή όχι.

Έτσι, σε περιπτώσεις με οριακές καταστάσεις παρατηρείται ασάφεια γνώσης. Για παράδειγμα, στην περίπτωση του ύψους δεν υπάρχουν ακριβή σύνορα μεταξύ των ψηλών ανθρώπων και των υπολοίπων. Συμπερασματικά, υπάρχουν δύο τρόποι για το διαχωρισμό των ασαφών κατηγορημάτων από τα μη ασαφή: α) τα ασαφή κατηγορήματα διαθέτουν

οριακές καταστάσεις ενώ τα μη ασαφή όχι, β) τα ασαφή κατηγορήματα δε διαθέτουν ακριβή σύνορα, όπως φάνηκε στο προηγούμενο παράδειγμα για τους ψηλούς ανθρώπους.

Ενώ όμως για έναν άνθρωπο είναι σχετικά εύκολο να χειριστεί και να χρησιμοποιήσει αυτήν την ανακρίβεια και ασάφεια, για ένα σύστημα είναι αρκετά δύσκολο. Δύσκολος επίσης είναι, για ένα σύγχρονο σύστημα διαχείρισης γνώσης λόγω της ασάφειας, ο χειρισμός της έννοιας της σχετικότητας (relevance), η οποία κατέχει κεντρικό ρόλο στα συστήματα αυτά καθώς σχετίζεται άμεσα με τη λειτουργία της ανάκτησης της γνώσης. Πιο συγκεκριμένα, η έννοια αυτή αφενός είναι ασαφής από μόνη της (καθώς το πόσο σχετικό είναι κάτι με κάτι άλλο είναι πάντα θέμα βαθμού) και αφετέρου γίνεται περισσότερο ασαφής όταν ο υπολογισμός της βασίζεται σε ασαφή ανθρώπινη γνώση. Σε ένα τρίτο επίπεδο, η ασάφεια της σχετικότητας προκύπτει και από την υποκειμενική αντίληψη του ανθρώπου για το τι είναι σχετικό και τι όχι.

1.3 Αντικείμενο Διπλωματικής

Σαν άμεσο συμπέρασμα των παραπάνω προκρίνεται η θέση ότι η γνώση που κατέχει μια επιχείρηση είναι σε μεγάλο βαθμό ασαφής. Έτσι, γίνεται ξεκάθαρη η ανάγκη για ολοκληρωμένη και αποτελεσματική διαχείριση αυτής της ασαφούς επιχειρηματικής γνώσης. Προτείνεται έτσι στο [2] ένα καινοτόμο υβριδικό πλαίσιο διαχείρισης γνώσης το οποίο καλύπτει την ανάγκη αυτή τόσο σε τεχνολογικό όσο και σε μεθοδολογικό επίπεδο. Το πλαίσιο αυτό βασίζεται στο συνδυασμό τριών προσεγγίσεων της TN, της Συλλογιστικής Βασισμένης σε Περιπτώσεις (Case Based Reasoning - CBR) της Οντολογικής Μηχανικής και της Ασαφούς Λογικής [2].

Πιο συγκεκριμένα ένα CBR σύστημα προσφέρει πολλαπλά πλεονεκτήματα σε ό,τι αφορά τα ζητήματα μοντελοποίησης και ανάκτησης της γνώσης. Σε συνδυασμό με τις οντολογίες που αποτελούν τον κύριο τρόπο μοντελοποίησης υπολογιστικά ερμηνεύσιμες (machine interpretable), κοινά αποδεκτές (shared) και επαναχρησιμοποιήσιμες (reusable) γνώσης, επιτρέπεται σε ένα CBR σύστημα να χρησιμοποιεί γνώση που έχει ήδη αναπτυχθεί μειώνοντας έτσι την ανάγκη για εξοχής συλλογή και μοντελοποίηση γνώσης. Ενσωματώνοντας στον παραπάνω συνδυασμό την Ασαφή Λογική με στόχο την όσο το δυνατόν πιο αποτελεσματική αντιμετώπιση της ασάφειας, προκύπτει το καινοτόμο υβριδικό πλαίσιο ασαφούς διαχείρισης γνώσης IKARUS-CBR (Imprecise Knowledge Acquisition Representation and Use).

Στο σημείο αυτό εισέρχεται η παρούσα διατριβή η οποία έχει ως στόχο το σχεδιασμό και την ανάπτυξη μιας γραφικής διεπιφάνειας-διεπαφής χρήστη (Graphical User Interface - GUI) που θα χειρίζεται το πλαίσιο IKARUS-CBR. Η σχεδίαση αυτή θα βασιστεί στη δομή και στις μεθόδους που χρησιμοποιεί το πλαίσιο IKARUS-CBR για να αναπαραστήσει και να ανακτήσει ασαφή γνώση. Η διεπαφή αυτή δηλαδή θα αποτελεί το τελευταίο επίπεδο του πλαισίου IKARUS-CBR που θα αναλαμβάνει την επικοινωνία του πλαισίου με το χρήστη.

Απώτερος σκοπός της γραφικής διεπιφάνειας θα είναι η αναπαράσταση και η ανάκτηση ασαφούς γνώσης σε ένα όσο το δυνατόν πιο φιλικό ως προς το χρήστη περιβάλλον.

1.4 Δομή Διπλωματικής

Η δομή που ακολουθείται στη διατριβή είναι η εξής:

- Στο Κεφάλαιο 1 περιγράφονται γενικές έννοιες που αφορούν τη διαχείριση και αναπαράσταση γνώσης, ενώ αναφέρονται ο σκοπός και η δομή της συγκεκριμένης διπλωματικής διατριβής.
- Στο Κεφάλαιο 2 αναλύεται το θεωρητικό υπόβαθρο της εργασίας η κατανόηση του οποίου κρίνεται αναγκαία από τον αναγνώστη πριν αναλυθούν η σχεδίαση και η υλοποίηση του συστήματος.
- Το Κεφάλαιο 3 περιγράφει τους τρόπους με τους οποίους η ασαφής γνώση αναπαρίσταται και ανακτάται από το σύστημα IKARUS-CBR. Η ύπαρξη του κεφαλαίου αυτού είναι καθοριστική αφού θα βοηθήσει τον αναγνώστη να κατανοήσει το σχεδιασμό της γραφικής διεπιφάνειας χρήστη που θα παρουσιασθεί στο Κεφάλαιο 4. Στο 4^ο Κεφάλαιο λοιπόν περιγράφεται η αρχιτεκτονική του συστήματος και οι βασικές λειτουργίες που αυτό επιτελεί.
- Το Κεφάλαιο 5 αναφέρει λεπτομέρειες που αφορούν την υλοποίηση της διεπιφάνειας χρήστη, ενώ περιγράφονται οι πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την περάτωση του συστήματος.
- Στο 6^ο Κεφάλαιο ελέγχεται η ορθή λειτουργία της διεπιφάνειας χρήστη με τη βοήθεια ενός σεναρίου χρήσης που θα οδηγήσει στην αξιολόγηση του συστήματος.
- Στο Κεφάλαιο 7 συνοψίζονται τα αποτελέσματα και η συμβολή της διπλωματικής και τέλος στο 8^ο Κεφάλαιο αναφέρεται η βιβλιογραφία στην οποία στηρίχθηκε η υλοποίηση της διπλωματικής.

2

Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται κάποιες βασικές έννοιες, η κατανόηση των οποίων κρίθηκε απαραίτητη για τη μελέτη των κεφαλαίων που ακολουθούν. Αναλύονται οι βασικές έννοιες της συλλογιστικής βασισμένης σε περιπτώσεις, της οντολογίας, ενώ γίνεται και αναφορά σε έννοιες που αφορούν την ασαφή λογική.

2.1 Συλλογιστική Βασισμένη σε Περιπτώσεις(CBR)

2.1.1 Επίλυση προβλημάτων βασισμένα σε περιπτώσεις

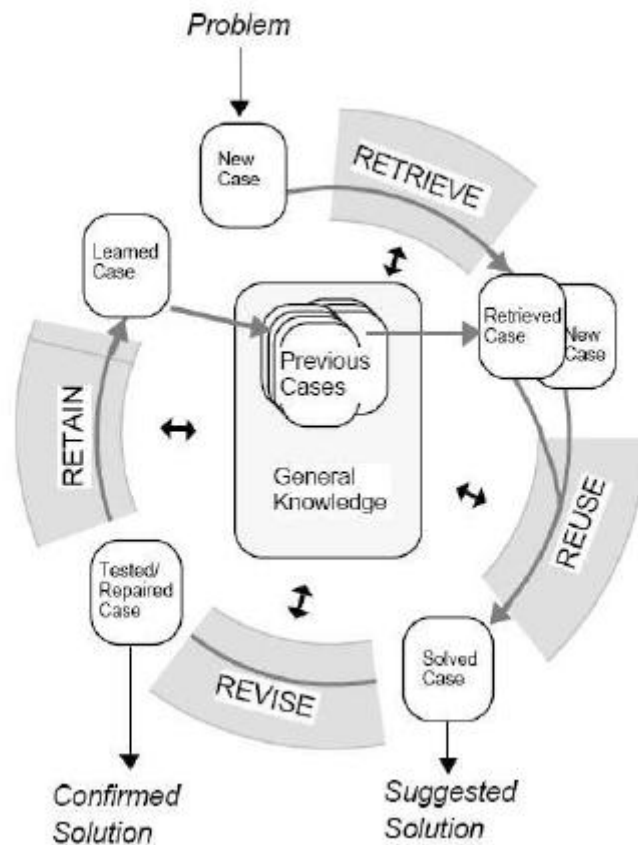
Η Συλλογιστική Βασισμένη σε Περιπτώσεις (Case-Based Reasoning-CBR) ως προσέγγιση της ΤΝ στηρίζεται στις μεθόδους που χρησιμοποιεί η ανθρώπινη νόηση με στόχο την επίλυση προβλημάτων. Σύμφωνα με τη θεωρία του Schank [4] η μάθηση και η υπενθύμιση στους ανθρώπους βασίζεται στη διατήρηση της εμπειρίας σε ένα δυναμικό και εξελισσόμενο σύστημα μνήμης. Κατά τον Anderson [5] οι άνθρωποι χρησιμοποιούν τις παλιές περιπτώσεις ως μοντέλα όταν μαθαίνουν να επιλύουν προβλήματα. Οι άνθρωποι πάντως όταν επιδιώκουν την επίλυση ενός προβλήματος ανακαλούν στη μνήμη τους μια παρόμοια παλαιότερη κατάσταση που έχουν ήδη αντιμετωπίσει από την οποία επαναχρησιμοποιούν την ήδη αποκτώμενη γνώση, βάσει της λογικής ότι παρόμοια προβλήματα θα έχουν και παρόμοιες λύσεις [6]. Χαρακτηριστικά είναι τα παρακάτω παραδείγματα τρόπου επίλυσης προβλημάτων παρμένα από την καθημερινότητα:

- Ένας γιατρός εξετάζει έναν ασθενή με συγκεκριμένο συνδυασμό συμπτωμάτων. Αν έχει ήδη αντιμετωπίσει κάποιον ασθενή με παρόμοια συμπτώματα στο παρελθόν, τότε θα ανακαλέσει στη μνήμη του τη συγκεκριμένη περίπτωση και θα καταλήξει έτσι στην παλιά διάγνωση ως προτεινόμενη λύση για τον υπό εξέταση ασθενή. Ο γιατρός βέβαια δε θεωρεί με απόλυτη βεβαιότητα ότι η παλιά διάγνωση θα είναι και η σωστή για τη νέα περίπτωση, αλλά θα την αξιολογήσει εκ νέου χωρίς να αποκλείεται πάντα να καταλήξει σε διαφορετικές πιθανές διαγνώσεις. Μολαταύτα ο γιατρός με το να ανακαλέσει μια παρόμοια παλιά περίπτωση θα παράγει εύκολα και γρήγορα μια εύλογη πιθανή λύση.
- Ένας μηχανικός αυτοκινήτων στην προσπάθειά του να επιδιορθώσει έναν κινητήρα είναι πιθανό να θυμηθεί παρόμοια επιλυμένα προβλήματα και έτσι να θεωρήσει ως λύση στο παρόν ζήτημα την παλιά λύση.
- Ένας δικηγόρος κατά την υπεράσπιση ενός πελάτη του στηρίζεται πάντα σε νομικά παρόμοια δεδικασμένα προκειμένου να καταλήξει σε συγκεκριμένη γραμμή υπεράσπισης.

Στην ορολογία CBR μια περίπτωση (case) συνίσταται σε ένα πρόβλημα μαζί με τη λύση του, ενώ το σύνολο των αποθηκευμένων περιπτώσεων ονομάζεται βάση περιπτώσεων (case base). Έτσι σε ένα CBR σύστημα η Συλλογιστική Βασισμένη σε Περιπτώσεις υλοποιείται μέσω μιας συστηματικής διαδικασίας που περιλαμβάνει την οργάνωση, αποθήκευση, ανάκτηση και προσαρμογή περιπτώσεων [6]. Πρόκειται δηλαδή για μια κυκλική και ολοκληρωμένη διαδικασία επίλυσης ενός προβλήματος, μάθησης από τη συγκεκριμένη εμπειρία, επίλυσης ενός νέου προβλήματος, κτλ.

Οι επιμέρους δραστηριότητες της συγκεκριμένης διαδικασίας φαίνονται στο σχήμα 2.1. και είναι οι εξής:

1. Ανάκτησε (RETRIEVE) την πιο όμοια περίπτωση από τη βάση περιπτώσεων.
2. Επαναχρησιμοποίησε (REUSE) την πληροφορία και γνώση από την ανακτημένη περίπτωση για να λύσεις τη νέα περίπτωση.
3. Αξιολόγησε (REVISE) την προτεινόμενη λύση.
4. Διατήρησε (RETAIN) το νέο πρόβλημα μαζί με τη λύση του αποθηκεύοντάς το ως νέα περίπτωση στη βάση περιπτώσεων για μελλοντική χρήση.



Σχήμα 2.1: Ο CBR κύκλος

Πιο αναλυτικά μια αρχική περιγραφή της περίπτωσης ορίζει την καινούρια περίπτωση (new case), η οποία χρησιμοποιείται για την ανάκτηση (RETRIEVE) μιας περίπτωσης από τη συλλογή των προηγούμενων αποθηκευμένων περιπτώσεων. Η περίπτωση που ανακτήθηκε συνδυάζεται με την καινούρια μέσω της επαναχρησιμοποίησης (REUSE) με αποτέλεσμα μια επιλυμένη περίπτωση, δηλαδή μια προτεινόμενη λύση στο αρχικό πρόβλημα. Μέσω της διαδικασίας της αξιολόγησης (REVISE) η προτεινόμενη λύση ελέγχεται για το αν πέτυχε στην επίλυση του νέου προβλήματος, είτε εφαρμοζόμενη υπό κανονικές συνθήκες ή αξιολογούμενη από έναν ειδικό, ενώ αν απέτυχε επιδιορθώνεται. Κατά την διατήρηση (RETAIN) τώρα χρήσιμη εμπειρία αποθηκεύεται για μελλοντική επαναχρησιμοποίηση και έτσι η βάση περιπτώσεων ανανεώνεται είτε από μια νέα επιλυμένη περίπτωση (learned case) ή από τροποποιήσεις σε ήδη υπάρχουσες περιπτώσεις.

2.1.2 Μοντέλο Αναπαράστασης CBR

Οι βασικές δυνατότητες που πρέπει να προσφέρει ένα CBR σύστημα περιλαμβάνουν:

- Τη δυνατότητα μοντελοποίησης των περιπτώσεων για ένα δεδομένο πεδίο εφαρμογής μέσω κατάλληλων μοντέλων αναπαράστασης.
- Τη δυνατότητα ανάκτησης παρόμοιων περιπτώσεων από τη βάση μέσω κατάλληλων μέτρων σύγκρισης.

Οι δύο αυτές δυνατότητες είναι αλληλένδετες υπό την έννοια ότι ο μηχανισμός σύγκρισης εξαρτάται από το μοντέλο αναπαράστασης. Έτσι, ένας από τους βασικότερους παράγοντες διαφοροποίησης των διάφορων CBR προσεγγίσεων που έχουν προταθεί από την ερευνητική κοινότητα και έχουν υλοποιηθεί σε πρακτικό επίπεδο είναι ο ακριβής τρόπος υλοποίησης του συνδυασμού αναπαράστασης-ανάκτησης των περιπτώσεων.

Οι θεμελιώδεις CBR προσεγγίσεις είναι οι εξής:

- **Προσέγγιση κειμένου [8]:** Οι περιπτώσεις αναπαριστώνται ως ελεύθερο κείμενο ενώ η ανάκτησή τους πραγματοποιείται με χρήση γλωσσολογικών τεχνικών αντιστοίχισης λέξεων και όρων ή και άλλων στατιστικών μέτρων ομοιότητας κειμένων. Για τη βελτίωση της ανάκτησης συνήθως χρησιμοποιούνται λεξικά όρων που καταγράφουν την τυχόν ομοιότητα μεταξύ διαφορετικών όρων (π.χ. συνώνυμους όρους). Γενικά όμως η ακρίβεια του μηχανισμού ανάκτησης μειώνεται σημαντικά με την αύξηση του αριθμού των περιπτώσεων.
- **Προσέγγιση διαλόγου [9]:** Οι περιπτώσεις θεωρούνται ως σύνολα ερωταπαντήσεων που έχουν προκύψει από κάποιο διάλογο και που εν γένει περιέχουν δυνητικά χρήσιμη πληροφορία και γνώση για την επίλυση ενός προβλήματος. Συνήθως υιοθετείται το δενδρικό μοντέλο δόμησης των ερωταπαντήσεων έτσι ώστε οι περιπτώσεις να έχουν τη μορφή ενός δένδρου αποφάσεων και η ανάκτηση να γίνεται μέσω διαλόγων του συστήματος με το χρήστη.
- **Δομημένη προσέγγιση [7]:** Η δομή των περιπτώσεων καθορίζεται από ένα μοντέλο πεδίου (domain model) το οποίο προδιαγράφει τα χαρακτηριστικά (features ή attributes) βάσει των οποίων περιγράφεται μια περίπτωση καθώς και το σύνολο των τιμών που αυτά μπορούν να λάβουν. Το συγκεκριμένο μοντέλο μπορεί να είναι τόσο απλό όσο ένας πίνακας ή τόσο περίπλοκο όσο ένα αντικειμενοστραφές μοντέλο ή μια οντολογία. Σε κάθε περίπτωση η λειτουργία της ανάκτησης λειτουργεί ως εξής: Το πρόβλημα του χρήστη αναπαρίσταται ως περίπτωση βάσει του μοντέλου πεδίου και στη συνέχεια συγκρίνεται με όλες τις περιπτώσεις που είναι αποθηκευμένες στη βάση. Έπειτα, οι αποθηκευμένες περιπτώσεις με τη μεγαλύτερη ομοιότητα ανακτώνται και επιστρέφονται στο χρήστη. Η διαδικασία σύγκρισης μεταξύ μιας

νέας περίπτωσης και μιας αποθηκευμένης γίνεται με χρήση μέτρων ομοιότητας τα οποία λειτουργούν ως εξής:

1. Υπολογίζουν αρχικά τις επιμέρους ομοιότητες των δύο περιπτώσεων συγκρίνοντας τις τιμές των επιμέρους χαρακτηριστικών τους.
2. Συναθροίζουν (aggregate) κατόπιν τις επιμέρους ομοιότητες σε μια τιμή η οποία υποδηλώνει το συνολικό βαθμό ομοιότητας των δύο περιπτώσεων.

2.2 Οντολογίες

2.2.1 Η Έννοια της Οντολογίας

Ο όρος Οντολογία προέρχεται από τον τομέα της φιλοσοφίας που μελετά το *ον*, την ύπαρξή του και την πραγματικότητα γενικά καθώς και τις βασικές κατηγορίες αντικειμένων που υπάρχουν και τις σχέσεις που αυτά εμφανίζουν μεταξύ τους. Παραδοσιακά οι οντολογίες αποτελούν αντικείμενο μεγάλου κλάδου της φιλοσοφίας γνωστό ως μεταφυσική, ενώ ασχολούνται με ερωτήσεις που αφορούν τους τύπους οντοτήτων που υπάρχουν ή μπορούν να ειπωθούν ότι υπάρχουν και πως οι οντότητες μπορούν να ομαδοποιηθούν, να συσχετισθούν μέσω μιας ιεραρχίας και υποκατηγοριοποιηθούν σύμφωνα με ομοιότητες και διάφορες. Ιστορικά τέτοιου τύπου ερωτήσεις τέθηκαν για πρώτη φορά από αρχαίους Έλληνες φιλοσόφους όπως ο Αριστοτέλης.

Στα τέλη του εικοστού και στις αρχές του εικοστού πρώτου αιώνα, οι οντολογίες εμφανίστηκαν ως αντικείμενο έρευνας και στην Πληροφορική και ιδιαίτερα στην περιοχή της Μηχανικής Γνώσης (Knowledge Engineering) ορίζοντας ένα σύνολο από αναπαραστατικά αρχέτυπα τα οποία μοντελοποιούν ένα πεδίο γνώσης. Τα αρχέτυπα αυτά τυπικά είναι κλάσεις ή σύνολα, ιδιότητες ή χαρακτηριστικά και σχέσεις που συνδέουν τις κλάσεις μεταξύ τους. Οι ορισμοί των παραπάνω συστατικών περιλαμβάνουν πληροφορία για τη σημασία που τους επιδέχεται και περιορισμούς με τη μορφή αξιωμάτων και κανόνων.

Ο διασημότερος ίσως ορισμός της οντολογίας έχει δοθεί από τον Gruber το 1993 [10]. Σύμφωνα με αυτόν μια οντολογία είναι «ένας τυπικός, ρητός προσδιορισμός μιας κοινής αντίληψης». Ο όρος αντίληψη υποδηλώνει ένα αφηρημένο μοντέλο το οποίο περιλαμβάνει έννοιες και σχέσεις που περιγράφουν κάποια αφηρημένη και απλοποιημένη όψη του κόσμου που για κάποιο λόγο θέλουμε να αναπαραστήσουμε. Η όψη αυτή ονομάζεται συνήθως γνωστικό πεδίο (domain) και αυτό που τελικά κάνει μια οντολογία είναι να ορίσει (και

ενδεχομένως να περιορίσει) το νόημα των εννοιών που περιγράφουν το πεδίο αυτό με τρόπο "τυπικό", "ρητό" και "κοινό".

Οι τρεις αυτοί επιθετικοί προσδιορισμοί είναι πολύ σημαντικοί γιατί περιγράφουν τις βασικές ιδιότητες που πρέπει να έχει μια οντολογία. Αναλυτικότερα, μια οντολογία είναι ρητή όταν το νόημα των εννοιών της ορίζεται με τρόπο ξεκάθαρο, συγκεκριμένο και μονοσήμαντο. Είναι επίσης κοινή όταν οι ορισμοί αυτοί είναι ομόφωνοι, έχουν δηλαδή συμφωνηθεί και αποδεχτεί από όλα τα μέλη της κοινότητας εντός της οποίας χρησιμοποιείται η οντολογία. Τέλος, μια οντολογία είναι τυπική όταν αναπαρίσταται με τρόπο μηχανικά ερμηνεύσιμο, με τέτοιο τρόπο δηλαδή ώστε το νόημά της να μπορεί να κατανοηθεί από υπολογιστικά συστήματα.

Ένας άλλος ορισμός που λειτουργεί συμπληρωματικά ως προς τον ορισμό του Gruber είναι αυτός του Neches [11] σύμφωνα με τον οποίο «μια οντολογία ορίζει τους βασικούς όρους και τις σχέσεις που απαρτίζουν ένα λεξιλόγιο μιας συγκεκριμένης θεματικής περιοχής καθώς και τους κανόνες βάσει των οποίων οι όροι και οι σχέσεις αυτές συνδυάζονται προκειμένου να εμπλουτίσουν το συγκεκριμένο λεξιλόγιο». Ο ορισμός αυτός, αν και λιγότερο επίσημος, αναδεικνύει μια άλλη σημαντική ιδιότητα των οντολογιών: το γεγονός ότι εκτός από έννοιες και σχέσεις μια οντολογία μπορεί να περιέχει και άλλα συστατικά στοιχεία (για παράδειγμα κανόνες ή αξιώματα) τα οποία επιτρέπουν την αυτόματη παραγωγή νέας γνώσης για ένα δεδομένο γνωστικό πεδίο βάσει της υπάρχουσας γνώσης. Η διαδικασία παραγωγής τέτοιας γνώσης ονομάζεται συμπερασμός (inference) ή συλλογισμός (reasoning) και μπορεί να λάβει διάφορες μορφές ανάλογα με τον τρόπο δόμησης και αναπαράστασης της οντολογίας.

Ένα ακόμη στοιχείο που χαρακτηρίζει τις οντολογίες δεν είναι τόσο το λεξιλόγιο μιας γλώσσας που χρησιμοποιείται για την περιγραφή των εννοιών και σχέσεων όσο είναι η εννοιολογική σύλληψη που αντιπροσωπεύουν οι όροι του λεξιλογίου που χρησιμοποιούνται. Έτσι, μεταφράζοντας τους όρους οντολογίας από μία γλώσσα σε μια άλλη, έστω από τα Αγγλικά στα Ελληνικά, δεν αλλάζει καθόλου η εννοιολογική μορφή της οντολογίας, εφόσον βέβαια ικανοποιηθεί ο δεύτερος επιθετικός προσδιορισμός μιας οντολογίας, δηλαδή "κοινή", και επομένως δεν τροποποιηθεί η σημασιολογία των όρων κατά τη μετάφραση.

Από τα παραπάνω προκύπτει εύλογα το συμπέρασμα ότι μια οντολογία είναι σε θέση να εξυπηρετήσει δύο σκοπούς:

1. Να επιτρέψει το διαμοιρασμό και την επαναχρησιμοποίηση της γνώσης που αφορά ένα συγκεκριμένο γνωστικό πεδίο μεταξύ ανθρώπων αλλά και υπολογιστικών συστημάτων.
2. Να επιτρέψει σε υπολογιστικά συστήματα να χρησιμοποιήσουν και να επεκτείνουν τη γνώση τους με τρόπο αρκετά παρόμοιο με την ανθρώπινη νόηση, με τη χρήση δηλαδή του συμπερασμού και της συλλογιστικής.

Τα επιστημονικά πεδία που εκμεταλλεύονται τα χαρακτηριστικά των οντολογιών είναι η Τεχνητή Νοημοσύνη, ο Σημασιολογικός Ιστός, η Τεχνολογία Λογισμικού και η Μηχανική Γνώσης.

2.2.2 Οντολογική Μηχανική

Το γενικό μεθοδολογικό πλαίσιο που ορίζει τη διαδικασία ανάπτυξης οντολογιών περιγράφεται από τις εξής κύριες δραστηριότητες:

- **Διαχείριση Ανάπτυξης (Management):** Περιλαμβάνει εργασίες με κύριο στόχο τη σωστή, αποτελεσματική και ποιοτική ανάπτυξη της οντολογίας. Κατά τη φάση αυτή λοιπόν διενεργούνται ο Προγραμματισμός των Εργασιών, ο Έλεγχος και η Διασφάλιση της Ποιότητας τους.
- **Ανάπτυξη (Development):** Στη συγκεκριμένη κατηγορία εμπίπτουν δραστηριότητες που εκτελούνται κατά την ανάπτυξη της οντολογίας ξεκινώντας από τη σύλληψη της ιδέας μέχρι την παράδοση του τελικού προϊόντος. Οπότε κατά τη φάση αυτή οι κύριες εργασίες είναι: Μελέτη Περιβάλλοντος και Σκοπιμότητας, Ανάλυση Προδιαγραφών (Specification), Εννοιολογική Μοντελοποίηση (Conceptualization), Τυποποίηση και Κωδικοποίηση (Formalization and Implementation), Συντήρηση (Maintenance).
- **Υποστήριξη (Support):** Οι υποστηρικτικές δραστηριότητες λαμβάνουν χώρα παράλληλα με τις δραστηριότητες ανάπτυξης και είναι απαραίτητες προκειμένου η εκτέλεση των τελευταίων να είναι δυνατή και επιτυχής. Έτσι, συντελούνται εργασίες Απόκτησης Γνώσης, Αξιολόγησης και Τεκμηρίωσης της διαδικασίας ανάπτυξης, Συγχώνευσης και Ολοκλήρωσης οντολογιών.

2.2.3 Αναπαράσταση Γνώσης και Γλώσσες Οντολογιών

Κατά τη διαδικασία της Εννοιολογικής Μοντελοποίησης συντελείται η δόμηση της γνώσης του γνωστικού πεδίου της οντολογίας με βάση ένα κοινά κατανοητό και αποδεκτό εννοιολογικό μοντέλο το οποίο αναπαριστά ξεκάθαρα τη σημασιολογία του πεδίου. Με τις

διαδικασίες της Τυποποίησης και Κωδικοποίησης επιτυγχάνεται η μετατροπή του παραπάνω εννοιολογικού μοντέλου σε ένα υπολογιστικά ερμηνεύσιμο μοντέλο αναπαράστασης γνώσης.

Όλες οι παραπάνω διαδικασίες πραγματοποιούνται κατά κανόνα μέσω κάποιας γλώσσας οντολογιών. Μια τέτοια γλώσσα ορίζει ένα σύνολο δομών έκφρασης νοήματος οι οποίες επιτρέπουν την αναπαράσταση γνώσης με τρόπο σαφή, τυποποιημένο και υπολογιστικά ερμηνεύσιμο. Οι διαφορετικές γλώσσες οντολογιών διακρίνονται με βάση την εκφραστικότητα που προσφέρουν, τη σημασιολογία και την πολυπλοκότητα εξαγωγής συμπερασμάτων. Η εκφραστικότητα αφορά την πληθώρα των δομών που προσφέρουν για την περιγραφή με τυπικό, ρητό, ακριβή και ευέλικτο τρόπο των συστατικών της οντολογίας. Η σημασιολογία αναφέρεται στην ερώτηση εάν είναι σαφές το τι σημαίνουν οι επιμέρους γλωσσικές δομές, ενώ η πολυπλοκότητα εξαγωγής συμπερασμάτων περιγράφει το μέγεθος των υπολογιστικών πόρων που απαιτούνται για να υπολογιστούν οι εκφράσεις στη δεδομένη γλώσσα αναπαράστασης γνώσης και αναπόφευκτα συσχετίζεται με την εκφραστικότητα της γλώσσας που θα χρησιμοποιηθεί.

Γενικά οι γλώσσες οντολογιών χωρίζονται σε αυτές που προέρχονται από την Τεχνητή Νοημοσύνη και σε αυτές από τις τεχνολογίες ιστού. Στην πρώτη κατηγορία ανήκουν γλώσσες που βασίζονται σε κάποιο λογικό φορμαλισμό (π.χ. λογική πρώτης τάξης, περιγραφικές λογικές, κλπ.), όπως οι γλώσσες KIF, CycL, Ontolingua και FLogic. Το συντακτικό των γλωσσών από την τεχνολογία ιστού βασίζεται σε γλώσσες όπως η HTML και η XML που αναπτύχθηκαν αρχικά με σκοπό την αναπαράσταση και ανταλλαγή δεδομένων. Κάποιες από αυτές είναι οι SHOE, RDF, RDF Schema και η OWL.

2.2.4 Συστατικά Οντολογίας

Μια οντολογία αναπαριστά τα εξής στοιχεία:

- **Έννοιες, Κλάσεις (Concepts, Classes):** Πρόκειται για αφηρημένα σύνολα, συλλογές ή τύποι αντικειμένων. Τα αντικείμενα δηλαδή του πεδίου εφαρμογής της οντολογίας κατηγοριοποιούνται σε ομάδες με κοινά χαρακτηριστικά και ιδιότητες. Για παράδειγμα η έννοια Άνθρωπος και Αυτοκίνητο που περιλαμβάνουν το σύνολο των ανθρώπων και των αυτοκινήτων αντίστοιχα.
- **Στιγμιότυπα (Instances):** Είναι τα βασικά αντικείμενα της οντολογίας, ενώ αποτελούν στιγμιότυπα των εννοιών ή κλάσεων. Για παράδειγμα, η έννοια Άνθρωπος έχει στιγμιότυπα τα αντικείμενα Γιάννης, Κώστας και Μαρία.

- **Ιδιότητες (Attributes):** Τα αντικείμενα μιας οντολογίας μπορούν να διαθέτουν χαρακτηριστικά γνωρίσματα, γενικά χαρακτηριστικά ή παραμέτρους. Συνδέονται έτσι τα αντικείμενα με τιμές που ανήκουν σε συγκεκριμένο πεδίο τιμών. Για παράδειγμα, η έννοια Άνθρωπος διαθέτει την ιδιότητα έχειΧρονιάΓέννησης που λαμβάνει τιμές από το πεδίο των ακεραίων αριθμών και έτσι το στιγμιότυπο Γιάννης στην ιδιότητα αυτή μπορεί να πάρει την τιμή 1987.
- **Σχέσεις (Relations):** Προσδιορίζουν τον τρόπο με τον οποίο συνδέονται και συσχετίζονται τα αντικείμενα μεταξύ τους. Για παράδειγμα, η έννοια Άνθρωπος διαθέτει τη σχέση έχειΑυτοκίνητο που συνδέει αντικείμενα που ανήκουν στην έννοια Άνθρωπος με αντικείμενα της έννοιας Αυτοκίνητο περιγράφοντας τη δυνατότητα των ανθρώπων να διαθέτουν ένα ή και περισσότερα αυτοκίνητα.
- **Ιεραρχίες και Ταξονομίες Εννοιών (Concept Hierarchies and Taxonomies):** Μια κλάση της οντολογίας μπορεί να διαθέτει υποκλάσεις που περιγράφουν αντικείμενα τα οποία όχι μόνο κληρονομούν τα χαρακτηριστικά και τις ιδιότητες της υπερκλάσης τους αλλά και μπορούν να διαθέτουν επιπλέον ιδιότητες κάνοντάς τα έτσι πιο συγκεκριμένα και ειδικευμένα σε σχέση με τα αντικείμενα της υπερκλάσης τους. Έτσι δημιουργούνται ιεραρχίες κλάσεων καταλήγοντας σε μια δενδρική δομή που απεικονίζει με σαφήνεια τον τρόπο που συνδέονται μεταξύ τους τα αντικείμενα. Η ταξονομία εννοιών τώρα αντιπροσωπεύει μια ιεραρχική δομή που συσχετίζει έννοιες με βάση σχέσεις γενίκευσης-ειδίκευσης.
- **Αξιόματα και Κανόνες (Axioms and Rules):** Πρόκειται για λογικούς κανόνες και περιορισμούς που πρέπει να διέπουν την οντολογία.

2.3 Ασαφής Λογική

Η ασαφής λογική (fuzzy logic) είναι ένα υπερσύνολο της κλασικής λογικής [3], η οποία έχει επεκταθεί ώστε να μπορεί να χειριστεί τιμές αληθείας μεταξύ του «απολύτου αληθούς» και του «απολύτου ψευδούς», δηλαδή τιμές στο διάστημα $[0, 1]$. Έχει τις ρίζες της στη θεωρία των ασαφών συνόλων (fuzzy set theory) που προτάθηκε από τον Lofti Zadeh στη δεκαετία του '60. Η βασική ιδέα είναι ότι ενώ στην κλασική συνολοθεωρία ένα αντικείμενο είτε ανήκει είτε δεν ανήκει σε ένα σύνολο, στην ασαφή συνολοθεωρία ένα αντικείμενο μπορεί να ανήκει με κάποιο βαθμό αληθείας.

2.3.1 Ασαφή Σύνολα

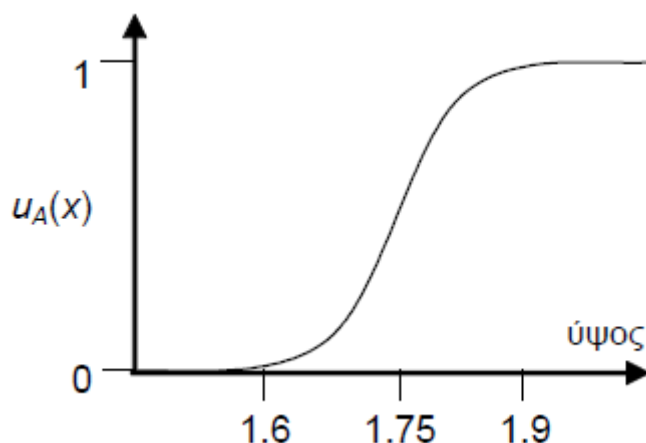
Ένα ασαφές σύνολο (fuzzy set) A ορίζεται ως ένα σύνολο διατεταγμένων ζευγών $(x, u_A(x))$ όπου $x \in S$ και $u_A(x) \in [0, 1]$ [3]. Το σύνολο S αποτελεί ένα ευρύτερο σύνολο αναφοράς

(universe of discourse) που περιλαμβάνει όλα τα αντικείμενα στα οποία μπορεί να γίνει αναφορά. Η τιμή $u_A(x)$ λέγεται βαθμός αληθείας (degree of truth) ή βαθμός ασάφειας ή βαθμός συγγένειας ή βαθμός συμμετοχής, παίρνει τιμές στο διάστημα $[0, 1]$ και συμβολίζει το βαθμό της συγγένειας του x στο A , δηλαδή, το κατά πόσο το στοιχείο x μπορεί να θεωρηθεί ότι ανήκει στο A . Τέλος, η συνάρτηση u_A ονομάζεται συνάρτηση συγγένειας (membership function) ή/και συνάρτηση συμμετοχής. Στην πράξη, η συνάρτηση συγγένειας μπορεί να προέρχεται από:

- υποκειμενικές εκτιμήσεις,
- προκαθορισμένες και απλοποιημένες μορφές,
- συχνότητες εμφανίσεων και πιθανότητες,
- φυσικές μετρήσεις,
- διαδικασίες μάθησης και προσαρμογής (συνήθως με νευρωνικά δίκτυα).

Επομένως, η διαφορά των ασαφών συνόλων με την κλασική συνολοθεωρία είναι ότι στη δεύτερη ισχύει $u_A(x) \in \{0, 1\}$, δηλαδή το x είτε ανήκει στο A και τότε $u_A(x)=1$ ή δεν ανήκει σε αυτό οπότε $u_A(x)=0$. Άρα, η ασαφής συνολοθεωρία μεταπίπτει στην αντίστοιχη κλασική, όταν οι δυνατές τιμές της συνάρτησης συγγένειας είναι μόνο 0 και 1.

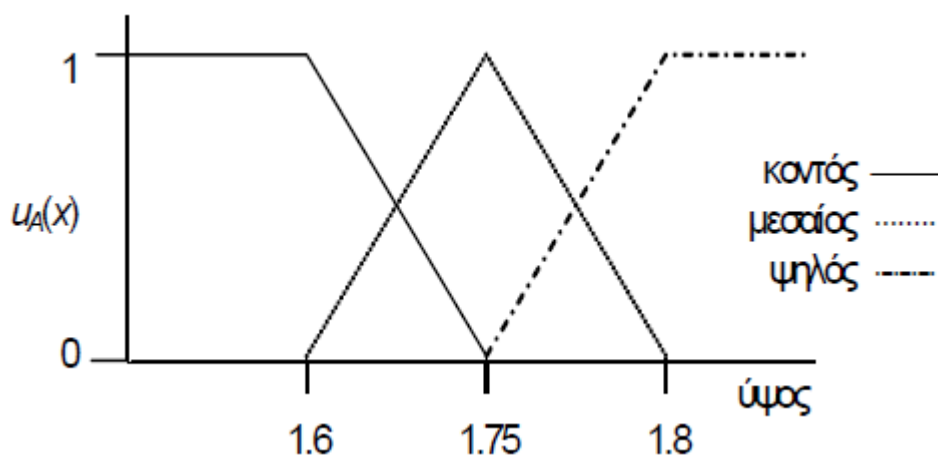
Ένας τρόπος αναπαράστασης των ασαφών συνόλων είναι μέσω της αναλυτικής έκφρασης της συνάρτησης συγγένειάς τους, u_A . Στο Σχήμα 2.2 απεικονίζεται μια πιθανή συνάρτηση συγγένειας (η ακριβής μορφή της εξαρτάται από το εκάστοτε πρόβλημα) που προσδιορίζει πότε κάποιος θεωρείται ψηλός.



Σχήμα 2.2: Συνάρτηση συγγένειας για το ασαφές σύνολο «ψηλός»

Έτσι, ένα άτομο με ύψος >1.95 θεωρείται ψηλό με $u_A(1.95) \approx 1$, κάθε άτομο με ύψος ≈ 1.75 θεωρείται ψηλό με $u_A(1.75) \approx 0.5$, ενώ κάθε άτομο με ύψος <1.60 θεωρείται ψηλό με $u_A(1.60) \approx 0$.

Συχνά, σε πολλές εφαρμογές χρησιμοποιείται η τμηματικώς γραμμική απεικόνιση συνάρτησης συγγένειας. Για παράδειγμα, το Σχήμα 2.3 απεικονίζει μια εκδοχή της συνάρτησης συγγένειας για τα ασαφή σύνολα «κοντός», «μεσαίος» και «ψηλός». Η αλληλοεπικάλυψη των συναρτήσεων συγγένειας για κάποιες τιμές του ύψους (για παράδειγμα στο διάστημα 1.6 ως 1.75) είναι εγγενές χαρακτηριστικό της θεωρίας ασαφών συνόλων και πάνω σε αυτό στηρίζεται η ασαφής συλλογιστική.



Σχήμα 2.3: Συνάρτηση συγγένειας για τα ασαφή σύνολα «κοντός», «μεσαίος» και «ψηλός»

2.3.2 Ιδιότητες Ασαφών Συνόλων

Για τα ασαφή σύνολα ορίζονται πράξεις και ισχύουν ιδιότητες ανάλογες με αυτές που ισχύουν στα κλασικά σύνολα [3]. Ορισμένες από αυτές είναι:

- Η ένωση (union) δύο ασαφών συνόλων A και B ορισμένων στο ίδιο σύνολο S είναι ένα νέο ασαφές σύνολο $u_{A \cup B}(x)$ για το οποίο ισχύει $u_{A \cup B}(x) = \max(u_A(x), u_B(x))$. Η ένωση δύο ασαφών συνόλων σχετίζεται με τη διάζευξη (OR) της ασαφούς λογικής.
- Η τομή (intersection) δύο ασαφών συνόλων A και B ορισμένων στο ίδιο σύνολο S είναι ένα νέο ασαφές σύνολο $u_{A \cap B}(x)$ για το οποίο ισχύει $u_{A \cap B}(x) = \min(u_A(x), u_B(x))$. Η τομή δύο ασαφών συνόλων σχετίζεται με τη σύζευξη (AND) της ασαφούς λογικής.
- Για τη συνάρτηση συγγένειας του συμπληρωματικού (complement) ενός ασαφούς συνόλου A ισχύει $c(x) = 1 - u_A(x)$. Η συμπληρωματικότητα στα ασαφή σύνολα σχετίζεται με την άρνηση (NOT) της ασαφούς λογικής.

- Για δύο ασαφή σύνολα A και B ορισμένα στο S , το A είναι υποσύνολο του B ($A \subseteq B$) αν $u_A(x) \leq u_B(x)$ για κάθε $x \in S$.

2.3.3 Ασαφείς Σχέσεις

Μια ασαφής σχέση R από το (κλασικό) σύνολο A στο (κλασικό) B είναι ένα ασαφές σύνολο επί του $A \times B$. Αντίστοιχα με την περίπτωση των ασαφών συνόλων ορίζονται η τομή και η ένωση ασαφών σχέσεων. Ορίζεται επίσης η αντίστροφη σχέση ως: $R^{-1}(s_1, s_2) = R(s_2, s_1)$

Η συμμετρικότητα ορίζεται ως: $R = R^{-1}$.

Επίσης υπάρχουν δύο περισσότερο γνωστές μέθοδοι σύνθεσης δύο ασαφών σχέσεων: η σύνθεση max-min (max-min composition) και η σύνθεση max-product (max-product composition). Αν $R_1(x, y)$ και $R_2(y, z)$ είναι δύο ασαφείς σχέσεις ορισμένες στα σύνολα $X \times Y$ και $Y \times Z$ αντίστοιχα, τότε η σύνθεσή τους δίνει μια νέα σχέση $R_1 \circ R_2$ ορισμένη στο $X \times Z$.

Για την ταυτοτική σχέση I ισχύει:

$$I(s_1, s_2) = 1, s_1 = s_2 \text{ και}$$

$$I(s_1, s_2) = 0, s_1 \neq s_2,$$

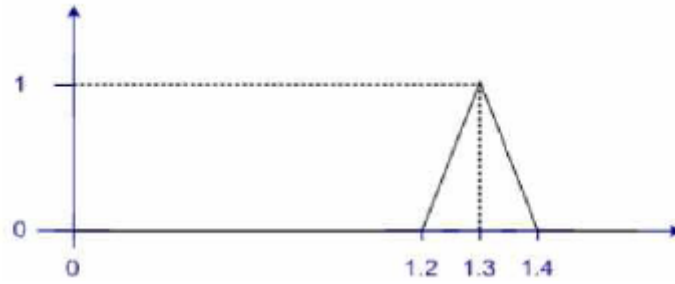
ενώ είναι το ουδέτερο στοιχείο της σύνθεσης, αφού ισχύει:

$$R \circ I = I \circ R = R, \text{ για κάθε } R.$$

Για την ανακλαστικότητα τέλος ισχύει: $R = I$.

2.3.4 Ασαφείς Αριθμοί και Γλωσσικές Μεταβλητές

Οι ασαφείς αριθμοί (fuzzy numbers) είναι ασαφή υποσύνολα του συνόλου των πραγματικών αριθμών και αντιπροσωπεύουν πραγματικούς αριθμούς [3]. Με τον όρο αντιπροσωπεύουν υπονοείται ότι η συνάρτηση συγγένειας λαμβάνει τιμές πιο κοντά στο 1 όσο προσεγγίζει κανείς τον αριθμό αυτό (είτε από αριστερά είτε από δεξιά) και πιο κοντά στο 0 όσο απομακρύνεται από αυτόν. Για παράδειγμα, η συνάρτηση συγγένειας του αριθμού ασαφές 1.3 θα μπορούσε να είναι όπως στο Σχήμα 2.4. Συνήθως χρησιμοποιούνται τριγωνικές, τραπεζοειδείς ή εκθετικές (για παράδειγμα γκαουσιανές) συναρτήσεις συγγένειας. Οι μη ασαφείς τιμές κάποιου μεγέθους αποκαλούνται crisp (σαφείς, συγκεκριμένες).

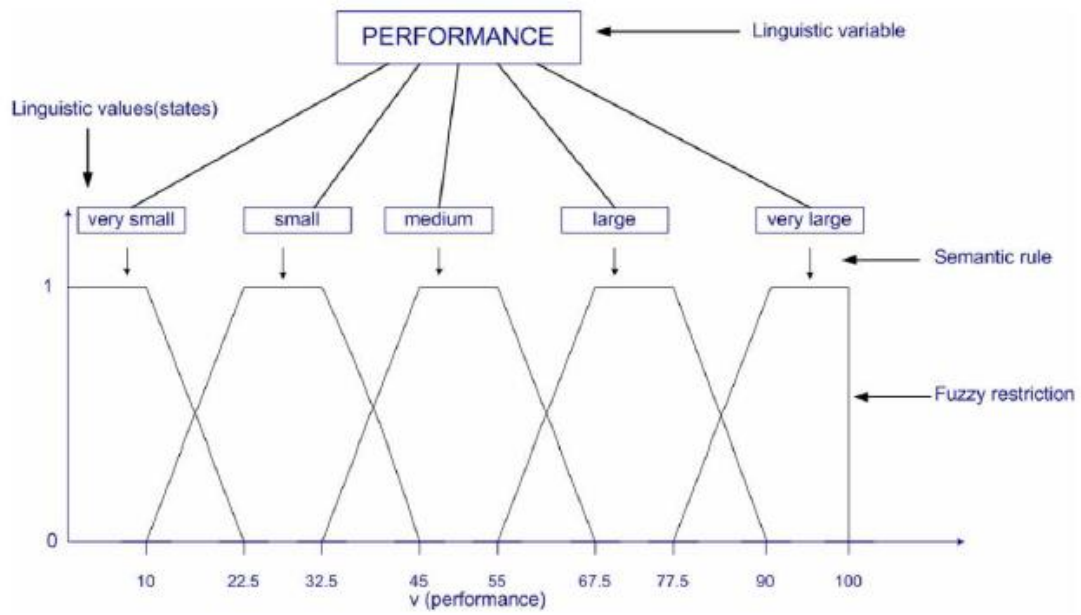


Σχήμα 2.4: Ασαφής τριγωνικός αριθμός

Οι γλωσσικές μεταβλητές (linguistic variables) ορίζονται με βάση τους ασαφείς αριθμούς. Μια γλωσσική μεταβλητή διαθέτει μία μεταβλητή βάση, οι τιμές της οποίας είναι πραγματικοί αριθμοί σε ένα προκαθορισμένο εύρος [14]. Μια μεταβλητή βάση μπορεί να έχει είτε φυσική υπόσταση (π.χ. θερμοκρασία) είτε να πρόκειται για κάτι μετρήσιμο (π.χ. αξιοπιστία). Στο πλαίσιο μιας γλωσσικής μεταβλητής χρησιμοποιούνται διαβαθμίσεις (π.χ. πολύ λίγο, λίγο, αρκετά, πάρα πολύ, κ.α.) οι οποίες αντιστοιχούν σε ασαφείς αριθμούς.

Με πιο τυπικό τρόπο τώρα, μια γλωσσική μεταβλητή αποτελείται από μια πεντάδα (u, T, X, g, m), όπου u είναι το όνομα της μεταβλητής, T είναι το σύνολο των γλωσσικών όρων που αναφέρονται στη μεταβλητή u και των οποίων η σημασία καλύπτει όλο το εύρος του καθολικού συνόλου X , g είναι ένας γραμματικός κανόνας για την παραγωγή γλωσσικών όρων και m είναι ένας σημασιολογικός κανόνας ο οποίος αναθέτει σε κάθε γλωσσικό όρο $t \in T$ μια σημασία $m(t)$ η οποία με τη σειρά της αντιστοιχεί σε ένα ασαφές σύνολο στο X .

Στο Σχήμα 2.5 δίνεται ένα διάγραμμα με τα ασαφή σύνολα που αντιστοιχούν στη γλωσσική μεταβλητή της απόδοσης. Παρατηρείται ο ρόλος που διαδραματίζουν σε μια γλωσσική μεταβλητή η μεταβλητή βάση (performance), οι γλωσσικές τιμές (very small, small, medium, large, very large) και οι σημασιολογικοί κανόνες που αντιστοιχίζουν κάθε γλωσσική τιμή στο αντίστοιχο ασαφές σύνολο.



Σχήμα 2.5: Παράδειγμα γλωσσικής μεταβλητής

3

Υβριδικό Πλαίσιο IKARUS-CBR

Στο παρόν κεφάλαιο παρουσιάζεται ο τρόπος διαχείρισης και εκμετάλλευσης ασαφούς γνώσης από το καινοτόμο υβριδικό πλαίσιο IKARUS-CBR μέσω της αποτελεσματικής ενσωμάτωσης της ασαφούς συνολοθεωρίας στο συνδυασμό οντολογιών και CBR, όπως παρουσιάστηκε και θεμελιώθηκε θεωρητικά στο [2]. Η προοπτική αυτή θα δια φωτίσει τους άξονες σχεδιασμού της πλατφόρμας διεπιφάνειας χρήστη που βασίζεται στο API (Application Programming Interface) του μοντέλου IKARUS-CBR. Οι άξονες αυτοί και το API θα παρουσιαστούν στο επόμενο κεφάλαιο.

Το πλαίσιο IKARUS-CBR λειτουργεί σε δύο άξονες, τον άξονα της αναπαράστασης ασαφούς γνώσης χρησιμοποιώντας οντολογίες και τον άξονα της ανάκτησης της γνώσης αυτής μέσω παραμετροποιήσιμου μηχανισμού υπολογισμού της σημασιολογικής ομοιότητας μεταξύ των περιπτώσεων.

3.1 Αναπαράσταση Ασαφούς Γνώσης

3.1.1 Οντολογίες και Ασάφεια

Σε επίπεδο αναπαράστασης γνώσης, μια οντολογία μπορεί να οριστεί ως ένα σύνολο εννοιών, στιγμιότυπων, ιδιοτήτων και σχέσεων (βλέπε παράγραφο 2.2.4). Μια έννοια (ή κλάση), όπως έχει ήδη αναφερθεί, αναπαριστά ένα σύνολο οντοτήτων που αφορούν ένα γνωστικό πεδίο

ενώ οι οντότητες που ανήκουν σε μια έννοια ονομάζονται στιγμιότυπα αυτής. Επίσης, μια σχέση συνδέει δύο στιγμιότυπα μεταξύ τους ενώ μια ιδιότητα συνδέει ένα στιγμιότυπο με τιμές προερχόμενες από κάποιο τύπο δεδομένων. Το πρόβλημα της ασάφειας σε μια οντολογία εντοπίζεται κυρίως στις σχέσεις και στις ιδιότητες.

Πιο συγκεκριμένα, μια οντολογική σχέση που συνδέει δύο στιγμιότυπα είναι ασαφής όταν η σύνδεση αυτή μπορεί, υπό ορισμένες συνθήκες, να θεωρηθεί ότι εμπεριέχει κάποιο ανακριβές ή αόριστο χαρακτηριστικό [1]. Η ύπαρξη και το νόημα της ανακρίβειας ή της αοριστίας αυτής εξαρτάται κάθε φορά από το προτιθέμενο νόημα της σχέσης και δεδομένου του τεράστιου εύρους των νοημάτων αυτών, η εξαντλητική αντιστοίχιση μεταξύ διαφορετικών τύπων σχέσεων και ασάφειας είναι πολύ δύσκολη. Παρόλα αυτά δύο γενικές κατηγορίες οντολογικών σχέσεων στις οποίες η ασάφεια μπορεί να λάβει συγκεκριμένο νόημα είναι οι ιεραρχικές και οι συσχετιστικές σχέσεις.

Μια ιεραρχία οργανώνει ένα σύνολο οντοτήτων σε μια δενδρική δομή ορίζοντας πρακτικά μια μερική διάταξη (ordering) των οντοτήτων αυτών βάσει κάποιας σχέσης. Δύο κοινοί τύποι ιεραρχιών είναι οι ταξονομίες και οι μερονομίες.

Μια ταξονομία βασίζεται στη γνωστή is-a σχέση η οποία συνδέει μια οντότητα ενός συγκεκριμένου τύπου με μια οντότητα ενός γενικότερου τύπου. Η σύνδεση αυτή μπορεί να θεωρηθεί ασαφής καθώς υποδηλώνει ότι το νόημα μιας οντότητας είναι πιο συγκεκριμένο από αυτό μιας άλλης αλλά δεν προσδιορίζει το ακριβές επίπεδο της ειδίκευσης αυτής. Έτσι, η ασάφεια σε μια ταξονομία αντικατοπτρίζει την έλλειψη πληροφορίας ως προς το πόσο "κοντά" βρίσκεται το νόημα μιας οντότητας με αυτό του γονέα της.

Από την άλλη πλευρά, μια μερονομία βασίζεται στη σχέση is-part-of (γνωστή και ως σχέση μέρους-όλου) η οποία συνδέει μια οντότητα με μια άλλη οντότητα της οποίας αποτελεί μέρος. Η φύση και το πιθανό νόημα της ασάφειας μιας μερονομικής σχέσης προκύπτει από το γεγονός ότι ένα μέρος είναι πάντα κάτι "λιγότερο" από το όλο. Αυτός ο όρος "λιγότερο" όταν δεν προσδιορίζεται ξεκάθαρα, είτε ποιοτικά είτε ποσοτικά, καθιστά τη σχέση ασαφή. Το τι ακριβώς σημαίνει αυτή η ασάφεια μπορεί να προσδιοριστεί από το γεγονός ότι η σχέση is-part-of προσπαθεί πάντα να λάβει υπόψη το βαθμό διαφοροποίησης του μέρους με το όλο.

Τέλος, οι συσχετιστικές σχέσεις συνδέουν οντότητες μεταξύ τους με μη ιεραρχικό τρόπο. Η ασάφεια στην περίπτωση αυτή προκύπτει από (και αντικατοπτρίζει την) έλλειψη ακριβούς

πληροφορίας σχετικά με την ισχύ της σύνδεσης μεταξύ των οντοτήτων είτε λόγω της εγγενούς αοριστίας της σχέσης είτε λόγω της απουσίας ακριβούς πληροφορίας για κάποιο χαρακτηριστικό της σχέσης.

Σε ό,τι αφορά τις ιδιότητες, θεωρούνται ασαφείς σε δύο περιπτώσεις [1]:

1. Όταν η σύνδεση μεταξύ στιγμιότυπου και τιμής που εκφράζει η ιδιότητα είναι ασαφής.
2. Όταν οι τιμές που μπορεί να λαμβάνει ένα στιγμιότυπο για αυτήν την ιδιότητα μπορεί να εκφραστεί με ασαφή τρόπο.

Η πρώτη περίπτωση είναι παρόμοια με αυτή των ασαφών σχέσεων καθώς η μόνη διαφορά μεταξύ μιας ιδιότητας και μιας σχέσης είναι ότι η πρώτη συνδέει στιγμιότυπα με τιμές αντί με άλλα στιγμιότυπα. Έτσι για παράδειγμα η ιδιότητα *κατηγορία* της έννοιας Βιβλίο, η οποία συνδέει συγκεκριμένα βιβλία με τις κατηγορίες τις οποίες τα βιβλία αυτά σχετίζονται, είναι ασαφής καθώς η έννοια της σχετικότητας είναι εγγενώς ασαφής. Στη δεύτερη περίπτωση ασαφείς τιμές μιας ιδιότητας αποτελούν εκφράσεις της μορφής "ψηλός", "ζεστός", "μέτριος", κλπ.

3.1.2 Μοντέλο Αναπαράστασης Ασαφών Οντολογιών

Με βάση την προηγούμενη παράγραφο, η αναπαράσταση της ασάφειας σε μια οντολογία μπορεί να γίνει χρησιμοποιώντας στοιχεία της Ασαφούς Άλγεβρας ως εξής:

- Μια ασαφής οντολογική σχέση μπορεί να αναπαρασταθεί ως μια ασαφής σχέση που συνδέει μεταξύ τους στιγμιότυπα με συγκεκριμένους βαθμούς.
- Μια ασαφής ιδιότητα που συνδέει στιγμιότυπα και τιμές με ασαφή τρόπο μπορεί να αναπαρασταθεί ως μια ιδιότητα οι τιμές της οποίας έχουν τη μορφή ασαφούς συνόλου.
- Μια ασαφής ιδιότητα που συνδέει στιγμιότυπα με ασαφείς τιμές μπορεί να αναπαρασταθεί ως μια ιδιότητα οι τιμές της οποίας αποτελούν όρους κάποιας ασαφούς γλωσσικής μεταβλητής.

Επομένως, μια Ασαφή Οντολογία ορίζεται [1] ως μια πλειάδα της μορφής: $O_F = \{C, I, FHR, FAR, FP, FLV, FVP\}$ όπου:

- C είναι ένα σύνολο εννοιών.

- I είναι ένα σύνολο στιγμιότυπων με κάθε στιγμιότυπο να ανήκει τουλάχιστον σε μια έννοια.
- FHR (Fuzzy Hierarchical Relation) και FAR (Fuzzy Associative Relation) είναι σύνολα ασαφών ιεραρχικών και συσχετιστικών σχέσεων αντίστοιχα. Κάθε σχέση $fr \in \{FHR \cup FAR\}$ είναι μια συνάρτηση της μορφής: $I^2 \rightarrow [0, 1]$.
- FP (Fuzzy Property) είναι ένα σύνολο ασαφών ιδιοτήτων, κάθε ασαφής ιδιότητα $fr \in FP$ είναι μια ασαφής συνάρτηση της μορφής $I \rightarrow F(X)$, όπου $F(X)$ είναι το σύνολον όλων των ασαφών συνόλων.
- FLV (Fuzzy Linguistic Variable) είναι ένα σύνολο ασαφών γλωσσικών μεταβλητών. Μια γλωσσική μεταβλητή $flv \in FLV$ είναι μια πλειάδα της μορφής $\{u, T, X, m\}$, όπου u είναι το όνομα της μεταβλητής, T (Fuzzy Linguistic Term) το σύνολο των γλωσσικών όρων της μεταβλητής u οι οποίοι αναφέρονται σε μια μεταβλητή βάσης οι τιμές της οποίας βρίσκονται στο σύνολο X και m είναι ένας σημασιολογικός κανόνας που αντιστοιχεί σε κάθε όρο $t \in T$ το νόημά του $m(t)$ το οποίο είναι ένα ασαφές δυναμοσύνολο (Fuzzy Set) του X .
- FVP (Fuzzy Valued Property) είναι ένα σύνολο ασαφώς αποτιμώμενων ιδιοτήτων (δηλαδή ιδιότητα με ασαφείς τιμές). Κάθε τέτοια ιδιότητα $fnr \in FVP$ είναι μια συνάρτηση της μορφής $I \rightarrow T$, όπου T είναι το σύνολο των γλωσσικών όρων κάποιας γλωσσικής μεταβλητής $flv \in FLV$.

Στην παραπάνω δομή και ιεραρχία του θεωρητικού μοντέλου της Ασαφούς Οντολογίας στηρίχθηκε το πλαίσιο IKARUS-CBR με στόχο τον ορισμό του σχήματος βάσει του οποίου μοντελοποιείται και αναπαρίσταται μια ασαφής οντολογία πεδίου. Επομένως, τα επιμέρους κομμάτια από τα οποία αποτελείται η ασαφής οντολογία που εισάγεται και δημιουργείται στο σύστημα φαίνονται στο Σχήμα 3.1. Το σχήμα αυτό ορίστηκε σε ένα αρχείο owl με τα ορθογώνια να αναπαριστούν κλάσεις και τις γραμμές διακεκομμένες ή μη που τα ενώνουν να αποτελούν σχέσεις μεταξύ των κλάσεων.

3.1.3 Οντολογικό Μοντέλο Αναπαράστασης Ασαφών Περιπτώσεων

Η αναπαράσταση των περιπτώσεων σε ένα CBR μοντέλο με βάση οντολογίες γίνεται θεωρώντας τις κατηγορίες των περιπτώσεων ως έννοιες, τις περιπτώσεις ως στιγμιότυπα των εννοιών αυτών και τα γνωρίσματα των περιπτώσεων ως οντολογικές σχέσεις ή ιδιότητες. Οι τιμές που μπορούν να λάβουν τα γνωρίσματα-σχέσεις είναι στιγμιότυπα που ορίζονται σε κάποια οντολογία πεδίου. Με βάση αυτά, το πλαίσιο IKARUS-CBR επιτρέπει την αναπαράσταση της ασάφειας σε μία περίπτωση με δύο τρόπους:

1. Επιτρέποντας στα γνωρίσματα των περιπτώσεων να έχουν τη μορφή ασαφών οντολογικών σχέσεων ή ασαφών οντολογικών ιδιοτήτων.
2. Επιτρέποντας στα γνωρίσματα-σχέσεις να λαμβάνουν τιμές από κάποια ασαφή οντολογία πεδίου.

Πιο τυπικά, μια οντολογία ασαφών περιπτώσεων ορίζεται [1] ως ένα υποσύνολο μιας ασαφούς οντολογίας, όπου:

- FCT (Fuzzy Case Types) $\subseteq C$ είναι ένα σύνολο από τύπους ασαφών περιπτώσεων.
- FC (Fuzzy Case) $\subseteq I$ είναι ένα σύνολο από ασαφείς περιπτώσεις. Κάθε ασαφής περίπτωση ανήκει σε ένα συγκεκριμένο τύπο ασαφών περιπτώσεων.
- FRA (Fuzzy Relation Attribute) $\subseteq \{FHR \cup FAR\}$ είναι ένα σύνολο από γνωρίσματα-χαρακτηριστικά περιπτώσεων τα οποία ορίζονται ως ασαφείς σχέσεις. Κάθε $fra \in FRA$ είναι μια συνάρτηση $FC \times I \rightarrow [0, 1]$.
- FPA (Fuzzy Property Attribute) $\subseteq FP$ είναι ένα σύνολο από γνωρίσματα-χαρακτηριστικά περιπτώσεων τα οποία ορίζονται ως ασαφείς ιδιότητες. Κάθε $fpa \in FPA$ είναι μια συνάρτηση $FC \rightarrow F(U)$, όπου $F(U)$ είναι το σύνολο όλων των ασαφών συνόλων.
- $FVPA$ (Fuzzy Valued Property Attribute) $\subseteq FVP$ είναι ένα σύνολο από γνωρίσματα περιπτώσεων τα οποία ορίζονται ως ασαφείς αποτιμώμενες ιδιότητες. Κάθε $fvpa \in FVPA$ είναι μια συνάρτηση $FC \rightarrow T$ όπου T είναι το σύνολο των γλωσσικών όρων μιας ασαφούς γλωσσικής μεταβλητής $flv \in FLV$.

Αξίζει να αναφερθεί ότι επειδή μια μη-ασαφής οντολογία είναι μια ειδική περίπτωση ασαφούς οντολογίας στην οποία όλοι οι ασαφείς βαθμοί είναι ίσοι με 1, ο παραπάνω φορμαλισμός καθιστά εφαρμόσιμες και τις τεχνικές και μεθόδους που έχουν αναπτυχθεί στο

πλαίσιο του κλασικού συνδυασμού CBR και οντολογιών και αφορούν μη-ασαφή (crisp) χαρακτηριστικά (σχέσεις και ιδιότητες).

3.2 Ανάκτηση Ασαφούς Γνώσης

Όπως έχει ήδη αναφερθεί στην αρχή του κεφαλαίου η διαδικασία σύγκρισης μεταξύ μιας ζητούμενης περίπτωσης και μιας αποθηκευμένης γίνεται με χρήση μέτρων ομοιότητας βασισμένα στις τιμές που έχουν λάβει τα επιμέρους γνωρίσματα-χαρακτηριστικά τους και στη συνέχεια συναθροίζοντας τα μέτρα αυτά σε μια τιμή. Ο υπολογισμός των μέτρων ομοιότητας εξαρτάται από τη φύση και το σκοπό των γνωρισμάτων και ιδιαίτερα από τον τύπο και το εύρος των τιμών που αυτά λαμβάνουν.

Έτσι, για την αποτελεσματική ανάκτηση περιπτώσεων το IKARUS-CBR πλαίσιο αναγνωρίζει τους διαφορετικούς τύπους γνωρισμάτων που διαθέτει, ορίζει μέτρα ομοιότητας για τον καθένα από αυτά και παρέχει τελεστές για συνάθροιση ομοιότητας [1]. Οι τύποι αυτοί είναι: Ασαφείς Σχέσεις-Γνωρίσματα (Fuzzy Relation Attributes - FRA), Ασαφείς Ιδιότητες-Γνωρίσματα (Fuzzy Properties Attributes - FPA) και Ασαφώς Αποτιμώμενες Ιδιότητες-Γνωρίσματα (Fuzzy Valued Properties Attributes - FVPA). Έτσι για τον καθένα τύπο γνωρισμάτων ορίζονται μέθοδοι εκτίμησης της ομοιότητας δίνοντας έμφαση στο ρόλο που παίζει η ασάφεια για τον υπολογισμό της ομοιότητας. Στη συνέχεια θα αναλυθούν για τον καθένα τύπο γνωρισμάτων τα μέτρα ομοιότητας που προτείνονται και στο τέλος θα αναφερθεί το σχήμα ομοιότητας που ορίζει το πλαίσιο IKARUS-CBR.

3.2.1 Ομοιότητα για Ασαφείς Ιδιότητες-Γνωρίσματα (FPA)

Έστω το σενάριο κατά το οποίο η αποθηκευμένη περίπτωση χαρακτηρίζεται από ένα γνώρισμα τύπου ασαφούς ιδιότητας με τιμές ένα ασαφές (fuzzy) σύνολο λεκτικών ενώ η ζητούμενη περίπτωση χαρακτηρίζεται και αυτή από μια ασαφή ιδιότητα-γνώρισμα με τιμές μη ασαφές (crisp) σύνολο λεκτικών. Στην ουσία η διαφορά των δύο παραπάνω συνόλων λεκτικών έγκειται στην απόδοση βαθμού αληθείας στο πρώτο.

Αν τώρα θεωρηθεί και το πρώτο σύνολο από τα παραπάνω μη ασαφές τότε η ομοιότητα μεταξύ των δύο περιπτώσεων θα προέκυπτε με τη σύγκριση των τιμών των γνωρισμάτων σε δύο επίπεδα: στο επίπεδο των λεκτικών των συνόλων και στο επίπεδο των συνόλων τους. Στο πρώτο επίπεδο η ομοιότητα λαμβάνει δυαδικές τιμές, δηλαδή την τιμή 1 εάν κατά τη σύγκριση των λεκτικών των υπό σύγκριση συνόλων βρεθούν ότι είναι πανομοιότυπα, ενώ σε διαφορετική περίπτωση λαμβάνει την τιμή 0. Στο επίπεδο συνόλων συναθροίζονται οι

ομοιότητες των λεκτικών σε μία τιμή ανάλογα με την εφαρμογή του σεναρίου. Για το λόγο αυτόν ορίζονται διαφορετικές συναρτήσεις συνάθροισης της ομοιότητας των περιπτώσεων στο πλαίσιο IKARUS-CBR.

Το σύνολο των λεκτικών της αποθηκευμένης περίπτωσης όμως χαρακτηρίζεται από έναν ασαφή βαθμό που θα πρέπει να λαμβάνεται υπόψη στην απόδοση ομοιότητας με τη ζητούμενη περίπτωση. Έτσι κατά τη σύγκριση των λεκτικών, με το ένα να ανήκει στο ασαφές σύνολο της αποθηκευμένης και να έχει ένα βαθμό ασάφειας, ενώ το άλλο να ανήκει στο μη ασαφές της ζητούμενης περίπτωσης, θα πρέπει να λαμβάνεται υπόψη και ο βαθμός αυτός.

Όμως η απόφαση για το αν η ασάφεια και η ανακρίβεια της συμμετοχής του λεκτικού στο σύνολο θα διαδραματίσουν ρόλο στην απόδοση ομοιότητας θα πρέπει να εξαρτάται από το συγκεκριμένο σενάριο εφαρμογής. Έτσι, για τη διάκριση της διαφορετικότητας του ρόλου της ασάφειας στην αποτίμηση της ομοιότητας, ορίζεται για μια ασαφής ιδιότητα-γνώρισμα (FPA) η τιμή Fuzzy Property Attribute Similarity Context ως συνάρτηση που λαμβάνει τιμή 0 ή 1 και προσδιορίζει αν η ασάφεια του συνόλου των λεκτικών της αποθηκευμένης περίπτωσης θα ληφθεί υπόψη στην απόδοση ομοιότητας. Η τιμή 0 δηλώνει το μη συγκερασμό της ασάφειας ενώ η τιμή 1 το ακριβές αντίθετο. Στην πρώτη περίπτωση το σύνολο των λεκτικών της αποθηκευμένης περίπτωσης θεωρείται μη ασαφές.

3.2.2 Ομοιότητα για Ασαφώς Αποτιμώμενες Ιδιότητες-Γνωρίσματα (FVPA)

Ας θεωρηθεί τώρα το σενάριο κατά το οποίο η αποθηκευμένη περίπτωση χαρακτηρίζεται από ένα γνώρισμα τύπου ασαφώς αποτιμώμενης ιδιότητας με τιμές γλωσσικούς όρους ενώ η ζητούμενη περίπτωση χαρακτηρίζεται και αυτή από μια ασαφώς αποτιμώμενη ιδιότητα - γνώρισμα με τιμή ένα γλωσσικό όρο. Η ομοιότητα τότε των περιπτώσεων με βάση μόνο το γνώρισμα αυτό βρίσκεται από τη σύγκριση των όρων αυτών.

3.2.3 Ομοιότητα για Ασαφείς Σχέσεις-Γνωρίσματα (FRA)

Έστω το σενάριο στο οποίο η αποθηκευμένη περίπτωση χαρακτηρίζεται από ένα γνώρισμα τύπου ασαφούς σχέσης με τιμές ένα ασαφές (fuzzy) σύνολο στιγμιότυπων, που είναι παρμένα από μία ασαφής οντολογία πεδίου, ενώ η ζητούμενη περίπτωση χαρακτηρίζεται και αυτή από μια ασαφή σχέση-γνώρισμα με τιμές μη ασαφές (crisp) σύνολο στιγμιότυπων. Τότε για την απόδοση ομοιότητας μεταξύ των συγκεκριμένων περιπτώσεων συγκρίνονται τα δύο παραπάνω σύνολα στιγμιότυπων. Η σύγκριση αυτή συντελείται με ανάλογο τρόπο με αυτή

που γίνεται για τις ασαφείς ιδιότητες-γνωρίσματα με τη μόνη διαφορά ότι συγκρίνονται στιγμιότυπα και όχι λεκτικά.

Για τη σύγκριση δύο στιγμιότυπων θα εξεταστούν οι σχέσεις της οντολογίας που συνδέουν με κάποιο τρόπο τα στιγμιότυπα αυτά. Επειδή όμως τα στιγμιότυπα αυτά συνδέονται με ασαφείς σχέσεις σημαντικό ρόλο στην απόδοση ομοιότητας παίζει και ο βαθμός ασάφειας-συμμετοχής των στιγμιότυπων στην ασαφή σχέση. Δεν αγνοούνται όμως σε καμία περίπτωση το πεδίο εφαρμογής και η ανάγκη για διαφορετική απόδοση πλαισίου ανάλογα με το χρήστη, στοιχεία που σε συνδυασμό με τα παραπάνω οδηγούν στην απόδοση ομοιότητας για δύο διαφορετικά στιγμιότυπα.

Για τους παραπάνω λόγους ορίζονται στο πλαίσιο IKARUS-CBR τα στοιχεία:

- Το Fuzzy Ontology Relation Similarity Context (FORSC), μοντέλο που προσδιορίζει το ρόλο των σχέσεων της ασαφούς οντολογίας στη διαδικασία απόδοσης ομοιότητας μεταξύ στιγμιότυπων.
- Έναν αλγόριθμο για την εκτίμηση της ομοιότητας μεταξύ στιγμιότυπων.

Πιο συγκεκριμένα, ένα στιγμιότυπο μιας οντολογίας συνδέεται με άλλα στιγμιότυπα μέσω σχέσεων ή και σύνθεση από αυτές. Ο στόχος δηλαδή του FORSC είναι ο προσδιορισμός του αν και σε ποιο βαθμό οι συνδέσεις αυτές θα ερμηνευτούν ως όμοιες. Επομένως, δεδομένης μιας ασαφούς οντολογίας όπως ορίστηκε στην παράγραφο 3.1.2 ($O_F = \{C, I, FHR, FAR, FP, FLV, FVP\}$), το FORSC προσδιορίζει:

- Πως κάθε ιεραρχική σχέση $R \in FHR$ θα χρησιμοποιηθεί για τον υπολογισμό της ομοιότητας μεταξύ δύο στιγμιότυπων
- Πως η αντίστροφη σχέση μιας ιεραρχικής σχέσης $R \in FHR$ θα χρησιμοποιηθεί για τον υπολογισμό της ομοιότητας μεταξύ δύο στιγμιότυπων
- Πως κάθε συσχετιστική σχέση $R \in FAR$ θα χρησιμοποιηθεί για τον υπολογισμό της ομοιότητας μεταξύ δύο στιγμιότυπων
- Πως η αντίστροφη σχέση μιας συσχετιστικής σχέσης $R \in FAR$ θα χρησιμοποιηθεί για τον υπολογισμό της ομοιότητας μεταξύ δύο στιγμιότυπων και τέλος
- Πως η σύνθεση n σχέσεων $R_1, R_2, \dots, R_n \in FHR \cup FHR^{-1} \cup FAR \cup FAR^{-1}$ θα χρησιμοποιηθεί για τον υπολογισμό της ομοιότητας μεταξύ δύο στιγμιότυπων, με τα σύνολα FHR^{-1} και FAR^{-1} να περιέχουν τις αντίστροφες σχέσεις των συνόλων FHR και FAR αντίστοιχα.

Για τους παραπάνω σκοπούς ορίζονται τα εξής πεδία:

- Το Fuzzy Hierarchical Relation Similarity Context ορισμένο ως συνάρτηση $fhrsc : FHR \rightarrow [-1, 1]$.
- Το Fuzzy Reverse Hierarchical Relation Similarity Context ορισμένο ως συνάρτηση $frhsc : FHR \rightarrow [-1, 1]$.
- Το Fuzzy Associative Relation Similarity Context ορισμένο ως συνάρτηση $farsc : FAR \rightarrow [-1, 1]$, που προσδιορίζει το βαθμό δήλωσης ομοιότητας μεταξύ δύο στιγμιότυπων που συνδέονται με μια σχέση $R \in FAR$.
- Το Fuzzy Reverse Associative Relation Similarity Context ορισμένο ως συνάρτηση $frarsc : FAR \rightarrow [-1, 1]$, που προσδιορίζει το βαθμό δήλωσης ομοιότητας μεταξύ δύο στιγμιότυπων που συνδέονται με μια σχέση $R \in FAR^{-1}$.
- Το Fuzzy Composite Relation Similarity Context ορισμένο ως συνάρτηση $fcpsc : (FHR \cup FHR^{-1} \cup FAR \cup FAR^{-1})^n \rightarrow [-1, 1]$, με n τον αριθμό των υπό σύνθεση σχέσεων.

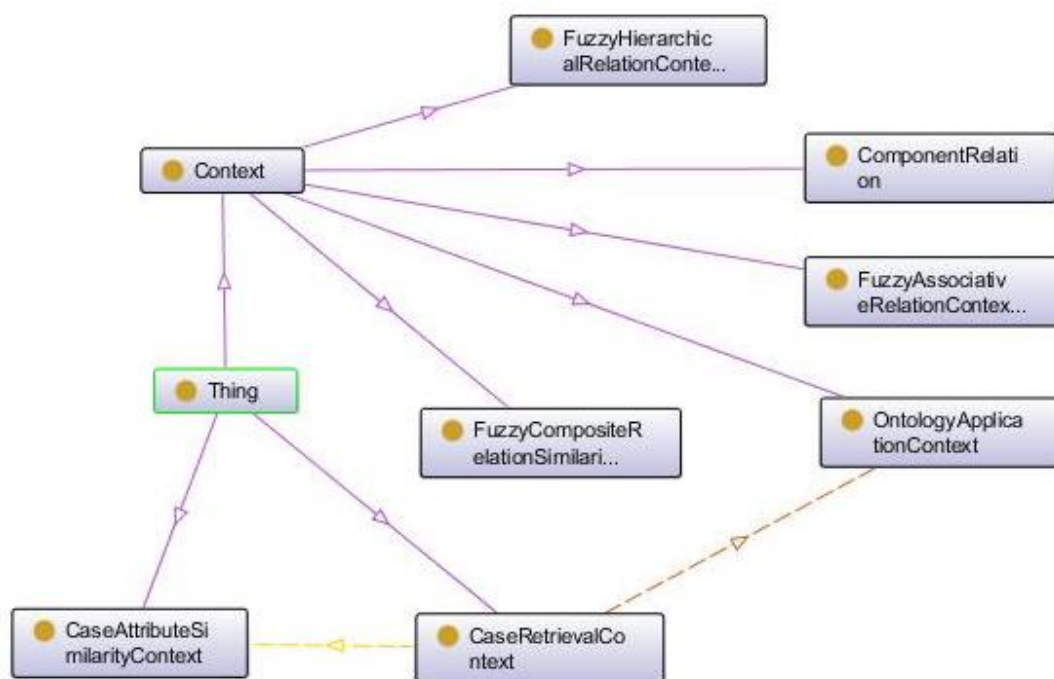
Η τιμή -1 υποδηλώνει ότι η σχέση ή ο συνδυασμός αυτών δε θα πρέπει να προσμετρηθεί στην εκτίμηση ομοιότητας, ενώ η τιμή 1 το ακριβές αντίθετο, δηλαδή ότι δύο στιγμιότυπα που συνδέονται με τη σχέση αυτή θα πρέπει να θεωρηθούν πανομοιότυπα. Οποιαδήποτε τιμή μεταξύ -1 και 1 υποδηλώνει μια ενδιάμεση κατάσταση.

3.2.4 Σχήμα Ομοιότητας πλαισίου

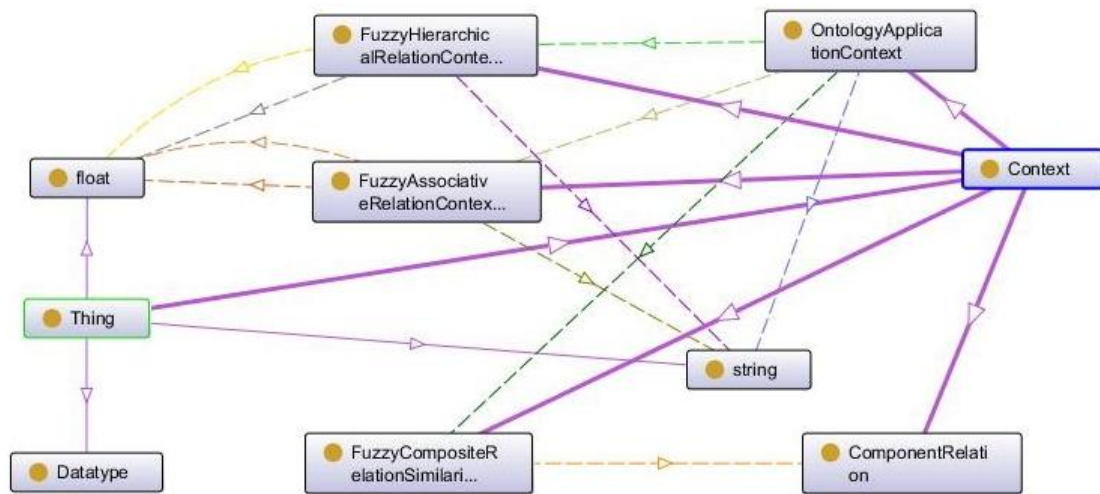
Για τη σύνθεση του σχήματος απόδοσης ομοιότητας ορίστηκαν σε δύο αρχεία owl, στο πλαίσιο IKARUS-CBR, η δομή του πλαισίου ομοιότητας για τα γνωρίσματα των περιπτώσεων (δηλαδή case attribute similarity context) και η δομή του πλαισίου ομοιότητας για τις οντολογικές σχέσεις (δηλαδή του fuzzy ontology relation similarity context).

Για την ανάθεση πλαισίου (context) συνολικά σε κάποιον τύπο περιπτώσεων ορίστηκε το πλαίσιο για ανάκτηση περιπτώσεων (case retrieval context). Τα case retrieval contexts αποτελούνται από πολλά case attribute similarity contexts, δηλαδή πλαίσιο που ανατίθεται στα γνωρίσματα των περιπτώσεων με στόχο την ανάκτηση. Αξίζει να αναφερθεί ότι ένας τύπος περιπτώσεων μπορεί να διαθέτει πολλά πλαίσια, αλλά μόνο ένα από αυτά πρέπει να είναι ενεργό, δηλαδή να συμμετέχει στον υπολογισμό των μέτρων ομοιότητας. Όλα αυτά περιγράφονται στο Σχήμα 3.2, στο οποίο με μωβ απεικονίζεται η σχέση hasSubclass, με διακεκομμένο καφέ η hasOntologySimilarityContext και με διακεκομμένο κίτρινο αυτή της hasAttributeSimilarityContext.

Στο ίδιο σχήμα διακρίνεται και η κλάση Πλαίσιο (Context) που αφορά τις οντολογικές σχέσεις και αναλύεται στο Σχήμα 3.3. Ορίζονται έτσι το Fuzzy Hierarchical Relation Context (για τις ιεραρχικές σχέσεις), το Fuzzy Associative Relation Context (για τις συσχετιστικές σχέσεις) και το Fuzzy Composite Relation Context (για τις σύνθετες σχέσεις), με το τελευταίο πλαίσιο να περιλαμβάνει πλαίσια από τις σχέσεις από τις οποίες αποτελείται (ComponentRelation). Το πλαίσιο αυτό δηλώνει το βαθμό κατά τον οποίο θα συμμετέχει στον υπολογισμό της ομοιότητας δύο στιγμιότυπων που συνδέονται με τη σχέση στην οποία ανήκει το εν λόγω πλαίσιο. Έτσι ορίζονται πεδία όπως τα fhrsc, frhrsc, farsc, frarsc και fcrsc που λαμβάνουν τιμές πραγματικών αριθμών στο διάστημα [-1, 1] και αναλύθηκαν στην προηγούμενη παράγραφο.



Σχήμα 3.2 Μοντέλο αναπαράστασης πλαισίου περιπτώσεων



Σχήμα 3.3 Μοντέλο αναπαράστασης πλαισίου ομοιότητας οντολογικών σχέσεων

4

Σχεδίαση Συστήματος

Στο κεφάλαιο αυτό περιγράφονται οι άξονες σχεδίασης της γραφικής διεπιφάνειας ή διεπαφής χρήστη (Graphical User Interface - GUI) βασισμένοι όχι μόνο στα σενάρια λειτουργίας του που επίσης θα παρουσιαστούν στο κεφάλαιο αυτό αλλά και στο μοντέλο αναπαράστασης και ανάκτησης της γνώσης του πλαισίου IKARUS-CBR το οποίο αναφέρθηκε στο 3^ο κεφάλαιο. Οι άξονες αυτοί θα βοηθήσουν τον αναγνώστη να κατανοήσει τις μεθόδους που χρησιμοποιήθηκαν για την υλοποίησή του, μέθοδοι που θα αναλυθούν στο επόμενο κεφάλαιο. Επίσης, θα παρουσιαστεί το μοντέλο αρχιτεκτονικής που χρησιμοποιήθηκε για τη σχεδίασή του. Στο τέλος αναφέρονται τα βασικά λειτουργικά συστατικά του API του IKARUS-CBR, δηλαδή τα interfaces από τα οποία αντλήθηκαν μέθοδοι που χρησιμοποιήθηκαν από το GUI.

4.1 Ανάλυση Απαιτήσεων

Πριν αναλυθεί το μοντέλο αρχιτεκτονικής του συστήματος, θα γίνει μια περιεκτική αναφορά στους χρήστες του συστήματος και τις περιπτώσεις χρήσης του από τους τελευταίους. Οι περιπτώσεις χρήσης ειδικά του συστήματος θα αποτελέσουν οδηγό ως προς τον τρόπο που θα δομηθεί η γραφική διεπιφάνεια χρήστη.

4.1.1 Χρήστες

Οι πιθανοί χρήστες του συστήματος χωρίζονται στις παρακάτω δύο ομάδες με διαφορετικές ανάγκες για αναπαράσταση και ανάκτηση ασαφούς γνώσης αλλά με τις ίδιες απαιτήσεις ως προς τον τρόπο λειτουργίας του συστήματος:

- Στη πρώτη ομάδα συγκαταλέγονται οι μηχανικοί γνώσης (knowledge engineers), οι οποίοι θα ασχοληθούν με τον ορισμό του μοντέλου γνώσης που θα καλύψει το πεδίο γνώσης που τους ενδιαφέρει. Διαθέτουν εξαιρετική εμπειρία και ευχέρεια στη διαχείριση συστημάτων πληροφορικής και έτσι δε θα δυσκολευτούν ιδιαίτερα στη μοντελοποίηση και αναπαράσταση ασαφούς γνώσης. Για το συγκεκριμένο σκοπό φυσικά θα συνεργαστούν με ειδικούς γνώσης (experts) οι οποίοι όμως θα λειτουργήσουν συμβουλευτικά και δε χρειάζεται να λάβουν μέρος στη διαδικασία αναπαράστασης ασαφούς γνώσης. Για παράδειγμα για μια αξιόπιστη μοντελοποίηση και αναπαράσταση γνώσης στον τομέα των ιστορικών γεγονότων και μαχών του Β' Παγκοσμίου Πολέμου θα ζητηθεί σίγουρα η γνώμη των ιστορικών ειδικά αυτών που έχουν ειδικευθεί στη συγκεκριμένη χρονική περίοδο. Έτσι οι μηχανικοί γνώσης ορίζουν με σαφήνεια το σχήμα γνώσης ανάλογα με την περίπτωση της εφαρμογής και έχουν τον πλήρη έλεγχο πάνω στο σχήμα αυτό σε αντίθεση με την επόμενη ομάδα. Απαιτούν φιλικό και εύχρηστο γραφικό περιβάλλον, ενώ πρέπει να συνυπολογιστεί και η ανάγκη τους για γρήγορη απόκριση του συστήματος στις εντολές τους.
- Στην δεύτερη ομάδα χρηστών διακρίνονται οι διαχειριστές των μοντέλων γνώσης οι οποίοι είναι υπεύθυνοι για τη λειτουργία της εφαρμογής με βάση το σχήμα γνώσης που ορίστηκε από τους μηχανικούς γνώσης. Έτσι, στην ομάδα αυτή δεν επιτρέπεται να αλλάξει το ήδη ορισμένο σχήμα γνώσης αλλά απλά να το συντηρεί και να παρεμβαίνει στο περιεχόμενό του. Εδώ συναντούνται τα στελέχη των επιχειρήσεων που αναλαμβάνουν τη λειτουργία της εφαρμογής IKARUS-CBR, τα οποία όμως ενδέχεται να μη διαθέτουν μεγάλη ή ακόμα και καθόλου εμπειρία διαχείρισης συστημάτων μοντελοποίησης και αναπαράστασης ασαφούς γνώσης. Υπάρχει επίσης η πιθανότητα να μην είναι εξοικειωμένοι με τα συστήματα πληροφορικής, η οποία πάντως είναι μικρή αφού στον επιχειρηματικό κόσμο δεν είναι και το πιο σίφρον για μια επιχείρηση να διαθέτει προσωπικό μη συμφιλωμένο με τα σύγχρονα πληροφοριακά συστήματα. Για τους παραπάνω λόγους καθίσταται σαφής η ανάγκη για φιλική ως προς το χρήστη, λιτή και απεριττή σχεδίαση της διεπιφάνειας χρήστη. Είναι εξίσου απαραίτητο για την ομάδα αυτή η προσαρμογή και η εξοικείωσή τους στο σύστημα να γίνεται ταχύτατα έτσι ώστε να χρησιμοποιήσουν άμεσα το σύστημα και να μην αναγκαστούν να διανύσουν μεγάλη περίοδο αφομοίωσης της λειτουργίας του, κάτι που ενδέχεται να τους οδηγήσει εν τέλει στην απόρριψή του. Έτσι, στις

απαιτήσεις τους συγκαταλέγονται όχι μόνο η αποφυγή ύπαρξης περιττής πληροφορίας στη διεπιφάνεια χρήστη κάτι που μπορεί να τους αποπροσανατολίσει αλλά και η προσεκτική και επιμελής σχεδίασή της με στόχο οι χρήστες να κατανοούν ανά πάσα στιγμή σε ποια κατάσταση βρίσκονται και να είναι σε θέση να γνωρίζουν τι αλλαγές θα προκύψουν στο σύστημα μετά από τη δικιά τους παρέμβαση με τη μορφή γεγονότων-συμβάντων, όπως για παράδειγμα με ένα κλικ του ποντικιού.

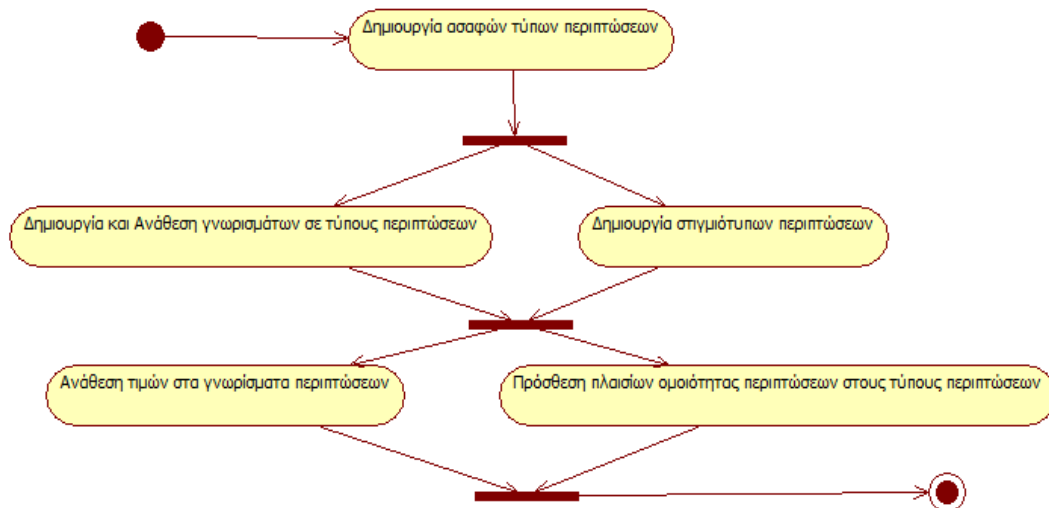
Κρίνοντας έτσι από τις απαιτήσεις των χρηστών, κύριος σχεδιαστικός γνώμονας του γραφικού περιβάλλοντος διεπιφάνειας χρήστη αποτελούν όχι μόνο η φιλικότητα, η σαφήνεια και η ευχρηστία ως προς το χρήστη αλλά και η δυνατότητα για διαχωρισμό των λειτουργιών των παραπάνω ομάδων έτσι ώστε η δεύτερη να μην παρεμβαίνει στο χώρο δράσης της πρώτης. Η παρούσα διπλωματική ασχολήθηκε πάντως αποκλειστικά με τις ανάγκες της πρώτης ομάδας για μοντελοποίηση και αναπαράσταση του σχήματος γνώσης χωρίς να προβλέπει μειωμένες διαχειριστικές λειτουργίες για τη δεύτερη ομάδα.

4.1.2 Περιπτώσεις Χρήσης

Η βασική περίπτωση χρήσης του συστήματος συνίσταται στη μοντελοποίηση και αναπαράσταση ασαφούς γνώσης για ένα δεδομένο πεδίο γνώσης με στόχο βέβαια την ανάκτησή της και την επαναχρησιμοποίησή της. Έτσι, για τους παραπάνω σκοπούς ο χρήστης μπορεί να κατασκευάσει μια οντολογία ασαφών περιπτώσεων, μια ασαφής οντολογία και ένα πλαίσιο ανάκτησης περιπτώσεων.

4.1.2.1 Κατασκευή και Διαχείριση Σχήματος Ασαφών Περιπτώσεων

Για τη σωστή διαδικασία ορισμού ενός σχήματος ασαφών περιπτώσεων όπως αυτό ορίστηκε στην παράγραφο 3.1.3 ο χρήστης αρχικά θα πρέπει να δημιουργήσει ένα σύνολο από ασαφείς τύπους περιπτώσεων. Στη συνέχεια σε κάθε έναν από αυτούς θα πρέπει να αναθέσει σύνολα γνωρισμάτων-χαρακτηριστικών περιπτώσεων, τα οποία μπορεί να είναι είτε ασαφείς σχέσεις ή μη ασαφείς είτε ασαφείς ιδιότητες ή μη ασαφείς είτε τέλος ασαφώς αποτιμώμενες ιδιότητες. Ταυτόχρονα σε κάθε τύπο περιπτώσεων προσθέτει στιγμιότυπα περιπτώσεων. Μόλις ολοκληρωθούν οι παραπάνω δύο ενέργειες, ο χρήστης σε κάθε περίπτωση πρέπει να θέσει τιμές για κάθε γνώρισμα (σχέσεις και ιδιότητες) που διαθέτει ο τύπος περιπτώσεων του οποίου αποτελεί στιγμιότυπο. Τέλος, σε κάθε τύπο περιπτώσεων θα προστεθούν πλαίσια ομοιότητας περιπτώσεων με τη δυνατότητα ενεργοποίησης ενός εξ αυτών. Οι ενέργειες αυτές φαίνονται στο διάγραμμα δραστηριοτήτων (activity diagram) του σχήματος 4.1.



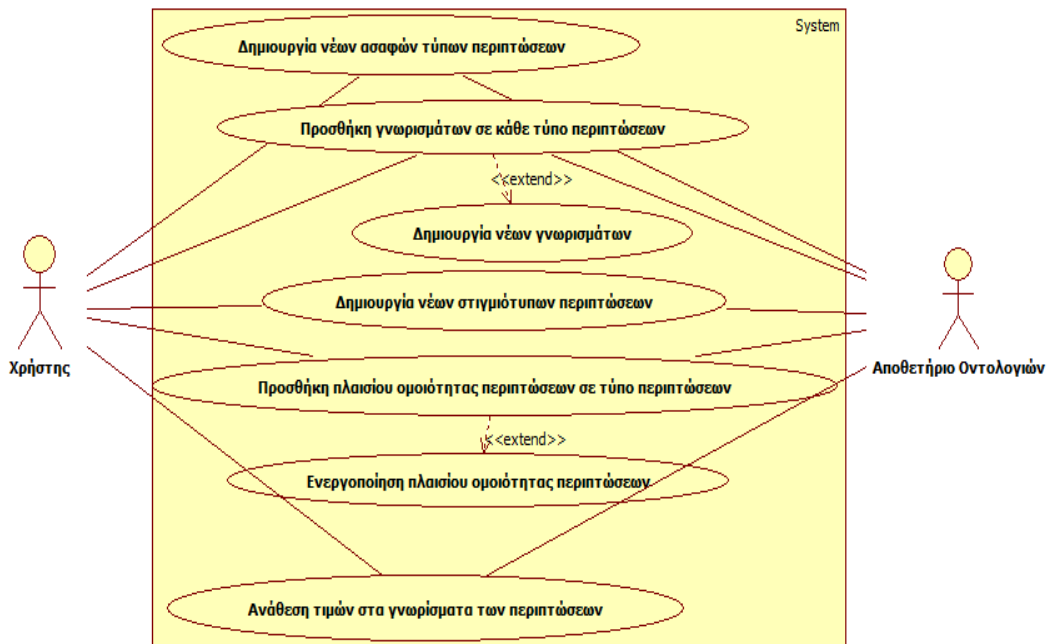
Σχήμα 4.1 Διάγραμμα δραστηριοτήτων για τη κατασκευή σχήματος περιπτώσεων

Επομένως, για την υποστήριξη των ενεργειών που απαιτούνται για το σωστό ορισμό σχήματος περιπτώσεων παρέχονται συγκεκριμένες δυνατότητες στο χρήστη από το GUI που αφορούν τον ορισμό του σχήματος των περιπτώσεων με τέτοιο τρόπο ώστε να δύναται να ορίσει νέους τύπους περιπτώσεων (case types) και να αναθέσει σε αυτούς ιδιότητες (attributes) με τη μορφή σχέσεων (relations) και ιδιοτήτων (properties). Τους τύπους περιπτώσεων μπορεί είτε να τους ορίσει εξ αρχής είτε να τους πάρει έτοιμους από την ασαφή οντολογία πεδίου.

Οι σχέσεις μπορεί να είναι ένα υποσύνολο των ασαφών συσχετιστικών και ιεραρχικών σχέσεων που έχουν οριστεί στην ασαφή οντολογία πεδίου αλλά και καινούριες τις οποίες τις ορίζει ως σχέσεις-γνωρίσματα των τύπων περιπτώσεων. Οι ιδιότητες είναι ασαφείς ιδιότητες ή ασαφείς αποτιμώμενες ιδιότητες ορισμένες και αυτές στην ασαφή οντολογία πεδίου ή καινούριες ορισμένες στους τύπους περιπτώσεων. Οι σχέσεις και οι ιδιότητες αυτές (εκτός φυσικά από τις ασαφώς αποτιμώμενες ιδιότητες) θα μπορούν να είναι είτε ασαφείς (fuzzy) είτε μη ασαφείς (crisp).

Σε κάθε τύπο περιπτώσεων μπορούν να προστεθούν στιγμιότυπα αυτών (case instances) και έτσι για το κάθε ένα στιγμιότυπο να τεθούν οι τιμές για τις σχέσεις και για τις ιδιότητές του. Τέλος για κάθε τύπο περιπτώσεων υπάρχει η δυνατότητα προσθήκης πλαισίων ανάκτησης περιπτώσεων (case retrieval context) με στόχο των υπολογισμό της ομοιότητας, ενώ μόνο ένα από αυτά μπορεί να ενεργοποιηθεί από το χρήστη. Η συγκεκριμένη περίπτωση χρήσης

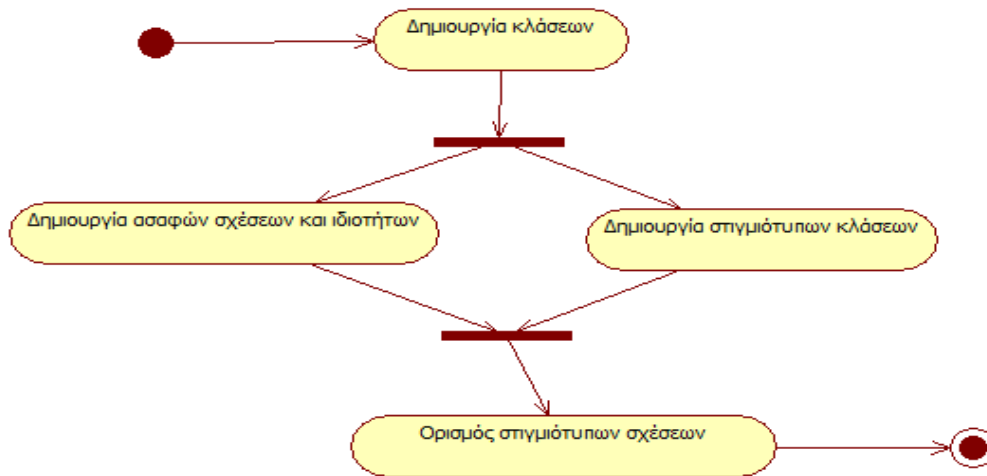
περιγράφεται στο παρακάτω Σχήμα 4.2 που απεικονίζει ένα διάγραμμα χρήσης της UML (use case diagram).



Σχήμα 4.2 Διάγραμμα χρήσης για τη μοντελοποίηση σχήματος περιπτώσεων

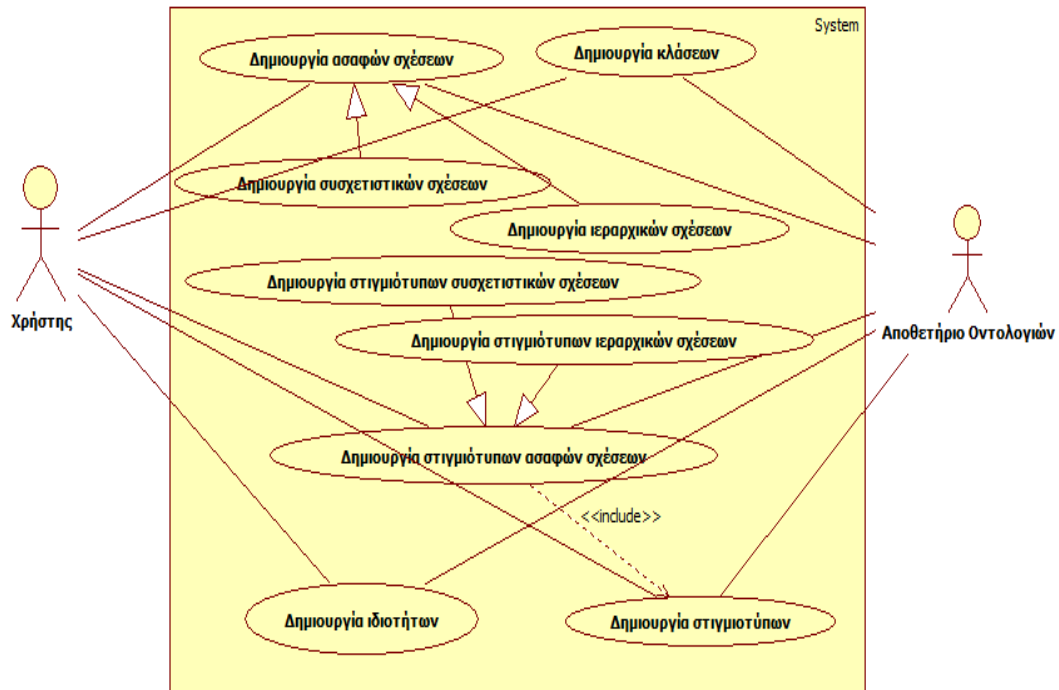
4.1.2.2 Κατασκευή και Διαχείριση Ασαφούς Οντολογίας

Για τη σωστή διαδικασία ορισμού τώρα της ασαφούς οντολογίας όπως αυτό ορίστηκε στην παράγραφο 3.1.2 ο χρήστης αρχικά θα πρέπει να δημιουργήσει ένα σύνολο από έννοιες-κλάσεις. Στη συνέχεια δημιουργεί τις ασαφείς σχέσεις (ιεραρχικές ή συσχετιστικές, ασαφείς ή μη) και τις ασαφείς ιδιότητες, ενώ ταυτόχρονα προσθέτει σε κάθε κλάση στιγμιότυπα. Κατά τη δημιουργία των σχέσεων και των ιδιοτήτων οφείλει να ορίσει φυσικά και τα πεδία ορισμού και τιμών τους. Αφού τελειώσουν οι παραπάνω δύο ενέργειες πρέπει να δημιουργήσει τα στιγμιότυπα των σχέσεων, συσχετίζοντας δηλαδή τα στιγμιότυπα των κλάσεων με βαθμούς ασάφειας. Οι ενέργειες αυτές και η σωστή αλληλουχία τους φαίνονται στο διάγραμμα δραστηριοτήτων (activity diagram) του σχήματος 4.3.



Σχήμα 4.3 Διάγραμμα δραστηριοτήτων για τη κατασκευή ασαφούς οντολογίας

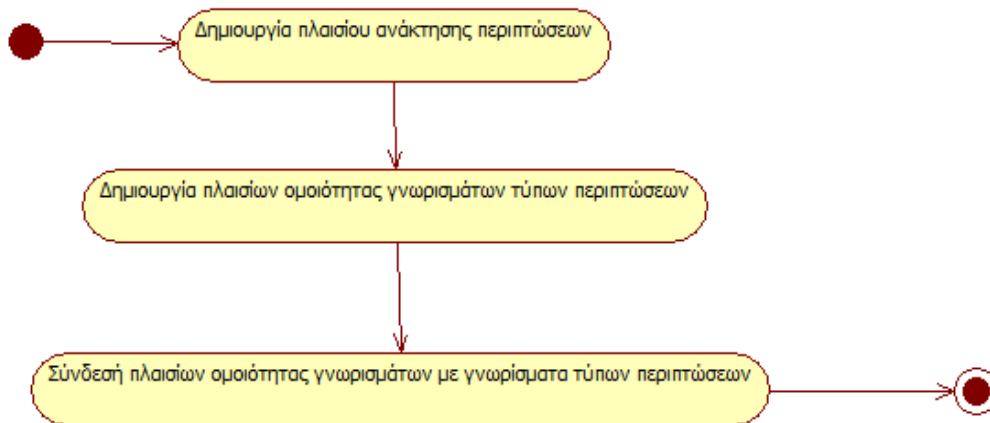
Έτσι για την υποστήριξη των απαραίτητων ενεργειών για την κατασκευή ασαφούς οντολογίας παρέχονται στο χρήστη συγκεκριμένες δυνατότητες που συντελούν στην παραπάνω κατεύθυνση. Μπορεί έτσι όχι μόνο να δημιουργήσει κλάσεις-έννοιες αλλά και να τις συσχετίσει με άλλες κλάσεις και με τύπους δεδομένων χρησιμοποιώντας σχέσεις (relations) και ιδιότητες (properties) αντίστοιχα. Οι σχέσεις και οι ιδιότητες αυτές θα μπορούν να είναι είτε ασαφείς (fuzzy) είτε μη ασαφείς (crisp), ενώ οι ασαφείς σχέσεις μπορεί να είναι ιεραρχικές ή συσχετιστικές. Επίσης, παρέχεται η δυνατότητα για ορισμό ατόμων-στιγμιότυπων (individuals-instances) για κάθε κλάση και έπειτα για συσχέτιση των ατόμων μέσω των σχέσεων ή των ιδιοτήτων. Στην περίπτωση των ασαφών σχέσεων, ο χρήστης ορίζει είτε ιεραρχικές (hierarchical) είτε συσχετιστικές (associative) ασαφείς σχέσεις συνδέοντας με τον τρόπο αυτό δύο άτομα της οντολογίας με κάποιο βαθμό συγγενείας-ασάφειας. Για τη σύνδεση δύο ατόμων-στιγμιότυπων ή ενός ατόμου με τιμή από κάποιο πεδίο τιμών, δηλαδή για τη δημιουργία στιγμιότυπων σχέσεων και ιδιοτήτων, απαιτείται φυσικά πρώτα να έχουν δημιουργηθεί τα στιγμιότυπα, ενέργεια που υποστηρίζεται από το GUI. Η συγκεκριμένη περίπτωση χρήσης περιγράφεται συνολικά στο διάγραμμα χρήσης του Σχήματος 4.4.



Σχήμα 4.4 Διάγραμμα χρήσης για τη μοντελοποίηση ασαφούς οντολογίας πεδίου

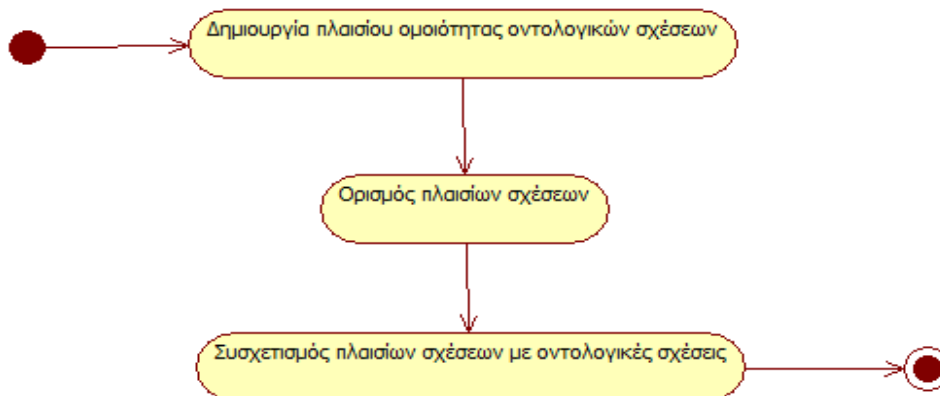
4.1.2.3 Παραμετροποίηση Πλαισίου Ανάκτησης Περιπτώσεων

Για την αποτελεσματική απόδοση πλαισίου με στόχο την ανάκτηση των περιπτώσεων ο χρήστης, όπως ορίστηκε στην παράγραφο 3.2.4, οφείλει να δημιουργήσει αρχικά πλαίσια ανάκτησης περιπτώσεων (case retrieval contexts). Σε αυτά πρέπει να δημιουργήσει πλαίσια ομοιότητας των γνωρισμάτων των τύπων περιπτώσεων (case attribute similarity context) συνδέοντάς τα με τα ανάλογα γνωρίσματα που έχουν οριστεί για τους τύπους περιπτώσεων στους οποίους θα ανατεθεί το συγκεκριμένο πλαίσιο ανάκτησης περιπτώσεων, ενώ οφείλει να δηλώσει τη συνάρτηση συνάθροισης ομοιότητας για το συγκεκριμένο πλαίσιο ανάκτησης περιπτώσεων. Η αλληλουχία ενεργειών αυτή φαίνεται στο παρακάτω διάγραμμα δραστηριοτήτων του σχήματος 4.5.



Σχήμα 4.5 Διάγραμμα δραστηριοτήτων για την κατασκευή πλαισίου ανάκτησης περιπτώσεων

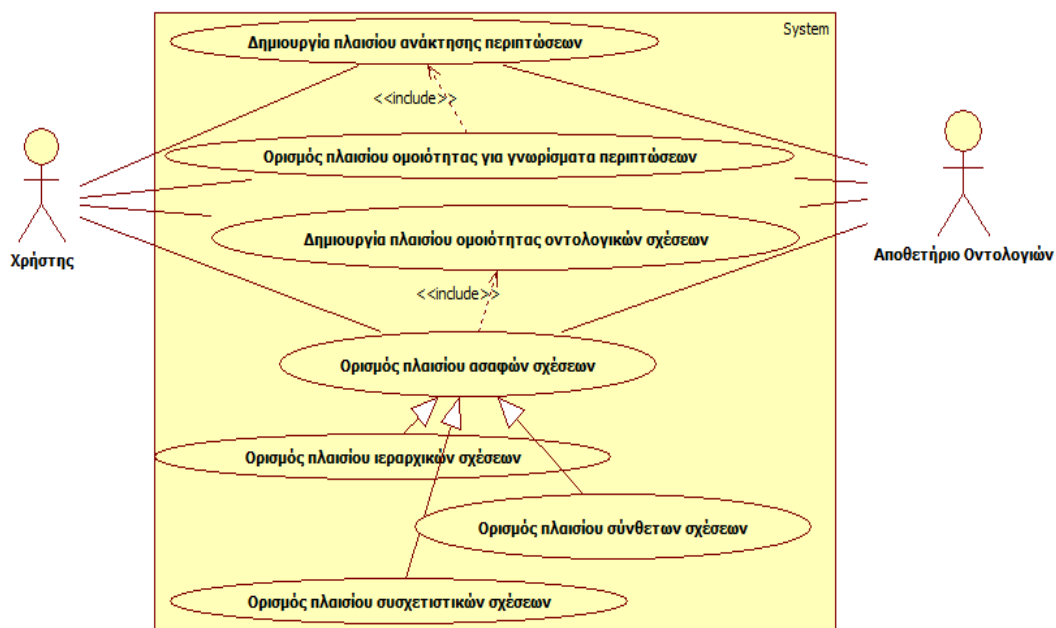
Πλαίσια ομοιότητας ανατίθενται και στις οντολογικές σχέσεις σύμφωνα με την παράγραφο 3.2.4 με στόχο την εύρεση ομοιότητας στιγμιότυπων που συνδέονται με τις σχέσεις αυτές. Για το συγκεκριμένο πλαίσιο ο χρήστης πρέπει να δημιουργήσει ένα πλαίσιο ομοιότητας οντολογικών σχέσεων (ontology similarity context) στο οποίο θα ορίσει πλαίσια ιεραρχικών, συσχετιστικών και σύνθετων σχέσεων. Τα τελευταία πρέπει να συσχετισθούν με τις ανάλογες σχέσεις από την ασαφή οντολογία πεδίου. Όλα αυτά φαίνονται στο σχήμα 4.6.



Σχήμα 4.6 Διάγραμμα δραστηριοτήτων για την κατασκευή πλαισίου ομοιότητας οντολογικών σχέσεων

Για τους παραπάνω σκοπούς παρέχονται συγκεκριμένες λειτουργίες από το GUI που αφορούν τον ορισμό πλαισίου ανάκτησης περιπτώσεων και πλαισίου ομοιότητας οντολογικών σχέσεων. Για παράδειγμα ο χρήστης δημιουργεί όσα πλαίσια ανάκτησης περιπτώσεων (Case Retrieval Contexts) επιθυμεί στα οποία αναθέτει πλαίσια ομοιότητας για

τα γνωρίσματα περιπτώσεων (Case Attribute Similarity Contexts). Τα τελευταία πρέπει φυσικά να τα συνδέσει με κάποιο γνώρισμα-χαρακτηριστικό τύπου περιπτώσεων (Case Attribute) για να αποκτήσουν αυτά σημασιολογικό πλαίσιο. Επίσης για το σχήμα πλαισίου οντολογικών σχέσεων μπορεί να δημιουργήσει πλαίσια ομοιότητας οντολογικών σχέσεων (ontology similarity contexts) στα οποία αναθέτει τα ανάλογα πλαίσια των ιεραρχικών, συσχετιστικών και σύνθετων σχέσεων (fuzzy hierarchical και associative, και composite relation context). Στα τελευταία πλαίσια ο χρήστης ορίζει τιμές σε πεδία που αναλύθηκαν στην παράγραφο 3.2.3, όπως είναι τα farsc και frarsc, αφού φυσικά συσχετίσει τις συγκεκριμένες σχέσεις με αυτές που υπάρχουν στην ασαφή οντολογία πεδίου. Η περίπτωση χρήσης αυτή περιγράφεται στο UML use case diagram του Σχήματος 4.7.



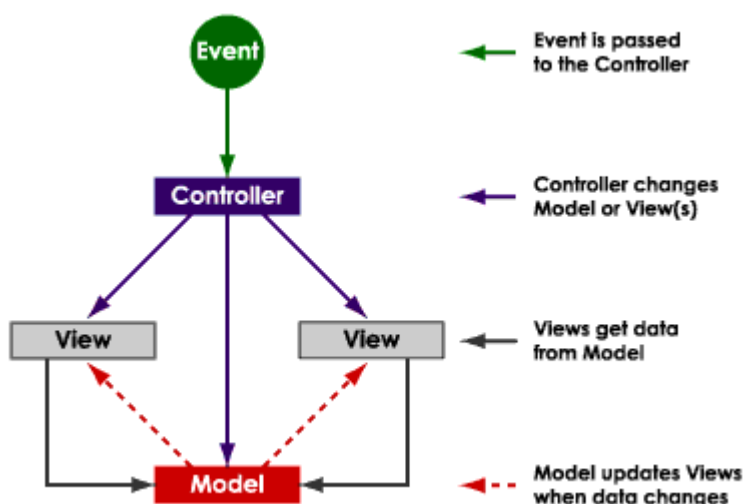
Σχήμα 4.7 Διάγραμμα χρήσης για τη μοντελοποίηση πλαισίου περιπτώσεων και οντολογικών σχέσεων

4.2 Αρχιτεκτονική

Η αρχιτεκτονική που χρησιμοποιήθηκε για το σχεδιασμό και την υλοποίηση του ολικού συστήματος ήταν η τεχνοτροπία *Μοντέλο-Παρουσίαση-Ελεγχος* (Model-View-Controller ή MVC), που θεωρείται ένα από τα αρχιτεκτονικά πρότυπα για παραγωγή λογισμικού. Σαν ολικό σύστημα θεωρείται το ολικό πλαίσιο που χρησιμοποιείται από τα χαμηλότερο επίπεδο έως το υψηλότερο επίπεδο με στόχο την αναπαράσταση και ανάκτηση ασαφούς γνώσης.

Το μοντέλο MVC χρησιμοποιείται ευρέως προσφέροντας δυναμική αρχιτεκτονική δομή ειδικά σε περιπτώσεις σχεδίασης γραφικού περιβάλλοντος για εφαρμογές βασισμένες στο διαδίκτυο (web-based applications), αφού προσφέρει τη δυνατότητα για απομόνωση του πεδίου των δεδομένων της εφαρμογής από τη διεπιφάνεια χρήστη (είσοδος και παρουσίαση) επιτρέποντας ανεξάρτητη ανάπτυξη, έλεγχο και συντήρηση των παραπάνω συστατικών. Χρησιμοποιείται επίσης και για εφαρμογές που ενδέχεται να χρειαστούν διαφορετικές και πολλαπλές αναπαραστάσεις των ίδιων δεδομένων.

Το μοντέλο MVC σχεδιάστηκε αρχικά με στόχο την αντιστοίχιση των παραδοσιακών ρόλων ενός συστήματος, δηλαδή της εισόδου (input), της επεξεργασίας (processing) και της εξόδου (output), σε ανάλογες λειτουργίες και διαδικασίες ενός γραφικού περιβάλλοντος, οι οποίες διακρίνονται στο Σχήμα 4.8.



Σχήμα 4.8 Μοντέλο MVC

Η λειτουργία Controller του MVC δέχεται και ερμηνεύει την είσοδο είτε από το ποντίκι είτε από το πληκτρολόγιο, η οποία παρουσιάζεται με τη μορφή συμβάντων (events) και αντιστοιχίζει στη συνέχεια την είσοδο αυτή σε εντολές που στέλνονται είτε στη λειτουργία View είτε στη Model. Η πλευρά του Model τώρα διαχειρίζεται δεδομένα και αποκρίνεται σε ερωτήματα που αφορούν στην κατάσταση των δεδομένων και σε οδηγίες με σκοπό την αλλαγή της κατάστασης αυτής. Η πλευρά του View είναι υπεύθυνη για την απεικόνιση στο χρήστη των δεδομένων που δέχεται από το Model με συνδυασμό γραφικών και κειμένου. Οι μονάδες παρουσίασης ή αλλιώς όψεις, στην περίπτωση μίας web εφαρμογής, είναι κάθε html σελίδα που παρέχεται στο χρήστη. Αυτή διαμορφώνεται κάθε φορά με την είσοδο από το χρήστη και αντλεί πληροφορίες απευθείας από τα μοντέλα που έχουν δημιουργηθεί, δίνοντας

πάντα μια έγκυρη εικόνα των δεδομένων. Οι όψεις αποτελούν ουσιαστικά τη διεπαφή του συστήματος με το χρήστη.

Είναι φανερό ότι το παραπάνω μοντέλο αρχιτεκτονικού σχεδιασμού χρησιμοποιείται ευρέως για εφαρμογές που δέχονται πολλαπλά συμβάντα μέσω εισόδου από το χρήστη με στόχο την εμφάνιση των δεδομένων και την επεξεργασία ή την αλλαγή των δεδομένων ενός συστήματος. Τα δεδομένα και οι όψεις των συστημάτων αυτών μπορεί να βρίσκονται σε έναν εξυπηρετητή ιστού (web server). Πρόκειται δηλαδή για διαδικτυακές εφαρμογές που στηρίζονται στη συνεχή δημιουργία συμβάντων.

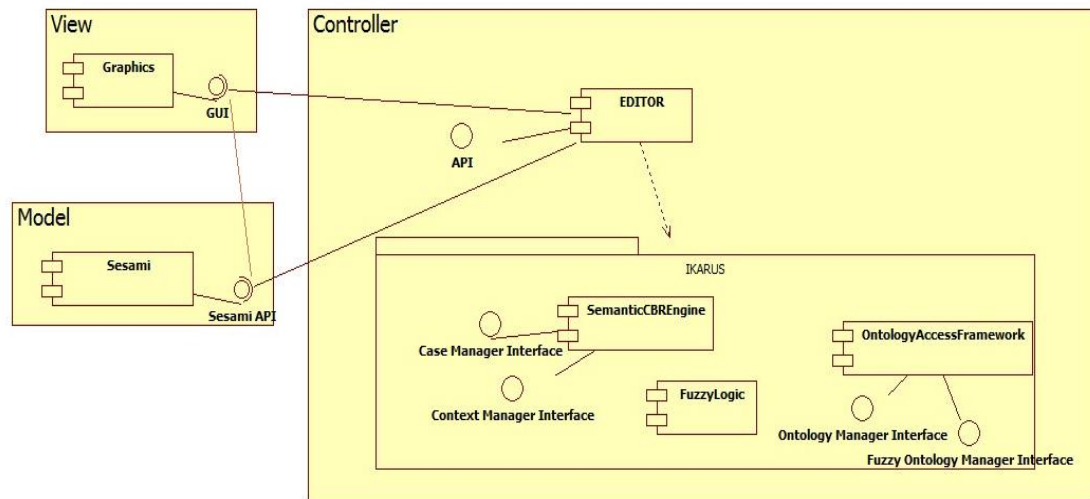
Βασισμένο στο διαδίκτυο είναι και το μοντέλο IKARUS-CBR καθώς τα δεδομένα της εφαρμογής βρίσκονται σε απομακρυσμένους εξυπηρετητές, ενώ ο χρήστης χρειάζεται να παρακολουθεί μια όψη του συστήματος και να έχει τη δυνατότητα να επέμβει στην κατάσταση των δεδομένων. Επομένως, κρίθηκε επιτακτική η ανάγκη υιοθέτησης του MVC μοντέλου για το σχεδιασμό και την υλοποίηση του γραφικού περιβάλλοντος που όχι μόνο θα απεικονίζει την ασαφής οντολογία πεδίου, τις περιπτώσεις και το Context αλλά και θα παρέχει στο χρήστη λειτουργίες διαμόρφωσης των παραπάνω δεδομένων.

Έτσι, στο πιο χαμηλό επίπεδο που ταυτίζεται με την πλευρά Model συναντάται το σύστημα αναπαράστασης των οντολογιών βάσει των μοντέλων που ορίστηκαν στα κεφάλαια 3.1.2, 3.1.3 και 3.2. Τα δεδομένα των οντολογιών αποθηκεύονται σε κάποιον απομακρυσμένο εξυπηρετητή (web server) με τη μορφή τριάδων με το κάθε οντολογικό στοιχείο της τριάδας να συνοδεύεται από ένα Uri (Uniform Resource Identifier). Ο χρήστης αλληλεπιδρά με τη βάση οντολογιών αυτή μέσω της γραφικής διεπιφάνειας χρήστη (GUI) μέσω του διαδικτύου και του πρωτοκόλλου HTTP. Το GUI εξαρτάται απόλυτα από τα σχήματα των οντολογιών καθώς οποιαδήποτε αλλαγή στη δομή τους επηρεάζει άμεσα και τους άξονες σχεδίασης του GUI.

Στα πλαίσια της υποστήριξης της δομής των συγκεκριμένων οντολογιών και ορισμού μεθόδων για τη διαχείριση και αλλαγή τους δημιουργήθηκε το API (Application Programming Interface) του IKARUS-CBR γραμμένο σε γλώσσα JAVA. Το API τώρα σαν σύστημα αποτελείται από τα προγραμματιστικά πακέτα SemanticCBREngine, OntologyAccessFramework και FuzzyLogicAPI. Το πρώτο υλοποιεί τα interfaces CaseManager και ContextManager, ενώ το δεύτερο τα OntologyManager και FuzzyOntologyManager. Μεθόδους από τα παραπάνω interfaces χρησιμοποιεί το GUI που

σχεδιάστηκε με το ZK framework. Τα παραπάνω πακέτα σε συνδυασμό με τις μεθόδους που έχουν γραφεί για τη διαχείριση των συμβάντων από την είσοδο του χρήστη (API) αποτελούν την πλευρά Controller.

Στο Σχήμα 4.9 διακρίνονται καθαρά οι διαφορετικές πλευρές του View, Controller και Model που περιγράφηκαν παραπάνω. Στην πλευρά View υπάρχει το GUI που επικοινωνεί με τον Controller και στο οποίο λαμβάνουν μέρος τα συμβάντα.



Σχήμα 4.9 Αρχιτεκτονικό Μοντέλο Συστήματος

4.2.1 Πλευρά Model

Σαν πεδίο Model θεωρείται ο τρόπος με τον οποίο δομείται το σύστημα IKARUS-CBR, δηλαδή οι μέθοδοι που χρησιμοποιούνται για την αναπαράσταση των δεδομένων του συστήματος. Οι μέθοδοι αυτοί έχουν να κάνουν με το σχήμα οντολογιών του συστήματος που αποθηκεύονται στο sesame ontology repository, που αφορά ένα διαδικτυακό αποθετήριο οντολογιών. Το σχήμα αυτό αναπτύχθηκε στο 3^ο κεφάλαιο. Σκοπός της παρούσας διπλωματικής αποτέλεσαν αποκλειστικά η διαμόρφωση της πλευράς View και η δημιουργία μεθόδων στην πλευρά Controller για το χειρισμό των συμβάντων.

4.2.2 Πλευρά View

Σύμφωνα με τις τρεις βασικές περιπτώσεις χρήσης του GUI ο χρήστης μπορεί να ορίσει μια ασαφής οντολογία πεδίου, μια οντολογία ασαφών περιπτώσεων και να αναθέσει σημασιολογικό πλαίσιο είτε στις οντολογικές σχέσεις είτε στα γνωρίσματα των περιπτώσεων με απώτερο στόχο την αποτελεσματικότερη ανάκτηση γνώσης. Για να διεξάγονται ομαλά οι

τρεις παραπάνω λειτουργίες που δε διαθέτουν κανένα κοινό χαρακτηριστικό χρειάζεται να υπάρχει μια σαφής διάκριση των παραπάνω λειτουργιών που επιτελεί το GUI. Για το λόγο αυτό σχεδιάστηκαν τρεις διαφορετικές όψεις του συστήματος με τη μορφή των καρτελών (tabs), οι οποίες είναι ο *Ontology Manager* για την κατασκευή και διαχείριση μιας ασαφούς οντολογίας, ο *Case Manager* για τον ορισμό του σχήματος ασαφών περιπτώσεων και ο *Context Manager* για την απόδοση πλαισίου στις περιπτώσεις και στις οντολογικές σχέσεις.

Στο σημείο αυτό αξίζει να σημειωθεί ότι έχοντας ως κύριο γνώμονα την ταχύτατη εξοικείωση του χρήστη με το γραφικό περιβάλλον υιοθετήθηκε ένα συγκεκριμένο σχεδιαστικό μοτίβο στην παρουσίαση των βασικών στοιχείων του συστήματος, όπως είναι οι κλάσεις, οι περιπτώσεις, τα στιγμιότυπα κλάσεων, κ.α. Σύμφωνα με το μοτίβο αυτό, στα αριστερά της όψης της εφαρμογής απαριθμούνται τα εκάστοτε στοιχεία που εξερευνεί ο χρήστης προσφέροντάς του επιλογές για προσθήκη κάποιου νέου στοιχείου και διαγραφή ενός ήδη υπάρχοντος. Στα δεξιά της διεπιφάνειας εμφανίζονται οι πληροφορίες για το στοιχείο που θα επιλεγεί με το χτύπημα του ποντικιού.

Για την καλύτερη κατανόηση των λειτουργιών που επιτελούνται στις επιμέρους καρτέλες εμφανίζονται screenshots από το GUI.

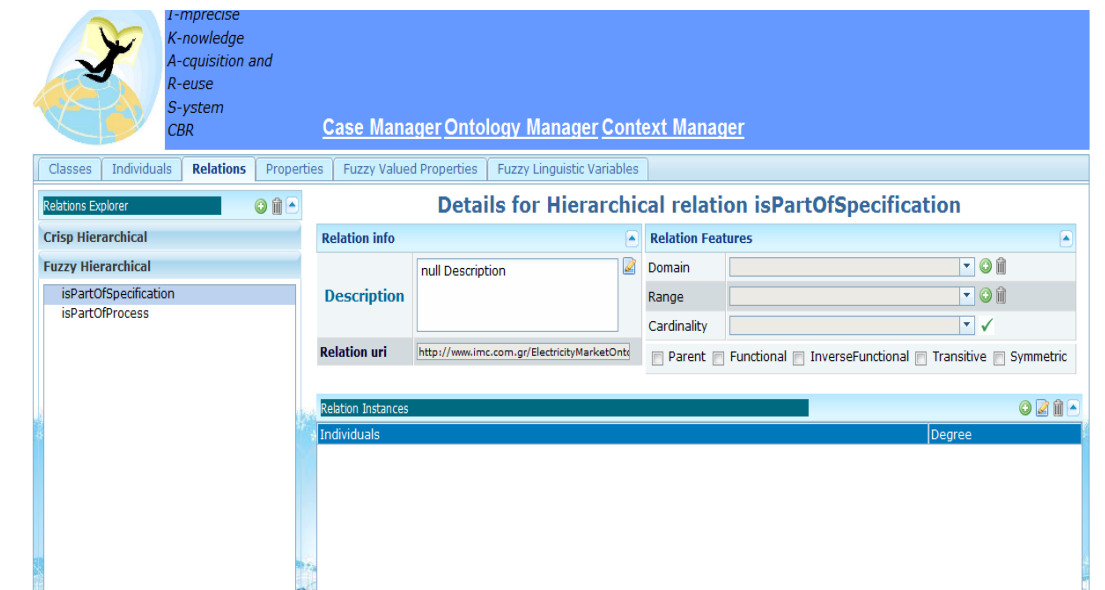
4.2.2.1 *Ontology Manager*

Σε αυτήν την καρτέλα συντελείται ο ορισμός της ασαφούς οντολογίας πεδίου. Λόγω των διαφορετικών λειτουργιών που απαιτούνται για το συγκεκριμένο ορισμό σχεδιάστηκαν και σε αυτό το σημείο έξι διαφορετικές όψεις του συστήματος με τη μορφή των καρτελών (tabs), οι οποίες είναι:

- Καρτέλα Classes: Στην πρώτη καρτέλα παρουσιάζονται οι κλάσεις-έννοιες της οντολογίας μαζί με ορισμένες σημαντικές πληροφορίες για αυτές, όπως μια σύντομη περιγραφή τους, τα αξιώματα και οι κανόνες τους, οι σχέσεις, οι ιδιότητες και οι ασαφώς αποτιμώμενες ιδιότητες οι οποίες έχουν ως πεδίο ορισμού αυτές. Στην καρτέλα αυτήν παρέχεται η δυνατότητα στο χρήστη όχι μόνο να ορίσει μια νέα κλάση αναθέτοντάς της όσες πληροφορίες από τις παραπάνω αυτός επιθυμεί αλλά και να διαγράψει από την οντολογία μια ήδη υπάρχουσα κλάση. Είναι σημαντικό να τονιστεί ότι για να δει ο χρήστης τα παραπάνω χαρακτηριστικά μιας κλάσης θα πρέπει να την επιλέξει με το ποντίκι. Βασική δυνατότητα επίσης που προσφέρεται στο χρήστη αποτελεί η προσθήκη μιας σχέσης ή ιδιότητας η οποία είτε υπάρχει ήδη ορισμένη στην ασαφή οντολογία πεδίου είτε θα δημιουργηθεί εκ του μηδενός από

αυτόν. Με την επιλογή μιας σχέσης ή ιδιότητας εμφανίζονται ορισμένες πληροφορίες για αυτήν, όπως μια σύντομη περιγραφή της και το πεδίο τιμών και ορισμού της.

- Καρτέλα Individuals: Σε αυτήν την καρτέλα ο χρήστης περιηγείται στα άτομο-στιγμιότυπα που διαθέτουν οι κλάσεις, ενώ δύναται όχι μόνο να δημιουργήσει καινούρια άτομα για κάθε κλάση αλλά και να διαγράψει ήδη ορισμένα άτομα. Για την προσθήκη ενός ατόμου σε μια κλάση απαιτείται πρώτα φυσικά να επιλεγεί η κλάση στην οποία θα προστεθεί το νέο στιγμιότυπο.
- Καρτέλα Relations (Εικόνα 4.1): Αρχικά παρουσιάζονται στο χρήστη οι ασαφείς σχέσεις που περιέχονται στην ασαφή οντολογία πεδίου που μπορεί να είναι ασαφείς ή μη ιεραρχικές και ασαφείς ή μη συσχετιστικές. Με την επιλογή μιας σχέσης εμφανίζονται μια σύντομη περιγραφή της, ορισμένα βασικά χαρακτηριστικά της όπως το πεδίο ορισμού και πεδίου τιμών της και ορισμένες ιδιότητες που αυτή μπορεί να έχει όπως μεταβατική (transitive), συμμετρική (symmetric), μονότιμη (functional) και αντίστροφα μονότιμη (inverse functional).
- Στο σημείο αυτό συναντάται ένα σημαντικό σημείο συμβολής της διπλωματικής που έχει να κάνει με την αναπαράσταση και ορισμό στιγμιότυπων των ασαφών σχέσεων (relation instances). Αν πρόκειται για ασαφή ιεραρχική σχέση τότε ο χρήστης για να δημιουργήσει μια νέα σχέση-στιγμιότυπο πρέπει να επιλέξει από μια λίστα ένα ήδη ορισμένο άτομο-στιγμιότυπο της κλάσης που αποτελεί πεδίο ορισμού και τιμών της σχέσης και να ορίσει για τη σχέση αυτή το βαθμό βεβαιότητας-συμμετοχής που όπως έχει ήδη αναφερθεί παίρνει τιμές στο διάστημα $[0, 1]$. Στην περίπτωση μιας νέας ασαφούς συσχετιστικής σχέσης ο χρήστης θα δηλώσει δύο άτομα, το ένα από μια λίστα με άτομα που ανήκουν σε κλάσεις από το πεδίο ορισμού της σχέσης και το άλλο από μια άλλη λίστα με άτομα από το πεδίο τιμών, ενώ οφείλει να δηλωθεί επίσης ο βαθμός συμμετοχής. Τέλος παρέχεται φυσικά η δυνατότητα στο χρήστη όχι μόνο να διαγράψει μία σχέση-στιγμιότυπο αλλά και να αλλάξει το βαθμό ασάφειας-συμμετοχής μιας σχέσης αφού πρώτα την επιλέξει.



Εικόνα 4.1 Καρτέλα Relations του Ontology Manager

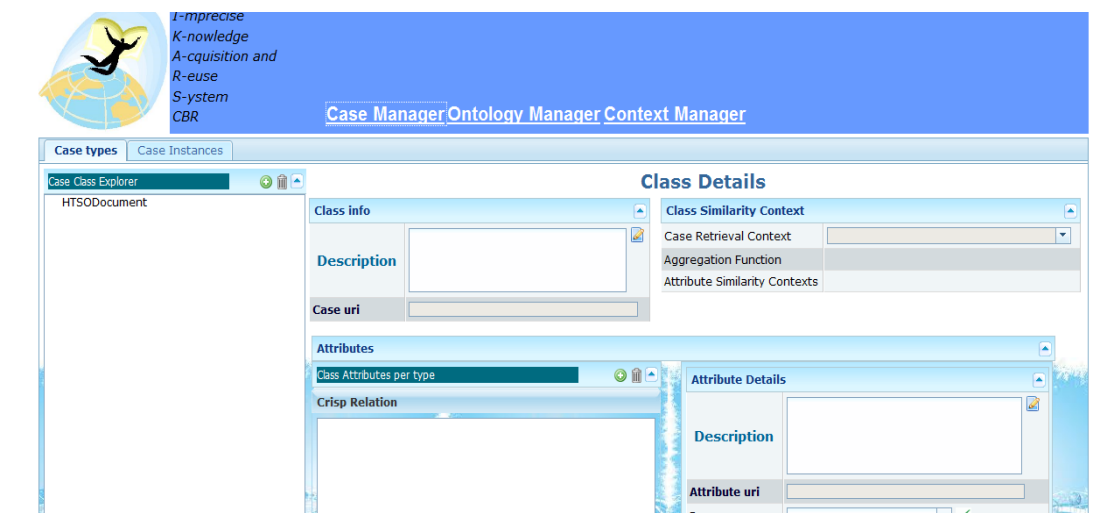
- **Καρτέλα Properties:** Στην καρτέλα αυτήν παρουσιάζονται στο χρήστη οι ασαφείς ιδιότητες που περιέχονται στην ασαφή οντολογία πεδίου που μπορεί να είναι ασαφείς και μη ασαφείς (fuzzy και crisp). Με την επιλογή μιας ιδιότητας εμφανίζονται μια σύντομη περιγραφή της, ορισμένα βασικά χαρακτηριστικά της όπως το πεδίο ορισμού και πεδίου τιμών της και ορισμένες ιδιότητες που αυτή μπορεί να έχει όπως μονότιμη (functional). Το πεδίο τιμών μιας ιδιότητας δεν είναι φυσικά κλάση αλλά μπορεί να είναι είτε ακέραιος (int) είτε πραγματικός (float – αριθμός κινητής υποδιαστολής) ή αλφαριθμητικό (string).

4.2.2.2 Case Manager

Σε αυτήν την καρτέλα ο χρήστης ορίζει τη δομή και το σχήμα που θέλει να δώσει στην οντολογία ασαφών περιπτώσεων. Για να επιτευχθεί με σωστό και αποτελεσματικό τρόπο ο συγκεκριμένος ορισμός, ο οποίος τρόπος έχει ήδη αναφερθεί στην ενότητα 3.1.3, δημιουργήθηκαν οι εξής δύο καρτέλες:

- **Καρτέλα Case Types (Εικόνα 4.2):** Στη συγκεκριμένη καρτέλα ο χρήστης περιηγείται στους τύπους ασαφών περιπτώσεων που έχουν ήδη οριστεί, ενώ μπορεί να δημιουργήσει αλλά και να διαγράψει ένα σύνολο από τους τύπους αυτούς. Στα πλαίσια της δημιουργίας ενός τύπου περίπτωσης παρέχεται στο χρήστη και η επιλογή για προσθήκη νέου τύπου από τις κλάσεις που έχουν ήδη οριστεί στην ασαφή οντολογία πεδίου. Επίσης, του παρουσιάζονται οι ιδιότητες-χαρακτηριστικά που διαθέτουν οι ασαφείς τύποι περιπτώσεων, αφού βέβαια πρώτα επιλέξει με το χτύπημα του ποντικιού τον τύπο περίπτωσης που τον ενδιαφέρει. Οι ιδιότητες αυτές μπορεί να

είναι: ασαφείς και μη σχέσεις (fuzzy και crisp relations), ασαφείς και μη ιδιότητες (fuzzy και crisp properties) και ασαφώς αποτιμώμενες ιδιότητες (fuzzy valued properties). Στην περίπτωση της επιλογής ενός τύπου θα εμφανιστούν και ορισμένες βασικές πληροφορίες για τον επιλεγμένο τύπο περιπτώσεων, όπως είναι μια σύντομη περιγραφή του σε απλό κείμενο, το uri του και το πλαίσιο ανάκτησης περιπτώσεων (case retrieval context) το οποίο του έχει δοθεί με στόχο την ανάκτηση, ενώ αν δεν του έχει δοθεί ακόμα ο χρήστης μπορεί να του το αναθέσει μέσα φυσικά από τη συγκεκριμένη καρτέλα. Όσον αφορά τις ιδιότητες, ο χρήστης αναθέτει στους τύπους περιπτώσεων καινούριες ιδιότητες, τις οποίες τις ορίζει ο ίδιος εκ του μηδενός. Με την επιλογή μιας συγκεκριμένης ιδιότητας εμφανίζονται ορισμένες βασικές λεπτομέρειές της, οι οποίες έχουν να κάνουν με απλές πληροφορίες για την περιγραφή της.



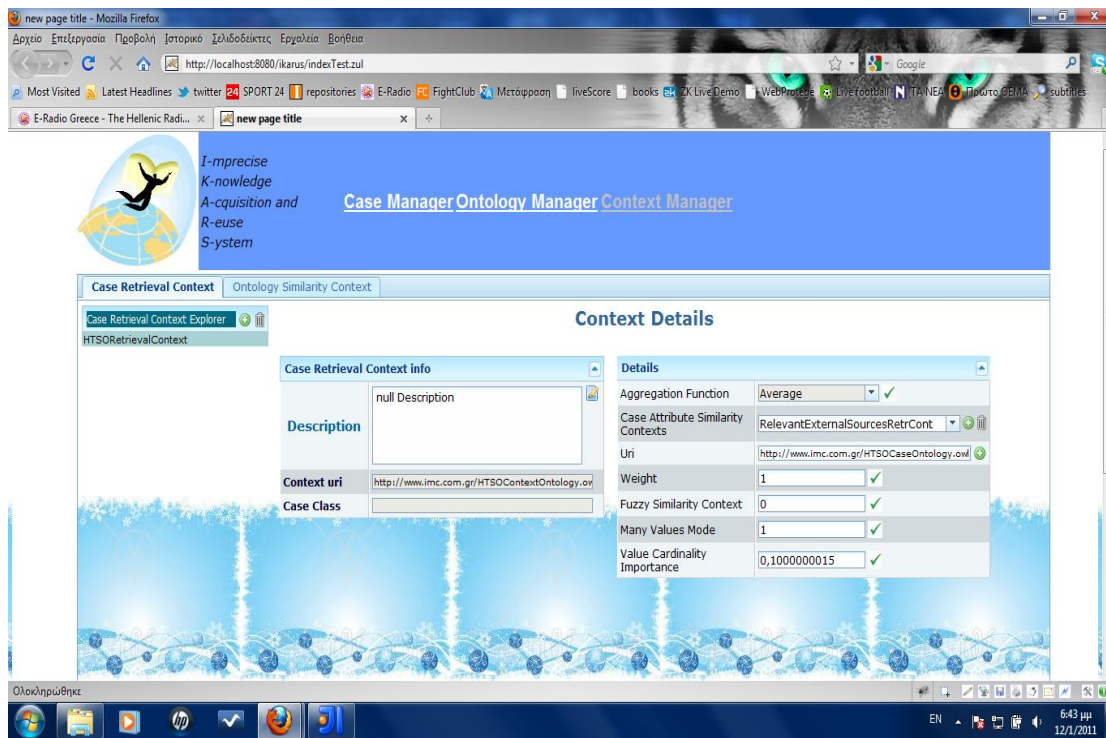
Εικόνα 4.2 Καρτέλα Case Types του Case Manager

- Καρτέλα Case Instances: Εδώ γίνεται η προσθήκη από το χρήστη περιπτώσεων, δηλαδή στιγμιότυπα των ασαφών τύπων περιπτώσεων. Έτσι δημιουργείται μια λίστα από περιπτώσεις που έχουν ήδη οριστεί. Με την επιλογή ενός τύπου περιπτώσεων θα εμφανιστούν οι περιπτώσεις-στιγμιότυπα του συγκεκριμένου τύπου, ενώ τώρα με την επιλογή μιας περίπτωσης ο χρήστης θα παρατηρήσει όχι μόνο τις ιδιότητες (attributes) της, οι οποίες φυσικά μπορεί να είναι ασαφείς ή μη σχέσεις και ιδιότητες και ασαφώς αποτιμώμενες ιδιότητες, αλλά και τις ήδη υπάρχουσες τιμές που έχουν ανατεθεί στις ιδιότητες αυτές. Στη συνέχεια του δίνεται η δυνατότητα να προσθέσει μια τιμή σε μια ιδιότητα ανεξάρτητα από το αν αυτή η ιδιότητα έχει ή δεν έχει ήδη τιμές. Οι τιμές που θέτει ο χρήστης ανταποκρίνονται φυσικά στα πεδία τιμών των γνωρισμάτων στα οποία ανατίθενται οι τιμές.

4.2.2.3 Context Manager

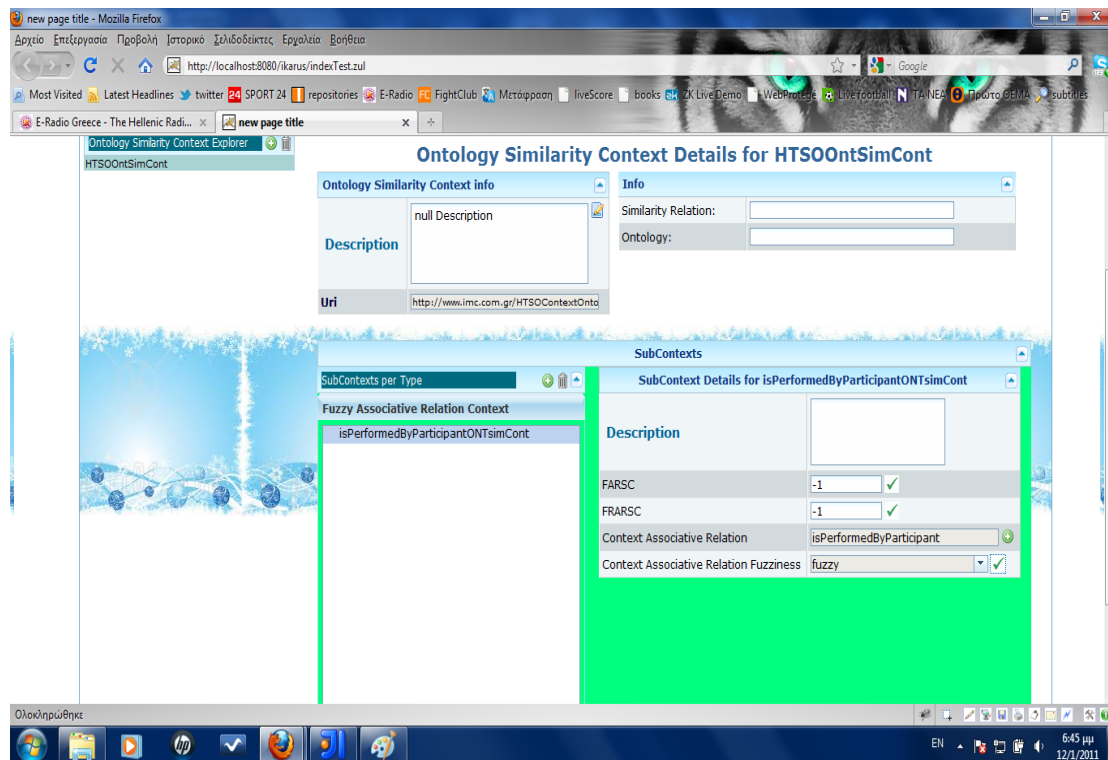
Στην καρτέλα αυτή συντελείται ο ορισμός του νοηματικού πλαισίου για τις περιπτώσεις και τις οντολογικές σχέσεις. Επειδή πρόκειται για δύο εντελώς διαφορετικές διαδικασίες σχεδιάστηκαν δύο διαφορετικές καρτέλες για το χειρισμό των συγκεκριμένων εργασιών:

- Καρτέλα Case Retrieval Context (Εικόνα 4.3): Ο χρήστης δημιουργεί και διαγράφει όσα πλαίσια ανάκτησης περιπτώσεων (case retrieval contexts) αυτός επιθυμεί τα οποία εμφανίζονται σε μορφή λίστας στην οθόνη. Αφού επιλέξει ένα από αυτά, εμφανίζονται κάποιες σημαντικές πληροφορίες του πλαισίου αυτού. Αυτές μπορεί να είναι μια απλή λεκτική περιγραφή του πλαισίου και το uri του τύπου περιπτώσεων στον οποίο έχει ανατεθεί το πλαίσιο. Εμφανίζονται επίσης τα πλαίσια ομοιότητας των γνωρισμάτων των περιπτώσεων (case attribute similarity context) που έχουν δημιουργηθεί για το δεδομένο πλαίσιο ανάκτησης περιπτώσεων, ενώ ο χρήστης μπορεί να προσθέσει όσα case attribute similarity context αυτός επιθυμεί. Για κάθε πλαίσιο ανάκτησης περιπτώσεων ο χρήστης δηλώνει τη συνάρτηση συνάθροισης ομοιότητας, η οποία μπορεί να είναι max, min, average ή Weighted Average. Για κάθε case attribute similarity context ο χρήστης θέτει τιμές που συντελούν στον υπολογισμό της ομοιότητας, ενώ δίνει το πλαίσιο αυτό με ένα γνώρισμα περίπτωση που αυτός επιθυμεί. Οι πιο σημαντικές τιμές από αυτές είναι οι εξής:
 1. Uri: πρόκειται για το uri του γνωρίσματος με το οποίο είναι συνδεδεμένο το πλαίσιο που θα επιλεγεί,
 2. Fuzzy Similarity Context (0 ή 1): πρόκειται για την τιμή f_{pasc} που περιγράφεται στην παράγραφο 3.2.1,
 3. Weight ([0, 1]): πρόκειται για το βάρος που θα λάβει το συγκεκριμένο πλαίσιο στον υπολογισμό της ομοιότητας σε περίπτωση που έχει επιλεγεί συνάρτηση συνάθροισης η Weighted Average.



Εικόνα 4.3 Καρτέλα πλαισίου ανάκτησης περιπτώσεων

- Καρτέλα Ontology Similarity Context** (Εικόνα 4.4): Στα αριστερά της καρτέλας αυτής εμφανίζονται τα πλαίσια ομοιότητας οντολογικών σχέσεων που έχουν ήδη δημιουργηθεί, ενώ στο σημείο αυτό παρέχεται η δυνατότητα στο χρήστη για ορισμό νέων ontology similarity contexts. Με την επιλογή ενός εξ αυτών εμφανίζονται στα δεξιά της οθόνης πληροφορίες που ποικίλλουν από μια απλή περιγραφή έως τα πλαίσια σχέσεων που έχουν ανατεθεί στο συγκεκριμένο ontology similarity context. Τα πλαίσια ομοιότητας των σχέσεων μπορούν να αφορούν ασαφείς ή μη συσχετιστικές και ιεραρχικές και σύνθετες σχέσεις. Με την επιλογή ενός πλαισίου σχέσης εμφανίζονται συγκεκριμένα χαρακτηριστικά του πλαισίου ομοιότητας που αφορούν τα πεδία farsc και frarsc για τις ασαφείς συσχετιστικές σχέσεις, τα πεδία fhrc και fhrc για τις ασαφείς ιεραρχικές σχέσεις και το πεδίο fcrc για τις σύνθετες σχέσεις. Φυσικά στο σημείο αυτό συσχετίζονται τα πλαίσια σχέσεων με συγκεκριμένες ασαφείς σχέσεις της ασαφούς οντολογίας πεδίου με χρήση του πεδίου Context Associative ή Hierarchical Relation, ενώ στο πεδίο Context Associative ή Hierarchical Relation Fuzziness ο χρήστης επιλέγει είτε fuzzy ή crisp που έχει να κάνει για το αν το πλαίσιο συνδέεται με ασαφή ή μη σχέση αντίστοιχα. Όσον αφορά τις σύνθετες σχέσεις ο χρήστης ορίζει τις σχέσεις της ασαφούς οντολογίας που συμμετέχουν στη σύνθεση της σχέσης αυτής και δημιουργεί για κάθε μία από αυτή ένα ξεχωριστό Component Relation Context.



Εικόνα 4.4 Καρτέλα πλαισίου ομοιότητας οντολογικών σχέσεων

4.2.3 Πλευρά Controller

Στην πλευρά αυτή συναντάται το στοιχείο Editor που χωρίζεται στα API και IKARUS. Το API αναφέρεται στις μεθόδους που προσφέρονται για τον ορισμό και το χειρισμό των συμβάντων που λαμβάνουν μέρος στο σύστημα δεχόμενων της εισόδου του χρήστη μέσω του χτυπήματος του ποντικιού και του πληκτρολογίου. Για κάθε γεγονός που χρήζει ιδιαίτερου χειρισμού δημιουργήθηκαν μέθοδοι (συναρτήσεις) γραμμένοι σε γλώσσα προγραμματισμού JAVA με στόχο την αλλαγή της όψης της διεπιφάνειας (View) και την προσθήκη ή αλλαγή δεδομένων στο σύστημα (Model). Για παράδειγμα διακρίνονται μέθοδοι για το χειρισμό της προσθήκης μιας κλάσης ή ασαφούς συσχετιστικής σχέσης στην ασαφή οντολογία πεδίου με το χρήστη να δηλώνει το όνομα και το uri για τα στοιχεία αυτά, μέθοδοι που θα παρουσιαστούν στο επόμενο κεφάλαιο.

Το στοιχείο IKARUS αναφέρεται στο API του πλαισίου IKARUS-CBR. Το API αυτό αποτελείται από τρία προγραμματιστικά πακέτα στα οποία διακρίνονται τέσσερις κύριες προγραμματιστικές διεπιφάνειες (interfaces):

1. Ontology Manager: Περιλαμβάνει μεθόδους που αφορούν όχι μόνο τη διαχείριση της ασαφούς οντολογίας αλλά και την παροχή πληροφοριών για συστατικά του

συστήματος όπως σχόλια για το κάθε στοιχείο. Χειρίζεται επομένως τις κλάσεις, σχέσεις και ιδιότητες της ασαφούς οντολογίας πεδίου παρέχοντας για παράδειγμα μεθόδους για δημιουργία και διαγραφή των παραπάνω στοιχείων από την οντολογία πεδίου.

2. Fuzzy Ontology Manager: Μέθοδοι για την διαχείριση της ασαφούς οντολογίας πεδίου προσφέρονται και από το interface του Fuzzy Ontology Manager, απλά αυτό ασχολείται σε μεγάλο βαθμό με τον ορισμό και τα στιγμιότυπα των ασαφών ιεραρχικών και συσχετιστικών σχέσεων.
3. Case Manager: Προσφέρει μεθόδους για το χειρισμό των τύπων περιπτώσεων και των στιγμιότυπών τους, όπως για παράδειγμα μέθοδοι για τη δημιουργία και διαγραφή των στοιχείων αυτών. Επίσης, ορίζει μεθόδους για τη δημιουργία ιδιοτήτων-γνωρισμάτων (attributes) και ανάθεσή τους στους τύπους περιπτώσεων.
4. Context Manager: Σε αυτό το interface περιλαμβάνονται μέθοδοι για την ανάθεση πλαισίου στα γνωρίσματα των περιπτώσεων και στις οντολογικές σχέσεις. Αυτές οι μέθοδοι θέτουν τιμές σε συγκεκριμένα χαρακτηριστικά των πλαισίων των γνωρισμάτων και σχέσεων, όπως τα πεδία τα πεδία farsc και frarsc για τις ασαφείς συσχετιστικές σχέσεις και τα πεδία fharsc και fharsc με στόχο την πιο αποτελεσματική ανάκτηση των περιπτώσεων.

5

Υλοποίηση

Στο κεφάλαιο αυτό θα αναφερθούν ορισμένες χαρακτηριστικές λεπτομέρειες υλοποίησης του γραφικού περιβάλλοντος με απώτερο σκοπό την κατανόηση των προγραμματιστικών εργαλείων και τους λόγους για του οποίους αυτά χρησιμοποιήθηκαν.

5.1 Τεχνικές λεπτομέρειες

Στο πρώτο μέρος του κεφαλαίου θα παρουσιασθούν οι προγραμματιστικές μέθοδοι και ορισμένες βασικές λεπτομέρειες από τα αντίστοιχα αρχεία που αναπτύχθηκαν με στόχο αρχικά τη δημιουργία μιας εύχρηστης διεπιφάνειας χρήστη (πλευρά View) και στη συνέχεια τον αποτελεσματικό χειρισμό των συμβάντων μέσω της εισόδου από το χρήστη (πλευρά Controller).

5.1.1 Δημιουργία Όψεων

Για το σχεδιασμό των όψεων δημιουργήθηκαν αρχεία zul (χρησιμοποιώντας το ZK framework, που παρουσιάζεται στην ενότητα 5.2.1) τα οποία εντέλει μέσω ενός εξυπηρετητή ιστού (web server) παράγουν σελίδες σε κώδικα html που παρουσιάζονται μέσω ενός φυλλομετρητή ιστού (browser). Για κάθε καρτέλα της πλευράς View σχεδιάστηκε ένα αρχείο zul. Υπάρχουν δηλαδή δύο αρχεία zul για την καρτέλα Case Manager, άλλα έξι αρχεία zul για τον Ontology Manager, ενώ εξαίρεση αποτελεί ο Context Manager για τον οποίο σχεδιάστηκε μόλις ένα αρχείο zul.

Το σχεδιαστικό μοτίβο των παρουσιάσεων επέβαλε στο αριστερό μέρος των όψεων να εμφανίζονται τα στοιχεία-χαρακτηριστικά της εκάστοτε καρτέλας, ενώ στο δεξιό μέρος οι ανάλογες πληροφορίες του στοιχείου που θα επιλεγθεί. Για παράδειγμα, στην καρτέλα Case Types για την παρουσίαση των ασαφών τύπων περιπτώσεων στα αριστερά γίνεται η καταγραφή των τύπων περιπτώσεων, ενώ στα δεξιά εμφανίζονται πληροφορίες που αφορούν τον τύπο που θα επιλεγθεί, όπως μια σύντομη περιγραφή του, τις ιδιότητές του (attributes) και το νοηματικό πλαίσιο ανάκτησης που του έχει ανατεθεί.

Για την παρουσίαση των στοιχείων στα αριστερά της εκάστοτε καρτέλας χρησιμοποιήθηκαν τα παρακάτω συστατικά στοιχεία (components) που προσφέρει το πλούσιο περιβάλλον του ZK framework, χαρακτηριστικό των οποίων είναι ότι ο χρήστης επιλέγοντας ένα στοιχείο δημιουργείται ένα συμβάν το οποίο το δέχεται φυσικά η πλευρά Controller, και είναι τα:

- Δέντρο (trees): Χρησιμοποιήθηκε στις περισσότερες των περιπτώσεων με στόχο την απαρίθμηση των εκάστοτε στοιχείων. Σημαντικό προτέρημα του component αυτού είναι η δυνατότητα για αναπαράσταση ιεραρχικών σχέσεων μεταξύ των στοιχείων που συνιστούν ένα δέντρο, δηλαδή μια δενδρική δομή. Η δομή αυτή κρίθηκε απαραίτητη για την αναπαράσταση των σχέσεων-στιγμιότυπων για τις ιεραρχικές σχέσεις της ασαφούς οντολογίας πεδίου. Σημαντική συμβολή της δομής αυτής καταγράφηκε και στην παρουσίαση των κλάσεων της οντολογίας που πρέπει να δίνουν στο χρήστη τη δυνατότητα για ορισμό ιεραρχίας. Στις υπόλοιπες των περιπτώσεων πάντως χρησιμοποιήθηκαν δέντρα λόγω της ευκολίας δημιουργίας τους, έχοντας βέβαια ως απώτερο στόχο την εκμετάλλευση της δομής που προσφέρουν, όπως έγινε για την αναπαράσταση των ασαφών τύπων περιπτώσεων που αν και δεν υποστηρίζουν ακόμα δενδρική δομή, σε μελλοντικές επεκτάσεις της εφαρμογής ίσως και να χρειαστεί.
- Λίστα (listbox): Μόνο στην παρουσίαση των πλαισίων για την ανάκτηση περιπτώσεων (case retrieval context) και αυτών για την ομοιότητα των οντολογικών σχέσεων (ontology similarity context) στον Context Manager χρησιμοποιήθηκε το component αυτό, καθώς τα χαρακτηριστικά αυτά δεν εμπεριέχουν κάποια δενδρική δομή για να αναπαρασταθούν με δέντρα.

Για την αναπαράσταση των πληροφοριών των στοιχείων που επιλέγονται από την αριστερή πλευρά και παρουσιάζονται στη δεξιά πλευρά των καρτελών χρησιμοποιήθηκαν συστατικά στοιχεία (components) του ZK framework που χρησιμοποιούνται όχι μόνο για την

παρουσίαση στο χρήστη των τιμών ορισμένων χαρακτηριστικών όπως το πεδίο τιμών και ορισμού μιας σχέσης, αλλά και για να δεχτούν την είσοδο από το χρήστη σε μορφή ανάλογη με τον τύπο του στοιχείου. Συνήθως ομαδοποιούνται υπό μορφή πλέγματος χρησιμοποιώντας για το λόγο αυτό το component grid. Αυτά είναι:

- Πεδίο κειμένου (textbox): Το component αυτό εμφανίζει και δέχεται περιεχόμενο σε μορφή κειμένου με όλους τους πιθανούς χαρακτήρες του πληκτρολογίου. Χαρακτηριστική η εφαρμογή του είναι στο πεδίο περιγραφής (Description) του εκάστοτε επιλεγμένου στοιχείου που αφορά μια σύντομη λεκτική αναφορά στο τι αναπαριστά το στοιχείο αυτό.
- Πεδίο ακεραίου (intbox): Το στοιχείο αυτό δε δέχεται άλλη είσοδο από το χρήστη πλην ακεραίων αριθμών. Αν δηλαδή δοθεί είσοδος με τη μορφή αλφαριθμητικών χαρακτήρων, αυτή θα αγνοηθεί μέχρις ότου να δοθεί ένας ακέραιος αριθμός. Χρησιμοποιήθηκε για πεδία οι τιμές των οποίων όφειλαν να είναι ακέραιοι.
- Πεδίο πραγματικού (doublebox): Παρόμοια χρησιμότητα με το intbox μόνο που αντί για ακεραίους δέχεται και αναπαριστά αριθμούς με υποδιαστολή, δηλαδή πραγματικούς.
- Πεδίο για επιλογή από drop-down λίστα (combobox): Πρόκειται για component το οποίο δίνει την ευκαιρία στο χρήστη αντί να γράφει μόνος του το κείμενο που θα εισάγει, να επιλέγει δεδομένα από μια drop-down λίστα. Με τον τρόπο αυτόν παρέχεται κάποιου είδους βοήθεια για το τι θα εισάγει ο χρήστης καθώς ορισμένα πεδία χρειάζονται συγκεκριμένες τιμές για είσοδο.

Τα στοιχεία (components) που χρησιμοποιήθηκαν για το ρόλο του υποδοχέα (container), δηλαδή με σκοπό να εμπεριέχουν άλλα συστατικά στοιχεία ως μια ενιαία δομή, είναι τα εξής:

- Window: Πρόκειται για τον πιο σημαντικό υποδοχέα, αφού όλα τα components ενός αρχείου `zui` είναι τοποθετημένα σε αυτόν. Σε κάθε window έχει δοθεί ένα μοναδικό `id` που το χαρακτηρίζει, `id` που χρησιμοποιείται για τις κλήσεις των μεθόδων που καλούνται όταν γίνει κάποιο συμβάν. Άξιες αναφοράς είναι συγκεκριμένες περιπτώσεις στις οποίες ο χρήστης επιθυμώντας να εισάγει κάποιο νέο στοιχείο, όπως για παράδειγμα ένα νέο ασαφή τύπο περιπτώσεων, μια νέα κλάση ή ασαφής σχέση, αναδύονται παράθυρα στα οποία ο χρήστης εισάγει το όνομα του νέου στοιχείου και το `uri` του.
- Panel: Χρησιμοποιείται σαν ένα μεγάλο κουτί με τίτλο που παρέχει δυνατότητες για μεγιστοποίηση, σύντομη απόκρυψη, κ.α., ενώ προσφέρει τρεις πιθανές θέσεις για τοποθέτηση toolbars (component με στόχο την είσοδο από το χρήστη). Μέσα σε

αυτόν τον υποδοχέα παρουσιάζονται τα περισσότερα των components όπως τα trees, listbox, grids.

- **Tabbox**: Αυτό το component επιτρέπει την ομαδοποίηση των components σε μερικά γκρουπ έχοντας ως στόχο να εμφανίζονται μόνο ένα από αυτά. Πρόκειται για τις γνωστές καρτέλες που χρησιμοποιήθηκαν για το διαχωρισμό των λειτουργιών του GUI σε τρεις βασικές ομάδες (Case, Ontology και Context Manager) και τον περαιτέρω διαχωρισμό των παραπάνω ομάδων σε άλλες ανάλογα με τις λειτουργίες και τους σκοπούς που επιτελούν. Έτσι, στο χρήστη εμφανίζονται τα περιεχόμενα μόνο της μιας καρτέλας, ενώ επιλέγοντας μια άλλη καρτέλα παρουσιάζονται τα περιεχόμενα της τελευταίας και αποκρύπτονται αυτά της προηγούμενης.

Τέλος σημαντικό ρόλο στη λειτουργία του GUI είχαν τα components toolbarbuttons. Συνήθως χρησιμοποιούνταν σε ομάδες με τη βοήθεια του υποδοχέα caption. Στόχος τους ήταν να παρέχουν στο χρήστη ένα σημείο με τη μορφή μικρής εικόνας, το οποίο αφού το επέλεγε με το ποντίκι έστελνε ένα συμβάν (event) στην πλευρά του Controller. Αυτές οι εικόνες που χρησιμοποιήθηκαν κατά σωρηδόν ήταν αυτές για την προσθήκη και τη διαγραφή στοιχείων της οντολογίας, όπως για παράδειγμα προσθήκη και διαγραφή μιας κλάσης, μιας ιεραρχικής σχέσης, ενός τύπου περιπτώσεων.

5.1.2 Δημιουργία Κλάσεων

Για το χειρισμό των συμβάντων μέσω της εισόδου από το χρήστη δημιουργήθηκαν τόσες κλάσεις JAVA όσες είναι και ο αριθμός των αρχείων zul. Κάθε αρχείο zul δηλαδή, που περιέχει ένα βασικό window, χειρίζεται από μια java κλάση. Επίσης για την αναπαράσταση των components ενός αρχείου zul μέσω του κώδικα java δημιουργούνται στιγμιότυπα κλάσεων με όνομα το αντίστοιχο component. Για παράδειγμα, αν σε μια σελίδα zul υπάρχουν τρία textbox τότε θα οριστούν και τρία στιγμιότυπα της κλάσης Textbox στην αντίστοιχη κλάση java. Οι κλάσεις αυτές γίνονται import από το πακέτο (package) org.zkoss.zul.*. Έτσι επιτυγχάνεται ο δυναμικός έλεγχος των components.

Όσον αφορά την java κλάση τώρα, αυτή κληρονομεί τις μεθόδους από την κλάση BasicWindow η οποία με τη σειρά της κληρονομεί την κλάση Window, βασική κλάση που προσφέρει η βιβλιοθήκη της java. Η BasicWindow υλοποιήθηκε με στόχο την ύπαρξη ορισμένων κοινών λειτουργιών των κλάσεων-παραθύρων, όπως η δημιουργία ενός δέντρου ή μιας λίστας. Τα πλεονεκτήματα αυτού του μοτίβου έχουν να κάνουν με τον αποτελεσματικό χειρισμό των συμβάντων και την αλλαγή ή απόδοση νέων χαρακτηριστικών των components του αρχείου zul.

Για την πρώτη περίπτωση, όταν δημιουργείται ένα συμβάν στο GUI τη στιγμή που εμφανίζεται ένα αρχείο zul, τότε αυτό θα το χειριστεί μια μέθοδος της κλάσης που έχει ανατεθεί στο συγκεκριμένο αρχείο zul, η οποία μέθοδος έχει φυσικά σχεδιαστεί αποκλειστικά για το χειρισμό του συγκεκριμένου συμβάντος. Με τον τρόπο αυτόν αν οποιαδήποτε συμβάν επιθυμεί ειδική μεταχείριση, τότε δημιουργείται μια ανάλογη μέθοδος που τοποθετείται στη java κλάση που χειρίζεται τη σελίδα zul στην οποία λαμβάνει μέρος το συμβάν. Για παράδειγμα, έχουν δημιουργηθεί μέθοδοι για το χειρισμό συμβάντων όταν ο χρήστης πατάει toolbarbuttons για προσθήκη κάποιου νέου στοιχείου, όπως μια νέα κλάση ή ασαφής σχέση, στις οποίες μεθόδους γίνεται όλη η διαδικασία προσθήκης του εν λόγω στοιχείου.

Για τη δεύτερη περίπτωση, σημαντικό ρόλο προς την κατεύθυνση της αλλαγής χαρακτηριστικών των components διαδραμάτισε η κλάση BasicWindow και οι μέθοδοι που υλοποιήθηκαν με στόχο την παραπάνω κατεύθυνση, ενώ και στη περίπτωση αυτή καλούνται συγκεκριμένες μέθοδοι ανάλογα με την ανάγκη για αλλαγή και το συμβάν που δημιουργεί η ανάγκη αυτή. Για παράδειγμα, μερικοί μέθοδοι από την κλάση BasicWindow είναι:

- checkForZeroInputTextbox: Πρόκειται για μέθοδο που ελέγχει αν το component textbox που δέχεται ως είσοδο περιέχει μηδενική είσοδο από το χρήστη. Η συγκεκριμένη μέθοδος κρίθηκε απαραίτητη σε περιπτώσεις που ο χρήστης πρέπει να δώσει κάποια είσοδο.
- showWarningMessageTextbox: Συνήθως συνδυάζεται με την παραπάνω αφού σε περίπτωση που δεν έχει δοθεί είσοδος σε κάποιο textbox, εμφανίζεται ένα νέο παράθυρο που ειδοποιεί το χρήστη για το γεγονός αυτό.
- buildTreeFromLista: Δέχεται ως όρισμα μια λίστα από Strings (uri από στοιχεία) και χτίζει με αυτά ένα δέντρο χωρίς ιεραρχία.
- buildListFromLista: Ανάλογη λειτουργία με την παραπάνω μέθοδο μόνο που αντί για δέντρο χτίζει μια λίστα.
- checkForValidUri: Σε περίπτωση που γίνεται είσοδος ενός νέου στοιχείου από το χρήστη απαιτείται είσοδος και του uri του. Η μέθοδος αυτή δέχεται ως όρισμα ένα String και ελέγχει αν αυτό μπορεί να είναι έγκυρο για να γίνει uri. Αν είναι έγκυρο η είσοδος του στοιχείου γίνεται αυτόματα, αλλιώς το στοιχείο δεν εισάγεται και εμφανίζεται ένα μήνυμα στο χρήστη που τον ειδοποιεί για το γεγονός αυτό.

5.1.2.1 Αναφορά Μεθόδων από API του IKARUS-CBR

Σημαντικό μέρος για την επίτευξη των λειτουργιών και στόχων στην πλευρά του Controller καταλαμβάνουν οι μέθοδοι που προσφέρει το API του πλαισίου IKARUS-CBR, καθώς με

βάση αυτές τις μεθόδους επιτυγχάνονται όχι μόνο η αλλαγή της κατάστασης του Model αλλά και αλλαγές στις όψεις της διεπιφάνειας, δηλαδή στην πλευρά View. Στο σημείο αυτό θα αναφερθούν οι μέθοδοι αυτές που χρησιμοποιήθηκαν από τις διεπιφάνειες (interfaces) του API αυτού.

Αρχικά διακρίνονται από το interface του Ontology Manager οι εξής μέθοδοι (στις παρενθέσεις αναφέρονται τα ορίσματα):

- `getOntologyClasses(String ontology_uri)`: Δέχεται ως όρισμα το uri της οντολογίας και επιστρέφει μια λίστα από κλάσεις που ανήκουν στη συγκεκριμένη οντολογία.
- `addClass(String class_uri)`: Προσθέτει μια κλάση στην οντολογία με uri αυτό του ορίσματος.
- `deleteClass(String class_uri)`: Διαγράφει την κλάση με το συγκεκριμένο uri.
- `getClassInferredIndividuals(String class_uri)`: Επιστρέφει μια λίστα από άτομα που είναι στιγμιότυπα της κλάσης που δέχεται ως όρισμα.
- `addClassIndividual(String individual_uri, String class_uri)`: Προσθέτει σε μια κλάση ένα άτομο που είναι στιγμιότυπό της.
- `deleteIndividual(String individual_uri)`: Διαγράφει το άτομο του ορίσματος.
- `getOntologyObjectProperties(String ontology_uri)`: Επιστρέφει τη λίστα με τις σχέσεις (object properties) της συγκεκριμένης οντολογίας.
- `getObjectPropertiesByDomain(String domain)`: Δεχόμενη ως όρισμα το uri μιας κλάσης επιστέφει μια λίστα με τις σχέσεις στις οποίες συμμετέχει η κλάση αυτή ως πεδίο ορισμού.

Στη συνέχεια από το interface του Fuzzy Ontology Manager έχουμε τις εξής μεθόδους:

- `getFuzzyOntologyHierarchicalRelationsByDomain(String domain)`: Επιστρέφει τη λίστα με τις ασαφείς ιεραρχικές σχέσεις με πεδίο ορισμού τη συγκεκριμένη κλάση. Αντίστοιχη μέθοδος υπάρχει και για τις συσχετιστικές σχέσεις.
- `addFuzzyOntologyHierarchicalRelation(String relation_uri)`: Προσθέτει την ιεραρχική σχέση το uri της οποίας περνιέται ως όρισμα. Αντίστοιχη μέθοδος υπάρχει και για τις συσχετιστικές σχέσεις.
- `deleteFuzzyOntologyRelation(String relation_uri)`: Διαγράφει τη συγκεκριμένη σχέση.
- `addFuzzyOntologyRelationInstance(String source_uri, String relation_uri, String destination_uri, String destination_class, float fuzzy_degree)`: Προσθέτει ένα

στιγμιότυπο μιας σχέσης με uri το `relation_uri` που συνδέει δύο στιγμιότυπα κλάσεων τα `source_uri` και `destination_uri` (με το 2^ο να ανήκει στην κλάση `destination_class`) με βαθμό ασάφειας `fuzzy_degree`.

Από το interface του Case Manager χρησιμοποιήθηκαν οι εξής μέθοδοι:

- `getCaseTypes()`: Επιστρέφει τη λίστα με τους τύπους των ασαφών περιπτώσεων.
- `addCaseType(String type, int mode)`: Προσθέτει ένα νέο ασαφή τύπο περίπτωσης με ορίσματα το uri του και έναν ακέραιο, ο οποίος αν είναι 0 πρόκειται για καινούριο τύπο, ενώ αν είναι 1 για τύπο που έχει ήδη οριστεί στην ασαφή οντολογία πεδίου ως ασαφής κλάση.
- `deleteCaseType(String type, int mode)`: Διαγράφει το συγκεκριμένο τύπο περιπτώσεων, ενώ το mode λειτουργεί όπως και στη παραπάνω μέθοδο.
- `getCaseTypeCrispRelationAttributes(String case_type)`: Επιστρέφει μια λίστα με τις μη ασαφείς σχέσεις-ιδιότητες του συγκεκριμένου τύπου. Αντίστοιχες μέθοδοι υπάρχουν και για τους υπόλοιπους τύπους ιδιοτήτων (attributes), δηλαδή για τις ασαφείς σχέσεις και ιδιότητες, τις μη ασαφείς ιδιότητες και τις ασαφώς αποτιμώμενες ιδιότητες.
- `addCaseTypeCrispRelationAttribute(String case_type, String attribute)`: Λειτουργία προσθήκης σε ένα τύπο περιπτώσεων ένα νέο attribute τύπου μη ασαφούς σχέσεως. Αντίστοιχες μέθοδοι φυσικά υπάρχουν και για τα υπόλοιπα είδη των attributes.
- `deleteCaseTypeCrispRelationAttribute(String attribute)`: Διαγράφει την ιδιότητα τύπου μη ασαφούς σχέσεως που δέχεται ως όρισμα.
- `getCases(String case_type)`: Για τον ασαφή τύπο περιπτώσεων του ορίσματος επιστρέφει μια λίστα με τις περιπτώσεις-στιγμιότυπά του.
- `addCase(String case_uri, String case_type)`: Λειτουργία προσθήκης σε έναν τύπο περιπτώσεων μια περίπτωση, με ορίσματα τα uri τους.
- `deleteCase(String case_uri)`: Διαγραφή της περίπτωσης-στιγμιότυπο.

Από το interface του Context Manager χρησιμοποιήθηκαν οι εξής μέθοδοι:

- `getCaseRetrievalContexts()`: Επιστρέφει τη λίστα με τα case retrieval contexts.
- `addCaseRetrievalContext(String case_retrieval_context)`: Προσθέτει το συγκεκριμένο case retrieval context.

- `deleteCaseRetrievalContext(String case_retrieval_context)`: Διαγράφει το case retrieval context του ορίσματος.
- `getCaseTypeRetrievalContexts(String case_type)`: Επιστρέφει τη λίστα με τα case retrieval contexts που έχουν συνδεθεί με το συγκεκριμένο case type.
- `addCaseTypeRetrievalContext(String case_type, String case_retrieval_context)`: Συνδέει το case retrieval context με το case type.
- `getCaseTypeActiveRetrievalContext(String case_type)`: Γυρίζει το ενεργοποιημένο case retrieval context του τύπου περιπτώσεων του ορίσματος.
- `setCaseTypeActiveRetrievalContext(String case_type, String context)`: Ενεργοποιεί το case retrieval context για το συγκεκριμένο τύπο περιπτώσεων.
- `getCaseAttributeSimilarityContexts(String case_retrieval_context)`: Επιστρέφει τα case attribute similarity context που ανήκουν στο συγκεκριμένο case retrieval context.
- `getOntologySimilarityContexts()`: Επιστρέφει τη λίστα με τα υπάρχοντα ontology similarity context.
- `getFuzzyHierarchicalRelationContexts(String ontology_similarity_context)`: Γυρίζει τη λίστα με τα contexts των ασαφών ιεραρχικών σχέσεων του συγκεκριμένου ontology similarity context. Αντίστοιχη μέθοδος και για τις συσχετιστικές σχέσεις.
- `addFuzzyHierarchicalRelationContext(String ontology_similarity_context, String fuzzy_hierarchical_relation_context)`: Προσθέτει το ένα context ασαφούς ιεραρχικής σχέσης στο συγκεκριμένο ontology similarity context. Αντίστοιχη μέθοδος και για τις συσχετιστικές σχέσεις.
- `deleteFuzzyHierarchicalRelationContext(String fuzzy_hierarchical_relation_context)`: Διαγράφει το ένα context ασαφούς ιεραρχικής σχέσης. Αντίστοιχη μέθοδος και για τις συσχετιστικές σχέσεις.
- `getContextHierarchicalRelation(String fuzzy_hierarchical_relation_context)`: Επιστρέφει την ιεραρχική σχέση με την οποία έχει δεθεί το context της ασαφούς ιεραρχικής σχέσης.
- `setContextHierarchicalRelation(String fuzzy_hierarchical_relation_context, String relation)`: Δένει τη συγκεκριμένη ιεραρχική σχέση με το fuzzy hierarchical relation context.
- `getContextComponentRelations(String fuzzy_composite_relation_context)`: Γυρίζει μια λίστα με τα context των σχέσεων που αποτελούν μια σύνθετη σχέση το context της οποίας περινιέται ως όρισμα.

- `addContextComponentRelation(String fuzzy_composite_relation_context, String relation)`: Προσθέτει ένα context μιας σχέσης που συναποτελεί μαζί με άλλες σχέσεις μια σύνθετη σχέση, το context της οποίας περνιέται ως όρισμα.
- `deleteContextComponentRelation(String relation)`:): Διαγράφει το context μιας σχέσης που συναποτελεί μαζί με άλλες σχέσεις μια σύνθετη σχέση, το context της οποίας περνιέται ως όρισμα.

5.2 Πλατφόρμες και προγραμματιστικά εργαλεία

Στο δεύτερο μέρος του κεφαλαίου θα αναλυθούν τα προγραμματιστικά εργαλεία και οι λόγοι για τους οποίους αυτά χρησιμοποιήθηκαν.

5.2.1 ZK framework

Πρόκειται για ένα ανοιχτού κώδικα (open source) πλαίσιο (framework) γραμμένο στη γλώσσα Java. Κύριος στόχος του πλαισίου αποτελεί η ανάπτυξη δυναμικών ιστοσελίδων και υπηρεσιών διαδικτύου (web services) παρέχοντας στο χρήστη εργαλεία για εύκολο και εύχρηστο σχεδιασμό πλούσιων γραφικών διεπιφανειών χρήστη (GUI).

Η ευκολία έγκειται στο γεγονός ότι απαιτούνται λίγες προγραμματιστικές γνώσεις, ενώ στο σχεδιασμό των GUIs δεν εμπλέκεται καθόλου η JavaScript. Η ευχρηστία υποστηρίζεται από την ύπαρξη βασισμένου σε AJAX μηχανισμού οδηγούμενου από γεγονότα (Ajax-based event-driven mechanism) [13]. Τα γεγονότα δημιουργούνται από τη δραστηριότητα του εκάστοτε χρήστη κατά την περιήγησή του σε μια ιστοσελίδα. Ο μηχανισμός αυτός που βασίζεται σε AJAX (Asynchronous JavaScript and XML) πρακτικά σημαίνει ότι δε χρειάζεται να φορτωθεί μια ολόκληρη ιστοσελίδα κάθε φορά που ο χρήστης αλληλεπιδρά με αυτήν, αφού τα δεδομένα μεταξύ του εξυπηρετητή και του περιηγητή ιστού μεταφέρονται ασύγχρονα στο «παρασκήνιο». Έτσι δίνεται η εντύπωση στον τελικό χρήστη μιας εφαρμογής GUI ανάλογης με αυτές που τρέχουν σε έναν υπολογιστή.

Για το σχεδιασμό του GUI το ZK πλαίσιο προσφέρει περίπου 200 συστατικά στοιχεία (components) βασισμένα στις XML και HTML. Τα συστατικά αυτά όπως είναι τα δέντρα, λίστες, πεδία εισαγωγής κειμένου, παρουσιάστηκαν στην παράγραφο 5.1.1. Για τον ίδιο σκοπό έχει σχεδιαστεί μια γλώσσα σήμανσης, η ZUML (ZK User Interface Markup Language), η οποία επιτρέπει πάντως στον προγραμματιστή όχι μόνο να αναμειξεί διαφορετικές γλώσσες σήμανσης όπως η Mozilla XUL και XHTML αλλά και να

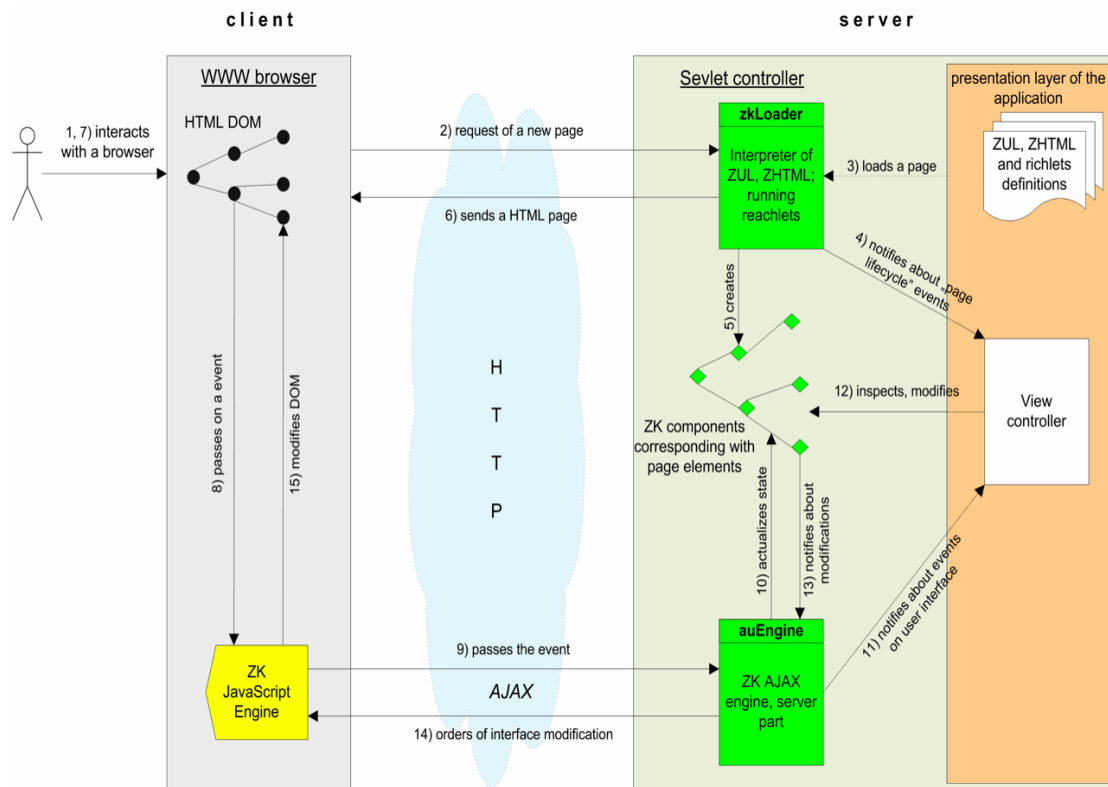
ενσωματώσει σε μια σελίδα κώδικα τη JAVA για το χειρισμό των components. Η γλώσσα αυτή δημιουργεί αρχεία με κατάληξη “.zul”.

Για το χειρισμό των συμβάντων τώρα παρέχεται η δυνατότητα στον προγραμματιστή είτε να δημιουργήσει μια κεντρική κλάση στην οποία θα προωθούνται όλα τα συμβάντα ή να ορίσει χειριστές συμβάντων μέσα στη σελίδα zul χρησιμοποιώντας είτε γλώσσα JAVA ή Ruby ή Python ή JavaScript. Στην παρούσα διπλωματική προτιμήθηκε η κεντρική διαχείριση των συμβάντων κάθε αρχείου zul από μία java κλάση, όπως έχει περιγραφεί και στην παράγραφο 5.1.2, καθώς με το δεύτερο θα δημιουργούνταν ογκώδη αρχεία που θα μετέτρεπαν τη συντήρηση του κώδικα σε δύσκολη κατάσταση.

5.2.2 Apache Tomcat 6

Με λίγα λόγια επομένως το ZK πλαίσιο δέχεται σελίδες-αρχεία ZUML ως είσοδο στην πλευρά του εξυπηρετητή και παράγει ως έξοδο dhtml (δυναμική html) που τη διαχειρίζεται ο browser. Για τον παραπάνω λόγο χρειάζεται μια εφαρμογή που να λειτουργεί ως web server και θα αναλαμβάνει την παραπάνω διαδικασία. Αυτή είναι το Apache Tomcat 6, το οποίο πρόκειται για εφαρμογή ανοιχτού κώδικα που παρέχει στο χρήστη ένα περιβάλλον εξυπηρετητή ιστού με στόχο να εκτελέσει java κώδικα. Αυτό εν ολίγοις σημαίνει ότι το Apache αναλαμβάνει λειτουργίες ενός web server που δέχεται αιτήσεις από έναν πελάτη μέσω του πρωτοκόλλου HTTP και τις διαχειρίζεται με τη βοήθεια java κλάσεων που ονομάζονται servlets. Τα τελευταία ανάλογα με την είσοδο του χρήστη παράγει και ανάλογο περιεχόμενο ως έξοδο σε dhtml.

Χαρακτηριστική των παραπάνω δύο παραγράφων είναι το Σχήμα 5.1 [12] που περιγράφει ένα σενάριο κατά το οποίο ο πελάτης επιδρά με μια σελίδα στον browser από τη στιγμή της πρώτης εμφάνισής της μέχρι και της στιγμή που ο server θα επιφέρει αλλαγές στη σελίδα ανάλογα με την είσοδο που δέχεται με τη μορφή συμβάντος. Το ρόλο του server παίζει το Apache Tomcat 6.



Σχήμα 5.1 Αρχιτεκτονική του ZK framework

5.2.3 Subversion

Ένα από τα πιο σημαντικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι το λογισμικό subversion. Το πρόγραμμα αυτό χαρακτηρίζεται ως σύστημα ελέγχου αναθεωρήσεων (revision control system) του κώδικα που γράφεται κατά τη διαδικασία ανάπτυξης μίας εφαρμογής. Η λειτουργία που επιτελεί είναι να κρατά τις επιμέρους αναθεωρήσεις σε μία κεντρική αποθήκη και να εντοπίζει τις διαφορές από έκδοση σε έκδοση και τις κάνει εμφανείς στους προγραμματιστές όταν επιχειρούν να αποθηκεύσουν κάποια νέα αναθεώρηση του κώδικα. Όταν εντοπίζεται κάποιο σημείο σύγκρουσης ανάμεσα σε δύο εκδόσεις του ίδιου κώδικα, το πρόγραμμα δεν επιτρέπει την αποθήκευση της νέας έκδοσης έως ότου επιλυθεί η σύγκρουση. Με τον τρόπο αυτό, πολλοί προγραμματιστές μπορούν να δουλεύουν στο ίδιο έργο χωρίς να φοβούνται μήπως σβήσουν ή χαλάσουν κάποιο κομμάτι κώδικα. Αυτό επιτυγχάνεται τόσο επειδή υπάρχουν διαθέσιμες ανά πάσα στιγμή όλες οι προηγούμενες εκδόσεις, όσο και γιατί εάν υπάρξει κάποια σύγκρουση αυτό γίνεται σαφές. Η χρήση λοιπόν του λογισμικού αυτού, συνετέλεσε στο να μην εμπλακούν οι διαφορετικές

προγραμματιστικές δουλειές των πακέτων SemanticCBREngine, OntologyAccessFramework και FuzzyLogicAPI και αυτού για το σχεδιασμό του GUI, καθώς η πρώτη διαδικασία επιτελέστηκε αποκλειστικά στα πλαίσια της διδακτορικής διατριβής που αφορά το IKARUS-CBR. Σαν αποτέλεσμα από τη μια πλευρά μπορούσα να δω και να αξιοποιήσω τις αλλαγές κώδικα στα παραπάνω πακέτα, ενώ από την άλλη δεν είχα το δικαίωμα παρέμβασης και αλλαγής στον κώδικα αυτόν.

5.2.4 StarUML

Πρόκειται για ένα πρόγραμμα-εφαρμογή που χρησιμοποιήθηκε με σκοπό το σχεδιασμό των διαγραμμάτων χρήσης (use case diagrams), δραστηριοτήτων (activity diagrams) καθώς και του διαγράμματος συστατικών (component diagram) για την αρχιτεκτονική του ολικού συστήματος, τα οποία διαγράμματα παρουσιάστηκαν στο 4^ο κεφάλαιο. Τα διαγράμματα αυτά περιγράφουν βασικές λειτουργίες και τη δομή του συστήματος σε μια παγκόσμια «γλώσσα» με τη βοήθεια γραφικών συμβολισμών που χρησιμοποιείται για τη μοντελοποίηση διαδικασιών ανάπτυξης λογισμικού, τη UML.

5.2.5 Πλατφόρμα ανάπτυξης

Η ανάπτυξη της εφαρμογής έγινε σε περιβάλλον Windows 7 Home Premium. Για τη συγγραφή του κώδικα χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment ή IDE) **IntelliJ IDEA** που παρέχει τόσο έναν συντάκτη κώδικα με πλήρη υποστήριξη για αρχεία html, zul, javascript και java, όσο και ένα περιβάλλον για τη λειτουργία του subversion. Η εφαρμογή δοκιμάστηκε κατά την ανάπτυξη της στους browser Mozilla Firefox και Google Chrome.

6

Σενάριο Χρήσης IKARUS-GUI

Στο παρόν κεφάλαιο θα γίνει η παρουσίαση ενός σεναρίου χρήσης της γραφικής διεπιφάνειας χρήστη (Graphic User Interface - GUI) που σχεδιάστηκε για το υβριδικό πλαίσιο IKARUS-CBR. Αρχικά θα περιγραφεί το σενάριο, ενώ στη συνέχεια θα παρουσιαστούν αναλυτικά οι ενέργειες που πρέπει να γίνουν ώστε να επιτευχθούν οι στόχοι του σεναρίου. Με στόχο την όσο το δυνατόν καλύτερη επεξήγηση των παραπάνω ενεργειών παρατίθενται ορισμένες όψεις οθονών (screenshots) που πάρθηκαν κατά τη λειτουργία του GUI.

6.1 Περιγραφή Σεναρίου

Το σενάριο αφορά την ανάπτυξη της ηλεκτρονικής βιβλιοθήκης του οργανισμού Διαχείρισης Ελληνικού Συστήματος Μεταφοράς Ηλεκτρικής Ενέργειας (ΔΕΣΜΗΕ) που παρέχει στο κοινό πρόσβαση σε γνώση που αφορά την ελληνική Αγορά Ενέργειας. Η γνώση αυτή έχει τη μορφή τεχνικών εγγράφων τα οποία μοντελοποιούνται στο πλαίσιο IKARUS-CBR ως περιπτώσεις στις οποίες και αποδίδεται σημασιολογικό περιεχόμενο μέσω μιας ασαφούς οντολογίας στο πεδίο της αγοράς ηλεκτρικής ενέργειας. Τέλος, η ανάκτηση των εγγράφων αυτών προσαρμόζεται κατάλληλα μέσω του ορισμού ενός κατάλληλου νοηματικού πλαισίου ανάκτησης.

6.1.1 Ορισμός Περιπτώσεων

Με τη βοήθεια ειδικών του τομέα της Ενεργειακής Αγοράς ορίστηκε η Οντολογία Ασαφών Περιπτώσεων του μοντέλου της ΔΕΣΜΗΕ. Αυτή αποτελείται από μόλις έναν ασαφή τύπο περιπτώσεων με όνομα HTSODocument, το οποίο με τη σειρά του περιλαμβάνει τα ακόλουθα πέντε γνωρίσματα περιπτώσεων:

- Title: Ο τίτλος του εγγράφου.
- Date: Η ημερομηνία δημοσιοποίησης του εγγράφου.
- Contributors: Άνθρωποι ή οργανισμοί που συνετέλεσαν στη δημιουργία του.
- ThematicContent: Λέξεις κλειδιά που χαρακτηρίζουν το σημασιολογικό περιεχόμενο του εγγράφου.
- RelevantExternalResources: Σύνδεσμοι σε ανάλογες με το έγγραφο πηγές που δεν περιέχονται στη βιβλιοθήκη.

Τα πρώτα δύο γνωρίσματα είναι πρακτικά μη ασαφή (crisp) και έτσι ορίζονται στην οντολογία ως μη ασαφείς ιδιότητες (crisp property attributes). Τα γνωρίσματα αντίθετα Contributors και RelevantExternalResources θεωρούνται ασαφή και παρουσιάζονται στην οντολογία ως ασαφείς ιδιότητες (fuzzy property attributes). Η ασάφεια στην ιδιότητα των Contributors αντικατοπτρίζει το πόσο ένας συγκεκριμένος οργανισμός ή άτομο έχει συνεισφέρει στη δημιουργία του εγγράφου, ενώ σε αυτήν των RelevantExternalResources δείχνει το βαθμό σχετικότητας των εξωτερικών πηγών με το έγγραφο.

Τέλος, το γνώρισμα ThematicContent μοντελοποιείται ως μία ασαφής σχέση που συνδέει τις περιπτώσεις με τις λέξεις κλειδιά που αντιπροσωπεύουν το νόημά τους. Αυτές οι λέξεις κλειδιά έχουν τη μορφή των οντολογικών στιγμιότυπων που ανήκουν σε έννοιες-κλάσεις ασαφούς οντολογίας που μοντελοποιεί τον τομέα της Αγοράς Ενέργειας. Πρόκειται για τις έννοιες Action, Extent, Obligation, Participant, Process, InformationSource, Right, Rule, Specification και System. Οι έννοιες αυτές συσχετίζονται με τις εξής σχέσεις:

Μη ασαφείς συσχετιστικές (Crisp Associative):

- *hasSpecification*(System, Specification)
- *isFoundInSource*(Extent, InformationSource)
- *regardsParticipant* (Obligation or Right, Participant)
- *regardsProcess* (Rule, Process)

Ασαφείς συσχετιστικές (Fuzzy Associative):

- $isPerformedByParticipant(Action, Participant)$
- $isPerformedInTheContextOfProcess(Action, Process)$
- $isUtilizedInProcess(InformationSource, Process)$
- $participatesInProcess(Participant, Process)$

Ασαφείς ιεραρχικές (Fuzzy Hierarchical):

- $isPartOfSpecification (Specification, Specification)$
- $isPartOfProcess(Process)$

Αξίζει να υπογραμμιστεί ότι η δημιουργία εγγράφων και η ανάθεση τιμών στα γνωρίσματά τους έγινε με τη βοήθεια ειδικών του τομέα της Αγοράς Ενέργειας.

6.1.2 Ορισμός Νοηματικού Πλαισίου Ανάκτησης

Σύμφωνα με το μοντέλο ομοιότητας που παρουσιάστηκε στην παράγραφο 3.2 ορίζονται τα εξής νοηματικά πλαίσια για κάθε ασαφές γνώρισμα:

- Ένα fuzzy property attribute similarity context (fpsc) για το γνώρισμα Contributors.
- Ένα fuzzy property attribute similarity context (fpsc) για το γνώρισμα RelevantExternalResources.
- Ένα fuzzy ontology relation similarity context για αυτές τις σχέσεις της ασαφούς οντολογίας της Αγοράς Ενέργειας που συνδέουν στιγμιότυπα τα οποία είναι πιθανές τιμές του γνωρίσματος ThematicContent.

Για το πρώτο γνώρισμα έχουμε $fpsc(Contributors)=1$, δηλαδή η ασάφεια του γνωρίσματος θα ληφθεί υπόψη στην απόδοση ομοιότητας, ενώ για το δεύτερο είναι $fpsc(RelevantExternalResources)=0$, δηλαδή αγνοείται η ασάφεια της ιδιότητας αυτής. Θα οριστεί επομένως ένα case retrieval context το οποίο θα αποτελείται από τα παραπάνω fpsc. Η συνάρτηση συνάθροισης που αποδίδεται στο συγκεκριμένο case retrieval context είναι η Average.

Για το τρίτο γνώρισμα του ThematicContent τα πεδία απόδοσης νοηματικού πλαισίου που δεν είναι ίσα με -1 είναι τα εξής:

- $frhrsc(isPartOfSpecification) = 1$
- $farsc(participatesInProcess) = 0$
- $frarsc(isPerformedByParticipant) = -0.2$

- $farsc(isPerformedInTheContextOfProcess) = -0.5$
- $frarsc(isPerformedInTheContextOfProcess) = -0.95$
- $farsc(regardsProcess) = -0.5$
- $frarsc(regardsProcess) = -0.7$
- $farsc(isFoundInSource) = -0.2$
- $frarsc(isFoundInSource) = -0.8$
- $fcrsc(participatesInProcess, isPartOfProcess) = 0$
- $farsc(ThematicContent) = 0$

Αξίζει να τονιστεί ότι η τελευταία τιμή στο γνώρισμα ThematicContent (θεωρούμενο ως συσχετιστική σχέση) πρακτικά δηλώνει ότι η ασάφεια του γνωρίσματος προσμετρείται όταν αποδίδεται ομοιότητα μεταξύ περιπτώσεων, παρόμοιο ρόλο δηλαδή με την τιμή frasc.

6.2 Ενέργειες στο GUI

Με βάση τον ορισμό του μοντέλου του ΔΕΣΜΗΕ θα παρουσιασθούν ακριβείς ενέργειες για τον ορισμό και ανάθεση τιμών στα παραπάνω χαρακτηριστικά. Οι όψεις που θα εμφανιστούν πάρθηκαν από τη λειτουργία του σεναρίου στο φυλλομετρητή ιστού (browser) Mozilla Firefox.

6.2.1 Δημιουργία Ασαφούς Οντολογίας Αγοράς Ενέργειας

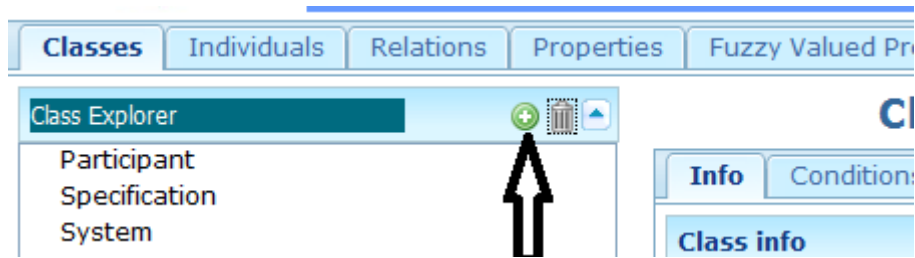
Κάνοντας κλικ στον Ontology Manager μεταφέρεται ο χρήστης στην καρτέλα όπου γίνεται ο ορισμός της δομής (schema) της ασαφούς οντολογίας πεδίου. Για τις ανάγκες του σεναρίου επομένως έχουμε:

6.2.1.1 Προσθήκη κλάσης-έννοιας

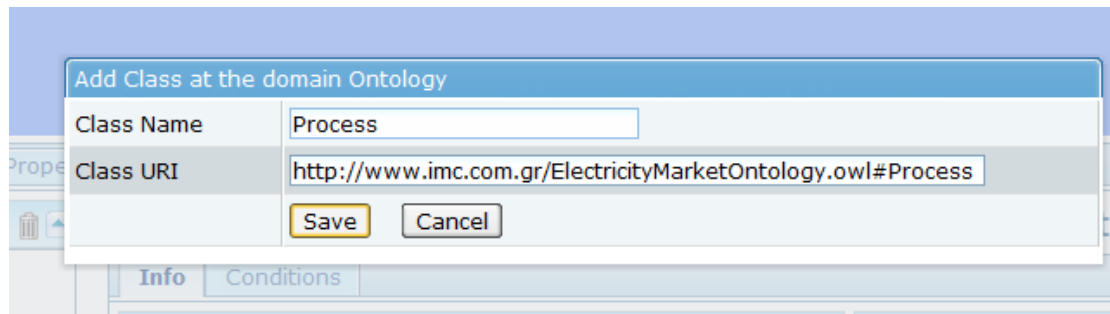
Ο χρήστης κάνει click στο εικονίδιο add που βρίσκεται στο panel Class Explorer της καρτέλας Classes, όπως φαίνεται και στην Εικόνα 6.1 και έπειτα του εμφανίζεται ένα νέο παράθυρο (molded window).

Στο παράθυρο αυτό μπορεί να προσθέσει μια καινούρια Class με το όνομά του και το uri του. Έχοντας ως παράδειγμα τις ανάγκες του σεναρίου ο χρήστης ορίζει την κλάση με όνομα Process, uri:

<http://www.imc.com.gr/ElectricityMarketOntology.owl#Process> και στη συνέχεια με το button Save (Εικόνα 6.2) επιτυγχάνεται η προσθήκη.

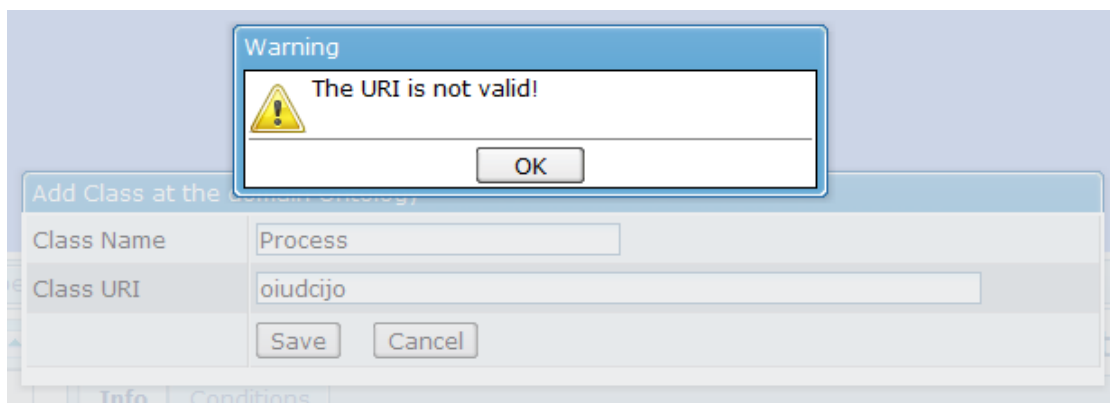


Εικόνα 6.1 Προσθήκη κλάσης στην οντολογία (1^ο βήμα)



Εικόνα 6.2 Προσθήκη κλάσης στην οντολογία (2^ο βήμα)

Στο σημείο αυτό τονίζεται ότι σε περίπτωση που δοθεί κάποιο μη έγκυρο uri εμφανίζεται ένα μήνυμα που το αναφέρει (Εικόνα 6.3) και δίνει στο χρήστη την ευκαιρία για νέο ορισμό uri. Το μήνυμα αυτό εμφανίζεται σε όλες τις περιπτώσεις που απαιτείται το uri.

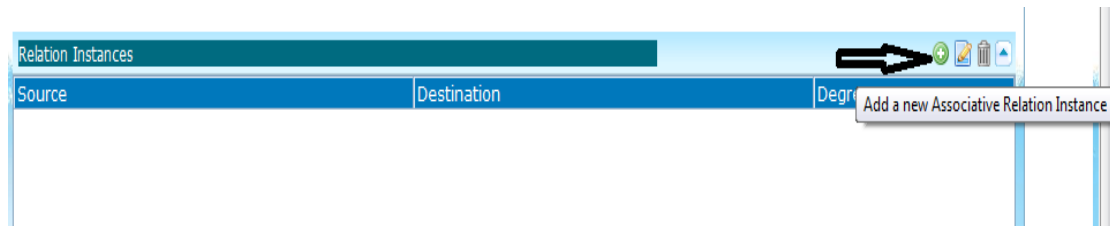


Εικόνα 6.3 Μήνυμα λάθους σε μη έγκυρο uri

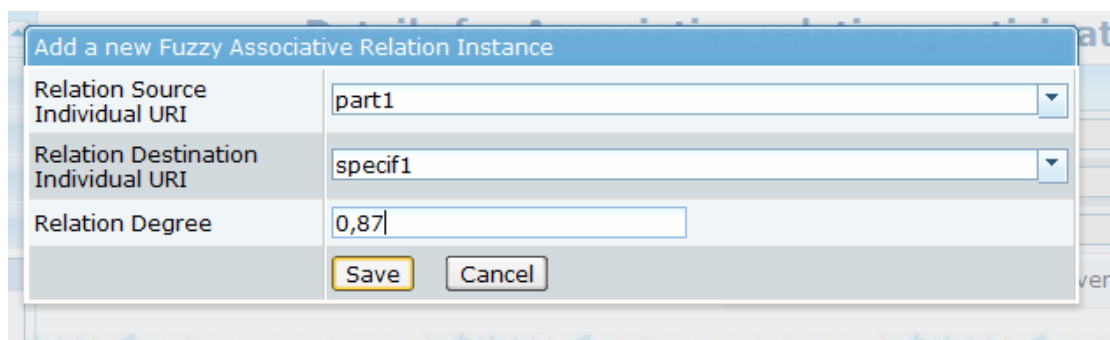
6.2.1.2 Προσθήκη στιγμιότυπου ασαφούς συσχετιστικής σχέσης

Έστω η ήδη ορισμένη ασαφής συσχετιστική σχέση *participatesInProcess* με πεδίο ορισμού την κλάση *Participant* και πεδίο τιμών την *Process*. Ο χρήστης θα μεταφερθεί στην καρτέλα *Relations* του *Ontology Manager* και θα επιλέξει τη σχέση αυτή. Στη συνέχεια κάνει click στο εικονίδιο *add* που βρίσκεται στο *panel Relation Instances* (Εικόνα 6.4) και έπειτα του

εμφανίζεται ένα νέο παράθυρο στο οποίο διαλέγει από δύο λίστες τα στιγμιότυπα (ήδη ορισμένα στην καρτέλα Individuals) που θα συνδέσει και το βαθμό της ασαφούς σύνδεσης αυτής (Εικόνα 6.5).

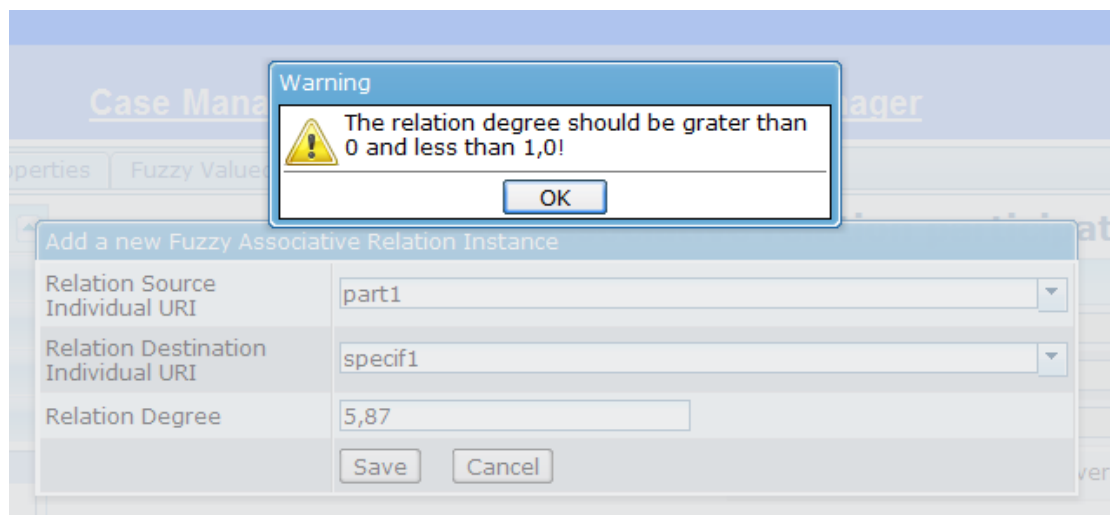


Εικόνα 6.4 Βήμα 1 για την προσθήκη στιγμιότυπου ασαφούς συσχετιστικής σχέσης



Εικόνα 6.5 2^ο βήμα για προσθήκη στιγμιότυπου ασαφούς συσχετιστικής σχέσης

Φυσικά γίνεται έλεγχος για το αν ο ασαφής βαθμός ανήκει στο διάστημα $[0, 1]$ και αν δεν ανήκει εμφανίζεται κατάλληλο μήνυμα (Εικόνα 6.6).



Εικόνα 6.6 Μήνυμα λάθους για μη έγκυρο ασαφή βαθμό

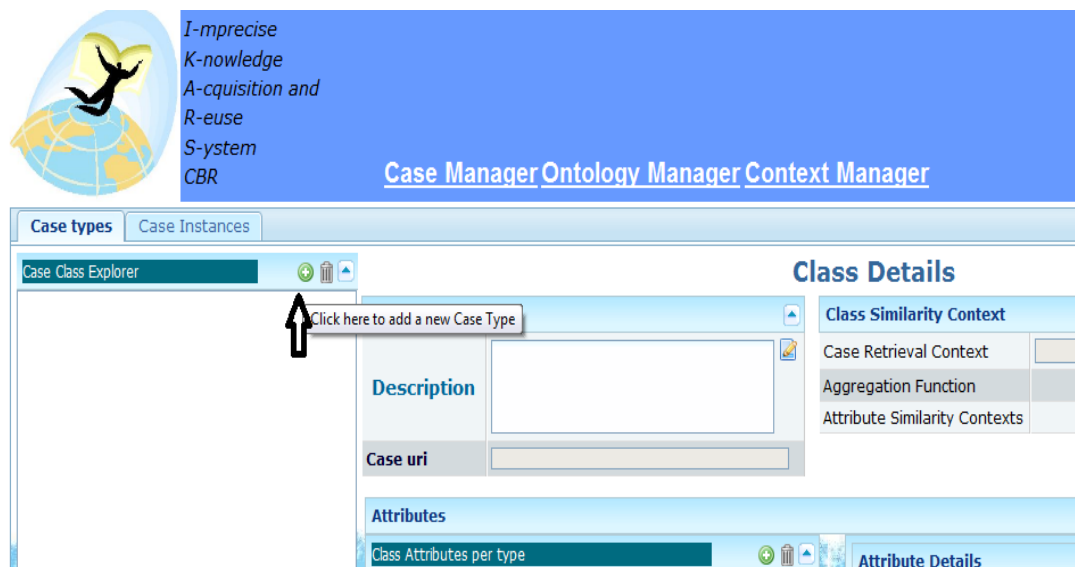
6.2.2 Δημιουργία Ασαφούς Οντολογίας Περιπτώσεων

Κάνοντας κλικ στον Case Manager μεταφέρεται ο χρήστης στην καρτέλα όπου γίνεται ο ορισμός της δομής (schema) όχι μόνο των τύπων των περιπτώσεων αλλά και των χαρακτηριστικών (attributes) τους. Για τις ανάγκες του σεναρίου επομένως έχουμε:

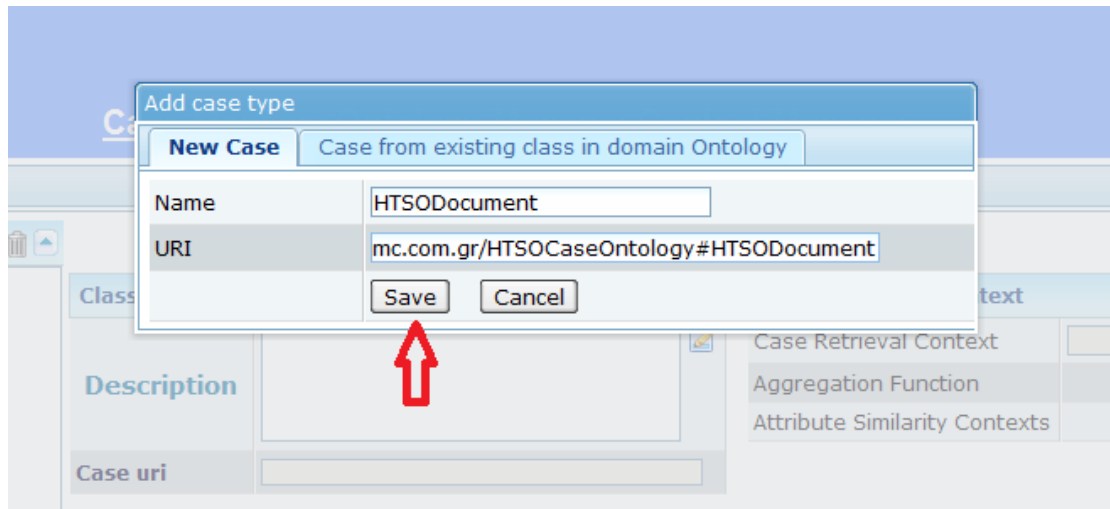
6.2.2.1 Προσθήκη τύπου περιπτώσεων (case type)

Ο χρήστης κάνει click στο εικονίδιο add που βρίσκεται στο panel Case Class Explorer της καρτέλας Case Types, όπως φαίνεται και στην εικόνα 6.7 και έπειτα του εμφανίζεται ένα νέο παράθυρο.

Στο παράθυρο αυτό μπορεί να προσθέσει είτε ένα καινούριο Case με το όνομά και το uri του ή ένα Case που υπάρχει ήδη στην Domain Ontology, εικόνα 6.8.



Εικόνα 6.7 Βήμα πρώτο για προσθήκη τύπου περιπτώσεων



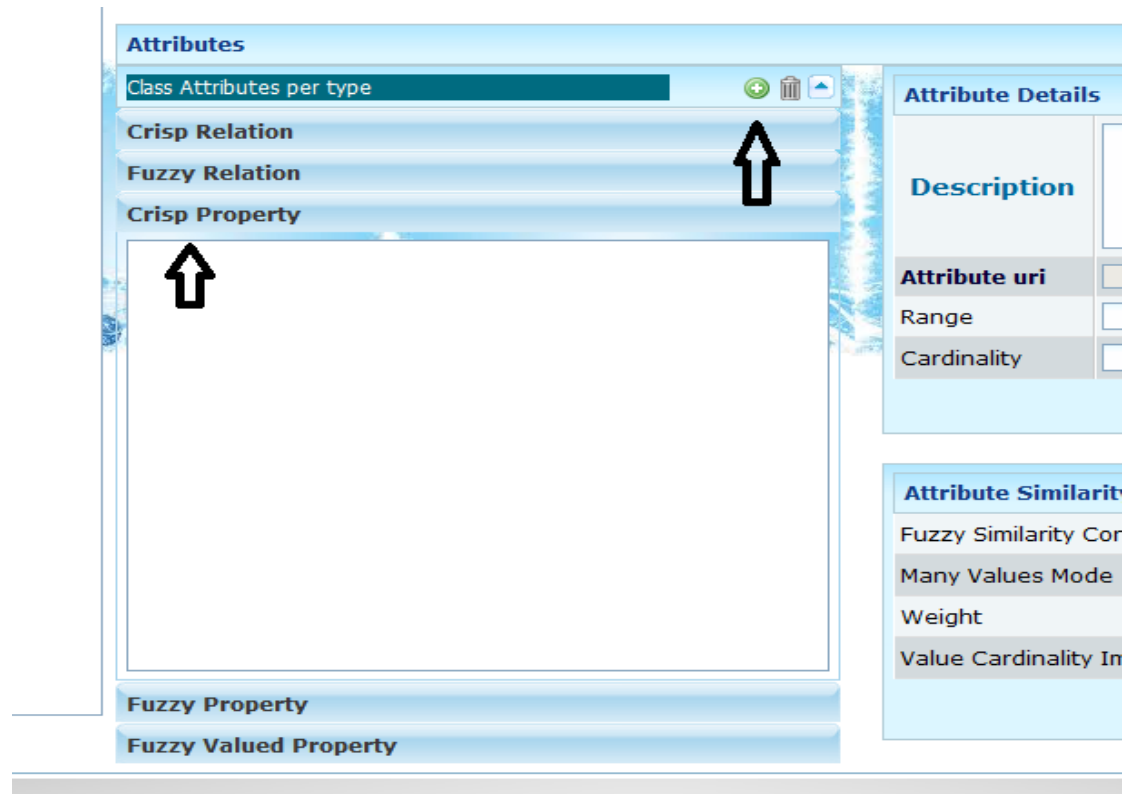
Εικόνα 6.8 Βήμα δεύτερο για προσθήκη τύπου περιπτώσεων

Στα πλαίσια του σεναρίου με όνομα: HTSODocument και uri:

<http://www.imc.com.gr/HTSOCaseOntology#HTSODocument> και πατώντας το κουμπί (button) Save επιτυγχάνεται εν τέλει η προσθήκη του τύπου περίπτωσης HTSODocument.

6.2.2.2 Προσθήκη γνωρισμάτων περιπτώσεων (case attributes)

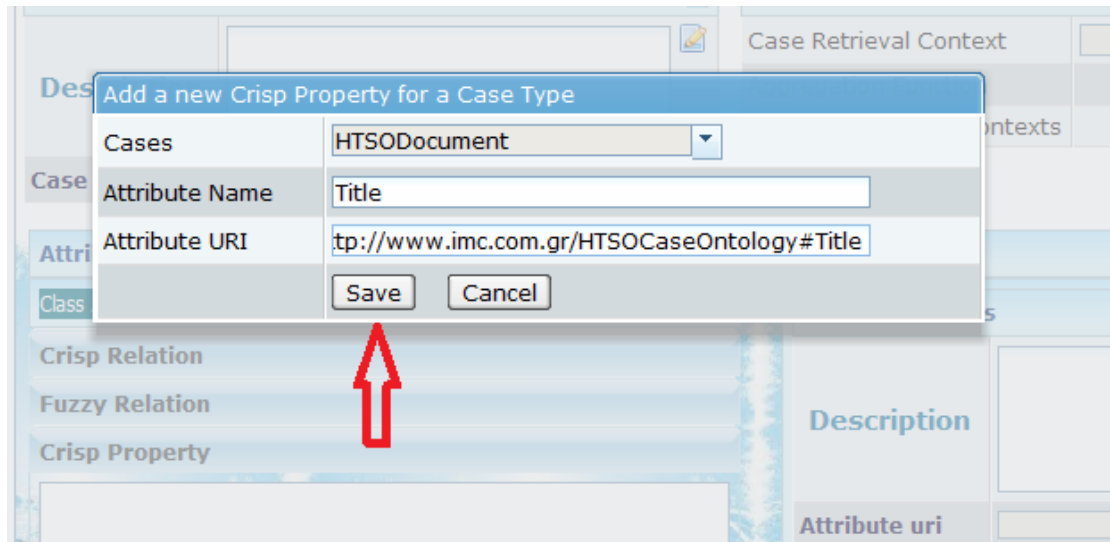
Ο χρήστης κάνει click στο εικονίδιο add που βρίσκεται στο panel Class Attributes per type αφού πρώτα βρεθεί στην καρτέλα εκείνη στην οποία θα ανήκει το γνώρισμα που θέλει να προσθέσει. Για παράδειγμα για να προσθέσει μια μη ασαφής ιδιότητα (Crisp Property) θα μεταφερθεί στην καρτέλα με όνομα Crisp Property και κάνοντας κλικ στο εικονίδιο add του εμφανίζεται ένα νέο παράθυρο, όπως φαίνεται στην εικόνα 6.9.



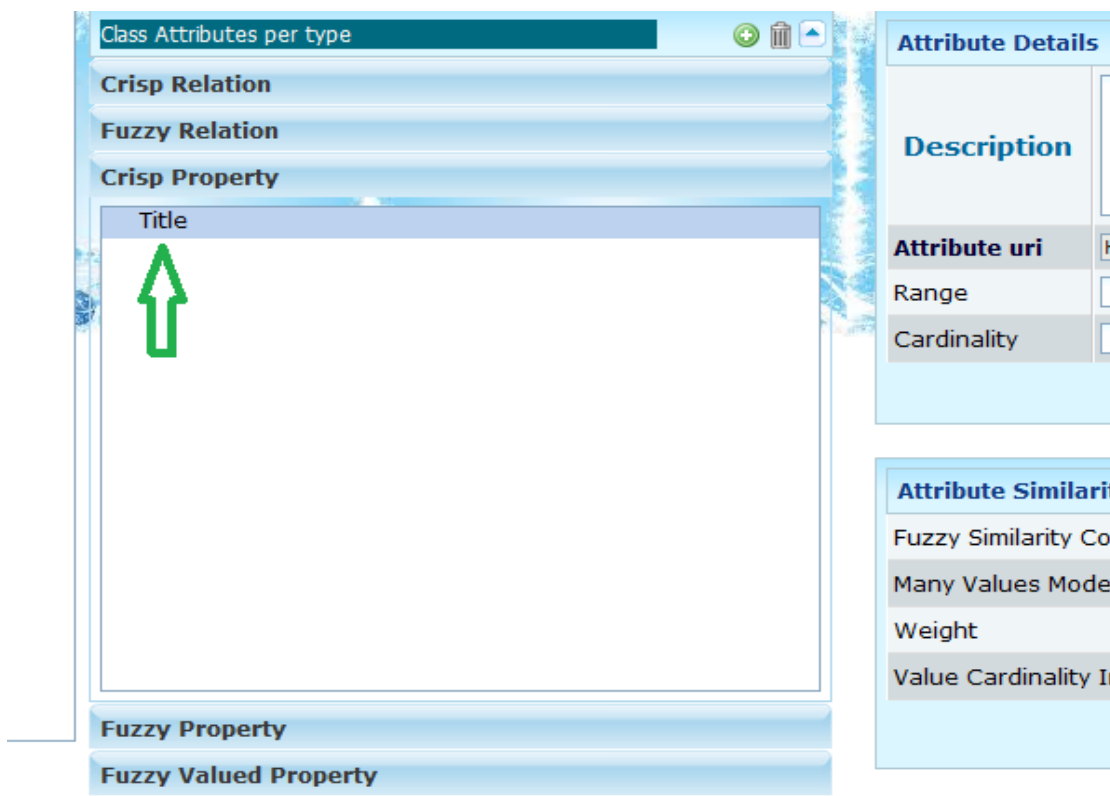
Εικόνα 6.9 Βήμα πρώτο για προσθήκη γνωρίσματος περιπτώσεων τύπου Crisp Attribute

Αν ο χρήστης έχει ήδη επιλέξει ένα Case Type θα του εμφανιστεί παράθυρο για προσθήκη ονόματος και uri του crisp property. Αν δεν έχει επιλεγεί Case Type τότε στο παράθυρο θα πρέπει εκτός του ονόματος και του uri να επιλέξει και συγκεκριμένο Case Type στο οποίο θα προστεθεί το crisp property.

Στο σενάριο χωρίς επιλογή του Case Type ο χρήστης θα διαλέξει από το combobox το HTSODocument. Έπειτα στα επόμενα πεδία γράφει όνομα: Title και uri: <http://www.imc.com.gr/HTSOCaseOntology#Title> και πατώντας το button Save, όπως φαίνεται στην εικόνα 6.10, επιτυγχάνεται εν τέλει η προσθήκη του Crisp Property Attribute Title στο Case Type HTSODocument και φαίνεται εν τέλει στην καρτέλα Crisp Property η εν λόγω προσθήκη (Εικόνα 6.11).



Εικόνα 6.10 Βήμα δεύτερο για προσθήκη γνωρίσματος περιπτώσεων τύπου Crisp Attribute



Εικόνα 6.11 Τελικό αποτέλεσμα προσθήκης γνωρίσματος περιπτώσεων τύπου Crisp Attribute

Με ανάλογο τρόπο θα γίνει και η προσθήκη των γνωρισμάτων Contributors, RelevantExternalResources, ThematicContent και PublicationDate.

6.2.3 Ορισμός Νοηματικού Πλαισίου Ανάκτησης

Κάνοντας κλικ στον Context Manager μεταφέρεται ο χρήστης στην καρτέλα όπου αποδίδεται σημασιολογικό πλαίσιο στις ιδιότητες (attributes) των τύπων περιπτώσεων και στις σχέσεις της οντολογίας πεδίου με στόχο την αποτελεσματική ανάκτηση ασαφούς γνώσης.

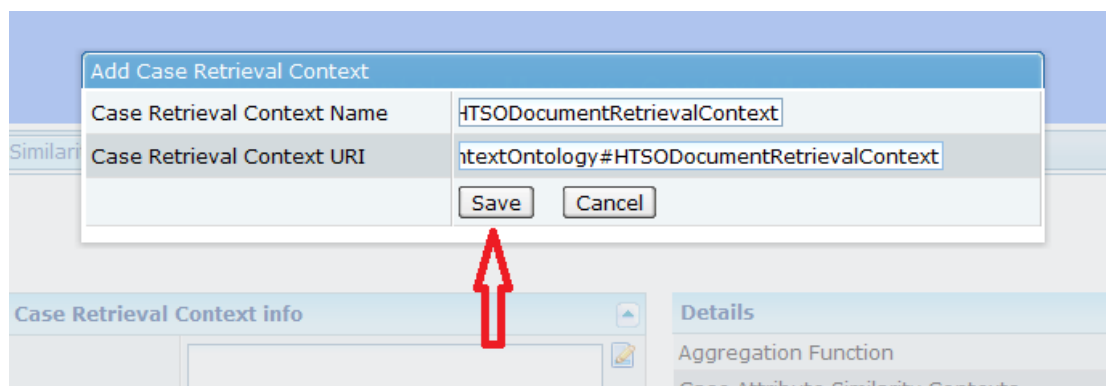
6.2.3.1 Δημιουργία Case Retrieval Context

Αρχικά θα βρεθεί ο χρήστης στην καρτέλα με label Case Retrieval Context. Ο χρήστης επιλέγοντας το εικονίδιο add που βρίσκεται στο panel Case Retrieval Context Explorer της καρτέλας Case Retrieval Context του εμφανίζεται ένα νέο παράθυρο. Στο παράθυρο αυτό προσθέτει ένα καινούριο Case Retrieval Context με το όνομά και το uri του.

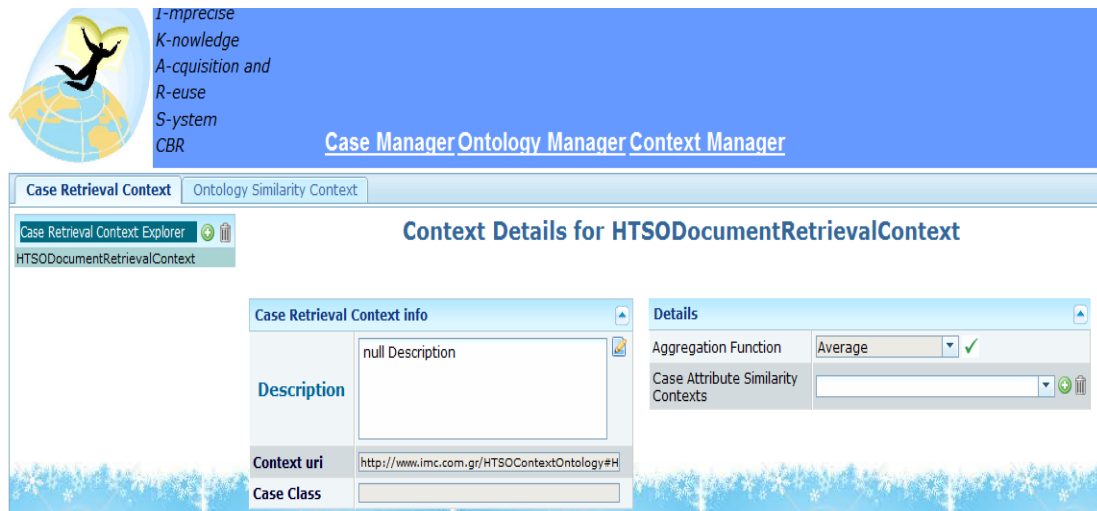
Άρα, με όνομα: HTSODocumentRetrievalContext, uri:

<http://www.imc.com.gr/HTSOContextOntology#HTSODocumentRetrievalContext> και πατώντας το button Save (Εικόνα 6.12) επιτυγχάνεται εν τέλει η εν λόγω προσθήκη.

Δημιουργείται έτσι ένα νέο Case Retrieval Context με τιμή στο πεδίο Attribute Aggregation Function: Average (Εικόνα 6.13).



Εικόνα 6.12 Δημιουργία νέου Case Retrieval Context

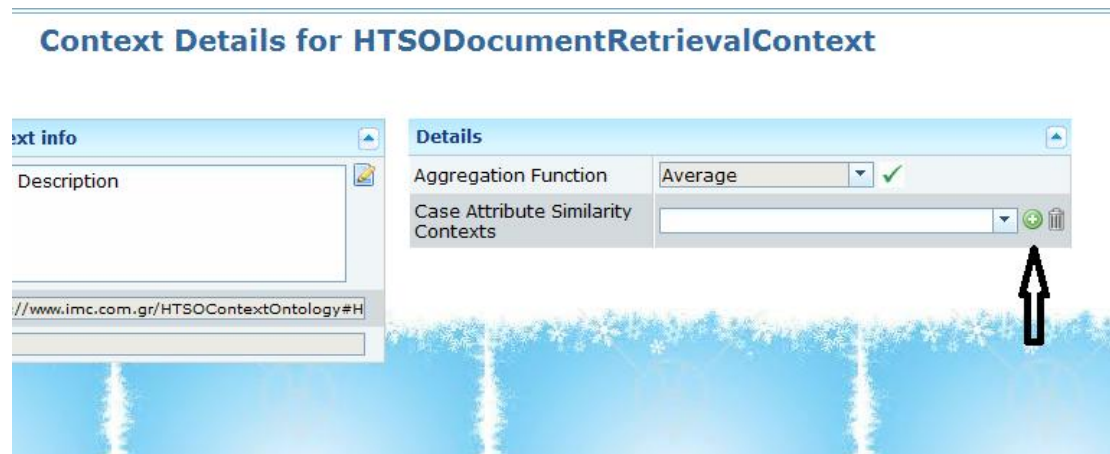


Εικόνα 6.13 Τελική εμφάνιση νέου Case Retrieval Context

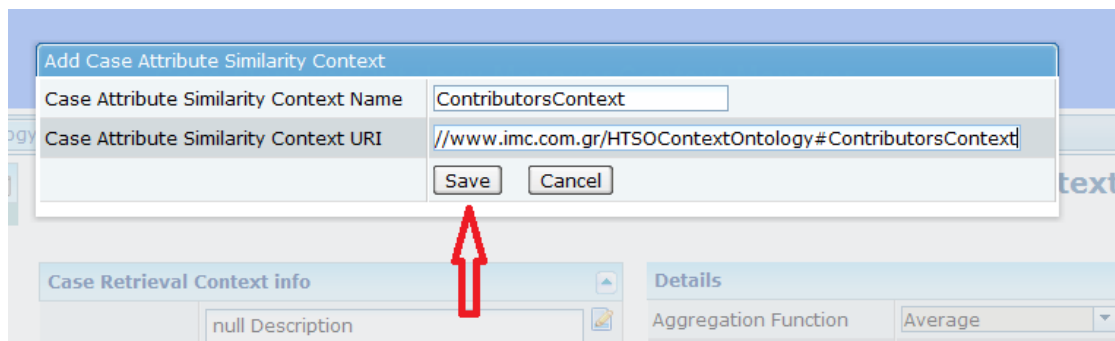
6.2.3.2 Προσθήκη Case Attribute Similarity Context

Παραμένοντας στην καρτέλα με label Case Retrieval Context, θα γίνει πρώτα η επιλογή του case retrieval context στο οποίο θα ανήκουν τα case attribute similarity contexts. Στο σενάριο ο χρήστης επιλέγει το Case Retrieval Context που δημιούργησε στην παράγραφο 6.2.3.1, με όνομα HTSODocumentRetrievalContext. Στη συνέχεια κάνοντας click στο εικονίδιο add που βρίσκεται στη σειρά με label Case Attribute Similarity Contexts του panel Details (Εικόνα 6.14) του εμφανίζεται ένα νέο παράθυρο. Στο παράθυρο αυτό μπορεί να προστεθεί ένα καινούριο Case Attribute Similarity Context με το όνομά και το uri του. Σύμφωνα με την πλοκή του σεναρίου ο χρήστης δημιουργεί για το γνώρισμα Contributors ένα Case Attribute Similarity Context. Άρα, με όνομα: ContributorsContext, uri:

<http://www.imc.com.gr/HTSOContextOntology#ContributorsContext> και πατώντας το button Save (Εικόνα 6.15) επιτυγχάνεται εν τέλει η εν λόγω προσθήκη.

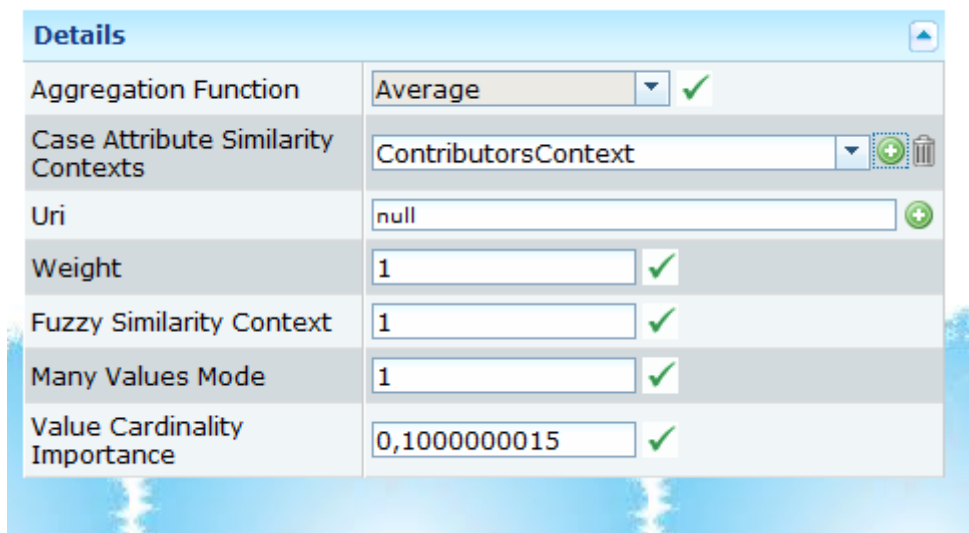


Εικόνα 6.14 Βήμα 1^ο για προσθήκη Case Attribute Similarity Context



Εικόνα 6.15 Βήμα 2^ο για προσθήκη Case Attribute Similarity Context

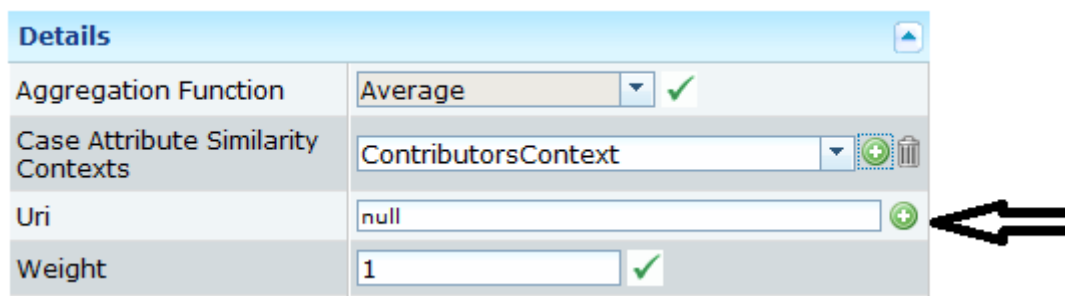
Σαν αποτέλεσμα της παραπάνω προσθήκης θα εμφανιστούν και οι παράμετροι του συγκεκριμένου Case Attribute Similarity Context με τις Default τιμές τους, οι οποίες είναι: URI = null, Weight = 1, Many Values Mode = 1, Fuzzy Similarity Context = 1 και Value Cardinality Importance = 0.1 (Εικόνα 6.16).



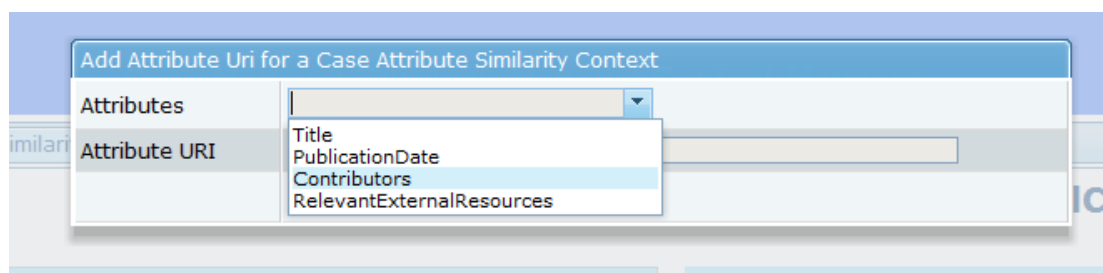
Εικόνα 6.16 Τελικό αποτέλεσμα προσθήκης Case Attribute Similarity Context με εμφάνιση των παραμέτρων του

Για την αλλαγή τιμής μιας παραμέτρου (αφού πρώτα έχει επιλεγθεί ένα συγκεκριμένο Case Attribute Similarity Context) ο χρήστης τοποθετεί την τιμή στο πεδίο με label την παράμετρο που θέλει να αλλάξει και τέλος επιλέγει το εικονίδιο δίπλα από το πεδίο. Το πεδίο Fuzzy Similarity Context δείχνει την τιμή $f_{frac}()$. Για το γνώρισμα Contributors επειδή είναι $f_{frac}(\text{Contributors})=1$ ήδη αφού είναι η default τιμή δε γίνεται κάποια αλλαγή. Σε περίπτωση πάντως που εισάγει τιμή μη έγκυρη θα του εμφανισθεί ανάλογο μήνυμα και δε θα συντελεστεί αλλαγή στην τιμή της παραμέτρου.

Για τη σύνδεση του case attribute Contributors με το Case Attribute Similarity Context με όνομα ContributorsContext που μόλις δημιουργήθηκε, αφού κάνει click στο εικονίδιο add του πεδίου Uri (Εικόνα 6.17) του εμφανίζεται ένα νέο παράθυρο που περιέχει μια λίστα (combobox) στην οποία ο χρήστης επιλέγει συγκεκριμένο Case Attribute από τα ήδη ορισμένα της Case Ontology (Εικόνα 6.18)για τη σύνδεση και έπειτα πατάει Save. Με την επιλογή τώρα από τη λίστα του Contributors επιτυγχάνεται η εν λόγω σύνδεση.

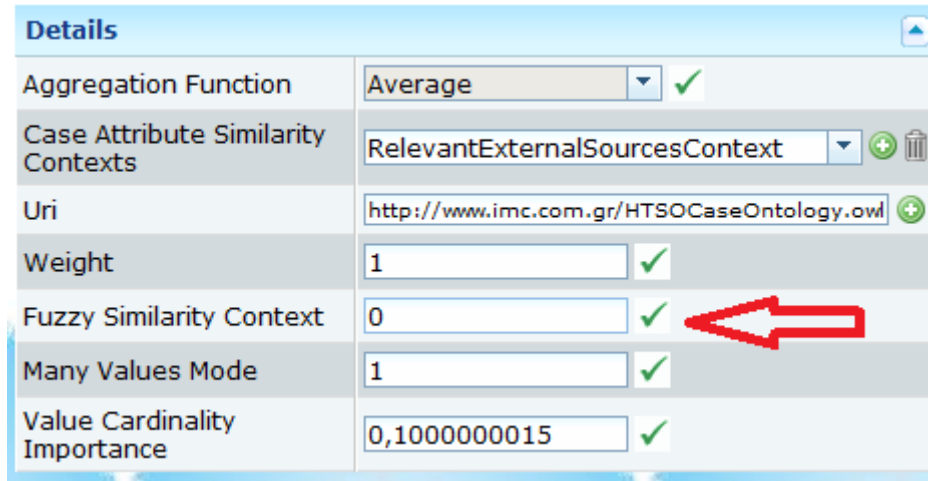


Εικόνα 6.17 Βήμα 1^ο σύνδεσης Case Attribute Similarity Context με Case Attribute



Εικόνα 6.18 Βήμα 2^ο σύνδεσης Case Attribute Similarity Context με Case Attribute

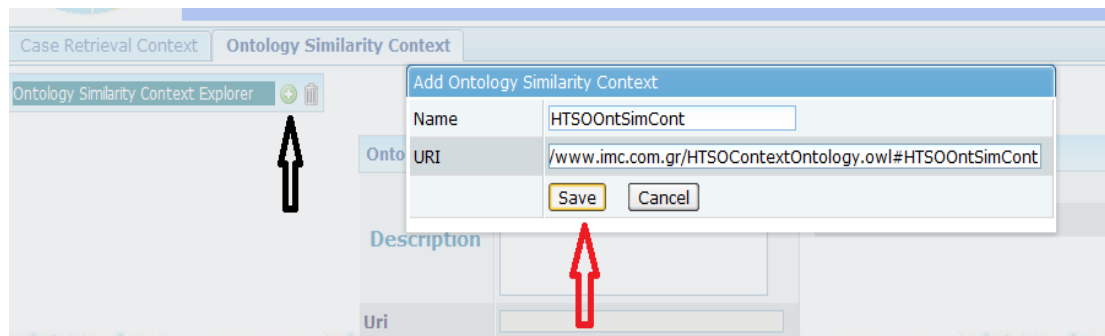
Με ανάλογο τρόπο γίνεται και η δημιουργία για το γνώρισμα RelevantExternalSources ένα Case Attribute Similarity Context με όνομα RelevantExternalSourcesContext και τιμή $f_{\text{rasc}}(\text{RelevantExternalResources})=0$, πράγμα που σημαίνει ότι ο χρήστης αφού δημιουργήσει το RelevantExternalSourcesContext θα γράψει στο πεδίο Fuzzy Similarity Context την τιμή 0 και θα πατήσει το εικονίδιο tick (Εικόνα 6.19).



Εικόνα 6.19 Αλλαγή πεδίου Fuzzy Similarity Context

6.2.3.3 Δημιουργία Ontology Similarity Context

Ο χρήστης θα βρεθεί στην καρτέλα με label Ontology Similarity Context. Ο χρήστης επιλέγοντας το εικονίδιο add που βρίσκεται στο panel Ontology Similarity Context Explorer του εμφανίζεται ένα νέο παράθυρο. Στο παράθυρο αυτό προσθέτει ένα καινούριο Ontology Similarity Context με το όνομά και το uri του. Άρα με όνομα HTSOOntSimCont, uri: http://www.imc.com.gr/HTSOContextOntology.owl#HTSOOntSimCont (Εικόνα 6.20) και πατώντας το button Save επιτυγχάνεται εν τέλει η εν λόγω προσθήκη.

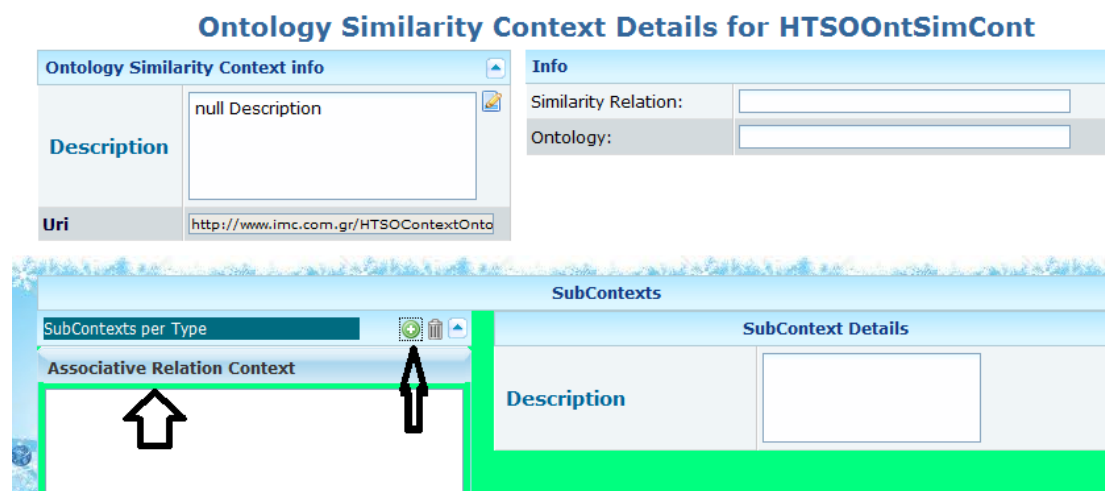


Εικόνα 6.20 Δημιουργία Ontology Similarity Context

6.2.3.4 Προσθήκη Fuzzy Associative Relation Context

Αφού πρώτα επιλεγθεί ένα ontology similarity context στο οποίο θα γίνει η προσθήκη του Fuzzy Associative Relation Context, ο χρήστης κάνει click στο εικονίδιο add που βρίσκεται στο panel SubContexts per type, με την προϋπόθεση φυσικά να βρίσκεται στην καρτέλα Fuzzy Associative Relation Context του panel SubContexts per type (Εικόνα 6.21). Έπειτα του εμφανίζεται ένα νέο παράθυρο στο οποίο αν ο χρήστης έχει ήδη επιλέξει

ένα Ontology Similarity Context θα του εμφανιστεί παράθυρο για προσθήκη ονόματος και uri του Fuzzy Associative Relation, ενώ αν δεν έχει επιλεγθεί Ontology Similarity Context τότε στο παράθυρο θα πρέπει εκτός του ονόματος και του uri να επιλέξει και συγκεκριμένο Ontology Similarity Context στο οποίο θα προστεθεί το Fuzzy Associative Relation Context.



Εικόνα 6.21 Πρώτο βήμα για δημιουργία Fuzzy Associative Relation Context

Στα πλαίσια του σεναρίου για την ασαφή συσχετιστική σχέση *participatesInProcess* προστίθεται στο Ontology Similarity Context με όνομα HTSOntSimCont, που δημιουργήθηκε στην παράγραφο 6.2.3.3, ένα Fuzzy Associative Relation Context με όνομα *participatesInProcessAssocRelContext* και uri:

<http://www.imc.com.gr/HTSOContextOntology.owl#participatesInProcessAssocRelContext>.

Με την επιλογή ενός Fuzzy Associative Relation Context θα εμφανιστούν στο panel SubContext Details οι παράμετροί του με default τιμές: FARSC = -1, FRARSC = -1, Context Associative Relation = κενό και Context Associative Relation Fuzziness = crisp. Οι παράμετροι αυτοί αρχικοποιούνται κατά τη δημιουργία ενός Fuzzy Relation Context.

Για την αλλαγή τιμής μιας παραμέτρου(αφού πρώτα έχει επιλεγθεί ένα συγκεκριμένο Fuzzy Associative Relation) ο χρήστης τοποθετεί την τιμή στο πεδίο με label την παράμετρο που θέλει να αλλάξει και τέλος επιλέγει το εικονίδιο tick δίπλα από το πεδίο. Σε περίπτωση που εισάγει τιμή μη έγκυρη θα του εμφανισθεί ανάλογο μήνυμα και δε θα συντελεστεί αλλαγή στην τιμή της παραμέτρου. Στο σενάριο δίνεται η τιμή *farsc(participatesInProcess)* = 0, οπότε στο πεδίο FARSC γράφει ο χρήστης την τιμή 0 και έπειτα επιλέγει το tick (Εικόνα 6.22), ενώ επειδή η σχέση *participatesInProcess* είναι ασαφής στο πεδίο Context Associative Relation Fuzziness θα επιλέξει την τιμή fuzzy από τη λίστα και θα κάνει click στο tick (Εικόνα 6.23).

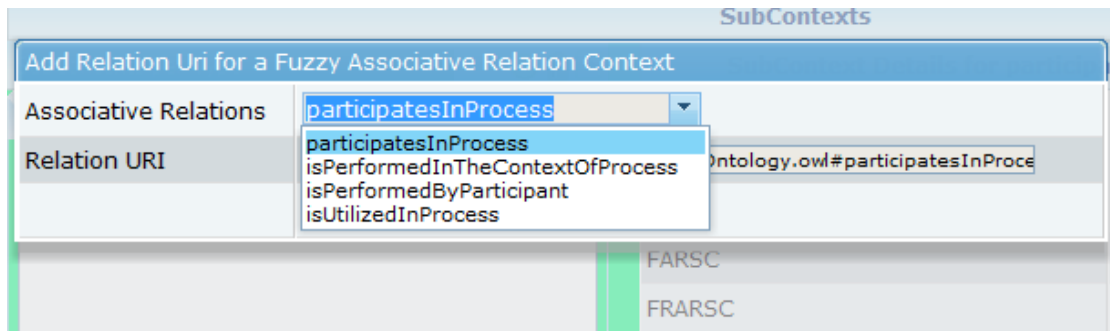
SubContext Details for participatesInProcessAssocRelContext	
Description	
FARSC	0 ✓
FRARSC	-1 ✓
Context Associative Relation	
Context Associative Relation Fuzziness	crisp ✓

Εικόνα 6.22 Αλλαγή πεδίου FARSC

SubContext Details for participatesInProcessAssocRelContext	
Description	
FARSC	0 ✓
FRARSC	-1 ✓
Context Associative Relation	participatesInProcess +
Context Associative Relation Fuzziness	crisp ✓

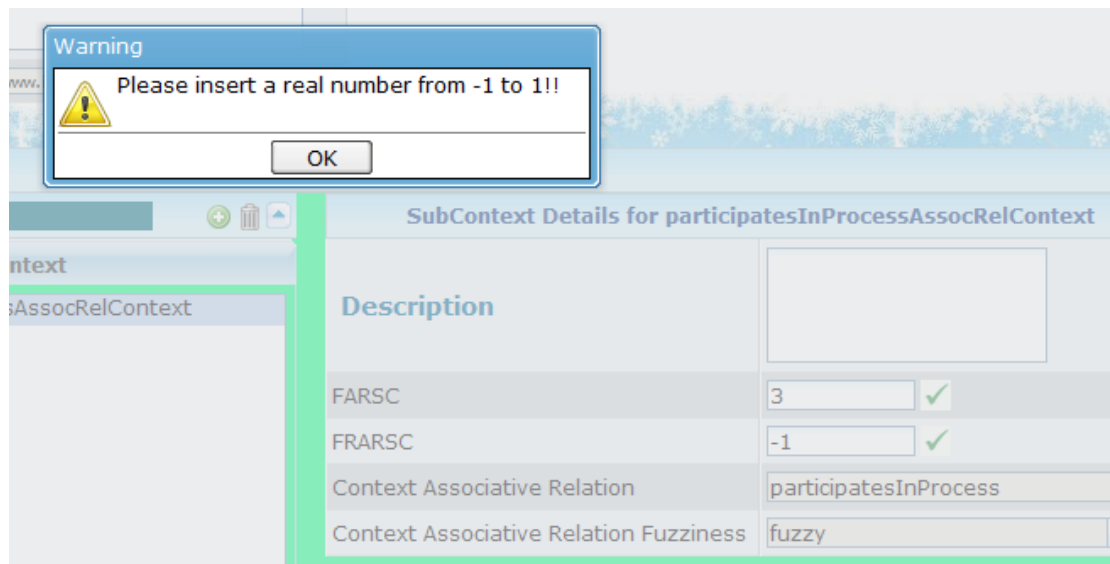
Εικόνα 6.23 Αλλαγή πεδίου Context Associative Relation Fuzziness

Για τη σύνδεση ενός Fuzzy Associative Relation Context ο χρήστης αφού επιλέξει το εικονίδιο add του πεδίου Context Associative Relation εμφανίζεται ένα παράθυρο που περιέχει μια λίστα (combobox) στην οποία ο χρήστης επιλέγει συγκεκριμένο Fuzzy Associative Relation από τα ήδη ορισμένα της οντολογίας πεδίου. Έτσι, για τη σύνδεση του participatesInProcessAssocRelContext με τη συσχετιστική σχέση participatesInProcess αφού γίνει η παραπάνω διαδικασία (Εικόνα 6.24) θα κάνει click στο Save του παραθύρου.



Εικόνα 6.24 Σύνδεση Fuzzy Associative Relation Context με Associative Relation

Σε περίπτωση που εισάγει τιμή μη έγκυρη σε κάποιο πεδίο θα του εμφανισθεί ανάλογο μήνυμα και δε θα συντελεστεί αλλαγή στην τιμή της παραμέτρου. Αν για παράδειγμα θέσει την τιμή 3 στο πεδίο FARSC θα εμφανιστεί το μήνυμα της Εικόνας 6.25, ενώ δε θα συντελεστεί αλλαγή στην τιμή του πεδίου.



Εικόνα 6.25 Μήνυμα προειδοποίησης για λάθος τιμή

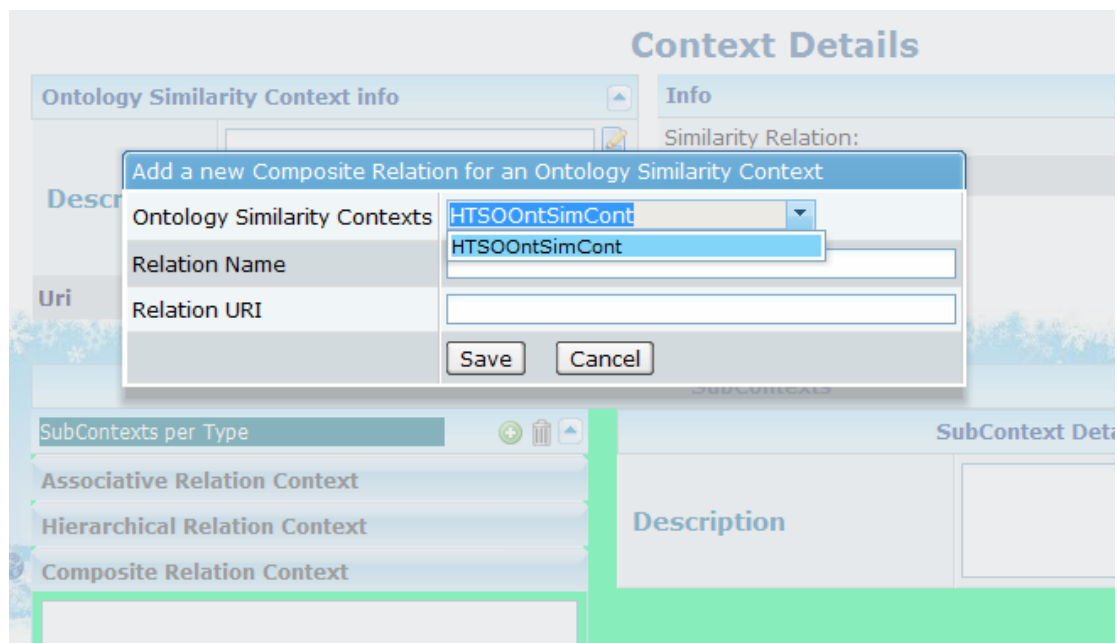
Με αντίστοιχο τρόπο δημιουργούνται Fuzzy Associative Relation Context για τις συσχετιστικές σχέσεις `isPerformedByParticipant`, `isPerformedInTheContextOfProcess`, `regardsProcess` και `isFoundInSource`, ενώ δημιουργείται και για το γνώρισμα περιπτώσεων `ThematicContent`.

6.2.3.5 Προσθήκη Fuzzy Hierarchical Relation Context

Ανάλογες με τις ενέργειες της παραγράφου 6.2.3.4 είναι και οι ενέργειες για τη δημιουργία Fuzzy Hierarchical Relation Context για την ιεραρχική σχέση `isPartOfSpecification` και την αλλαγή τιμών στις παραμέτρους του.

6.2.3.6 Προσθήκη Composite Relation Context

Ο χρήστης κάνει click στο εικονίδιο add της καρτέλας Composite Relation Context που βρίσκεται στο panel SubContexts per type. Αν ο χρήστης έχει ήδη επιλέξει ένα Ontology Similarity Context θα του εμφανιστεί παράθυρο για προσθήκη ονόματος και uri του Fuzzy Composite Relation. Αν δεν έχει επιλεγεί Ontology Similarity Context τότε στο παράθυρο θα πρέπει εκτός του ονόματος και του uri να επιλέξει και συγκεκριμένο Ontology Similarity Context στο οποίο θα προστεθεί το Fuzzy Composite Relation. Για την προσθήκη τώρα ενός Composite Relation Context που να συνδέει τις σχέσεις participatesInProgress και isPartOfProcess ο χρήστης έστω ότι δεν επιλέγει πρώτα το Ontology Similarity Context στο οποίο θα προστεθεί. Τότε θα εμφανιστεί το παράθυρο της Εικόνας 6.26, στο οποίο παράθυρο θα επιλέξει από τη λίστα το HTSOOntSimCont και με όνομα CompositeRelPartInIsPart, uri: <http://www.imc.com.gr/HTSOContextOntology.owl#CompositeRelPartInIsPart> και πατώντας Save θα γίνει η εν λόγω προσθήκη.



Εικόνα 6.26 Δημιουργία Composite Relation Context χωρίς επιλογή Ontology Similarity Context

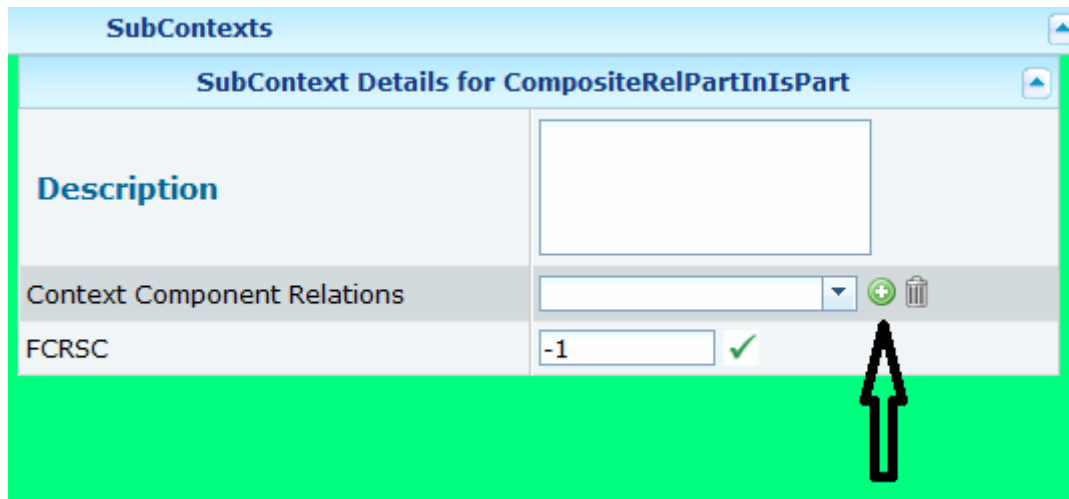
Με την παραπάνω αρχικοποιείται η παράμετρος του Composite Relation Context με Default τιμή: $fcpsc = -1$.

Αφού γίνει επιλογή του Composite Relation Context ο χρήστης πρέπει να ορίσει τα πλαίσια των σχέσεων που την αποτελούν. Τότε με click στο εικονίδιο add που βρίσκεται στη σειρά με label Context Component Relations του panel SubContext Details (Εικόνα 6.27),

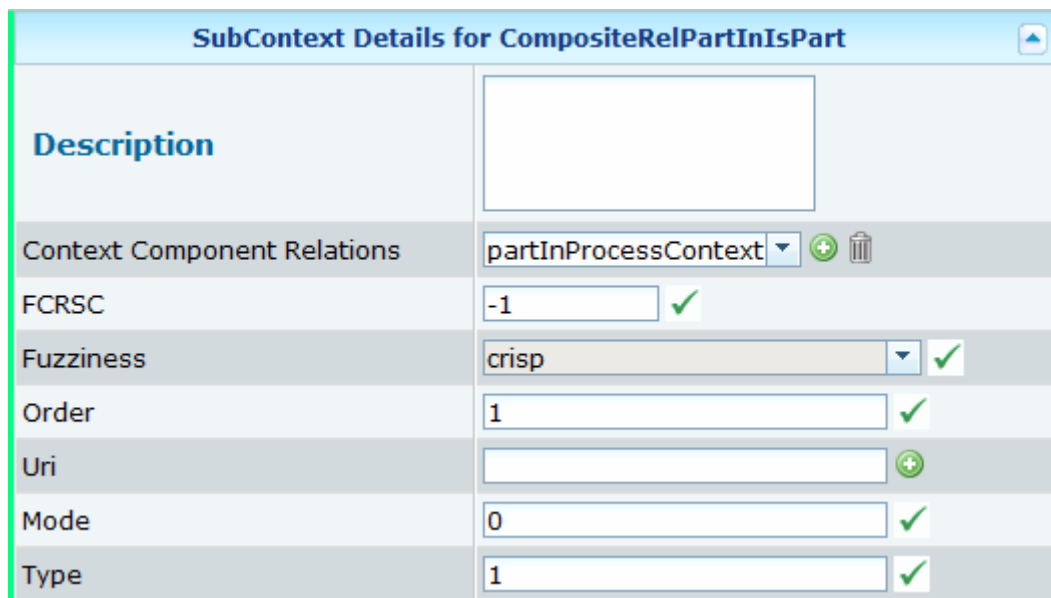
του εμφανίζεται ένα νέο παράθυρο για την προσθήκη Context Component Relation, στο οποίο βάζει το όνομα: partInProcessContext, το uri:

<http://www.imc.com.gr/HTSOContextOntology.owl#partInProcessContext> και πατάει Save.

Έτσι, δημιουργείται ένα Context Component Relation ενός Composite Relation Context, ενώ αρχικοποιούνται οι παράμετροι με Default τιμές: Relation URI = κενό, Relation Order = 1, Relation Mode = 0, Relation Fuzziness = crisp και Relation Type = 1 (Εικόνα 6.28).



Εικόνα 6.27 Βήμα 1^ο για προσθήκη Context Component Relation



Εικόνα 6.28 Default τιμές Context Component Relation

Για την αλλαγή τιμής μιας παραμέτρου(αφού πρώτα έχει επιλεγθεί ένα συγκεκριμένο Context Component Relation) ο χρήστης τοποθετεί την τιμή στο πεδίο με label την παράμετρο που θέλει να αλλάξει και τέλος επιλέγει το εικονίδιο δίπλα από το πεδίο. Για να συνδεθεί

επομένως το συγκεκριμένο Context Component Relation με μια υπάρχουσα σχέση της ασαφούς οντολογίας, δηλαδή το πλαίσιο partInProcessContext με τη σχέση participatesInProcess, ακολουθείται η διαδικασία της παραγράφου 6.2.3.4 για τη σύνδεση συσχετιστικής σχέσης με πλαίσιο.

Για την ολοκλήρωση του πλαισίου CompositeRelPartInIsPart, ο χρήστης δημιουργεί με αντίστοιχο τρόπο ένα Context Component Relation που το συνδέει με τη σχέση isPartOfProcess.

7

Επίλογος

Στόχος της διπλωματικής αποτελούσε η υλοποίηση ενός γραφικού περιβάλλοντος υποστήριξης του πλαισίου IKARUS-CBR. Στο σημείο αυτό θα εξετασθεί αν και κατά πόσο αυτός ο στόχος κατέστη επιτυχής, ενώ στο τέλος θα παρουσιαστούν μελλοντικές επεκτάσεις με σκοπό την περαιτέρω βελτίωση του GUI.

7.1 Σύνοψη και συμπεράσματα

Αρχικά κατέστη απαραίτητη για το σχεδιασμό του GUI η πλήρης κατανόηση του θεωρητικού μοντέλου του IKARUS-CBR όπως παρουσιάστηκε στα [1] και [2]. Στη συνέχεια τέθηκαν οι βάσεις για το σχήμα αναπαράστασης ασαφούς γνώσης όπως αυτό παρουσιάζεται στο Κεφάλαιο 3, το οποίο υλοποιήθηκε στο επόμενο κεφάλαιο με γνώμονα την εύκολη και άμεση εξοικείωση του χρήστη με ένα σύστημα αναπαράστασης γνώσης.

Το GUI που υλοποιήθηκε επομένως προσφέρει με απλό και απέριττο τρόπο τη δυνατότητα στο χρήστη για αναπαράσταση ασαφούς γνώσης με τον ακριβή τρόπο που περιγράφεται στα πλαίσια του IKARUS-CBR. Ο χρήστης μοντελοποιεί τη γνώση ενός συγκεκριμένου γνωστικού πεδίου ορίζοντας το σχήμα της οντολογίας πεδίου με τη δημιουργία κλάσεων-εννοιών και σχέσεων μεταξύ αυτών. Στη συνέχεια ορίζει τύπους περιπτώσεων και σχέσεις ή ιδιότητες αυτών προδιαγράφοντας έτσι το σχήμα της οντολογίας ασαφών περιπτώσεων. Έπειτα προσδιορίζει τα πλαίσια ομοιότητας των τύπων ασαφών περιπτώσεων, των σχέσεων

και ιδιοτήτων τους και των οντολογικών σχέσεων έχοντας ως απώτερο στόχο την ανάκτηση γνώσης. Στο τέλος της διαδικασίας αναπαράστασης γνώσης ο χρήστης δημιουργεί στιγμιότυπα όχι μόνο των κλάσεων και των τύπων περιπτώσεων αλλά και των σχέσεων και ιδιοτήτων που αυτά διαθέτουν. Πιο συγκεκριμένα ορίζει στιγμιότυπα κλάσεων και έπειτα τα συσχετίζει μέσω στιγμιοτυποποίησης ασαφών συσχετιστικών και ιεραρχικών σχέσεων. Η τελευταία διαδικασία γίνεται απλά με τον προσδιορισμό από το χρήστη του βαθμού ασάφειας της σχέσης που συνδέει δύο συγκεκριμένα στιγμιότυπα.

Ο σχεδιασμός του GUI έγινε ως εφαρμογή βασισμένη στο διαδίκτυο (web-based application) καθώς έτσι τα δεδομένα των οντολογιών (κλάσεις, σχέσεις, στιγμιότυπα, κτλ.) αποθηκεύονται σε απομακρυσμένους εξυπηρετητές (web servers) και έτσι καθίστανται προσβάσιμα ανεξαρτήτου σταθμού εργασίας του χρήστη. Έτσι χρησιμοποιήθηκε το ZK Framework αφού όχι μόνο επιτρέπει στο χρήστη ασύγχρονη επικοινωνία με τους servers (AJAX τεχνολογία) αλλά και προσφέρει στο σχεδιαστή πολύτιμα και εύχρηστα εργαλεία για το σχεδιασμό διεπιφανειών χρήστη. Επομένως ο χρήστης αναπαριστά ασαφής γνώση με απλό, εύχρηστο και γρήγορο τρόπο.

7.2 Μελλοντικές επεκτάσεις

Σίγουρα υπάρχουν αρκετές προσθήκες που κρίνονται απαραίτητες για την ολοκλήρωση του GUI ώστε να υποστηρίζει όλες τις λειτουργίες του πλαισίου IKARUS-CBR και να ανταπεξέρχεται πλήρως σε όλες τις ανάγκες και απαιτήσεις των διαφορετικών χρηστών. Σαφής διαχωρισμός πρέπει να γίνει σε ορισμένες διαφορετικές ανάγκες των ομάδων χρηστών καθώς για παράδειγμα οι απλοί διαχειριστές του συστήματος δε θα πρέπει να εμπλέκονται σε διαδικασίες μοντελοποίησης και αλλαγής του σχήματος γνώσης, ενέργειες που απασχολούν τους μηχανικούς γνώσης. Ο διαχωρισμός αυτός θα γίνεται με ανάθεση λογαριασμών χρήστη στις παραπάνω ομάδες με τους οποίους θα συνδέονται στην εφαρμογή.

Ως προς την κατεύθυνση της αναπαράστασης γνώσης δεν έχουν υλοποιηθεί οι καρτέλες του Ontology Manager που αφορούν τις ασαφώς αποτιμώμενες ιδιότητες και τις ασαφείς γλωσσικές μεταβλητές παρέχοντας λειτουργίες για σαφή ορισμό τους όπως αυτές έχουν ήδη περιγραφεί στα Κεφάλαια 2 και 3.

Επίσης, μπορούν να προστεθούν λειτουργίες έχοντας ως στόχο την ανάκτηση γνώσης, καθώς ενώ ο χρήστης ορίζει το πλαίσιο ομοιότητας των περιπτώσεων και των σχέσεών τους, δεν του παρέχεται η δυνατότητα για σύγκριση περιπτώσεων και αναζήτηση όμοιων περιπτώσεων.

Έτσι, κρίνεται απαραίτητη η προσθήκη πεδίων με στόχο την αναζήτηση όμοιων περιπτώσεων έχοντας ως βάση μία συγκεκριμένη περίπτωση με στόχο την ανάκτησή της περίπτωσης με το μεγαλύτερο βαθμό ομοιότητας. Προς την ίδια κατεύθυνση χρειάζεται η λειτουργία της άμεσης σύγκρισης μεταξύ στιγμιότυπων περιπτώσεων έχοντας ως αποτέλεσμα το βαθμό ομοιότητάς τους.

Ως προς την ανάγκη δημιουργίας και συντήρησης στους web servers πολλών και διαφορετικών ασαφών οντολογιών πεδίου και περιπτώσεων από διαφορετικούς χρήστες πρέπει ο κάθε χρήστης να διαθέτει λογαριασμό σύνδεσης. Έτσι θα γίνει διαχωρισμός μεταξύ των απλών επισκεπτών στο GUI και σε αυτούς που διαθέτουν λογαριασμό και τη δυνατότητα ορισμού οντολογιών. Επίσης θα πρέπει να παρέχεται η δυνατότητα για συμμετοχή πολλών διαφορετικών χρηστών στη διαδικασία ορισμού οντολογιών πεδίου και ασαφών περιπτώσεων και πλαισίου ομοιότητας προάγοντας έτσι τη συνεργασία τους για την όσο το δυνατόν πιο ολοκληρωμένη μοντελοποίηση και αναπαράσταση γνώσης.

8

Βιβλιογραφία

- [1] Panos Alexopoulos, Manolis Wallace, Konstantinos Kafentzis, Dimitris Askounis, A Methodology for Developing Fuzzy Ontologies, accepted for publication in Knowledge and Information Systems.
- [2] Panos Alexopoulos, Manolis Wallace, Konstantinos Kafentzis, Dimitris Askounis, Utilizing Imprecise Knowledge in Ontology-based CBR Systems by Means of Fuzzy Algebra in *International Journal of Fuzzy Systems*, Vol. 12, No. 1, March 2010
- [3] Βλαχάβας Ιωάννης, Κεφαλάς Πέτρος, Βασιλειάδης Νίκος, Κόκκορας Φώτης, Σακελλαρίου Ηλίας. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ. Γ' Έκδοση – Φεβρουάριος 2006. Εκδόσεις Β. Γκιούρδας.
- [4] Schank R.C. Dynamic Memory: A Theory of Reminding and Learning in Computers and People in *Cambridge University Press*, 1982.
- [5] Anderson, J.R., (1983). The architecture of cognition in *Harvard University Press*, Cambridge.
- [6] Aamodt A. and Plaza E., “Case-based Reasoning: Foundational Issues, Methodological Variations and Systems Approaches”, *AI-Communications*, vol. 7(1), 35-59, 1998.
- [7] Bonissone P. and Ayub S., “Representing cases and rules in plausible reasoning systems”, *Proceeding of the 1994 ARPA/RL Planning Initiative*, (Tucson, AZ), 305-316, 1994.

- [8] Schreiber G., Akkermans H., Anjewierden A., Hoog R., Shadbolt N., Velde W. and Wielinga B.J., *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, 2002.
- [9] Aha D. and Breslow L., "Refining conversational case libraries", *Case-based Reasoning Research and Development (ICCBR'97), Lecture Notes in AI, Springer*.
- [10] Gruber T.R., "A Translation Approach to Portable Ontology Specification", *Knowledge Acquisition*, vol. 5(2), 199-220, 1993.
- [11] Neches R., Fikes R., Finin T., Gruber T., Patil R., Senator T. and Swartout W.R., "Enabling technology for knowledge sharing", *AI Mag.*, vol. 12, No. 3, 36-56, 1991.
- [12] The ZK Framework from *Dr. Dobb's, The World of Software Development*, Available in: <http://drdobbs.com/architecture-and-design/205800349>.
- [13] Top Reasons in using ZK Framework, 15 January 2011, Available in: <http://www.zkoss.org/WhyZK/TopReasons.dsp>.
- [14] Zhai J., Y. Liang, J. Jiang, Y. Yu (2008), "Fuzzy ontology models based on fuzzy linguistic variable for knowledge management and information retrieval", *Intelligent Information Processing IV*, 2008, pp. 58-67.