



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Σύντηξη δεδομένων βάσει θεωρίας ενδείξεων:
Εφαρμογή στην έγκαιρη προειδοποίηση πυρκαγιών**

Άγγελος Χ. Μπιμπούδης

**Επιβλέποντες: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Χρήστος Αναγνωστόπουλος, Διδάκτωρ ΕΚΠΑ**

ΑΘΗΝΑ

ΜΑΡΤΙΟΣ 2009

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σύντηξη δεδομένων βάσει θεωρίας ενδείξεων:
Εφαρμογή στην έγκαιρη προειδοποίηση πυρκαγιών

Άγγελος Χ. Μπιμπούδης

A.M.: 1115200300092

ΕΠΙΒΛΕΠΟΝΤΕΣ

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ
Χρήστος Αναγνωστόπουλος, Διδάκτωρ ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία πραγματεύεται το θέμα σύντηξης δεδομένων πολλαπλών αισθητήρων μέσω θεωρίας ενδείξεων. Συμπερασμός, σχετικά με την πιθανότητα εμφάνισης φωτιάς σε μία γεωγραφική περιοχή, καλυμμένη από ένα δίκτυο αισθητήρων θερμοκρασίας, υγρασίας και καμερών, πετυχαίνεται μέσω μίας τέτοιας μεθόδου. Στην παρούσα εργασία η σύντηξη δεδομένων εξετάζεται μέσω της θεωρίας ενδείξεων, αξιοποιώντας δεδομένα – μετρήσεις που έχουν ήδη επεξεργαστεί.

Η προσέγγιση που ακολουθείται είναι η θεωρία που θεμελίωσαν οι Dempster και Shafer, μία θεωρία για αναπαράσταση γνώσης και σύντηξης εισερχόμενης πληροφορίας από διαφορετικούς και ανεξάρτητους ειδικούς. Αυτή η προσθήκη στην θεωρία ενδείξεων, αναλύεται από τη ρίζα της, η οποία είναι η θεωρία Bayes, φτάνοντας μέχρι το μοντέλο TBM (Transferable Belief Model) που αποτελεί επέκταση της θεωρίας των Dempster και Shafer.

Επιπροσθέτως, αριθμητικές εφαρμογές παρουσιάζονται ως απόδειξη λειτουργίας, για την καλύτερη κατανόηση και απεικόνιση των πλεονεκτημάτων αυτής της προσέγγισης. Παρουσιάζονται απλά και περισσότερο σύνθετα παραδείγματα.

Στο τελευταίο κεφάλαιο της πτυχιακής εργασίας παρατίθεται η υλοποίηση σε Java, του μηχανισμού που σχεδιάστηκε.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Τεχνικές σύντηξης δεδομένων πολλαπλών αισθητήρων για ανίχνευση πυρκαγιάς

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: διάχυτος υπολογισμός, θεωρία ενδείξεων, σύντηξη δεδομένων, ασύρματα δίκτυα αισθητήρων

ABSTRACT

The current BSc thesis deals with multisensor data fusion using evidential theory. Reasoning about the probability of fire in a geographical area covered by temperature, humidity and vision sensors is achieved through a multilevel data fusion process. In the current work, fusion is examined through the evidential reasoning perspective, assuming raw data are already accessed.

The approach being examined is the Dempster – Shafer theory of evidence, a technique for knowledge representation and fusion, of incoming information from various, and independent experts. This installment of evidence theory, is traced back at its root – Bayesian inference. We proceed with the analysis of it, which is initiated from that point, up to Transferable Belief Model, an extension of the Dempster – Shafer theory.

Furthermore, arithmetical applications are presented as proof of concept, to depict the advantages of this theory. Both novice examples and more complex ones are presented through this thesis.

In conclusion, a Java implementation of this mechanism is analyzed and demonstrated.

SUBJECT AREA: Multisensor data fusion techniques for forest fire detection

KEYWORDS: pervasive computing, evidential theory, data fusion, wireless sensor network

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω πρωτίστως, τον καθηγητή μου, κύριο Χατζηευθυμιάδη Ευστάθιο, για την εμπιστοσύνη που μου έδειξε από την πρώτη στιγμή της γνωριμίας μας. Η εργασία αυτή δεν αποτέλεσε μόνο το αποτέλεσμα μιας ανάθεσης, αλλά ήταν το προϊόν μιας ουσιαστικής συνεργασίας και πολύπλευρης ενασχόλησης στην ομάδα ερευνών Διάχυτου Υπολογισμού στο τμήμα Πληροφορικής και Τηλεπικοινωνιών, ΕΚΠΑ.

Θα ήθελα να ευχαριστήσω επίσης τους συναδέλφους μου, στην ομάδα Διάχυτου Υπολογισμού που έκαναν την έρευνα και ανάπτυξη που παρουσιάζεται στην παρακάτω εργασία, μία ξεχωριστή εμπειρία γνώσης και συναδελφικότητας. Ειδικότερα θα ήθελα να ευχαριστήσω τους: Χρήστο Αναγνωστόπουλο, Οδυσσέα Σέκκα, Μίλτο Κυριακάκο, Γιάννη Πρίγγουρη, Γιάννη Αλαγιάννη, Μάνο Αθανασούλη, Βασίλη Τσέτσο, Βασίλη Μπαούση καθώς επίσης και τον καθηγητή κ. Ευάγγελο Ζέρβα· σας ευχαριστώ για την πολύτιμη βοήθειά σας και υποστήριξη σε αυτό το ταξίδι έρευνας και υλοποίησης.

Τέλος απεριόριστη και βαθύτατη εκτίμηση στους γονείς μου, Δέσποινα και Χρήστο, που μου στάθηκαν και στέκονται στο πλευρό μου δυνατοί, σε κάθε μου πορεία, επαγγελματική, ακαδημαϊκή και προσωπική.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	10
ΕΙΣΑΓΩΓΗ.....	10
1.1 Διαχείριση κινδύνου σε περίπτωση πυρκαγιάς.....	10
1.2 Συστήματα Υποστήριξης Αποφάσεων μέσω Ασύρματου Δικτύου Αισθητήρων.....	13
ΚΕΦΑΛΑΙΟ 2	16
ΣΥΝΤΗΞΗ ΔΕΔΟΜΕΝΩΝ	16
2.1 Εισαγωγή στην Σύντηξη Δεδομένων	16
2.1.1 Χαμηλό επίπεδο σύντηξης	17
2.1.2 Μέσο επίπεδο	17
2.1.3 Υψηλό Επίπεδο	17
2.2 Αβεβαιότητα.....	19
2.3 Θεωρία ενδείξεων στην σύντηξη δεδομένων	21
2.3.1 Συναρτήσεις Πεποίθησης.....	21
2.3.2 Εννοιολογική Θεμελίωση: Από τον Bayes στους Dempster & Shafer.....	23
ΚΕΦΑΛΑΙΟ 3	36
ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ	36
3.1 Περιγραφή	36
3.2 Τύποι δεδομένων	38
3.3 Κλάσεις υλοποίησης	39
3.4 Εξομοίωση.....	44
3.4.1 Σενάριο 1	44
3.4.2 Σενάριο 2	44
3.4.3 Σενάριο 3	45
3.5 Σύνοψη.....	47

ΠΑΡΑΡΤΗΜΑ.....	49
ΚΩΔΙΚΑΣ	49
Κώδικας Υλοποίησης	49
Κλάση DSFusion	49
Κλάση HypothesisSet.....	63
Κλάση Hypothesis	70
Κλάση Hypothesis	71
Κλάση Evidence	72
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	75
ΑΝΑΦΟΡΕΣ	76

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Ροή πληροφορίας και ενεργειών.....	11
Εικόνα 2: Διαδικασίες που ενισχύονται από το πληροφοριακό σύστημα.....	13
Εικόνα 3: Ασύρματο δίκτυο αισθητήρων για παρακολούθηση πυρκαγιάς σε πραγματικό χρόνο.....	15
Εικόνα 4: Διαφορετικές αρχιτεκτονικές.....	19
Εικόνα 5: Upper Lower Probability.....	27
Εικόνα 6: Αναπαράσταση αντιστοιχίας M.....	28
Εικόνα 7: Dempster - Shafer Workflow.....	37
Εικόνα 8: Dempster Shafer Algorithm Dependencies.....	39
Εικόνα 9: BeliefInterval κλάση.....	39
Εικόνα 10: Combinations κλάση.....	40
Εικόνα 11: Combinatoric κλάση.....	40
Εικόνα 12: Dempster Shafer κλάση υλοποίησης.....	41
Εικόνα 13: Evidence κλάση.....	42
Εικόνα 14: Hypothesis κλάση.....	42
Εικόνα 15: HypothesisSet κλάση.....	43
Εικόνα 16: Στατιστική πίστης από διαδοχικούς ειδικούς.....	46

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία εκπονήθηκε στο πλαίσιο του χρηματοδοτούμενου από την ευρωπαϊκή κοινότητα ερευνητικού έργου SCIER (Sensor Computing Infrastructure for Environmental Risks). Ο στόχος του SCIER είναι να καταστήσει ασφαλέστερες, σε θέματα φυσικών κινδύνων ή καταστροφών, τις περιοχές στη ζώνη μίξης δασών – οικισμών (Wildland – Urban Interface, ή σε συντομία WUI) όπου τα δάση και οι αγροτικές καλλιέργειες έρχονται σε άμεση επαφή με σπίτια και άλλες υποδομές. Για το σκοπό αυτό, στο πλαίσιο του SCIER, σχεδιάστηκε και αναπτύχθηκε ένα ολοκληρωμένο σύστημα αισθητήρων και υπολογιστικού συστήματος, για τον έγκαιρο εντοπισμό μιας επερχόμενης καταστροφής και την άμεση αντίδραση. Η παρούσα εργασία διαπραγματεύεται το πώς σχεδιάστηκε το υποσύστημα σύντηξης δεδομένων (data fusion) για ανίχνευση του κινδύνου πυρκαγιάς με βάση τη θεωρία ενδείξεων.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία εντάσσεται, στην επιστημονική περιοχή της πληροφορικής και των τηλεπικοινωνιών που ασχολείται με τον Διάχυτο Υπολογισμό. Ο Adam Greenfield αναφέρει στο βιβλίο του [1], ότι ο Διάχυτος Υπολογισμός αποτελεί την περιοχή της πληροφορικής που θα κατακτήσει την ζωή και το περιβάλλον μας με διάφορους τρόπους. Αναφέρει στην εισαγωγή του βιβλίου του ότι με τον εν λόγω όρο, περιγράφουμε την ύπαρξη μιας δύναμης τόσο κατανεμημένης μέσα στο περιβάλλον, που αφανίζει τα υπολογιστικά συστήματα αυτά καθαυτά από το επίκεντρο του ενδιαφέροντος. Αποκαλεί, χαρακτηριστικά, την εν λόγω περιοχή, με τον όρο “Everyware”, δίνοντας έμφαση στο γεγονός ότι ο Διάχυτος Υπολογισμός δεν αφορά μία συγκεκριμένη υπολογιστική υποδομή όπως την ξέρουμε μέχρι σήμερα, αλλά αναβαθμίζει την επικοινωνία ανθρώπου μηχανής, με επικοινωνία ανθρώπου περιβάλλοντος. Η πληροφορία διαχέεται στο περιβάλλον και αποκτά αξία μέσα από τις αλληλεπιδράσεις με αυτό. Το περιβάλλον μπορεί να είναι οτιδήποτε. Ένα αυτοκίνητο, ένα σπίτι ή ένα δάσος. Το τελευταίο, είναι το περιβάλλον εκείνο που θα μας απασχολήσει στην παρούσα εργασία και ειδικότερα το πώς η πληροφορία που θα ανακτήσουμε από την κατάστασή του, μπορεί να μας βοηθήσει στο να αποφανθούμε γρήγορα, αν υπάρχει πυρκαγιά ή όχι.

Η παρούσα εργασία αφορά την εννοιολογική θεμελίωση της σύντηξης δεδομένων σε περιβάλλον διάχυτου υπολογισμού με βάση την θεωρία των ενδείξεων και ειδικότερα του μαθηματικού μοντέλου των Dempster και Shafer.

1.1 **Διαχείριση κινδύνου σε περίπτωση πυρκαγιάς**

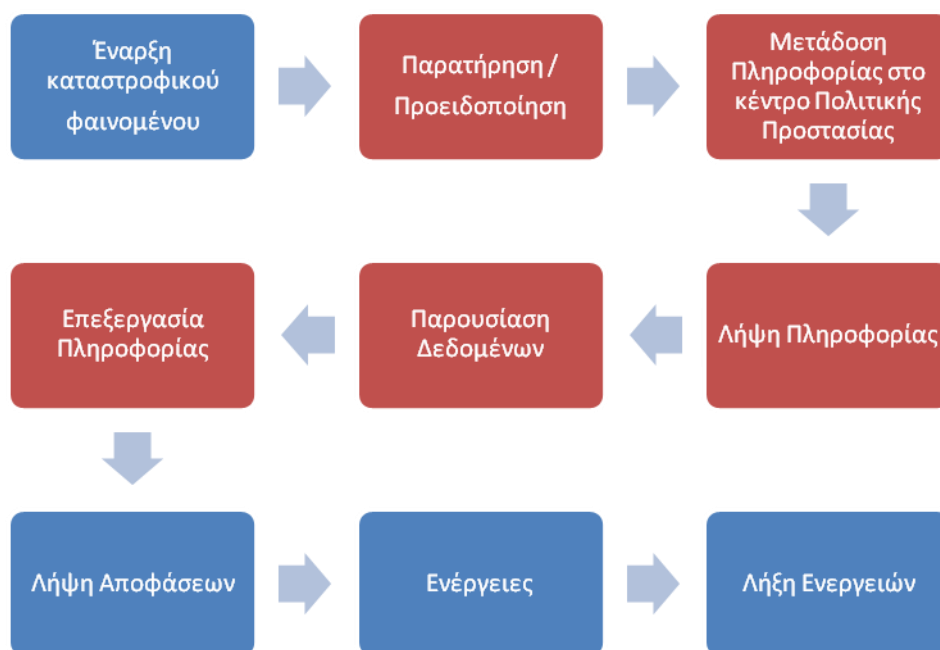
Το πλάνο της διαχείρισης μιας καταστροφής από την πολιτεία, περιλαμβάνει έναν αριθμό συγκεκριμένων βημάτων, για την καλύτερη αντιμετώπιση της κρίσης:

1. Εκτίμηση της έκτασης της καταστροφής
2. Προσδιορισμός δρόμων διαφυγής και συνεχής παρακολούθησή τους για να μην διακοπεί η ροή κίνησης από και προς τα σημεία ενδιαφέροντος, παρεμποδίζοντας την διαδικασία διαχείρισης της καταστροφής
3. Δυναμική κατανομή πόρων (Μηχανοκίνητα μέσα και ανθρώπινο δυναμικό πολιτικής προστασίας)
4. Ανεύρεση και συντονισμένη διάσωση

5. Απομάκρυνση και παροχή βοήθειας σε τραυματίες

6. Ενημέρωση πολιτών

Αναδεικνύοντας εκείνο ακριβώς το σημείο στο οποίο συμβάλει η παρούσα εργασία, αναλύω παρακάτω τα πολύ συγκεκριμένα στάδια στα οποία θα μπορούσε να συμβάλει ένα σύστημα το οποίο θα ενίσχυε την ανθρώπινη παρέμβαση σε τέτοιες περιπτώσεις (Εικόνα 1)



Εικόνα 1: Ροή πληροφορίας και ενεργειών

1. Έναρξη καταστροφικού φαινομένου

Αυτή είναι η πρώτη φάση στην πορεία πληροφορίας/διαδικασιών για την αντιμετώπιση ενός καταστροφικού φαινομένου. Συνήθως δεν υπάρχει κάποια πληροφόρηση και η πορεία μιας επιτυχημένης επιχείρησης έγκειται στο πόσο γρήγορα θα φτάσει αυτή η πληροφορία στον αρμόδιο φορέα.

2. Προειδοποίηση

Η πρώτη προειδοποίηση μπορεί να έρθει από κάποια πηγή που θα έρθει πρώτη σε επαφή με το φαινόμενο. Μια συνηθισμένη πηγή είναι κάποιος πολίτης που μπορεί να βρεθεί σε περιοχή και να ειδοποιήσει τις αρχές. Άλλο είδος μπορεί να είναι κάποιο εξειδικευμένο σύστημα ή κάποια αρμόδια αρχή (δασοφύλακας)

3. Μετάδοση πληροφορίας

Κατά την συγκεκριμένη διαδικασία, η πληροφορία για το συμβάν μεταφέρεται στο σημείο από όπου μπορεί να συντονιστεί μια επιχείρηση αντιμετώπισης του φαινομένου.

Οι τρόποι που μπορεί να επιτευχθεί αυτό είναι αρκετοί (από στόμα σε στόμα, τηλέφωνο, telex, ραδιόφωνο, συστήματα συναγερμού).

4. Λήψη πληροφορίας

Κάποιο κέντρο πληροφοριών (αστυνομικό τμήμα, πυροσβεστικός σταθμός,...) λαμβάνει την πληροφορία, την διατυπώνει καθαρά και επίσημα, καταγράφοντας μαζί και όλες τις διαθέσιμες πληροφορίες μέχρι εκείνη τη στιγμή.

5. Παρουσίαση δεδομένων

Στην συνέχεια, με στόχο την καλύτερη απεικόνιση της κατάστασης η αρμόδια αρχή παρουσιάζει τα δεδομένα σε χάρτες, πίνακες, κτλ με στόχο να έχει καλύτερη αντίληψη της κατάστασης, ξεκινώντας τον σχεδιασμό αντιμετώπισης της κρίσης.

6. Επεξεργασία πληροφορίας

Αυτή η φάση, είναι το σημείο που γίνεται επεξεργασία των δεδομένων και ένα άτομο ή με την συνδρομή ενός συμβουλίου εκτιμά την κατάσταση.

7. Λήψη αποφάσεων

Αυτή είναι η κρίσιμη φάση στην διαχείριση της κρίσης. Περιλαμβάνει την δρομολόγηση ενεργειών, κατανομή πόρων, κτλ. Η λήψη αποφάσεων γίνεται είτε από ένα άτομο είτε από ομάδα που είναι αρμόδια σε τέτοιες καταστάσεις.

8. Ενέργειες

Η φάση της λήψης αποφάσεων συνοδεύεται από τις αντίστοιχες ενέργειες οι οποίες έχουν δύο σκέλη. Είναι είτε δυναμικές (αποστολές διάσωσης, ενημέρωσης πολιτών, εκκένωση, κτλ) είτε στατικές (αναμονή για αποτελέσματα ενεργειών, κτλ)

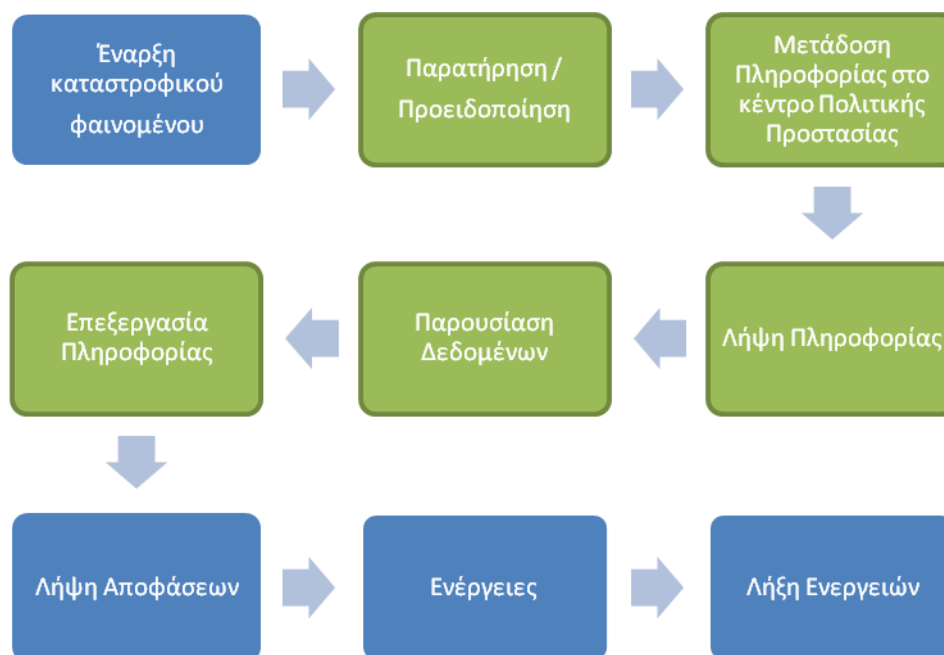
9. Λήξη ενεργειών

Η τελική φάση περιλαμβάνει εκτίμηση της κατάστασης, της ζημίας και όλων των επιχειρήσεων.

Είναι εύκολο να κατανοήσουμε, ότι αν τα πρώτα στάδια ολοκληρωθούν γρήγορα είναι σχεδόν σίγουρο ότι η καταστροφή μπορεί να αποφευχθεί [2]. Αυτό δικαιολογεί και την ανάγκη για την εφαρμογή συστημάτων, που θα υποβοηθούν τον ανθρώπινο παράγοντα μειώνοντας τον χρόνο απόκρισης από την έναρξη του φαινομένου μέχρι την ανίχνευσή του.

1.2 Συστήματα Υποστήριξης Αποφάσεων μέσω Ασύρματου Δικτύου Αισθητήρων

Ο όρος «Σύστημα Υποστήριξης Απόφασης» αναφέρεται στην προσέγγιση κατά την οποία θεωρούμε ότι η ανθρώπινη κρίση ενισχύεται με αυτοματοποιημένη επεξεργασία πληροφορίας, με την χρήση υπολογιστικών συστημάτων [3]. Έτσι αυξάνεται η ακρίβεια, βελτιώνεται η ποιότητα και εκδίδεται γρηγορότερα μία απόφαση η οποία απαντά σε μία υπόθεση που έχει οριστεί εκ των προτέρων. Σε αυτό το κεφάλαιο παρουσιάζονται επιγραμματικά τα βασικά μέρη ενός συστήματος που έρχεται να υποστηρίξει την συνήθη διαδικασία διαχείρισης καταστάσεων κρίσης ενισχύοντας λειτουργίες που περιγράφονται στην προαναφερόμενη ροή. Στην Εικόνα 2 με πράσινο χρώμα, βλέπουμε τις διαδικασίες που ενισχύονται με βάση το προτεινόμενο σύστημα. Η αντιστοιχία με τα εν λόγω μέρη παρουσιάζεται στον παρακάτω πίνακα (Πίνακας 1).



Εικόνα 2: Διαδικασίες που ενισχύονται από το πληροφοριακό σύστημα

Πίνακας 1: Αντιστοιχίες διαδικασιών

Συνήθης διαδικασία	Μέρη συστήματος που την ενισχύουν
Έναρξη καταστροφικού φαινομένου	Ανίχνευση φαινομένου από Δίκτυο Αισθητήρων ¹

¹ Wireless Sensor Network

Παρατήρηση / Προειδοποίηση	Αλγόριθμος ανίχνευσης
Μετάδοση Πληροφορίας	Κεντρικό Υπολογιστικό Σύστημα
Λήψη Πληροφορίας	
Παρουσίαση Δεδομένων	Παρουσίαση ηλεκτρονικών χαρτών με πληροφορίες σε πραγματικό χρόνο
Επεξεργασία Πληροφορίας	Σύντηξη Δεδομένων / Πρόβλεψη
Λήψη Αποφάσεων	Logistics ² , Κατανομή Πόρων

Για την ανίχνευση των προαναφερόμενων φαινομένων υποθέτουμε την αξιοποίηση ασύρματου δικτύου αισθητήρων. Τα δίκτυα αισθητήρων αποτελούν εξέλιξη των απλών αισθητήρων και λειτουργούν σύμφωνα με τους δύο παρακάτω τρόπους [4].

- Τοποθετούνται αισθητήρες που μπορεί να βρίσκονται είτε πολύ κοντά είτε πολύ μακριά από την εξέλιξη του φαινομένου, πράγμα που σημαίνει ότι αναπτύσσονται ειδικές θεωρίες γύρω από το θέμα της «αίσθησης» σε συνάρτηση με τον περιβάλλοντα θόρυβο
- Η εν λόγω τοποθέτηση εξυπηρετεί έναν και μόνο σκοπό. Την αποστολή σειρών από τιμές φυσικής σημασίας σε κάποιους κεντρικούς κόμβους ώστε να συντηχτεί η περιεχόμενη πληροφορία και να μεταφραστεί σε απόφαση.

Παραδείγματα τέτοιων αισθητήρων για ανίχνευση φωτιάς είναι

- 1 Θερμοκρασίας
- 2 Καπνού
- 3 Οπτικοί αισθητήρες
- 4 Υγρασίας

Όλα αυτά τα είδη αισθητήρων μας καθιστούν διαθέσιμη πληροφορία με ιδιαίτερη φυσική σημασία όχι μόνο για την ανίχνευση του φαινομένου αλλά και για την παρατήρηση της εξέλιξής του στον χρόνο. Στην Εικόνα 3 βλέπουμε ένα τέτοιο δίκτυο αναπτυγμένο σε δασική έκταση. Οι κόμβοι μεταφέρουν πληροφορία θερμοκρασίας σε έναν συγκεκριμένο κόμβο, όπου γίνεται και η αποθήκευση και επεξεργασία της πληροφορίας (sink node) με σκοπό την σύντηξη των δεδομένων και την εξαγωγή συμπεράσματος από αυτά, μία διαδικασία που θα εξετάσουμε αναλυτικά εν συνεχεία.

² Υποσύστημα σχεδιασμού διαδρομών



Εικόνα 3: Ασύρματο δίκτυο αισθητήρων για παρακολούθηση πυρκαγιάς σε πραγματικό χρόνο

ΚΕΦΑΛΑΙΟ 2

ΣΥΝΤΗΞΗ ΔΕΔΟΜΕΝΩΝ

2.1 *Εισαγωγή στην Σύντηξη Δεδομένων*

Από την στιγμή που τα δεδομένα λαμβάνονται, η επόμενη ενέργεια είναι να γίνει σύντηξη και επεξεργασία τους και να βγει κάποιο νόημα από αυτά. Βασισμένοι στην ευρεία έννοια των αισθητήρων, δεδομένα μπορεί να έρθουν σε οποιαδήποτε μορφή, σε συνεχή ροή ή ασυνεχή. Η περιοχή που εξετάζει τον συνδυασμό δεδομένων και πληροφορίας από αισθητήρες και γενικά από οποιαδήποτε πηγή πληροφορίας λέγεται σύντηξη δεδομένων (data fusion) και έχει εξελιχθεί κυρίως μέσα από στρατιωτικές εφαρμογές ξεκινώντας από το τμήμα ερευνών του Υπουργείου Άμυνας των Ηνωμένων Πολιτειών Αμερικής. Οι αρχικές εφαρμογές αυτής της νέας επιστημονικής περιοχής για την εποχή ήταν παρακολούθηση εναέριων και θαλάσσιων στόχων με σκοπό την ανίχνευση, τον εντοπισμό, αναγνώριση πολεμικών οχημάτων. Οι πηγές που χρησιμοποιούνταν, ήταν δεδομένα από ραντάρ, από ελεγκτές κυκλοφορίας, ακουστικά σήματα και ακτινοβολίες (εφαρμόζοντας μαθηματικά μοντέλα όπως Expectation maximization [5], για την ανεύρεση των άγνωστων παραμέτρων). Στην συνέχεια ο κλάδος επεκτάθηκε και εξελίχθηκε μέσω των πανεπιστημίων (ρομποτική, ιατρικές διαγνώσεις, παρακολούθηση περιβάλλοντος) και των βιομηχανιών αναγκών (παρακολούθηση συστημάτων κυρίως). Η σύντηξη δεδομένων λαμβάνει χώρα ενώνοντας παρατηρήσεις από έναν ή πολλούς αισθητήρες και εφαρμόζοντας θεωρίες πάνω στα δεδομένα των ενδείξεων. Οι τιμές αυτών των μαθηματικών εφαρμογών είναι εκείνες που θα εξεταστούν, θα συγκριθούν με γνωστά όρια και θα εξεταστούν με τεχνικές τεχνητής νοημοσύνης για να εξαχθούν χρήσιμα και αξιόπιστα συμπεράσματα.

Μερικές θεωρίες που υιοθετούνται για την περιοχή της σύντηξης δεδομένων (χωρίς να τις αναλύσουμε περαιτέρω) είναι οι εξής:

- Μαθηματικά μοντέλα όπως Kalman filters
- Τεχνικές συμπερασμού όπως αλγόριθμοι συσταδοποίησης, αναγνώρισης προτύπων, νευρωνικά δίκτυα, Dempster – Shafer, θεωρία ενδείξεων
- Διάχυτος υπολογισμός, λογικός συμπερασμός

Στην διαδικασία σύντηξης δεδομένων υπάρχουν τρεις διαφορετικές τεχνικές που υιοθετούνται [6]:

2.1.1 Χαμηλό επίπεδο σύντηξης

Ονομάζεται επίσης και σύντηξη δεδομένων, συνδυάζει διάφορες πηγές ανεπεξέργαστων δεδομένων για την παραγωγή νέων δεδομένων που αναμένεται να είναι περισσότερο ενημερωτικά και συνθετικά από εκείνα που λάβαμε στην αρχή ως είσοδο. Συνήθως, στην επεξεργασία εικόνας, εικόνες παρουσιάζουν αρκετές ζώνες φάσματος της ίδιας σκηνής. Ένας αλγόριθμος επεξεργασίας εικόνας για παράδειγμα, θα μπορούσε στη συνέχεια να χρησιμοποιήσει τις προ-επεξεργασμένες αυτές εικόνες, αντί της αρχικής εικόνας. Αυτό είναι ιδιαίτερα σημαντικό όταν ο αριθμός των διαθέσιμων φασματικών ζωνών γίνεται τόσο μεγάλος, ώστε είναι αδύνατο να δούμε τις εικόνες χωριστά. Αυτό το είδος της σύντηξης προϋποθέτει ακριβή καταγραφή των διαθέσιμων εικόνων σε πολύ αναλυτικό επίπεδο (pixel-επίπεδο). Αυτή η εγγραφή είναι εγγενείς όταν τα διάφορα κομμάτια προέρχονται από τον ίδιο αισθητήρα αλλά είναι πολύ πιο περίπλοκη, όταν προέρχονται από διαφορετικές πηγές (υπέρυθρες, κάμερα, κτλ). Αυτό το είδος σύντηξης στοχεύει στον συνδυασμό πληροφορίες από διαφορετικές εικόνες. Το πρόβλημα σε τέτοιου είδους προβλήματα είναι να προσδιορίσουμε πόσο σχετική και ακριβής είναι αυτή η πληροφορία

2.1.2 Μέσο επίπεδο

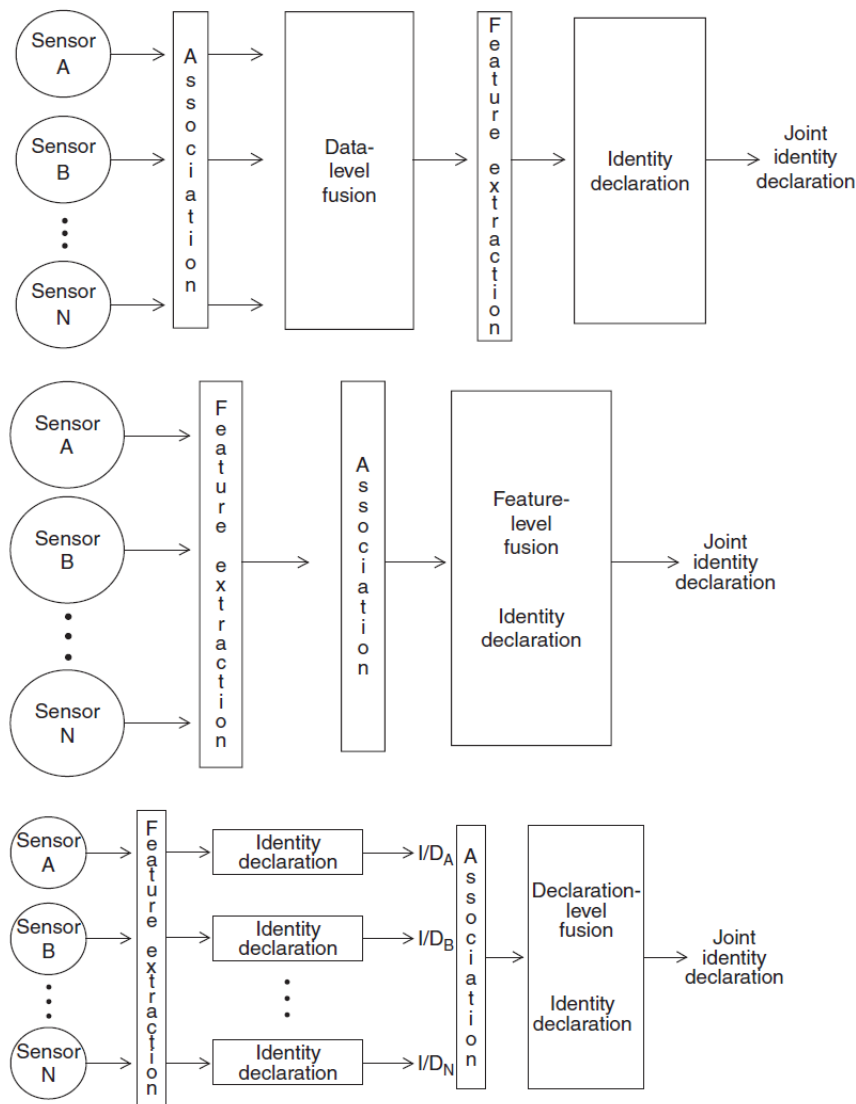
Το μέσο επίπεδο σύντηξης, που ονομάζονται επίσης επίπεδο σύντηξης χαρακτηριστικών, συνδυάζει διάφορα χαρακτηριστικά. Αυτά τα χαρακτηριστικά μπορεί να προέρχονται από διάφορες πηγές ακατέργαστων δεδομένων (αισθητήρες, διαφορετικές στιγμές, κ.λπ.) ή από τα ίδια τα ανεπεξέργαστα δεδομένα. Στην τελευταία περίπτωση, ο στόχος είναι η εξεύρεση χαρακτηριστικών που σχετίζονται μεταξύ τους. Η πηγή αυτών των χαρακτηριστικών είναι συνήθως διαφορετικοί αλγόριθμοι.

2.1.3 Υψηλό Επίπεδο

Το υψηλό επίπεδο, που καλείται επίσης σύντηξη αποφάσεων συνδυάζει αποφάσεις που προέρχονται από πολλούς πράκτορες. Κατ'επέκταση, κάποιος μπορεί να μιλήσει

για σύντηξη αποφάσεων ακόμα και αν οι πράκτορες επιστρέφουν επίπεδα εμπιστοσύνης (το πόσο πιστεύει τα δεδομένα που λαμβάνει από τον αισθητήρα – belief) και όχι κάποια απόφαση. Για να γίνει διαχωρισμός ανάμεσα στις δύο περιπτώσεις μιλάμε για δυνατή και απλή σύντηξη δεδομένων. Μέθοδοι για σύντηξη αποφάσεων περιλαμβάνουν voting αλγορίθμους, στατιστικές μεθόδους και μεθόδους που βασίζονται σε θεωρία ασαφών συνόλων. Στην παρούσα εργασία θα εξετάσουμε την σύντηξη δεδομένων βάση θεωρίας ενδείξεων κατά την οποία συνδυάζεται πληροφορία, αφότου έχει γίνει η προεπεξεργασία από τους αρμόδιους αισθητήρες.

Στην Εικόνα 4 βλέπουμε ακριβώς τρεις μεθόδους για σύντηξη δεδομένων. Στην πρώτη βλέπουμε απευθείας σύντηξη δεδομένων, χωρίς κάποια προεπεξεργασία. Στην δεύτερη περίπτωση τα δεδομένα των αισθητήρων αναπαριστώνται με διανύσματα χαρακτηριστικών (feature vectors) και στην τρίτη γίνεται πλήρης επεξεργασία αισθητηριακών δεδομένων και συντήκονται κατόπιν.



Εικόνα 4: Διαφορετικές αρχιτεκτονικές

Στην παρούσα εργασία εξετάζουμε την εν λόγω διαδικασία στο υψηλό επίπεδο.

2.2 Αβεβαιότητα

Η μεθοδολογία που ακολουθούμε για να αποφασίσουμε αν θα πραγματοποιήσουμε μία ενέργεια ή άλλη, με δεδομένο ένα συγκεκριμένο ποσό άγνοιας (ignorance), δεν εμπίπτει στην κλασσική θεωρία πιθανοτήτων, αφού δεν μπορούμε να απεικονίσουμε σωστά κάποια μεγέθη, όπως την πλήρη άγνοια.

Η κλασσική θεωρία πιθανοτήτων λειτουργεί εν γένει ως εξής: Όταν ένα νέο αποτέλεσμα είναι διαθέσιμο (αληθές 100%), το συνδυάζουμε με τους γνωστούς κανόνες (Bayes) και εξαγάγουμε πιθανότητα συμβάντος. Όταν υπάρχουν δεδομένα με πλήρη βεβαιότητα (το κέρμα στο 50% των προσπαθειών έδειξε το δεδομένο «κεφαλή») τότε η κλασσική θεωρία των πιθανοτήτων λειτουργεί κανονικά. Πριν όμως προχωρήσουμε, είναι καλό να

σχολιάσουμε τον ορισμό της άγνοιας που θα μας βοηθήσει να καταλάβουμε τι μας ωθεί στο να χρησιμοποιήσουμε την μέθοδο D-S και πώς οδηγηθήκαμε σε αυτή θεωρητικά.

Καταρχάς με την λέξη «άγνοια» περιγράφουμε την κατάσταση κατά την οποία μας λείπει πληροφορία πριν αποφανθούμε για μία έκφραση. Παρόλα αυτά δεν είναι όλα τα είδη άγνοιας το ίδιο και αυτά ακριβώς τα είδη, είναι εκείνα που μας έκαναν να μην μοντελοποιούμε όλα τα σχετικά προβλήματα με την θεωρία κλασσικής πιθανότητας [7], [8]. Η αβεβαιότητα χωρίζεται σε τρεις μεγάλες κατηγορίες:

- ανακρίβεια (imprecision): αναφέρεται σε περιπτώσεις όπου η τιμή μίας μεταβλητής δίνεται, αλλά όχι με την απαραίτητη ακρίβεια που απαιτεί το πεδίο του προβλήματος (π.χ., κάνει ζέστη).
- ανεπάρκεια (incompleteness): αναφέρεται σε περιπτώσεις όπου οι τιμές κάποιων μεταβλητών λείπουν (π.χ., βάση δεδομένων που δεν έχει όλα τα δεδομένα στα πεδία των πλειάδων της).
- αβεβαιότητα (uncertainty): αναφέρεται σε περιπτώσεις όπου ένας πράκτορας (agent) ³ μπορεί να δομήσει μία γνώμη για ένα γεγονός, η οποία όμως δεν βασίζεται σε επαρκή αποδεικτικά στοιχεία και μπορεί να είναι σωστή ή λάθος (π.χ., ο αισθητήρας αναφέρει φωτιά).

Συλλογιζόμενοι την ανωτέρω ταξινόμηση, μπορούμε να έχουμε οποιονδήποτε συνδυασμό τους, όπου κάθε ένας μπορεί να αντιμετωπιστεί με διαφορετικό τρόπο θεωρητικά. Για παράδειγμα η έλλειψη πληροφορίας μπορεί να αντιμετωπιστεί με *μη μονοτονική λογική* (non-monotonic logic), όταν έχουμε ανακρίβεια στα δεδομένα μπορούμε να χρησιμοποιήσουμε την θεωρία των *ασαφών συνόλων* (fuzzy sets) και αν έχουμε αβεβαιότητα μπορεί να χρησιμοποιήσουμε την θεωρία των πιθανοτήτων, ενώ αν έχουμε συνδυασμό ανακρίβειας και αβεβαιότητας τότε θα πρέπει να χρησιμοποιήσουμε άλλα μοντέλα όπως *θεωρία δυνατοτήτων* (possibility theory) ή *θεωρία ενδείξεων* (evidence theory). Πριν συνεχίσουμε στην εισαγωγή της θεωρίας ενδείξεων θα αναφερθούμε λίγο πιο διεξοδικά στους διαφορετικούς τρόπους αντιμετώπισης που προαναφέραμε.

³ Μία αυτόνομη οντότητα που παρατηρεί και δρα στα πλαίσια ενός περιβάλλοντος

2.3 Θεωρία ενδείξεων στην σύντηξη δεδομένων

2.3.1 Συναρτήσεις Πεποίθησης

Δύο μοντέλα έχουν εισαχθεί για την ποσοτικοποίηση του βαθμού πεποίθησης κάποιου. Το ένα βασίζεται στην κλασσική συνάρτηση πιθανότητας, που χρησιμοποιείται στην θεωρία Bayes και το άλλο στις συναρτήσεις πεποίθησης που εισήχθησαν στην θεωρία των Dempster-Shafer [9]. Η θεωρία των *συναρτήσεων πεποίθησης* (belief functions) παρέχει την εναλλακτική προσέγγιση από την θεωρία του Bayes, αξιοποιώντας την θεωρία των πιθανοτήτων, για την ποσοτικοποίηση υποκειμενικών κρίσεων. Ο Bayes χρησιμοποιεί απ' ευθείας τις πιθανότητες για να αποφανθεί για μία πρόταση, ενώ τα μοντέλα που χρησιμοποιούν συναρτήσεις πεποίθησης χρησιμοποιούν πιθανότητες για να αποφανθούν για σχετικές προτάσεις των οποίων ο συνδυασμός θα δώσει την τελική απάντηση στην αρχική μας πρόταση. Οι πιθανότητες που κατανέμονται στις επιμέρους προτάσεις ονομάζονται βαθμοί πεποίθησης (degrees of belief) και είναι ένας αριθμός μεταξύ 0 και 1. Έστω για παράδειγμα ότι θέλουμε να χειριστούμε αβέβαιη γνώση σε ένα πεδίο προβλήματος ιατρικής χρησιμοποιώντας λογική πρώτης τάξης [10]. Έστω ότι έχουμε τον παρακάτω κανόνα:

$$\forall p \text{ Σύμπτωμα } (p, \text{Πονόδοντος}) \Rightarrow \text{Ασθένεια } (p, \text{Κοιλότητα})$$

Ο κανόνας είναι λανθασμένος γιατί δεν είναι απαραίτητο ότι όποιος έχει πονόδοντο θα έχει και κοιλότητα στο δόντι. Μία πρόχειρη λύση στο πρόβλημα αυτό θα ήταν να προσθέσουμε μία άπειρη αλληλουχία από πιθανές ασθένειες προσθέτοντας στην ουσία μία μεγάλη λίστα αιτίων.

$$\begin{aligned} \forall p \text{ Σύμπτωμα } (p, \text{Πονόδοντος}) \\ \Rightarrow \text{Ασθένεια } (p, \text{Κοιλότητα}) \vee \text{Ασθένεια } (p, \text{Ουλίτιδα}) \\ \vee \text{Ασθένεια } (p, \text{Απόστημα}) \vee \dots \end{aligned}$$

Αυτή η λύση δεν είναι πρακτική και μία καλή παραλλαγή είναι να αντιστρέψουμε τον κανόνα μετατρέποντάς τον σε αιτιολογικό, έτσι έχουμε:

$$\forall p \text{ Ασθένεια } (p, \text{Κοιλότητα}) \Rightarrow \text{Σύμπτωμα } (p, \text{Πονόδοντος})$$

Ούτε σε αυτή την περίπτωση έχουμε σωστή πρόταση από την στιγμή που δεν προκαλούν όλες οι κοιλότητες πονόδοντο! Βλέπουμε ότι η παραπάνω διαδικασία αποτυγχάνει για τρεις κυρίως λόγους:

1. Για να έχουμε μία ασφαλή και σωστή πρόταση θα πρέπει να εξαντλήσουμε όλα τα διαθέσιμα συμπτώματα
2. Δεν έχουμε όλη την απαραίτητη γνώση εκ των προτέρων, η ιατρική επιστήμη δεν έχει την πλήρη θεωρία (μερική άγνοια)
3. Ακόμα και αν τα 1,2 ισχύουν πάντα για έναν ασθενή μπορεί να μην μπορούμε να εκτελέσουμε όλους τους απαραίτητους ελέγχους.

Έστω λοιπόν ότι ο πράκτορας μπορεί να παρέχει έναν βαθμό πεποίθησης για τις σχετικές προτάσεις που όπως είπαμε μοντελοποιούνται στο πεδίο των πιθανοτήτων. Ο πράκτορας τώρα έχει ένα καλό εργαλείο για να συναθροίζει την αβεβαιότητα που προκύπτει από τα παραπάνω. Στην περίπτωση μας, δηλαδή, έστω ότι πιστεύουμε κατά 80% ότι ο ασθενής έχει κοιλότητα, αφού μας επισκέφτηκε με πονόδοντο. Το 80% πίστης προκύπτει είτε στατιστικά είτε από διάφορους κανόνες είτε από σύντηξη πεπιοθήσεων για το συγκεκριμένο γεγονός! Το υπόλοιπο 20% χρησιμοποιείται για την αναπαράσταση της συνάθροισης όλων των άλλων περιπτώσεων. Εδώ αξίζει να τονίσουμε την λεπτή διαφορά του βαθμού πεποίθησης με τον βαθμό αλήθειας. Αν έχουμε 0 πιθανότητα σε μία δεδομένη πρόταση τότε η πρόταση είναι ψευδής, ενώ αν έχουμε 1 τότε η πρόταση θεωρούμε ότι είναι αληθής. Τιμές βαθμού πεποίθησης ανάμεσα στο 0 και στο 1 σημαίνει ότι πιστεύουμε με αντίστοιχη πιθανότητα ότι το γεγονός είναι αληθές. Για παράδειγμα μία πιθανότητα 0.8 δεν σημαίνει ότι η πρόταση είναι 80% αληθής αλλά έχουμε 80% βαθμό πεποίθησης. Το 0.8 αναφέρεται στην πεποίθηση του πράκτορα και όχι άμεσα στον πραγματικό κόσμο και επηρεάζεται μόνο από το τι έχει αισθανθεί μέχρι εκείνη της στιγμή. Οι αισθήσεις που έχει λάβει ονομάζονται *ενδείξεις* (evidence) και είναι εκείνες οι πιθανολογικές δηλώσεις που συνδυάζονται (με κάποιο μαθηματικό φορμαλισμό, θα τους εξετάσουμε στην συνέχεια).

Μία άλλη διαφορά είναι ότι οι βαθμοί πεποίθησης μπορεί να μην αθροίζουν στο 100% όπως συμβαίνει με τις κλασσικές πιθανότητες. Έστω για παράδειγμα ότι έχουμε έναν μάρτυρα για το γεγονός πτώσης ενός κλαδιού στο αμάξι μας. Το όνομα του μάρτυρα είναι Χρήστος. Ο Χρήστος μας πληροφορεί για το γεγονός. Εμείς λόγω της πείρας μας και της γνώμης που έχουμε σχηματίσει για τον Χρήστο είναι ότι τον θεωρούμε αξιόπιστο κατά 90% και ότι τον θεωρούμε αναξιόπιστο κατά 10% (άθροισμα μονάδας ως κλασσικές πιθανότητες). Σε περίπτωση που είναι αξιόπιστος η δήλωση του Χρήστου

είναι αληθής μάλλον, ενώ αν είναι αναξιόπιστος τότε δεν μας είπε απαραίτητα ψέμα. Αναλογιζόμενοι λοιπόν, την απάντηση που μας έδωσε, έχουμε την πεποίθηση ότι κατά 90% ισχύει αυτό που είπε (ότι έπεσε κλαδί στο αυτοκίνητο), αλλά 0% ότι δεν έπεσε το κλαδί πάνω στο αυτοκίνητό μας (δεν έχουμε κάποια μαρτυρία για την αντίθετη πρόταση), πράγμα που σημαίνει ότι δεν έχουμε κανένα λόγο να πιστέψουμε ότι δεν έπεσε κάποιο κλαδί στο αυτοκίνητό μας. Το 90% και το 0% αποτελούν την συνάρτηση πεποίθησης.

2.3.2 Εννοιολογική Θεμελίωση: Από τον Bayes στους Dempster & Shafer

Όπως είδαμε οι βαθμοί πεποίθησης απορρέουν από τον βαθμό αβεβαιότητας για ένα γεγονός. Το πώς αντιμετωπίζεται το πρόβλημα αυτό, εξαρτάται από το πεδίο του προβλήματος (όπως αναδείξαμε στο 2.3.1). Στα επόμενα θα υιοθετήσουμε την εξής σειρά παρουσίασης: 1) παρουσιάζουμε το κλασσικό μοντέλο πιθανότητας 2) στην συνέχεια το μοντέλο Ανώτερων – Κατώτερων Πιθανοτήτων (Upper Lower Probabilities ή αλλιώς ULP), 3) το μοντέλο του Dempster 4) οι παραλλαγές του Shafer σύμφωνα με την μαθηματική θεμελίωση του Dempster και 5) το Μοντέλο Μεταφερόμενων Πεποιθήσεων (Transferable Belief Model ή αλλιώς TBM). Η πορεία που ακολουθούμε συναντάται στην εργασία του Philippe Smets [11] και διαγράφεται τοιουτοτρόπως για καλύτερη κατανόηση των θεωριών, τι συμπληρώνει η κάθε μία, τι προβλήματα λύνει και ποια ερωτηματικά γέννησε.

Κάθε μοντέλο που χρησιμοποιεί την έννοια της πεποίθησης έχει δύο στοιχεία: ένα στατικό και ένα δυναμικό. Στο στατικό περιγράφεται η κατάσταση της πεποίθησης και στο δυναμικό περιλαμβάνονται οι κανόνες που ανανεώνουν την κατάσταση στην περίπτωση εμφάνισης νέων ενδείξεων. Μία τέτοια, ολοκληρωτική προσέγγιση είναι αναγκαία γιατί οι διαφορές μπορούν να εντοπιστούν μόνο με την προσεκτική μελέτη του δυναμικού μέρους.

Έστω ότι έχουμε έναν αισθητήρα που ανιχνεύει θερμοκρασία και ανάβει μία κόκκινη ένδειξη όταν καταλαβαίνει ότι το περιβάλλον είναι ζεστό ενώ ανάβει μία μπλε ένδειξη σε αντίθετη περίπτωση. Ο μόνος περιορισμός που τίθεται τεχνικά είναι ότι λόγω κατασκευής δεν μπορούμε να τον εμπιστευτούμε πλήρως, πράγμα που μας γνωστοποιείται από τον κατασκευαστή ο οποίος μας αναφέρει ότι με πιθανότητα 20% ο αισθητήρας μπορεί να μην αναφέρει σωστά αποτελέσματα. Αυτό συμβαίνει όταν ο αισθητήρας είναι χαλασμένος και σε αυτή τη περίπτωση η ένδειξη που αναφέρεται

μπορεί να μην είναι αληθής σχετικά με τις πραγματικές συνθήκες. Ποιος είναι λοιπόν ο βαθμός πεποίθησής μου για το αν έχουμε ζέστη ή κρύο αν η ένδειξη είναι μπλε;

Ένα εργαλείο που θα μας βοηθήσει στο να διατηρούμε ενοποιημένη την πληροφορία είναι το λεγόμενο πλαίσιο παρατήρησης (Frame of Discernment) Ω , το οποίο καλείται αλλιώς και σύνολο καταστάσεων. Με απλή Άλγεβρα Bool και θεωρία συνόλων το σύνολο αυτό ισούται με όλους τους δυνατούς συνδυασμούς ένδειξης-πραγματικής θερμοκρασίας-κατάστασης αισθητήρα. Έτσι το Ω είναι το καρτεσιανό γινόμενο των:

- Ένδειξη (S): {Μπλε, Κόκκινο}
- Θερμοκρασία (T): {Ζέστη, Κρύο}
- Κατάσταση (Θ): {Λειτουργικό, Χαλασμένο}

Όπως καταλαβαίνουμε το πλαίσιο παρατήρησης ισούται με:

$$\Omega = \text{Ένδειξη} \times \text{Θερμοκρασία} \times \text{Κατάσταση}$$

και ο πίνακας ο οποίος θα χρησιμοποιήσουμε για τους υπολογισμούς του παραδείγματος θα είναι ο εξής:

Πίνακας 2: Τα στοιχεία του συνόλου Ω

	Μπλε		Κόκκινο	
	Ζεστό	Κρύο	Ζεστό	Κρύο
Λειτουργικό	a	b	c	d
Χαλασμένο	e	f	g	h

2.3.2.1 Μοντέλο Πιθανοτήτων

Στο κλασσικό μοντέλο πιθανοτήτων το στατικό μέρος αποτελείται από την αποτίμηση της συνάρτησης πυκνότητας πιθανότητας p πάνω στα στοιχεία του Ω έτσι ώστε $p: \Omega \rightarrow [0,1]$ με $\sum_{\omega \in \Omega} p(\omega) = 1$. Οι βαθμοί πεποίθησης στα υποσύνολα του Ω αποτιμώνται σύμφωνα με την συνάρτηση κατανομής $P: 2^\Omega \rightarrow [0,1]$ έτσι ώστε για κάθε $\omega \in \Omega, P(\{\omega\}) = p(\omega)$ και για κάθε $A, B \subseteq \Omega$ με $A \cap B = \emptyset, P(A \cup B) = P(A) + P(B)$ και $P(A) = \sum_{\omega \in A} p(\omega)$. Ο μόνος δυναμικός κανόνας που υπάρχει εδώ είναι πώς όταν έχουμε μία καινούρια ένδειξη για το γεγονός B που ξέρουμε ότι είναι αληθές, τότε το

εισάγουμε ως νέα γνώση στο σύστημα μας με την μορφή της υπό συνθήκη πιθανότητας

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Σύμφωνα με την κλασσική πιθανότητα ισχύει ότι $P(\text{Χαλασμένο}) = 0.2$ και $P(\text{Λειτουργικό}) = 0.8$, δηλαδή

$$p(a) + p(b) + p(c) + p(d) = 0.8$$

$$p(e) + p(f) + p(g) + p(h) = 0.2$$

Επίσης αν ο αισθητήρας λειτουργεί σωστά, τότε θεωρώ ότι το αποτέλεσμα είναι 100% αληθές οπότε αν η ένδειξη είναι κόκκινη τότε η θερμοκρασία είναι υψηλή, ενώ αν είναι μπλε τότε είναι χαμηλή. Με αυτό τον συλλογισμό εξάγουμε το συμπέρασμα ότι

$$p(a) = p(d) = 0$$

Αν ο αισθητήρας είναι χαλασμένος τότε θεωρώ ότι η ένδειξη, δεν σχετίζεται με την πραγματική θερμοκρασία, πράγμα που μεταφράζεται στην θεωρία των πιθανοτήτων ως εξής:

$$P(\text{Μπλε}|\text{Χαλασμένο}, \text{Ζεστό}) = P(\text{Μπλε}|\text{Χαλασμένο}, \text{Κρύο})$$

Η πιθανότητα ο αισθητήρας να είναι μπλε όταν το θερμόμετρο είναι χαλασμένο είναι η

$$P(\text{Μπλε}|\text{Χαλασμένο}) = \frac{P(\text{Μπλε} \cap \text{Χαλασμένο})}{P(\text{Χαλασμένο})} = \frac{p(e) + p(f)}{p(e) + p(f) + p(g) + p(h)} = x$$

Επίσης η πιθανότητα του να είναι το θερμόμετρο χαλασμένο και η θερμοκρασία να είναι κρύο είναι $P(\text{Χαλασμένο}) = p(b) + p(f) + p(h) = \pi$. Αυτή είναι μία πιθανότητα που δεν μπορεί να αποκτηθεί ρεαλιστικά γιατί υποδεικνύει γνώση πριν την μέτρηση. Τέλος η πιθανότητα της κατάστασης του αισθητήρα είναι άσχετη με την κατάσταση που επικρατεί στο περιβάλλον (π.χ., $P(\text{Λειτουργικό}|\text{Ζεστό}) = P(\text{Λειτουργικό}) = 0.8$).

Ο Πίνακας 2 συμπληρώνεται ως εξής:

Πίνακας 3: Κατανομή πιθανότητας στο Ω

	Μπλε		Κόκκινο	
	Ζεστό	Κρύο	Ζεστό	Κρύο

Λειτουργικό	0	0.8π	0.8(1-π)	0
Χαλασμένο	0.2(1-π)x	0.2πx	0.2(1-π)(1-x)	0.2π(1-x)

Ο στόχος μας είναι να μπορέσουμε να βρούμε την πιθανότητα $P(Κρύο|Μπλε)$ και χρησιμοποιώντας τα παραπάνω αποτελέσματα και τον επεκταμένο αλγόριθμο του Bayes έχουμε τα εξής:

$$\begin{aligned}
 &P(Κρύο|Μπλε) \\
 &= P(Κρύο|Μπλε, Λειτουργικό)P(Λειτουργικό|Μπλε) \\
 &+ P(Κρύο|Μπλε, Χαλασμένο)P(Χαλασμένο|Μπλε) \\
 &= \frac{P(Μπλε|Λειτουργικό)P(Λειτουργικό)}{P(Μπλε)} + P(Κρύο) \frac{P(Μπλε|Χαλασμένο)P(Χαλασμένο)}{P(Μπλε)} \\
 &= \frac{0.8P(Κρύο) + 0.2P(Κρύο)P(Μπλε|Χαλασμένο)}{P(Μπλε)} \\
 &= \frac{P(Κρύο)(0.8 + 0.2P(Μπλε|Χαλασμένο))}{P(Μπλε|Λειτουργικό)P(Λειτουργικό) + P(Μπλε|Χαλασμένο)P(Χαλασμένο)} \\
 &= \frac{P(Κρύο)(0.8 + 0.2P(Μπλε|Χαλασμένο))}{P(Κρύο)0.8 + P(Μπλε|Χαλασμένο)0.2}
 \end{aligned}$$

Στον παραπάνω υπολογισμό έχουμε τα εξής προβλήματα. Δεν μπορούμε να ξέρουμε a priori την πιθανότητα $P(Κρύο)$ αλλά ούτε και την $P(Μπλε|Χαλασμένο)$, δηλαδή την δεσμευμένη πιθανότητα του να δείχνει μπλε ενώ είναι χαλασμένος ο αισθητήρας. Ο ενδιαφερόμενος σε αυτή την περίπτωση θα πρέπει να καταφύγει σε τεχνάσματα για να καλύψει το κενό σε αυτές τις τιμές όπως το να δώσει τυχαίες αποτιμήσεις.

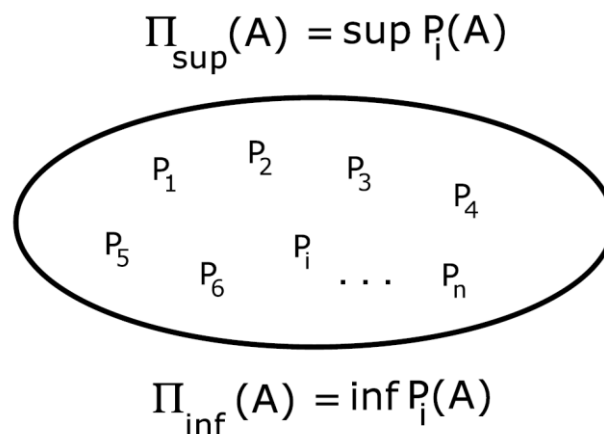
2.3.2.2 Μοντέλο Ανώτερων – Κατώτερων Πιθανοτήτων

Η κύρια ιδέα πίσω από το μοντέλο ανώτερων-κατώτερων πιθανοτήτων (ULP στο εξής), είναι να μην εργαζόμαστε κυρίως με μία κατανομή πιθανότητας αλλά με ένα σύνολο από πιθανές κατανομές που προκύπτουν από διαφορετικές τιμές άγνωστων πιθανοτήτων όπως στην περίπτωση του 2.3.2.1. Αυτό το σύνολο μπορεί να αναπαρασταθεί μαζί με το κατώτερο και ανώτερο όριο όπως φαίνεται στην Εικόνα 5.

Στην συζήτηση του 2.3.2.1 έχουμε δύο άγνωστες πιθανότητες, τις: $x = P(Mπλε|Χαλασμένο)$ και $y = P(Κρύο)$. Το μόνο που μπορούμε να πούμε είναι ότι οι τιμές τους είναι ανάμεσα στο 0,1 μιας και είναι τιμές πιθανοτήτων. Αυτό δίνει τα εξής όρια στην x

$$\Pi_{inf}(Κρύο|Mπλε) = \inf_{(x,y) \in [0,1]^2} P(Κρύο|Mπλε) = 0$$

$$\Pi_{sup}(Κρύο|Mπλε) = \sup_{(x,y) \in [0,1]^2} P(Κρύο|Mπλε) = 1$$



Εικόνα 5: Upper Lower Probability

Έτσι και μετά την εφαρμογή της ανάλυσης είμαστε ακόμα σε κατάσταση άγνοιας. Αυτό συμβαίνει γιατί ο βαθμός ελευθερίας της μοντελοποίησης είναι πολύ μεγάλος. Αν γνωρίζαμε την a priori πιθανότητα y , θα είχαμε:

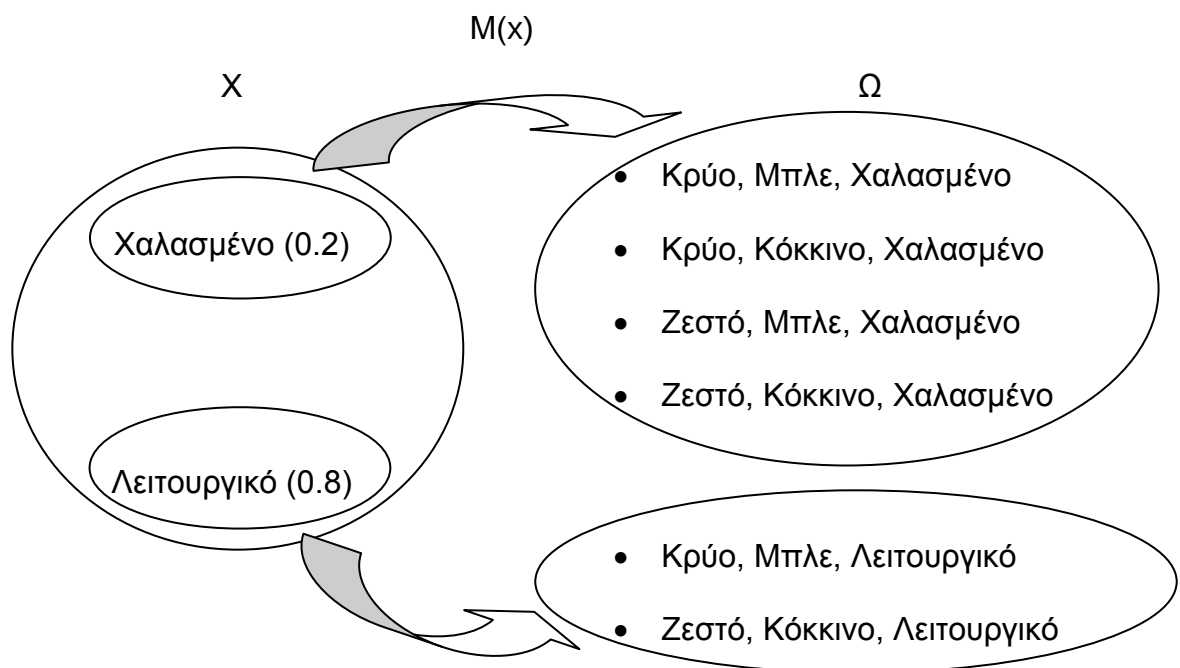
$$\Pi_{inf}(Κρύο|Mπλε) = \inf_{(x,y) \in [0,1]^2} P(Κρύο|Mπλε) = \frac{y}{0.8y + 0.2}$$

$$\Pi_{sup}(Κρύο|Mπλε) = \sup_{(x,y) \in [0,1]^2} P(Κρύο|Mπλε) = 1$$

Οπότε για να μπορούμε να λάβουμε απόφαση για ένα γεγονός, ακόμα και αν γνωρίζουμε το $P(Κρύο)$ το ULP, παραμένει σε μία ανοικτή κατάσταση. Για να αποφύγει αυτού του είδους τα προβλήματα ο Arthur Dempster [12], [9], εισάγει τεχνική συμπερασμού δίχως a-priori πιθανότητες, όπως στα προηγούμενα.

2.3.2.3 Μοντέλο του Dempster

Ο Dempster θεωρεί μία ειδική περίπτωση του ULP. Για το στατικό μέρος του μοντέλου, θεωρεί έναν χώρο X όπου είναι εφοδιασμένος με μία γνωστή κατανομή στα στοιχεία του και μία αντιστοιχία M από τα στοιχεία του M στα στοιχεία του δυναμοσυνόλου 2^Y . Στο δικό μας παράδειγμα έχουμε έναν καλά ορισμένο χώρο πιθανότητας, εκείνο της λειτουργίας του αισθητήρα. Έχουμε $P(\text{Λειτουργικό}) = 0.8$ και $P(\text{Χαλασμένο}) = 0.2$. Αν γνωρίζουμε ότι ο αισθητήρας λειτουργεί καλά τότε ο συνδυασμός (Κρύο, Κόκκινο, Λειτουργικό) δεν μπορεί να υπάρξει. Το ίδιο συμβαίνει και για τον ανάστροφο συνδυασμό (Ζεστό, Κρύο, Λειτουργικό) έτσι ανάμεσα στα δύο σύνολα X και Ω έχουμε την εξής αντιστοιχία που φαίνεται στην Εικόνα 6.



Εικόνα 6: Αναπαράσταση αντιστοιχίας M

Από αυτή την αντιστοιχία ο Dempster ορίζει μία ελάχιστη και μία μέγιστη συνάρτηση πιθανότητας στο σύνολο Ω .

$$\Pi_{inf}(A \in \Omega) = \sum_{x \in X; M(x) \subseteq A} P(X)$$

$$\Pi_{sup}(A \in \Omega) = \sum_{x \in X; M(x) \cap A \neq \emptyset} P(X)$$

Η συνάρτηση ελάχιστης πιθανότητας του $A \in \Omega$, μπορεί να θεωρηθεί ως το άθροισμα όλων των στοιχείων του X των οποίων όλα τα συμβατά στοιχεία ανήκουν στο A . Η κύρια ιδέα πίσω από αυτό είναι, ότι αν κάποιο $x \in X$ είναι συμβατό μόνο με στοιχεία από το σύνολο A , αν υποθέσω ότι λαμβάνω γνώση για το x σημαίνει ότι έχω λάβει γνώση και για το A . Οπότε η πιθανότητα του x υποστηρίζει πλήρως και το A . Η συνάρτηση μέγιστης πιθανότητας μπορεί να θεωρηθεί ως το συμπλήρωμα της συνάρτησης ελάχιστης πιθανότητας. Η ιδέα πίσω από την συνάρτηση μέγιστης πιθανότητας είναι ότι αν ένα στοιχείο $x \in X$ είναι δεν έχει κανένα κοινό στοιχείο με κάποιο A , τότε αν έχουμε x τότε δεν σημαίνει κάτι για την πιθανότητα του x . Σε αντίθετη περίπτωση αν έστω και ένα στοιχείο είναι κοινό με το A , τότε αν έχουμε x τότε μπορεί να συνεπάγεται το A . Κοινώς, η συνάρτηση μέγιστης πιθανότητας είναι η ελάχιστη επαυξημένη με τα στοιχεία που υποστηρίζονται μερικώς.

Στο δυναμικό μέρος του μοντέλου που πρότεινε, ο Dempster πραγματοποιεί την τομή του συνόλου Ω με ένα νέο ενδεικτικό στοιχείο (που προέρχεται π.χ., από γνώση για ένα γεγονός από αισθητήρα). Το σύνολο X παραμένει σταθερό. Από την φύση της ένδειξης ως προς το ποια στοιχεία περιέχει (π.χ., «Μπλε»), κρατιούνται μόνο τα στοιχεία που περιέχουν το στοιχείο Μπλε.

Ενώ η μέθοδος του Dempster στο μέρος της αντιστοίχισης είναι αποδεκτός, εκεί που προκαλεί διαφωνία ([13], [14]) είναι στο γεγονός ότι ενώ έρχεται ένα νέο δεδομένο, το σύνολο X δεν ανανεώνεται σύμφωνα με τις δεσμευμένες πιθανότητες $P(\text{Χαλασμένο}|\text{Μπλε})$, $P(\text{Λειτουργικό}|\text{Μπλε})$. Στην περίπτωση που είχαμε ένδειξη μπλε, σχετικά με την πρώτη περίπτωση,

$$P(\text{Χαλασμένο}|\text{Μπλε}) = \frac{P(\text{Χαλασμένο})P(\text{Μπλε}|\text{Χαλασμένο})}{P(\text{Μπλε})} = \frac{0.2x}{0.2x + 0.8\pi}$$

για να μπορέσει κάποιος να προχωρήσει πρέπει να υποθέσει ότι $x = \pi$. Αυτό σημαίνει ότι θεωρείται γνωστή και υπαρκτή η κατανομή στον χώρο στον οποίο αντιστοιχίζονται τα στοιχεία του συνόλου X . Η έρευνα και η μαθηματική μοντελοποίηση του Dempster χρησιμοποιήθηκε από τον Shafer και άλλους για την επέκταση του παραπάνω μοντέλου. Συγκεκριμένα ο επικεντρώθηκαν στα εξής:

- Στο πώς αναπαριστάται η *υπάρχουσα* γνώση στα σύνολα

- Στην συσχέτιση μεταξύ του γνωστού κόσμου (σύνολο X) και στο πλαίσιο παρατήρησης.
- Στα άνω και κάτω όρια πιθανοτήτων

2.3.2.4 Μοντέλο Μεταφερόμενων Πεποιθήσεων

Το μοντέλο μεταφερόμενων πεποιθήσεων (TBM στο εξής) είναι ένα μαθηματικό μοντέλο το οποίο είναι σχεδιασμένο για να αναπαριστά βαθμούς πεποίθησης. Έχει πολύ κοντινή συγγένεια με την προσέγγιση των Dempster και Shafer με την μόνη και κύρια διαφορά ότι δεν θεωρείται καμία άγνωστη κατανομή πιθανότητας πάνω στο Ω . Το TBM δεν είναι ούτε γενίκευση, ούτε εξειδίκευση, αλλά δημιουργεί μία απόσταση.

Το TBM έχει δύο στάδια:

- Το στάδιο της *πίστης* (credal), κατά το οποίο οι βαθμοί πεποίθησης ανανεώνονται
- Το στάδιο της *απόφασης* (pignistic⁴), κατά το οποίο λαμβάνεται κάποια απόφαση

2.3.2.4.1 Στάδιο Πίστης

Το TBM βασίζεται στην έννοια των βασικών *συναρτήσεων μάζας πεποίθησης* (Basic Belief Mass Function – στο εξής bbm). Ως μάζα ορίζουμε το πεπερασμένο μέγεθος πίστης που δίνεται σε κάποια υπόθεση από το σύνολο Ω . Η ανάθεση μίας τιμής $m(A)$ σε κάποιο υποσύνολο A του Ω , σημαίνει ότι δίνεται ακριβώς $m(A)$ βαθμός πεποίθησης σε αυτό το υποσύνολο. Η μάζα πεποίθησης πρέπει να αθροίζει στο 1 για όλα τα υποσύνολα του Ω .

$$\sum_{A \subseteq \Omega} m(A) = 1$$

Η κύρια διαφορά είναι ότι δεν ακολουθεί τον κανόνα της ένωσης πιθανοτήτων όπως στην κλασσική πιθανότητα, δηλαδή δεν ισχύει ο κανόνας της πρόσθεσης: $m(A \cup B) \neq m(A) + m(B)$. Τα υποσύνολα του Ω που έχουν $m(A) \neq 0$ λέγονται *εστιακά σύνολα* (focal sets). Η ανάθεση σε κάποιο υποσύνολο της τιμής $m(A)$ μπορεί να γίνει είτε με βάση υποκειμενικά στοιχεία, είτε με γνωστές κατανομές πιθανοτήτων.

⁴ Από την λατινική λέξη *Pignus* που σημαίνει στοιχηματίζω.

Ακολουθεί μία λίστα με τις συναρτήσεις που απαρτίζουν το μοντέλο TBM.

Πίστη (Belief) ονομάζεται η αποδεδειγμένη εμπιστοσύνη που έχω στο σύνολο A. Αν υπάρχει κάποιο σύνολο B που είναι υποσύνολο του A τότε αυτό παρέχει αποδεδειγμένο βαθμό εμπιστοσύνης στο σύνολο A. Αυτό μαθηματικά εκφράζεται ως εξής:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B)$$

Σε σχέση με όσα προαναφέρθηκαν στην ανάλυση του Μοντέλου του Dempster, ως belief μπορούμε να δούμε την συνάρτηση ελάχιστης πιθανότητας.

Ευλογοφάνεια (Plausibility) είναι η μέγιστη εμπιστοσύνη που μπορώ να αποδώσω σε ένα υποσύνολο A του Ω.

$$pl(A) = 1 - bel(A) = \sum_{\emptyset \neq B \cap A} m(B)$$

Όπως τοποθετηθήκαμε για τις συναρτήσεις μέγιστης/ελάχιστης πιθανότητας στο κεφάλαιο 2.3.2.3, έτσι και εδώ ισχύει η σχέση

$$pl(A) = bel(A) + εν\ δυναμη\ εμπιστοσύνη(A)$$

Άγνοια (Ignorance) ενός υποσυνόλου A είναι η διαφορά της πίστης και της ευλογοφάνειας.

$$ignorance(A) = pl(A) - bel(A)$$

Αμφιβολία (Doubt) είναι η μέτρηση της πίστης που ποτέ δεν θα δοθεί για το υποσύνολο ενδιαφέροντος A

$$doubt(A) = 1 - pl(\bar{A})$$

Κοινοτυπία (Commonality) είναι μία έννοια χωρίς άμεση εφαρμογή, αλλά χρησιμεύει στην βελτιστοποίηση υπολογισμών, καθώς αποτελεί ένα μέγεθος μέτρησης της υποστήριξης που μπορεί να επιδεχθεί κάθε στοιχείο ενός υποσυνόλου ενός συνόλου A.

$$com(A) = \sum_{A \subseteq B \subseteq \Omega} m(B)$$

Τα μεγέθη που αναφέρθηκαν συνδέονται με την εξής σχέση (αξιοποιώντας τον μετασχηματισμό Möbius [15]).

$$m(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} bel(B)$$

Στο δυναμικό μέρος του TBM, θεωρούμε ότι έχουμε υποστήριξη για ένα γεγονός A, με μάζα πιθανότητας $m(A)$. Έστω ότι δεχόμαστε γνώση για ένα γεγονός B, που μας αποκλείει την πραγματοποίησή του. Οπότε η προηγούμενη μάζα πιθανότητας παραμένει η ίδια αλλά τώρα αναφέρεται στο γεγονός $A \cup \bar{B}$. **Μεταφέρεται** δηλαδή η μάζα από $m(A)$ στο $m(A \cup \bar{B})$. Έστω λοιπόν ότι ζητάμε να μάθουμε με τι πιθανότητα συμβαίνει ένα γεγονός από έναν αισθητήρα. Έστω ότι ο αισθητήρας A αναφέρει ότι υπάρχει κάποια μάζα πιθανότητας για το γεγονός φωτιάς. Θεωρούμε έναν αισθητήρα B (μία ανεξάρτητη πηγή) με μία διαφορετική αναφερόμενη μάζα πιθανότητας. Πώς γίνεται να συνδυαστούν και οι δύο. Στην βιβλιογραφία υπάρχουν πολλοί κανόνες γι' αυτό ακριβώς το θέμα [16], παρόλα αυτά ο πιο διαδεδομένος είναι ο κανόνας του Dempster (Dempster's Rule of Combination):

$$m_{1,2} = (m_1 \oplus m_2)(A) = \frac{1}{1 - K} \sum_{B \cap C = A \neq \emptyset} m_1(B)m_2(C)$$

Εξίσωση 1: Κανόνας Συνδυασμού Ενδείξεων στο αλγόριθμο Dempster Shafer

Όπου

$$K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C)$$

Αν δει κάποιος τον κανόνα που έχει τις ρίζες του στην θεωρία των πιθανοτήτων τότε θα τον κατανοήσει καλύτερα. Ο εν λόγω κανόνας έχει άμεση σχέση με τον κλασσικό συνδυασμό στις πιθανότητες $P(A \cap B) = P(A)P(B)$. Παρόμοιος είναι και ο τρόπος που συνδυάζει γεγονότα ο Dempster.

Αξίζει να σημειώσουμε τον λόγο ύπαρξης του συντελεστή κανονικοποίησης K. Έστω ότι έχουμε δύο ενδείξεις οι οποίες υποστηρίζουν δύο εντελώς διαφορετικά γεγονότα.

Δηλαδή έστω ότι έχουμε $m(X)$ και $m(Y)$ διάφορα του μηδενός και παράλληλα $X \cap Y = \emptyset$.

Αυτό σημαίνει ότι συγκρούονται οι δύο παρατηρήσεις με αποτέλεσμα το γινόμενό τους να είναι μηδενικό. Ένας τρόπος να αποφευχθεί αυτό και για να διατηρήσουμε την

ιδιότητα του αθροίσματος πιθανοτήτων στο 1 είναι απαραίτητο να προχωρήσουμε στην παραπάνω κανονικοποίηση.

Η υλοποίηση του παραπάνω απλού κανόνα αποτελεί ένα πεδίο με ενεργό έρευνα σήμερα. Ο λόγος είναι γιατί η υπολογιστική πολυπλοκότητα του συνδυασμού είναι #P Complete. Συνοπτικά (και διαισθητικά) αυτό φαίνεται, από την ανάγκη υπολογισμού των τιμών κάθε δυνατού υποσυνόλου που πρέπει να θεωρηθεί κατά την ανανέωση της μάζας.

2.3.2.4.2 Στάδιο Απόφασης

Σε ένα δεδομένο πεδίο προβλήματος έστω ότι έχουμε ανανεώσει τις απαραίτητες μεταβλητές με βάση όλη τη διαθέσιμη γνώση (π.χ., από αισθητήρες) και θέλουμε να κατανέμουμε πιθανότητες στα μοναδιαία σύνολα του κόσμου μας, δηλαδή στις υποθέσεις που έχουμε θεωρήσει.

Οι κανόνες για το στάδιο της απόφασης εν γένει ανήκουν σε δύο κατηγορίες ανάλογα τα κριτήρια που απαιτούνται από το πρόβλημα:

Μέγιστη πίστη: Με βάση αυτό το συλλογισμό, απαιτούμε να διαλέγουμε πάντα το γεγονός με τη μεγαλύτερη και εξακριβωμένη υποστήριξη

Μέγιστη ευλογοφάνεια: Με βάση αυτό το συλλογισμό, απλά μας ενδιαφέρει κάποια λύση που δεν περιέχει αντιφάσεις

Το TBM χρησιμοποιεί έναν απλό κανόνα για την μετατροπή μίας μάζας πιθανότητας σε πιθανότητα. Έστω ότι έχουμε δύο υποθέσεις έτσι ώστε $m(A \cup B) = 0.8$, το TBM θα κατανέμει την πίστη σε ίσες πιθανότητες, διαιρώντας με το πλήθος των στοιχείων.

$$\forall x \in \Omega, Bet_p(x) = \sum_{x \in A \subseteq \Omega} \frac{m(A)}{|A|}$$

Εξίσωση 2: Μετατροπή μάζας πιθανότητας σε πιθανότητα κατά TBM

2.3.2.5 Παραδειγματική εφαρμογή DST

Έστω ότι υπάρχει ένα σύνολο από υποθετικά σενάρια. Το σενάριο H, C, P . Αυτά τα (μοναδιαία) σύνολα αποτελούν το frame of discernment $\theta = \{H, C, P\}$ και έστω ότι λαμβάνουμε ενδείξεις για τα παρακάτω σύνολα.

$$m(\{H\}) = 0.3$$

$$m(\{H, C\}) = 0.2$$

$$m(\{H, C, P\}) = 0.5$$

Η πίστη $Bel(A)$ που έχουμε για κάθε δυνατό υποσύνολο του θ υπολογίζεται προσθέτοντας όλα τα $m(B)$ για τα οποία ισχύει ότι το B είναι υποσύνολο του A έχοντας τα παρακάτω (αναφέρουμε χαρακτηριστικά 4 παραδείγματα εκ των 8).

$$Bel(\{H\}) = 0.3$$

$$Bel(\{H, C\}) = 0.3 + 0.2$$

$$Bel(\{H, P\}) = 0.3$$

$$Bel(\{H, C, P\}) = 0.3 + 0.2 + 0.5$$

Η συνάρτηση κοινοτυπίας, $common(A)$ κάθε δυνατού υποσυνόλου θ υπολογίζεται προσθέτοντας όλα τα $m(B)$ των οποίων το A είναι υποσύνολο του B .

$$Common(\{H\}) = 0.3 + 0.2 + 0.5$$

$$Common(\{H, C\}) = 0.3 + 0.5$$

$$Common(\{H, P\}) = 0$$

$$Common(\{H, C, P\}) = 0.5$$

Έστω ότι θέλουμε να συνδυάσουμε τους βαθμούς πεποίθησης και παρατηρούμε ένα πολύ απλό παράδειγμα για την κατανόησή του, θεωρώντας το frame of discernment ως $\theta = \{D, D'\}$ και έστω ότι έχουμε τις εξής μάζες πιθανότητας σε κάποια χρονική στιγμή.

Πίνακας 4: Παραδειγματικό Frame of Discernment

$m_1 \{D\} = 0.8$	$m_2 \{D\} = 0.9$
-------------------	-------------------

$m_1 \{(D')\} = 0$	$m_2 \{(D')\} = 0$
$m_1 \{(D, D')\} = 0.2$	$m_2 \{(D, D')\} = 0.1$

Έχοντας δύο ενδείξεις τις συνδυάζουμε σύμφωνα με τον γνωστό κανόνα συνδυασμού των Dempster και Shafer σύμφωνα με τις εξισώσεις: Εξίσωση 1, Εξίσωση 2.

Πίνακας 5: Πίνακας γινομένων

		m_2		
		$\{(D)\} = 0.9$	$\{(D')\} = 0$	$\{(D, D')\} = 0.1$
m_1	$\{(D)\} = 0.8$	0.72	0	0.08
	$\{(D')\} = 0$	0	0	0
	$\{(D, D')\} = 0.2$	0.18	0	0.02

Το conflict K ισούται με $K=1$

Και σύμφωνα με το κανόνα συνδυασμού οι νέες ανανεωμένες τιμές βαθμού πίστης είναι οι εξής:

$$m_1 \oplus m_2 \{(D)\} = 1 \cdot (0.72 + 0.08 + 0.18) = 0.98$$

$$m_1 \oplus m_2 \{(D')\} = 1 \cdot 0 = 0$$

$$m_1 \oplus m_2 \{(D, D')\} = 1 \cdot 0.02 = 0.02$$

Σύμφωνα με την νέα αποτίμηση με βάση τις ενδείξεις που λάβαμε θεωρούμε ότι το πιο πιθανό belief είναι το D .

ΚΕΦΑΛΑΙΟ 3

ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

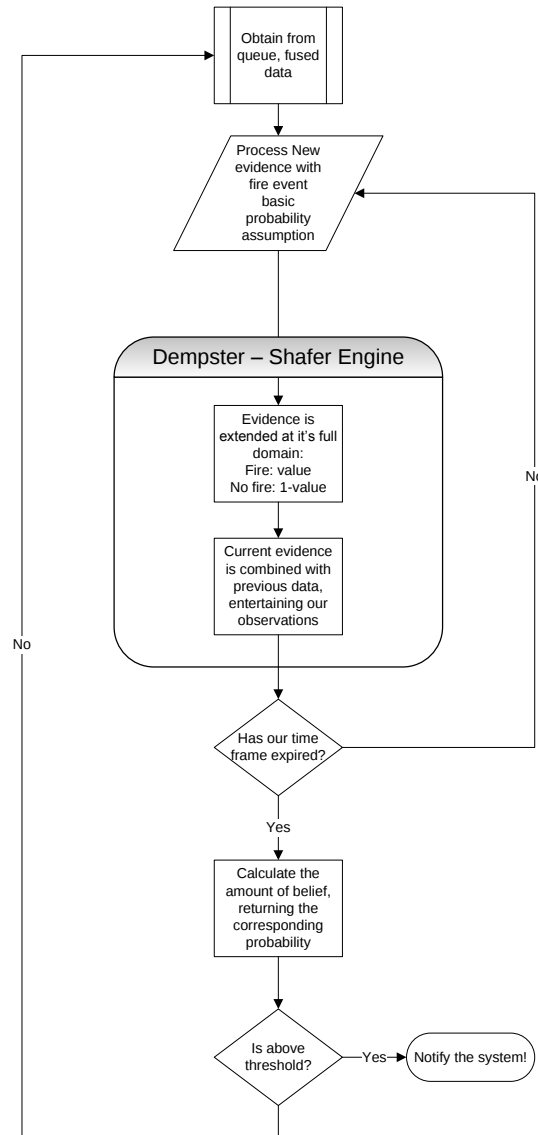
3.1 Περιγραφή

Στην υλοποίησή⁵ που θα παρουσιαστεί παρακάτω, έχει θεωρηθεί το σχήμα που προτείνεται από τους Dempster-Shafer. Η κύρια λειτουργία του DST είναι να ενημερώνει σύνολα από υποθέσεις (hypotheses sets) που περιέχουν συνδυασμούς από δύο υποθέσεις: Η μία ονομάζεται Fire Event και η δεύτερη No Fire Event. Τα δεδομένα της φωτιάς από αισθητήρες μετατρέπονται σε μάζες πιθανότητας, μέσα από ειδικές διαδικασίες [17] και αποτελούν την είσοδο στην εφαρμογή απόφασης. Σε κάθε κύκλο τα εισερχόμενα δεδομένα ενημερώνουν το σύνολο των 2^{Ω} των hypotheses sets με τις νέες μάζες πιθανότητας, χρησιμοποιώντας τον κανόνα του συνδυασμού από την θεωρία DST. Στην Εικόνα 7 παρουσιάζεται η λειτουργικότητα της κλάσης DSEngine η οποία υλοποιεί τον αλγόριθμο. Ο αλγόριθμος λαμβάνει υπόψη αισθητήρες θερμοκρασίας με βαθμό πεποίθησης $t \in [0,1]$ και οπτικούς αισθητήρες βαθμό πεποίθησης $t \in [0,1]$. Αναπαριστούμε τις συναρτήσεις μάζας πιθανότητας πάνω στο δυναμοσύνολο $F = \{\{\}, \{f\}, \{f'\}, \{f, f'\}\}$, όπου f = ένδειξη γεγονότος φωτιάς, f' = ένδειξη για γεγονός μη φωτιάς. Κάθε υποσύνολο του F αναπαριστάται από την HypothesisSet κλάση η οποία περιέχει τα στοιχεία hypotheses φωτιάς και μη φωτιάς, αναπαριστάμενα από την κλάση Hypothesis. Το δυναμοσύνολο F υλοποιείται ως ένα άνυσμα από HypothesesSet αντικείμενα.

- $\{\}$ = κενό σύνολο σημαίνει ότι είμαστε σίγουροι ότι δεν μπορούμε να θεωρήσουμε κάποιο στοιχείο ένδειξης στο F (πλήρης σύγκρουση)
- $\{f\}$ = ανίχνευση φωτιάς για συγκεκριμένο (t,v) στις μετρήσεις των αισθητήρων
- $\{f'\}$ = συμπερασμός μη φωτιάς για συγκεκριμένο (t,v) στις μετρήσεις των αισθητήρων
- $\{f, f'\}$ = πλήρης άγνοια ότι ούτε φωτιά έχει ανιχνευτεί (δεν μπορεί να υπολογιστεί το $\{f\}$) ούτε ότι δεν έχει ανιχνευτεί (αδύνατος ο συμπερασμός $\{f'\}$)
- $\text{mass}_v(A) = v \cdot \text{mass}(A)$,
- $\text{mass}_t(A) = t \cdot \text{mass}(A)$

⁵ Η υλοποίηση είναι μέρος κώδικα που έχει εφαρμοστεί στο ερευνητικό έργο SCIER (Sensor Computing Infrastructure for Environmental Risks) υπ' αριθμόν IST-5035164

Η είσοδος του αλγορίθμου μοντελοποιείται ως μία Evidence class. Αυτή η κλάση λειτουργεί ως adapter (wrapper) της πραγματικής πληροφορίας που παρέχεται από τους ειδικούς (experts). Αυτή ακριβώς η πληροφορία είναι η μάζα πιθανότητας mass (A) $\in [0,1]$.



Εικόνα 7: Dempster - Shafer Workflow

Η βασική συνάρτηση της DSEngine στο διατηρούμενο άνυσμα, με την έλευση ενός νέου στοιχείου, εντοπίζει το σχετικό HypothesisSet με το αντίστοιχο ποσοστό της πεπτοίησης αν το στοιχείο του συνόλου είναι αληθές ή όχι. Αν το A είναι μοναδιαίο σύνολο (singleton), π.χ, $A=\{e\}$, τότε η μάζα είναι $mass(A) = prob(A)$ (Basic Probability Assumption - BPA). Σε αυτό το σημείο πρέπει να σημειώσουμε ότι υιοθετώντας την θεωρία ενδείξεων κάποιος είναι βέβαιος ότι συμπερασμός για το γεγονός της φωτιάς γίνεται αν και μόνο αν τα δεδομένα από τους αισθητήρες υποστηρίζουν γνώση μόνο για την ύπαρξή της και όχι για την μη ύπαρξή της). Αυτό σημαίνει ότι η ευλογοφάνεια ($\{NO$

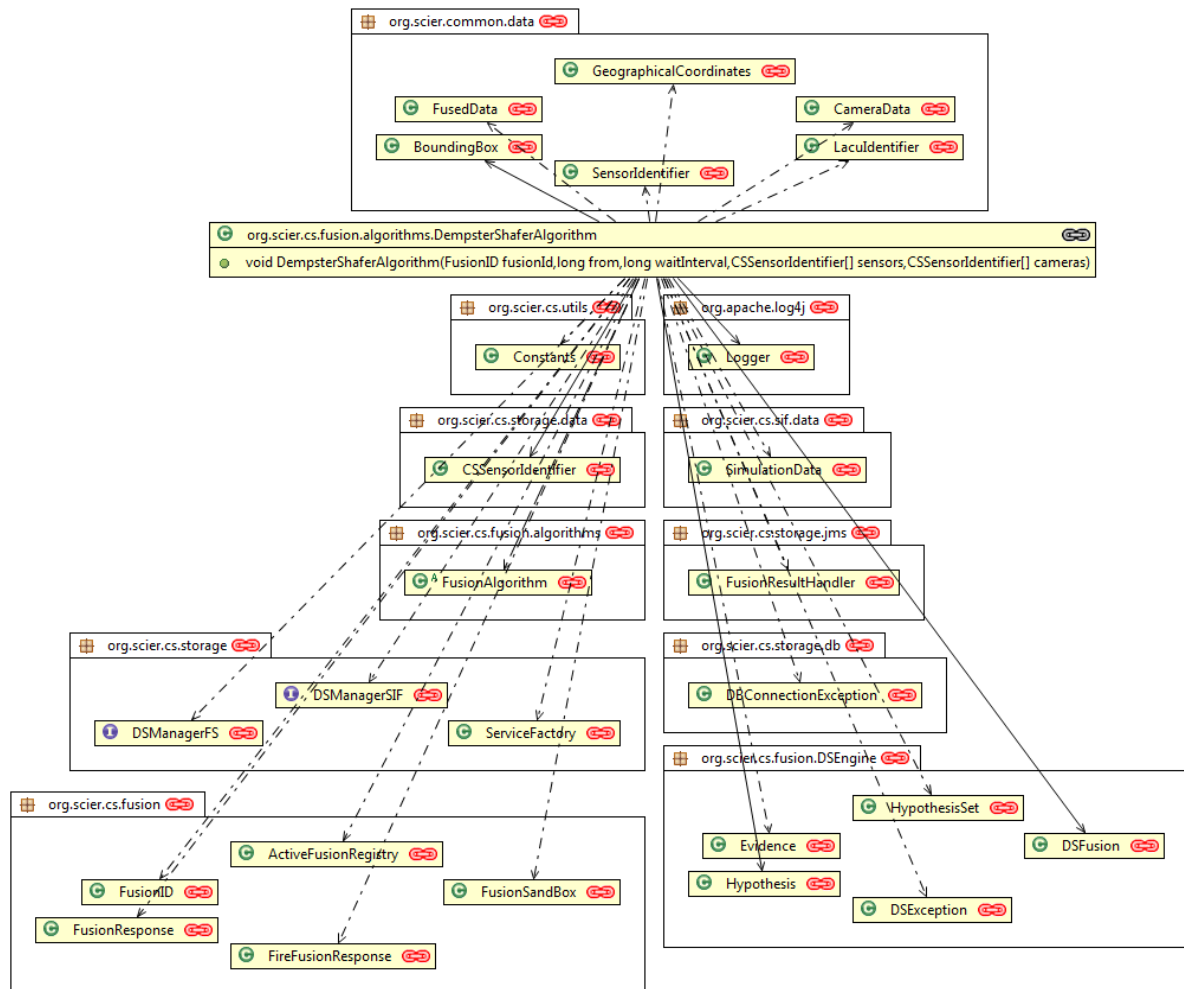
FIRE EVENT}) εξάγεται από την μάζα (FIRE EVENT). Συμπερασματικά η πιθανότητα ενός συνόλου A in Ω υπάρχει ανάμεσα στην ποσότητα πεποίθησης και ευλογοφάνειας. Ο συνδυασμός των ενδείξεων γίνεται μέσω του κανόνα συνδυασμού των Dempster και Shafer (Εξίσωση 1).

Τέλος η DSEngine κλάση, στην συνέχεια, ενημερώνει τους βαθμούς πεποίθησης μέχρι να εξαντληθούν τα δεδομένα στο συγκεκριμένο χρονικό παράθυρο. Το σύστημα συμπερασμού διατηρεί τις συνδυασμένες καλύτερες πεποιθήσεις σε μία δομή δεδομένων που καλείται *beliefpool data structure*, μέσα στο ένα και μοναδικό στιγμιότυπο (singleton design pattern) της DSEngine.

3.2 Τύποι δεδομένων

Για να είναι δυνατή η πρόσβαση παρέχονται διεπαφές του αλγορίθμου μέσα από την κλάση *DempsterShaferAlgorithm* που βρίσκεται μέσα στο βασικό πακέτο της εφαρμογής.

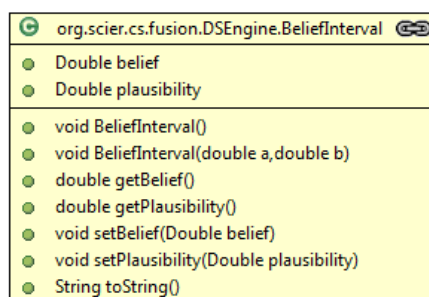
Στην Εικόνα 8 παρουσιάζονται οι απαιτήσεις τύπων που χρησιμοποιούνται από την ίδια την κλάση και παρέχονται σε άλλα namespaces.



Εικόνα 8: Dempster Shafer Algorithm Dependencies

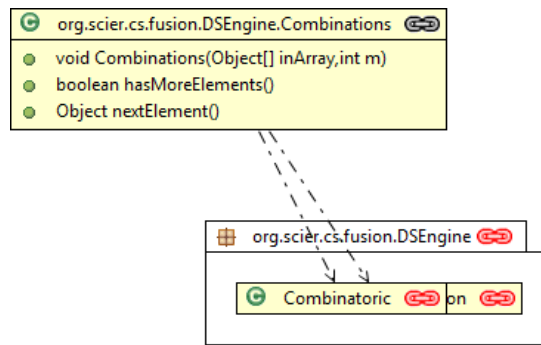
3.3 Κλάσεις υλοποίησης

- Η κλάση BeliefInterval αναπαριστά το διάστημα [belief, plausibility].



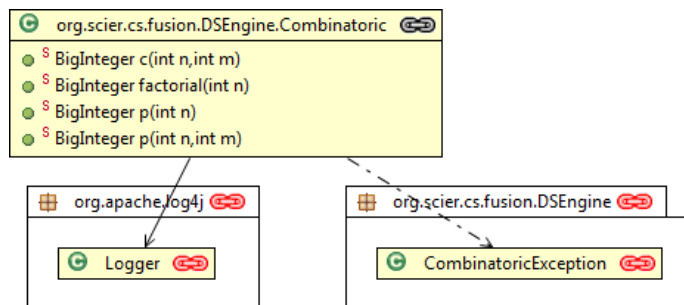
Εικόνα 9: BeliefInterval κλάση

- Η κλάση Combinations παρέχει μία απαρίθμηση από όλα τα υποσύνολα από μία ομάδα n αντικειμένων ανά r .



Εικόνα 10: `Combinations` κλάση

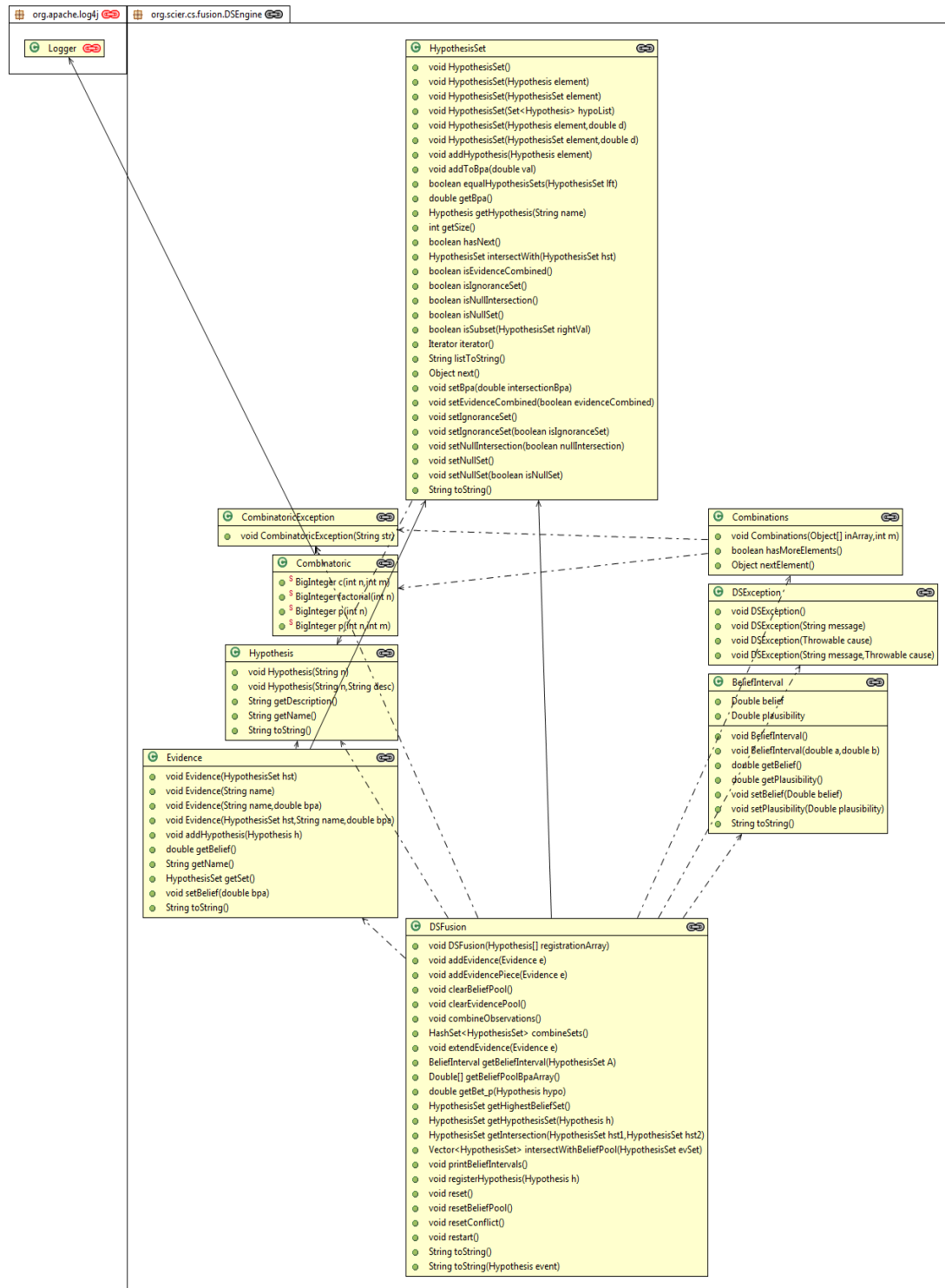
- Η κλάση `Combinatoric` περιέχει μεθόδους για την πραγμάτωση των βασικών συναρτήσεων που παράγουν τους συνδυασμούς



Εικόνα 11: `Combinatoric` κλάση

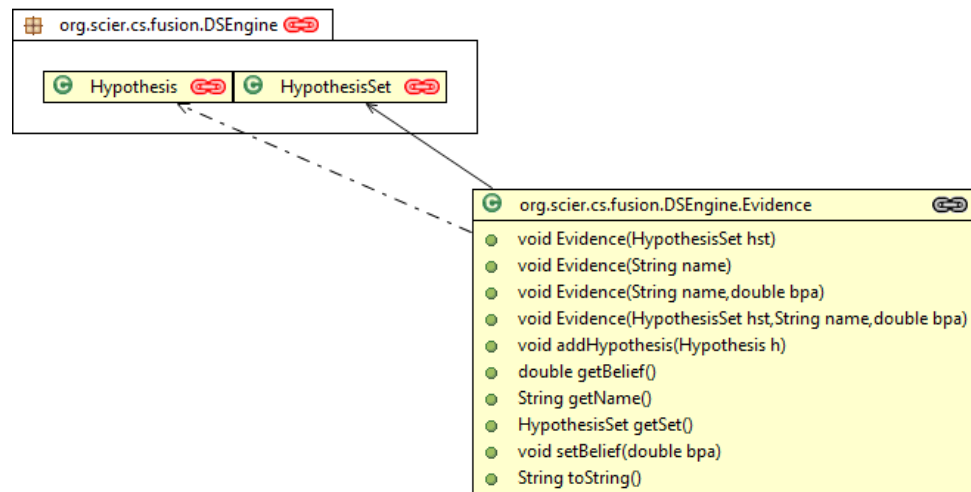
- `DSFusion` είναι η κύρια κλάση του συστήματος σύντηξης δεδομένων με Dempster Shafer. Περιέχει όλα τις hypotheses που θα χρησιμοποιηθούν στην μηχανή συμπερασμού Dempster

Shafer.



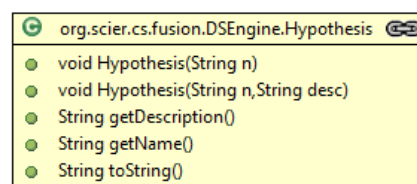
Εικόνα 12: Dempster Shafer κλάση υλοποίησης

- Η κλάση Evidence περιέχει ένα hypothesis set και ένα string περιέχοντας ονομαστική αναφορά.



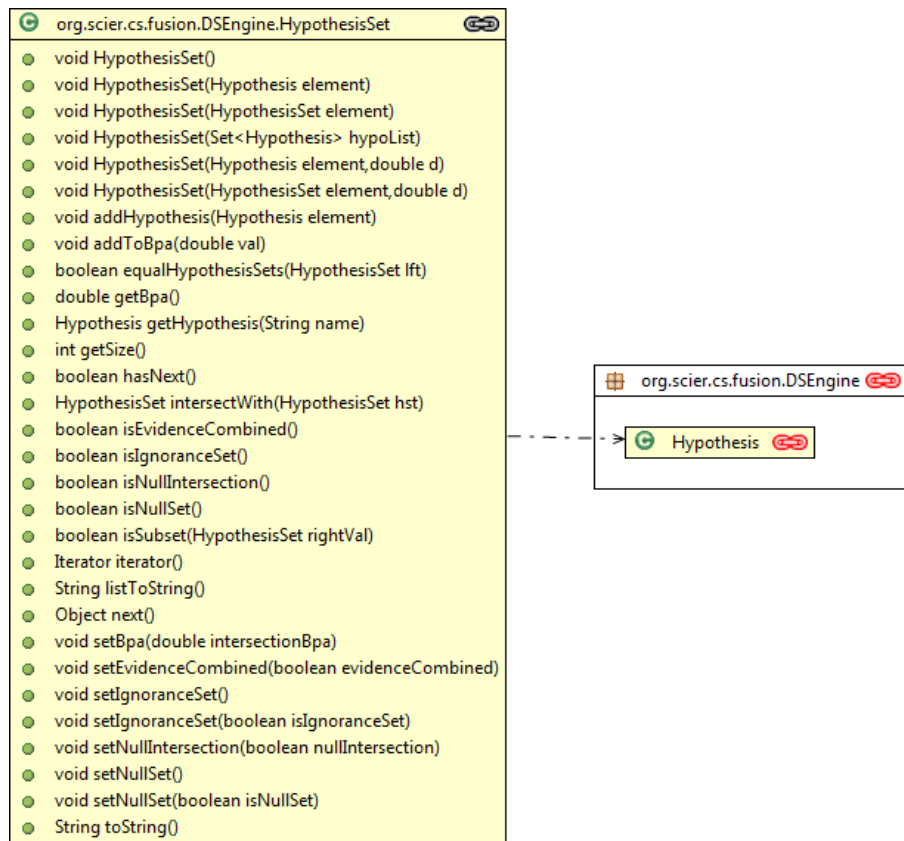
Εικόνα 13: Evidence κλάση

- Η κλάση Hypothesis αναπαριστά ένα hypothesis αντικείμενο



Εικόνα 14: Hypothesis κλάση

- Η κλάση HypothesisSet αναπαριστά ένα αντικείμενο evidence στον DS αλγόριθμο. Μπορεί να περιέχει κανένα, μέχρι και όλα τα στοιχεία (π.χ., $A \cup B \cup C$). Είναι μηδέν αν αναπαριστά το κενό σύνολο και αν έχει όλα τα στοιχεία τότε αναπαριστά το σύνολο άγνοιας



Εικόνα 15: HypothesisSet κλάση

3.4 Εξομοίωση

Για λόγους τεκμηρίωσης θεωρούμε τρία σενάρια σύντηξης δεδομένων. Τα δύο πρώτα αποτελούν παρουσίαση αποτελεσμάτων για απλές περιπτώσεις όταν έχουμε τρεις πηγές, μία κάμερα και δύο αισθητήρες φωτιάς. Στο τρίτο σενάριο, θεωρούμε σειρά από δεδομένα πιθανοτήτων εμφάνισης φωτιάς.

3.4.1 Σενάριο 1

Σε αυτό το σενάριο υποθέτουμε ότι έχουμε έναν αισθητήρα θερμοκρασίας και έναν οπτικό αισθητήρα (κάμερα εδώ και εξής). Υποθέτουμε ότι λαμβάνουμε τρεις πιθανότητες φωτιάς από την κάμερα (0.1, 0.5, 0.9), εκ των οποίων κάθε μία συνδυάζεται με τρεις πιθανότητες που προκύπτουν ως ενδείξεις ενός αισθητήρα θερμοκρασίας (0.1, 0.5, 0.8). Ο Πίνακας 6 δείχνει τα αποτελέσματα του συνδυασμού. Όπως παρατηρείται, αν και μόνο αν και οι δύο πιθανότητες που συμμετέχουν στην διαδικασία είναι μεγαλύτερες από 0.5, τότε η τελική πιθανότητα παρουσιάζεται σχετικά αυξημένη. Συμπερασματικά, αυτό μας παρουσιάζει μία κατάσταση που ερμηνεύεται ως κλασσικές αντιπαρατιθέμενες ενδείξεις (όπως στην ανεπιθύμητη περίπτωση που ένας αισθητήρας δεν δείχνει φωτιά ενώ κάποιος άλλος υποστηρίζει ότι υπάρχει υψηλή πιθανότητα).

3.4.2 Σενάριο 2

Στο δεύτερο σενάριο συνδυάζουμε τις συντηγμένες πιθανότητες από το πρώτο σενάριο με την πιθανότητα ανίχνευσης φωτιάς από έναν ακόμα αισθητήρα. Για τον δεύτερο αισθητήρα θεωρούμε δύο ενδεχόμενες πιθανότητες (0.2 και 0.6). Τα αποτελέσματα παρουσιάζεται στους επόμενους πίνακες (Πίνακας 7, Πίνακας 8). Όπως παρατηρείται από τις εγγραφές (Πίνακας 7, γραμμές 3, 5), η τιμή 0.5 όταν συνδυαστεί με μία μικρή πιθανότητα τότε εξαναγκάζεται το αποτέλεσμα να πέσει στο 0.2.

Αντιθέτως, όπως παρατηρείται, (Πίνακας 8 γραμμές 3, 5), η τιμή 0.5 αν συνδυαστεί με υψηλότερη πιθανότητα, π.χ., 0.6, τότε εξαναγκάζεται να σκαρφλώσει σε αυτή την τιμή ο βαθμός πίστης. Αν όλες οι πιθανότητες είναι μεγαλύτερες από 0.5 τότε η πίστη μας για ένα γεγονός φωτιάς ενισχύεται (όπως φαίνεται από τους προαναφερόμενους πίνακες).

3.4.3 Σενάριο 3

Στο σενάριο 3 θεωρούμε ότι χειριζόμαστε βαθμούς πίστης στο γεγονός της φωτιάς με περισσότερους από 2 αισθητήρες να συμμετέχουν επί της διαδικασίας. Στην Εικόνα 16 αναπαριστούμε την διαδικασία χειρισμού των βαθμών πίστης (κάθετος άξονας) από έναν συγκεκριμένο αριθμό ειδικών (οριζόντιος άξονας). Οι συναρτήσεις πίστης στην πραγματικότητα είναι κατανομές πιθανοτήτων πάνω στο δυναμοσύνολο που ορίσαμε στα προηγούμενα κεφάλαια. Θεωρούμε ότι η συνάρτηση μάζας για το γεγονός της φωτιάς, αναπτύσσεται με γραμμικό αυξητικό τρόπο. Εξετάζοντας την κατανομή, μπορεί κάποιος να καταλάβει ότι όσο τροφοδοτούμε το σύστημα με συγκρουόμενη γνώση τότε η πίστη για το αντίστοιχο γεγονός μειώνεται (πρώτες 20 τιμές, Πίνακας 9).

Πίνακας 6: Data Fusion ενός αισθητήρα θερμοκρασίας με κάμερα

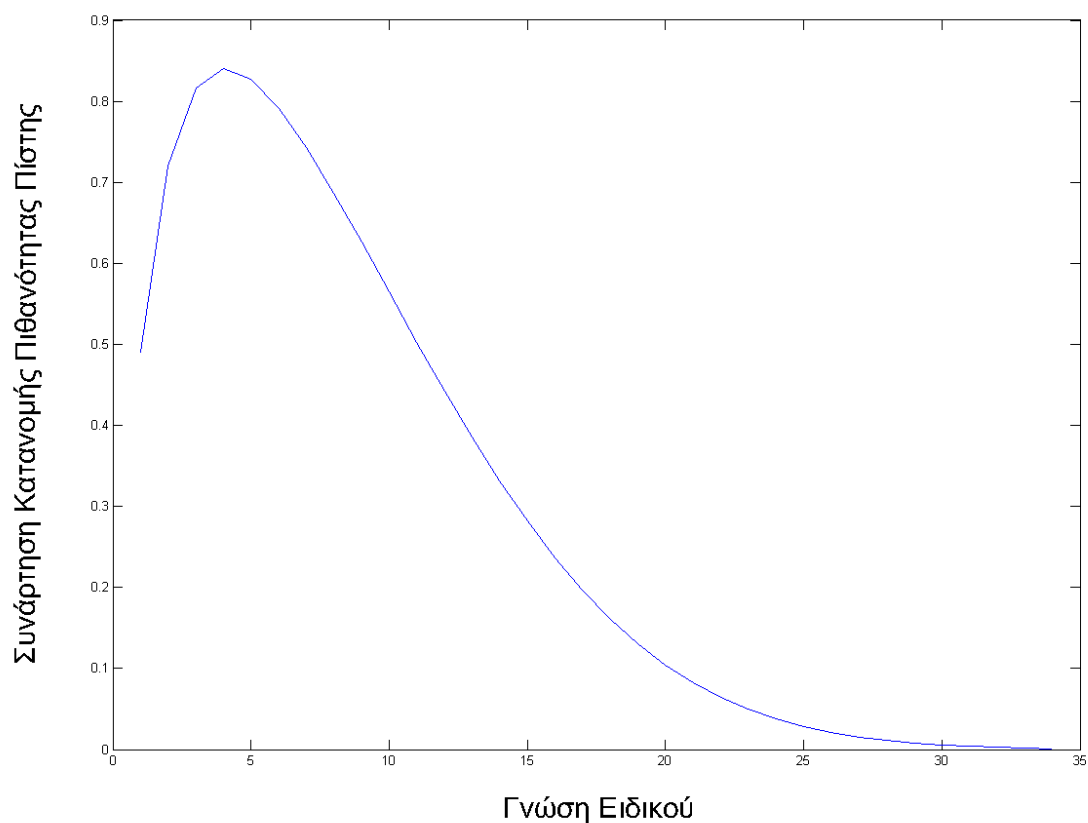
Sensor1	Vision Sensor	Conflict	Sensor1 + Vision
0,1	0,1	0,18	0,0122
0,1	0,5	0,5	0,1
0,1	0,9	0,82	0,5
0,5	0,1	0,5	0,1
0,5	0,5	0,5	0,5
0,5	0,9	0,5	0,9
0,8	0,1	0,74	0,3077
0,8	0,5	0,5	0,8
0,8	0,9	0,26	0,973

Πίνακας 7: Data Fusion ενός δεύτερου αισθητήρα θερμοκρασίας (με πιθανότητα 0.2)

Sensor1 + Vision	Sensor2	Conflict	Fused
0.0122	0.2	0.2073	0.0031
0.1000	0.2	0.26	0.0270
0.5000	0.2	0.5	0.2000
0.1000	0.2	0.26	0.0270
0.5000	0.2	0.5	0.6000
0.9000	0.2	0.74	0.6923
0.3077	0.2	0.3846	0.1000
0.8000	0.2	0.68	0.5000
0.9730	0.2	0.7837	0.9000

Πίνακας 8: Data Fusion ενός δεύτερου αισθητήρα θερμοκρασίας (με πιθανότητα 0.6)

Sensor1 + Vision	Sensor2	Conflict	Fused
0.0122	0.6	0.5975	0.0182
0.1000	0.6	0.58	0.1429
0.5000	0.6	0.5	0.6000
0.1000	0.6	0.58	0.1429
0.5000	0.6	0.5	0.6000
0.9000	0.6	0.42	0.9310
0.3077	0.6	0.5384	0.4000
0.8000	0.6	0.44	0.8571
0.9730	0.6	0.4054	0.9818



Εικόνα 16: Στατιστική πίστης από διαδοχικούς ειδικούς

σ

Πίνακας 9: 20 πρώτες πιθανότητες φαινομένου

0.49
0.7202
0.8160939999999999
0.8409502399999998
0.8270277279999998
0.7911560643199997
0.7424938898175997
0.6863756286321918
0.6262033845552946
0.5643642960997651
0.502665082903791
0.44253081983033604
0.38509214528364233
0.3312231902564324
0.281561074022655
0.23652167815559272
0.19631802826953257
0.1609832245872667
0.13039759447183835
0.10431864778206051

3.5 Σύνοψη

Από τα τρία σενάρια που εξετάσαμε στο Κεφάλαιο 3.4 παρατηρούμε ότι αν δύο συναρτήσεις πεποίθησης είναι πανομοιότυπες τότε ο συνδυασμός τους δεν είναι απαραίτητα ίδιος. Σαν αποτέλεσμα σε έναν επαναληπτικό συνδυασμό τιμών με ίδια τιμή BPA το αποτέλεσμα θα πηγαίνει είτε στο μηδέν είτε στη μονάδα. Μία τέτοια προσέγγιση μπορεί να λυθεί με σταθμισμένες πηγές ενδείξεων, κατανέμοντας υψηλό βαθμό εμπιστοσύνης σε συγκεκριμένες πηγές (π.χ., για το δικό μας πεδίο προβλήματος την κάμερα) και σε άλλες μικρότερο ποσοστό. Επιπροσθέτως ο κανόνας συνδυασμού που εφαρμόσαμε δεν έχει την δυνατότητα να υποστηρίξει μεγάλο αριθμό ενδείξεων. Το αποτέλεσμα μπορεί να θεωρηθεί αξιόπιστο, μόνο αν τα δεδομένα είναι ουσιαστικής συμφωνίας. Επίσης όπως είδαμε για την απόφαση για το σημείο φωτιάς την παίρνουμε με βάση συγκρίσεις προκαθορισμένων κατωφλίων. Σε ένα φαινόμενο όπως η φωτιά μία καλύτερη προσέγγιση θα ήταν να εξετάσουμε την εφαρμογή θεωρίας ασαφών συνόλων και νευρωνικών δικτύων για μία πιο δυναμική προσέγγιση.

Η θεωρία ενδείξεων είναι μία πολύ καλή λύση για σύντηξη δεδομένων, γιατί παρέχει μία πολύ καλά θεμελιωμένη αναπαράσταση γνώσης (μοντελοποιημένη με μαθηματικό και εύληπτο τρόπο). Αναπαριστά πάρα πολύ πιστά την αβεβαιότητα, την ανακρίβεια στα δεδομένα, ακόμα και την πλήρη άγνοια, χωρίς να χρειάζονται παραδοχές. Το μοντέλο ειδικότερα που εξετάσαμε, χρησιμοποιεί μόνο διαθέσιμες πληροφορίες και δεν απαιτεί καμία άλλη πρόσθετη ενημέρωση (πράγμα που εγκυμονεί κινδύνους). Παρέχει έναν απλό τρόπο για απόδοση τιμών σε σύνολα από υποθέσεις για γεγονότα που μας ενδιαφέρουν. Επίσης μία πολύ σημαντική ιδιότητα είναι ότι μπορούν να εξαχθούν πολλά συμπεράσματα, όπως άγνοια, αμφιβολία, πίστη και ευλογοφάνεια. Συγκεκριμένα για τον κανόνα των Dempster – Shafer και τις επεκτάσεις του όπως αυτόν που εξετάσαμε – TBM, μπορούμε να πούμε ότι αποτελεί μία από τις συχνότερες λύσεις στο πεδίο της σύντηξης δεδομένων λόγω των ιδιοτήτων του. Αξιοσημείωτο είναι πως με τον τρόπο που μοντελοποιήθηκε κάποιος μπορεί να σπάσει τον αλγόριθμο στο στατικό και δυναμικό μέρος όπως είδαμε και σχετικά με τον τρόπο ανανέωσης να χρησιμοποιήσει την δική του τακτική, ή όποια θεωρεί καταλληλότερη.

ΠΑΡΑΡΤΗΜΑ

ΚΩΔΙΚΑΣ

Κώδικας Υλοποίησης

Παραθέτονται οι βασικότερες κλάσεις που απαρτίζουν την υλοποίηση του Dempster Shafer αλγόριθμου.

Κλάση DSFusion

```
public class DSFusion {  
  
    private final HypothesisSet NULL_SET;  
  
    private final HypothesisSet IGNORANCE_SET;  
  
    private final String nullHypothesis = "NULL";  
  
    /**  
     * Contains the hypotheses that the current instance of DSFusionEngine will  
     * calculate in Dempster Shafer reasoning procedure.  
     * the universal set: the set of all states under consideration  
     */  
  
    private HashSet<Hypothesis> hypoList = null;  
  
  
    private double conflict;  
  
    /**  
     * The reasoning system maintains its combined best believes  
     * in the beliefPool data structure, which is updated  
     * whenever new evidence is provided by the sensor.  
     * Focal Elements are computed with the combineSets function.  
     * Contains the Theta set (Frame of Discernment).  
     */  
}
```

```
private HashSet<HypothesisSet> beliefPool;

/**
 * Container as the evidence pool
 */

private HashSet<HypothesisSet> evidencePool;

/**
 * Constructor of the main DSFusion Class
 * Static method registerHypothesis must be called first as many times
 * as necessary to register the Frame of Discernment. Then the constructor
initializes
 * the new fusion engine of Demster Shafer reasoning system for the sensor
provided
 * as parameter
 */

public DSFusion (Hypothesis registrationArray []) {
    this.NULL_SET = new HypothesisSet(new Hypothesis(nullHypothesis));
    this.NULL_SET.setNullSet();
    this.IGNORANCE_SET = new HypothesisSet();
    this.IGNORANCE_SET.setIgnoranceSet();
    this.hypoList = new HashSet<Hypothesis> ();
    this.evidencePool = new HashSet<HypothesisSet>();
    this.beliefPool = new HashSet <HypothesisSet> ();
    for (int i=0;i<registrationArray.length; i++) {
        this.hypoList.add(registrationArray[i]);
        this.IGNORANCE_SET.addHypothesis(registrationArray[i]);
    }
}
```

```
        conflict=0;
    }

/**
 * Constructing the Frame Of Discernment.
 * This method must always precede the constructing of DSFusion objects
 * because it is after the creation of the Frame of Discernment, when the private
fields
 * IGNORANCE_SET and NULL_SET will get their values
 *
 * @param hypothesis
 * @throws Exception
 */
public void registerHypothesis (Hypothesis h){
    hypoList.add(h);
}

/**
 * Simply adds one evidence piece to the evidence pool
 * @param e evidence to be combined
 */
public void addEvidencePiece(Evidence e) {
    this.evidencePool.add(e.getSet());
}

/**
 * An evidence piece is added (after extension) at this evidence pool.
```

```
* If this beliefpool is empty then the evidence is simply copied there,  
* if not a combination is called. Evidence Pool is reseted afterwards.  
* @param e evidence to be combined  
* @throws DSEException  
*/
```

```
public void addEvidence (Evidence e) throws DSEException{  
    this.evidencePool.add(e.getSet());  
    extendEvidence(e);  
    if (this.beliefPool.isEmpty()) {  
        this.beliefPool.addAll(this.evidencePool);  
    }  
    else {  
        combineObservations();  
    }  
    clearEvidencePool();  
}
```

```
/**
```

* Extends the given evidence object with an ignorance set with $bpa = 1$ - the given evidence's bpa

* @param e evidence to be extended on the current domain.

* @throws DSEException if basic probability assignment cannot be calculated due to

* breaking the $[0, 1]$ range of the probability.

```
*/
```

```
public void extendEvidence (Evidence e) throws DSEException{  
    double sumOfBpa = 0;
```

```
        for (HypothesisSet i: this.evidencePool) {  
            sumOfBpa+=i.getBpa();  
        }  
        if (sumOfBpa>1 || sumOfBpa<0)  
            throw new DSEException("Evidence caused: out of probability  
range");  
        this.evidencePool.add(new HypothesisSet(this.IGNORANCE_SET, 1-  
sumOfBpa));  
    }
```

```
public void reset(){  
    this.hypoList.clear();  
    clearBeliefPool();  
    clearEvidencePool();  
}
```

```
public void restart(){  
    clearBeliefPool();  
    clearEvidencePool();  
}
```

```
public void clearBeliefPool() {this.beliefPool.clear();}  
public void clearEvidencePool() {this.evidencePool.clear();}
```

```
/**
```

* All possible combinations. For example if the frame of discernment is A, B, C

* the set returned is A, B, C, AB, AC, BC, ABC(theta).

* @return The combination of the frame of discernment. Constructs the theta set.

*/

```
public HashSet<HypothesisSet> combineSets(){
    try {
        HashSet<HypothesisSet> ret = new HashSet<HypothesisSet>();
        ret.add(new HypothesisSet(NULL_SET));
        Object[] hypoListArray = hypoList.toArray();
        for (int r = 1; r <= hypoListArray.length; r++) {
            Combinations c = new Combinations(hypoListArray, r);
            while (c.hasMoreElements()) {
                HypothesisSet hs = new HypothesisSet ();
                Object[] combo = (Object[])c.nextElement();
                for (int i = 0; i < combo.length; i++) {

                    System.out.print(((Hypothesis)combo[i]).getName() + " ");

                    hs.addHypothesis((Hypothesis)combo[i]);

                }
                System.out.println();
                ret.add(hs);
            }
        }
        return ret;
    } catch (CombinatoricException e) {
        e.printStackTrace();
    }
    return null;
}
```

```
/**
 * The belief(A) for a set A is defined as the sum of all the masses of
 * (not necessarily proper) subsets of the set of interest, which in the current
 * context is the beliefPool.
 *
 * @param A set
 * @return the BeliefInterval: [Belief, Plausability]
 */
public BeliefInterval getBeliefInterval(HypothesisSet A){
    double belief=0, plausibility=0;
    HypothesisSet tmp;
    for (Iterator i=this.beliefPool.iterator(); i.hasNext();){
        tmp=(HypothesisSet)i.next();
        if (tmp.isSubset(A))
            belief+=tmp.getBpa();
        if (tmp.intersectWith(A).getSize()!=0)
            plausibility+=tmp.getBpa();
    }
    return new BeliefInterval(belief, plausibility);
}

public void resetBeliefPool(){
    this.beliefPool.clear();
}

public HypothesisSet getHypothesisSet(Hypothesis h){
```

```
        return new HypothesisSet(h);
    }

    public Double [] getBeliefPoolBpaArray(){
        Vector <Double> ret = new Vector<Double>();
        for ( HypothesisSet i : this.beliefPool){
            ret.add(i.getBpa());
        }
        Double[] tmp = new Double[ret.size()];
        ret.toArray(tmp);
        return tmp;
    }

    public HypothesisSet getHighestBeliefSet(){
        HypothesisSet ret = null;
        if (!this.beliefPool.isEmpty()){
            double bpa = 0f;
            double tmp;
            for (Iterator i = this.beliefPool.iterator(); i.hasNext();){
                tmp = ((HypothesisSet)i).getBpa();
                if ( tmp > bpa ){
                    bpa=tmp;
                    ret = (HypothesisSet )i;
                }
                i.next();
            }
        }
    }
```

```
        return ret;

    }

    /**
     * It returns the intersection of the sets
     * @param hst1 First set
     * @param hst2 Second set
     * @return Intersection set with bpa the product of the two sets if are intersected
     */
    public HypothesisSet getIntersection (HypothesisSet hst1, HypothesisSet hst2) {
        HypothesisSet ret = hst1.intersectWith(hst2);
        if (ret == null){
            ret = new HypothesisSet();
            ret.setNullIntersection(true);
        }
        ret.setBpa(hst1.getBpa()*hst2.getBpa());
        return ret;
    }

    /**
     * @param evSet The hypothesis set which is part of the evidence pool
     * @return A vector containing the intersections for the current set with the
     system's belief pool
     */
    public Vector<HypothesisSet> intersectWithBeliefPool(HypothesisSet evSet){
        Vector <HypothesisSet> ret = new Vector <HypothesisSet>();
        HypothesisSet tmp_intersection;
```

```
        for (HypothesisSet i: this.beliefPool) {
            if      (!(tmp_intersection = getIntersection(evSet,
i)).isNullIntersection()) {
                ret.add(tmp_intersection);
            }
            else
                conflict+=tmp_intersection.getBpa();
        }
        evSet.setEvidenceCombined(true);
        return ret;
    }

    /**
     * @param Evidence new Evidence to be combined with the belief pool
     *
     * The use of this function is theoritically demonstrated here
     *
     * http://en.wikipedia.org/wiki/Dempster-Shafer\_theory#Dempster.27s\_rule\_of\_combination
     */
    @SuppressWarnings("unchecked")
    public void combineObservations(){
        Vector <HypothesisSet> intersections = new Vector <HypothesisSet>();
        //Calculating cartesian product based on DS rule combination
        for (HypothesisSet evidence_it : this.evidencePool) {
            intersections.addAll(this.intersectWithBeliefPool(evidence_it));
        }

        Collections.sort(intersections, new Comparer ());
    }
```

```
//Updating BPA's of the belief
for (HypothesisSet i: calcIntersectionSums(intersections)){
    HypothesisSet bpSet = this.getHypothesisSetFromBP(i);
    if (bpSet!= null)
        bpSet.setBpa(i.getBpa());
    else
        this.beliefPool.add(i);
}

}

private HashSet<HypothesisSet> calcIntersectionSums (Vector<HypothesisSet>
intersections){
    HashSet<HypothesisSet> ret = new HashSet<HypothesisSet>();
    HypothesisSet oldHypo = null;

    for (HypothesisSet hypo: intersections){
        if (oldHypo==null || !oldHypo.equalHypothesisSets(hypo)){
            oldHypo = new HypothesisSet(hypo);
            ret.add(oldHypo);
        }
        oldHypo.addToBpa(hypo.getBpa());
    }
    return ret;
}

class Comparer implements Comparator {
```

```
public int compare(Object obj1, Object obj2)
{
    String hst1 = ((HypothesisSet)obj1).listToString();
    String hst2 = ((HypothesisSet)obj2).listToString();

    return hst1.compareTo(hst2);
}

}

/**
 * Prints all the BeliefIntervals
 */
public void printBeliefIntervals(){
    for (HypothesisSet i: this.beliefPool){
        getBeliefInterval(i).toString();
    }
}

/**
 * Checks for existence of the given object in Belief Pool. If yes, it returns the
corresponding
 * hypothesis set
 * @param val: Given hypothesis set
 * @return
 */
private HypothesisSet getHypothesisSetFromBP (HypothesisSet val) {
    for (HypothesisSet i : this.beliefPool) {
```

```
        if (i.equalHypothesisSets(val))
            return i;
    }
    return null;
}
```

```
private HypothesisSet getHypothesisSetFromBP (String val) {
    HypothesisSet hVal = wrapToHypothesisSet(val);
    for (HypothesisSet i : this.beliefPool) {
        if (i.equalHypothesisSets(hVal))
            return i;
    }
    return null;
}
```

```
private HypothesisSet wrapToHypothesisSet(String val) {
    return new HypothesisSet(new Hypothesis(val));
}
```

```
/**
```

```
 * PIGNISTIC LEVEL
```

```
 * The goal of the pignistic level is to construct probabilities from belief functions
when decision
```

```
 * must be made. Probabilities are constructed only at the final stage
```

```
 * @return Probability of the hypothesis
```

```
 */
```

```
public double getBet_p (Hypothesis hypo){
    double prob=0;
    HypothesisSet tmp;
    HypothesisSet A = new HypothesisSet(hypo);
    for (HypothesisSet B: this.beliefPool){
        tmp = getIntersection(A,B);
        if (tmp != null && !tmp.isNullIntersection())
            prob+= (B.getBpa()* tmp.getSize())/(B.getSize()*(1-
this.conflict));
    }
    return prob;
}

public String toString() {
    return "System's belief mass function: \n" +
        this.beliefPool.toString() +
        "\nSystem's Conflict: " +
        this.conflict +
        "\n";
}

public String toString(Hypothesis event) {
    Double val = this.getHypothesisSetFromBP(new
HypothesisSet(event)).getBpa();
    return val.toString();
}

public void resetConflict() {
```

```
        this.conflict=0;

    }

}
```

Κλάση HypothesisSet

```
/**
 * Structure representing an evidence object in DS algorithm.
 * It may contain from zero to all hypotheses. If zero then we represent
 * the null set, if it contains all the elements of the basic hypotheses
 * it represents the ignorance set.
 *
 */

public class HypothesisSet {

    /**
     * hypoList represents the hypothesis set, containing objects of type Hypothesis
     */

    private boolean isIgnoranceSet=false;
    private boolean isNullSet=false;
    private boolean evidenceCombined=false;
    private Set<Hypothesis> hypoList;
    private boolean nullIntersection=false;

    /**
     * Conflict of calculating bpa
     */

    /**
```

* bpa stands for basic probability assingment (mass)

*/

private double bpa=0;

private Iterator internalIteratorVar;

public boolean isNullIntersection() {

return nullIntersection;

}

public void setNullIntersection(boolean nullIntersection) {

this.nullIntersection = nullIntersection;

}

/**

* Constructors for NULL set

*/

public HypothesisSet(){

this.hypoList = new HashSet<Hypothesis>();

this.bpa = 0;

}

public HypothesisSet(Hypothesis element){

this.hypoList = new HashSet<Hypothesis>();

this.hypoList.add(element);

this.bpa = 0;

}

```
public HypothesisSet(Hypothesis element, double d) {  
    this.hypoList = new HashSet<Hypothesis>();  
    this.hypoList.add(element);  
    this.bpa = d;  
}
```

```
public HypothesisSet(HypothesisSet element) {  
    this.hypoList = element.hypoList;  
    this.bpa=0;  
}
```

```
public HypothesisSet(HypothesisSet element, double d) {  
    this.hypoList = element.hypoList;  
    this.bpa=d;  
}
```

```
public void addHypothesis(Hypothesis element){  
    this.hypoList.add(element);  
}
```

```
public double getBpa() {  
    return bpa;  
}
```

```
public void setBpa(double intersectionBpa) {
```

```
        this.bpa = intersectionBpa;
    }

    public void addToBpa(double val) {
        this.bpa+=val;
    }

    public Hypothesis getHypothesis(String name){
        Hypothesis s;
        for (Iterator it = this.hypoList.iterator(); it.hasNext();){
            s = (Hypothesis)it.next();
            if (s.getName()==name){
                return s;
            }
        }
        return null;
    }

    public boolean isSubset(HypothesisSet rightVal){
        if (this.hypoList.containsAll(rightVal.hypoList))
            return true;
        else
            return false;
    }

    public int getSize(){
        return this.hypoList.size();
    }
}
```

```
}
```

```
public HypothesisSet intersectWith(HypothesisSet hst){  
    HashSet<Hypothesis> intersection = new  
    HashSet<Hypothesis>(this.hypoList);  
    intersection.retainAll(hst.hypoList);  
    if (intersection.isEmpty())  
        return null;  
    else  
        return new HypothesisSet(intersection);  
}
```

```
public Iterator iterator(){ return internalIteratorVar = this.hypoList.iterator(); }  
public boolean hasNext(){ return internalIteratorVar.hasNext(); }  
public Object next() { return internalIteratorVar.next(); }
```

```
public boolean equalHypothesisSets (HypothesisSet lft){  
    if (this.hypoList.equals(lft.hypoList))  
        return true;  
    else  
        return false;  
}
```

```
/**
```

```
 * @return the isIgnoranceSet
```

```
*/

public boolean isIgnoranceSet() {

    return isIgnoranceSet;

}

/**

 * @param isIgnoranceSet the isIgnoranceSet to set

 */

public void setIgnoranceSet(boolean isIgnoranceSet) {

    this.isIgnoranceSet = isIgnoranceSet;

}

public void setIgnoranceSet() {

    this.isIgnoranceSet = true;

}

/**

 * @return the isNullSet

 */

public boolean isNullSet() {

    return isNullSet;

}

/**

 * @param isNullSet the isNullSet to set

 */

public void setNullSet(boolean isNullSet) {
```

```
        this.isNullSet = isNullSet;
    }

    public void setNullSet() {
        this.isNullSet = true;
    }

    public String toString() {
        String ret = new String();
        for (Hypothesis i : this.hypoList) {
            ret += i.toString() + " ";
        }
        ret += "with BPA: " + this.bpa + "\n";
        return ret;
    }

    public String listToString() {
        String ret = new String();
        for (Hypothesis i : this.hypoList) {
            ret += i.toString() + " ";
        }
        return ret;
    }

    public HypothesisSet(Set<Hypothesis> hypoList) {
        super();
        this.hypoList = hypoList;
    }
}
```

```
}

/**
 * @return True if it has already been combined, false otherwise. Default is false
 */
public boolean isEvidenceCombined() {
    return evidenceCombined;
}

/**
 * @param evidenceCombined True if we has already taken under consideration
this hypothesis set
 * in combination sequence
 */
public void setEvidenceCombined(boolean evidenceCombined) {
    this.evidenceCombined = evidenceCombined;
}

}
```

Κλάση Hypothesis

```
public class Hypothesis {
    private String name;
    private String description;
    public Hypothesis(String n){
        name = new String(n);
    }
}
```

```
public Hypothesis(String n, String desc){  
    name = new String(n);  
    description = new String(desc);  
}  
  
public String getDescription() {  
    return description;  
}  
  
public String getName() {  
    return name;  
}  
  
public String toString() {  
    return name;  
}  
}
```

Κλάση Hypothesis

```
public class Hypothesis {  
    private String name;  
    private String description;  
    public Hypothesis(String n){  
        name = new String(n);  
    }  
    public Hypothesis(String n, String desc){  
        name = new String(n);  
        description = new String(desc);  
    }  
    public String getDescription() {
```

```
        return description;
    }

    public String getName() {
        return name;
    }

    public String toString() {
        return name;
    }
}
```

Κλάση Evidence

```
public class Evidence {
    HypothesisSet hst;
    String name;

    public Evidence(HypothesisSet hst, String name, double bpa) {
        super();
        this.hst = hst;
        this.name = name;
        this.hst.setBpa(bpa);
    }

    public Evidence(String name, double bpa) {
        super();
        this.name = name;
        this.hst.setBpa(bpa);
    }
}
```

```
public Evidence(HypothesisSet hst) {  
    super();  
    this.hst = hst;  
}  
  
public Evidence(String name) {  
    super();  
    this.hst = new HypothesisSet();  
    this.name = name;  
}  
  
public void addHypothesis(Hypothesis h){  
    this.hst.addHypothesis(h);  
}  
  
public double getBelief (){  
    return this.hst.getBpa();  
}  
  
public void setBelief(double bpa){  
    this.hst.setBpa(bpa);  
}  
  
public HypothesisSet getSet() {  
    return hst;  
}
```

```
public String getName() {  
    return name;  
}  
  
public String toString() {  
    return "Received evidence: " +this.getSet().toString();  
}  
}
```

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

σύντηξη δεδομένων	data fusion
διανύσματα χαρακτηριστικών	feature vectors
άγνοιας	ignorance
ανακρίβεια	imprecision
ανεπάρκεια	incompleteness
αβεβαιότητα	uncertainty
μη μονοτονική λογική	non-monotonic logic
ασαφή σύνολα	fuzzy sets
θεωρία δυνατοτήτων	possibility theory
θεωρία ενδείξεων	evidence theory
συναρτήσεις πεποίθησης	belief functions
βαθμοί πεποίθησης	degrees of belief
ένδειξη	evidence
μοντέλο ανώτερων κατώτερων	upper lower probabilities
μοντέλο μεταφερόμενων πεποιθήσεων	transferable belief model
πλαίσιο παρατήρησης	frame of discernment
στάδιο της πίστης	credal level
στάδιο της απόφασης	pignistic level
συναρτήσεις μάζας πεποίθησης	basic belief mass functions
εστιακά σύνολα	focal sets
πίστη	belief
ευλογοφάνεια	plausibility
άγνοια	ignorance
αμφιβολία	doubt

ΑΝΑΦΟΡΕΣ

- [1] A. Greenfield, *"Everyware: The Dawning Age of Ubiquitous Computing,"* New Riders Publishing, 2006.
- [2] W.N. Carter, *"Disaster Management: A Disaster Manager's Handbook,"* Asian Development Bank, 1991.
- [3] P.G.W. Keen, *"Decision Support Systems: An Organizational Perspective,"* Addison-Wesley, 1978.
- [4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 38, 2002, pp. 393--422.
- [5] C. Elkan, "The Expectation-Maximization Algorithm," 2007.
- [6] M.E. Liggins, D.L. Hall, and J. Llinas, *"Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition,"* CRC, 2008.
- [7] P. Smets, "Varieties of ignorance and the need for well-founded theories," *Information Sciences*, vol. 57, 1991, pp. 135--144.
- [8] P. Smets, "Imperfect information: Imprecision-Uncertainty," *Uncertainty Management in Information Systems. From Needs to Solutions*, 1997, pp. 225-254.
- [9] G. Shafer, *"A mathematical theory of evidence,"* Princeton University Press, 1976.
- [10] S. Russell and P. Norvig, *"Artificial Intelligence: A Modern Approach (2nd Edition),"* Prentice Hall, 2002.
- [11] P. Smets, "What is Dempster-Shafer's model?," *Advances in the Dempster-Shafer theory of evidence*, John Wiley & Sons, Inc., 1994, pp. 5-34.
- [12] A.P. Dempster and H.U.C.M.D.O. STATISTICS, "Upper and lower probabilities induced by a multi-valued mapping," Springer, 1966.
- [13] I. Levi, "Consonance, dissonance and evidentiary mechanisms," *Evidentiary Value; Philosophical, Judicial and Psychological Aspects of a Theory*, pp. 27--43.
- [14] G. Shafer, "Lindley's paradox," *Journal of the American Statistical Association*, 1982, pp. 325-334.
- [15] N. Wilson, "Algorithms for dempster-shafer theory," *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5, 2000, pp. 421--475.
- [16] K. Sentz and S. Ferson, "Combination of evidence in Dempster-Shafer theory," Report No. SAND2002, vol. 835, 2002.
- [17] E. Gombay and D. Serban, "An adaptation of Page's CUSUM test for change detection," *Periodica Mathematica Hungarica*, vol. 50, 2005, pp. 135-147.