



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Περιβάλλον-Πλαίσιο Εκτίμησης Εμπιστοσύνης
σε Υπηρεσίες
και
σε Πόρους Υπηρεσιοκεντρικών Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
Κωνσταντίνου Τσιούνη

Επιβλέπων : Κωνσταντίνος Κοντογιάννης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2018



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

**Περιβάλλον-Πλαίσιο Εκτίμησης Εμπιστοσύνης
Σε Υπηρεσίες
και
σε Πόρους Υπηρεσιοκεντρικών Συστημάτων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

Κωνσταντίνου Τσιούνη

Επιβλέπων : Κωνσταντίνος Κοντογιάννης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14^η Ιουνίου 2018

(Υπογραφή)

.....

Κωνσταντίνος Κοντογιάννης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Γεώργιος Στάμου
Αναπλ. Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2018

.....
Κωνσταντίνος Ε. Τσιούνης

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

Copyright @ Τσιούνης Ε. Κωνσταντίνος, 2018.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής μου εργασίας, Καθηγητή κύριο Κώστα Κοντογιάννη, για την αμέριστη βοήθεια, στήριξη και υπομονή που επέδειξε καθ' όλη την διάρκεια της εκπόνησής της. Είναι ο άνθρωπος που με ώθησε να ασχοληθώ εκτενέστερα με ερευνητικά θέματα της Τεχνολογίας Λογισμικού. Οι γνώσεις του, η διάθεση του να τις μοιραστεί μαζί μου, και η καθοδήγησή του, συντέλεσαν στο να αποκομίσω τα μέγιστα από την παρούσα εργασία.

Ένα μεγάλο ευχαριστώ αξίζει, επίσης, σε όλους τους συμφοιτητές μου, που με στήριξαν σε όλες τις στιγμές της φοίτησής μου, εύκολες αλλά και δύσκολες. Μοιραστήκαμε στιγμές και γνώσεις μέσα σε αυτά τα χρόνια, και έπαιξαν καθοριστικό ρόλο στην διαμόρφωση του ακαδημαϊκού μου υπόβαθρου αλλά και της προσωπικότητάς μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που υπήρξε αρωγός και υποστηρικτής, τόσο των αποφάσεών μου, όσο και των επιλογών μου, παρ' όλο που αυτό δεν ήταν πάντα εύκολο για εκείνους.

Κώστας

Περίληψη

Σκοπός της παρούσας διπλωματικής είναι η ανάπτυξη ενός περιβάλλοντος-πλαίσιου που επιτρέπει σε χρήστες να λαμβάνουν μια εκτίμηση σχετικά με την εμπιστοσύνη που απολαμβάνει μια υπηρεσία, που προσφέρεται από έναν πάροχο σε συγκεκριμένο περιεχόμενο. Δεδομένης της αυξανόμενης συχνότητας χρησιμοποίησης υπηρεσιών στο διαδίκτυο, η ανάγκη για γρήγορη και αποδοτική αξιολόγηση της εμπιστοσύνης τους αυξάνεται αντίστοιχα. Το περιβάλλον-πλαίσιο καλείται να καλύψει αυτή την ανάγκη υπολογίζοντας μια εκτίμηση της εμπιστοσύνης στην υπηρεσία που παρουσιάζει ενδιαφέρον, λαμβάνοντας υπόψιν τις προτιμήσεις του χρήστη, και όσον αφορά τις απαιτήσεις του από την υπηρεσία, αλλά και όσον αφορά τις διακριτές πληροφορίες που συντελούν στην κάλυψη ή όχι της εκάστοτε απαίτησης.

Για κάθε απαίτηση που χρησιμοποιείται στο περιβάλλον-πλαίσιο, απαραίτητος θεωρείται ένας ορισμός της. Ο ορισμός αυτός περιλαμβάνει τις εξωτερικές πληροφορίες, που επηρεάζουν την απαίτηση, και οι συσχετίσεις ανάμεσά τους. Το framework μοντελοποιεί τις ανάγκες του χρήστη και παράγει ένα δέντρο στόχων, που αντιστοιχεί στην κάθε απαίτηση. Με χρήση τεχνικών λογικού συμπερασμού που βασίζονται στα Markov Logic Networks, υπολογίζεται η εκτίμηση ικανοποίησης της εκάστοτε απαίτησης. Πλεονέκτημα αυτής της μεθόδου είναι ότι ο υπολογισμός αυτός μπορεί να γίνει και με έλλειψη του συνόλου των πληροφοριών.

Ακολούθως, ζητείται ένας ορισμός της συνολικής εκτίμησης της εμπιστοσύνης στην υπηρεσία. Ένας γράφος που ανταποκρίνεται στον ορισμό αυτό παράγεται σύμφωνα με συγκεκριμένα κριτήρια. Στην συνέχεια, εκτελείται ο αλγόριθμος TidalTrust, ο οποίος αρχικά προτάθηκε για κοινωνικά δίκτυα, πάνω στον γράφο αυτό. Το αποτέλεσμα του αλγορίθμου παρέχεται στον χρήστη ως εκτίμησης της εμπιστοσύνης που απολαμβάνει η υπηρεσία, η οποία ερευνάται.

Αλλαγές σε πληροφορίες και μοντέλα μπορεί, επίσης, να προκύψουν. Το περιβάλλον-πλαίσιο ανταποκρίνεται σε αυτές τις μεταβολές ανακαλύπτοντάς τις και επαναυπολογίζοντας την εκτίμηση της εμπιστοσύνης, όπου αυτό κρίνεται αναγκαίο. Αντίστοιχη συμπεριφορά ακολουθείται και σε περίπτωση που το χρονικό πλαίσιο εγκυρότητας των πληροφοριών παρέλθει. Αυτό το πλαίσιο ορίζεται από τον χρήστη και οδηγεί σε ανανέωση των παρεχόμενων εκτιμήσεων ανά τακτά χρονικά διαστήματα.

Η αρχιτεκτονική του περιβάλλοντος-πλαίσιου που ορίστηκε και υλοποιήθηκε, στο πλαίσιο της παρούσας διπλωματικής, στηρίζεται στα σχεδιαστικά μορφήματα του Μαυροπίνακα και του Publisher/Subscriber. Αυτή η επιλογή επιτρέπει την αποσύνδεση ανάμεσα στα στάδια της διαδικασίας εκτίμησης της εμπιστοσύνης, και, κατ' επέκταση την δυνατότητα εκτέλεσης μόνο των σταδίων που είναι απολύτως απαραίτητα όταν παρουσιαστεί μια οποιαδήποτε μεταβολή στα δεδομένα.

Λέξεις κλειδιά: *«Δέντρα στόχων, προτασιακή ικανοποιησιμότητα, τεχνολογία λογισμικού, συλλογή εξωτερικών πληροφοριών, ικανοποίηση απαιτήσεων, εκτίμηση εμπιστοσύνης»*

Abstract

The main goal of this diploma thesis is the development of a framework that allows users to receive an assessment of the trust a service, provided by a single provider on specific context, is entitled to. Given the increased frequency online services are used, the need for swift and efficient evaluation of their reliability increases respectively. The framework is designed to cover this need by assessing the extend to which a service in interest is trusted, keeping in mind the user's preferences, regarding requirements from the service on one hand, and external information, associated with each individual requirement, and its level of satisfaction, on the other.

For each and every requirement, that is used in the framework, a definition is required. The definition consists of external information, that have an impact on the requirement in question, as well as their respective correlations. The framework models the user's needs and, subsequently, creates a goal tree, representing each requirement. Using logic inference techniques, relying heavily on Markov Logic Networks, an appreciation of a requirement's satisfiability is calculated. Calculation can be performed with lack of information as well, which is an advantage of this method.

A definition of the overall trust, and the requirements from which it consists of, is, moreover, required by the framework. A graph corresponding to the given definition is constructed, based on certain criteria. The TidalTrust algorithm, which was initially proposed for social networks, is then applied upon this graph. The algorithm's result value is provided to the user as an assessment of the trustworthiness of the service, which is evaluated.

Alterations on information and models can emerge later. The framework responds to these changes by discovering them, at first, and reinferring trust on a service, wherever it's deemed necessary, afterwards. Similar behavior is portrayed when the time frame of information validity expires. This frame is supplied to the framework by the user, and is responsible for renewing the provided information, in specific time intervals.

The framework's architecture, which is defined and materialized, as part of this diploma thesis, is based on the Blackboard architectural pattern, as well as the Publisher/Subscriber architectural pattern. Such a choice enables the decoupling among the discrete stages of the trust inference process, and thus the possibility of executing the stages that are absolutely essential for the service assessment, given the alterations that have occurred at that certain point.

Keywords: *«goal trees, propositional satisfiability, software engineering, external information acquirement, requirement satisfiability, trust inference»*

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	1
Πίνακας Σχημάτων	5
Κεφάλαιο 1: Εισαγωγή	8
1.1 Κίνητρο	8
1.2 Περιγραφή Προβλήματος.....	9
1.3 Συνεισφορά της διπλωματικής εργασίας	9
1.4 Οργάνωση κειμένου	10
Κεφάλαιο 2: Σχετικές Εργασίες	12
2.1 Εισαγωγή.....	12
2.2 Μοντελοποίηση συστημάτων λογισμικού	12
2.2.1 MOF	12
2.2.2 XMI.....	13
2.2.3 Eclipse Modeling Framework.....	14
2.3 Δέντρα Στόχων (Goal Trees).....	15
2.4 Περιβάλλοντα συλλογιστικής	17
2.4.1 Λογική Πρώτης Τάξης.....	18
2.4.2 Μαρκοβιανά Λογικά Δίκτυα (MLN).....	19
2.5 Συμπερασμός εμπιστοσύνης	22
Κεφάλαιο 3: Αρχιτεκτονική Συστήματος.....	25
3.1 Εισαγωγή.....	25
3.2 Επισκόπηση της αρχιτεκτονικής	26
3.3 Αναλυτική περιγραφή των Components	30
3.3.1 Strategy	30
3.3.2 InformationGathering	31
3.3.3 ReasoningModule	33
3.3.4 Alarmer	34

3.3.5	TrustEvaluator.....	36
3.3.6	Pub/Sub Module.....	37
3.4	Επικοινωνία μεταξύ των Components του συστήματος	39
3.4.1	Ακολουθιακό Διάγραμμα Δημιουργίας Νέων Δεδομένων	39
3.4.2	Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Δέντρου Στόχων Απαίτησης	41
3.4.3	Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Απαίτησης	42
3.4.4	Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Γράφου Εκτίμησης Εμπιστοσύνης.	43
3.4.5	Ακολουθιακό Διάγραμμα Λήξης Χρόνου	44
Κεφάλαιο 4:	Περιγραφή Διαδικασίας	47
4.1	Εισαγωγή.....	47
4.2	Τυποποίηση συμμετεχόντων μερών.....	47
4.2.1	Πάροχος υπηρεσίας (Service Provider)	47
4.2.2	Απαιτήσεις από την υπηρεσία (Requirements).....	47
4.2.3	Χρονικό πλαίσιο (Time Frame)	48
4.2.4	Περιεχόμενο (Context)	48
4.3	Εννοιολογικό Μοντέλο (Domain model).....	48
4.3.1	Κλάση Entity.....	50
4.3.2	Κλάση TrustSpecification.....	50
4.3.3	Κλάση TrustComputation	50
4.3.4	Κλάση TidalTrust	50
4.3.5	Κλάση Expression.....	51
4.3.6	Κλάση Context.....	51
4.3.7	Κλάση ContextualElement	51
4.3.8	Κλάση Service	52
4.3.9	Κλάση ContextualInfo	52
4.3.10	Κλάση Requirement.....	52
4.3.11	Κλάση RequirementEvaluation	53
4.3.12	Κλάση EventHandler	53
4.3.13	Κλάση Impact	53
4.3.14	Κλάση Experience	54

4.3.15	Κλάση Constraint.....	54
4.3.16	Κλάση Time	54
4.3.17	Κλάση Resources	55
4.4	Τυποποίηση λειτουργικότητας.....	55
4.4.1	Διαγράμματα δραστηριοτήτων	55
4.4.2	Ακολουθιακά διαγράμματα	58
Κεφάλαιο 5:	Εκτίμηση εμπιστοσύνης (Trust Evaluation)	64
5.1	Κατασκευή γράφου απαιτήσεων.....	64
5.1.1	Ορισμός Εμπιστοσύνης.....	64
5.1.2	Γράφος απαιτήσεων	65
5.1.3	Απόδοση πιθανοτήτων σε κόμβους	66
5.2	Αλγόριθμος εκτίμησης εμπιστοσύνης σε μια υπηρεσία.....	70
Κεφάλαιο 6:	Εκτίμηση απόδοσης	79
6.1	Χρονική Απόδοση (Time Performance)	79
6.1.1	Ανάλυση μεγέθους γράφου.....	80
6.1.2	Ανάλυση πυκνότητας γράφου.....	80
6.1.3	Ανάλυση Μήκους Μονοπατιού	81
6.1.4	Ανάλυση αριθμού Κόμβων ανά Επίπεδο.....	82
6.1.5	Ανάλυση αριθμού Συσσωρεύσεων	83
6.2	Σταθερότητα Συστήματος	84
Κεφάλαιο 7:	Επίλογος.....	87
7.1	Σύνοψη και συμπεράσματα.....	87
7.2	Μελλοντικές επεκτάσεις	88
Κεφάλαιο 8:	Βιβλιογραφία.....	91

Πίνακας Σχημάτων

Σχήμα 2.2-1 MOF Levels	13
Σχήμα 2.3-1 Παράδειγμα δέντρου με hard και soft goals.....	16
Σχήμα 2.3-2 Παράδειγμα δέντρου στόχων με soft goals και contributions.....	17
Σχήμα 3.2-1 Ψηφιδικό διάγραμμα αρχιτεκτονικής περιβάλλοντος-πλαισίου	27
Σχήμα 3.3-1 Ψηφιδικό διάγραμμα κλάσης Strategy	31
Σχήμα 3.3-2 Ψηφιδικό διάγραμμα κλάσης InformationGathering	32
Σχήμα 3.3-3 Ψηφιδικό διάγραμμα κλάσης ReasoningModule	34
Σχήμα 3.3-4 Ψηφιδικό διάγραμμα κλάσης Alarm	35
Σχήμα 3.3-5 Ψηφιδικό διάγραμμα κλάσης TrustEvaluator	36
Σχήμα 3.3-6 Ψηφιδικό διάγραμμα κλάσης Pub/Sub Module	38
Σχήμα 3.4-1 Ακολουθιακό Διάγραμμα Δημιουργίας Νέων Δεδομένων	40
Σχήμα 3.4-2 Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Δέντρου Στόχων Απαίτησης.....	41
Σχήμα 3.4-3 Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Απαίτησης.....	43
Σχήμα 3.4-4 Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Γράφου Εκτίμησης Εμπιστοσύνης ...	44
Σχήμα 3.4-5 Ακολουθιακό Διάγραμμα Λήξης Χρόνου.....	45
Σχήμα 4.3-1 Διάγραμμα Κλάσεων του Εννοιολογικού Μοντέλου	49
Σχήμα 4.4-1 Διάγραμμα Δραστηριότητας Δημιουργίας Νέας Πληροφορίας.....	56
Σχήμα 4.4-2 Διάγραμμα Δραστηριότητας Λήξης Χρονικού Πλαισίου.....	56
Σχήμα 4.4-3 Διάγραμμα Δραστηριότητας Αλλαγής Μοντέλου Απαίτησης.....	57
Σχήμα 4.4-4 Διάγραμμα Δραστηριότητας Αλλαγής Μοντέλου Εκτίμησης Εμπιστοσύνης	58
Σχήμα 4.4-5 Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Πληροφορίας	59
Σχήμα 4.4-6 Ακολουθιακό Διάγραμμα Λήξης Χρονικού Πλαισίου	60
Σχήμα 4.4-7 Ακολουθιακό Διάγραμμα Αλλαγής Μοντέλου Απαίτησης	61
Σχήμα 4.4-8 Ακολουθιακό Διάγραμμα Αλλαγής Μοντέλου Εκτίμησης Εμπιστοσύνης.....	62
Σχήμα 5.1-1 Παράδειγμα Γράφου Απαιτήσεων	66
Σχήμα 5.2-1 Διάγραμμα ροής αλγόριθμου TidalTrust	72
Σχήμα 5.2-2 Παράδειγμα: Βήμα 1	72
Σχήμα 5.2-3 Παράδειγμα: Βήμα 2	73
Σχήμα 5.2-4 Παράδειγμα: Βήμα 3	73
Σχήμα 5.2-5 Παράδειγμα: Βήμα 4	74
Σχήμα 5.2-6 Παράδειγμα: Βήμα 5	74
Σχήμα 5.2-7 Παράδειγμα: Βήμα 6	75
Σχήμα 5.2-8 Παράδειγμα: Βήμα 7	75
Σχήμα 5.2-9 Παράδειγμα: Βήμα 8	76
Σχήμα 5.2-10 Παράδειγμα: Βήμα 9	76
Σχήμα 5.2-11 Παράδειγμα: Βήμα 10	77
Σχήμα 5.2-12 Παράδειγμα: Βήμα 11	77
Σχήμα 6.1-1 Διάγραμμα Χρόνου ως προς το Μέγεθος του γράφου.....	80

Σχήμα 6.1-2Διάγραμμα Χρόνου ως προς την Πυκνότητα του γράφου	81
Σχήμα 6.1-3Διάγραμμα Χρόνου ως προς το Μήκος Μονοπατιού	81
Σχήμα 6.1-4Διάγραμμα Χρόνου ως προς των Αριθμό Κόμβων ανά Επίπεδο.....	82
Σχήμα 6.1-5Διάγραμμα Χρόνου ως προς τον Αριθμό των Συσσωρεύσεων.....	83

Κεφάλαιο 1: Εισαγωγή

Στην παρούσα εισαγωγή περιγράφεται αρχικά το κίνητρο πίσω από την δημιουργία ενός περιβάλλοντος-πλαισίου εκτίμησης εμπιστοσύνης, το οποίο με την συλλογή εξωτερικών πληροφοριών δύναται να εκτιμήσει αποδοτικά την εμπιστοσύνη μιας υπηρεσίας που προσφέρεται από έναν πάροχο σχετικά με συγκεκριμένο περιεχόμενο (κεφάλαιο 1.1). Ακολουθώς παρουσιάζονται τα θέματα που προκύπτουν κατά την διαδικασία του σχεδιασμού και της ανάπτυξης του περιβάλλοντος-πλαισίου (κεφάλαιο 1.2). Τέλος αναφέρεται η επιστημονική συμβολή της παρούσας εργασίας (κεφάλαιο 1.3) και παρουσιάζεται η δομή της (κεφάλαιο 1.4).

1.1 Κίνητρο

Στην σύγχρονη εποχή, οι ηλεκτρονικές υπηρεσίες καταλαμβάνουν όλο και μεγαλύτερο ποσοστό των παρεχόμενων υπηρεσιών, ανεξαρτήτως κλάδου. Οι προσωπικές, αλλά και οι επιχειρηματικές, μας δραστηριότητες εξαρτώνται όλο και περισσότερο από υπηρεσίες, που παραδοσιακά ήταν ανθρωποκεντρικές, τα τελευταία χρόνια, όμως, έχουν αντικατασταθεί από τις ισοδύναμες ηλεκτρονικές τους. Χαρακτηριστικά τέτοια παραδείγματα είναι οι τραπεζικές και δημόσιες υπηρεσίες, το εμπόριο κάθε είδους, η ιατρική και ο έλεγχος ασφαλείας εγκαταστάσεων. Η αντικατάσταση αυτή, προϊόντος του χρόνου, τείνει να γίνει καθολική σε κάποιες περιπτώσεις, με άμεσο αποτέλεσμα την δυσκολία εκτίμησης της εμπιστοσύνης στην υπηρεσία και τον πάροχό της.

Η δυσκολία εκτίμησης της εμπιστοσύνης υπηρεσιών οδηγεί αναπόφευκτα σε δύο διαφορετικά, αλλά εξίσου αρνητικά αποτελέσματα:

- a) Ο χρήστης εμπιστεύεται μια υπηρεσία για να καλύψει την ανάγκη του, η οποία δεν χρίζει εμπιστοσύνης. Το αποτέλεσμα είναι η ανάγκη αυτή να μην καλύπτεται και ο χρήστης να υπόκειται σε απώλεια, οικονομική ή χρονική.
- b) Ο χρήστης δεν εμπιστεύεται μια υπηρεσία για να καλύψει την ανάγκη του, παρ' όλο που χρίζει εμπιστοσύνης. Το αποτέλεσμα είναι ο χρήστης να καλύπτει την ανάγκη του αλλά ίσως όχι με τον αποδοτικότερο, χρονικά ή οικονομικά, τρόπο.

Εύκολα διαπιστώνεται, επομένως, η άμεση ανάγκη για την δημιουργία ενός συστήματος, το οποίο να είναι ικανό να εκτιμήσει αποδοτικά, χρησιμοποιώντας πολλαπλές πηγές, την εμπιστοσύνη που απολαμβάνει ο οποιοσδήποτε πάροχος, και συγκεκριμένα οι παρεχόμενες από αυτόν υπηρεσίες [23], [30].

Αυτή η ανάγκη, λοιπόν, αποτέλεσε το κίνητρο για την δημιουργία του περιβάλλοντος-πλαισίου εκτίμησης εμπιστοσύνης υπηρεσιών.

1.2 Περιγραφή Προβλήματος

Το πρόβλημα που καλείται να επιλύσει η παρούσα διπλωματική αφορά στον σχεδιασμό και υλοποίηση ενός περιβάλλοντος-πλαισίου, το οποίο θα παρέχει μια εκτίμηση για τον βαθμό εμπιστοσύνης μιας υπηρεσίας που παρέχεται από έναν πάροχο. Η εκτίμηση αυτή θα βασίζεται σε πληροφορίες που θα προέρχονται από πολλαπλές και διαφορετικές, κατά περίπτωση, εξωτερικές πηγές.

Αρχικά, η *εμπιστοσύνη* ορίζεται ως ο *βαθμός της προσδοκίας (degree of expectation)* ότι ο πόρος ή η υπηρεσία θα ικανοποιήσει τις προδιαγεγραμμένες λειτουργικές και μη λειτουργικές απαιτήσεις της.

Συγκεκριμένα, η εκτίμηση της συνολικής εμπιστοσύνης μιας υπηρεσίας στηρίζεται στην εκτίμηση επιμέρους χαρακτηριστικών της. Τα χαρακτηριστικά αυτά αφορούν θέματα όπως η ασφάλεια των ζητούμενων δεδομένων, η ταχύτητα της υπηρεσίας, το σωστό αποτέλεσμα κ.α. Ο χρήστης πρέπει να έχει την δυνατότητα να επιλέξει τα χαρακτηριστικά που τον ενδιαφέρουν, καθώς και τον βαθμό στον οποίο είναι σημαντικά για την εξαγωγή της τελικής εκτίμησης.

Τα επιμέρους χαρακτηριστικά εξαρτώνται με την σειρά τους από κάποιους παράγοντες. Παραδείγματα τέτοιων παραγόντων είναι η ασφάλεια του καναλιού επικοινωνίας με την υπηρεσία, ο εξοπλισμός που χρησιμοποιείται και η εμπειρία προηγούμενων χρηστών της υπηρεσίας. Κάθε, σχετικός με το χαρακτηριστικό, παράγοντας εκτιμάται, χρησιμοποιώντας πληροφορίες από εξωτερικές πηγές, και συνεισφέρει στο αποτέλεσμα.

Ξεκινώντας, λοιπόν, από φαινομενικά ασύνδετους παράγοντες, εκτιμάται η εμπιστοσύνη σε συγκεκριμένα χαρακτηριστικά του παρόχου και εν συνεχεία παρέχεται μια συνολική εκτίμηση για την εμπιστοσύνη σε μια παρεχόμενη υπηρεσία, με βάση, πάντα, τις ανάγκες του εκάστοτε χρήστη.

1.3 Συνεισφορά της διπλωματικής εργασίας

Η παρούσα διπλωματική ασχολείται με τον σχεδιασμό και την κατασκευή ενός περιβάλλοντος-πλαισίου, το οποίο εξετάζει μια υπηρεσία, με βάση ένα πλήθος χαρακτηριστικών ορισμένα από τον χρήστη, και εκτιμά την εμπιστοσύνη σε αυτή, εκφράζοντας το αποτέλεσμα σε μορφή πιθανότητας. Η συνεισφορά της διπλωματικής συνοψίζεται στα εξής σημεία:

- Τον ορισμό και την υλοποίηση μιας αρχιτεκτονικής για το περιβάλλον-πλαίσιο, που υλοποιεί το **αρχιτεκτονικό μόρφωμα μαυροπίνακα (blackboard architectural pattern)**, η οποία δίνει την δυνατότητα μοντελοποίησης των επιμέρους χαρακτηριστικών της υπηρεσίας και την χρήση αλγορίθμου για την εκτίμηση της συνολικής εμπιστοσύνης στην υπηρεσία και τον πάροχό της.
- Την χρήση των μοντέλων στόχων για την μοντελοποίηση των επιμέρους χαρακτηριστικών. Τα **μοντέλα στόχων** είναι δένδρα, η ρίζα των οποίων αντιπροσωπεύει το προς εκτίμηση χαρακτηριστικό και οι υπόλοιποι κόμβοι τους παράγοντες που το

επηρεάζουν. Οι κόμβοι συνδέονται μεταξύ τους με διαζεύξεις(OR) ή συζεύξεις(AND). Παρέχεται, επίσης, η δυνατότητα θετικής ή αρνητικής συνεισφοράς μεταξύ των κόμβων.

- Την χρήση αλγορίθμου, που χρησιμοποιείται κατά βάση στα κοινωνικά δίκτυα, για την περιγραφή των συσχετίσεων ανάμεσα στα επιμέρους χαρακτηριστικά μιας υπηρεσίας, με στόχο την μελλοντική επέκταση για υποστήριξη συσχετίσεων και ανάμεσα σε διαφορετικούς παρόχους και υπηρεσίες.

1.4 Οργάνωση κειμένου

Η δομή της διπλωματικής είναι η ακόλουθη:

- *Κεφάλαιο 1:* Παρουσιάζονται οι βασικές έννοιες που αναφέρονται στην παρούσα διπλωματική, περιγράφεται το πρόβλημα που αντιμετωπίζεται και αναφέρεται και ο τρόπος με τον οποίο προσεγγίζεται.
- *Κεφάλαιο 2:* Παρουσιάζεται το θεωρητικό υπόβαθρο που είναι απαραίτητο για την πλήρη κατανόηση της παρούσας διπλωματικής εργασίας. Επιπλέον, αναφέρονται όλες οι τεχνολογίες και σχετικές εργασίες, που χρησιμοποιήθηκαν κατά την διάρκεια εκπόνησης της εργασίας.
- *Κεφάλαιο 3:* Δίνεται λεπτομερώς η αρχιτεκτονική που προέκυψε από τον σχεδιασμό του περιβάλλοντος-πλαίσιου. Επιπροσθέτως, αναλύονται τα επιμέρους συστατικά της αρχιτεκτονικής, όσον αφορά την χρήση και την διεπαφή τους.
- *Κεφάλαιο 4:* Περιγράφεται αναλυτικά η διαδικασία, ορίζοντας τις εμπλεκόμενες οντότητες και δίνοντας το domain model και τα αντίστοιχα διαγράμματα δραστηριοτήτων και ακολουθιακά διαγράμματα.
- *Κεφάλαιο 5:* Παρουσιάζεται ο αλγόριθμος που χρησιμοποιείται για τον συμπερασμό της εκτίμησης της εμπιστοσύνης στην υπηρεσία, μέσα από τις τιμές που έχουν υπολογιστεί για τα επιμέρους χαρακτηριστικά.
- *Κεφάλαιο 6:* Περιέχονται ένα σύνολο περιπτώσεων χρήσης, με στόχο την αξιολόγηση της απόδοσης του περιβάλλοντος-πλαίσιου ανάλογα με την πολυπλοκότητα και το μέγεθος του γράφου που παράγεται από τον τελικό αλγόριθμο.
- *Κεφάλαιο 7:* Ανακεφαλαιώνονται τα βασικά στοιχεία της παρούσας διπλωματικής, με σκοπό την εξαγωγή συμπερασμάτων και ιδεών για μελλοντική ανάπτυξη του περιβάλλοντος-πλαίσιου.

Κεφάλαιο 2: Σχετικές Εργασίες

2.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί παρατίθενται οι τεχνολογίες που επιστρατεύτηκαν κατά τη δημιουργία του περιβάλλοντος-πλαισίου που περιγράφεται και αναλύεται στην παρούσα διπλωματική εργασία. Επιπροσθέτως, αναφέρεται το θεωρητικό υπόβαθρο που απαιτείται για την κατανόηση κάποιων μερών της εργασίας. Το υπόβαθρο, αυτό, βέβαια, είναι εκτενέστερο των αναγκών μας και ως εκ τούτου θα αναφερθούν μόνο οι βασικές έννοιες, που θεωρούνται απαραίτητες. Περαιτέρω πληροφορίες βρίσκονται στην βιβλιογραφία που περιέχεται στο κεφάλαιο 8.

2.2 Μοντελοποίηση συστημάτων λογισμικού

Η μοντελοκεντρική αρχιτεκτονική (Model-Driven Architecture – MDA) αποτελεί μια αρχιτεκτονική για σχεδίαση λογισμικού, η οποία βασίζεται σε κάποιες οδηγίες σχετικά με την δόμηση των απαιτήσεων και συγκεκριμένα στην αναπαράστασή τους με την μορφή μοντέλων. Είναι μια προσέγγιση ευρέως διαδεδομένη στα σύγχρονα συστήματα και βασίζεται σε κάποιες υπάρχουσες τεχνολογίες, όπως είναι η Unified Modeling Language (UML), το Meta Object Facility (MOF), το XML Metadata Interchange (XMI) κ.α.

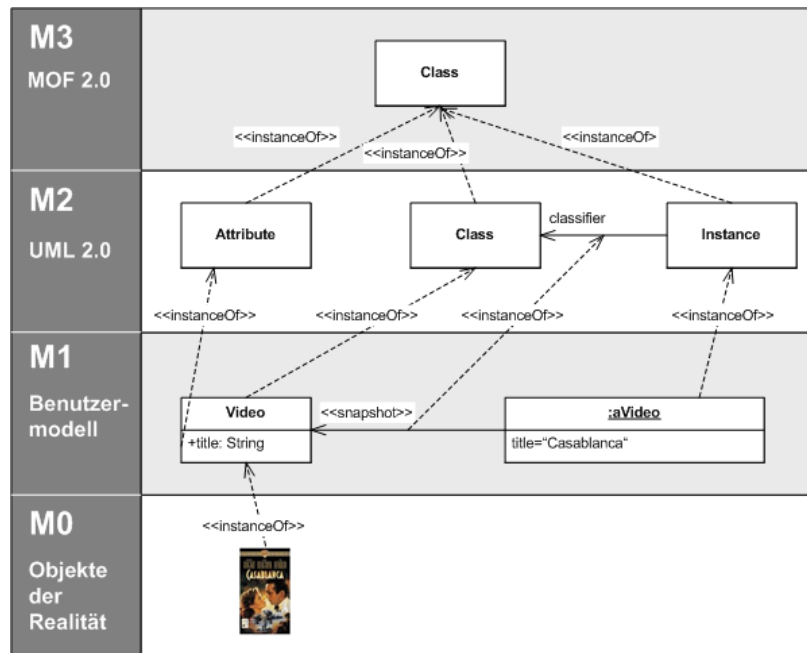
2.2.1 MOF

Το **Meta Object Facility(MOF)**[1][2] αποτελεί ένα πρότυπο του Object Management Group (OMG). Σκοπός της δημιουργίας του ήταν ο επίσημος ορισμός της UML και πλέον αποτελεί ένα από τα θεμέλια της οδηγούμενης από μοντέλα μηχανικής (model-driven engineering). Συγκεκριμένα, το πρότυπο ορίζει μια ιεραρχία μοντέλων τεσσάρων επιπέδων, και είναι προδιαγεγραμμένο με τέτοιο τρόπο ώστε να υποστηρίζεται απεριόριστη επεκτασιμότητα. Τα επίπεδα είναι τα εξής:

1. Το επίπεδο M3 ή επίπεδο μετα-μεταμοντέλου (meta-metamodel layer). Αποτελεί το ανώτερο επίπεδο της ιεραρχίας και η χρησιμότητά του έγκειται στην περιγραφή μοντέλων του επιπέδου M2, δηλαδή μεταμοντέλων (όπως π.χ. η UML). Τα μετα-μεταμοντέλα που βρίσκονται σε αυτό το επίπεδο στην ουσία είναι ορισμοί γλωσσών, που χρησιμοποιούνται στην προδιαγραφή μεταμοντέλων, και είναι MOF μοντέλα. Άξιο αναφοράς είναι το γεγονός ότι το επίπεδο αυτό είναι αυτοαναφορικό, πράγμα που σημαίνει ότι τα μετα-μεταμοντέλα είναι ορισμένα από μετα-μεταμοντέλα.
2. Το επίπεδο M2 ή επίπεδο μεταμοντέλου (metamodel layer). Στο επίπεδο αυτό βρίσκονται τα μεταμοντέλα, δηλαδή οι γλώσσες μοντελοποίησης, τα οποία χρησιμοποιούνται για την

περιγραφή των μοντέλων του επόμενου επιπέδου. Χαρακτηριστικό τέτοιο παράδειγμα αποτελεί η UML.

3. Το επίπεδο M1 ή επίπεδο μοντέλου (model layer). Στο επίπεδο αυτό βρίσκονται τα μοντέλα περιγραφής των μετα-δεδομένων που βρίσκονται στο επόμενο επίπεδο, δηλαδή στο M0.
4. Το επίπεδο M0 ή επίπεδο πληροφορίας (information layer). Στο επίπεδο αυτό αναπαρίσταται κάθε δεδομένο ή αντικείμενο που επιθυμούμε να μοντελοποιήσουμε. Τα αντικείμενα αυτά αποτελούν στιγμιότυπα μοντέλων του προηγούμενου επιπέδου.



Σχήμα 2.2-1 MOF Levels

Αξιοσημείωτος είναι ο ορισμός στο επίπεδο M2 του μεταμοντέλου που περιγράφει την UML. Επιπλέον, το MOF, πέρα από την ευκολία σχεδιασμού που παρέχει, επιτρέπει την αποθήκευση των μοντέλων με πρότυπα όπως το XMI, με πλεονεκτήματα όπως η μεταφορά σε άλλη εφαρμογή και η επέκταση του μεταγενέστερα, πιθανόν και από διαφορετικό σχεδιαστή.

2.2.2 XMI

Το **XML Metadata Interchange (XMI)** είναι ένα πρότυπο του OMG, που δημιουργήθηκε και πιστοποιήθηκε για ανταλλαγή μετα-δεδομένων μέσω της eXtensible Markup Language (XML). Το XMI είναι χρησιμοποιήσιμο για κάθε μετα-δεδομένο, του οποίου το μεταμοντέλο μπορεί να περιγραφεί σε MOF. Παρ' ότι η ικανότητα του προτύπου περιλαμβάνουν ανταλλαγή μοντέλων σε διάφορες γλώσσες, αλλά και τις ίδιες τις γλώσσες περιγραφής, η συνηθέστερη χρήση του είναι η μεταφορά UML μοντέλων.

Το OMG ορίζει δύο διακριτά είδη μοντέλων, τα αφηρημένα (abstract models) και τα συγκεκριμένα (concrete models). Τα αφηρημένα αφορούν μοντέλα που περιγράφουν την σημασιολογία της πληροφορίας, ενώ τα συγκεκριμένα είναι μοντέλα αναπαράστασης εποπτικών διαγραμμάτων. Συνηθισμένα παραδείγματα αφηρημένων μοντέλων είναι οι γλώσσες μοντελοποίησης, όπως η UML και η SysML [3], που βασίζονται στο πρότυπο MOF.

Το XMI δημιουργήθηκε προκειμένου να διευκολύνει δύο βασικές λειτουργίες. Η μια είναι η εύκολη ανταλλαγή μετα-δεδομένων ανάμεσα σε εργαλεία μοντελοποίησης που βασίζονται στην UML και αποθήκες μετα-δεδομένων, όταν αυτά βρίσκονται σε ετερογενή υπολογιστικά συστήματα. Η άλλη αφορά την δημιουργία μοντέλων που θα μπορούν να εισαχθούν σε εργαλεία αυτόματης παραγωγής κώδικα. Η ασυμβατότητα μεταξύ των εργαλείων μοντελοποίησης, ακόμα και σε περιπτώσεις αφηρημένων μοντέλων, όμως, δυσχεραίνει την εκτενή χρήση του XMI.

2.2.3 Eclipse Modeling Framework

Το **Eclipse Modeling Framework (EMF)**[4] είναι ένα έργο ανοιχτού κώδικα που χρηματοδοτείται και βρίσκεται υπό την επίβλεψη της IBM. Αποτελεί μια προσθήκη στο γραφικό περιβάλλον ανάπτυξης Eclipse και χρησιμοποιείται για την δημιουργία συστημάτων λογισμικού που υπακούν στον ορισμό της Μοντελοκεντρικής Αρχιτεκτονικής. Το EMF είναι, την στιγμή συγγραφής της παρούσας διπλωματικής, ίσως η καλύτερη προσέγγιση σε αυτή την πρακτική.

Το EMF επιτρέπει στον σχεδιαστή την δημιουργία μοντέλων για την περιγραφή ενός συστήματος λογισμικού. Λαμβάνοντας υπόψιν την γνώση ότι ο πηγαίος αντικειμενοστραφής κώδικας, τα διαγράμματα UML και τα XML σχήματα αποτελούν όλα μοντέλα περιγραφής ενός συστήματος, γίνεται μια προσπάθεια ενοποίησης των διαφορετικών τεχνικών μοντελοποίησης. Αυτή η προσπάθεια δίνει την δυνατότητα να περιγράφει το σύστημα με όποια τεχνική επιθυμεί και να έχει διαθέσιμη την περιγραφή και των υπόλοιπων μοντέλων.

Το EMF υποστηρίζει δυο ευρέως διαδεδομένα πρότυπα του Object Management Group, το MOF και το XMI. Με το EMF, λοιπόν, μπορεί να περιγραφεί ένα δομημένο και ιεραρχικό MOF μοντέλο. Αυτό το μοντέλο αποτελεί από μόνο του μια προδιαγραφή ενός συστήματος λογισμικού ή/και των δεδομένων που χρησιμοποιεί το σύστημα αυτό. Επιπλέον, το EMF μπορεί να παράγει αυτόματα κώδικα είτε από μοντέλα που δημιουργήθηκαν από αυτό ή από μοντέλα που εισήχθησαν από άλλη πηγή σε μορφή που συνάδει με το XMI πρότυπο.

Το πρότυπο XMI χρησιμοποιείται ευρέως από το EMF και για την ανταλλαγή μοντέλων με άλλες εφαρμογές, όπως αναφέρθηκε προηγουμένως, αλλά και για την αποθήκευση και αρχικοποίηση μοντέλων στον κώδικα της εφαρμογής. Όλες οι λειτουργίες του EMF είναι διαθέσιμες μέσω του API που προσφέρει για την καλύτερη και ευκολότερη χρήση των δυνατοτήτων του.

Στα πλαίσια της παρούσας διπλωματικής, το EMF χρησιμοποιήθηκε για την μοντελοποίηση ολόκληρου του συστήματος που αποτελεί το περιβάλλον-πλαίσιο.

2.3 Δέντρα Στόχων (Goal Trees)

Η συλλογή και η ανάλυση των απαιτήσεων ενός συστήματος λογισμικού είναι ένα από τα βασικότερα και σημαντικότερα στάδια της διαδικασίας ανάπτυξης λογισμικού. Ενδεχόμενη αποτυχία ορισμού σαφών, πλήρων και μη ελλειπών απαιτήσεων του συστήματος οδηγεί σε κατακόρυφη μείωση της ποιότητας και πολύ μεγάλη αύξηση του κόστους του λογισμικού που παράγεται. Για τον λόγο αυτό, το στάδιο αυτό είναι εξαιρετικά σημαντικό για την όλη διαδικασία. Με το στάδιο αυτό, λοιπόν, ασχολείται ένας ξεχωριστός κλάδος της Τεχνολογίας Λογισμικού που ονομάζεται Μηχανική Απαιτήσεων.

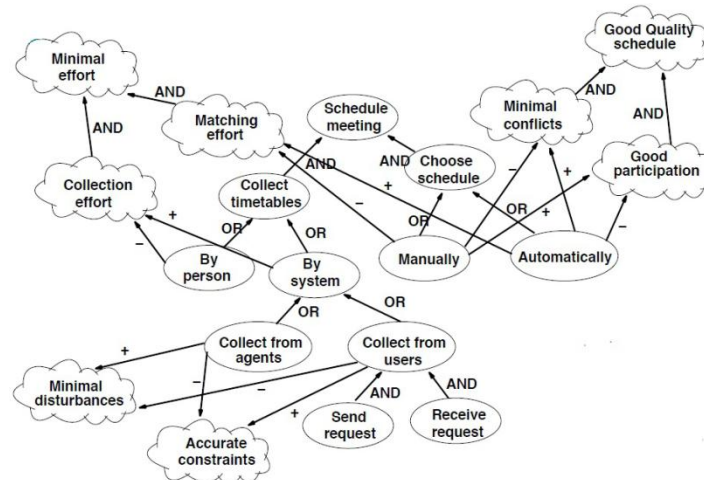
Η προσανατολισμένη σε στόχους μηχανική απαιτήσεων αποτελεί μια δημοφιλή προσέγγιση στην Μηχανική Απαιτήσεων. Τα πλεονεκτήματα της προσέγγισης αυτής αναλύονται διεξοδικά στα [5] και [6]. Άμεση απόδειξη της αποδοχής της συγκεκριμένης προσέγγισης αποτελεί η ύπαρξη πολλαπλών μεθοδολογιών που έχουν προταθεί κατά καιρούς για την προσανατολισμένη σε στόχους Μηχανική Απαιτήσεων, π.χ τα KAOS [7], GRL [8], GBRAM [9], TROPOS [10], CREWS [11], i* [12] κτλ.

Η μοντελοποίηση των απαιτήσεων, λειτουργικών και μη, είναι από τα σημαντικότερα κομμάτια της Μηχανικής Απαιτήσεων. Λόγω της σημαντικότητας αυτής, λοιπόν, έχουν προταθεί αρκετές μέθοδοι για την μοντελοποίησή τους και έχει δημιουργηθεί και αντίστοιχος, αν όχι μεγαλύτερος, αριθμός εργαλείων που στηρίζονται σε αυτές τις μεθόδους. Το [13] περιγράφει εκτενέστερα την εισαγωγή τους στον χώρο της μηχανικής και μοντελοποίησης λογισμικού.

Τα δέντρα στόχων είναι γράφοι, ή πιο συγκεκριμένα δέντρα, όπως μαρτυρά και το όνομα, των οποίων οι κόμβοι αποτελούν στόχους (goals). Κάθε στόχος, όμως, μπορεί να αποτελεί παιδί κάποιου άλλου στόχου στο δέντρο, ή υποστόχο (subgoal) όπως ονομάζεται. Οι υποστόχοι μπορεί να συνθέτουν κάποιον στόχο, μέσω λογικής σύζευξης (KAI/AND) ή διάζευξης (H/OR), και η επαλήθευσή τους ή όχι επηρεάζει την επαλήθευση ή όχι του στόχου με διακριτό τρόπο ανάλογα με την περίπτωση. Συγκεκριμένα, στην περίπτωση της σύζευξης απαιτείται η ικανοποίηση όλων των υποστόχων-παιδιά προκειμένου να ικανοποιηθεί ο στόχος-πατέρας. Αντίστοιχα, στην περίπτωση της διάζευξης η ικανοποίηση ενός τουλάχιστον υποστόχου-παιδί εγγυάται την ικανοποίηση του στόχου-πατέρα.

Οι παραπάνω στόχοι ονομάζονται στην βιβλιογραφία, όπως στο [13], **hard goals** και αναφέρονται σε λειτουργικές απαιτήσεις. Ορίζεται έτσι μια μονοσήμαντη σημασιολογία επαλήθευσης ανάμεσα στον στόχο και τους υποστόχους του. Στην πραγματικότητα, όμως, κατά την διάρκεια της μοντελοποίησης απαιτήσεων λογισμικού προκύπτουν περιπτώσεις που η

μονοσήμαντη επαλήθευση δεν αρκεί. Κάποιες φορές ένας στόχος επαληθεύεται ή όχι ανάλογα με το ποσοστό θετικής σε σχέση με την αρνητική συνεισφορά που λαμβάνει από τους υποστόχους του. Αυτοί είναι οι γνωστοί ως **soft goals**. Παρακάτω παρατίθεται ένα παράδειγμα ενός δέντρου που αποτελείται από συνδυασμό hard και soft goals.



Σχήμα 2.3-1 Παράδειγμα δέντρου με hard και soft goals

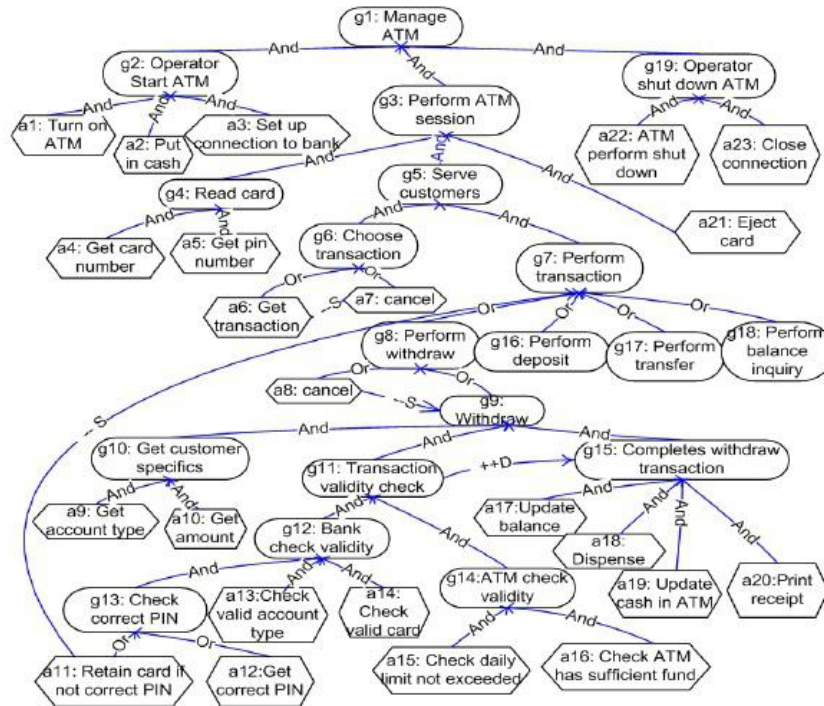
Στο διάγραμμα αυτό διακρίνεται η δενδρική δομή που αποτελείται από hard goals (οβάλ), soft goals (σύννεφα) και συνεισφορές θετικού ή αρνητικού προσήμου.

Για τα hard goals υπάρχει και μια επιπλέον δυνατότητα συνεισφοράς που αποτελείται από τέσσερις περιπτώσεις. Οι περιπτώσεις αυτές είναι οι ακόλουθες:

- ++S (a, b) : επιτυχία του a συνεπάγεται επιτυχία του b
- --S (a, b) : επιτυχία του a συνεπάγεται αποτυχία του b
- ++D (a, b) : αποτυχία του a συνεπάγεται επιτυχία του b
- --D (a, b) : αποτυχία του a συνεπάγεται αποτυχία του b

Επιπρόσθετες επεκτάσεις των μοντέλων δέντρων που αφορούν πράκτορες (Agents), ρόλους (Roles), δεσμεύσεις (Commitments) μεταξύ πρακτόρων και συνεισφορές (Contributions) μεταξύ κόμβων μπορούν να βρεθούν στα [14], [15], [16].

Ένα πλήρες παράδειγμα έχει παρθεί από το [17] και περιλαμβάνει όλα τα είδη συνεισφορών που περιγράψαμε παραπάνω:



Σχήμα 2.3-2 Παράδειγμα δέντρου στόχων με soft goals και contributions

Στην παρούσα διπλωματική χρησιμοποιήσαμε τα δέντρα στόχων για να μοντελοποιήσουμε τα επιμέρους χαρακτηριστικά της υπηρεσίας, που ενδιαφέρουν τον χρήστη. Κάθε χαρακτηριστικό βρίσκεται στην ρίζα του δέντρου και οι παράγοντες που το επηρεάζουν δομούνται σαν υποστόχοι, χρησιμοποιώντας κατά περίπτωση σύζευξη ή διάζευξη μεταξύ τους, καθώς επίσης και συνεισφορές. Προκύπτει τελικά ένα αποτέλεσμα που είναι ενδεικτικό του αν το χαρακτηριστικό αυτό ικανοποιείται ή όχι και σε τι βαθμό.

2.4 Περιβάλλοντα συλλογιστικής

Τα τελευταία χρόνια, η ανάπτυξη της τεχνητής νοημοσύνης στην επιστήμη των υπολογιστών είναι ραγδαία, με αποτέλεσμα την άμεση συνεισφορά της και σε άλλους τομείς διάφορων επιστημονικών κλάδων. Σκοπός, λοιπόν, της τεχνητής νοημοσύνης είναι η δημιουργία μαθηματικά τεκμηριωμένων μεθοδολογιών και εργαλείων, που θα επιτρέπουν την επίλυση προβλημάτων λογικής, σημαντικά δυσκολότερων από εκείνα που ο ανθρώπινος εγκέφαλος μπορεί να διαχειριστεί. Η έρευνα που έχει γίνει πάνω στον τομέα αυτό έχει παράγει αξιόλογα αποτελέσματα. Κάποια τέτοια παραδείγματα αποτελούν η Λογική Πρώτης Ταξης και τα Λογικά Μαρκοβιανά Δίκτυα. Αν αναλογιστούμε και την ευκολία μετατροπής των δέντρων στόχων σε προτάσεις λογικής πρώτης τάξης, γίνεται εμφανής η ανάγκη για μελέτη τέτοιων εργαλείων στο πλαίσιο της παρούσας διπλωματικής. Περιγράφουμε, λοιπόν, σε αυτή την ενότητα την Λογική

Πρώτης Τάξης και τα Λογικά Μαρκοβιανά Δίκτυα, που στηρίζονται πάνω στην Λογική Πρώτης Τάξης.

2.4.1 Λογική Πρώτης Τάξης

Η Λογική Πρώτης Τάξης (First Order Logic) αποτελεί έναν φορμαλιστικό τρόπο αναπαράστασης γνώσης και εκτέλεσης συμπερασμών, ο οποίος προσπαθεί να προσεγγίσει τον ανθρώπινο λογικό τρόπο σκέψης [18]. Επιπλέον, η Λογική Πρώτης Τάξης θεμελιώνεται πάνω στην Μαθηματική Λογική, και την Προτασιακή Λογική κατ' επέκταση, και λειτουργεί, κατά βάση, ορίζοντας έναν κόσμο, στον οποίο ισχύουν κάποιοι συγκεκριμένοι κανόνες. Στην συνέχεια, μπορούν να χρησιμοποιηθούν τεχνικές συμπερασμού (reasoning) και να απαντηθούν ερωτήματα που αφορούν τον κόσμο αυτό.

Η Λογική Πρώτης Τάξης επεκτείνει την Προτασιακή Λογική εισάγοντας κάποια νέα στοιχεία. Τα στοιχεία αυτά είναι οι όροι (terms), τα κατηγορήματα (predicates) και οι ποσοδείκτες (quantifiers). Υιοθετείται, επιπροσθέτως, και το ακόλουθο αλφάβητο, προκειμένου να υποστηριχθεί η επέκταση αυτή:

- Σταθερές: a, b, c κ.λ.π
- Μεταβλητές: x, y, z κ.λ.π
- Σύμβολα συναρτήσεων: f, g κ.λ.π. Οι συναρτήσεις δέχονται ένα πλήθος ορισμάτων, που τελικά ορίζει και την τάξη της συνάρτησης.
- Σύμβολα κατηγορημάτων: P, Q κ.λ.π. Τα κατηγορήματα χρησιμοποιούνται για την τυποποίηση σχέσεων που ισχύουν στον κόσμο. Δέχονται ένα πλήθος ορισμάτων, που ορίζει και την τάξη του κατηγορήματος.
- Σύνδεσμοι: $\neg$ (άρνηση), $\wedge$ (σύζευξη), $\vee$ (διάζευξη), $\rightarrow$ (συνεπαγωγή), $\leftrightarrow$ (ισοδυναμία).
- Ποσοδείκτες: $\exists$ (υπαρξιακός) και $\forall$ (καθολικός)
- Σημεία στίξης: « (» , «) » , « , »

Οι όροι (terms), οι οποίοι αποτελούν νέο στοιχείο της Λογικής Πρώτης Τάξης, ορίζονται ως εξής και μόνον με αυτούς τους τρόπους:

- Μια σταθερά είναι όρος.
- Μια μεταβλητή είναι όρος.
- Έστω μια f συνάρτηση n -τάξης και $t_1, \dots, t_n$ n το πλήθος όροι. Τότε η $f(t_1, \dots, t_n)$ είναι όρος.

Επιπλέον, ονομάζονται καλά σχηματισμένοι τύποι, ή απλά τύποι, οι «οντότητες» που κατασκευάζονται με τους ακόλουθους τρόπους, και μόνον με αυτούς:

- Εάν P είναι k -τάξης κατηγορημα και $t_1, \dots, t_k$ k στο πλήθος όροι, τότε $P(t_1, \dots, t_k)$ είναι ατομικός τύπος (atomic formula) ή απλά άτομο (atom) ή θετικό λεκτικό στοιχείο (positive literal).
- Εάν ϕ είναι τύπος, τότε και $\neg\phi$ είναι τύπος.

- Εάν ϕ, ψ είναι τύποι, τότε και $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, $(\phi \leftrightarrow \psi)$ είναι τύποι.
- Εάν ϕ είναι τύπος και x μεταβλητή, τότε και $(\exists x)\phi$, $(\forall x)\phi$ είναι τύποι.

Τέλος, βασικός όρος (ground term), ή τύπος ή άτομο, είναι ένας όρος, ή τύπος ή άτομο, που δεν περιέχει καμία μεταβλητή στον ορισμό του. Επί παραδείγματι, ο όρος $g(a,e)$ και ο τύπος $P(b,h(c,d))$ είναι βασικοί, ενώ ο όρος $g(a,x)$ και ο τύπος $P(x,h(c,y))$ όχι.

Η Λογική Πρώτης Τάξης χρησιμοποιεί δύο βασικές έννοιες σαν εργαλεία για τους αλγορίθμους συμπερασμού γνώσης: την αντικατάσταση (substitution) και την ενοποίηση (unification).

Η έννοια της αντικατάστασης ορίζεται ως η διαδικασία αντικατάστασης κάποιας μεταβλητής που εμφανίζεται σε έναν τύπο με τον κατάλληλο όρο. Αυτή η διαδικασία συμβολίζεται με $\{X_i/t_i\}$, όπου X_i είναι η μεταβλητή προς αντικατάσταση και t_i ο όρος που θα την αντικαταστήσει. Για παράδειγμα η αντικατάσταση $\{X/\text{Smith}\}$ στον τύπο:

Supervisor(X, Williams)

Θα μας δώσει τον τύπο:

Supervisor(Smith, Williams)

Η έννοια της ενοποίησης ορίζεται ως η διαδικασία κατά την οποία, με την χρήση κατάλληλων αντικαταστάσεων, δύο τύποι γίνονται συντακτικά όμοιοι. Για παράδειγμα, οι ακόλουθοι τύποι:

Supervisor(X, Williams)

Supervisor(Smith, Y)

ενοποιούνται με την χρήση των αντικαταστάσεων $\{X, \text{Smith}\}$, $\{Y, \text{Williams}\}$.

2.4.2 Μαρκοβιανά Λογικά Δίκτυα (MLN)

Η θεωρία των Μαρκοβιανών Λογικών Δικτύων αναπτύχθηκε σαν μια προσπάθεια συνδυασμού των θετικών στοιχείων της Λογικής Πρώτης Τάξης και των πιθανοτικών γραφικών μοντέλων, ή πιο συγκεκριμένα των Μαρκοβιανών Δικτύων. Την αρχική εισαγωγή έκαναν οι Richardson και Domingos[19],[20]. Ένα MLN, λοιπόν, αποτελείται από μια βάση γνώσης που προσομοιάζει σε μια αντίστοιχη βάση λογικής πρώτης τάξης. Στην βάση αυτή δηλαδή περιέχονται κατηγορήματα και θεμελιώδη άτομα, που κι αυτά με την σειρά τους συνδέονται με λογικούς συνδέσμους και ποσοδείκτες, προκειμένου να σχηματιστούν περαιτέρω τύποι που ανήκουν στην βάση. Η βασική διαφορά, όμως, προκύπτει από τον τρόπο χειρισμού της βάσης αυτής. Όπως συζητήθηκε και στην προηγούμενη ενότητα, ανάθεση αληθοτιμών στα κατηγορήματα δημιουργεί έναν κόσμο στον οποίο η βάση γνώσης ανταποκρίνεται ή όχι. Αδυναμία επιβεβαίωσης έστω και ενός λογικού κανόνα της βάσης ακυρώνει, με βάση την Λογική Πρώτης Τάξης, την δυνατότητα ερμηνείας της βάσης με χρήση του δημιουργηθέντος κόσμου. Εν αντιθέσει, στα Μαρκοβιανά Λογικά Δίκτυα κάτι τέτοιο δεν ισχύει. Η αδυναμία επιβεβαίωσης ενός λογικού κανόνα απλώς μειώνει την

πιθανότητα ο κόσμος, που έχει δημιουργηθεί, να αποτελεί ερμηνεία της βάσης γνώσης. Καταλαβαίνουμε, λοιπόν, ότι η πιθανότητα ο κόσμος να ορίζει μια ερμηνεία της βάσης, είναι αντιστρόφως ανάλογη του πλήθους κανόνων που παραβιάζονται και των περιορισμών που θέτουν αυτές οι παραβιάσεις[21].

Ένα Μαρκοβιανό Δίκτυο ορίζεται ως ένα μοντέλο της από κοινού κατανομής ενός συνόλου μεταβλητών $X=(X_1, X_2, \dots, X_n)$, όπως έχει αναλυθεί στο [19]. Το μοντέλο αυτό αποτελείται από έναν κατευθυνόμενο γράφο G και μια σειρά από συναρτήσεις δυναμικού (potential functions) ϕ_k . Ένα προς ένα αντιστοιχία έχουν οι κόμβοι του γράφου με τις μεταβλητές του συνόλου X , και οι κλίκες του γράφου με τις συναρτήσεις. Οι συναρτήσεις δυναμικού αναπαριστούν την κατάσταση της αντίστοιχης σε αυτές κλίκας, και ορίζονται με πεδίο τιμών τους μη αρνητικούς πραγματικούς αριθμούς. Η από κοινού κατανομή πιθανότητας που προκύπτει από τα ανωτέρω είναι η ακόλουθη:

$$P(x) = \frac{1}{Z} \prod \phi_k(x_k) \quad (2.4.1)$$

όπου x_k αντιστοιχεί στην κατάσταση των μεταβλητών που εμφανίζονται στην κλίκα k , και Z στην συνάρτηση διαμέρισης που δίνεται από την ακόλουθη σχέση:

$$Z = \sum_x \prod_k \phi_k(x_k) \quad (2.4.2)$$

Τα Μαρκοβιανά Δίκτυα συχνά αναπαριστώνται ως λογαριθμογραμμικά μοντέλα (log – linear models). Εν τοιαύτη περιπτώσει, έχουμε αντικατάσταση των συναρτήσεων δυναμικού με εκθετικά αθροίσματα βαρών των χαρακτηριστικών της κατάστασης του δικτύου. Κάθε χαρακτηριστικό αποτελεί συνάρτηση της κατάστασης με πραγματικό πεδίο ορισμού. Τελικά, η από κοινού κατανομή διαμρφώνεται ως εξής:

$$P(x) = \frac{1}{Z} \exp \left(\sum_i w_i f_i(x) \right) \quad (2.4.3)$$

όπου w_i αντιπροσωπεύουν τα βάρη και f_i τα χαρακτηριστικά. Θεωρούμε, επίσης, ότι τα χαρακτηριστικά έχουν σύνολο τιμών το $\{0, 1\}$.

Η αναπαράσταση γνώσης με την χρήση Λογικής Πρώτης Τάξης είναι, όπως είδαμε και στην προηγούμενη ενότητα, εξαιρετικά εύκολη. Προβλήματα προκύπτουν, όμως, από την αδυναμία υποστήριξης ελαστικότητας στις περιπτώσεις που ένας ή περισσότεροι κανόνες της βάσης δεν ικανοποιούνται. Σε αυτές τις περιπτώσεις, ο κόσμος που δημιουργείται ακυρώνεται και δεν θεωρείται, πλέον, πιθανή ερμηνεία της βάσης γνώσης. Συνδυασμός, ωστόσο, με την ευελιξία των Μαρκοβιανών Δικτύων θα μπορούσε να δώσει έναν κόσμο με συγκεκριμένη πιθανότητα εγκυρότητας δοθείσας μιας βάσης. Τα Μαρκοβιανά Λογικά Δίκτυα δημιουργήθηκαν με ακριβώς

αυτή την λογική συνδυασμού των δύο θεωριών. Δημιουργείται έτσι μια αντίστροφη αναλογία ανάμεσα στην πιθανότητα ο κόσμος να είναι έγκυρος και στο πλήθος κανόνων που παραβιάζονται. Επιπλέον, κάθε κανόνας, της βάσης γνώσης της Λογικής Πρώτης Τάξης, συνοδεύεται από ένα βάρος, αντιπροσωπευτικό της σημαντικότητας του κανόνα για το Μαρκοβιανό Δίκτυο ή με άλλα λόγια της επιρροής του κανόνα στην πιθανότητα εγκυρότητας του κόσμου.

Αυτές οι πληροφορίες οδηγούν στην δημιουργία του ακόλουθου τυπικότερου ορισμού των Μαρκοβιανών Λογικών Δικτύων. Έστω ένα σύνολο $L = \{(F_i, w_i)\}$, όπου F_i αντιστοιχεί σε έναν τύπο Λογικής Πρώτης Τάξης και w_i στο βάρος του συγκεκριμένου τύπου, εκφρασμένο σε μορφή πραγματικού αριθμού. Το σύνολο L , σε συνδυασμό με ένα πεπερασμένο σύνολο σταθερών $C = \{C_1, C_2, \dots, C_{|C|}\}$, ορίζουν ένα Μαρκοβιανό Λογικό Δίκτυο $M_{L,C}$ ως εξής:

1. Το $M_{L,C}$ περιέχει έναν δυαδικό κόμβο για κάθε πιθανή θεμελίωση (grounding) κάθε κατηγορήματος που εμφανίζεται στο L . Η τιμή του εκάστοτε κόμβου είναι 1 αν το βασικό άτομο είναι αληθές, και 0 αν είναι ψευδές.
2. Το $M_{L,C}$ περιέχει ένα χαρακτηριστικό για κάθε πιθανή θεμελίωση κάθε πιθανού τύπου F_i που εμφανίζεται στο L . Το εκάστοτε χαρακτηριστικό έχει τιμή 1 αν ο ατομικός τύπος είναι αληθής, και 0 αν είναι ψευδής. Το βάρος του χαρακτηριστικού είναι το w_i που σχετίζεται με το F_i στο L .

Διαφορετικά σύνολα σταθερών μπορεί να οδηγήσουν ένα Μαρκοβιανό Λογικό Δίκτυο σε παραγωγή διαφορετικών δικτύων. Τα δίκτυα αυτά ονομάζονται βασικά Μαρκοβιανά Δίκτυα (ground Markov Networks), και, παρ' όλο που μπορεί να διαφέρουν αρκετά σε έκταση, παρουσιάζουν ομοιότητες στην δομή και τις παραμέτρους τους. Λογικά, λοιπόν, προκύπτει ότι η συνάρτηση κατανομής πιθανότητας όλων των x , που ορίζονται από το βασικό μαρκοβιανό δίκτυο, έχει την ακόλουθη μορφή:

$$P(X = x) = \frac{1}{Z} \exp \left(\sum_i w_i n_i(x) \right) = \frac{1}{Z} \left(\prod_i \varphi_i(x_i)^{n_i(x)} \right) \quad (2.4.4)$$

Όπου $n_i(x)$ είναι ο αριθμός των αληθών θεμελιώσεων του F_i στο x , x_i είναι η κατάσταση των ατόμων που εμφανίζονται στο F_i και $\varphi_i(x_i) = e^{w_i}$. Σύμφωνα με την θεωρία που έχει αναπτυχθεί για τα Μαρκοβιανά Λογικά Δίκτυα, οι αλγοριθμικές διαδικασίες που μπορούν να εφαρμοστούν σε ένα δίκτυο είναι οι εξής δύο:

- **Μάθηση (Learning):** Η διαδικασία αυτή μας επιτρέπει να εξάγουμε τα βάρη του Δικτύου, χρησιμοποιώντας μια ή περισσότερες γνωσιακές βάσεις. Αυτό γίνεται κάνοντας μια θεώρηση κλειστού κόσμου, πράγμα που σημαίνει ότι κάθε βασικό άτομο που δεν συναντάται στην βάση θεωρείται ψευδές.
- **Συμπερασμός (Inference):** Η διαδικασία αυτή επιτρέπει την εκτίμηση της πιθανότητας να ισχύει ή όχι ένας ατομικός τύπος F_i , δοθείσας της ισχύος κάποιου άλλου τύπου F_j .

Πρακτική εφαρμογή των αλγορίθμων στατιστικής μάθησης και συμπερασμού, που βασίζονται στην θεωρία των Μαρκοβιανών Λογικών Δικτύων, κάνουν διάφορα εργαλεία. Παράδειγμα τέτοιου εργαλείου είναι το Alchemy[22].

2.5 Συμπερασμός εμπιστοσύνης

Η μεγάλη ανάπτυξη των ηλεκτρονικών υπηρεσιών [23] τα τελευταία χρόνια δημιούργησε αναπόφευκτα την ανάγκη για αποδοτικούς τρόπους ανάλυσής τους και διαπίστωσης της εμπιστοσύνης τους. Η έλλειψη φυσικής παρουσίας, λοιπόν, έχει οδηγήσει στην ανάπτυξη διαφόρων τεχνικών και αλγορίθμων για να υποστηριχθεί αυτή η διαδικασία αξιολόγησης των υπηρεσιών, ή μάλλον, για να είμαστε πιο ακριβείς, των χρηστών που παρέχουν την εκάστοτε υπηρεσία.

Η εμπιστοσύνη, λοιπόν, μπορεί να αφορά πολλά και διαφορετικά περιεχόμενα. Στην παρούσα διπλωματική, όμως, με τον όρο εμπιστοσύνη εννοούμε την πεποίθηση, που αποκτά το σύστημα μέσω της διαδικασίας αξιολόγησης, ότι η εμπιστευόμενη πλευρά μπορεί να φέρει εις πέρας την υπηρεσία την οποία έχει υποσχεθεί στον χρήστη που την χρειάζεται. Αυτή είναι η πιο δημοφιλής ερμηνεία και χρησιμοποιείται στις περισσότερες ερευνητικές προσπάθειες, όπως π.χ στο [24], [25], [26]

Το μεγαλύτερο κομμάτι της έρευνας, πάνω στο συγκεκριμένο θέμα, αφορά τον συμπερασμό εμπιστοσύνης σε κοινωνικά δίκτυα. Συγκεκριμένα, διάφοροι αλγόριθμοι έχουν προταθεί κατά καιρούς με τον καθένα να προσθέτει επιπλέον χαρακτηριστικά ή να επιλέγει διαφορετικό τρόπο εκτίμησης του τελικού αποτελέσματος. Παραδείγματα τέτοιων αλγορίθμων αναφέρονται στα [24], [25], [26], [27] και είναι ο EigenTrust, Advogato, TidalTrust, Sunny, MoleTrust, ModelTrust, TrustWalker, AppleSeed κ.α.

Κοινό σημείο, παρ' όλα αυτά, αποτελεί, λόγω της φύσης του προβλήματος, η αναπαράσταση των χρηστών σε έναν «κοινωνικό» γράφο. Κάθε χρήστης αποτελεί έναν κόμβο και οι συνδέσεις μεταξύ τους αντιπροσωπεύουν την εμπιστοσύνη που μοιράζονται εκφρασμένη σε μορφή ποσοστού. Η διαφορά μεταξύ των αλγορίθμων παρουσιάζεται βασικά στον τρόπο εκτίμησης της εμπιστοσύνης μεταξύ χρηστών που δεν συνδέονται άμεσα αλλά μέσω άλλων χρηστών.

Ο Advogato εκφράζει την εμπιστοσύνη που καθολικά απολαμβάνει ο χρήστης, υπολογίζοντας την ροή του γράφου, αγνοώντας ενδεχόμενα βάρη στις ακμές. Ο AppleSeed λαμβάνει υπ' όψιν τα βάρη αυτά και άρα υπολογίζει την συνολική ενέργεια που φτάνει στον εκάστοτε χρήστη μέσω διάφορων μονοπατιών. Ο TidalTrust, που προτάθηκε και αναλύεται στο [28], από την άλλη υπολογίζει ελάχιστα μονοπάτια με μέγιστο βαθμό εμπιστοσύνης. Διαφοροποιήσεις έχουν προταθεί όσον αφορά τον τρόπο διάσχισης του γράφου (MoleTrust) και την ύπαρξη διακριτών καταστάσεων στους κόμβους (Sunny).

Στην παρούσα διπλωματική ο πλέον κατάλληλος αλγόριθμος είναι ο TidalTrust λόγω των χαρακτηριστικών που υλοποιεί. Ο αλγόριθμος, βέβαια, προσαρμόστηκε στις ανάγκες του

περιβάλλοντος-πλαισίου. Οι κόμβοι αντιπροσωπεύουν τα χαρακτηριστικά που καταδείχθηκαν ως σημαντικά από τον χρήστη της υπηρεσίας και επισημειώσεις έχουν προστεθεί που αφορούν ενδεχόμενη διαφοροποίηση σημαντικότητας ανάμεσα στα χαρακτηριστικά.

Η επιλογή αυτού του αλγορίθμου δίνει, επιπλέον, την δυνατότητα για μελλοντική επέκταση που αφορά συσχετίσεις ανάμεσα σε διαφορετικούς παρόχους και χαρακτηριστικά διαφορετικών υπηρεσιών.

Κεφάλαιο 3: Αρχιτεκτονική Συστήματος

3.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η αρχιτεκτονική του συστήματος που αφορά το περιβάλλον-πλαίσιο που πραγματεύεται η παρούσα διπλωματική. Συγκεκριμένα, αναφέρονται όλα τα δομικά συστατικά του συστήματος και αναλύονται οι λειτουργικότητές τους και ο τρόπος διασύνδεσης μεταξύ τους. Αναπτύσσονται, επιπλέον, οι σχεδιαστικές αποφάσεις που πάρθηκαν και οι τεχνοτροπίες που χρησιμοποιήθηκαν. Για την ανάλυση χρησιμοποιούνται ψηφιδικά διαγράμματα της γλώσσας UML.

Διαπιστώνουμε, αρχικά, ότι το σύστημά μας πρέπει να παρέχει στον χρήστη κάποιες λειτουργίες. Οι λειτουργίες είναι οι ακόλουθες:

- Πρέπει να υπάρχει η δυνατότητα να ορίζει ο χρήστης ποια χαρακτηριστικά, και με τι λογική σχέση μεταξύ τους, αφορούν την εκτίμηση της εμπιστοσύνης στην υπηρεσία υπο εξέταση.
- Πρέπει να δίνεται η δυνατότητα να διαλέγει ο χρήστης τα συστατικά που οδηγούν στην ανάλυση της εμπιστοσύνης του εκάστοτε χαρακτηριστικού. Η επιλογή πρέπει να αφορά και τις αλληλεπιδράσεις ανάμεσα στα συστατικά και το τελικό αποτέλεσμα.
- Πρέπει να μπορεί ο χρήστης να επιλέξει, επιπροσθέτως των χαρακτηριστικών, και την στρατηγική με την οποία γίνεται η τελική αποτίμηση της εμπιστοσύνης στην υπηρεσία.
- Πρέπει ενδεχόμενες αλλαγές σε κάποιο συστατικό χαρακτηριστικού να γίνονται αντιληπτές και να αλλάζει κατάλληλα και το τελικό αποτέλεσμα.

Για κάθε μια από τις ακόλουθες λειτουργίες απαιτείται ένα ή περισσότερα components. Component χρειάζεται, επιπλέον, για την λήψη πληροφοριών από εξωτερικές πηγές και για τον τελικό υπολογισμό της εμπιστοσύνης στην υπηρεσία.

Στην επόμενη παράγραφο δίνεται το συνολικό ψηφιδικό διάγραμμα της αρχιτεκτονικής του συστήματος και δίνεται μια σύντομη περιγραφή του κάθε component. Ακολουθεί η αναλυτική περιγραφή του κάθε component και στην τελευταία παράγραφο παρουσιάζεται η επικοινωνία μεταξύ τους, με την χρήση ακολουθιακών διαγραμμάτων.

3.2 Επισκόπηση της αρχιτεκτονικής

Η παρούσα ενότητα παρουσιάζει την αρχιτεκτονική του περιβάλλοντος-πλαίσιου για την εκτίμηση της εμπιστοσύνης σε υπηρεσίες. Στην αρχιτεκτονική περιλαμβάνονται τόσο τα βασικά components που χρησιμοποιεί το περιβάλλον-πλαίσιο για την λειτουργία του, όσο και επιπρόσθετα components που το επεκτείνουν, όπως το UI που επιτρέπει την αλληλεπίδραση με τον χρήστη.

Η αρχιτεκτονική που προτείνουμε και χρησιμοποιούμε στο περιβάλλον-πλαίσιο, που αναπτύσσεται στο παρούσα διπλωματική, στηρίζεται κατά βάση στο αρχιτεκτονικό πρότυπο του μαυροπίνακα. Αναλυτικότερες πληροφορίες υπάρχουν στο [86] και χρησιμοποιείται όταν η λύση δεν είναι γνωστή ντετερμινιστικά. Το μοτίβο αυτό βασίζεται στην παραδοχή ότι ένα βασικό component, που θα ονομάζεται Blackboard, περιέχει όλη την γνώση που χρειάζεται για την επίλυση του προβλήματος. Όλα τα υπόλοιπα components χρησιμοποιούν αυτή την γνώση για επιλύσουν το πρόβλημα που τους ανατίθεται και στην συνέχεια επιστρέφουν τα αποτελέσματά τους στο Blackboard. Αυτό εξασφαλίζει ότι η γνώση αφ' ενός είναι διαθέσιμη σε όλα τα components και αφ' ετέρου ενδεχόμενη αλλαγή σε κάποιο δεδομένο του προβλήματος δεν επανεκκινεί την διαδικασία, αλλά αντιμετωπίζεται τοπικά από το αντίστοιχο component.

Το πρόβλημα που καλούμαστε να επιλύσουμε στο συγκεκριμένο μοτίβο αφορά την ενημέρωση του εκάστοτε component ότι όλα τα δεδομένα που χρειάζεται υπάρχουν και η εκτέλεση του είναι δυνατή. Μια πιθανότητα είναι το **polling**, δηλαδή η περιοδική αναζήτηση καινούριας γνώσης από το κάθε component. Αυτό, όμως, είναι μη αποδοτικό υπολογιστικά. Έτσι αποφασίστηκε η χρήση του σχεδιαστικού μοτίβου Publisher/Subscriber. Σε αυτή την περίπτωση χρησιμοποιείται ένα επιπλέον component το οποίο γνωρίζει ποιά συγκεκριμένα δεδομένα αφορούν το κάθε component και παρέχει ενημέρωση στην εκάστοτε περίπτωση. Συγκεκριμένα, τα υπόλοιπα component ενημερώνουν το κεντρικό αυτό component για το ενδιαφέρον τους, όσον αφορά κάποια δεδομένα, και εκείνο τους ενημερώνει σε περίπτωση τροποποίησης.

Στην συνέχεια ακολουθεί το ψηφιδωτικό διάγραμμα της αρχιτεκτονικής του συστήματος βασιζόμενοι σε όσα αναπτύξαμε παραπάνω:

καθώς επίσης και τον βαθμό στον οποίο την επηρεάζουν. Τέλος δίνεται η δυνατότητα, μέσω αυτού του component να επιλέξει και την περαιτέρω συσχέτιση ανάμεσα στα χαρακτηριστικά.

- **TrustSpecificationModeler:** Αυτό το component αναλαμβάνει να μοντελοποιήσει τον ορισμό της εμπιστοσύνης σε μια υπηρεσία. Αυτό γίνεται σε μορφή γράφου ή εξίσωσης, ανάλογα με την στρατηγική που έχει επιλεγεί από τον χρήστη του περιβάλλοντος-πλαισίου. Αν ο χρήστης δεν έχει κάνει κάποια διαφορετική επιλογή, χρησιμοποιείται ως στρατηγική ο αλγόριθμος TidalTrust που περιγράφεται στο κεφάλαιο 5.
- **RequirementEditor:** Αυτό το component παρέχει την δυνατότητα στον χρήστη να ορίσει και να παραμετροποιήσει λεπτομερώς τα επιμέρους χαρακτηριστικά που συμβάλουν στην εμπιστοσύνη της κάθε υπηρεσίας. Πιο συγκεκριμένα, δίνεται η επιλογή δημιουργίας νέων δέντρων στόχων, που αποτελούνται από τα συστατικά και έχουν ως ρίζα τα εν λόγω χαρακτηριστικά της υπηρεσίας. Έτσι υπάρχει ευελιξία ορισμού πλήθους διαφορετικών χαρακτηριστικών.
- **RequirementModeler:** Αυτό το component χρησιμοποιείται προκειμένου να γίνει η μοντελοποίηση των δέντρων στόχων που ορίστηκαν από το προηγούμενο component και να αναπαρασταθεί το μεταμοντέλο στο οποίο πρέπει να υπακούν τα μοντέλα δέντρων στόχων.
- **Strategy:** Αυτό το component χρησιμοποιείται για να δοθεί στον χρήστη η δυνατότητα επιλογής της στρατηγικής αποτίμησης της εμπιστοσύνης. Παρέχεται ένα εύρος από επιλογές αλγορίθμων ανάμεσα στις οποίες μπορεί να διαλέξει ο ενδιαφερόμενος.
- **InformationGathering:** Αυτό το component χρησιμοποιείται για την λήψη πληροφοριών από εξωτερικές πηγές. Κάθε συστατικό, που απαρτίζει το κάθε χαρακτηριστικό το οποίο συμμετέχει στην εκτίμηση της εμπιστοσύνης της υπηρεσίας, έχει μια διακριτή τιμή, που εκφράζεται με την μορφή ποσοστού. Αυτή η τιμή υπολογίζεται από αυτό το component με διαφορετικούς τρόπους ανάλογα με την περίπτωση. Κάποιες τιμές προέρχονται από άμεση εισαγωγή μέσω αρχείου, αν π.χ. πρόκειται για πληροφορία που παρέχεται από φυσικό έλεγχο. Άλλες παρέχονται από κάποια άλλη υπηρεσία που θεωρείται αξιόπιστη. Και τέλος, κάποιες άλλες μπορεί να προέρχονται από κάποια άλλη εφαρμογή που τρέχει τοπικά στο σύστημα, όπως π.χ. κάποιο εργαλείο εκτίμησης ηλεκτρονικής ασφάλειας της υπηρεσίας που εκτιμάται.
- **Reasoning Module:** Αυτό το component είναι υπεύθυνο για τον υπολογισμό της εμπιστοσύνης που απολαμβάνει το κάθε επιμέρους χαρακτηριστικό της υπηρεσίας που προσφέρεται. Συγκεκριμένα αναθέτει τις τιμές των συστατικών στους αντίστοιχους κόμβους του δέντρου στόχου (goal model) του χαρακτηριστικού. Μόλις ολοκληρωθεί

αυτή η διαδικασία, υπολογίζονται με τους κατάλληλους αλγορίθμους οι τιμές όλων των κόμβων, και τελικά και η τιμή της ρίζας του δέντρου, που στην ουσία αναπαριστά, με την μορφή ποσοστιαίας τιμής και πάλι, την εμπιστοσύνη που απολαμβάνει το κάθε χαρακτηριστικό της υπηρεσίας που μας ενδιαφέρει.

- **Alarmer:** Το component έχει επιφορτιστεί με μια πολύ συγκεκριμένη λειτουργία. Φροντίζει μετά την πάροδο ενός συγκεκριμένου χρονικού διαστήματος να κηρύσσει ως παρωχημένες όλες τις τιμές των συστατικών. Αυτή η ενέργεια εξασφαλίζει ότι, ακόμα και αν μια αλλαγή δεν ανιχνευθεί από το σύστημα, μετά από κάποιο χρονικό διάστημα οι τιμές θα ανανεωθούν υποχρεωτικά. Έτσι μας επιτρέπεται να ορίζουμε εμπιστοσύνη σε μια υπηρεσία, η οποία ισχύει για ένα, καθορισμένο από τον χρήστη, χρονικό περιθώριο και όχι για πάντα. Αυτό είναι απολύτως επιθυμητό διότι και στον φυσικό κόσμο η εμπιστοσύνη δεν υπάρχει για πάντα.
- **TrustEvaluator:** Η λειτουργία αυτού του component έγκειται στον τελικό υπολογισμό της εμπιστοσύνης που απολαμβάνει μια υπηρεσία, που παρέχεται από έναν πάροχο. Συγκεκριμένα, το component αυτό χρησιμοποιεί τον γράφο ή την εξίσωση που παρέχεται, ανάλογα με την στρατηγική που επιλέγεται. Αντιστοιχεί τις υπολογισμένες τιμές των χαρακτηριστικών της υπηρεσίας στους αντίστοιχους κόμβους ή μεταβλητές, και τέλος χρησιμοποιώντας των κατάλληλο αλγόριθμο εξάγει την τελική τιμή εμπιστοσύνης.
- **Report:** Το component αυτό χρησιμοποιείται για την ενημέρωση του χρήστη για τα γεγονότα που λαμβάνουν χώρα κατά την διάρκεια του υπολογισμού της εμπιστοσύνης. Ειδικότερα, καταγράφονται οι λήψεις πληροφοριών που αφορούν τα χρησιμοποιούμενα συστατικά. Αναφέρονται, επίσης, και οι αλλαγές που οδηγούν σε ακύρωση και επαναυπολογισμό των συστατικών, των χαρακτηριστικών και της εμπιστοσύνης στην υπηρεσία. Τέλος, παρουσιάζονται στον χρήστη οι τιμές που υπολογίζονται για τα επιμέρους χαρακτηριστικά και η τιμή που υπολογίζει το περιβάλλον-πλαίσιο για την εμπιστοσύνη στην υπηρεσία που ενδιαφέρει το χρήστη.
- **UI:** Μέσω αυτού του component πραγματοποιείται η επικοινωνία ανάμεσα στον χρήστη και το περιβάλλον-πλαίσιο. Με την χρήση του component αυτού, ο χρήστης αποκτά πλήρη ελευθερία στην διαδικασία ορισμού της εμπιστοσύνης μιας υπηρεσίας. Δίνεται, λοιπόν, η δυνατότητα ορισμού goal trees που περιγράφουν τα χαρακτηριστικά μιας υπηρεσίας. Αυτό περιλαμβάνει και την δομή του δέντρου, αλλά και την προέλευση των

πληροφοριών που αφορούν τα συστατικά του κάθε χαρακτηριστικού. Πλήρως παραμετροποιήσιμη από τον χρήστη είναι, επίσης, η αλληλουχία και η διασύνδεση των χαρακτηριστικών, που οδηγούν στον υπολογισμό της εμπιστοσύνης.

3.3 Αναλυτική περιγραφή των Components

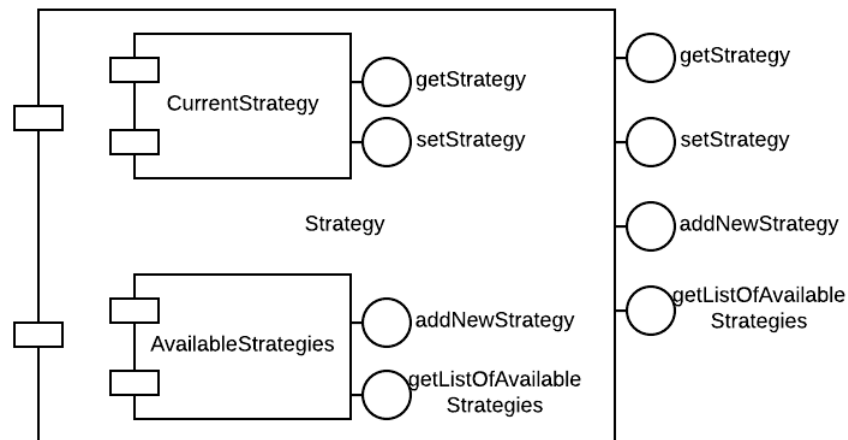
Σε αυτό το κεφάλαιο παρουσιάζονται αναλυτικά ένα ένα τα επιμέρους components του συστήματος που αναφέρθηκαν επιγραμματικά στην προηγούμενη ενότητα. Εξηγούνται αναλυτικά οι διεπαφές του κάθε component και αναλύεται η εσωτερική τους δομή. Παρέχεται, δε, όπου θεωρείται απαραίτητο για την κατανόηση, επιπλέον UML διάγραμμα προκειμένου να δειχθούν τα επιμέρους συστατικά του εκάστοτε component και οι αλληλεπιδράσεις μεταξύ τους.

Σε κάποιες περιπτώσεις, όπως φαίνεται και από το ψηφιδωτικό διάγραμμα της προηγούμενης παραγράφου, οι διασυνδέσεις κάποιων components έχουν απλοποιηθεί αρκετά. Συγκεκριμένα, θεωρούμε ότι όσα components εξαρτώνται από την γνώση που περιέχεται στο blackboard δρουν αυτόνομα. Κάνουν, δηλαδή, αρχικά *subscribe* στο *Pub/Sub Module* και ενεργοποιούνται μόνον όταν ειδοποιηθούν ότι παρουσιάστηκε κάποια γνώση που τα αφορά. Η μοναδική διεπαφή, λοιπόν, που πρέπει να υλοποιείται εξωτερικά από αυτά τα components είναι αυτή που θα χρησιμοποιείται για την ειδοποίηση τους όταν ένα τέτοιο γεγονός λαμβάνει χώρα. Η μέθοδος, που υλοποιεί αυτή την διεπαφή, μπορεί να ονομαστεί *notifyNewEvent* και τα components που την παρέχουν είναι ο *TrustEvaluator* και ο *ReasoningModule*.

3.3.1 Strategy

Το component Strategy είναι υπεύθυνο για τον ορισμό και την ανάκτηση της στρατηγικής με την οποία υπολογίζεται η τελική εμπιστοσύνη στην υπηρεσία που μας προσφέρεται. Μια περίπτωση περιλαμβάνει τον αλγόριθμο *TidalTrust*, που υπολογίζει την τελική τιμή μέσω της κατασκευής κατάλληλου γράφου. Ο αλγόριθμος αυτός είναι ο προεπιλεγμένος και παρουσιάζεται αναλυτικά στο κεφάλαιο 5. Εναλλακτικά ο ορισμός μπορεί να δοθεί με την μορφή εξίσωσης που χρησιμοποιεί λογικούς τελεστές. Στρατηγικές μπορούν επίσης να προστεθούν και στο μέλλον σε μορφή επέκτασης του περιβάλλοντος-πλαισίου.

Το ψηφιδωτικό του διάγραμμα είναι το ακόλουθο:



Σχήμα 3.3-1 Ψηφιακό διάγραμμα κλάσης Strategy

Όπως παρατηρούμε τα sub-components του είναι τα εξής:

- **CurrentStrategy:** Στο component αυτό αποθηκεύεται η πληροφορία σχετικά με την στρατηγική που χρησιμοποιείται την παρούσα στιγμή
- **AvailableStrategies:** Εδώ υπάρχουν όλες οι διαθέσιμες στρατηγικές προς χρήση και η δυνατότητα για προσθήκη νέων.

Οι διεπαφές που προσφέρει είναι οι εξής:

- **getStrategy:** Επιστρέφει την στρατηγική που έχει επιλεγεί προς χρήση στον υπολογισμό της εμπιστοσύνης
- **setStrategy:** Ορίζει την στρατηγική που θα χρησιμοποιηθεί από εδώ και πέρα στον υπολογισμό της εμπιστοσύνης
- **getListOfAvailableStrategies:** Επιστρέφει την λίστα με τις, προς το παρόν, διαθέσιμες στρατηγικές για χρήση
- **addNewStrategy:** Προσθέτει μια καινούργια επιλογή στρατηγικής, που μπορεί να χρησιμοποιηθεί σαν εναλλακτική

3.3.2 InformationGathering

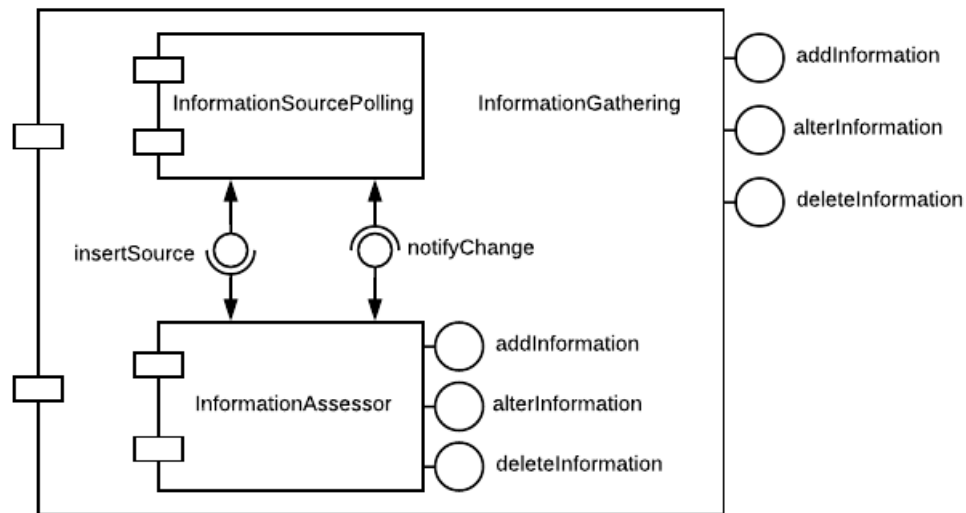
Το component InformationGathering είναι επιφορτισμένο με την λειτουργία ανάκτησης πληροφοριών από, εξωτερικές του περιβάλλοντος-πλαισίου, πηγές.

Η εμπιστοσύνη σε μια υπηρεσία εξαρτάται, όπως είναι φυσικό, από κάποια χαρακτηριστικά. Τα χαρακτηριστικά αυτά ορίζουν πρακτικά τους παράγοντες που επηρεάζουν την αποτίμηση της γενικής εμπιστοσύνης. Τέτοια χαρακτηριστικά είναι, για

παράδειγμα, η ασφάλεια, η ταχύτητα, η σταθερότητα κ.α. Τα χαρακτηριστικά, όμως, με την σειρά τους εξαρτώνται από κάποια συστατικά. Παραδείγματα τέτοιων συστατικών είναι η αξιολόγηση από προηγούμενους χρήστες και το hardware που χρησιμοποιείται για την υπηρεσία αυτή.

Αρμοδιότητα, λοιπόν, του παρόντος component είναι η λήψη πληροφοριών από εξωτερικές πηγές, προκειμένου να εκτιμηθεί, με την μορφή ποσοστού, η θετική συνεισφορά του κάθε συστατικού. Η πληροφορία αυτή θα αποθηκευθεί στο blackboard και θα χρησιμοποιηθεί από όποιο άλλο component την χρειάζεται.

Το ψηφιακό διάγραμμά του είναι το ακόλουθο:



Σχήμα 3.3-2 Ψηφιακό διάγραμμα κλάσης InformationGathering

Τα sub-components που παρατηρούμε είναι τα εξής:

- **InformationSourcePolling:** Το component αυτό κάνει polling ανά κάποια ορισμένα χρονικά διαστήματα στις εξωτερικές πηγές. Αν παρατηρηθεί κάποια μεταβολή στην παρεχόμενη πληροφορία, ενημερώνεται το επόμενο component.
- **InformationAssessor:** Το component αυτό ενεργοποιείται όταν ενημερωθεί για κάποια αλλαγή και, αφού βρει από την βάση δεδομένων του σε ποια πληροφορία αντιστοιχεί αυτή η πηγή, εκτιμά την ποσοστιαία συνεισφορά της πληροφορίας/συστατικού. Στην συνέχεια, δημοσιεύει την ανανεωμένη τιμή στο blackboard προς κατανάλωση από άλλα components.

Οι διεπαφές που προσφέρονται είναι οι εξής και αφορούν το δεύτερο component που επικοινωνεί στην συνέχεια εσωτερικά με το πρώτο:

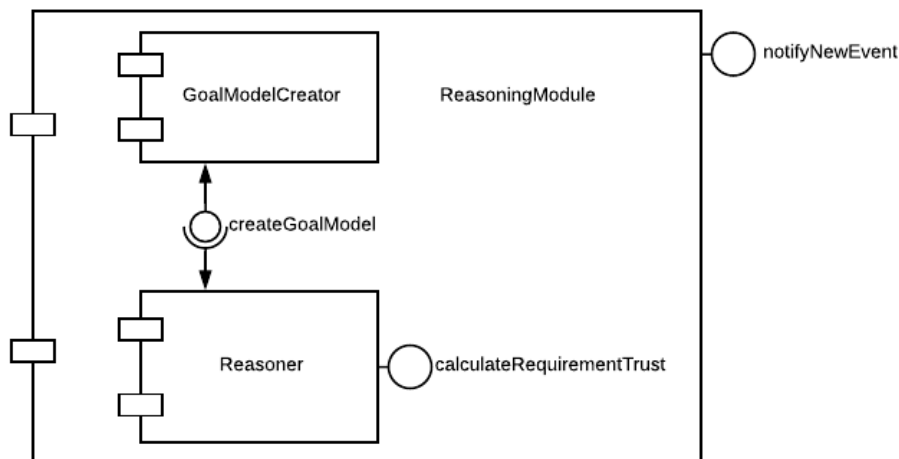
- **addInformation:** Προσθέτει ένα καινούργιο συστατικό, που μπορεί να χρησιμοποιηθεί για την εκτίμηση της εμπιστοσύνης ενός ή περισσότερων χαρακτηριστικών της υπηρεσίας. Μαζί προστίθενται φυσικά και η πηγή λήψης της απαραίτητης πληροφορίας.
- **alterInformation:** Αλλάζει την πηγή λήψης της πληροφορίας για ένα συστατικό που έχει προστεθεί στο παρελθόν.
- **deleteInformation:** Διαγράφει ένα συστατικό από την μνήμη, είτε επειδή είναι παρωχημένο σαν πληροφορία ή διότι δεν αφορά πλέον την υπηρεσία υπό εξέταση, και συγκεκριμένα τα χαρακτηριστικά που την απαρτίζουν.

3.3.3 ReasoningModule

Το component αυτό χρησιμοποιείται για τον συμπερασμό της εμπιστοσύνης, που απολαμβάνει κάθε χαρακτηριστικό της υπηρεσίας που μας ενδιαφέρει.

Συγκεκριμένα, το component λαμβάνει το μεταμοντέλο του δέντρου στόχων, που αντιπροσωπεύει το χαρακτηριστικό που μας ενδιαφέρει. Αφού λάβει και γεμίσει τους κόμβους με τις πληροφορίες για τα συστατικά που το αποτελούν, εκτελεί reasoning πάνω στο δέντρο στόχων με βάση την θεωρία που έχει αναπτυχθεί στο κεφάλαιο 2. Η τιμή που ανατίθεται στην ρίζα του δέντρου αποτελεί την εμπιστοσύνη του χαρακτηριστικού, πάντα βέβαια για την δεδομένη δομή του δέντρου που ορίσαμε.

Το ψηφιδωτικό του διάγραμμα είναι το ακόλουθο:



Σχήμα 3.3-3 Ψηφιδικό διάγραμμα κλάσης ReasoningModule

Τα sub-components που το αποτελούν είναι τα εξής:

- **GoalModelCreator:** Το component αυτό παίρνει το μεταμοντέλο που έχει δημιουργηθεί για το χαρακτηριστικό που μας ενδιαφέρει, καθώς επίσης και τα συστατικά/πληροφορίες που απαιτούν οι κόμβοι του δέντρου στόχων. Στην συνέχεια δημιουργεί το δέντρο στόχων και το περνάει στο επόμενο component.
- **Reasoner:** Το component παίρνει ως είσοδο το δέντρο στόχων, εκτελεί το reasoning πάνω σε αυτό και τελικά επιστρέφει την τιμή που έχει η ρίζα του δέντρου. Το αποτέλεσμα αυτό το γράφει στο blackboard, προκειμένου να χρησιμοποιηθεί από όποιο component το χρειάζεται.

Το component διαθέτει, όπως είπαμε και στην εισαγωγή του κεφαλαίου, μία και μοναδική διεπαφή. Αυτή είναι η notifyNewEvent. Όταν κάποιο από τα δεδομένα που χρειάζεται για να εκτελεστεί γίνει αρχικά διαθέσιμο, ή αλλάξει τιμή, το component ειδοποιείται και υπολογίζεται η εμπιστοσύνη που απολαμβάνει το χαρακτηριστικό που μας ενδιαφέρει.

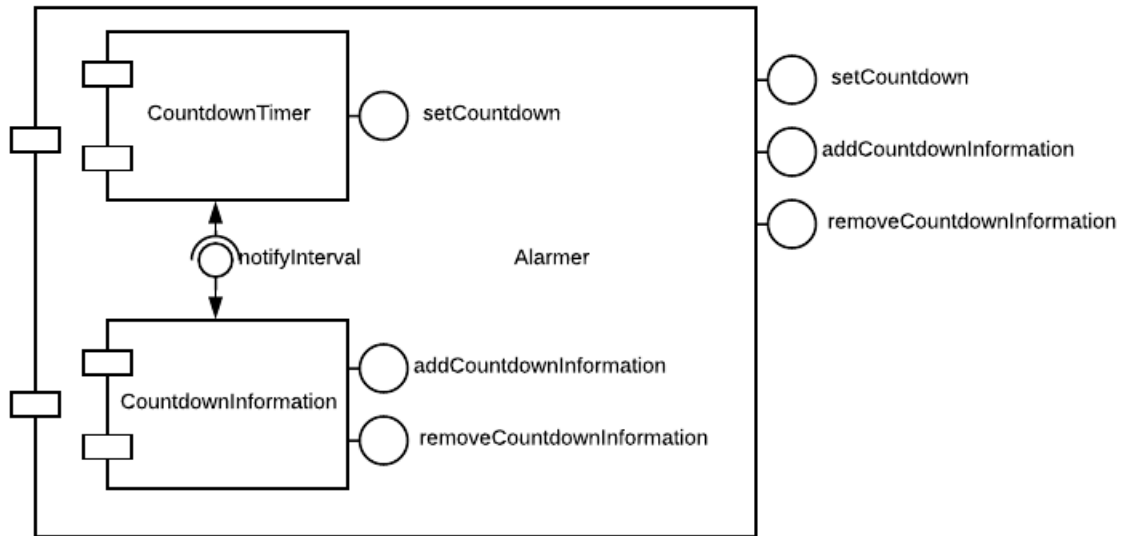
3.3.4 Alarmer

Το component αυτό χρησιμοποιείται για να ειδοποιείται το σύστημα τότε πρέπει να επαναυπολογιστούν μια, ή περισσότερες, τιμές.

Συγκεκριμένα, περιέχεται ένας εσωτερικός αντίστροφος χρονομετρητής. Όταν λήξει η αντίστροφη μέτρηση, κάποια, ή όλα, τα δεδομένα, που υπάρχουν μέχρι στιγμής αποθηκευμένα στο blackboard, θεωρούνται άκυρα και επιβάλλεται ο εξ' αρχής επαναπροσδιορισμός τους. Αυτό εξασφαλίζει την βεβαιότητα ότι κανένα δεδομένο δεν

είναι αιώνιο. Έτσι αν μια αλλαγή δεν ανιχνευθεί είμαστε σίγουροι ότι δεν θα παραμείνει το σύστημα με λάθος δεδομένα για πάντα. Ο χρόνος ακύρωσης αλλά και οι πληροφορίες που ακυρώνονται ορίζονται από τον χρήστη.

Το ψηφιακό του διάγραμμα είναι το εξής:



Σχήμα 3.3-4 Ψηφιακό διάγραμμα κλάσης Alarmer

Τα sub-components που το αποτελούν είναι τα ακόλουθα:

- **CountdownTimer**: Το component αυτό είναι υπεύθυνο για την αντίστροφη μέτρηση. Σε αυτό το component ορίζονται ο χρόνος που μεσολαβεί ανάμεσα στις ακυρώσεις
- **CountdownInformation**: Το component αυτό είναι υπεύθυνο για την διαχείριση της πληροφορίας που αφορά ποια δεδομένα ακυρώνονται όταν παρέλθει ο απαιτούμενος χρόνος.

Οι διεπαφές που προσφέρονται από το συγκεκριμένο component είναι οι ακόλουθες:

- **setCountdown**: Εδώ ορίζεται ο χρόνος που πρέπει να περάσει μέχρι οι πληροφορίες που έχουν επιλεγεί να θεωρηθούν παρωχημένες και να ζητηθεί ο επαναυπολογισμός τους.
- **addCountdownInformation**: Αυτή η διεπαφή επιτρέπει την προσθήκη συγκεκριμένων πληροφοριών που θεωρούνται παρωχημένες μετά το πέρας του χρόνου, που έχει οριστεί.

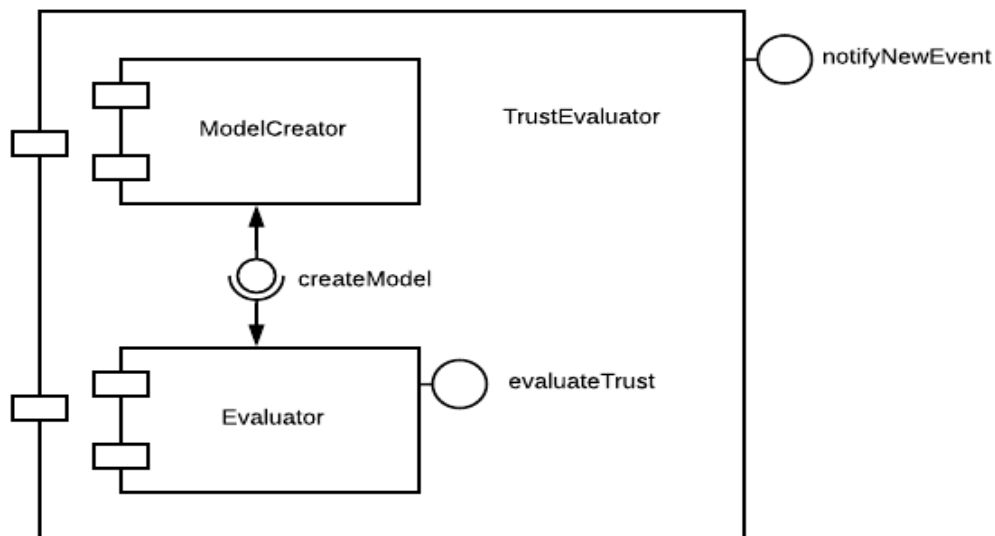
- **removeCountdownInformation:** Αυτή η διεπαφή επιτρέπει την αφαίρεση συγκεκριμένων πληροφοριών που δεν θεωρούνται πλέον εξαρτώμενες από τον χρόνο.

3.3.5 TrustEvaluator

Το component αυτό είναι υπεύθυνο για τον τελικό υπολογισμό της εμπιστοσύνης που απολαμβάνει μια υπηρεσία που παρέχει κάποιος συγκεκριμένος πάροχος.

Αρχικά, αφού έχει γνωστοποιηθεί στο component η στρατηγική που θα χρησιμοποιηθεί, λαμβάνεται ο γράφος ή η μαθηματική έκφραση που αντιπροσωπεύει την εμπιστοσύνη στην υπηρεσία, ανάλογα με την περίπτωση. Στην συνέχεια, στον γράφο, στην περίπτωση του TidalTrust που είναι η προεπιλογή, συμπληρώνονται οι τιμές που έχουν ήδη υπολογιστεί για τα αντίστοιχα χαρακτηριστικά. Τέλος, εκτελείται ο αλγόριθμος που έχει επιλεγεί και επιστρέφεται το αποτέλεσμα, που αντιπροσωπεύει και το τελικό ποσοστό εμπιστοσύνης που δίνει το περιβάλλον-πλαίσιο στην υπηρεσία που μας ενδιαφέρει.

Το ψηφιδωτικό διάγραμμά του είναι το ακόλουθο:



Σχήμα 3.3-5 Ψηφιδωτικό διάγραμμα κλάσης TrustEvaluator

Τα sub-components που το αποτελούν είναι τα εξής:

- **ModelCreator:** Το component αυτό λαμβάνει τα δεδομένα που χρειάζεται για την δημιουργία του μοντέλου που απαιτείται για τον υπολογισμό της εμπιστοσύνης

στην υπηρεσία. Μαθαίνει, δηλαδή, την στρατηγική που θα χρησιμοποιηθεί και στην συνέχεια, παίρνει την αντίστοιχη μορφή μοντέλου που ορίζεται. Τέλος, λαμβάνει και συμπληρώνει τις τιμές των χαρακτηριστικών, στις θέσεις που αντιστοιχούν.

- **Evaluator:** Το component αυτό λαμβάνει όλα τα απαραίτητα στοιχεία από το προηγούμενο component και εκτελεί τον αλγόριθμο που έχει επιλεγεί προκειμένου να υπολογιστεί το τελικό αποτέλεσμα. Στην περίπτωση του προεπιλεγμένου TidalTrust αλγορίθμου, η τιμή της εμπιστοσύνης στην υπηρεσία βρίσκεται στον τελευταίο κόμβο του κατευθυνόμενου γράφου, που χρησιμοποιείται ως sink.

Το component αυτό, όπως αναφέρθηκε και στην εισαγωγή του κεφαλαίου, προσφέρει μία και μοναδική διεπαφή. Αυτή είναι η `notifyNewEvent`. Όταν κάποιο από τα δεδομένα που χρειάζεται για να εκτελεστεί γίνει αρχικά διαθέσιμο, ή αλλάξει τιμή, το component ειδοποιείται και υπολογίζεται η εμπιστοσύνη που απολαμβάνει το χαρακτηριστικό που μας ενδιαφέρει.

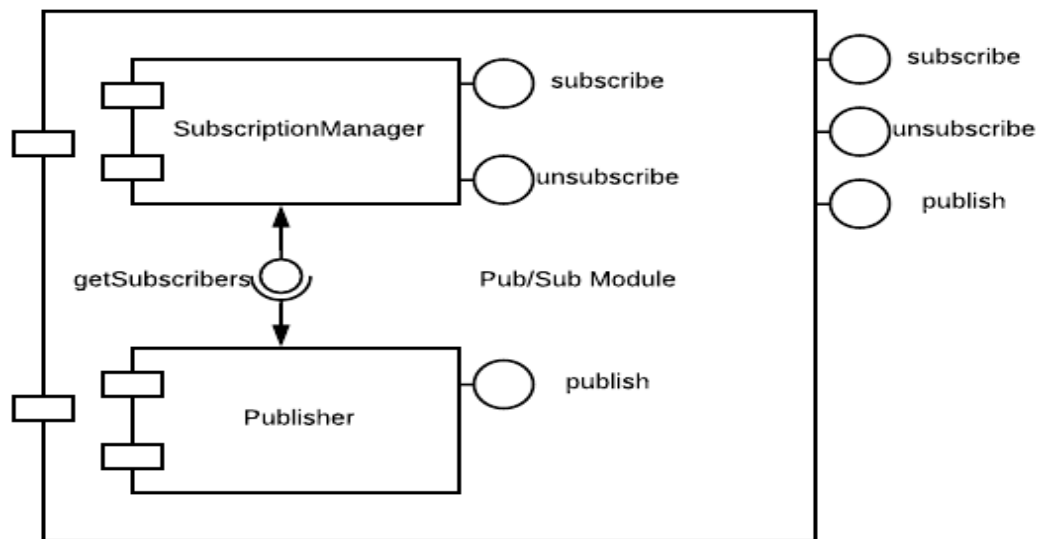
3.3.6 Pub/Sub Module

Το component αυτό χρησιμοποιείται για την ειδοποίηση συγκεκριμένων components για συγκεκριμένα events.

Αναλυτικότερα, επιτρέπει στα υπόλοιπα components να εκδηλώνουν ενδιαφέρον για κάποια γεγονότα(events) που λαμβάνουν χώρα κατά την διάρκεια εκτέλεσης του περιβάλλοντος-πλαισίου. Όταν κάποιο από τα γεγονότα αυτά συμβεί, τα αντίστοιχα components ενημερώνονται γι' αυτό. Αυτό διευκολύνει την αποσυνδεδεμένη επικοινωνία, χωρίς την ανάγκη ενεργοβόρων τεχνικών, όπως είναι το polling.

Για να γίνει δυνατόν αυτό, προσφέρεται η δυνατότητα εγγραφής των components για συγκεκριμένους τύπους γεγονότων(events). Με αυτό τον τρόπο το Pub/Sub Module κρατάει λίστες ενδιαφερόμενων για κάθε event. Έτσι ούτε ο ενδιαφερόμενος γνωρίζει ποιος δημιούργησε το event, αλλά ούτε και αυτός που το δημιούργησε ενδιαφέρεται για το ποιον επηρεάζει.

Το ψηφιδωτικό του διάγραμμα είναι το ακόλουθο:



Σχήμα 3.3-6 Ψηφιακό διάγραμμα κλάσης Pub/Sub Module

Τα sub-components που το απαρτίζουν είναι τα εξής:

- **SubscriptionManager:** Το component αυτό είναι υπεύθυνο για την διατήρηση των λιστών, που περιέχουν όλα τα δυνατά events και τα components, που έχουν εκδηλώσει ενδιαφέρον για το κάθε ένα από αυτά. Παρέχει την δυνατότητα εγγραφής ή διαγραφής από τις λίστες αυτές. Η εύρεση των ενδιαφερόμενων για κάθε γεγονός πρέπει να γίνεται αποδοτικά.
- **Publisher:** Το component αυτό λαμβάνει ειδοποίηση για τα νέα γεγονότα που προκύπτουν. Στην συνέχεια, είναι υπεύθυνο να βρει και να ενημερώσει αυτούς που έχουν δηλώσει ενδιαφέρον για το συγκεκριμένο event.

Οι διεπαφές που προσφέρει το component είναι οι ακόλουθες:

- **subscribe:** Μέσω αυτής της διεπαφής τα components εκδηλώνουν ενδιαφέρον για συγκεκριμένα events, που προκύπτουν κατά την εκτέλεση του περιβάλλοντος-πλαισίου, προκειμένου να ενημερώνονται όταν αυτά συμβαίνουν.
- **unsubscribe:** Μέσω αυτής της διεπαφής τα components δηλώνουν ότι δεν ενδιαφέρονται πλέον για κάποια events, με αποτέλεσμα να σταματούν οι ειδοποιήσεις όταν αυτά συμβαίνουν.
- **publish:** Σε αυτή την διεπαφή τα components δηλώνουν την δημιουργία ενός συγκεκριμένου τύπου event, που προκλήθηκε κατά την εκτέλεσή τους. Αυτό το event, ακολούθως, γνωστοποιείται σε όλους τους ενδιαφερόμενους.

3.4 Επικοινωνία μεταξύ των Components του συστήματος

Σε αυτή την ενότητα παρουσιάζεται η λειτουργία του συστήματος και ο τρόπος με τον οποίο αλληλεπιδρούν τα components που το αποτελούν. Αυτό γίνεται με την χρήση ακολουθιακών διαγραμμάτων.

Όπως έχει αναφερθεί σε προηγούμενες ενότητες, τα components επικοινωνούν μεταξύ τους με την βοήθεια του Pub/Sub Module. Το εκάστοτε component εκτελεί μια βασική υπο-λειτουργία του συστήματος. Ενεργοποιείται, όμως, μόνο όταν λάβει ενημέρωση για κάποιο γεγονός, για το οποίο έχει εκδηλώσει ενδιαφέρον στο Pub/Sub Module. Εξασφαλίζεται έτσι η ύπαρξη των απαραίτητων δεδομένων για την διεκπεραίωση της υπο-λειτουργίας. Οι λειτουργίες αυτές αφορούν την συλλογή δεδομένων, την ανάλυση και υπολογισμό των χαρακτηριστικών της υπηρεσίας, την εκτίμηση της τελικής εμπιστοσύνης στην υπηρεσία και τέλος την διατήρηση χρονικού πλαισίου, στο οποίο όλα τα παραπάνω ισχύουν. Κάθε component φροντίζει, μετά την εκτέλεση της υπο-λειτουργίας που του έχει ανατεθεί, να γράφει τα αποτελέσματά του στο Blackboard.

Η λειτουργία ανάκτησης πληροφοριών από εξωτερικές πηγές εκτελείται διαρκώς και ανεξάρτητα από τις υπόλοιπες λειτουργίες. Επομένως, κρίνεται σκόπιμη η ανάθεσή της σε χωριστό νήμα της εφαρμογής, προκειμένου να μην επηρεάζει την εξέλιξη των υπολοίπων λειτουργιών. Αντίστοιχης αντιμετώπισης χρίζει και η λειτουργία ειδοποίησης του συστήματος σχετικά με πιθανό timeout των αποτελεσμάτων, μιας και μπορεί να προκύψει γεγονός οποιαδήποτε στιγμή. Οι υπόλοιπες λειτουργίες εξαρτώνται σειριακά η μία από την άλλη και, ως εκ τούτου, το κύριο νήμα της εφαρμογής είναι αρκετό. Παρ' όλα αυτά, αλλαγή μπορεί να προκύψει σε οποιοδήποτε σημείο της διαδικασίας, και γι' αυτό κάθε κομμάτι εξετάζεται χωριστά.

Παρατίθενται, λοιπόν, ακολουθιακά διαγράμματα για όλες τις βασικές λειτουργίες του περιβάλλοντος-πλαισίου, προκειμένου να διευκολυνθεί η περιγραφή τους.

3.4.1 Ακολουθιακό Διάγραμμα Δημιουργίας Νέων Δεδομένων

Στο διάγραμμα που έπεται περιγράφεται η διαδικασία που ακολουθείται όταν προκύψουν νέα δεδομένα από τις εξωτερικές πηγές που παρακολουθεί το σύστημά μας.

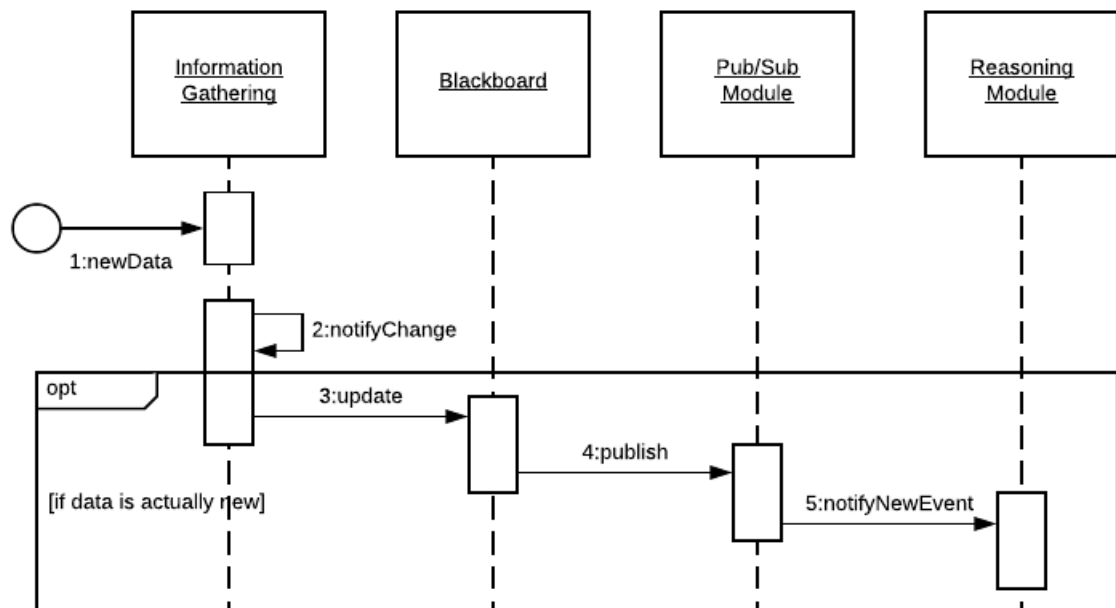
Περιλαμβάνει την συλλογή των δεδομένων, την εγγραφή τους στο Blackboard και την ενημέρωση των components που έχουν εκδηλώσει ενδιαφέρον για αυτά. Αναλυτικά τα βήματα είναι τα εξής:

1. **newData:** Το component InformationGathering λαμβάνει νέα δεδομένα από κάποια εξωτερική πηγή που παρακολουθείται

2. **notifyChange:** Ενημερώνεται εσωτερικά το αντίστοιχο sub-component για την αλλαγή που παρουσιάστηκε. Εκεί ελέγχεται κατα πόσο η πληροφορία που προέκυψε είναι όντως καινούργια.

Εφόσον οι πληροφορίες που δημιουργήθηκαν είναι όντως διαφορετικές από τις ήδη υπάρχουσες, ακολουθούν οι επόμενες ενέργειες:

3. **update:** Η αλλαγή στο περιεχόμενο των πληροφοριών γνωστοποιείται στο Blackboard, προκειμένου να γίνει διαθέσιμη σε όποιο άλλο component την έχει ανάγκη.
4. **publish:** Το Blackboard στέλνει τα νεοαποκτηθέντα δεδομένα στο Pub/Sub Module, προκειμένου αυτό να ενημερώσει με την σειρά του όλα τα ενδιαφερόμενα μέλη για το νέο γεγονός.
5. **notifyNewEvent:** Το Pub/Sub Module ενημερώνει το ReasoningModule για την αλλαγή στο περιεχόμενο των πληροφοριών.

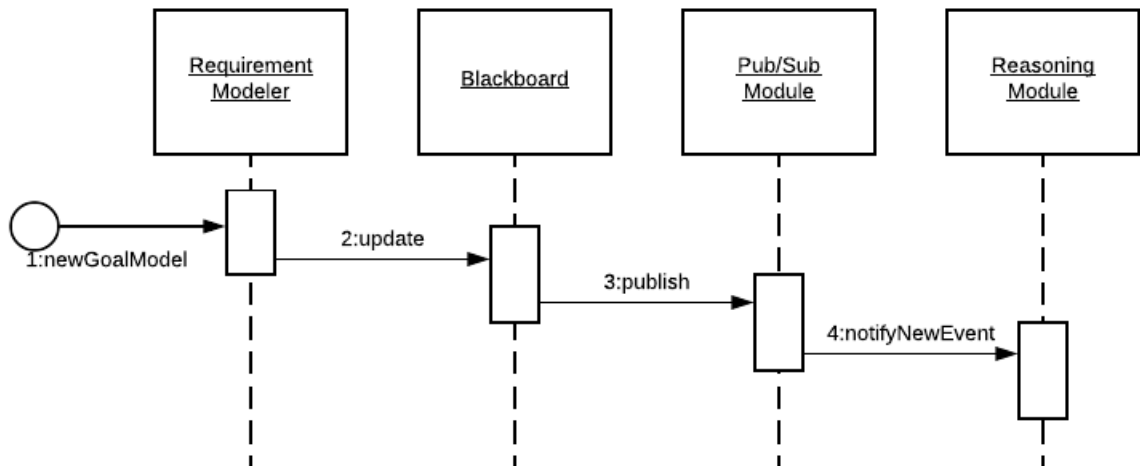


Σχήμα 3.4-1 Ακολουθιακό Διάγραμμα Δημιουργίας Νέων Δεδομένων

3.4.2 Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Δέντρου Στόχων Απαίτησης

Στο διάγραμμα που έπεται περιγράφεται η διαδικασία που ακολουθείται όταν παραχθεί κάποιο νέο δέντρο στόχων για κάποιο από τα επιμέρους χαρακτηριστικά της υπηρεσίας. Αυτό, πρακτικά, σημαίνει ότι ο χρήστης του περιβάλλοντος-πλαισίου μετέβαλλε τον ορισμό κάποιου χαρακτηριστικού προκειμένου να ανταποκρίνεται στις νέες του ανάγκες. Αναλυτικά η διαδικασία είναι η εξής:

1. **newGoalModel:** Το RequirementModeler λαμβάνει ένα καινούργιο δέντρο στόχων από τον χρήστη, και συγκεκριμένα από το RequirementEditor.
2. **update:** Το ανανεωμένο δέντρο στόχων γνωστοποιείται στο Blackboard, προκειμένου να γίνει διαθέσιμη σε όποιο άλλο component την έχει ανάγκη.
3. **publish:** Το Blackboard στέλνει το καινούργιο δέντρο στόχων στο Pub/Sub Module, προκειμένου αυτό να ενημερώσει με την σειρά του όλα τα ενδιαφερόμενα μέλη για το νέο γεγονός.
4. **notifyNewEvent:** Το Pub/Sub Module ενημερώνει το ReasoningModule για την αλλαγή στο δέντρο στόχων.

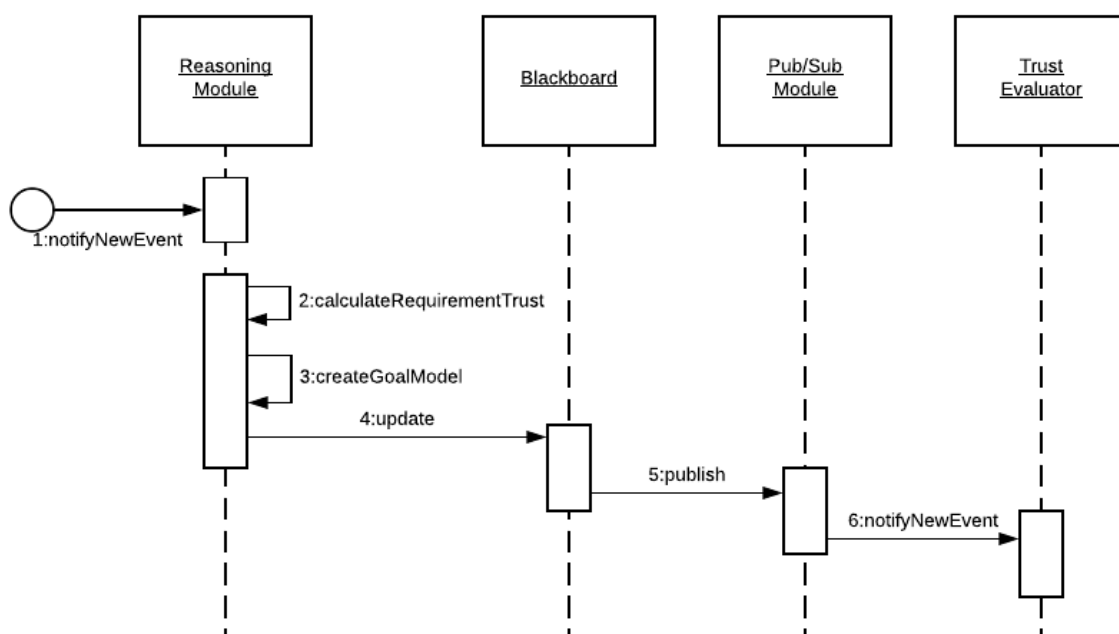


Σχήμα 3.4-2 Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Δέντρου Στόχων Απαίτησης

3.4.3 Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Απαίτησης

Στο διάγραμμα που έπεται περιγράφεται η διαδικασία που ακολουθείται όταν μεταβληθεί μια απαίτηση, που επηρεάζει την τελική εμπιστοσύνη στην υπηρεσία που ερευνάται. Αυτή η περίπτωση προκύπτει όταν πραγματοποιηθούν ένα, ή και τα δύο, γεγονότα που παρουσιάζονται στις υπο-ενότητες 3.4.1, 3.4.2. Αναλυτικά η διαδικασία είναι η εξής:

1. **notifyNewEvent:** Το ReasoningModule ενημερώνεται για την δημιουργία ενός, ή περισσότερων, event που αντιστοιχούν σε αυτά που περιγράφηκαν στις υπο-ενότητες 3.4.1 και 3.4.2, στην παραγωγή, δηλαδή, καινούργιων πληροφοριών ή καινούργιου δέντρου στόχων για την απαίτηση.
2. **calculateRequirementTrust:** Στο σημείο αυτό εκκινείται εσωτερικά ο επαναυπολογισμός της τιμής εμπιστοσύνης στην απαίτηση, στα συστατικά της οποίας παρουσιάστηκε μεταβολή. Αναπόσπαστο κομμάτι του βήματος αυτού, όμως, είναι και το επόμενο βήμα.
3. **createGoalModel:** Εδώ δημιουργείται το δέντρο στόχων, ολοκληρωμένο και με συμπληρωμένες τις τιμές στους κόμβους. Τελικά ολοκληρώνεται και το προηγούμενο βήμα και εκτελείται reasoning πάνω στο δέντρο. Έτσι προκύπτει η τιμή της απαίτησης.
4. **update:** Η ανανεωμένη τιμή της απαίτησης γνωστοποιείται στο Blackboard, προκειμένου να γίνει διαθέσιμη σε όποιο άλλο component την έχει ανάγκη.
5. **publish:** Το Blackboard στέλνει την νεοαποκτηθείσα τιμή στο Pub/Sub Module, προκειμένου αυτό να ενημερώσει με την σειρά του όλα τα ενδιαφερόμενα μέλη για το νέο γεγονός.
6. **notifyNewEvent:** Το Pub/Sub Module ενημερώνει τον TrustEvaluator για την αλλαγή στην τιμή μίας, ή περισσότερων, απαιτήσεων.

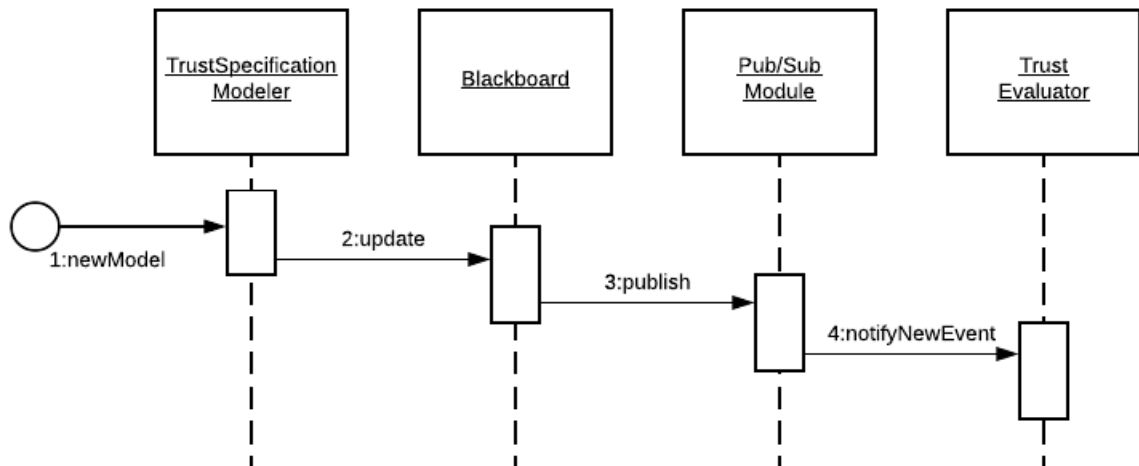


Σχήμα 3.4-3 Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Απαίτησης

3.4.4 Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Γράφου Εκτίμησης Εμπιστοσύνης

Στο διάγραμμα που έπεται περιγράφεται η διαδικασία που ακολουθείται όταν μεταβάλλεται ο γράφος εκτίμησης εμπιστοσύνης. Αυτό ανταποκρίνεται στην αλλαγή ανάμεσα στα επιμέρους χαρακτηριστικά, που ορίζουν την εμπιστοσύνη σε μια υπηρεσία, καθώς επίσης και στην σημαντικότητά τους αλλά και την αλληλεπίδραση μεταξύ τους. Αναλυτικά η διαδικασία είναι η εξής:

1. **newModel:** Το TrustSpecificationModeler λαμβάνει έναν νέο γράφο εκτίμησης εμπιστοσύνης που δημιούργησε ο χρήστης, και συγκεκριμένα το TrustSpecificationEditor.
2. **update:** Ο ανανεωμένος γράφος εκτίμησης εμπιστοσύνης γνωστοποιείται στο Blackboard, προκειμένου να γίνει διαθέσιμη σε όποιο άλλο component την έχει ανάγκη.
3. **publish:** Το Blackboard στέλνει τον καινούργιο γράφο εκτίμησης εμπιστοσύνης στο Pub/Sub Module, προκειμένου αυτό να ενημερώσει με την σειρά του όλα τα ενδιαφερόμενα μέλη για το νέο γεγονός.
4. **notifyNewEvent:** Το Pub/Sub Module ενημερώνει τον TrustEvaluator για την αλλαγή που προέκυψε στον γράφο εκτίμησης εμπιστοσύνης.



Σχήμα 3.4-4 Ακολουθιακό Διάγραμμα Δημιουργίας Νέου Γράφου Εκτίμησης Εμπιστοσύνης

3.4.5 Ακολουθιακό Διάγραμμα Λήξης Χρόνου

Στο διάγραμμα που έπεται περιγράφεται η διαδικασία που ακολουθείται όταν παρέλθει ο, ορισμένος από τον χρήστη, χρόνος. Αναλυτικότερα, όταν περάσει το παράθυρο εγκυρότητας των δεδομένων που έχει ορισθεί, τα δεδομένα, που έχουν παραχθεί, δηλώνονται ως άκυρα και όλη η διαδικασία εκτίμησης της εμπιστοσύνης αρχίζει από την αρχή με την ενημέρωση όλων των μελών που την αποτελούν. Η διαδικασία είναι η εξής:

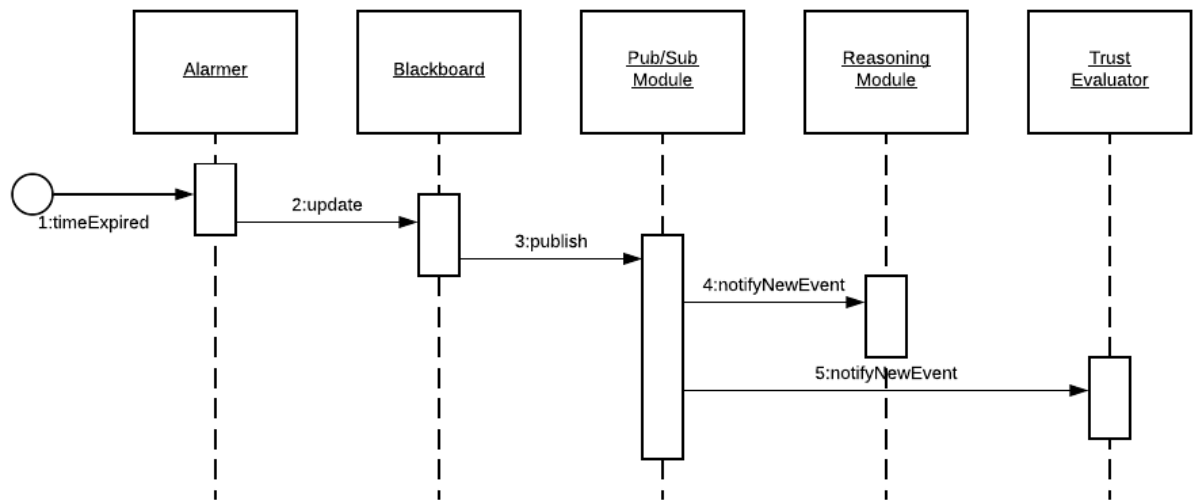
Κεφάλαιο 1: **timeExpired:** Ο Alarmer ενημερώνεται από τον εσωτερικό του χρονομετρητή για το πέρας του χρονικού περιθωρίου που έχει δοθεί για την εγκυρότητα των δεδομένων που έχει αποκτήσει το περιβάλλον-πλαίσιο.

Κεφάλαιο 2: **update:** Η πληροφορία για την ανάγκη ακύρωσης των δεδομένων, λόγω λήξης χρόνου, γράφεται στο Blackboard, προκειμένου να γνωρίζουν όλα τα components ότι οι πληροφορίες που έχουν είναι παρωχημένες.

Κεφάλαιο 3: **publish:** Το Blackboard στέλνει την πληροφορία για την ακύρωση των δεδομένων που υπάρχουν στο Pub/Sub Module, προκειμένου αυτό να ενημερώσει με την σειρά του όλα τα ενδιαφερόμενα μέλη για το νέο γεγονός

Κεφάλαιο 4: **notifyNewEvent:** Το Pub/Sub Module ενημερώνει το ReasoningModule για την ακύρωση των πληροφοριών που έχουν παραχθεί μέχρι στιγμής. Αυτό, όπως, είδαμε και προηγουμένως, οδηγεί στον επαναυπολογισμό της τιμής της κάθε απαίτησης που χρησιμοποιείται στο περιβάλλον-πλαίσιο.

Κεφάλαιο 5: **notifyNewEvent**: Το Pub/Sub Module ενημερώνει τον TrustEvaluator για την ακύρωση των πληροφοριών που έχουν παραχθεί μέχρι στιγμής. Αυτό, όπως, είδαμε και προηγουμένως, οδηγεί στον επαναυπολογισμό της τιμής της εκτίμησης της εμπιστοσύνης που απολαμβάνει η υπηρεσία που ερευνάται από το περιβάλλον-πλαίσιο.



Σχήμα 3.4-5 Ακολουθιακό Διάγραμμα Λήξης Χρόνου

Κεφάλαιο 4: Περιγραφή Διαδικασίας

4.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται το εννοιολογικό μοντέλο (domain model) του προτεινόμενου περιβάλλοντος-πλαισίου για την εκτίμηση της εμπιστοσύνης που απολαμβάνει μια υπηρεσία, η οποία αφορά συγκεκριμένο περιεχόμενο και παρέχεται από συγκεκριμένο πάροχο. Εδώ περιγράφονται όλες οι διακριτές οντότητες που απαρτίζουν το περιβάλλον-πλαίσιο, καθώς επίσης και οι σχέσεις που υπάρχουν μεταξύ τους. Το μοντέλο προδιαγράφεται με βάση το MOF και χρησιμοποιείται για την παραγωγή του πηγαίου κώδικα του περιβάλλοντο-πλαισίου με χρήση του EMF.

4.2 Τυποποίηση συμμετεχόντων μερών

Η εκτίμηση της εμπιστοσύνης σε μια υπηρεσία προϋποθέτει την πλήρη αποσαφήνιση των παραμέτρων που αποτελούν και επηρεάζουν την εκάστοτε υπηρεσία. Ο ορισμός μιας υπηρεσίας, λοιπόν, αποτελείται από κάποια συμμετέχοντα μέρη που χρίζουν τυποποίησης. Τα μέρη αυτά χωρίζονται στις τέσσερις κατηγορίες που αναλύονται στην συνέχεια.

4.2.1 Πάροχος υπηρεσίας (Service Provider)

Κάθε υπηρεσία παρέχεται από μια συγκεκριμένη οντότητα. Η οντότητα αυτή μπορεί να είναι ένα και μοναδικό πληροφοριακό σύστημα, ή πολλά πληροφοριακά συστήματα διασυνδεδεμένα μεταξύ τους. Σε κάποιες περιπτώσεις η οντότητα αυτή μπορεί να αντιπροσωπεύεται και από μια φυσική παρουσία που παρέχει την υπηρεσία που ενδιαφέρει τον χρήστη.

4.2.2 Απαιτήσεις από την υπηρεσία (Requirements)

Ο χρήστης, που ενδιαφέρεται για μια παρεχόμενη υπηρεσία, δηλώνει τις απαιτήσεις που έχει από την υπηρεσία αυτή. Ορίζει, δηλαδή, τους τομείς τους οποίους θεωρεί, περισσότερο ή λιγότερο, ανάλογα με την περίπτωση, σημαντικούς για την εμπιστοσύνη της ελεγχόμενης υπηρεσίας. Κάθε χρήστης μπορεί να έχει διαφορετικές απαιτήσεις από

την ίδια υπηρεσία, όπως επίσης και διαφορετικούς βαθμούς σημαντικότητας για την εκαστοτε απαίτηση. Όλα αυτά θεωρούνται παραμετροποιήσιμα με βάση τις ανάγκες του χρήστη.

4.2.3 Χρονικό πλαίσιο (Time Frame)

Η εμπιστοσύνη παροχής μιας υπηρεσίας έχει νόημα εκτίμησης μόνο μέσα σε κάποιο συγκεκριμένο χρονικό πλαίσιο. Αποτελεί, λοιπόν, κομμάτι του ορισμού της εμπιστοσύνης ο χρόνος που δίνεται σε μια υπηρεσία για να παραδώσει το αποτέλεσμα που ζητάει ο χρήστης. Ο χρόνος αυτός εξαρτάται από την φύση της εκάστοτε υπηρεσίας μεν, και από τις συγκεκριμενοποιημένες ανάγκες του χρήστη δε. Για τους λόγους αυτούς, το χρονικό πλαίσιο θεωρείται, επίσης, παραμετροποιήσιμο.

4.2.4 Περιεχόμενο (Context)

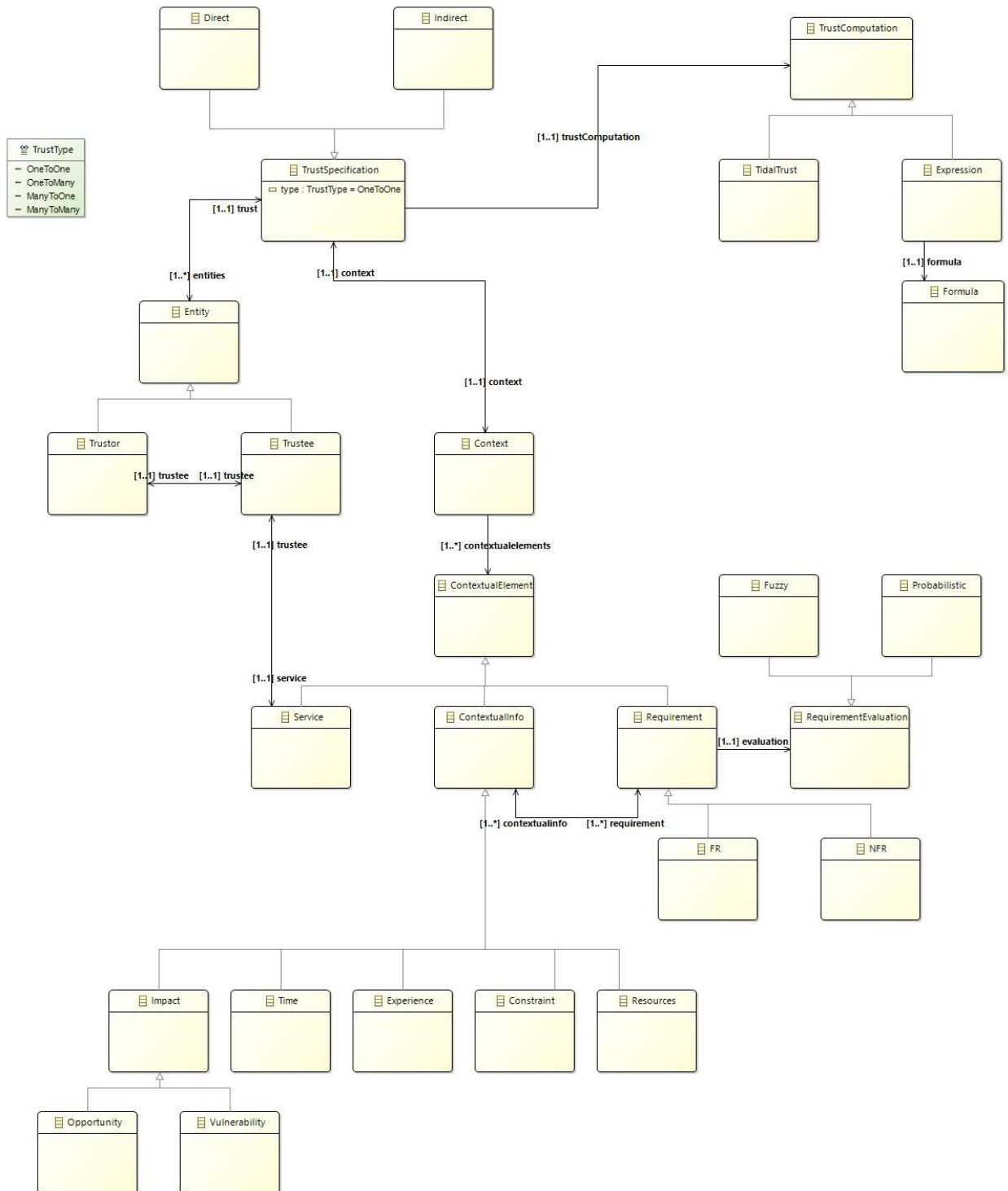
Σημαντικός παράγοντας, στην εκτίμηση της εμπιστοσύνης στην παροχή μιας υπηρεσίας, αποτελεί και το περιεχόμενο, το οποίο καλείται να χειριστεί η υπηρεσία. Εξ' ορισμού, κάποιες υπηρεσίες και το περιεχόμενο που χειρίζονται έχουν μεγαλύτερη επίπτωση στο περιβάλλον τους. Για παράδειγμα, υπηρεσίες που περιλαμβάνουν την διεξαγωγή τραπεζικών συναλλαγών πρέπει να εκτιμώνται με μεγαλύτερη αυστηρότητα από το περιβάλλον-πλαίσιο, λόγω της αυξημένης σημαντικότητας του περιεχομένου που χειρίζονται.

4.3 Εννοιολογικό Μοντέλο (Domain model)

Το εννοιολογικό μοντέλο του περιβάλλοντος-πλαισίου για την εκτίμηση της εμπιστοσύνης μιας παρεχόμενης υπηρεσίας παρουσιάζεται στο διάγραμμα που βρίσκεται στην επόμενη σελίδα. Το μοντέλο αυτό δίνει την δυνατότητα στον χρήστη να ορίζει πλήρως όλα τα συμμετέχοντα μέρη που ορίστηκαν προηγουμένως.

Πιο συγκεκριμένα, επιτρέπει τον ορισμό των απαιτήσεων με βάση τα επιμέρους συστατικά/πληροφορίες και τις μεταξύ τους αλληλεπιδράσεις, με την μορφή δέντρων στόχων. Επιπλέον, δηλώνεται ο πάροχος, το χρονικό πλαίσιο μέσα στο οποίο εκτιμάται και το περιεχόμενο που χειρίζεται η υπηρεσία. Τέλος, ο χρήστης μπορεί να ορίσει την σημαντικότητα των απαιτήσεων στην εκτίμηση της εμπιστοσύνης, καθώς επίσης και τις μεταξύ τους σχέσεις αλλά και τον αλγόριθμο που θα χρησιμοποιηθεί για την τελική αποτίμηση.

Στην συνέχεια παρέχεται το διάγραμμα και ακολούθως περιγράφονται αναλυτικά οι κλάσεις του μοντέλου προκειμένου να κατανοηθεί πλήρως η λειτουργία τους και η συμβολή τους στην λειτουργία του περιβάλλοντος-πλαισίου.



Σχήμα 4.3-1 Διάγραμμα Κλάσεων του Εννοιολογικού Μοντέλου

4.3.1 Κλάση Entity

Η κλάση αυτή αντιπροσωπεύει κάθε οντότητα που αποτελεί πάροχο ή αποδέκτη μιας υπηρεσίας. Περιέχει την ιδιότητα *name* που χρησιμοποιείται για την αποθήκευση του διακριτικού ονόματος της οντότητας. Η κλάση αυτή λειτουργεί ως βασική κλάση για τις δύο υπο-κλάσεις που την επεκτείνουν. Οι υπο-κλάσεις αυτές είναι:

- Trustor: Η κλάση αυτή χρησιμοποιείται για την αναπαράσταση του χρήστη που ενδιαφέρεται για την υπηρεσία που εκτιμάται
- Trustee: Η κλάση αυτή χρησιμοποιείται για την αναπαράσταση του παρόχου της υπηρεσίας που εκτιμάται

4.3.2 Κλάση TrustSpecification

Η κλάση αυτή αντιπροσωπεύει τον ορισμό, με βάση τα συμμετέχοντα μέρη, της εμπιστοσύνης σε μια υπηρεσία. Αποτελεί την βασική κλάση του περιβάλλοντος-πλαισίου και συνδέεται με τις κλάσεις που αντιπροσωπεύουν τα συστατικά του ορισμού της εμπιστοσύνης. Συγκεκριμένα, η TrustSpecification είναι άρρηκτα συνδεδεμένη με ένα και μόνο περιεχόμενο, γι' αυτό και συνδέεται με πολλαπλότητα 1 με την κλάση Context. Αντίστοιχα, η εμπιστοσύνη υπολογίζεται με έναν και μόνο τρόπο κάθε φορά, πράγμα που δικαιολογεί και την σύνδεση με πολλαπλότητα 1 με την κλάση TrustComputation. Τέλος, η υπηρεσία μπορεί να παρέχεται από έναν ή περισσότερους παρόχους και μπορεί να ενδιαφέρονται για αυτήν ένας ή περισσότεροι χρήστες. Έτσι η κλάση TrustSpecification συνδέεται με την κλάση Entity με πολλαπλότητα 1..*. Με το σκεπτικό μελλοντικής επέκτασης του περιβάλλοντος-πλαισίου, η κλάση TrustSpecification επεκτείνεται από δύο άλλες υπο-κλάσεις. Οι υπο-κλάσεις αυτές είναι η Direct και η Indirect, που αντιπροσωπεύουν το είδος του ορισμού και το εξειδικεύουν ανάλογα με τον αν πρόκειται για εκτίμηση που αφορά άμεση παροχή υπηρεσίας από κάποιον πάροχο ή με την παρεμβολή μεσάζοντα, αντίστοιχα.

4.3.3 Κλάση TrustComputation

Η κλάση TrustComputation αντιπροσωπεύει τον τρόπο με τον οποίο υπολογίζεται η τελική εμπιστοσύνη στην υπηρεσία που παρέχεται, με το συγκεκριμένο περιεχόμενο που δίνεται. Πρακτικά, αποτελεί κλάση-βάση για την υλοποίηση των διάφορων εναλλακτικών αλγορίθμων που είναι διαθέσιμοι στο περιβάλλον-πλαίσιο, για τον υπολογισμό της εμπιστοσύνης της υπηρεσίας. Υλοποιεί την μέθοδο `computeTrust()`, που καλείται από την TrustSpecification, όταν όλα τα απαραίτητα δεδομένα είναι διαθέσιμα.

4.3.4 Κλάση TidalTrust

Η κλάση `TidalTrust` αποτελεί υπο-κλάση της `TrustComputation` και αντιπροσωπεύει την υλοποίηση του προεπιλεγμένου αλγόριθμου του περιβάλλοντος-πλαίσιου για την εκτίμηση της εμπιστοσύνης σε μια υπηρεσία. Ο αλγόριθμος αυτός είναι ο `TidalTrust` που περιγράφεται διεξοδικά στο κεφάλαιο 5. Η κλάση κάνει `override` και παρέχει ανανεωμένη την μέθοδο `computeTrust()`, προκειμένου να η ανεξαρτησία υλοποίησης από την επιλογή κάποιου αλγορίθμου.

4.3.5 Κλάση `Expression`

Η κλάση `Expression` αποτελεί υπο-κλάση της `TrustComputation` και αντιπροσωπεύει την υλοποίηση αλγορίθμου εκτίμησης της εμπιστοσύνης σε μια υπηρεσία, που στηρίζεται στον ορισμό μέσω λογικών εκφράσεων. Η κλάση αυτή κάνει επίσης `override` την μέθοδο `computeTrust()`, μέσω της οποίας παρέχει το αποτέλεσμα που υπολογίζει.

4.3.6 Κλάση `Context`

Η κλάση `Context` αντιπροσωπεύει το γενικότερο περιεχόμενο που αφορά τον ορισμό της εμπιστοσύνης στην υπηρεσία. Το περιεχόμενο αυτό αφορά αφ' ενός τις απαιτήσεις που έχει ορίσει ο χρήστης ότι έχει από την υπηρεσία, καθώς επίσης και την σημαντικότητα της κάθε μιας. Αφ' ετέρου, όμως, περιεχόμενο θεωρούνται και οι επιμέρους πληροφορίες/συστατικά που απαρτίζουν, και κατ'επέκταση ορίζουν, την κάθε απαίτηση ξεχωριστά. Επιπλέον, κομμάτι το περιεχομένου είναι η ίδια η υπηρεσία που παρέχεται. Επειδή, λοιπόν, οι κλάσεις `Service`, `ContextualInfo` και `Requirement` είναι υπο-κλάσεις της κλάσης `ContextualElement`, η κλάση `Context` συνδέεται με πολλαπλότητα 1..* με την κλάση `ContextualElement`. Συνδέεται, επίσης, με πολλαπλότητα 1..1 με την κλάση `EventHandler`, προκειμένου να έχει ενημέρωση σχετικά με τις αλλαγές που προκύπτουν στα συστατικά του περιεχομένου.

4.3.7 Κλάση `ContextualElement`

Η κλάση `ContextualElement` είναι μια κλάση βάση για όλα τα στοιχεία που δύναται να αποτελούν κομμάτι του περιεχομένου, που αφορά τον ορισμό της εμπιστοσύνης σε μια υπηρεσία. Οτιδήποτε, λοιπόν, είναι στοιχείο του περιεχομένου θεωρείται στιγμιότυπο της κλάσης `ContextualElement`. Η κλάση αυτή, βέβαια, δεν χρησιμοποιείται ποτέ απευθείας αλλά χρησιμεύει για τον ορισμό υπο-κλάσεων, οι οποίες αντιπροσωπεύουν τα εκάστοτε στοιχεία του περιεχομένου με μεγαλύτερη λεπτομέρεια. Υπο-κλάσεις της κλάσης `ContextualElement` είναι οι κλάσεις `Service`, `ContextualInfo` και `Requirement`.

4.3.8 Κλάση Service

Η κλάση Service αποτελεί μια εκ των υπο-κλάσεων της κλάσης ContextualElement και αντιπροσωπεύει την υπηρεσία υπό εξέταση από το περιβάλλον-πλαίσιο. Περιέχει δύο ιδιότητες, οι οποίες χρησιμοποιούνται για αναλυτικότερη περιγραφή της υπηρεσίας. Η πρώτη ιδιότητα είναι η *name*, στην οποία περιέχεται ένα συνοπτικό όνομα, που είναι αντιπροσωπευτικό της υπηρεσίας, και η δεύτερη είναι η *description*, όπου περιέχεται, προαιρετικά, μια εκτενέστερη περιγραφή, σχετική με την υπηρεσία της οποίας η εμπιστοσύνη ερευνάται. Τέλος, η κλάση αυτή συνδέεται με την Trustee με πολλαπλότητα 1..1, προκειμένου να είναι εύκολα προσβάσιμος ο πάροχος της υπηρεσίας.

4.3.9 Κλάση ContextualInfo

Η κλάση ContextualInfo αποτελεί μια εκ των υπο-κλάσεων της κλάσης ContextualElement και αντιπροσωπεύει όλα τα συστατικά/πληροφορίες, από τα οποία εξαρτάται η κάθε απαίτηση, την οποία έχει ο χρήστης και συμβάλλει στον υπολογισμό της τελικής εμπιστοσύνης της υπηρεσίας. Η κλάση, λοιπόν, αυτή περιέχει μια ιδιότητα με το όνομα *value*, που περιέχει την τιμή της συγκεκριμένης πληροφορίας, η οποία έχει προέλθει από εξωτερικές πηγές. Η τιμή της ιδιότητας αυτής ανανεώνεται με την υλοποίηση της μεθόδου setValue() και μέσω αυτής ενημερώνεται για την αλλαγή και ο EventHandler, με τον οποίο συνδέεται το περιεχόμενο της συγκεκριμένης υπηρεσίας. Η κλάση ContextualInfo χρησιμοποιείται για τις γενικές περιπτώσεις, στις οποίες η τιμή της πληροφορίας αποκτάται από μια συγκεκριμένη πηγή, όπως π.χ. κάποιο αρχείο. Χρησιμοποιείται, όμως, και σαν κλάση βάση για πιο εξειδικευμένες περιπτώσεις. Έτσι κάθε ανάγκη, για δημιουργία καινούργιας κλάσης που να αντιπροσωπεύει κάποιο χαρακτηριστικό, υλοποιείται με υπο-κλάση της κλάσης ContextualInfo. Για τις ανάγκες τις διπλωματικής έχουν προβλεφθεί κάποιες τέτοιες ανάγκες, χωρίς, όμως, αυτό να σημαίνει ότι δεν υπάρχει η δυνατότητα για προσθήκη επιπλέον επιλογών. Περιπτώσεις τέτοιων υπο-κλάσεων είναι οι κλάσεις Impact, Experience, Constraint, Time και Resources. Κάθε κλάση ContextualInfo, ή οποιαδήποτε υπο-κλάση της, μπορεί να αφορά μια ή περισσότερες απαιτήσεις και επομένως συνδέεται με την κλάση Requirement με πολλαπλότητα 1..*.

4.3.10 Κλάση Requirement

Η κλάση Requirement αντιπροσωπεύει την κάθε απαίτηση, που έχει ο χρήστης από την υπηρεσία, της οποίας η εμπιστοσύνη αποτιμάται από το περιβάλλον-πλαίσιο. Η κλάση αυτή έχει μια, και μόνο, ιδιότητα που ονομάζεται *name* και χρησιμοποιείται καθαρά για αναφορικούς λόγους. Σημαντική σε αυτή την περίπτωση είναι η μέθοδος evaluate(), που

υλοποιείται από την κλάση και είναι υπεύθυνη για τον υπολογισμό της εμπιστοσύνης που απολαμβάνει η κάθε απαίτηση ξεχωριστά. Προκειμένου να επιτευχθεί αυτός ο υπολογισμός, η κλάση συνδέεται με πολλαπλότητα 1..1 με την κλάση RequirementEvaluation. Η κλάση Requirement εξαρτάται για τον υπολογισμό της τιμής της από μια, ή περισσότερες, πληροφορίες. Έτσι έχουμε σύνδεση με πολλαπλότητα 1..* με την κλάση ContextualInfo.

4.3.11 Κλάση RequirementEvaluation

Η κλάση RequirementEvaluation αντιπροσωπεύει τον υπολογισμό που διεξάγεται για την εύρεση της εμπιστοσύνης που απολαμβάνει η κάθε απαίτηση ξεχωριστά. Συγκεκριμένα, η κλάση υλοποιεί την μέθοδο evaluate(), η οποία, δοθέντος του δέντρου στόχων και των τιμών που αντιστοιχούν στις πληροφορίες, χρησιμοποιεί συλλογιστική και υπολογίζει τις τιμές όλων των κόμβων του δέντρου. Τέλος, επιστρέφει την τιμή της ρίζας του δέντρου, η οποία αντιστοιχεί στην εμπιστοσύνη που έχει υπολογιστεί για την εκάστοτε απαίτηση. Διαφορετικές υπο-κλάσεις της κλάσης RequirementEvaluation έχουν υλοποιηθεί με μοναδική διαφοροποίηση την μορφή επιστροφής της τιμής εμπιστοσύνης. Οι υπο-κλάσεις είναι οι ακόλουθες:

- Probabilistic: Η υπο-κλάση αυτή επιστρέφει την τιμή σε μορφή πιθανοτικού ποσοστού
- Fuzzy: Η υπο-κλάση αυτή επιστρέφει ένα ζεύγος τιμών (low, high) που δείχνει τα όρια ανάμεσα στα οποία μπορεί να κυμαίνεται η τιμή της εμπιστοσύνης που υπολογίστηκε.

4.3.12 Κλάση EventHandler

Η κλάση EventHandler χρησιμοποιείται για την ανιχνεύση πιθανών αλλαγών στα δεδομένα του προβλήματος και την ειδοποίηση συγκεκριμένων μερών, προκειμένου να γίνει ο απαιτούμενος επαναυπολογισμός, όπου αυτό κρίνεται απαραίτητο. Είναι, στην ουσία, ένας Listener γεγονότων που αφορούν αλλαγές σε τιμές πληροφοριών, δέντρα στόχων και γράφων υπολογισμού τελικής εμπιστοσύνης, αλλά και περιπτώσεις λήξης χρόνου ισχύς των δεδομένων (timeouts).

4.3.13 Κλάση Impact

Η κλάση Impact είναι μια υπο-κλάση της κλάσης ContextualInfo. Αντιπροσωπεύει πληροφορίες που αφορούν τον αντίκτυπο που έχει μια υπηρεσία στον χρήστη που την χρειάζεται, όσον αφορά την κάθε απαίτηση που ορίζει. Πιο συγκεκριμένα, μια κλάση

Impact προσθέτει γνώση σχετικά με την πιθανότητα αποτυχίας μιας υπηρεσίας να παραδώσει τα αποτελέσματα που υπόσχεται. Προφανώς, υπηρεσίες με μεγαλύτερο αντίκτυπο αντιμετωπίζουν αυστηρότερα κριτήρια υπολογισμού της εμπιστοσύνης της απαίτησης, την οποία αφορούν. Επί παραδείγματι, οι τραπεζικές συναλλαγές έχουν μεγάλο αντίκτυπο στην απαίτηση της ασφάλειας. Η κλάση αυτή υλοποιεί την μέθοδο `setValue()`, κάνοντας `override` την αντίστοιχη μέθοδο της κλάσης-βάσης `ContextualInfo`.

4.3.14 Κλάση Experience

Η κλάση `Experience` είναι μια υπο-κλάση της κλάσης `ContextualInfo`. Αντιπροσωπεύει την γνώση που προσφέρουν προηγούμενοι χρήστες της υπηρεσίας. Η πληροφορίες αυτές μπορεί να αφορούν κάποιο συγκεκριμένο χαρακτηριστικό της υπηρεσίας, ή αντιστοιχα απαίτηση που ενδιαφέρει τον χρήστη της, ή μπορεί να κρίνουν την υπηρεσία σαν ολότητα. Τα δεδομένα μπορεί να λαμβάνονται από κάποια εξωτερική πηγή, ή μπορεί μελλοντικά να το περιβάλλον-πλαίσιο να διατηρεί κάποια εσωτερική βάση δεδομένων, η οποία θα αφορά υπηρεσίες που έχουν εκτιμηθεί στο παρελθόν. Η κλάση αυτή υλοποιεί την μέθοδο `setValue()`, κάνοντας `override` την αντίστοιχη μέθοδο της κλάσης-βάσης `ContextualInfo`.

4.3.15 Κλάση Constraint

Η κλάση `Constraint` είναι μια υπο-κλάση της κλάσης `ContextualInfo`. Αντιπροσωπεύει πιθανούς περιορισμούς που πρέπει να ικανοποιούνται. Οι περιορισμοί αυτοί αναπαριστούνται στο δέντρο στόχων με την μορφή κόμβων. Οι κόμβοι αυτοί αποτελούν *hard goals* με `--D(a,b)` συνεισφορά, όπως αναπτύχθηκε εκτενέστερα στο Κεφάλαιο 2 και συγκεκριμένα στην ενότητα 2.3. Η κλάση αυτή υλοποιεί την μέθοδο `setValue()`, κάνοντας `override` την αντίστοιχη μέθοδο της κλάσης-βάσης `ContextualInfo`.

4.3.16 Κλάση Time

Η κλάση `Time` είναι μια υπο-κλάση της κλάσης `ContextualInfo`. Αντιπροσωπεύει το χρονικό διάστημα κατά το οποίο οι πληροφορίες και οι εκτιμήσεις, που έχουν υπολογιστεί, βρίσκονται σε ισχύ. Χρησιμοποιείται για την ειδοποίηση του περιβάλλοντος-πλαισίου σχετικά με την λήξη του, επιλεγμένου από τον χρήστη, χρονικού περιθωρίου. Περιέχει μια ιδιότητα, που ονομάζεται *delay* και περιέχει το χρονικό διαστήμα που ισχύουν τα δεδομένα. Υλοποιεί, επιπλέον, την μέθοδο `start()`, η οποία ενεργοποιεί έναν timer, επιφορτισμένο με την υποχρέωση ειδοποίησης του περιβάλλοντος-πλαισίου όταν ο χρόνος παρέλθει.

4.3.17 Κλάση Resources

Η κλάση Resources είναι μια υπο-κλάση της κλάσης ContextualInfo. Αντιπροσωπεύει τους πόρους που χρησιμοποιεί ο εκάστοτε πάροχος, προκειμένου να προσφέρει την υπηρεσία υπο εξέταση. Αυτοί μπορεί να είναι φυσικοί πόροι, όπως υπολογιστικά συστήματα, κτιριακές εγκαταστάσεις κ.α., ή μπορεί να είναι υπηρεσίες παρεχόμενες από τρίτους. Σε κάθε περίπτωση, η σημασία στην εκτίμηση κάθε απαίτησης είναι παραπάνω από εμφανής. Η κλάση αυτή υλοποιεί την μέθοδο setValue(), κάνοντας override την αντίστοιχη μέθοδο της κλάσης-βάσης ContextualInfo.

4.4 Τυποποίηση λειτουργικότητας

Στην ενότητα αυτή παρουσιάζεται η λειτουργικότητα του συστήματος, όπως αναπτύχθηκε τελικά σε κώδικα με την βοήθεια του EMF. Συγκεκριμένα, περιγράφεται, με την χρήση διαγραμμάτων δραστηριοτήτων, τα ακριβή βήματα που ακολουθούνται σε όλες τις περιπτώσεις που δύναται να προκύψουν κατά την χρησιμοποίηση του περιβάλλοντος-πλαισίου. Τέλος, παρέχονται και τα αντίστοιχα ακολουθιακά διαγράμματα, προκειμένου να διαφανούν πλήρως οι αλληλεπιδράσεις μεταξύ των κλάσεων που ορίζονται με την χρήση του εννοιολογικού μοντέλου που δόθηκε στην προηγούμενη ενότητα.

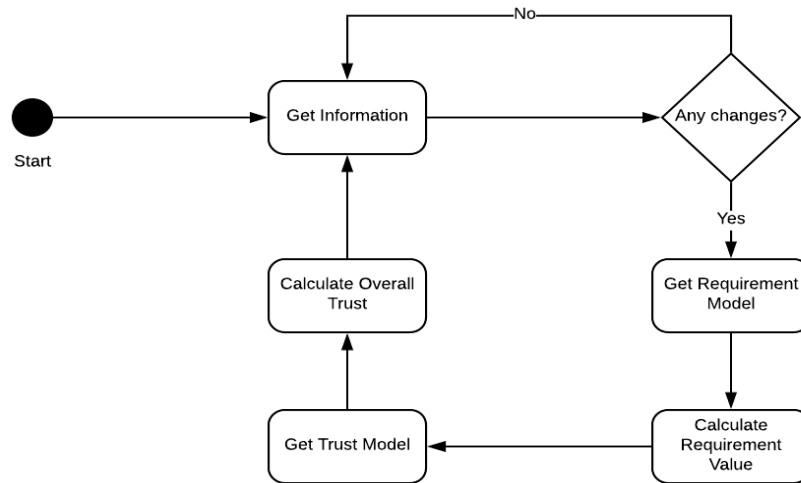
4.4.1 Διαγράμματα δραστηριοτήτων

Τα διαγράμματα δραστηριοτήτων περιγράφουν την σειρά των ενεργειών που πραγματοποιούνται σε όλες τις περιπτώσεις μεταβολών σε πληροφορίες, μοντέλα ή ορισμούς που χρησιμοποιούνται για την εκτίμηση της εμπιστοσύνης που απολαμβάνει μια υπηρεσία που ερευνάται.

Η εκτίμηση της εμπιστοσύνης, βεβαίως, είναι σε γενικές γραμμές μια σειριακή διαδικασία. Παρατηρούμε, λοιπόν, ότι τα διαγράμματα μοιάζουν αρκετά, ειδικά όσον αφορά τα τελευταία βήματα, που αφορούν τον υπολογισμό της εμπιστοσύνης. Οι διαφοροποιήσεις μεταξύ τους βρίσκονται στις αρχικές ενέργειες, που τελικά πυροδοτούν την υπόλοιπη διαδικασία.

4.4.1.1 Διάγραμμα Δραστηριότητας Δημιουργίας Νέας Πληροφορίας

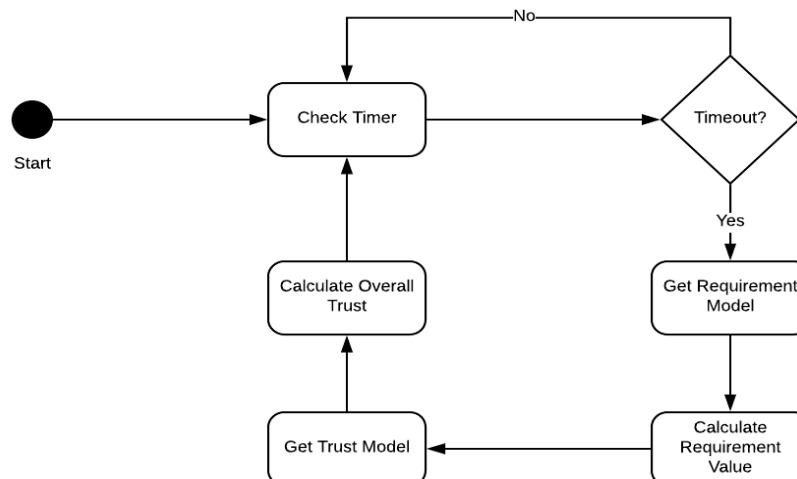
Σε αυτή την περίπτωση το σύστημα εξετάζει τις πληροφορίες, που παρέχονται από εξωτερικές πηγές, και αφορούν τις απαιτήσεις που έχει θέσει ο χρήστης για την υπηρεσία που τον ενδιαφέρει. Αν παρατηρηθεί κάποια αλλαγή, πυροδοτείται η διαδικασία που αναφέρθηκε και εκτελούνται σειριακά οι ενέργειες, που αφορούν την δημιουργία του goal model και τον υπολογισμό της εμπιστοσύνης, για την απαίτηση που μας αφορά, και στην συνέχεια, την δημιουργία του μοντέλου και τον υπολογισμό της εκτίμησης για την συνολική εμπιστοσύνη στην υπηρεσία.



Σχήμα 4.4-1 Διάγραμμα Δραστηριότητας Δημιουργίας Νέας Πληροφορίας

4.4.1.2 Διάγραμμα Δραστηριότητας Λήξης Χρονικού Πλαισίου

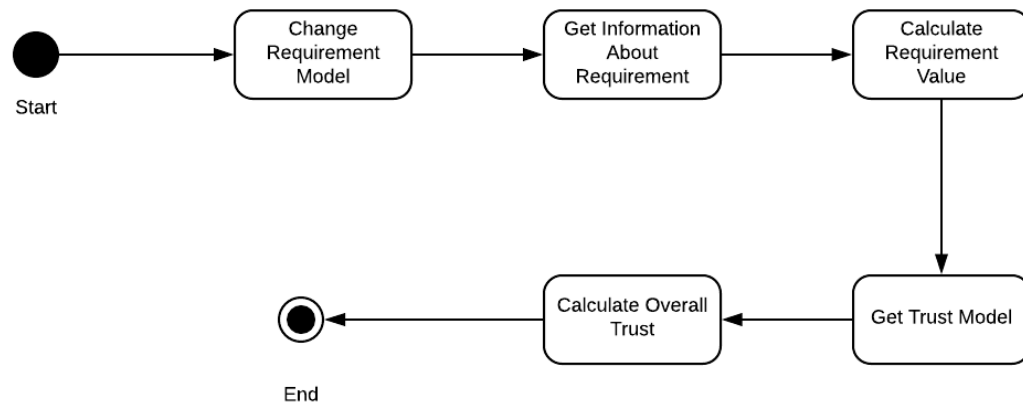
Σε αυτή την περίπτωση, όταν το χρονικό πλαίσιο, το οποίο έχει οριστεί από τον χρήστη, λήξει, το περιβάλλον-πλαίσιο ενημερώνεται και, σε πλήρη αντιστοιχία με την προηγούμενη περίπτωση, πυροδοτείται η διαδικασία που καταλήγει στον επαναυπολογισμό της εκτίμησης για την συνολική εμπιστοσύνη στην υπηρεσία που μας ενδιαφέρει.



Σχήμα 4.4-2 Διάγραμμα Δραστηριότητας Λήξης Χρονικού Πλαισίου

4.4.1.3 Διάγραμμα Δραστηριότητας Αλλαγής Μοντέλου Απαίτησης

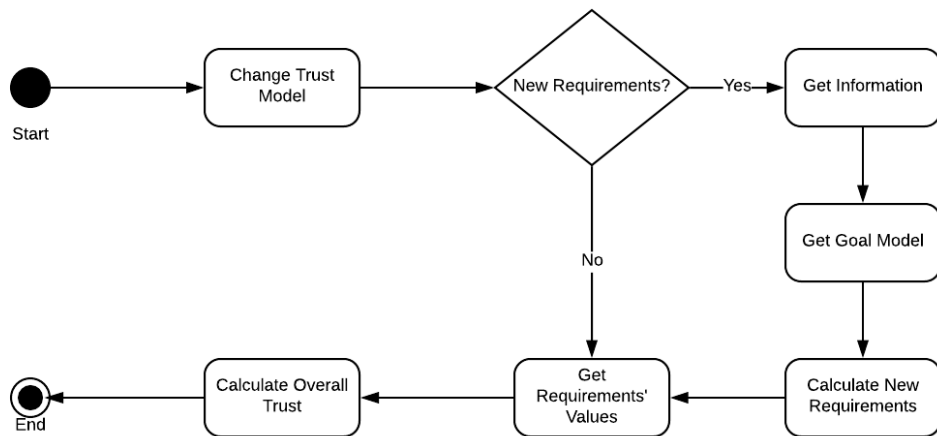
Σε αυτή την περίπτωση, ο χρήστης μεταβάλλει το goal model μιας εκ των απαιτήσεων που αποτελούν την υπηρεσία. Αυτό μπορεί να γίνει είτε προσθέτοντας μια καινούργια πληροφορία, από την οποία εξαρτάται ο υπολογισμός της εμπιστοσύνης της απαίτησης, ή αλλάζοντας την δομή και τις αλληλεξαρτήσεις στο δέντρο στόχων, που περιγράφουν την δεδομένη απαίτηση. Με την ανίχνευση της αλλαγής το σύστημα εκκινεί την διαδικασία επαναυπολογισμού της εκτίμησης της συνολικής εμπιστοσύνης στην υπηρεσία, όπως περιγράφηκε και προηγουμένως.



Σχήμα 4.4-3 Διάγραμμα Δραστηριότητας Αλλαγής Μοντέλου Απαίτησης

4.4.1.4 Διάγραμμα Δραστηριότητας Αλλαγής Μοντέλου Εκτίμησης Εμπιστοσύνης

Σε αυτή την περίπτωση, ο χρήστης μεταβάλλει το μοντέλο εκτίμησης της εμπιστοσύνης στην υπηρεσία. Αυτή η ενέργεια αφορά είτε στην προσθήκη νέας απαίτησης στον ορισμό της εμπιστοσύνης, ή στην μεταβολή του γράφου εκτίμησης της εμπιστοσύνης, που πρακτικά σημαίνει αλλαγή των εξαρτήσεων ανάμεσα στις απαιτήσεις που έχει θέσει ο χρήστης για την συγκεκριμένη υπηρεσία.



Σχήμα 4.4-4 Διάγραμμα Δραστηριότητας Αλλαγής Μοντέλου Εκτίμησης Εμπιστοσύνης

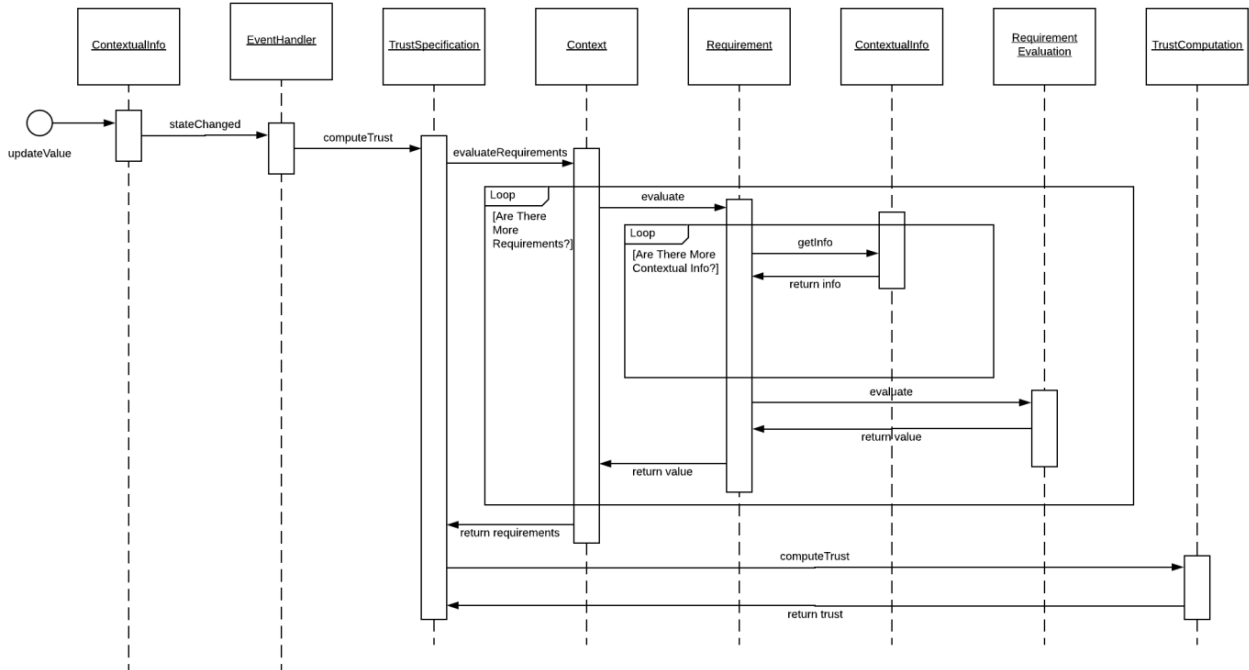
4.4.2 Ακολουθιακά διαγράμματα

Τα ακολουθιακά διαγράμματα περιγράφουν την αλληλουχία κλήσεων μεθόδων, που γίνονται κατά τις περιπτώσεις μεταβολών στις πληροφορίες, μοντέλα ή ορισμούς που χρησιμοποιούνται για την εκτίμηση της εμπιστοσύνης που απολαμβάνει μια υπηρεσία που ερευνάται.

Τα διάγραμματα που δίνονται σε αυτή την ενότητα ανταποκρίνονται στα αντίστοιχα διαγράμματα που δόθηκαν στην προηγούμενη υπο-ενότητα. Αντίστοιχα, λοιπόν, μεγάλο κομμάτι της διαδικασίας είναι όμοιο σε όλες τις περιπτώσεις που παρουσιάζονται.

4.4.2.1 Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Πληροφορίας

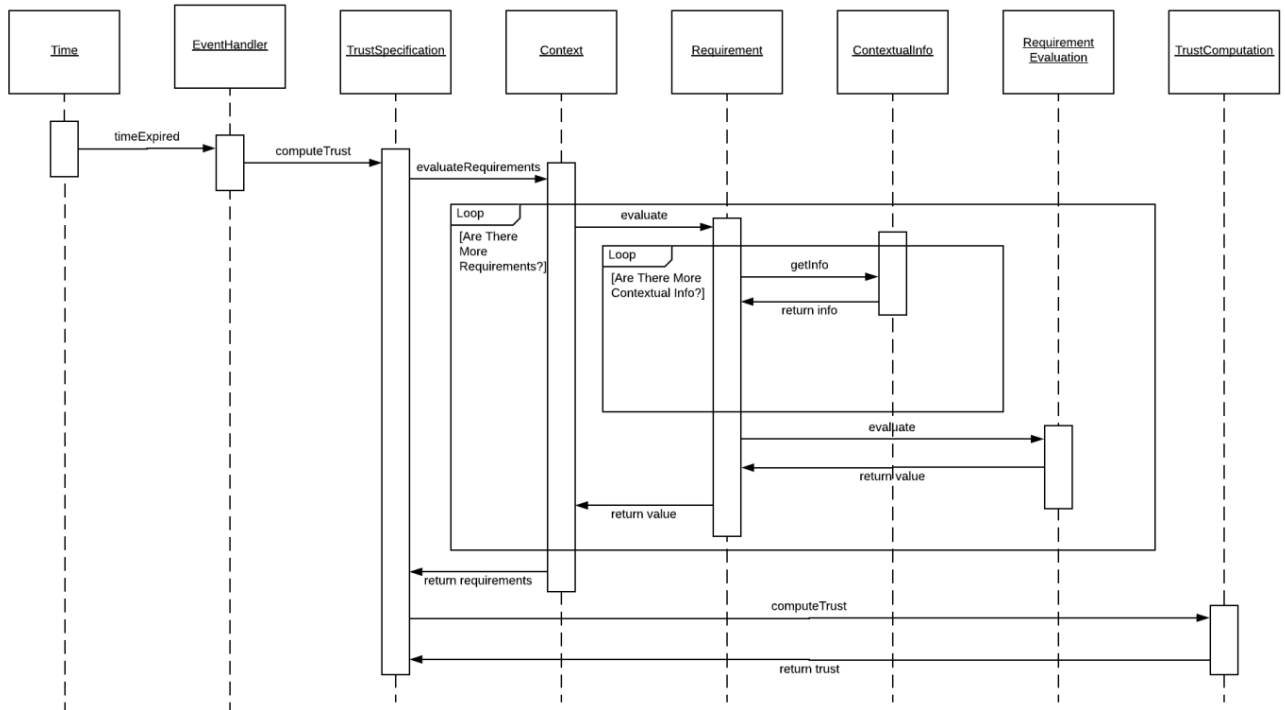
Σε αυτή την περίπτωση, παρατηρείται κάποια αλλαγή στις πληροφορίες, που αφορούν την εμπιστοσύνη που απολαμβάνει μια συγκεκριμένη απαίτηση. Αμέσως, λοιπόν, εκκινείται η αλληλουχία κλήσεων στις αντίστοιχες κλάσεις, προκειμένου να επαναυπολογιστεί η εκτίμηση της εμπιστοσύνης στην υπηρεσία που ερευνάται.



Σχήμα 4.4-5 Ακολουθιακό Διάγραμμα Δημιουργίας Νέας Πληροφορίας

4.4.2.2 Ακολουθιακό Διάγραμμα Λήξης Χρονικού Πλαισίου

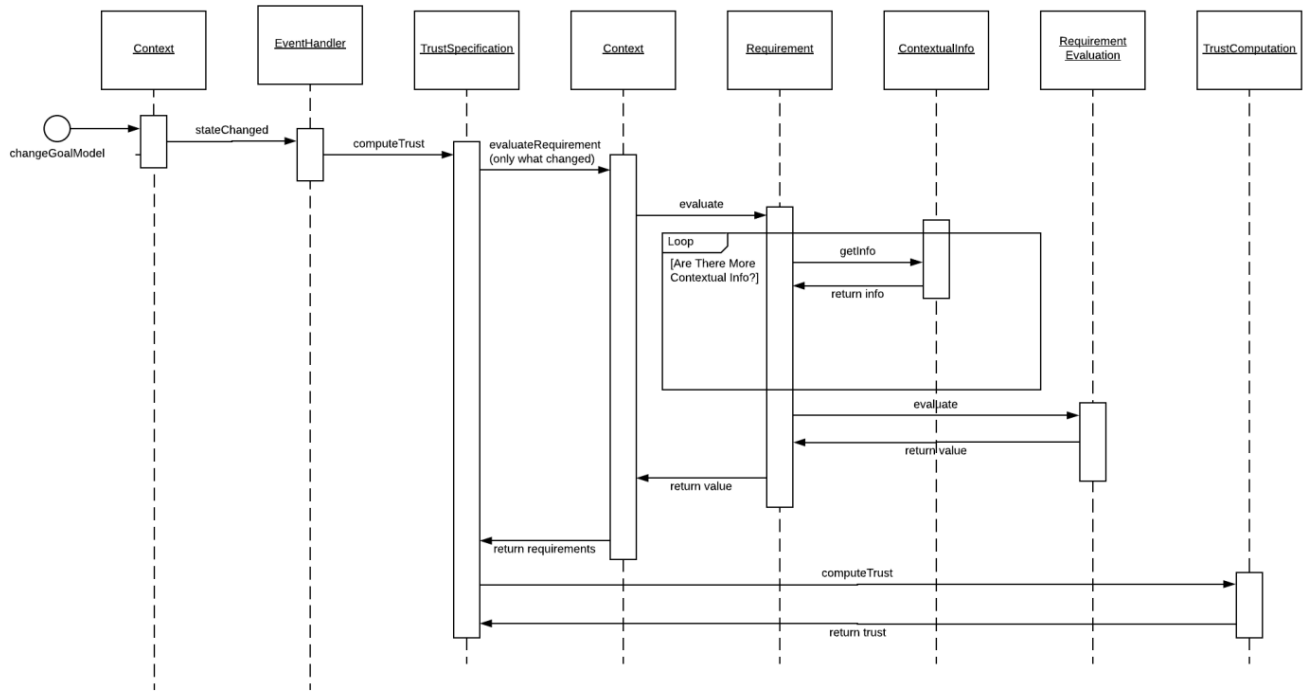
Σε αυτή την περίπτωση, παρέρχεται το χρονικό πλαίσιο ισχύος των παρεχόμενων πληροφοριών. Όταν αυτό συμβεί, τότε κρίνεται απαραίτητος από το περιβάλλον-πλαίσιο ο επαναυπολογισμός της εκτίμησης της εμπιστοσύνης στην υπηρεσία που ερευνάται. Έτσι πυροδοτείται η διαδικασία που εκτελεί όλες τις απαραίτητες διεργασίες προκειμένου να γίνει ο υπολογισμός αυτός.



Σχήμα 4.4-6 Ακολουθιακό Διάγραμμα Λήξης Χρονικού Πλαισίου

4.4.2.3 Ακολουθιακό Διάγραμμα Αλλαγής Μοντέλου Απαίτησης

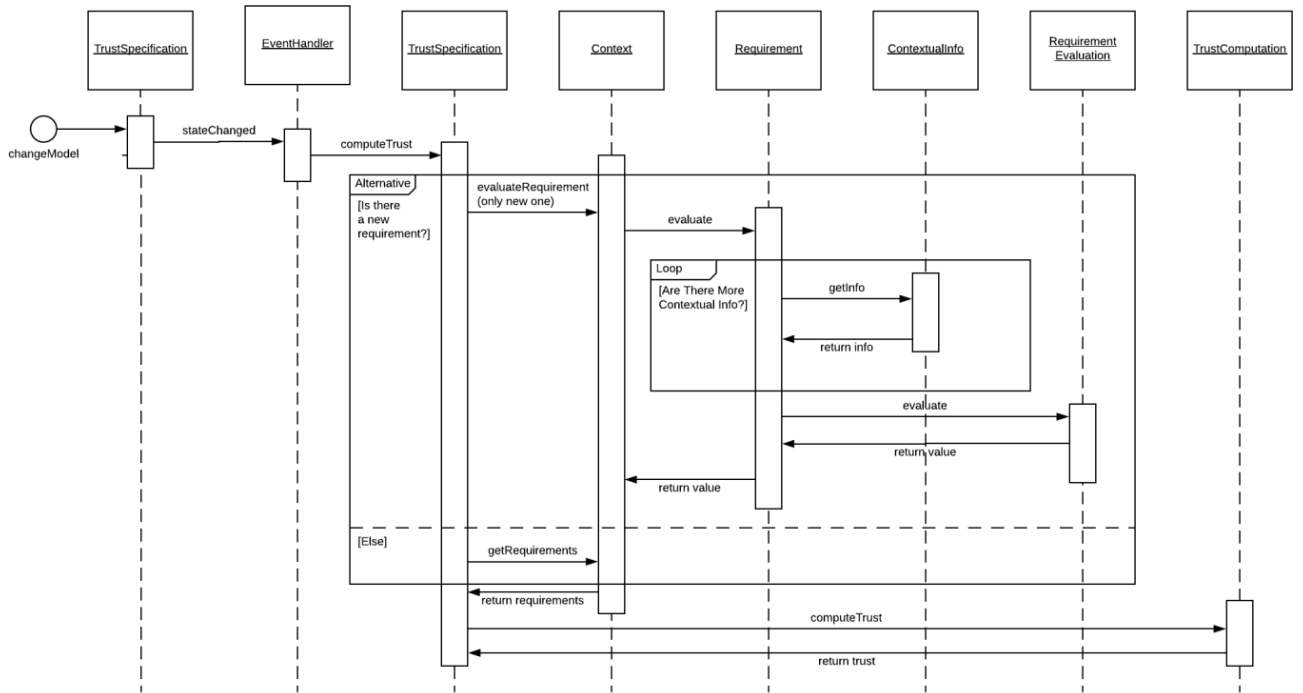
Σε αυτή την περίπτωση, ο χρήστης μεταβάλλει το μοντέλο που αντιστοιχεί σε μια εκ των απαιτήσεων που αποτελούν τον ορισμό της εμπιστοσύνης. Η προσθήκη νέων πληροφοριών και η μεταβολή του δέντρου στόχων ανήκουν αμφότερες σε αυτή την κατηγορία. Επομένως, κρίνεται σκόπιμος ο υπολογισμός της νέας απαίτησης και στην συνέχεια και ο επαναυπολογισμός της συνολικής εμπιστοσύνης στην υπηρεσία.



Σχήμα 4.4-7 Ακολουθιακό Διάγραμμα Αλλαγής Μοντέλου Απαίτησης

4.4.2.4 Ακολουθιακό Διάγραμμα Αλλαγής Μοντέλου Εκτίμησης Εμπιστοσύνης

Σε αυτή την περίπτωση, ο χρήστης μεταβάλλει τον γράφο εκτίμησης της εμπιστοσύνης στην υπηρεσία που ερευνάται. Μεταβολή του γράφου προκαλεί τόσο η προσθήκη μιας νέας απαίτησης, η οποία, σύμφωνα πάντα με τον χρήστη, είναι σημαντική για την εκτίμηση της υπηρεσίας, όσο και η ενδεχόμενη αλλαγή στις ακμές που συνδέουν τις απαιτήσεις. Αν έχει προστεθεί μια νέα απαίτηση, τότε χρειάζεται να γίνει ο υπολογισμός της και μόνο. Διαφορετικά, παίρνουμε τις ήδη υπολογισμένες τιμές για τις απαιτήσεις. Τέλος, επαναυπολογίζεται η εκτίμηση της εμπιστοσύνης στην υπηρεσία.



Σχήμα 4.4-8 Ακολουθιακό Διάγραμμα Αλλαγής Μοντέλου Εκτίμησης Εμπιστοσύνης

Κεφάλαιο 5: Εκτίμηση εμπιστοσύνης (Trust Evaluation)

Στο κεφάλαιο αυτό περιγράφεται ο προεπιλεγμένος αλγόριθμος που χρησιμοποιείται από το περιβάλλον-πλαίσιο, το οποίο προτείνεται στην παρούσα διπλωματική εργασία. Ο αλγόριθμος αφορά τον υπολογισμό της τελικής εμπιστοσύνης που απολαμβάνει μια υπηρεσία, όπως περιγράφεται στα [24], [25], [26], [27]. Βασικοί στόχοι του κεφαλαίου είναι τόσο η κατανόηση της μεθοδολογίας, βάση της οποίας δημιουργείται ο γράφος αναπαράστασης των σχέσεων ανάμεσα στις απαιτήσεις, όσο και η περιγραφή του αλγόριθμου που εκτελείται πάνω στον γράφο αυτό και παράγει την τελική απάντηση στην ερώτηση του πόσο άξια εμπιστοσύνης θεωρείται ένα σύστημα, μια υπηρεσία, ή ένας πόρος, δεδομένων των συνθηκών.

5.1 Κατασκευή γράφου απαιτήσεων

Όπως αναφέρθηκε και στην εισαγωγή, ορίζουμε την εμπιστοσύνη σαν τον βαθμό της βεβαιότητας, που έχει ένας χρήστης (system stakeholder) ότι το σύστημα, ο πόρος, ή η υπηρεσία, θα ικανοποιήσουν τις λειτουργικές και μη λειτουργικές απαιτήσεις, που αυτό το σύστημα, πόρος, ή υπηρεσία, έχει προδιαγραφεί να παραδώσει ως αποτέλεσμα, δεδομένου του περιεχομένου για το οποίο ορίζεται.

5.1.1 Ορισμός Εμπιστοσύνης

Σε αυτό το πλαίσιο ορισμού, η μαθηματική μοντελοποίηση του προβλήματος ορίζεται ως εξής:

- ✓ Έστω ένα σύνολο $\mathcal{R}$ που αντιπροσωπεύει όλες τις δυνατές επιμέρους απαιτήσεις (Requirements) που μπορεί να έχει ο χρήστης. Έστω, επίσης, ένα σύνολο $\mathbf{R}$, που συνίσταται από τις επιμέρους απαιτήσεις που χρήζουν ικανοποίησης σε κάθε περίπτωση, και είναι υποσύνολο του $\mathcal{R}$ ($R \subseteq \mathcal{R}$).
- ✓ Έστω $\mathbf{A}$ ο πάροχος (Provider) του συστήματος, του πόρου, ή της υπηρεσίας, του οποίου η εμπιστοσύνη ερευνάται.
- ✓ Έστω $\mathbf{S}$ η υπηρεσία (Service), ή ο πόρος, που παρέχεται προς χρήση και αξιολόγηση της εμπιστοσύνης του
- ✓ Έστω $\mathbf{C}$ το περιεχόμενο (Context), πάνω στο οποίο εφαρμόζεται η υπηρεσία, ή ο παρεχόμενος πόρος.
- ✓ Έστω $\mathbf{K}$ ο εκάστοτε χρήστης (Stakeholder) του συστήματος, πόρου, ή υπηρεσίας.

Λαμβάνοντας υπ' όψιν τους παραπάνω ορισμούς, θεωρούμε ότι η εμπιστοσύνη ορίζεται ως εξής:

$$T(R, A, S, C, K)$$

Επί παραδείγματι, ένας χρήστης εμπιστεύεται την Ευρωπαϊκή Τράπεζα να πιστοποιεί πωλητές ηλεκτρονικών συσκευών εντός της Ευρωπαϊκής Ένωσης, με απαιτήσεις την ασφάλεια και την απόδοση. Η ίδια τράπεζα, παρ' όλα αυτά, δεν χρήζει αντίστοιχης εμπιστοσύνης σε περιπτώσεις αγοράς αγαθών εκτός της Ευρωπαϊκής Ένωσης.

5.1.2 Γράφος απαιτήσεων

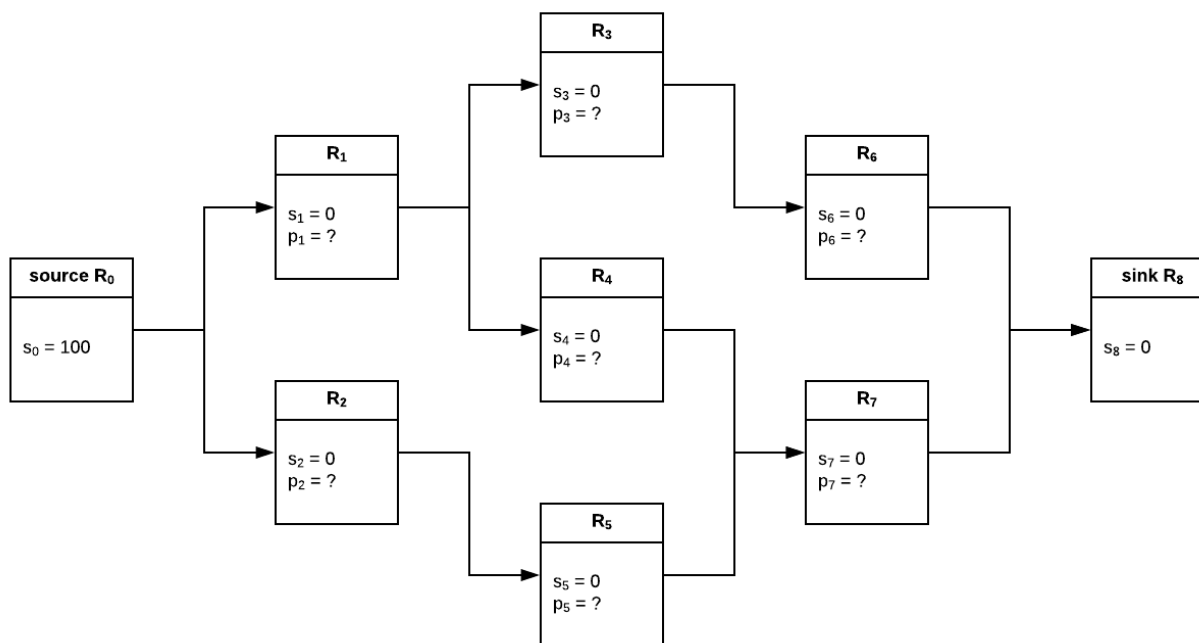
Στην προηγούμενη παράγραφο ορίσαμε ένα σύνολο $R = \{R_i\}$, $i=1,2,\dots,n$, που συνίσταται από τις επιμέρους απαιτήσεις, που χρίζουν ικανοποίησης στην εκάστοτε περίπτωση. Έστω, λοιπόν, οι τιμές p_i , $i=1,2,\dots,n$, οι οποίες αντιπροσωπεύουν τις αντίστοιχες πιθανότητες ικανοποίησης αυτών των απαιτήσεων. Οι τιμές που λαμβάνει κάθε πιθανότητα p_i υπολογίζονται από μοντέλα στόχων, όπως περιγράφεται στην επόμενη παράγραφο.

Για κάθε μέλος του συνόλου R δημιουργείται ένας κόμβος. Κάθε κόμβος ονοματίζεται με βάση την απαίτηση που αντιπροσωπεύει και φέρει τις ακόλουθες πληροφορίες. Περιέχει την αντίστοιχη τιμή p_i , την πιθανότητα, δηλαδή, ικανοποίησης της απαίτησης, , καθώς επίσης και την τιμή s_i , που αρχικοποιείται με την τιμή 0, και χρησιμοποιείται, όπως θα δούμε αναλυτικότερα και στην **ενότητα 5.2**, για τον υπολογισμό της δύναμης του μέχρι τώρα μονοπατιού.

Ο χρήστης, επιπλέον, έχει πλήρη έλεγχο όσον αφορά τις αλληλεπιδράσεις και την σημαντικότητα τις κάθε απαίτησης. Οι απαιτήσεις που συνδέονται μεταξύ τους με τον λογικό τελεστή **H (OR)** αναπαριστούνται με κόμβους παράλληλους, δηλαδή κόμβους με κοινό πρόγονο στον γράφο απαιτήσεων. Οι απαιτήσεις με τέτοια σύνδεση επιτρέπουν τον ορισμό εναλλακτικών περιπτώσεων κατά τις οποίες ο χρήστης είναι ικανοποιημένος με την εμπιστοσύνη της υπηρεσίας. Οι απαιτήσεις, όμως, μπορεί να συνδέονται μεταξύ τους και με τον λογικό τελεστή **KAI (AND)**. Σε αυτή την περίπτωση έχουμε σύνδεση σε σειρά, δηλαδή η μια απαίτηση είναι πρόγονος της άλλης στον γράφο απαιτήσεων. Αυτού του τύπου η σύνδεση ορίζει την ανάγκη ικανοποίησης όλων των απαιτήσεων που συνδέονται με αυτόν τον τρόπο.

Τέλος, χρησιμοποιούνται δύο επιπλέον βοηθητικοί κόμβοι. Ο ένας είναι ο κόμβος **source R_0** , ο οποίος αρχικοποιείται με τιμή δύναμης μονοπατιού $s_0 = 100$, και συνδέεται ως πηγή του γράφου απαιτήσεων. Ο δεύτερος είναι ο κόμβος **sink R_{n+1}** , ο οποίος αρχικοποιείται με τιμή δύναμης μονοπατιού $s_{n+1} = 0$, και συνδέεται ως καταβόθρα στην οποία καταλήγουν όλα τα μονοπάτια του γράφου απαιτήσεων. Ο κόμβος αυτός χρησιμεύει για την λήψη της τελικής τιμής εμπιστοσύνης που απολαμβάνει η υπηρεσία, ή ο πόρος, που εξετάζεται.

Ακολούθως παρουσιάζεται παράδειγμα γράφου που δημιουργείται με την παραπάνω διαδικασία.



Σχήμα 5.1-1 Παράδειγμα Γράφου Απαιτήσεων

5.1.3 Απόδοση πιθανοτήτων σε κόμβους

Τα δέντρα στόχων AND/OR αποτελούν έναν φορμαλισμό μοντελοποίησης που χρησιμοποιείται εκτεταμένα σε περιπτώσεις περιγραφής απαιτήσεων, λειτουργικών και μη[5]. Βασική έννοια αποτελεί η από πάνω προς τα κάτω αποσύνθεση των στόχων σε υποστόχους, όπως αναφέρθηκε εκτενώς στην υποενότητα 2.3.

Ενας στόχος G με AND-αποσύνθεση ικανοποιείται μόνο όταν όλοι οι υποστόχοι του ικανοποιούνται, ενώ ένας στόχος G με OR-αποσύνθεση, όταν έστω και ένας από τους υποστόχους είναι αληθής.

Ακολουθώντας την σημειογραφία που ακολουθείται στο [5], συμβολίζουμε την αποσύνθεση AND ενός στόχου G στο σύνολο των επιμέρους στόχων του ως εξής:

$$G \xrightarrow{AND} \alpha$$

Αντίστοιχα για η αποσύνθεση OR συμβολίζεται ως εξής:

$$G \xrightarrow{OR} \alpha$$

Επιπροσθέτως ορίζεται και η έννοια της συνεισφοράς ανάμεσα σε στόχους. Οι περιπτώσεις συνεισφοράς που προσφέρονται είναι τέσσερις και είναι οι ακόλουθες:

- $++S(g, g') : \eta \text{ ικανοποίηση του } g \text{ συνεισφέρει θετικά στην ικανοποίηση του } g'$
($g \rightarrow g'$)
- $--S(g, g') : \eta \text{ ικανοποίηση του } g \text{ συνεισφέρει αρνητικά στην ικανοποίηση του } g'$
($g \rightarrow \neg g'$)
- $++D(g, g') : \eta \text{ άρνηση του } g \text{ συνεισφέρει θετικά στην άρνηση του } g'$
($\neg g \rightarrow \neg g'$)
- $--D(g, g') : \eta \text{ άρνηση του } g \text{ συνεισφέρει αρνητικά στην άρνηση του } g'$
($\neg g \rightarrow g'$)

Με βάση, λοιπόν, τις παραπάνω αποσυνθέσεις και συνεισφορές, δημιουργείται ένα ξεχωριστό δέντρο στόχων, που αντιπροσωπεύει κάθε μια από τις λειτουργικές και μη λειτουργικές απαιτήσεις, που ανήκουν στο σύνολο **R**. Στην συνέχεια, σύμφωνα με συλλογιστική που είναι βασισμένη σε δέντρα στόχων, όπως έχει περιγραφεί στα [31][32], υπολογίζεται η πιθανότητα ικανοποίησης της απαίτησης που αποτελεί τον βασικό στόχο του δέντρου.

Πιο συγκεκριμένα, τα δέντρα στόχων μετασχηματίζονται αρχικά σε CNF κανόνες[31], ή Weighted Fuzzy κανόνες[32], και εν συνεχεία, αφού αντικατασταθούν οι τιμές που αντιστοιχούν στους εκάστοτε υποστόχους, υπολογίζεται η τιμή του βασικού στόχου.

5.1.3.1 Μετασχηματισμός σε CNF κανόνες

Στην περίπτωση μετασχηματισμού του δέντρου στόχων σε CNF μορφή, ο κάθε υποστόχος μετασχηματίζεται, σύμφωνα με τους μετασχηματισμούς που ακολουθούν, μέχρις ότου παραχθεί η αναπαράσταση του βασικού στόχου σε CNF μορφή.

Μετασχηματισμός αποσυνθέσεων

- *Αποσύνθεση AND*

$$G \xrightarrow{AND} \alpha \Rightarrow g_1 \wedge g_2 \wedge \dots \wedge g_n \leftrightarrow \alpha \Rightarrow$$

$$\Rightarrow (g_1 \wedge g_2 \wedge \dots \wedge g_n \rightarrow \alpha) \wedge (\alpha \rightarrow g_1 \wedge g_2 \wedge \dots \wedge g_n) \Rightarrow$$

$$\Rightarrow (\neg(g_1 \wedge g_2 \wedge \dots \wedge g_n) \vee \alpha) \wedge (\neg \alpha \vee (g_1 \wedge g_2 \wedge \dots \wedge g_n)) \Rightarrow$$

$$\Rightarrow (\neg g_1 \vee \neg g_2 \vee \dots \vee \neg g_n \vee \alpha) \wedge (\neg \alpha \vee g_1) \wedge (\neg \alpha \vee g_2) \wedge \dots \wedge (\neg \alpha \vee g_n)$$
- *Αποσύνθεση OR*

$$G \xrightarrow{OR} \alpha \Rightarrow g_1 \vee g_2 \vee \dots \vee g_n \leftrightarrow \alpha \Rightarrow$$

$$\begin{aligned}
&\Rightarrow (g_1 \vee g_2 \vee \dots \vee g_n \rightarrow \alpha) \wedge (\alpha \rightarrow g_1 \vee g_2 \vee \dots \vee g_n) \Rightarrow \\
&\Rightarrow (\neg(g_1 \vee g_2 \vee \dots \vee g_n) \vee \alpha) \wedge (\neg\alpha \vee (g_1 \vee g_2 \vee \dots \vee g_n)) \Rightarrow \\
&\Rightarrow ((\neg g_1 \wedge \neg g_2 \wedge \dots \wedge \neg g_n) \vee \alpha) \wedge (\neg\alpha \vee g_1 \vee g_2 \vee \dots \vee g_n) \Rightarrow \\
&\Rightarrow (\neg g_1 \vee \alpha) \wedge (\neg g_2 \vee \alpha) \wedge \dots \wedge (\neg g_n \vee \alpha) \wedge (\neg\alpha \vee g_1 \vee g_2 \vee \dots \vee g_n)
\end{aligned}$$

Μετασχηματισμός Συνεισφορών

- $g_1 \xrightarrow{++S} g_2 \Rightarrow g_1 \rightarrow g_2 \Rightarrow \neg g_1 \vee g_2$
- $g_1 \xrightarrow{--S} g_2 \Rightarrow g_1 \rightarrow \neg g_2 \Rightarrow \neg g_1 \vee \neg g_2$
- $g_1 \xrightarrow{++D} g_2 \Rightarrow \neg g_1 \rightarrow \neg g_2 \Rightarrow g_1 \vee \neg g_2$
- $g_1 \xrightarrow{--D} g_2 \Rightarrow \neg g_1 \rightarrow g_2 \Rightarrow g_1 \vee g_2$

Οι συνεισφορές προστίθενται στην εκάστοτε αποσύνθεση AND ή OR με τον λογικό τελεστή Ή. Στην συνέχεια η έκφραση μετασχηματίζεται ώστε να είναι σε CNF μορφή, από όπου και υπολογίζεται το τελικό αποτέλεσμα. Το αποτέλεσμα αυτό παρέχεται σαν πιθανοτική τιμή ικανοποίησης της απαίτησης που ερευνάται και καταχωρείται στο αντίστοιχο κόμβο του γράφου απαιτήσεων.

5.1.3.2 Μετασχηματισμός σε *Weighted Fuzzy κανόνες*

Σε περίπτωση που οι περιορισμένες επιλογές που παρέχονται από την **απλή λογική (Boolean logic)**, το λογικό **0(false)** και το λογικό **1(true)** δηλαδή, δεν επαρκούν για την περιγραφή ενός στόχου, μπορεί να χρησιμοποιηθεί η **ασαφής λογική (fuzzy logic)**. Στην **ασαφή λογική (fuzzy logic)**, που είναι μια λογική πολλών μεταβλητών, η κάθε μεταβλητή μπορεί να λάβει οποιαδήποτε τιμή αληθείας στο διάστημα $[0,1]$. Λαμβάνοντας αυτό υπόψιν μας μπορούμε να ορίσουμε μια ασαφή γνωσιακή βάση που αποτελείται από **weighted fuzzy κανόνες (wf-rules)**. Ακολουθώντας την σημειογραφία που χρησιμοποιείται στο [32], ορίζουμε έναν weighted fuzzy κανόνα ως εξής:

$$w : FuzzyAtom_k(p) \leftarrow (w_1; FuzzyAtom_1(p)) \tilde{\wedge} (w_2; FuzzyAtom_2(p)) \tilde{\wedge} \dots \tilde{\wedge} (w_n; FuzzyAtom_n(p))$$

Όπου:

- a) Το δεξί μέλος αποτελείται από μια ασαφή σύνδεση μεταξύ *ασαφών ατόμων (fuzzy atoms)*, όπου τα βάρη $w_1, w_2, \dots, w_n$ συμβολίζουν την σχετική σημασία ανάμεσα στα άτομα που βρίσκονται δεξιά.
- b) Στο αριστερό μέλος βρίσκεται το *ασαφές άτομο (fuzzy atom)* στο οποίο αναφέρεται ο κανόνας, καθώς επίσης και το βάρος w αντιπροσωπεύει την σχετική σημασία του κανόνα, ανάμεσα σε άλλους κανόνες που αναφέρονται στο ίδιο άτομο.

Ειδική περίπτωση **weighted fuzzy κανόνα (wf-rule)** αποτελεί ένα **ασαφές δεδομένο (fuzzy fact)**, το οποίο ορίζεται ως εξής:

$$w : FuzzyAtom_k(p) \leftarrow (1.0 ; t)$$

Όπου:

- a) w είναι η τιμή αληθείας του $FuzzyAtom_k$
- b) t είναι ένα προκαθορισμένο ασαφές άτομο που αντιπροσωπεύει την τιμή απόλυτης αληθείας.

Δοθέντων όλης της ασαφούς γνωσιακής βάσης (fuzzy knowledge base), χρησιμοποιείται μια s -νόρμα (fuzzy OR operator), για τον συνδυασμό των κανόνων με κοινό αριστερό μέλος, και μια t -νόρμα (fuzzy AND operator), για τον συνδυασμό των ατόμων που βρίσκονται στο δεξί μέρος. Οι νόρμες αυτές χρησιμοποιούνται στο [32] και είναι οι ακόλουθες.

Η s -νόρμα ορίζεται ως εξής:

$$\perp_{sum} (a, b) = a + b - a \cdot b$$

Η t -νόρμα ορίζεται ως εξής:

$$\Omega_{\langle w_1, \dots, w_n \rangle}^{prod} = \max \left\{ 0, \bar{w} - 1 + \prod_{i=1}^n \alpha_i^{w_i} \right\}$$

όπου α_i είναι οι τιμές αληθείας των ατόμων και w_i τα αντίστοιχα βάρη. Σε περίπτωση που όλα τα βάρη είναι 1, η t -νόρμα γίνεται η ακόλουθη:

$$\Gamma_{prod} (a, b) = a \cdot b$$

Με βάση, λοιπόν, τους **weighted fuzzy κανόνες (wf-rules)** και τους αντίστοιχους τελεστές (**fuzzy OR, AND operator**) προχωρούμε στον μετασχηματισμό του δέντρου στόχων στους αντίστοιχους κανόνες χρησιμοποιώντας τους μετασχηματισμούς που ακολουθούν.

Μετασχηματισμός αποσυνθέσεων

- Αποσύνθεση AND : $\langle AND, g, \{g_1, \dots, g_n\}, 1 \rangle$
 - ✓ $1.0 : HighSat(g) \leftarrow (1.0; HighSat(g_1)) \tilde{\wedge} \dots \tilde{\wedge} (1.0; HighSat(g_n))$
 - ✓ $1.0 : LowSat(g) \leftarrow (1.0; LowSat(g_1))$
 - ✓ ...
 - ✓ $1.0 : LowSat(g) \leftarrow (1.0; LowSat(g_n))$
- Αποσύνθεση OR : $\langle OR, g, \{g_1, \dots, g_n\}, 1 \rangle$
 - ✓ $1.0 : LowSat(g) \leftarrow (1.0; LowSat(g_1)) \tilde{\wedge} \dots \tilde{\wedge} (1.0; LowSat(g_n))$
 - ✓ $1.0 : HighSat(g) \leftarrow (1.0; HighSat(g_1))$
 - ✓ ...
 - ✓ $1.0 : HighSat(g) \leftarrow (1.0; HighSat(g_n))$

Μετασχηματισμός συνεισφορών

- ++S : $\langle S^P, g_t, \{g_s\}, w \rangle$
✓ 1.0 : $HighSat(g_t) \leftarrow (w; HighSat(g_s))$
- --S : $\langle S^N, g_t, \{g_s\}, w \rangle$
✓ 1.0 : $LowSat(g_t) \leftarrow (w; HighSat(g_s))$
- ++D : $\langle D^P, g_t, \{g_s\}, w \rangle$
✓ 1.0 : $LowSat(g_t) \leftarrow (w; LowSat(g_s))$
- --D : $\langle D^N, g_t, \{g_s\}, w \rangle$
✓ 1.0 : $HighSat(g_t) \leftarrow (w; LowSat(g_s))$

Χρησιμοποιώντας τους παραπάνω μετασχηματισμούς και τελεστές, υπολογίζουμε τις *ασαφείς (fuzzy) τιμές* των υποστόχων g_t μέχρι να καταλήξουμε στον βασικό στόχο g και το αντίστοιχο ζεύγος τιμών του $[LowSat, HighSat]$. Το ζεύγος αυτό, τελικά, παράγει μια πιθανοτική τιμή με χρήση μιας **διαδικασίας αποσαφήνισης (defuzzification process)**, η οποία περιγράφεται αναλυτικά στο [32]. Το αποτέλεσμα αυτής της διαδικασίας παρέχεται σαν πιθανοτική τιμή ικανοποίησης της απαίτησης που ερευνάται και καταχωρείται στο αντίστοιχο κόμβο του γράφου απαιτήσεων.

5.2 Αλγόριθμος εκτίμησης εμπιστοσύνης σε μια υπηρεσία

Στην προηγούμενη ενότητα κατασκευάστηκε ο γράφος αναπαράστασης των απαιτήσεων που έχει ο χρήστης από την υπηρεσία που ερευνάται. Ο γράφος δημιουργήθηκε έτσι ώστε να αναπαριστά ακριβώς τις ανάγκες του χρήστη από την υπηρεσία, και θα πάνω σε αυτόν θα εφαρμοστεί ο αλγόριθμος TidalTrust.

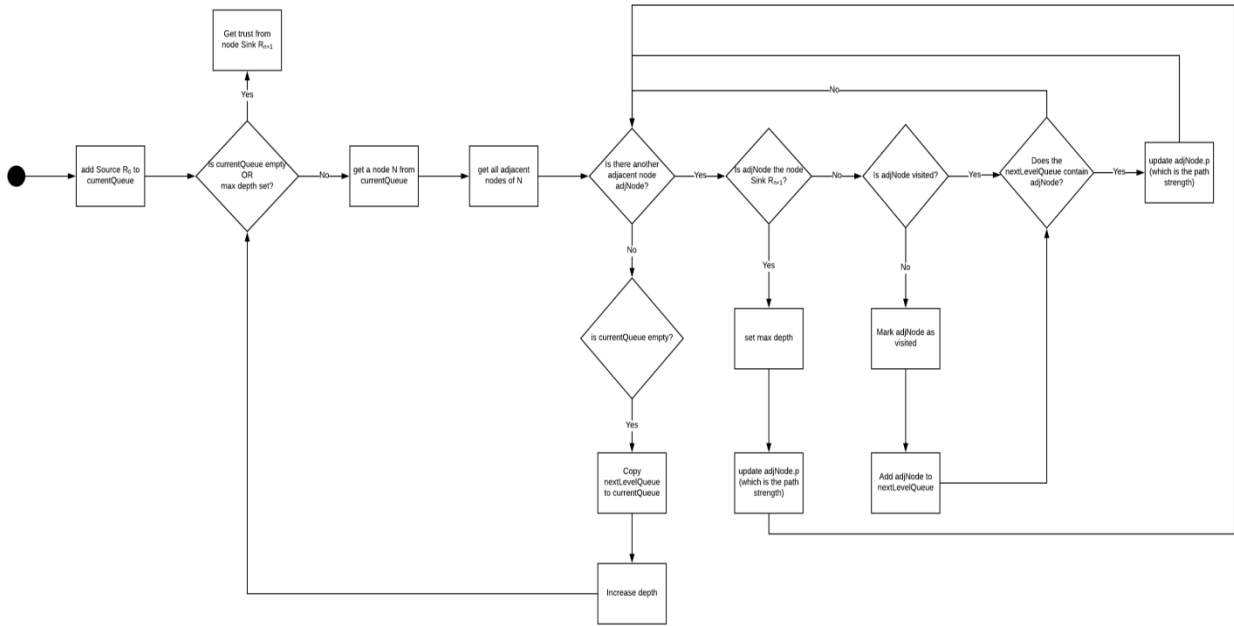
Ο αλγόριθμος TidalTrust προτάθηκε και χρησιμοποιήθηκε αρχικά από τον Golbeck[28] για την εκτίμηση εμπιστοσύνης σε κοινωνικά δίκτυα. Είναι, πλέον, από τους πιο γνωστούς αλγορίθμους στον τομέα των κοινωνικών δικτύων και εκτεταμένη έρευνα έχει γίνει πάνω στην αποδοτικότητά του[27], [28], αλλά και σε κάποιες παραλλαγές του [29].

Οι απαιτήσεις, όμως, που παρέχονται από τον χρήστη, σχετικά με την εκτίμηση της εμπιστοσύνης μιας παρεχόμενης υπηρεσίας, μπορούν να συνδεθούν και να αλληλεπιδρούν μεταξύ τους, όπως ακριβώς θα έκαναν και οι αντίστοιχοι κόμβοι ενός κοινωνικού δικτύου. Αν, επιπροσθέτως, λάβουμε υπ' όψιν μας και την δυνατότητα μελλοντικής επέκτασης του περιβάλλοντος-πλαισίου, προκειμένου να υποστηριχθεί αλληλεπίδραση μεταξύ διαφορετικών παρόχων, υπηρεσιών και απαιτήσεων που τις απαρτίζουν, τότε γίνεται παραπάνω από εμφανής η προτίμηση και τελική επιλογή αυτού του αλγορίθμου σαν προεπιλογή.

Δοθέντος, λοιπόν, ένας γράφος G , ο οποίος κατασκευάστηκε με την διαδικασία που περιγράφεται στην προηγούμενη ενότητα, ο αλγόριθμος TidalTrust υπολογίζει την εμπιστοσύνη T , που απολαμβάνει η υπηρεσία προς εξέταση, «ανακαλύπτοντας» το μονοπάτι που είναι συντομότερο και έχει τον μεγαλύτερο βαθμό δύναμης. Για να το επιτύχει αυτό, προσθέτει τον κόμβο $source R_0$ σε μία ουρά $currentQueue$ και ακολούθως διασχίζει τον γράφο χρησιμοποιώντας μια BFS υπολοίηση, τα βήματα της οποίας περιγράφονται στην συνέχεια:

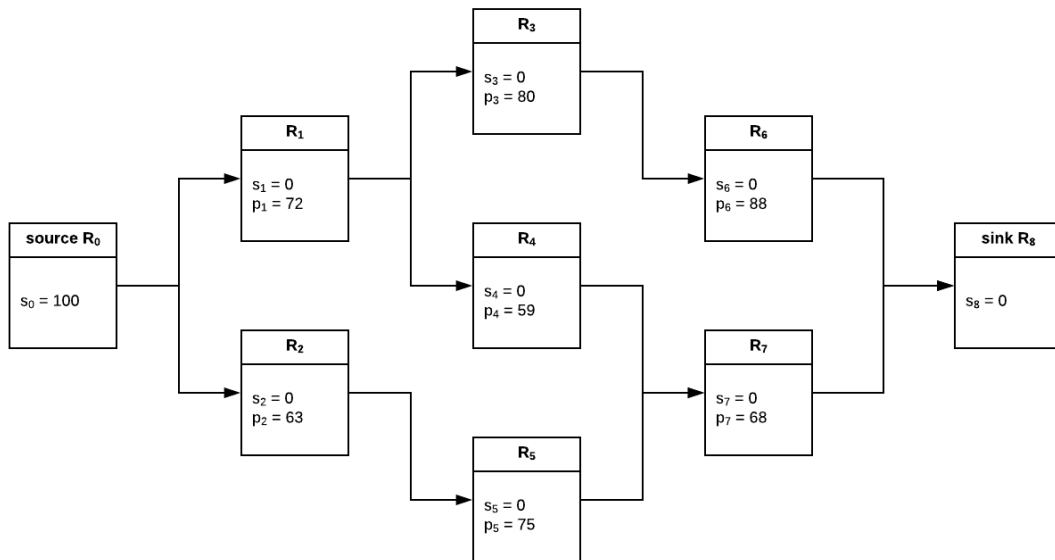
1. Έστω ένας κόμβος $N \in currentQueue$ και έστω $adjNode$ γειτονικός κόμβος του N .
 $\forall adjNode$ του N .
 - a. Εάν $adjNode.visited = false$, τότε πρόσθεσε $adjNode$ στην ουρά $nextLevelQueue$.
 - b. Εάν $adjNode \equiv sink R_{n+1}$, τότε
 - i. $adjNode.s = \max\{N.s, adjNode.s\}$, δηλαδή ενημερώνεται με την μέγιστη τιμή ανάμεσα στην προηγούμενη τιμή του και την δύναμη μονοπατιού του προηγούμενου κόμβου ($N.s$).
 - ii. $maxDepth = depth$, όπου $depth$ είναι το μέχρι τώρα βάθος του γράφου.
 - c. Αλλιώς εάν $adjNode \in nextLevelQueue$, τότε υπολόγισε $adjNode.s$, δηλαδή τη δύναμη μονοπατιού του.
 - i. Εύρεση $\min\{N.s, adjNode.p\}$, δηλαδή την ελάχιστη τιμή ανάμεσα στην δύναμη μονοπατιού του προηγούμενου κόμβου ($N.s$) και την πιθανοτική τιμή ικανοποίησης του κόμβου που ερευνάται ($adjNode.p$).
 - ii. Εάν η $\min\{N.s, adjNode.p\} > adjNode.s$, τότε $adjNode.s = \min\{N.s, adjNode.p\}$.
 - iii. Έτσι εξασφαλίζουμε ότι $adjNode.s = \max\{\min\{N.s, adjNode.p\}, adjNode.s\}$.
2. Μόλις $currentQueue.isEmpty = true$, έλεγξε αν έχει πάρει τιμή το $maxDepth$.
 - a. Εάν δεν έχει πάρει τιμή το $maxDepth$, τότε αντίγραψε την $nextLevelQueue$ στην $currentQueue$ και , επέστρεψε στο βήμα 1.
 - b. Εάν έχει πάρει τιμή το $maxDepth$, τότε γίνεται αντιληπτό ότι τα ελάχιστα μονοπάτια έχουν εξερευνηθεί και δεν υπάρχει λόγος περαιτέρω εκτέλεσης του αλγορίθμου. Έτσι έχουμε τερματισμό του αλγορίθμου.
3. Η εκτίμηση εμπιστοσύνης, για την υπηρεσία που ερευνάται, είναι ίση με την δύναμη μονοπατιού του κόμβου $sink R_{n+1}$. Είναι, δηλαδή $trustValue = sink.s$.

Παρακάτω παρουσιάζεται και ο αλγόριθμος σε μορφή διαγράμματος ροής προκειμένου να γίνει ευκολότερη η κατανόηση.



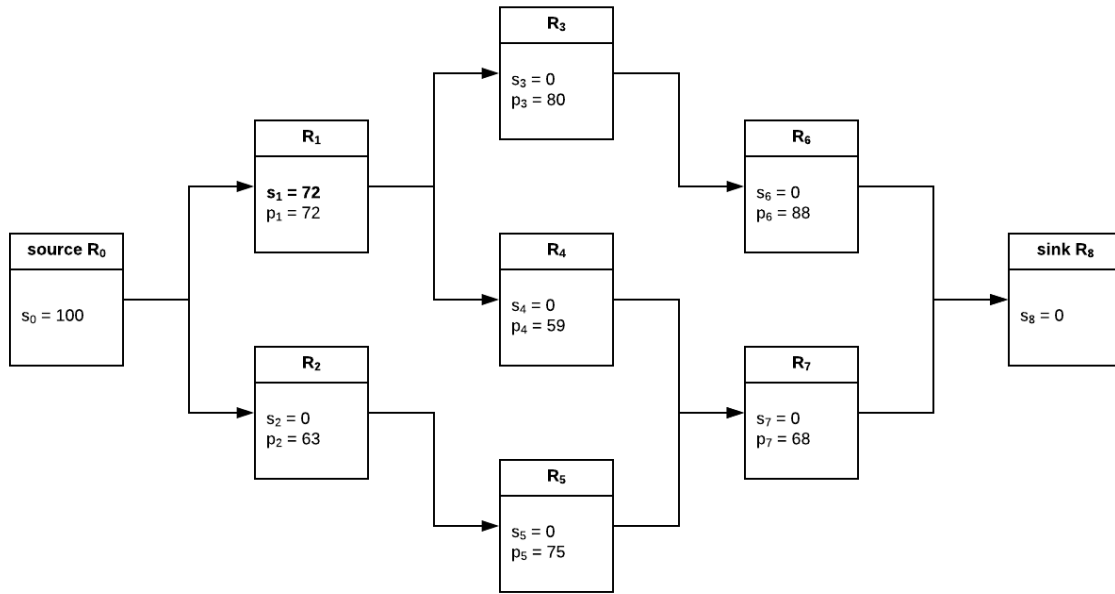
Σχήμα 5.2-1 Διάγραμμα ροής αλγόριθμου TidalTrust

Στην συνέχεια παρουσιάζεται ένα παράδειγμα εκτέλεσης του αλγορίθμου TidalTrust με χρήση ενός πιθανού γράφου απαιτήσεων και των ενδιάμεσων τιμών του.



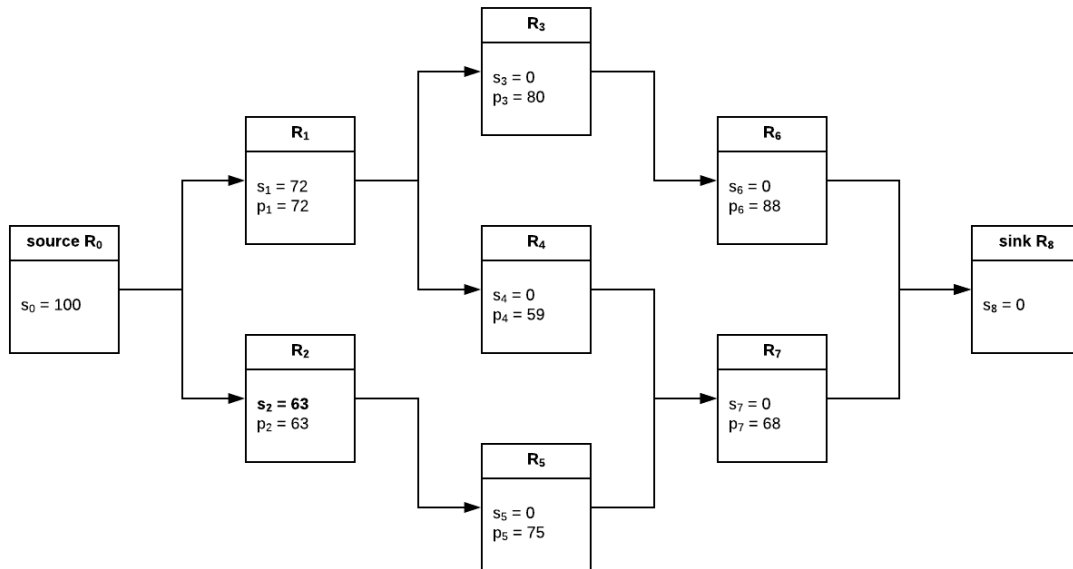
Σχήμα 5.2-2 Παράδειγμα: Βήμα 1

1. Στην αρχή όλοι οι κόμβοι του γράφου έχουν μηδενική δύναμη μονοπατιού, εκτός από τον αρχικό Source R_0 .



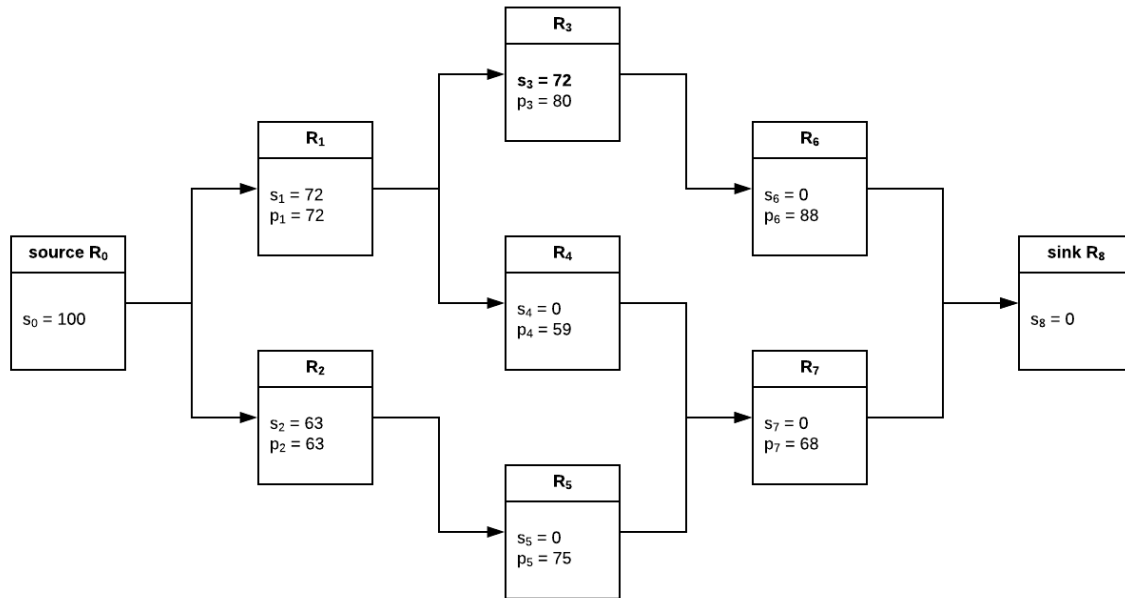
Σχήμα 5.2-3 Παράδειγμα: Βήμα 2

2. Βρισκόμαστε στον κόμβο R_1 . Συγκρίνουμε το p_1 με το s_0 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 72 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_1 . Επομένως $s_1 = 72$.



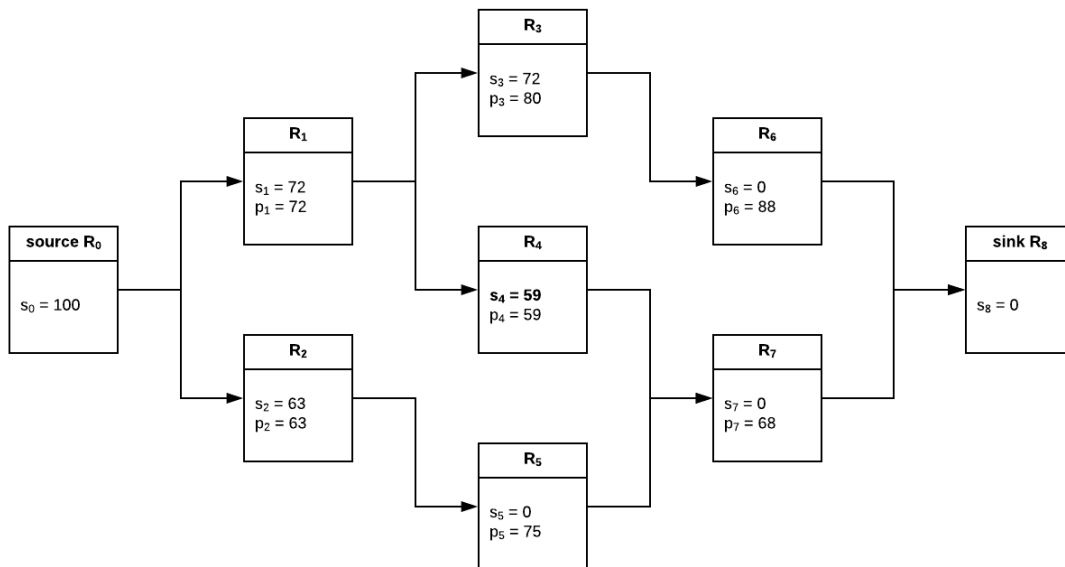
Σχήμα 5.2-4 Παράδειγμα: Βήμα 3

3. Βρισκόμαστε στον κόμβο R_2 . Συγκρίνουμε το p_2 με το s_0 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 63 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_2 . Επομένως $s_2 = 63$.



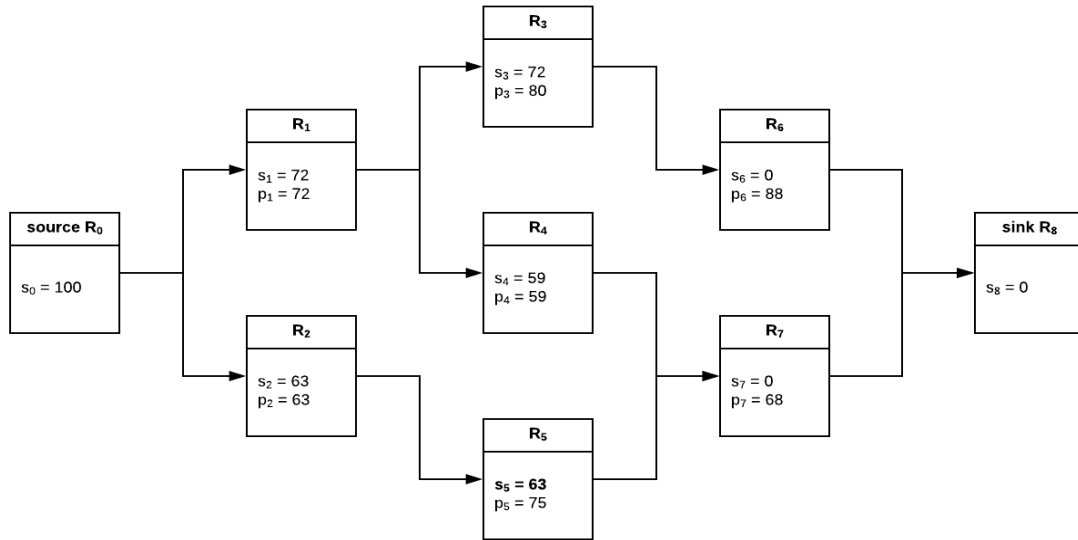
Σχήμα 5.2-5 Παράδειγμα: Βήμα 4

4. Βρισκόμαστε στον κόμβο R_3 . Συγκρίνουμε το p_3 με το s_1 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 72 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_3 . Επομένως $s_3 = 72$.



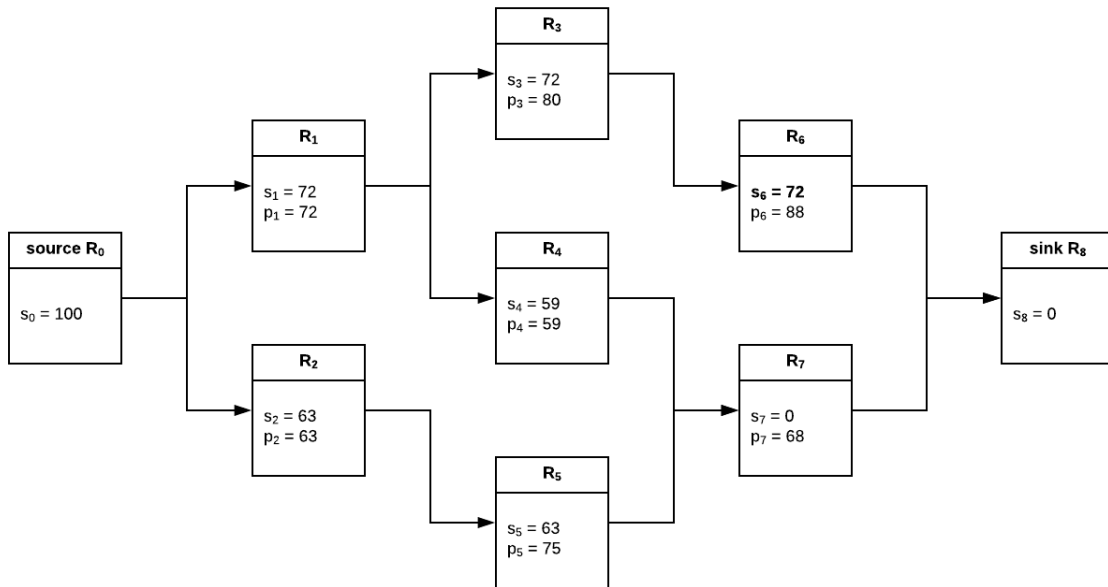
Σχήμα 5.2-6 Παράδειγμα: Βήμα 5

5. Βρισκόμαστε στον κόμβο R_4 . Συγκρίνουμε το p_4 με το s_1 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 59 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_4 . Επομένως $s_4 = 59$.



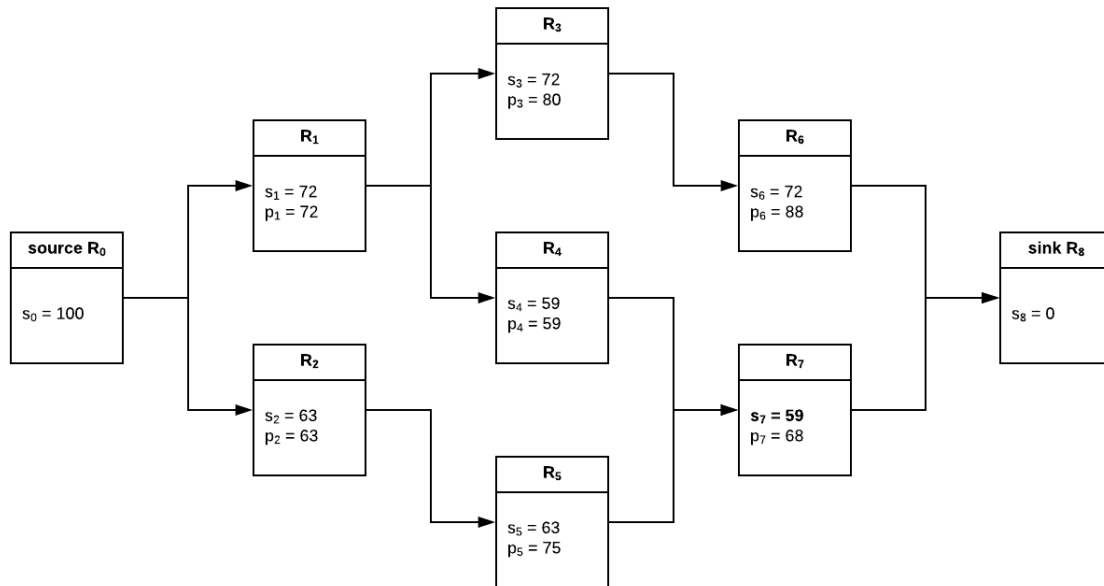
Σχήμα 5.2-7 Παράδειγμα: Βήμα 6

6. Βρισκόμαστε στον κόμβο R_5 . Συγκρίνουμε το p_5 με το s_2 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 63 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_5 . Επομένως $s_5 = 63$.



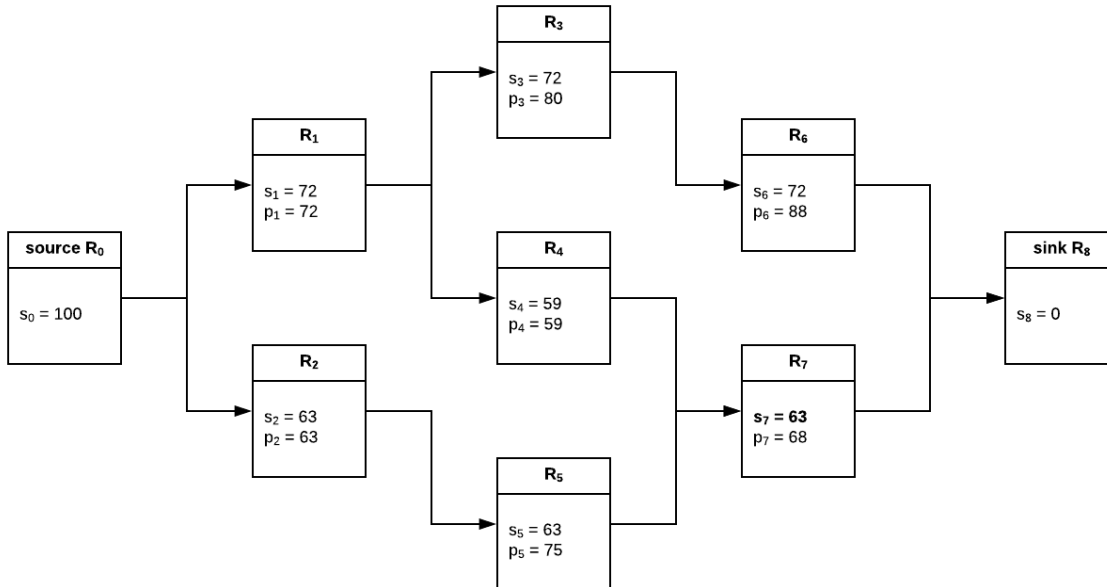
Σχήμα 5.2-8 Παράδειγμα: Βήμα 7

7. Βρισκόμαστε στον κόμβο R_6 . Συγκρίνουμε το p_6 με το s_3 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 72 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_6 . Επομένως $s_6 = 72$.



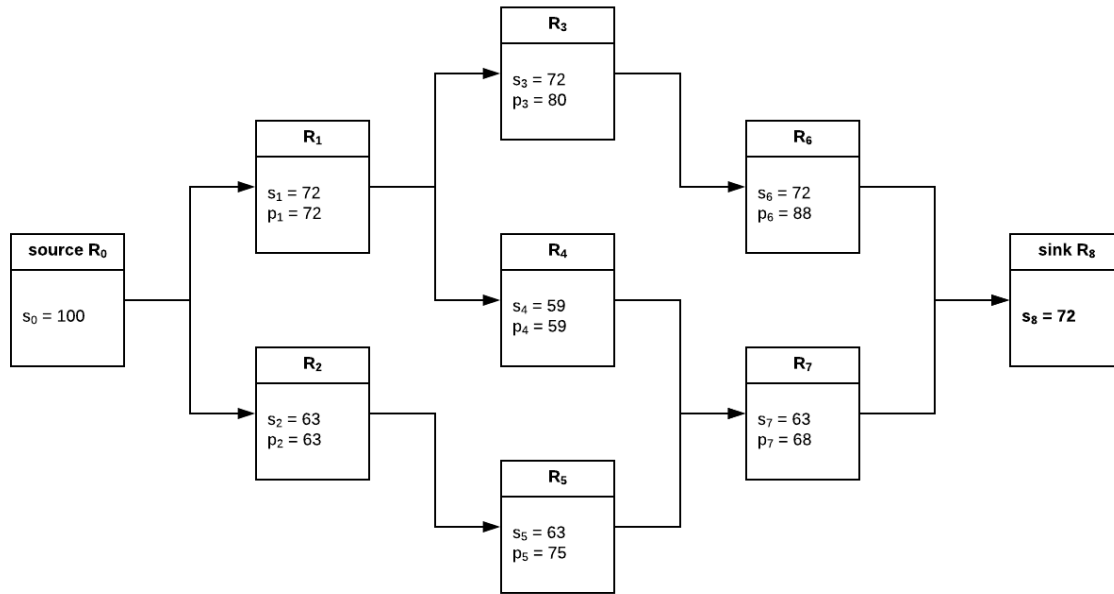
Σχήμα 5.2-9 Παράδειγμα: Βήμα 8

8. Βρισκόμαστε στον κόμβο R_7 . Συγκρίνουμε το p_7 με το s_4 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 59 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_7 . Επομένως $s_7 = 59$.



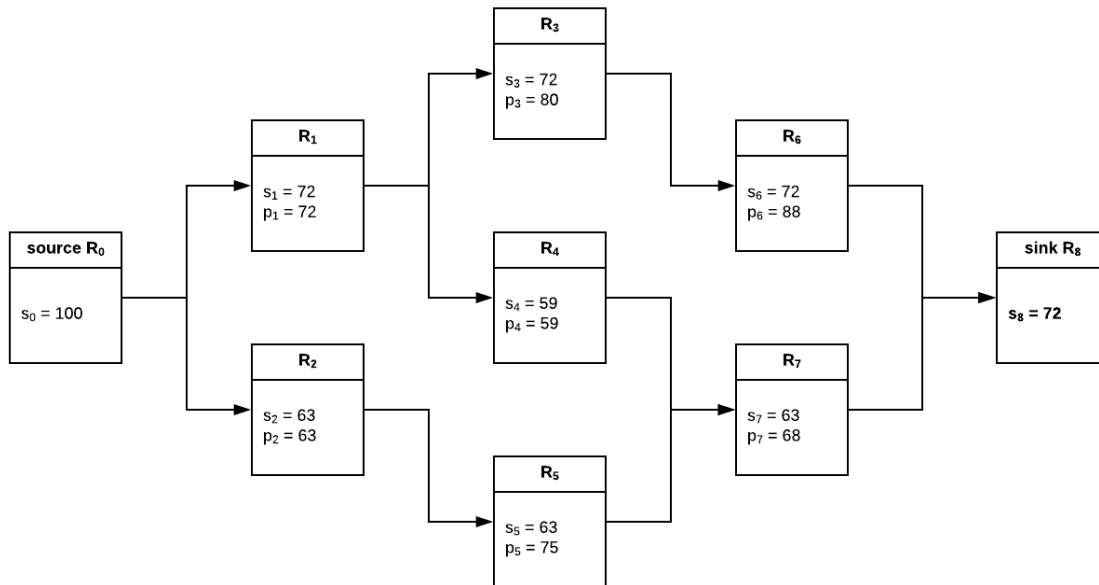
Σχήμα 5.2-10 Παράδειγμα: Βήμα 9

9. Βρισκόμαστε στον κόμβο R_7 . Συγκρίνουμε το p_7 με το s_5 . Βρίσκουμε ότι η ελάχιστη τιμή είναι το 63 και στην συνέχεια βλέπουμε ότι είναι μεγαλύτερη αυτή η τιμή από το παλιό s_7 . Επομένως $s_7 = 63$.



Σχήμα 5.2-11 Παράδειγμα: Βήμα 10

10. Επειδή έχουμε φτάσει στον τελικό κόμβο Sink R_8 ελέγχουμε την τιμή του s_6 και βλέπουμε ότι είναι μεγαλύτερη από το παλιό s_8 . Επομένως $s_8 = 72$.



Σχήμα 5.2-12 Παράδειγμα: Βήμα 11

11. Επειδή είμαστε και πάλι στον τελικό κόμβο Sink R_8 ελέγχουμε την τιμή του s_7 και βλέπουμε ότι δεν είναι μεγαλύτερη το παλιό s_8 . Επομένως παραμένει όπως πριν $s_8 = 72$.

Κεφάλαιο 6: Εκτίμηση απόδοσης

Στο κεφάλαιο αυτό καταγράφονται τα αποτελέσματα μιας σειράς δοκιμών που έγιναν, με σκοπό την εκτίμηση της απόδοσης του περιβάλλοντος-πλαισίου. Μετρήθηκε ο απαιτούμενος χρόνος για την εκτίμηση της εμπιστοσύνης σε μια υπηρεσία, πόρο, ή σύστημα, για διαφορετικούς γράφους απαιτήσεων, και όσον αφορά το μέγεθος του γράφου αλλά και σχετικά με το πλήθος και τους τύπους των διασυνδέσεων ανάμεσα στους κόμβους. Επιπλέον, ερευνάται η σταθερότητα του συστήματος, ελέγχοντας την ποσοστιαία μεταβολή του τελικού αποτελέσματος όταν μεταβάλλονται οι πιθανοτικές τιμές ικανοποίησης των λειτουργικών και μη λειτουργικών απαιτήσεων.

Δεδομένου ότι η απόδοση της διαδικασίας εξαγωγής πιθανοτικής τιμής ικανοποίησης από δέντρα στόχων έχει ερευνηθεί εκτεταμένα στο [31], επέλεξαμε ένα σταθερό μέγεθος δέντρου στόχων για κάθε απαίτηση. Το κάθε δέντρο, συγκεκριμένα, αποτελείται από 10 κόμβους, συνδεδεμένους σε συγκεκριμένη τοπολογία. Έτσι διευκολύνεται η αξιολόγηση της απόδοσης του TidalTrust αλγορίθμου για διαφορετικούς γράφους απαιτήσεων.

Τέλος, το περιβάλλον-σύστημα που αναπτύχθηκε δεν περιέχει πρόσθετα (plug-ins) για την συλλογή πληροφοριών από εξωτερικές πηγές. Οι τιμές, λοιπόν, των χαρακτηριστικών, που επηρεάζουν την πιθανοτική τιμή ικανοποίησης της κάθε απαίτησης, προέρχονται είτε από αρχεία διαμόρφωσης (configuration files) ή από μια συνάρτηση παραγωγής τυχαίων αριθμών.

6.1 Χρονική Απόδοση (Time Performance)

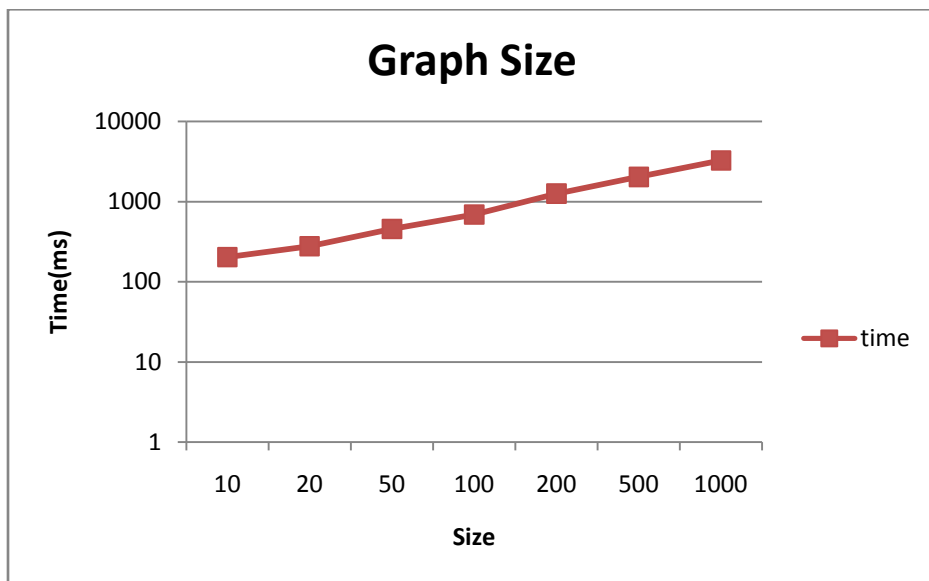
Σε αυτή την ενότητα αναλύεται η χρονική απόδοση του συστήματος για γράφους απαιτήσεων διαφορετικού :

- μεγέθους (size)
- πυκνότητας (density)
- μήκους μονοπατιού (path length)
- αριθμού κόμβων ανά επίπεδο (nodes/level)
- αριθμού συσσωρεύσεων (clusters)

6.1.1 Ανάλυση μεγέθους γράφου

Προκειμένου να αξιολογηθεί η χρονική απόδοση συναρτήσει του μεγέθους του γράφου, δημιουργήθηκαν γράφοι απαιτήσεων διαφορετικών μεγεθών με σταθερή πυκνότητα 0,5. Τα αποτελέσματα είναι τα ακόλουθα:

Μέγεθος	Χρόνος(ms)
10	203
20	277
50	456
100	686
200	1254
500	2023
1000	3241



Σχήμα 6.1-1 Διάγραμμα Χρόνου ως προς το Μέγεθος του γράφου

Από τα αριθμητικά στοιχεία, και το αντίστοιχο διάγραμμα, φαίνεται μια γραμμική αύξηση του χρόνου που απαιτείται για την εκτίμηση της εμπιστοσύνης σε μια υπηρεσία, πόρο, ή σύστημα, που ερευνάται, όσο αυξάνεται ο αριθμός των κόμβων που αποτελούν τον γράφο απαιτήσεων. Παρατηρούμε, επίσης, ότι, ακόμα και για περιπτώσεις 1000 κόμβων, οι χρόνοι εκτέλεσης είναι σε φυσιολογικά επίπεδα.

6.1.2 Ανάλυση πυκνότητας γράφου

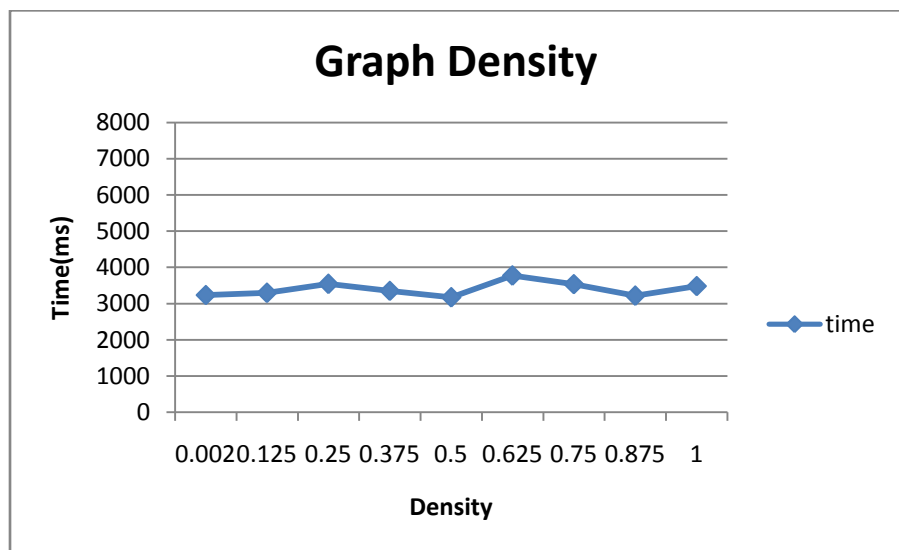
Η πυκνότητα ενός γράφου είναι ένα μέγεθος που περιγράφει την αναλογία πιθανών ακμών και ακμών που όντως υπάρχουν στον γράφο αυτό. Ως αναλογία, η μέγιστη τιμή που μπορεί να λάβει είναι 1, όταν όλοι οι κόμβοι είναι συνδεδεμένοι μεταξύ τους, και η ελάχιστη 0, όταν δεν υπάρχει καμία ακμή στον γράφο. Η πυκνότητα ορίζεται από τον ακόλουθο τύπο:

$$D = \frac{|E|}{|V|(|V| - 1)}$$

όπου E ο αριθμός των ακμών, που υπάρχουν στον γράφο, και V ο αριθμός των κόμβων.

Προκειμένου να αξιολογηθεί η χρονική απόδοση συναρτήσει της πυκνότητας του γράφου, δημιουργήθηκαν γράφοι απαιτήσεων διαφορετικής πυκνότητας με σταθερό αριθμό 1000 κόμβων. Τα αποτελέσματα είναι τα ακόλουθα:

Πυκνότητα	Χρόνος(ms)
0.002	3235
0.125	3296
0.25	3541
0.375	3355
0.5	3170
0.625	3773
0.75	3540
0.875	3215
1	3482



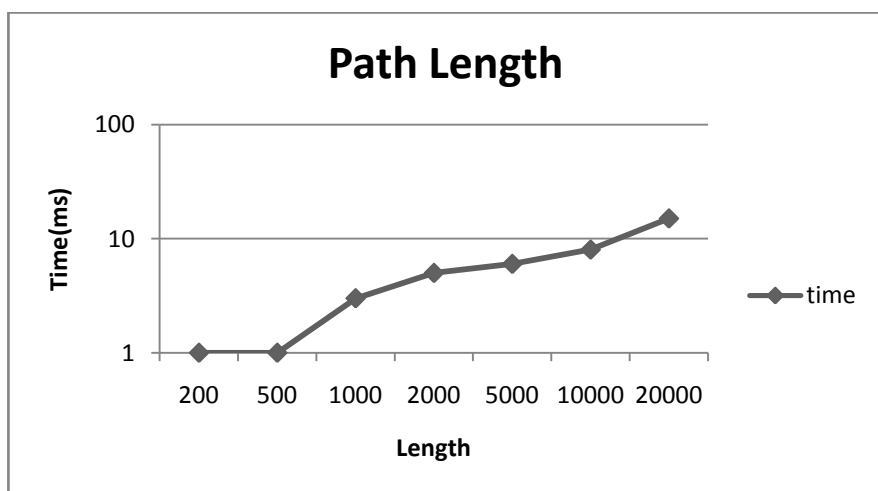
Σχήμα 6.1-2 Διάγραμμα Χρόνου ως προς την Πυκνότητα του γράφου

Από τα αριθμητικά στοιχεία, και το αντίστοιχο διάγραμμα, φαίνεται ότι ο χρόνος, που απαιτείται για την εκτίμηση της εμπιστοσύνης σε μια υπηρεσία, πόρο, ή σύστημα, που ερευνάται, δεν επηρεάζεται από την πυκνότητα του γράφου απαιτήσεων. Αυτό συνάδει με αυτό που αναμέναμε, δεδομένου ότι ο αλγόριθμος φέρει ομοιότητες με τον BFS(Breadth First Search) αλγόριθμο και τερματίζει μόλις ανακαλύψει το ελάχιστο μήκος μονοπατιού και τα μονοπάτια που έχουν αυτό το μήκος. Επομένως, επιπλέον ακμές δεν επηρεάζουν σημαντικά τον χρόνο εκτέλεσης.

6.1.3 Ανάλυση Μήκους Μονοπατιού

Προκειμένου να αξιολογηθεί η χρονική απόδοση συναρτήσει του μήκους μονοπατιού ανάμεσα στον αρχικό κόμβο *Source* και τον τελικό κόμβο *Sink*, δημιουργήθηκαν γράφοι απαιτήσεων διαφορετικού μήκους μονοπατιού. Χρησιμοποιήθηκαν 50000 κόμβοι, των οποίων ο χρόνος αρχικοποίησης ανέρχεται σε 125,5s κατά μέσο όρο. Τα αποτελέσματα αφορούν μόνο τον χρόνο, που αναλώνεται από τον αλγόριθμο TidalTrust πάνω στον γράφο απαιτήσεων, και είναι τα ακόλουθα:

Μήκος Μονοπατιού	Χρόνος(ms)
200	1
500	1
1000	3
2000	5
5000	6
10000	8
20000	15

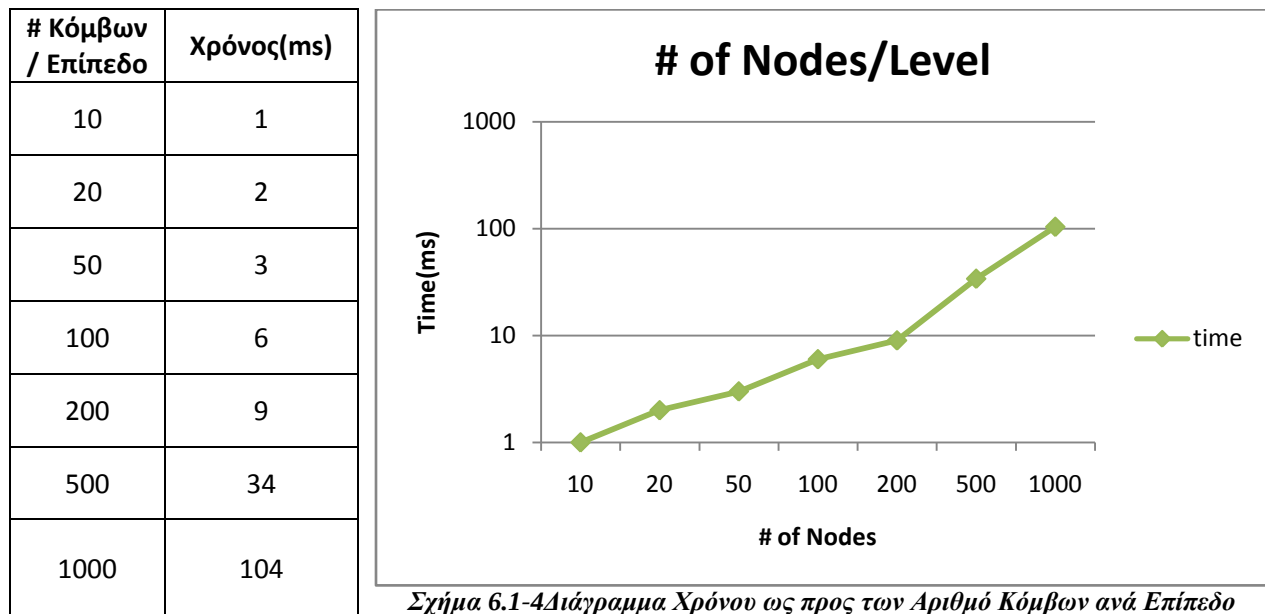


Σχήμα 6.1-3 Διάγραμμα Χρόνου ως προς το Μήκος Μονοπατιού

Αφαιρώντας τον χρόνο που απαιτείται για τον υπολογισμό των πιθανοτικών τιμών των απαιτήσεων και την αρχικοποίηση των κόμβων, βλέπουμε ότι ο χρόνος που απαιτείται για τον τελικό υπολογισμό της εμπιστοσύνης σε μια υπηρεσία, πόρο, ή σύστημα, είναι ελάχιστος. Παρατηρούμε δε, με βάση τα αριθμητικά στοιχεία και το αντίστοιχο διάγραμμα, ότι ο χρόνος αυξάνεται γραμμικά, όσο αυξάνεται το μήκος του μονοπατιού. Αυτή η συμπεριφορά είναι και η αναμενόμενη, δεδομένης της φύσης του αλγορίθμου, που αναζητά το ελάχιστο μονοπάτι.

6.1.4 Ανάλυση αριθμού Κόμβων ανά Επίπεδο

Προκειμένου να αξιολογηθεί η χρονική απόδοση συναρτήσει του αριθμού των κόμβων ανά επίπεδο του γράφου, δημιουργήθηκαν γράφοι απαιτήσεων με σταθερό μήκος ελάχιστου μονοπατιού 50 και μεταβλητό αριθμό κόμβων ανά επίπεδο. Η ανάλυση αυτού μεγέθους είναι εκ των πλέον σημαντικών, καθώς ο αλγόριθμος εκτελεί διάσχιση κατά πλάτος και επομένως η απόδοσή του επηρεάζεται από την συγκεκριμένη παράμετρο. Χρησιμοποιήθηκαν 50000 κόμβοι, των οποίων ο χρόνος αρχικοποίησης ανέρχεται σε 125,5s κατά μέσο όρο. Τα αποτελέσματα αφορούν μόνο τον χρόνο, που αναλώνεται από τον αλγόριθμο TidalTrust πάνω στον γράφο απαιτήσεων, και είναι τα ακόλουθα:



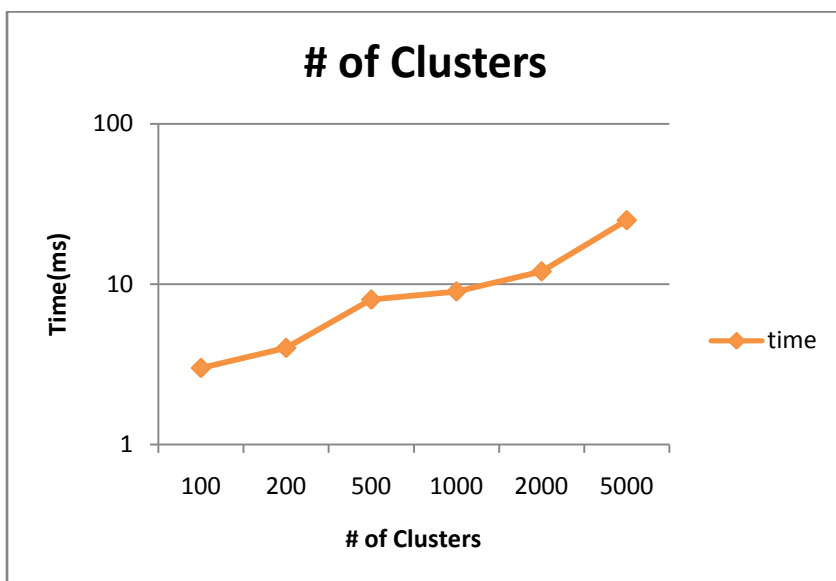
Όπως αναφέρθηκε και προηγουμένως, ο χρόνος, που απαιτείται για τον υπολογισμό των πιθανοτικών τιμών των απαιτήσεων και την αρχικοποίηση των κόμβων, είναι τάξεις μεγέθους μεγαλύτερος από αυτόν που αναλώνεται για τον υπολογισμό της τελικής εμπιστοσύνης στην υπηρεσία, πόρο, ή σύστημα, που ερευνάται. Τα αριθμητικά στοιχεία, και το αντίστοιχο

διάγραμμα, δείχνουν μια γραμμική αύξηση, όσο ο αριθμός των κόμβων παραμένει χαμηλός, καθώς, επίσης, και χαμηλούς χρόνους εκτέλεσης. Όταν, όμως, ο αριθμός των κόμβων ξεπεράσει τους 200, ο χρόνος εκτίμησης αυξάνεται με μεγαλύτερο ρυθμό. Παρ' όλα αυτά, παραμένει στο επίπεδο των μερικών ms κατά περίπτωση.

6.1.5 Ανάλυση αριθμού Συσσωρεύσεων

Προκειμένου να αξιολογηθεί η χρονική απόδοση συναρτήσει των αριθμών συσσωρεύσεων (clusters), με μεγάλη πυκνότητα, που υπάρχουν στον γράφο, δημιουργήθηκαν γράφοι απαιτήσεων με συσσωρεύσεις που αποτελούνται από 10 κόμβους και μεταβλητό αριθμό συσσωρεύσεων. Χρησιμοποιήθηκαν 50000 κόμβοι, των οποίων ο χρόνος αρχικοποίησης ανέρχεται σε 125,5s κατά μέσο όρο. Τα αποτελέσματα αφορούν μόνο τον χρόνο, που αναλώνεται από τον αλγόριθμο TidalTrust πάνω στον γράφο απαιτήσεων, και είναι τα ακόλουθα:

# Συσσωρεύσεων	Χρόνος(ms)
100	3
200	4
500	8
1000	9
2000	12
5000	25



Σχήμα 6.1-5 Διάγραμμα Χρόνου ως προς τον Αριθμό των Συσσωρεύσεων

Ακόμα και με δημιουργία αρκετών συσσωρεύσεων, με μεγάλη πυκνότητα, στον γράφο, οι χρόνοι εκτίμησης της τελικής εμπιστοσύνης στην υπηρεσία, πόρο, ή σύστημα, παραμένουν σε εξαιρετικά χαμηλά επίπεδα. Συγκεκριμένα, η αύξηση χρόνου είναι γραμμικώς ανάλογη με την αύξηση των clusters. Παρατηρούμε, λοιπόν, ότι οι συσσωρεύσεις δεν επηρεάζουν τον χρόνο εκτέλεσης. Στην ουσία, η αύξηση του χρόνου οφείλεται περισσότερο στην αύξηση του συνολικού μεγέθους του γράφου, παρά στην ύπαρξη των συσσωρεύσεων.

6.2 Σταθερότητα Συστήματος (Stability)

Σε αυτή την ενότητα εξετάζεται η σταθερότητα του περιβάλλοντος-πλαισίου, δοθέντων μεταβολών σε μία ή περισσότερες μεταβλητές που επηρεάζουν την εκτίμηση στην τελική εμπιστοσύνη στην υπηρεσία, πόρο, ή σύστημα, που ερευνάται.

*Ένα σύστημα ορίζεται ως **σταθερό** όταν, δεδομένης μιας εισόδου φραγμένου εύρους, το αποτέλεσμα που παράγεται είναι επίσης φραγμένο.*

Προκειμένου, λοιπόν να διαπιστωθεί η σταθερότητα (stability) του framework που δημιουργήσαμε, εκτελέσαμε μια σειρά από ελέγχους. Οι έλεγχοι αυτοί διενεργούνται μεταβάλλοντας ποσοστιαία τις πιθανοτικές τιμές ικανοποίησης των απαιτήσεων και, στην συνέχεια, υπολογίζοντας την ποσοστιαία μεταβολή του τελικού αποτελέσματος, που είναι η εμπιστοσύνη στην υπηρεσία, πόρο, ή σύστημα, και αναλύοντας την συμμόρφωση με τις αναμενόμενες τιμές. Στους ελέγχους χρησιμοποιούνται γράφοι με υψηλή πυκνότητα, προκειμένου να εξασφαλίσουμε την μέγιστη δυνατή διάδοση των μεταβληθέντων τιμών.

6.2.1 Μεταβολή μιας τιμής

Αρχικά μεταβάλλαμε ποσοστιαία κάθε πιθανοτική τιμή ικανοποίησης των απαιτήσεων ξεχωριστά και καταγράψαμε την ποσοστιαία μεταβολή του αποτελέσματος που προέκυψε. Για λόγους πληρότητας, βέβαια, επαναλάβαμε τα πειράματα σε γράφους διαφορετικών μεγεθών. Τα αποτελέσματα είναι τα ακόλουθα:

50 κόμβοι		1000 κόμβοι		5000 κόμβοι	
Μεταβολή εισόδου (%)	Μεταβολή εξόδου (%)	Μεταβολή εισόδου (%)	Μεταβολή εξόδου (%)	Μεταβολή εισόδου (%)	Μεταβολή εξόδου (%)
5	0 - 5	5	0 - 5	5	0 - 5
10	0 - 10	10	0 - 10	10	0 - 10
20	0 - 20	20	0 - 20	20	0 - 20
50	0 - 50	50	0 - 50	50	0 - 50

Τα αποτελέσματα που προέκυψαν είναι απολύτως λογικά και συνάδουν με τα αναμενόμενα. Ο αλγόριθμος σε κάθε βήμα υπολογίζει την ελάχιστη τιμή του μονοπατιού και τελικά επιλέγει το μονοπάτι με την μεγαλύτερη τιμή. Αν, επομένως, η μεταβλητή ανήκει στο μέγιστο μονοπάτι και είναι η ελάχιστη του μονοπατιού αυτού, τότε το αποτέλεσμα θα μεταβληθεί κατά το ίδιο ποσοστό. Αυτό, βέβαια, δεν ισχύει στην περίπτωση που η ποσοστιαία μεταβολή αλλάξει το μέγιστο μονοπάτι ή την ελάχιστη τιμή σε αυτό. Τότε η ποσοστιαία μεταβολή του αποτελέσματος θα ανήκει στο ανοιχτό διάστημα (0,P), όπου P η ποσοστιαία μεταβολή της πιθανοτικής τιμής της απαίτησης.

6.2.2 Μεταβολή πολλαπλών τιμών

Εν συνεχεία, μεταβάλλαμε ποσοστιαία τις πιθανοτικές τιμές ικανοποίησης πολλαπλών απαιτήσεων και καταγράψαμε ποσοστιαία μεταβολή του αποτελέσματος που προέκυψε. Χρησιμοποιήθηκε γράφος 1000 κόμβων για τις μετρήσεις. Τα αποτελέσματα είναι τα ακόλουθα:

Ανά δύο		Ανά τρεις		Ανά τέσσερις	
Μεταβολή εισόδων (%)	Μεταβολή εξόδου (%)	Μεταβολή εισόδων (%)	Μεταβολή εξόδου (%)	Μεταβολή εισόδων (%)	Μεταβολή εξόδου (%)
5	0 - 5	5	0 - 5	5	0 - 5
10	0 - 10	10	0 - 10	10	0 - 10
20	0 - 20	20	0 - 20	20	0 - 20
50	0 - 50	50	0 - 50	50	0 - 50

Τα αποτελέσματα είναι ακριβώς τα ίδια με την περίπτωση μιας μεταβλητής και, όπως και πριν, είναι σε πλήρη συμφωνία με αυτά που περιμέναμε θεωρητικά. Λόγω της φύσης του αλγορίθμου TidalTrust, η ποσοστιαία μεταβολή του αποτελέσματος θα είναι από 0, όταν η μεταβλητή δεν ανήκει στο μέγιστο μονοπάτι ακόμα και μετά την αλλαγή, μέχρι το P, που αντιστοιχεί στην ποσοστιαία μεταβολή της πιθανοτικής τιμής της απαίτησης. Οι ενδιάμεσες τιμές της μεταβολής προκύπτουν στην περίπτωση που η μεταβολή είτε αλλάζει το μονοπάτι που θεωρείται μέγιστο ή την μεταβλητή που θεωρείται ελάχιστη στο παρόν μονοπάτι.

Κεφάλαιο 7:Επίλογος

Στο κεφάλαιο αυτό γίνεται μια σύνοψη του περιβάλλοντος-πλαισίου, όπως περιγράφηκε στην διάρκεια της παρούσας διπλωματικής, και ανακεφαλαιώνονται συνοπτικά τα συμπεράσματα που εξήχθησαν ως προς την επίτευξη των στόχων που έχουν τεθεί. Τέλος, κλείνει το κεφάλαιο με μια αναφορά στις μελλοντικές προσθήκες και προεκτάσεις που μπορούν να γίνουν στο περιβάλλον-πλαίσιο.

7.1 Σύνοψη και συμπεράσματα

Στην παρούσα διπλωματική εργασία εξετάστηκε το πρόβλημα της εκτίμησης της εμπιστοσύνης σε μια υπηρεσία που προσφέρεται από έναν συγκεκριμένο πάροχο πάνω σε συγκεκριμένο περιεχόμενο, με δυνατότητα ορισμού σαφών απαιτήσεων που αφορούν την υπηρεσία αυτή. Η παροχή μιας αποδοτικής λύσης στο πρόβλημα αυτό θα παρέχει την δυνατότητα γρήγορης, αυτοματοποιημένης και, πάνω απ' όλα, αξιόπιστης αξιολόγησης μιας υπηρεσίας, η οποία ενδιαφέρει τον χρήστη. Τελικά το αποτέλεσμα θα είναι η ευκολότερη αξιοποίηση υπηρεσιών από χρήστες που μέχρι πρότινος είχαν λίγη, ή καθόλου, πρόσβαση σε πληροφόρηση για την εμπιστοσύνη που αυτή απολαμβάνει.

Σκοπός, λοιπόν, της διπλωματικής εργασίας είναι η προδιαγραφή και κατασκευή ενός περιβάλλοντος-πλαισίου, το οποίο επιτρέπει τον ορισμό της εμπιστοσύνης που απολαμβάνει μια υπηρεσία, και στην συνέχεια τον υπολογισμό μιας πιθανοτικής εκτίμησης για την εμπιστοσύνη αυτή. Συγκεκριμένα, δυνατότητες ορισμού παρέχονται τόσο για την συνολική εμπιστοσύνη, δίνοντας την ελευθερία στον χρήστη να δηλώσει τις απαιτήσεις που έχει από την υπηρεσία, αλλά και την μεταξύ τους αλληλεπίδραση, όσο και για την ξεχωριστή παραμετροποίηση της εκάστοτε απαίτησης, δηλαδή των πληροφοριών που την απαρτίζουν και τον τρόπο που αυτές συνδέονται μεταξύ τους.

Στα παραπάνω κεφάλαια παρουσιάστηκε η δομή της αρχιτεκτονικής και η ουσιαστική λειτουργία του περιβάλλοντος-πλαισίου. Σημαντικά σημεία ενδιαφέροντος που εμφανίστηκαν κατά την διάρκεια εκπόνησης της εργασίας είναι τα ακόλουθα:

- Αρχιτεκτονική του περιβάλλοντος πλαισίου: Η ανάπτυξη του πλαισίου βασίστηκε στα σχεδιαστικά μορφήματα Μαυροπίνακα και Publisher/Subscriber. Η επιλογή αυτής της αρχιτεκτονικής επιτρέπει την ανακάλυψη μεταβολών, σε πληροφορίες και μοντέλα, και τον επαναυπολογισμό μόνο των κομματιών που επηρεάζονται από αυτές. Ένα άλλο πλεονέκτημά της είναι η ευελιξία που παρέχεται όσον αφορά την προσθήκη μελλοντικών επεκτάσεων.
- Μοντελοποίηση απαιτήσεων: Για κάθε υπηρεσία παρέχεται ένας ορισμός εμπιστοσύνης, που αποτελείται από ένα σύνολο απαιτήσεων. Κάθε απαίτηση συνοδεύεται από μια αντίστοιχη πιθανοτική τιμή ικανοποίησής της. Για να υπολογιστεί, λοιπόν, αυτή η τιμή, η απαίτηση μοντελοποιείται με χρήση δέντρων στόχων. Η προσέγγιση αυτή παρέχει έναν εύκολο τρόπο ορισμού της απαίτησης, με την ανάλυσή της σε παρεχόμενες πληροφορίες και συνεισφορές ανάμεσα στις πληροφορίες αυτές. Στην συνέχεια, το δέντρο στόχων μπορεί εύκολα να

μετατραπεί σε προτασιακή λογική και τέλος σε CNF μορφή, προκειμένου να γίνει ο υπολογισμός της πιθανοτικής τιμής ικανοποίησης της απαίτησης.

- Εκτίμηση εμπιστοσύνης: Βασικός στόχος του περιβάλλοντος-πλαισίου που προτάθηκε είναι η εκτίμηση της συνολικής εμπιστοσύνης σε μια υπηρεσία. Προκειμένου αυτή η εκτίμηση να γίνει δυνατή, προτείνεται η χρήση του αλγόριθμου TidalTrust, ο οποίος χρησιμοποιήθηκε αρχικά σε κοινωνικά δίκτυα, αλλά η φύση του προβλήματος τον καθιστά κατάλληλο και για αυτή την περίπτωση. Η επιλογή αυτού του αλγόριθμου δίνει, επιπλέον, και την δυνατότητα για επεκτάσεις που θα αναφερθούν στην επόμενη ενότητα.
- Αξιολόγηση περιβάλλοντος-πλαισίου: Το περιβάλλον-πλαίσιο αξιολογήθηκε αρχικά με βάση την χρονική απόδοση. Ο αριθμός των απαιτήσεων είναι η παράμετρος που επηρεάζει σε μέγιστο βαθμό τον χρόνο εκτίμησης της εμπιστοσύνης. Αύξηση της, επιπλέον, προκαλεί ανάλογη αύξηση του συνολικού χρόνου. Η πυκνότητα του γράφου και το πλήθος κόμβων ανά επίπεδο επηρεάζουν επίσης τον χρόνο, αλλά είναι αμελητέες ποσότητες σε σχέση με τον αριθμό των κόμβων. Πιθανές μεταβολές σε άλλα χαρακτηριστικά του γράφου απαιτήσεων δεν επηρεάζουν σημαντικά τους χρόνους εκτέλεσης.

7.2 Μελλοντικές επεκτάσεις

Όπως αναφέρθηκε και σε προηγούμενες ενότητες, το περιβάλλον-πλαίσιο αναπτύχθηκε με βασικούς πυλώνες την ευελιξία και την δυνατότητα επέκτασης. Επομένως, μελλοντικές αλλαγές, επεκτάσεις και βελτιστοποιήσεις είναι εύκολο να υλοποιηθούν και να ενσωματωθούν στο παρόν πλαίσιο.

Σημείο που θα μπορούσε, αρχικά, να βελτιωθεί σε μελλοντική επέκταση του περιβάλλοντος-πλαισίου είναι ο ορισμός των απαιτήσεων και η αλληλεπίδραση των εξωτερικών πληροφοριών. Με βάση την τωρινή υλοποίηση, κάθε απαίτηση μοντελοποιείται με χρήση ενός ξεχωριστού δέντρου στόχων. Συνεισφορές μπορούν να δημιουργηθούν ανάμεσα στους κόμβους του συγκεκριμένου δέντρου και μόνο. Στην πραγματικότητα, όμως, ενδεχόμενη αλλαγή σε μια πληροφορία που αποτελεί μια απαίτηση μπορεί να επηρεάσει μια άλλη πληροφορία, που είναι κομμάτι μιας εντελώς διαφορετικής απαίτησης. Θα ήταν, λοιπόν, εξαιρετικά χρήσιμη η προσθήκη της δυνατότητας να ορίζονται και συνεισφορές μεταξύ κόμβων που ανήκουν σε διαφορετικά δέντρα στόχων.

Μια άλλη επέκταση που προτείνεται είναι του γράφου εκτίμησης εμπιστοσύνης στην υπηρεσία. Συγκεκριμένα, ο γράφος αυτός ορίζει την εμπιστοσύνη μιας υπηρεσίας, λαμβάνοντας, όμως, υπ' όψιν ότι η υπηρεσία παρέχεται από έναν συγκεκριμένο πάροχο, εντελώς αποκομμένο από τους υπόλοιπους παρόχους υπηρεσιών. Αυτή η σύμβαση περιορίζει τις δυνατότητες του περιβάλλοντος-πλαισίου, διότι ο πάροχος είναι κομμάτι ενός μεγαλύτερου δικτύου υπηρεσιών, οι οποίες αλληλεπιδρούν με την δική του υπηρεσία. Αυτές οι αλληλεπιδράσεις επηρεάζουν άμεσα και έμμεσα την εκτίμηση της εμπιστοσύνης. Προτείνεται η επέκταση των εργαλείων ορισμού της εμπιστοσύνης, έτσι ώστε να περιλαμβάνουν την δυνατότητα ορισμού ενός πολύ

μεγαλύτερου δικτύου που προσιδιάζει σε ένα κοινωνικό δίκτυο παρόχων. Η επιλογή, άλλωστε, του αλγόριθμου TidalTrust έγινε με την προοπτική αυτή.

Προτείνεται, επιπλέον, η προσαρμογή του αλγόριθμου εκτίμησης εμπιστοσύνης, με τέτοιο τρόπο, ώστε η πιθανή αλλαγή της τιμής μιας απαίτησης, ή η προσθήκη μιας νέας, να μην απαιτεί επαναυπολογισμό με χρήση όλου του γράφου, αλλά μόνο του τμήματος που επηρεάζεται. Κάτι τέτοιο θα επιταχύνει αρκετά το σύστημα και θα δώσει την δυνατότητα ορισμού πιο περίπλοκων δικτύων, που παρόλα αυτά θα έχουν απόκριση πραγματικού χρόνου.

Τέλος, μπορεί μια μελλοντική επέκταση να περιλαμβάνει ένα πιο αποδοτικό υποσύστημα λήψης εξωτερικών πληροφοριών και διαχείρισής τους. Κάτι τέτοιο θα επέτρεπε την χρήση του περιβάλλοντος-πλαισίου σε πραγματικές συνθήκες και από μεγαλύτερο εύρος χρηστών.

Κεφάλαιο 8: Βιβλιογραφία

- [1] O.M.G., “Meta Object Facility (MOF) Core Specification,” Management, 2006.
- [2] A.C. Completion and C. Outline, “MOF Essentials,” Library
- [3] O.M.G., “OMG Systems Modeling Language (OMG SysML TM),” Source, 2010.
- [4] E. Foundation. (2009, Oct.) Eclipse Modeling Framework. [Online]. <http://www.eclipse.org/modeling/emf/>
- [5] A. Lapouchnian. Goal-Oriented Requirements Engineering: An Overview of the Current Research. Depth Report, University of Toronto, 2005.
- [6] Yu E, Mylopoulos J. Why goal-oriented requirements engineering. In: Dubois E, Opdahl AL, Pohl K (eds). Proceedings of the 4th international workshop on requirements engineering: foundations of software quality, Pisa, Italy, 8–9 June 1998, Pisa, Italy. Presses Universitaires de Namur, 1998, pp 15–22
- [7] Dardenne, A., van Lamsweerde A. and Fickas, S., “Goal Directed Requirements Acquisition”, Science of Computer Programming, Vol. 20, No. 1-2, pp. 3–50, 1993
- [8] ITU - Telecommunication Standardization Sector Draft Specification of the Goal-oriented Requirement Language (Z.151), September 2013.
- [9] Anton, A.I., Goal Identification and Refinement in the Specification of Software-Based Information Systems, Ph.D. Dissertation, Georgia Institute of Technology, Atlanta GA, 1997.
- [10] Mylopoulos, J., Kolp, M., and Castro, J. “UML for Agent-Oriented Software Development: The Tropos Proposal,” in Proceedings of UML 2001.
- [11] Rolland, C., Souveyet, C., and Ben Achour, C. “Guiding goal modeling using scenarios,” IEEE Trans. Software Eng., vol. 24, pp. 1055–1071, Dec. 1998.91
- [12] Yu, E.S.K. “Towards modelling and reasoning support for early-phase requirements engineering,” in Proceedings of the Third IEEE International Symposium on Requirements Engineering, Annapolis, Maryland, January 1997
- [13] J. Mylopoulos, L. Chung, and E. Yu, "From object-oriented to goal-oriented requirements analysis," Commun. ACM, vol. 42, no. 1, pp. 31--37, 1999.
- [14] Amit K. Chopra, Fabiano Dalpiaz, Paolo Giorgini, and John Mylopoulos. Modeling and reasoning about service-oriented applications via goals and commitments. In Proceedings of the 22nd international conference on Advanced information systems engineering, CAiSE'10, pages 113—128. Springer-Verlag, 2010

- [15] Amit K. Chopra, Fabiano Dalpiaz, Paolo Giorgini, and John Mylopoulos. Reasoning about agents and protocols via goals and commitments. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1, AAMAS '10, pages 457--464, 2010.
- [16] Amit K. Chopra, John Mylopoulos, Fabiano Dalpiaz, Paolo Giorgini, and unindar P. Singh. Requirements as Goals and Commitments too. In *Intentional Perspectives on Information Systems Engineering*, chapter 8, pages 137--153. Springer, 2010.
- [17] Y. Wang, S. a McIlraith, Y. Yu, and J. Mylopoulos, "An automated approach to monitoring and diagnosing requirements," *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering - ASE '07*, 2007, p. 293.
- [18]http://en.wikipedia.org/wiki/First-order_logic
- [19] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107--136, 2006.
- [20] D. Pedro and D. Lowd, *Markov Logic: An Interface Layer for Artificial Intelligence*, Morgan & Claypool, 2009.
- [21] Tivadar Papai, Parag Singla, and Henry Kautz. Constraint propagation for efficient inference in markov logic. In *Proceedings of 17th International Conference on Principles and Practice of Constraint Programming (CP 2011)*, number 6876 in *Lecture Notes in Computer Science (LNCS)*, pages 691—705, 2011.
- [22] Pedro Domingos Marc Sumner. The alchemy tutorial [online]. <http://alchemy.cs.washington.edu/tutorial/tutorial.pdf>.
- [23] Eric Yu, "Information Systems (in the Internet Age)". In *Practical Handbook of Internet Computing*, M.P. Singh, ed. 2004 CRC Press.
- [24] Sherchan, W., Nepal, S., and Paris, C. 2013. A Survey of trust in social networks. *ACM Comput. Surv.* 45, 4, Article 47 (August 2013), 33 pages.
- [25] Morteza Ghorbani Moghaddam, Norwati Mustapha, Aida Mustapha, Nurfadhlin Mohd Sharef, Anousheh Elahian, "TRUST METRICS IN RECOMMENDER SYSTEMS: A SURVEY". In *Advanced Computational Intelligence: An International Journal (ACII)*, Vol.2, No.3, July 2015
- [26] Cai-Nicolas Ziegler and Georg Lausen, "Propagation Models for Trust and Distrust in Social Networks". In *Information Systems Frontiers* 7:4/5, 337–358, 2005

[27] Lesani M. and Montazeri N., “FUZZY TRUST AGGREGATION AND PERSONALIZED TRUST INFERENCE IN VIRTUAL SOCIAL NETWORKS” . In Computational Intelligence, Volume 25, Number 2, 2009

[28] Golbeck, J. 2005. Computing and Applying Trust in Webbased Social Networks. Ph.D. Dissertation, University of Maryland, College Park, MD, USA.

[29] U. Kuter and J. Golbeck, “Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models,” in AAAI’07 Proceedings of the 22nd national conference on Artificial intelligence, 2007, vol. 2, pp. 1377–1382.

[30] M. Qin, Q. Yang, and F. Fu, “A context-aware media content personalized recommendation for community networks,” 2013 Int Conf Wirel Commun Signal Process, no. 2, pp. 1–6, Oct. 2013.

[31] Γ. Παπαδόπουλος, Κ. Κοντογιάννης, «Περιβάλλον-Πλαίσιο Προσαρμοστικής Παρακολούθησης Συστημάτων», Διπλωματική εργασία για την σχολή Η.Μ.Μ.Υ Ε.Μ.Π, Φεβρουάριος 2014

[32] G. Chatzikonstantinou, K. Kontogiannis, “Run-Time Requirements Verification for Reconfigurable Systems”. In Information and Software Technology, vol. 75, pages 105-121, July 2016