ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

Μοντέλο Εμπιστοσύνης για τα Κοινωνικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιλεάνας Δρίβα

Επιβλέπων: Π. Στεφανέας
Λέκτορας Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2015

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

Μοντέλο Εμπιστοσύνης για τα Κοινωνικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιλεάνας Δρίβα

Επιβλέπων: Π. Στεφανέας
Λέκτορας Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2015

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:

Π. Στεφανέας	Ν. Παπασπύρου	Σ. Παπαβασιλείου
Λέκτορας Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.

Ιλεάνα Δρίβα
Διπλωματούχος Σχολής Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών, Ε.Μ.Π.

Ευχαριστίες

Μετά την ολοκλήρωση της διπλωματικής εργασίας αισθάνομαι την ανάγκη να ευχαριστήσω ειλικρινά τον επιβλέποντα κ. Πέτρο Στεφανέα, Λέκτορα του τομέα Μαθηματικών της Σχολής Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών του Ε.Μ.Π., ο οποίος με εμπιστεύθηκε, μου έδωσε την ευκαιρία να συνεργαστώ μαζί του και να ασχοληθώ με ένα τόσο πρωτοποριακό θέμα με πολλαπλές δυνατότητες ανάπτυξης και επέκτασης. Η πολύτιμη υποστήριξη, η συνεχής συμπαράσταση και καθοδήγησή του, ο χρόνος που αφιέρωσε και οι εύστοχες υποδείξεις του σε όλα τα στάδια της παρούσας διπλωματικής εργασίας υπήρξαν βασικοί παράγοντες της επιτυχούς ολοκλήρωσής της.

Επίσης, οφείλω να ευχαριστήσω την υποφήφια διδάκτορα κ. Κατερίνα Ξύστρα για τη βοήθειά της στο πρακτικό κομμάτι της εργασίας καθώς και τον διδάκτορα κ. Νίκο Τριανταφύλλου για τις παρατηρήσεις του.

Αφιερώνω την εργασία αυτή σε αυτούς που πάντα πίστευαν σε μένα και μου έμαθαν να βασίζομαι στις δικές μου δυνάμεις, να ξεπερνάω τα εμπόδια και τις δυσκολίες και να προχωράω με βάση το δίκαιο και την καλοσύνη. Ευχαριστώ τους γονείς και τον αδερφό μου για την ηθική, και όχι μόνο, υποστήριξη και συμπαράστασή τους.

Περίληψη

Στις μέρες μας τα κοινωνικά δίκτυα αποτελούν ένα αναπόσπαστο κομμάτι της καθημερινότητας όλο και περισσότερων ανθρώπων, καθώς προσφέρουν ποικίλες δυνατότητες στους χρήστες τους. Ενδεικτικά, διαθέτουν υπηρεσίες, οι οποίες βασισμένες στις πιο σύγχρονες τεχνολογίες του Διαδικτύου, επιτρέπουν την ανταλλαγή κάθε είδους πληροφορίας και εξυπηρετούν στην έμφυτη ανάγκη του ανθρώπου για κοινωνικοποίηση. Τα άτομα μέσα από τα κοινωνικά δίκτυα δημιουργούν δεσμούς προκειμένου να επικοινωνούν μεταξύ τους και να έχουν πρόσβαση σε κοινές πληροφορίες. Στο πλαίσιο αυτό, η δημιουργία δεσμών εμπιστοσύνης μεταξύ των χρηστών είναι αναγκαία προϋπόθεση για την ομαλή λειτουργία του κοινωνικού δικτύου στο σύνολό του.

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι μία πρόταση για τη μοντελοποίηση κοινωνικών δικτύων βασισμένη στην εμπιστοσύνη που αναπτύσσουν οι χρήστες τους. Προτείνεται λοιπόν ένα γενικό μοντέλο που εξαρτάται από την αποτελεσματική συνεργασία των χρηστών του κοινωνικού δικτύου με σκοπό να είναι όσο το δυνατόν πιο αξιόπιστο και αποδοτικότερο στους σκοπούς για τους οποίους έχει δημιουργηθεί. Για τη συγκεκριμένη μοντελοποίηση χρησιμοποιήθηκε η γλώσσα αλγεβρικών προδιαγραφών CafeOBJ που ανήκει στις τυπικές μεθόδους σχεδιασμού συστημάτων και υποστηρίζει τη συμπεριφοριακή προδιαγραφή μέσω τελεστών, τύπων και εξισώσεων. Πιο συγκεκριμένα χρησιμοποιήθηκε η μέθοδος ΠΣΜ (Παρατηρήσιμα Συστήματα Μετάβασης - OTS) για τη σύνθεση των τμημάτων προδιαγραφών και την τελική περιγραφή των συμπεριφοριακών αντικειμένων που χαρακτηρίζουν την επιθυμητή μοντελοποίηση.

Σχετικά με τη δομή της διπλωματικής εργασίας, στο πρώτο κεφάλαιο γίνεται μία αρχική εισαγωγή στα κοινωνικά δίκτυα και πως η εμπιστοσύνη πρέπει να εφαρμόζεται σε αυτά ως βασικό τους συστατικό. Στο δεύτερο κεφάλαιο, γίνεται μια εισαγωγή στα γνωστά δίκτυα ad hoc και παρουσιάζεται το πρωτόκολλο CONFIDANT, από το οποίο ξεκίνησε η ιδέα για τη μοντελοποίηση που προτείνεται στα κοινωνικά δίκτυα. Το τρίτο κεφάλαιο αποτελεί το θεωρητικό κομμάτι της μοντελοποίησης στα κοινωνικά δίκτυα, δηλαδή την προσαρμογή του πρωτόκολλου CONFIDANT από τα ad hoc στα κοινωνικά δίκτυα. Στο τέταρτο κεφάλαιο υπάρχει η περιγραφή της γλώσσας CafeOBJ καθώς και αναλυτικά η μέθοδος ΠΣΜ. Τέλος, στο πέμπτο κεφάλαιο υπάρχει η εφαρμογή της CafeOBJ/OTS στη μοντελοποίηση για τα κοινωνικά δίκτυα

Λέξεις Κλειδιά

Κοινωνικά Δίκτυα, εμπιστοσύνη, CafeOBJ, Παρατηρήσιμα Συστήματα Μετάβασης, γλώσσες αλγεβρικών προδιαγραφών, πρωτόκολλο CONFIDANT

ABSTRACT

Nowadays, social networks are an integral part of everyday life for more and more people, due to the fact that they offer a wide range of capabilities to their users. More specifically, they provide services, which are based on the most modern technologies of the Internet, allow the exchange of information and service the inherent need of man to socialise. People through social networks create links in order to communicate with each other and have access to the same information. In this context, the establishment of trustworthy links between users is a prerequisite for the proper functioning of the social network as a whole.

The scope of this thesis is to propose a formalisation for social networks based on trust developed by users. Therefore, a general model is proposed, which depends on the efficient cooperation of the users of the social network in order to be ensured that is reliable and effective for the purposes of its creation. For this system modeling the algebraic specification language CafeOBJ was used, which is one of the standard formal methods of systems design and supports the behavioural specification techniques through operators, types and equations. The OTS method was also used for the composition of the specification modules and the final description of behavioural objects which describe the wanted system.

According to the structure of the thesis, the first chapter is an initial introduction to social networks and how the trust should be applied to these as their main feature. The second chapter is a description to ad hoc networks and the CONFIDANT protocol, from which the idea was born for the trust model proposed in social networks. The third chapter is the theoretical part of how the trust model is working on social networks, i.e. the adaptation of protocol CONFIDANT by the ad hoc networks in social networks. In the fourth chapter there is a presentation of the language CafeOBJ and the OTS method. Finally, the fifth chapter is the application of CafeOBJ/OTS on the social networks trust model.

Keywords

Social Networks, trust, CafeOBJ, OTS, Algebraic Specification Languages, Observational Transition Systems, CONFIDANT protocol

Περιεχόμενα

Κεφάλαιο 3

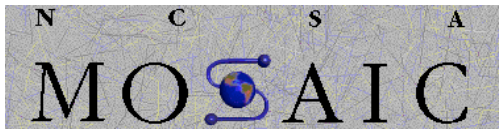
Κεφάλαιο 4

Κεφάλαιο 1

1.1 Ο Παγκόσμιος Ιστός και τα Κοινωνικά Δίκτυα

Η τεχνολογία του Παγκόσμιου Ιστού (ή σύντομα Ιστού) δημιουργήθηκε το 1989 από τον Βρετανό Τίμοθι Τζον Μπέρνερς-Λι (Sir Timothy John Berners-Lee) και ονομάστηκε WWW: World Wide Web [1-2]. Αυτό που οδήγησε τον Lee στην εφεύρεση του Παγκόσμιου Ιστού ήταν το όραμά του για ένα κόσμο όπου ο καθένας θα μπορούσε να ανταλλάσσει πληροφορίες άμεσα προσβάσιμες από τους υπολοίπους.

Βασικό χαρακτηριστικό του WWW είναι ότι αρκεί μία απλή σύνδεση στο Διαδίκτυο και ένα πρόγραμμα για πλοήγηση ή φυλλομέτρηση ιστοσελίδων (συνήθως διατίθονται δωρεάν ή περιλαμβάνονται εξαρχής στο λειτουργικό σύστημα του υπολογιστή), και ο χρήστης μπορεί να χρησιμοποιήσει τον Ιστό από όπου και αν βρίσκεται. Το πρώτο ιδιαίτερα δημοφιλές πρόγραμμα πλοήγησης (web browser), για το λειτουργικό σύστημα των Windows, του Παγκόσμιου Ιστού δημιουργήθηκε το 1993, από τους Marc Andreessen και Eric Bina και το ονόμασαν Mosaic (ή NCSA Mosaic [3] γιατί υλοποιήθηκε στο National Center for Supercomputing Applications) με μία σειρά από επιπλέον δυνατότητες όπως υποστήριξη γραφικών, ήχου και video, ηλεκτρονικό σελιδοδείκτη, κ.α. Για το λόγο αυτό, το Mosaic ήταν το πρόγραμμα περιήγησης που οδήγησε στην επανάσταση της επικοινωνίας μέσω του Διαδικτύου τη δεκαετία του 1990.



Το 2004 οι Dale Dougherty και O' Reilly VP, παρατήρησαν ότι το διαδίκτυο είχε αρχίσει να γίνεται εκτός από δημοφιλές και αρκετά σημαντικό κομμάτι της καθημερινότητας όλο και μεγαλύτερου ποσοστού ανθρώπων. Συνεχώς έβγαιναν νέες εφαρμογές και ιστοσελίδες οι οποίες αναγνωρίζονταν από το ευρύ κοινό σε σύντομο χρονικό διάστημα. Έτσι, η έλευση μίας νέας γενιάς Διαδικτύου, του ονομαζόμενου Νέου Παγκόσμιου Ιστού ή αλλιώς Web 2.0 [4] ήταν κάτι αναπόφευκτο. Στον Παγκόσμιο Ιστό μόνο κάποιος διαχειριστής μπορούσε να προσθέσει περιεχόμενο σε κάποιο δικτυακό τόπο ενώ με τον Web 2.0 ο χρήστης είναι σε θέση να αλλάξει τόσο το περιβάλλον της σελίδας όσο και να παρέμβει στο περιεχόμενό της. Η καινοτομία λοιπόν του Νέου Παγκόσμιου Ιστού είναι ότι καθιερώθηκε μία διαδραστική πλατφόρμα που υποστηρίζει την επικοινωνία εκατομμυρίων χρηστών σε οικουμενικό επίπεδο και παρέχει τα εργαλεία για ευρεία συμμετοχή στη δημοσίευση ψηφιακού περιεχομένου (user generated content).

Με τη δυνατότητα που έφερε το Web 2.0 ώστε οι ιστοσελίδες να είναι και διαδραστικές στο χρήστη και την καθιέρωση νέων υπηρεσιών το Διαδίκτυο έγινε μία μεγάλη πλατφόρμα επικοινωνίας με χαρακτήρα χρηστο-κεντρικό. Αποτέλεσμα ήταν το παραδοσιακό WWWeb να αλλάξει μορφή και να γίνει κάτι πολύ περισσότερο από ένα μέσο για εύκολη αναζήτηση πληροφοριών. Οι χρήστες το είδαν σαν μία ευκαιρία ώστε να συνδεθούν μεταξύ τους, στην αρχή κυρίως για επαγγελματικούς και εμπορικούς σκοπούς και στη συνέχεια για να αναπτύξουν διαπροσωπικές σχέσεις. Η επιθυμία τους αυτή σε συνδυασμό με τις

δυνατότητες που τους έδινε η νέα τεχνολογία έφερε τη δημιουργία των ψηφιακών κοινωνικών δικτύων (Online Social Networks, OSNs), ή αλλιώς απλά για συντομία τα λέμε και κοινωνικά δίκτυα (Social Networks, SNs), όπου, όπως είναι φυσικό, το ιδιαίτερο χαρακτηριστικό τους είναι ότι η ανάπτυξη τους ξεκίνησε από τους ίδιους τους χρήστες.

«Τα κοινωνικά δίκτυα γεννήθηκαν από κάτω προς τα πάνω, δεν είναι προϊόν κάποιας εταιρείας και άλλαξαν το Διαδίκτυο ποικιλοτρόπως», εξηγεί ο Δρ. Γιώργος Μητακίδης, καθηγητής του Πανεπιστημίου Πατρών και επικεφαλής του RISEPTIS - Research and Innovation for Security, Privacy and Trustworthiness in the Information Society. Με την έννοια αυτή, η εξέλιξη του Παγκόσμιου Ιστού, ως προς τη δομή και το περιεχόμενο, έφερε το χρήστη στο επίκεντρο να αντιμετωπίζει την νέα αυτή τεχνολογία βάση των αναγκών του για επικοινωνία και αλληλεπίδραση.

Ο άνθρωπος είναι από τη φύση του κοινωνικό ον, έχει δηλαδή την ανάγκη να εντάσσεται σε κοινωνικές ομάδες. Έτσι, με το πέρασμα του χρόνου τα κοινωνικά δίκτυα αναπτύσσονται όπως και μία πραγματική κοινότητα. Για να το καταλάβει αυτό κανείς καλύτερα μπορεί να σκεφτεί πώς λειτουργούν οι κοινωνικές ομάδες, όπου τα μέλη τους έχουν κάποιες κοινές πεποιθήσεις, στόχους, πιστεύω, αναπτύσσουν μαζί δραστηριότητες και εξελίσσονται κάτω από κοινές αρχές. Αντίστοιχα, στα κοινωνικά δίκτυα δημιουργούνται ομάδες, μικρότερες, όπως μία ομάδα μίας τάξης ενός πανεπιστημίου, ή μεγαλύτερες, όπως μία παγκόσμια ομάδα ενημέρωσης για τεχνολογικά νέα, που σκοπό έχουν να μοιράζονται πληροφορίες πάνω σε κοινά θέματα. Οι πληροφορίες αυτές δημοσιεύονται με τρόπο εύκολο και απλό στα κοινωνικά δίκτυα και μέσω αυτών οι χρήστες αναπτύσσουν σχέσεις ανεξάρτητα της γεωγραφικής τους απόστασης. Στόχος τελικά της κοινωνικής δικτύωσης είναι η δημιουργία μίας ομάδας χρηστών που επικοινωνούν μέσω του Διαδικτύου, χωρίς κανένα γεωγραφικό περιορισμό, για διαμοιρασμό πληροφοριών και δεδομένων μέσω της μεταξύ τους αλληλεπίδρασης. Τελικά, καταρρίπτουν την προϋπόθεση της φυσικής συνύπαρξης των μελών της κοινωνίας για την ύπαρξη αυτής, κάτι για το οποίο παλαιότερα δεν υπήρχε τρόπος να γίνει εφικτό.

1.2 Τα Κοινωνικά Δίκτυα

1.2.1 Ορισμός

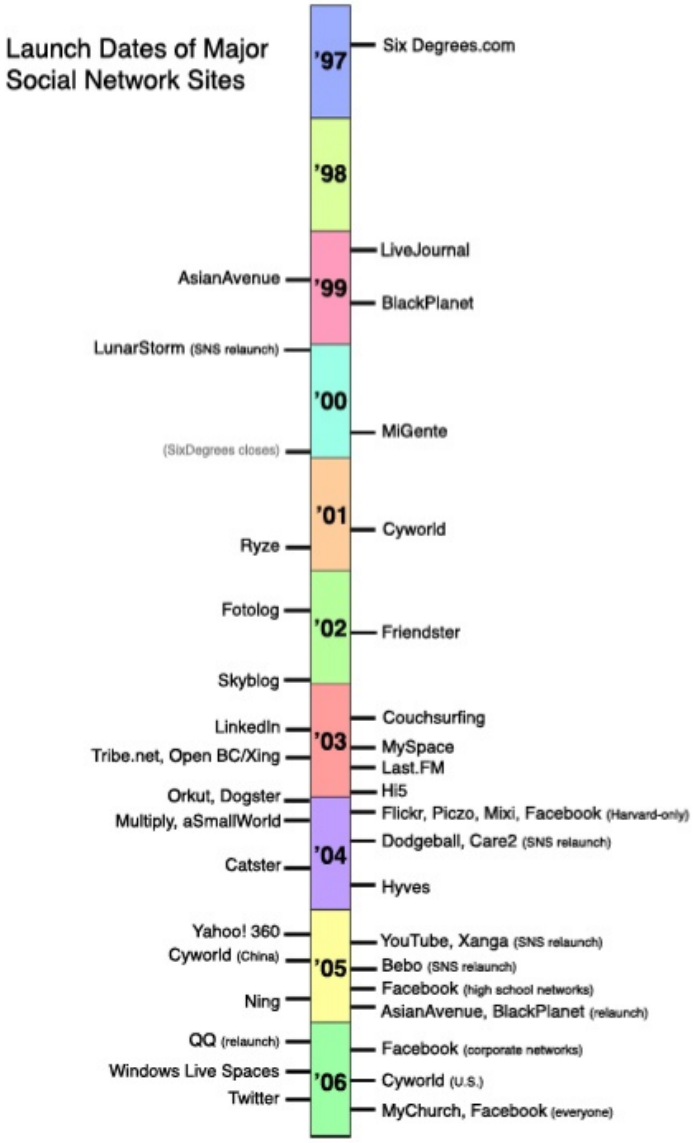
Οι Boyd & Ellison [5] δίνουν έναν ορισμό για τα online κοινωνικά δίκτυα.
Τα online κοινωνικά δίκτυα είναι διαδικτυακές (web-based) υπηρεσίες που επιτρέπουν στα άτομα:

1) να δημιουργήσουν ένα δημόσιο ή ημι-δημόσιο προφίλ μέσα σε ένα οριοθετημένο σύστημα
2) να επικοινωνήσουν με μία λίστα από άλλους χρήστες με τους οποίους μοιράζονται μία μορφή σύνδεσης και
3) να βλέπουν και να διαμοιράζονται τη δική τους λίστα των συνδέσεων με αυτές που δημιουργήθηκαν από άλλους μέσα στο σύστημα.

Με τον όρο σύνδεση εννοούμε μία οποιαδήποτε σχέση μεταξύ δύο ή περισσοτέρων ατόμων (δρόντων) που έχουν κάποιο κοινό σημείο αναφοράς. Η καινοτομία των κοινωνικών δικτύων είναι ότι επιτρέπει τη δημιουργία γνωριμιών με σκοπό τη μεταβίβαση πληροφοριών, ή γενικά ανταλλαγή πόρων. Κάθε είδος τέτοιας ανταλλαγής θεωρείται σχέση, σύνδεση, ή αλλιώς όπως συχνά αναφέρεται, δεσμός (tie) στο δίκτυο και είναι ο τρόπος αλληλεπίδρασης κάθε χρήστη με τους υπολοίπους, δηλαδή τις επαφές του. Οι παράμετροι που ορίζουν το είδος της επικοινωνίας αυτής εξαρτώνται από το κοινωνικό δίκτυο, τόσο τη δομή του όσο και τις λειτουργίες του.

Εικόνα 1: Η αναπαράσταση του κοινωνικού δικτύου

Τα κοινωνικά δίκτυα σχηματικά, για την καλύτερη κατανόησή τους, μπορούν να αναπαρασταθούν ως γράφοι (graphs) όπου το σύνολο των κόμβων τους (ή κορυφές, nodes) είναι οι χρήστες του κοινωνικού δικτύου και οι ακμές τους (edges) οι σχέσεις που τους



συνδέουν. Όπως είναι φυσικό, όσο πιο μεγάλο είναι το κοινωνικό δίκτυο τόσο και πιο περίπλοκος θα είναι και ο γράφος. Οι κόμβοι, που αναπαριστούν τους χρήστες, συνδέονται με άλλους που αποτελούν τις επαφές του και οι σχέσεις που τους συνδέουν καθώς και οι αλληλεπιδράσεις που μπορούν να έχουν, όπως είπαμε, ορίζονται κάθε φορά διαφορετικά ανάλογα με την ιστοσελίδα, ή αλλιώς τον Ιστόχωρο, κοινωνικής δικτύωσης (social networking).

Η αναπαράσταση αυτή είναι πολύ σημαντική για την ανάλυση των κοινωνικών δικτύων (Social Network Analysis), δηλαδή την ανάλυση των δεδομένων εκείνων που μπορεί κανείς να αντλήσει από τα κοινωνικά δίκτυα. Στην ανάλυση αυτή το διαφορετικό, σε σχέση με τις πιο παραδοσιακές μελέτες στον τομέα των κοινωνικών επιστημών, είναι ότι δίνει βάση στις σχέσεις που αναπτύσουν οι χρήστες μέσα στο δίκτυο παρά στις ιδιότητες τους σαν άτομα-χρήστες. Τα χαρακτηριστικά δηλαδή των ατόμων είναι λιγότερης σημασίας σε σχέση με τους δεσμούς που έχουν αναπτύξει μεταξύ τους μέσα στο δίκτυο [6]. Παρακάτω, θα δούμε

μερικά σύγχρονα εργαλεία που χρησιμοποιούνται στην ανάλυση των κοινωνικών δικτύων και στηρίζονται στη θεωρία των γράφων (graph theory).

1.2.2 Οι Ιστόχωροι Κοινωνικής Δικτύωσης

Η ιστοσελίδα Classmates.com [7] θεωρείται από τις πρώτες ιστοσελίδες που επέτρεψε στους χρήστες να συνδεθούν με άλλους χρήστες. Άρχισε μόλις το 1995 ως μία ιστοσελίδα που στόχο είχε να συνδέσει τους σπουδαστές με τους παλιούς συμφοιτητές τους και αυτή τη στιγμή έχει 57 εκατομμύρια εγγραμμένους χρήστες. Εντούτοις, το Classmates.com δεν επέτρεψε στους χρήστες να δημιουργήσουν συνδέσεις γενικά με άλλους χρήστες μόνο μέσω κοινού σχολείου ή κολλεγίου στο οποίο είχαν φοιτήσει.

Το 1997 ξεκίνησε το πρώτο ουσιαστικά online κοινωνικό δίκτυο, το SixDegrees.com [8] καθώς έδωσε τη δυνατότητα στους χρήστες του να συνδέονται απευθείας με άλλους και μπορεί να περιγραφεί πλήρως από τον ορισμό των Boyd & Ellison που δώσαμε παραπάνω. Οι χρήστες μπορούσαν να δημιουργήσουν το δικό τους προφίλ, να έχουν έναν κατάλογο με τους φίλους τους και να μπορούν να έρθουν σε επαφή μαζί τους. Το κοινωνικό δίκτυο όμως έκλεισε μόλις τρία χρόνια αργότερα. Ο λόγος για αυτό ήταν ότι πολλοί άνθρωποι που χρησιμοποιούσαν το Διαδίκτυο εκείνη την περίοδο δεν είχαν διαμορφώσει πολλά κοινωνικά δίκτυα ως εκ τούτου υπήρξε μικρό περιθώριο ελιγμών [9].

Εικόνα 2: Σχηματική απεικόνιση της ανάπτυξης των ιστοσελίδων κοινωνικής δικτύωσης. Πηγή: (http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html)

Καθώς όμως όλο και περισσότεροι άνθρωποι είχαν πρόσβαση στο διαδίκτυο φυσικό επακόλουθο είναι τα κοινωνικά δίκτυα να γίνονται όλο και πιο δημοφιλή. Από τις αρχές του 21ου αιώνα, άρχισαν να δημιουργούνται διάφορα κοινωνικά δίκτυα που ενθάρρυναν τους χρήστες να δημιουργήσουν ομάδες και να αλληπιδράσουν μέσω αυτών, με πιο χαρακτηριστικό και ιδιαίτερα πετυχημένο το Friendster [10]. Το Friendster ήταν ο πρωτοπόρος των σελίδων γνωριμιών. Παρόμοια site που έκαναν την εμφάνισή τους την ίδια χρονική περίοδο είναι η κορεατική ιστοσελίδα Cyworld [11], το Ryze [12] και το LinkedIn [13]. Πιο συγκεκριμένα το Ryze ιδρύθηκε προκειμένου οι χρήστες του να ενδυναμώσουν τα επιχειρησιακά τους δίκτυα. Ο ιδρυτής του Ryze αναφέρει ότι εισήγαγε αρχικά την ιστοσελίδα στους φίλους του, μέλη της κοινότητας επιχειρήσεων και τεχνολογίας του Σαν Φρανσίσκο, συμπεριλαμβανομένων των επιχειρηματιών και των επενδυτών πίσω από πολλά μελλοντικά online κοινωνικά δίκτυα (A. Scott, June 14, 2007) [5]. Από το 2011 μέχρι και σήμερα, το Friendster έχει αλλάξει ταυτότητα και πλέον λειτουργεί ως ιστοσελίδα κοινωνικής διασκέδασης εστιάζοντας κυρίως σε εφαρμογές παιχνιδιών.

Το 2003 δημιουργείται το MySpace [14] που αντίθετα με τα άλλα έδινε στους χρήστες την ευχέρεια να μπορούν να αλλάξουν την εμφάνιση της προσωπικής σελίδας του προφίλ τους όπως εκείνοι επιθυμούσαν, μέχρι και να μπορέσουν να ενσωματώσουν σε αυτή widgets, όπως κάποιο κινούμενο κείμενο. Πολύ γρήγορα έγινε απίστευτα δημοφιλές, με το 2009 να φτάνει να έχει 125 εκατομμύρια χρήστες με αποτέλεσμα να γίνει το μεγαλύτερο online κοινωνικό δίκτυο.

Με την άνοδο στη δημοτικότητα των online κοινωνικών δικτύων, πολλοί άλλοι τύποι ιστοσελίδων άρχισαν να περιλαμβάνουν τα χαρακτηριστικά γνωρίσματα της κοινωνικής δικτύωσης. Τα παραδείγματα περιλαμβάνουν τις ιστοσελίδες διαμοίρασης πολυμέσων (multimedia content sharing sites) (όπως Flickr [15], YouTube [16], και Zoomr [17]), τις ιστοσελίδες blogging (όπως LiveJournal [18] και BlogSpot [19]), τις επαγγελματικές ιστοσελίδες κοινωνικής δικτύωσης (όπως LinkedIn και Ryze), και τις ιστοσελίδες ειδήσεων (Digg [20], Reddit [21], και del.icio.us [22]). Όλα αυτά τα sites έχουν διαφορετικούς στόχους, αλλά υιοθετούν την κοινή στρατηγική που υπάρχει για να βελτιωθούν.

Το LinkedIn σήμερα είναι ίσως το πιο εστιασμένο για επιχειρηματικά ενδιαφέροντα κοινωνικό δίκτυο καθώς δίνει τη δυνατότητα στους χρήστες να δημιουργήσουν ένα δίκτυο από υπάρχουσες και καινούργιες επαγγελματικές επαφές (συνδέσεις-connections). Ένας επαγγελματίας είναι σε θέση να έχει ένα πολύ μεγάλο πεδίο επαγγελματικών επαφών, ενώ παράλληλα ένας οποιοσδήποτε χρήστης μπορεί να αναζητήσει δουλειά και επαγγελματικές ευκαιρίες από χρήστες-εργοδότες που έχουν αναρτήσει ανάλογες περιγραφές θέσεων εργασίας. Ακόμα πολλοί χρήστες το χρησιμοποιούν για να ανταλλάξουν απόψεις, ιδέες και πληροφορίες σχετικά με αντικείμενα του επαγγέλματός τους.

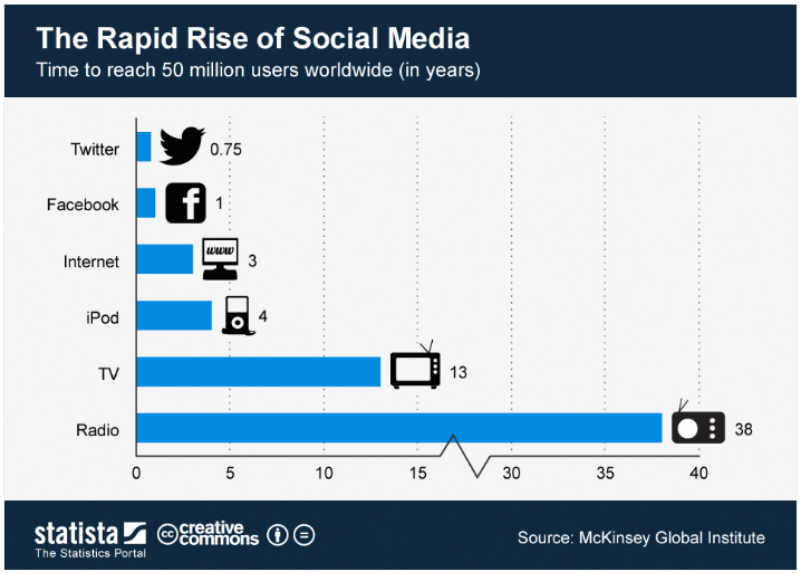
Το Facebook [23] άρχισε στις αρχές του 2004 από τον Mark Zuckerberg, όντας τότε δευτεροετής φοιτητής του Πανεπιστημίου του Harvard. Αρχικά, μπορούσαν να γίνουν μέλη μόνο οι φοιτητές του Πανεπιστημίου προκειμένου να γνωριστούν μεταξύ τους. Αρχίζοντας από τον Σεπτέμβριο του 2005, το Facebook επεκτάθηκε και για μαθητές σχολείων και το 2006 για όλους άνω των 13 ετών. Οι χρήστες μπορούν να προσκαλέσουν και να προσθέσουν άλλους χρήστες, να τους στείλουν μηνύματα, να ενημερώσουν το προσωπικό τους προφίλ και να γνωστοποιήσουν στους άλλους τις δραστηριότητες τους, θα δούμε αναλυτικά και παρακάτω την λειτουργία του. Το 2012 το Facebook ανακοίνωσε ότι έχει πάνω από ένα δισεκατομμύριο ενεργούς χρήστες κάτι που σίγουρα το κατατάσσει στα δημοφιλέστερα κοινωνικά δίκτυα παγκοσμίως.

Το Twitter [24] δημιουργήθηκε το Μάρτιο του 2006 από τον Jack Dorsey και η πλήρης μορφή του παρουσιάστηκε τον Ιούλιο του ίδιου έτους. Απέκτησε γρήγορα παγκόσμια δημοτικότητα καθώς είναι μία πλατφόρμα που σε πραγματικό χρόνο οι χρήστες ενημερώνουν την κατάστασή τους και την κοινοποιούν στους υπόλοιους χρήστες. Η ενημέρωση αυτή ονομάζεται “tweet”. Η δημοτικότητά του προκύπτει από τα στατιστικά στοιχεία, όπου μπορεί να δει κανείς ότι τον Ιούνιο του 2010 αποστέλλονται περίπου 65 εκατομμύρια tweets την ημέρα, που ισοδυναμεί με περίπου 750 tweets κάθε δευτερόλεπτο,

σύμφωνα με το Twitter.com (Ιούλιος 2010). Αναφορικά να πούμε επίσης ότι το 2011 είχε περίπου 500 εκατομμύρια χρήστες, που δημιουργούσαν πάνω από 200 εκατομμύρια tweets κάθε δευτερόλεπτο και άρα πάνω από 1,6 δισ. tweets την ημέρα.

Το 2011 η Google ανακοίνωσε μία δοκιμαστική πλατφόρμα για υπηρεσίες κοινωνικής δικτύωσης, το Google Plus (Google+) [25]. Σκοπός της Google για αυτή την πλατφόρμα ήταν να ενσωματώσει όλες τις υπηρεσίες της σε μία ενιαία και ταυτόχρονα να προσεγγίσει διαφορετικά τον τρόπο που μέχρι στιγμής δημιουργούνται οι σχέσεις μεταξύ των χρηστών. Η καινοτομία αυτή οφείλεται στο γεγονός ότι κάθε χρήστης μπορεί να έχει "κύκλους" γνωριμιών και να τους διαχειρίζεται ο ίδιος ανάλογα πως θέλει να αλληλεπιδρά με τους χρήστες που έχει οργανώσει σε αυτούς.

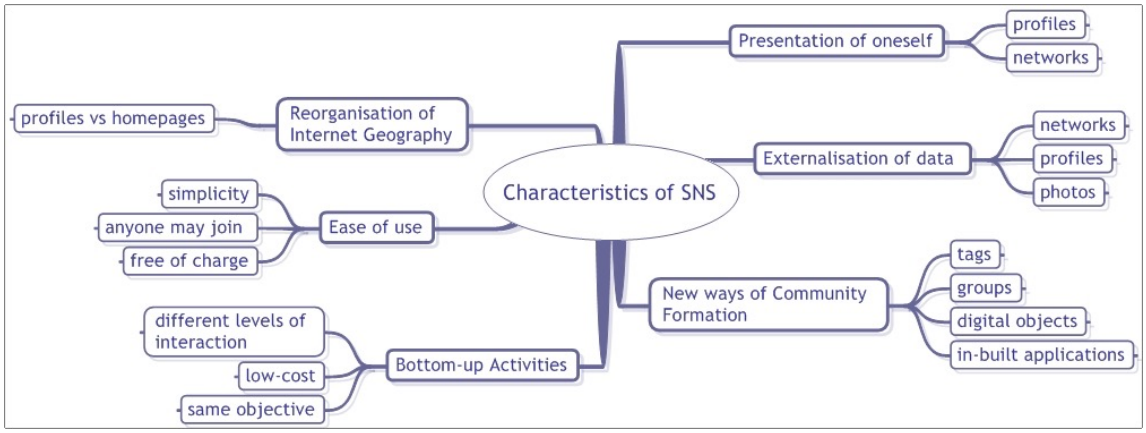
Παράλληλα με αυτά τα sites που αναφέρουμε έχουν αναπτυχθεί κατά καιρούς και άλλα σε διάφορες χώρες του κόσμου και το πλήθος τους αυξάνεται συνεχώς. Εμείς, στο παρόν σύντομο ιστορικό, παρουσιάσαμε αυτά που είχαν, ή έχουν ακόμα, τη μεγαλύτερη απήχηση στο κοινό. Αναφορικά στη χώρα μας, μερικά από τις πιο διαδεδομένα, είναι το Facebook, το Twitter και το Youtube. Όπως φαίνεται, οι ιστόχωροι κοινωνικής δικτύωσης είναι ήδη αναπόσπαστο κομμάτι της καθημερινότητας για τη συντριπτική πλειοψηφία των χρηστών και αναμένεται να ενσωματωθούν ακόμα περισσότερο στη ζωή όλων όσο εξυπηρετούν την ανάγκη αυτή για κοινωνικοποίηση. Αυτό φαίνεται και από το γεγόνος ότι η υιοθέτηση των νέων αυτών τεχνολογιών έγινε πολύ γρήγορα. Είναι γεγονός ότι ενώ για να συνδεθούν στο Διαδίκτυο 50 εκατομμύρια συνδρομητές πήρε 3 χρόνια, για να φτάσει να έχει το Facebook τον ίδιο αριθμό σε χρήστες χρειάστηκε μόλις ένα χρόνο και το Twitter ακόμα λιγότερο.



Εικόνα 3: Γράφημα που δείχνει το χρόνο που χρειάστηκαν κάποιες τεχνολογίες και υπηρεσίες για να φτάσουν τα 50 εκατομμύρια χρήστες σε όλο τον κόσμο. (Πηγή: http://www.statista.com)

1.2.3 Χαρακτηριστικά γνωρίσματα των κοινωνικών δικτύων

Τα χαρακτηριστικά εκείνα που κάνουν μία σελίδα κοινωνικής δικτύωσης διαφορετική από έναν άλλο ιστόχωρο είναι, σύμφωνα με την R. Cachia [26], τα εξής:



Εικόνα 4: Τα χαρακτηριστικά των κοινωνικών δικτύων

Η παρουσίαση του εαυτού μας:

Η προϋπόθεση εγγραφής (sign up) σε μία ιστοσελίδα κοινωνικής δικτύωσης είναι η δημιουργία ενός προφίλ (profile), μίας δηλαδή προσωπικής σελίδας (personal homepage) που δημιουργεί ο χρήστης και στην οποία έχει την δυνατότητα μέσω κειμένου, φωτογραφιών, βίντεο και άλλων επιλογών να παρουσιάσει και τον εαυτό του. Επιπλέον, μπορεί να οργανώνει τις κοινωνικές τους επαφές και να επιτρέπει ή όχι στα άλλα μέλη να βλέπουν το προφίλ του.

Η εξωτερίκευση της πληροφορίας:

Οι περισσότερες ιστοσελίδες κοινωνικής δικτύωσης επιτρέπουν στα μέλη τους να δουν τα δίκτυα των επαφών τους. Με την εξωτερίκευση, ή αλλιώς δημοσίευση των σελίδων, είναι ενδεχομένως μία από τις πρώτες φορές που οι online χρήστες είναι σε θέση να δουν τα online κοινωνικά δίκτυά τους και να τα μοιραστούν με τους φίλους τους και το ευρύ κοινό. Μερικές ιστοσελίδες υποστηρίζουν επίσης εφαρμογές που επιτρέπουν στους χρήστες για περιγράψουν τη σχέση μεταξύ τους και μεταξύ των άλλων μελών.

Νέοι τρόποι δημιουργίας κοινότητας:

Αν και οι έννοιες των εικονικών κοινοτήτων (virtual communities) έχουν υπάρξει από την αρχή των online εφαρμογών στον διαδίκτυο, οι ιστοσελίδες κοινωνικής δικτύωσης παρέχουν στους χρήστες νέους τρόπους να συνδεθούν μεταξύ τους. Οι χρήστες αυτών των ιστοσελίδων μπορούν να επιλέξουν να επικοινωνήσουν μέσω των διαφόρων ψηφιακών εργαλείων, όπως τα tags και διάφορες εφαρμογές ειδικά σχεδιασμένες για τις συγκεκριμένες ιστοσελίδες.

Bottom-up activities:

Οι ιστοσελίδες κοινωνικής δικτύωσης παρέχουν τις ιδανικές πλατφόρμες μέσω των οποίων οι χρήστες με παρόμοιες αξίες και ενδιαφέροντα μπορούν να ενωθούν και να συνεργαστούν αποτελεσματικά και χωρίς κόστος. Για παράδειγμα, οι γιατροί μπορούν να μοιραστούν πληροφορίες για σπάνιες ιατρικές περιπτώσεις, σε sites όπως το Within 3 [27], ή ακόμη οι ενεργοί ακτιβιστές μπορούν να οργανώσουν μία διαμαρτυρία μέσα από κοινωνικά δίκτυα όπως το Care2 [28].

Ευκολία χειρισμού:

Μία σημαντική ιδιότητα της δημοτικότητας των SNs είναι η απλότητά τους. Ο καθένας μόνο με τις βασικές δεξιότητες χρήσης του διαδικτύου μπορεί να δημιουργήσει και να διαχειριστεί το δικό του προφίλ σε μία ιστοσελίδα κοινωνικής δικτύωσης. Επιπλέον, τα κοινωνικά δίκτυα είναι δωρεάν και ανοικτά για τον καθένα. Τα περισσότερα από αυτά απαιτούν μία εγγραφή, ενώ άλλα περιορίζουν την ιδιότητα μέλους μέσω μίας πρόσκλησης από ήδη υπάρχοντες χρήστες, όπως το πολύ καινούργιο κοινωνικό δίκτυο με την ονομασία Ello [29].

Ο γεωγραφικός επαναπροσδιορισμός του διαδικτύου:

Οι ιστοσελίδες κοινωνικής δικτύωσης υποστηρίζουν ένα νέο σημείο εισόδου στον προσωπικό κόσμο των ανθρώπων. Μέχρι τώρα οι άνθρωποι μιλούσαν στο διαδίκτυο μεταφορικά για περιοχές όπως πόλεις, διευθύνσεις, προσωπικές ιστοσελίδες. Τώρα οι ιστοσελίδες κοινωνικής δικτύωσης έχουν μετατρέψει αυτές τις μεταφορές σε πιο προσωπικό επίπεδο μιλώντας για προφίλ, blogs, εικόνες, κ.α.

Λαμβάνοντας υπόψη αυτά τα χαρακτηριστικά, παρατηρούμε σημαντικές αλλαγές στον τρόπο με τον οποίο οι χρήστες χρησιμοποιούν το διαδίκτυο και στο πως χειρίζονται τις κοινωνικές τους επαφές σύμφωνα με διαφορετικά κοινωνικά περιβάλλοντα. Ειδικότερα, τα κοινωνικά δίκτυα φαίνεται να επηρρεάζουν και διαμορφώνουν τον τρόπο με τον οποίο επικοινωνούμε και τον τρόπο με τον οποίο διαχειριζόμαστε τις κοινωνικές μας επαφές.

1.2.4 Δημοφιλή Κοινωνικά Δίκτυα

Τα μεγάλα κοινωνικά δίκτυα είναι συνήθως διαθέσιμα σε πολλές γλώσσες και επιτρέπουν στους χρήστες να συνδεθούν με φίλους τους ή άτομα ανεξάρτητα από γεωγραφικά, πολιτικά ή οικονομικά σύνορα. Από το 2012, περισσότεροι από 1,4 δισ. χρήστες του Διαδικτύου έχουν πρόσβαση στα κοινωνικά δίκτυα και αυτά τα στοιχεία εξακολουθούν να αυξάνονται λόγω της προσβασης σε αυτά μέσω ειδικών εκδόσεων για κινητές συσκευές τηλεφώνου.

Ωστόσο είναι σημαντικό να δούμε τα χαρακτηριστικά κάθε κοινωνικού δικτύου καθώς κάθε κοινωνικό δίκτυο έχει διαφορετικό τρόπο αλληλεπίδρασης μεταξύ των χρηστών του, αφού προσφέρει διαφορετικές δυνατότητες. Ακόμα, άλλα κοινωνικά δίκτυα επικεντρώνονται στις σχέσεις που δημιουργούν μεταξύ τους οι χρήστες ενώ άλλα έχουν σκοπό να προβάλλουν το περιεχόμενο που δημιουργείται από τους χρήστες τους. Παρακάτω γίνεται μια συνοπτική αναφορά στα πιο δημοφιλή κοινωνικά δίκτυα του Παγκόσμιου Ιστού.

Facebook

10 χρόνια μετά το έτος ίδρυσής του, το Facebook [23] είναι σήμερα ο μεγαλύτερος ιστόχωρος κοινωνικής δικτύωσης τόσο σε παγκόσμια εμβέλεια όσο και βάση των ενεργών χρηστών του. Εδώ, ενεργοί χρήστες ορίζονται οι χρήστες εκείνοι που έχουν συνδεθεί στο Facebook μέσα στο διάστημα των τελευταίων 30 ημερών. Με βάση τη στατιστική έρευνα της statista.com, φαίνεται ότι το τρίτο τρίμηνο του 2014, το Facebook είχε 1,35 δισ. μηνιαία ενεργούς χρήστες.

Στο Facebook κάθε χρήστης θα πρέπει να εγγραφεί προτού χρησιμοποιήσει το κοινωνικό δίκτυο και ύστερα είναι ελεύθερος να δημιουργήσει μία προσωπική σελίδα-προφίλ (profile page), προκειμένου να επικοινωνεί με άλλους χρήστες, τους οποίους μπορεί να τους προσθέσει ως φίλους. Το προφίλ ενός χρήστη αποτελεί τη σελίδα προσωπικής του έκφρασης στην οποία παρουσιάζει αρχικά κάποιες βασικές πληροφορίες όπως όνομα, επώνυμο, ηλικία, πόλη διαμονής, σπουδές, πολιτικές πεποιθήσεις κτλ. Εκτός από τις βασικές αυτές πληροφορίες, οι οποίες θεωρούνται χαρακτηριστικά σταθερά χωρίς να αλλάζουν ή να επηρεάζονται από τους άλλους χρήστες του δικτύου, υπάρχουν και οι προσωπικές πληροφορίες για την καθημερινότητα ή τα ενδιαφέροντα του οι οποίες αναρτώνται στις ενημερώσεις κατάστασης (status updates) και περιγράφουν τι κάνει ή πώς αισθάνεται τη συγκεκριμένη στιγμή ο χρήστης όχι απαραίτητα μόνο μέσα από κάποιο κείμενο αλλά και μέσα από φωτογραφίες, βίντεο ή άλλους συνδέσμους. Οι φίλοι του τότε είναι σε θέση να δούν και να σχολιάσουν, αν επιθυμούν, αυτές τις καταστάσεις μέσα από μια κεντρική σελίδα όπου εμφανίζονται όλα τα δεδομένα της δραστηριότητας των χρηστών που είναι συνδεδεμένοι με τον χρήστη. Το ποια από τα παραπάνω στοιχεία θα

κοινοποιούνται και σε ποιούς ελέγχονται από το χρήστη μέσω ρυθμίσεων (privacy settings) που μπορεί να κάνει καθώς μπορεί να επιλέγει ο ίδιος το τρόπο που θα τα δημοσιοποιεί.

Μερικές από τις πλέον ενδιαφέρουσες υπηρεσίες που προσφέρει το Facebook είναι ότι κάθε χρήστης μπορεί είτε να γίνει μέλος είτε να δημιουργήσει ομάδες (groups) για κάποιο σκοπό, ή αντίστοιχα να κάνει like σε κάποια σελίδα αρεσκείας (page). Ακόμα υπάρχουν οι μελλοντικές εκδηλώσεις (events) στις οποίες δηλώνει αν θα παρευρεθεί ή τις δημιουργεί για να τις γνωστοποιήσει σε άλλους χρήστες.

Το Facebook είναι επίσης προσβάσιμο και μέσω συσκευής κινητού και ειδικά για το σκοπό αυτό έχει δημοσιεύσει μια σειρά από εφαρμογές με βάση τις λειτουργίες του, όπως είναι το Facebook Messenger για ανταλλαγή προσωπικών μηνυμάτων (instant messages). Με αφορμή την τεράστια δημοτικότητά του, το Facebook έχει υποστεί κριτική και κατηγορείται για θέματα που αφορούν την ασφάλεια των προσωπικών δεδομένων των χρηστών του. Ωστόσο, η συγκεκριμένη ιστοσελίδα παραμένει η πιο διάσημη εφαρμογή κοινωνικής δικτύωσης.

Twitter

Παρόλο που οι δυνατότητες οι οποίες παρέχονται στους χρήστες είναι αρκετά λιγότερες σε σχέση με το Facebook, ένα ακόμα αρκετά πετυχημένο κοινωνικό δίκτυο είναι το Twitter [24]. Ο λόγος αυτής του της επιτυχίας, βασίζεται στο γεγονός ότι εισήγαγε ένα καινοτόμο διαδραστικό τρόπο επικοινωνίας που διεξάγεται με μόλις 140 χαρακτήρες. Αυτή η ιδέα ήταν βασισμένη στη λογική του λεγόμενου micro blogging. Οι χρήστες μπορούν να κοινοποιούν στους υπολοίπους την κατάστασή τους, δηλαδή να κάνουν tweet. Το διαφορετικό που αξίζει να σημειωθεί για το Twitter είναι ότι μία σχέση "φιλίας" ανάμεσα σε δύο χρήστες δεν πρέπει να είναι απαραίτητα αμφίδρομη. Έτσι, παρατηρεί κανείς ότι υπάρχουν δύο κατηγορίες φίλων για κάθε χρήστη, οι followers και οι following, ή αλλιώς αυτοί που ακολουθούν το χρήστη και αυτούς που ο χρήστης ακολουθεί. Οι πρώτοι έχουν επιλέξει το χρήστη ως φίλο, ενώ τους άλλους τους έχει επιλέξει ο χρήστης ως φίλους.

Αξιοσημείωτο είναι επίσης να αναφέρουμε την επισήμανση περιεχομένου (tagging) για την κατηγοριοποίηση του περιεχομένου των πληροφοριών που υπάρχουν στο Διαδίκτυο χρησιμοποιώντας ετικέτες με το σύμβολο της δίεσης (#). Όταν ακριβώς ίδιες ετικέτες χρησιμοποιούνται σε έναν αρκετά μεγάλο αριθμό από tweets τότε δημιουργούνται οι λεγόμενες τάσεις (trends). Ακόμα, όταν ένας χρήστης θέλει να αναφερθεί σε έναν άλλον μέσα από δημόσια tweets τότε η αναφορά αυτή γίνεται μέσω του συμβόλου @. Τέλος, υπάρχει η δυνατότητα και για ανταλλαγή προσωπικών μηνυμάτων μεταξύ των χρηστών του.

LinkedIn

Το LinkedIn [13] είναι το μεγαλύτερο κοινωνικό δίκτυο με χαρακτήρα επαγγελματικό. Λογαριασμό στο LinkedIn μπορεί να έχει τόσο ένα φυσικό πρόσωπο όσο και μία εταιρεία. Έτσι, δημιουργείται ένα δίκτυο από επαγγελματίες και επιχειρήσεις με στόχο κυρίως την ανεύρεση εργασίας ή προσωπικού αντίστοιχα, ανταλλαγή επιστημονικών απόψεων, διασύνδεση με άλλους χρήστες (εργοδότες, συναδέλφους, φίλους κτλ) ακόμα και εύρεση επαγγελματικών ευκαιριών (π.χ μίας πρόσληψης ή προσέλκυση νέων πελατών). Το δίκτυο των "φίλων" εδώ αποτελείται από συνδέσεις ή όπως αλλιώς ονομάζονται connections. Κάθε χρήστης έχει τη δυνατότητα να δημιουργήσει ένα λογαριασμό, και να τον κοινοποιεί στις συνδέσεις του, μέσω του οποίου θα προβάλλει το επαγγελματικό του δημόσιο προφίλ. Σε αυτό υπάρχει μία περίληψη των επαγγαλματιών εμπειριών του χρήστη και τους στόχους του και παρακάτω αναλυτικά, βάση χρονολογικής σειράς, η επαγγελματική του πορεία. Το προφίλ μπορεί να ανανεωθεί ανά πάσα στιγμή και ο χρήστης να προσθέσει κάποια κατάρτιση, πιστοποίηση, βράβευση ή οποιαδήποτε άλλη πληροφορία σημαντική για το πλαίσιο του δικτύου των επαγγελματιών που ανήκει.

Google +

Από τα πιο καινούργια κοινωνικά δίκτυα, αλλά με μεγάλη προσέλκυση χρηστών είναι το Google+ [25]. Το διαφορετικό από όλα τα άλλα κοινωνικά δίκτυα είναι ότι κάθε χρήστης έχει τη δυνατότητα να οργανώσει τις επαφές του σε κύκλους επαφών (circles) και να επιλέξει ο ίδιος σε ποιούς κύκλους θα κοινοποιεί τα δεδομένα του. Το είδος αυτό φιλτραρίσματος είναι σημαντικό γιατί ορίζει ένα τρόπο ιδιωτικότητας των πληροφοριών που δημοσιοποιούνται. Ακόμα ο χρήστης διαλέγει αν θέλει να λαμβάνει ειδοποιήσεις για τις δραστηριότητες που αφορούν τους κύκλους του καθώς και τι δικαιώματα θα ορίζει σε αυτούς. Επιπλέον, τα Hangouts είναι μια από τις πρόσθετες εφαρμογές του Google+ και υποστηρίζει την άμεση συνομιλία (chat) μεταξύ των χρηστών. Άλλες εφαρμογές συμβατές με κινητές συσκευές είναι διάφορα παιχνίδια, η υπηρεσία Instant Upload για διαμοίραση δεδομένων όπως φωτογραφίες, βίντεο κτλ.

1.2.5 Ανάλυση Κοινωνικών Δικτύων

Ιστοσελίδες όπως το Facebook, το Twitter κτλ είναι από τα πιο γνωστά παραδείγματα κοινωνικών δικτύων για το λόγο ότι έχουν προσελκύσει τόσο μεγάλο αριθμό χρηστών. Με την ανάλυση τέτοιων δικτύων μπορούμε να δούμε θέματα δομής και ιδιοτήτων του δικτύου, καθώς και τον τρόπο διάδοσης της πληροφορίας.

Αυτό που μας ενδιαφέρει να μελετήσουμε είναι ο τρόπος που μπορούν να αντληθούν μετρήσιμα δεδομένα (social metrics) από ένα κοινωνικό δίκτυο και στη συνέχεια πως αυτά τα δεδομένα να χρησιμοποιηθούν προκειμένου να βγουν αναλυτικά συμπεράσματα για τη συμπεριφορά των χρηστών μέσα σε αυτά (social analytics). Με άλλα λόγια είναι χρήσιμο να γίνει ένα είδος οργάνωσης των δεδομένων που υπάρχουν στα κοινωνικά δίκτυα μέσα από την ανάπτυξη κατάλληλων εργαλείων.

Προκειμένου να γίνει η μελέτη αυτή (Social Network Analysis) πρώτα πρέπει να γίνει η απεικόνιση κάθε στοιχείου που χρειάζεται επεξεργασία, είτε είναι κάποιος χρήστης είτε μία ομάδα χρηστών, κάποια δεδομένα ή και σχέσεις που έχουν ορισθεί ανάμεσα στους χρήστες και τα δεδομένα. Η πιο γνωστή αναπαράσταση των δικτύων γίνεται με τη χρήση γράφων. Οι κόμβοι απεικονίζουν τους χρήστες ή τις ομάδες χρηστών, ενώ οι ακμές δείχνουν τις σχέσεις ή τις ροές μεταξύ των χρηστών. Στο κοινωνικό δίκτυο όπως το Twitter, όπου οι σχέσεις μεταξύ των χρηστών μπορεί να είναι και μονομερείς, είναι χρήσιμο η αναπαράσταση να γίνεται με κατευθυνόμενους γράφους. Άλλη αναπαράσταση μπορεί να γίνει μέσω πινάκων, αποτελούμενοι από γραμμές και στήλες όσοι οι χρήστες του δικτύου και τα στοιχεία να είναι οι δεσμοί που τους ενώνουν.

Ενδεικτικά, μερικές απλές μετρικές (graph metrics) όταν η αναπαράσταση των κοινωνικών δικτυών γίνεται μέσω γράφων είναι:

α) σε επίπεδο δικτύου

- η συνοχή (connectivity) του, όπου εξετάζεται ο βαθμός (degree) με τον οποίο τα μέλη του συνδέονται μεταξύ τους με συνεκτικούς δεσμούς.

- η πυκνότητα (density), δηλαδή ο αριθμός των δεσμών που υπάρχουν στο δίκτυο προς τον αριθμό που θα ήταν εφικτό να υπάρχουν, δηλαδή το μέγιστο δυνατό.

- το μήκος διαδρομής-μονοπατιού (path) που είναι η απόσταση μεταξύ ενός ζεύγους κόμβων στο δίκτυο, το μέσο μήκος διαδρομών (average path length) είναι ο μέσος όρος όλων των αποστάσεων των πιθανών αυτών ζευγαριών και το συντομότερο μονοπάτι (shortest path) είναι η μικρότερη δυνατή διαδρομή σε σχέση με όλες τις υπάρχουσες για δύο κόμβους.

- η κλίκα (clique) γενικά είναι μια υποομάδα κόμβων του δικτύου που είναι μεταξύ τους ισχυρά συνδεδεμένοι. Οι χρήστες ενός δικτύου μπορούμε να πούμε ότι σχηματίζουν κλίκες όταν συνδέονται σε ομάδες με κοινά ενδιαφέροντα.

β) σε επίπεδο κόμβου

- η κεντρικότητα (centrality), δηλαδή το πλήθος των δεσμών που έχει ο κόμβος, που μπορεί σε ένα κοινωνικό δίκτυο όπως το Facebook να είναι το πλήθος των φίλων του χρήστη. Συνήθως, όσο πιο πολλούς δεσμούς έχει ένας κόμβος τόσο πιο ισχυρή είναι η θέση του στο δίκτυο σε σχέση με τη μετάδοση της πληροφορίας.

- η κεντρικότητα ενδιαμεσότητας (betweenness centrality) δείχνει πώς ο κόμβος συνδέεται με άλλους ως ενδιάμεσος σταθμός σε σχέση με τη ροή των πληροφοριών.

Το πλεονέκτημα που έχει μια αναπαράσταση με τη χρήση γράφων είναι ότι δίνεται έμφαση στις σχέσεις μεταξύ των χρηστών αντί να μελετηθούν χαρακτηριστικά των ίδιων των χρηστών όπως για παράδειγμα το φύλο, η ηλικία, ο τόπος διαμονής. Έτσι, βγαίνουν συμπεράσματα για τις αλληλεπιδράσεις μεταξύ των χρηστών και τις σχέσεις που αναπτύσσουν μέσα από τις τεχνολογίες που τους δίνει το κοινωνικό δίκτυο. Για παράδειγμα μπορεί να βρεθεί η επιρροή που έχει ένα χρήστης και αν η θέση του στο δίκτυο είναι σημαντική σε σχέση με κάποιο παράγοντα. Μετριέται πόσο δημοφιλής είναι ένας χρήστης, ή πόσο κοινωνικός ανάλογα και με το πόσο ενεργός είναι στο κοινωνικό δίκτυο συναρτήσει του χρόνου και την αποδοχή που έχει από τους υπολοίπους.

Εργαλεία που χρησιμοποιούνται για τους παραπάνω σκοπούς υπάρχουν πολλά και δίνουν την ευκαιρία σε οποιονδήποτε επιθυμεί να μελετήσει με αποτελεσματικό και εύκολο τρόπο πως αναπτύσσονται οι σχέσεις μεταξύ των χρηστών ενός κοινωνικού δικτύου. Αναλυτικά όλα τα προγράμματα λογισμικού, είτε για ερευνητικούς, είτε για εμπορικούς σκοπούς, υπάρχουν στο International Network for Social Network Analysis (INSNA) [34].

1.3 Η Εμπιστοσύνη στα Κοινωνικά Δίκτυα

1.3.1 Βασικοί Ορισμοί

Η έννοια της εμπιστοσύνης (trust) είναι ένα έμφυτο και παράλληλα αναπόσπαστο κομμάτι της ανθρώπινης ύπαρξης. Η λήψη κάθε απόφασης, κοινωνικής, επαγγελματικής, οικονομικής καθορίζεται άμεσα ή έμμεσα από την εμπιστοσύνη που υπάρχει στην εκάστοτε σχέση. Συνεπώς, η εμπιστοσύνη διαδραματίζει πολύ σημαντικό ρόλο στην καθημερινή ζωή και αυτό συμβαίνει και στα κοινωνικά δίκτυα εφόσον στις μέρες μας αποτελούν κομμάτι αυτής της καθημερινότητας.

Στην ανθρώπινη κοινωνία, σύμφωνα με την J. Golbeck [30], η εμπιστοσύνη εξαρτάται από ένα πλήθος παραγόντων οι οποίοι δεν μπορούν εύκολα να μοντελοποιηθούν σε ένα υπολογιστικό σύστημα. Μερικοί τέτοιοι παράγοντες στη σχέση εμπιστοσύνης με ένα άτομο είναι η προηγούμενη εμπειρία που έχουμε μαζί του και με τους φίλους του, οι γνώμες που έχουμε λάβει για τις πράξεις του, οι ψυχολογικοί παράγοντες που έχουν επηρρεάσει στο παρελθόν τη ζωή μας και κάποια γεγονότα (τα οποία δεν έχουν καμία άμεση σχέση με το πρόσωπο αυτό που αποφασίζουμε τη δεδομένη στιγμή αν θα το εμπιστευτούμε ή όχι), η φήμη του, δηλαδή η επιρροή μας από τις γνώμες των άλλων, και τέλος τα κίνητρα μας για να κερδίσουμε κάτι σαν αποτέλεσμα της εμπιστοσύνης που θα δείξουμε.

Κάποιος ακριβής ορισμός για την εμπιστοσύνη γενικά δεν είναι εύκολο να δοθεί καθώς ο καθένας έχει την δική του αντίληψη σχετικά με την έννοια και τη σημασία της. Ένα μοντέλο εμπιστοσύνης παρουσιάστηκε το 1994 από τον Marsh [31] το οποίο είναι αρκετά πολύπλοκο καθώς περιέχει παράγοντες τόσο κοινωνικούς όσο και ψυχολογικούς, αλλά καταλήγει στην άποψη ότι η εμπιστοσύνη θεωρείται υποκειμενική για κάθε άτομο.

Ένας αρκετά δημοφιλής ορισμός είναι αυτός που επινοήθηκε από τον Deutch [32]:

΄Ενα άτομο

(α) αντιμετωπίζει ένα διφορούμενο μονοπάτι, ένα μονοπάτι που μπορεί να το οδηγήσει σε ένα γεγονός το οποίο το αντιλαμβάνεται ως ωφέλιμο ή ως επιζήμιο

(β) θεωρεί ότι η πραγματοποίηση αυτών των γεγονότων συνδέεται άμεσα με τη δράση ενός άλλου ατόμου, και

(γ) θεωρεί ότι οι αρνητικές επιπτώσεις του επιζήμιου γεγονότος είναι μεγαλύτερες από τις θετικές του ωφέλιμου. Εάν διαλέξει να ακολουθήσει το διφορούμενο μονοπάτι με αυτές τις ιδιότητες, λαμβάνει μια απόφαση εμπιστοσύνης, αλλιώς λαμβάνει μια απόφαση μη εμπιστοσύνης (δυσπιστίας).

Βλέπουμε πως ο Deutsch ορίζει την εμπιστοσύνη σαν μία προσδοκία γεγονότων, δηλαδή ότι υπάρχει εμπιστοσύνη αν οι προσδοκίες του το οδηγούν σε συμπεριφορά την οποία αντιλαμβάνεται να έχει μεγαλύτερες αρνητικές επιπτώσεις εάν η προσδοκία του δεν επιβεβαιωθεί από ότι θετικές επιπτώσεις εάν η προσδοκία του επιβεβαιωθεί.

Η μαθηματική εκδοχή του παραπάνω ορισμού του Deutsch έρχεται το 2002 από τον C. Dellarocas [33] που παρουσιάζει ιδιαίτερο ενδιαφέρον προκειμένου να δει κανείς ένα τρόπο μοντελοποίησης τέτοιων εννοιών στο διαδικτυακό κόσμο. Το ενδιαφέρον στην εκδοχή αυτή ήταν ότι μοντελοποιώντας μία σχέση όπως η εμπιστοσύνη με μαθηματικά μπορεί κανείς να μελετήσει ιδιότητές της όπως συμμετρία, μεταβατικότητα, αυτοπάθεια κτλ.

1.3.2 Ιδιότητες Εμπιστοσύνης

Τα ιδιαίτερα χαρακτηριστικά της εμπιστοσύνης διαφέρουν από περίπτωση σε περίπτωση και γιαυτό κάθε σχέση εμπιστοσύνης είναι μοναδική. Παρόλα αυτά, δημιουργείται και εξελίσσεται εντός ενός συγκεκριμένου πλαισίου σε σχέση με την ικανότητα του ατόμου που εμπιστευόμαστε να προσφέρει κάποιες υπηρεσίες. Για παράδειγμα, μπορεί να εμπιστευόμαστε κάποιον να διεξάγει οικονομικές συναλλαγές μόνον όταν η αξία τους δεν ξεπερνά ένα συγκεκριμένο χρηματικό ποσό. Ακόμα, το ότι εμπιστευόμαστε κάποιον να επιτελέσει μια συγκεκριμένη εργασία δεν σημαίνει απαραίτητα ότι τον εμπιστευόμαστε να επιτελέσει οποιαδήποτε άλλη εργασία ή ακόμα και την ίδια εργασία εάν διαφοροποιηθούν ορισμένα χαρακτηριστικά της. Χρονικά, η εμπιστοσύνη δεν είναι κάτι που συμβαίνει στιγμιαία ούτε κάτι που διαρκεί για πάντα. Αντίθετα εξελίσσεται μέσα στο χρόνο και είναι πιθανόν να μεταβάλλεται με την πάροδό του.

Προκειμένου να μετρήσουμε την εμπιστοσύνη σε διάφορους κλάδους όπως το μάρκετινγκ, χρειάστηκε να ορισθούν κάποιοι δείκτες οι οποίοι να χρησιμοποιούν μαθηματικές πράξεις ώστε να αντιστοιχούν ποιοτικές τιμές σε αριθμητικά αποτελέσματα. Οι αριθμητικές αυτές τιμές μπορεί να συμβολίζουν μέσω μίας κλίμακας το πόσο εμπιστεύεται ένας χρήστης έναν άλλον για μία ηλεκτρονική συναλλαγή. Πληροφορίες, όπως η προηγούμενη εμπειρία ενός χρήστη για μία τέτοια συναλλαγή, μπορούν να ληφθούν ως δείγμα πρόβλεψης μελλοντικής συμπεριφοράς βασισμένη σε στατιστικές μεθόδους.

Για το λόγο αυτό, η παροχή τέτοιων αποτελεσμάτων έγινε ιδιαίτερα γνωστή και η εγκυρότητα τους αναπτύχθηκε μέσω των λεγόμενων συστημάτων διαχείρισης εμπιστοσύνης. Κάθε τέτοιο σύστημα κατηγοριοποιεί τις μεθόδους, ή αλλιώς μετρικές, που χρησιμοποιεί για την έκφραση μίας σχέσης εμπιστοσύνης και στο τέλος γίνεται μια αξιολόγηση τους, βάση παραγόντων όπως καταλληλότητα, πολυπλοκότητα, πληρότητα κτλ.

Πολλές εφαρμογές υπάρχουν σε P2P δίκτυα, συστήματα υπολογιστών γενικά, καθώς ακόμα και σε κοινωνικά δίκτυα και η αναπαράστασή τους έγινε μέσω γράφων με βάρη στις ακμές τους.

1.3.3 Κοινωνικά Δίκτυα - Δίκτυα Εμπιστοσύνης

Με το πέρασμα του χρόνου ο άνθρωπος δημιούργησε τις κοινωνίες προκειμένου να διευκολύνει τη ζωή του και με βάση τη συνεργασία να μπορέσει να βρει τις τεχνικές εκείνες που θα κάνουν την αλληλεπίδραση με τους συνανθρώπους τους όσο πιο αποτελεσματική. Όπως είναι φυσικό, για να πετύχει αυτή η επιθυμητή συνεργασία έμαθε να εμπιστεύεται τους ανθρώπους. Όταν έπρεπε να πάρει μια απόφαση λάμβανε υπόψη του τη γνώμη εκείνων που εμπιστευόταν επειδή τους θεωρούσε αξιόπιστους είτε λόγω κάποιου χαρακτηριστικού, π.χ του κύρους τους, είτε εφόσον στο παρελθόν ανάλογη εμπειρία είχε θετικά αποτελέσματα.

Με παρόμοιο τρόπο λειτουργεί και ένας χρήστης σε ένα κοινωνικό δίκτυο, ο οποίος σε ένα συνεχώς μεταβαλλόμενο περιβάλλον δημιουργεί σχέσεις, ανταλλάσει πληροφορίες, αξιολογεί υπηρεσίες και ενσωματώνει με φυσικό τρόπο τις καινούργιες τεχνολογίες του ηλεκτρονικού κόσμου στη ζωή του. Η εμπιστοσύνη δηλαδή είναι μία έννοια γνωστή στη διαμόρφωση των διαπροσωπικών σχέσεων παρόλα αυτά σε ένα διαδικτυακό κόσμο είναι απαραίτητη για κάθε ηλεκτρονική συναλλαγή και με την έννοια αυτή εννοούμε και την πρόσβαση στην πληροφορία.

Πολύ σημαντικό είναι ότι ο ίδιος ο χρήστης με την εγγραφή του σε ένα κοινωνικό δίκτυο βρίσκεται εκτεθειμένος στους υπολοίπους χρήστες, με την έννοια ότι τους δημοσιοποιεί τα προσωπικά του δεδομένα. Οι πληροφορίες αυτές είτε είναι προσωπικά στοιχεία όπως όνομα, διεύθυνση κτλ είτε λεπτομέρειες για τον τρόπο ζωής του, τις προτιμήσεις του, φωτογραφίες, η δραστηριότητα στο κοινωνικό δίκτυο γενικά, παύουν να είναι ιδιωτικά δεδομένα. Έτσι, ειδικά τα τελευταία χρόνια έχουν παρατηρηθεί από ειδικούς θέματα στην ασφάλεια τέτοιων πληροφοριών καθώς υπάρχουν περιστατικά "κλοπής" και εκμετάλλευσής τους με τρόπο μη επιθυμητό. Χρήστες που συμμετέχουν σε τέτοιες διαδικασίες ορίζονται ως κακόβουλοι (malicious users) για το κοινωνικό δίκτυο καθώς εμποδίζουν την ομαλή λειτουργία του.

Σκοπός της διπλωματικής αυτής εργασίας είναι να φτιάξουμε ένα μοντέλο εμπιστοσύνης μεταξύ των χρηστών ενός κοινωνικού δικτύου το οποίο θα μπορεί να εντοπίζει και να απομονώνει τους χρήστες εκείνους που θεωρούνται κακόβουλοι. Τελικά, το κοινωνικό δίκτυο που θα δημιουργείται είναι ένα δίκτυο εμπιστοσύνης (trust network) στο οποίο οι χρήστες θα εμπιστεύονται αυτούς με τους οποίους συνδέονται.

Κεφάλαιο 2

2.1 Τα Ασύρματα ad hoc δίκτυα

2.1.1 Τι είναι ένα ασύρματο ad hoc δίκτυο: Δομή και λειτουργίες

Γενικά, ένα δίκτυο υπολογιστών (computer network) [35] είναι ένα τηλεπικοινωνιακό σύστημα το οποίο αποτελείται από δύο ή περισσότερους υπολογιστές. Σκοπός της δημιουργίας του είναι η επικοινωνία μεταξύ των υπολογιστών προκειμένου να υπάρξει ανταλλαγή δεδομένων. Το πιο γνωστό και σίγουρα και το μεγαλύτερο τέτοιου είδους δίκτυο είναι το Διαδίκτυο (Internet).

Όταν ένα τηλεπικοινωνιακό δίκτυο υπολογιστών χρησιμοποιεί ασύρματους συνδέσμους για τη διασύνδεση τους, συνήθως ραδιοσυνδέσμους, τότε το ονομάζουμε ασύρματο δίκτυο (wireless network). Με βάση τη δομή υπάρχουν δύο τύποι ασύρματων δικτυών. Με προϋπάρχουσα υποδομή καθώς κάποιοι υπολογιστές, οι ονομαζόμενοι servers, έχουν συγκεκριμένη θέση, είναι τα δίκτυα κινητής τηλεφωνίας και τα τοπικά δίκτυα (WLANs). Η άλλη κατηγορία, τα ad hoc δίκτυα, wireless ad hoc networks (WANETs) δεν απαιτούν καμία προυπάρχουσα δομή καθώς, όπως θα εξηγηθεί και παρακάτω, αναπτύσσονται με δυναμικό τρόπο.

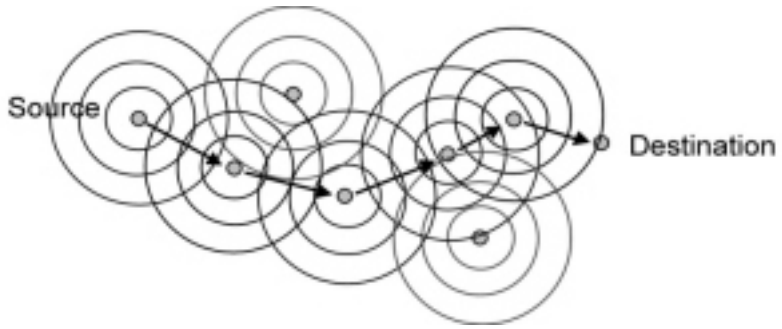
Κατά τη διάρκεια της δεκαετίας του ’90, παρατηρείται ραγδαία εξάπλωση των ασύρματων τηλεπικοινωνιακών συστημάτων συνεπώς και των αντίστοιχων τεχνολογιών που χρησιμοποιούν. Η μεγάλη ανάπτυξη που γνωρίζουν οι ασύρματες τεχνολογίες οφείλεται στο γεγονός ότι δίνουν στους χρήστες τη δυνατότητα να έχουν εύκολη πρόσβαση σε κάθε είδους πληροφορία ανεξάρτητα από την θέση τους. Η επιπλέον δυνατότητα που δίνει ένα ad hoc δίκτυο είναι η ασύρματη μεταφορά δεδομένων ως αποτέλεσμα μίας δυναμικής και αυτόνομης διαδικασίας. Η διαδικασία αυτή δεν είναι απλή αλλά στηρίζεται στην υλοποίηση αλγορίθμων και πρωτόκολλων ειδικά προσαρμοσμένων στα πρότυπα του νέου Παγκόσμιου Ιστού (World Wide Web ή WWW) και τις εφαρμογές που προσφέρει στους χρήστες.

Ένας σύντομος ορισμός για τα ad hoc δίκτυα είναι ο εξής:
Ad hoc δίκτυο είναι ένα αυτορυθμιζόμενο δίκτυο αποτελούμενο από ένα σύνολο κινητών κόμβων που διασυνδέονται με ασύρματες ζεύξεις.

Το πρωτότυπο στη λειτουργία των ad hoc δικτύων είναι ότι οι κόμβοι τους επικοινωνούν χωρίς να απαιτούν συγκεκριμένη προϋπάρχουσα διαδικτυακή υποδομή. Αυτός ο τρόπος ασύρματης επικοινωνίας των κόμβων τους καθιστά αυτόνομους και τελικά σχηματίζουν ένα δίκτυο στο οποίο δεν υπάρχει κεντρική οργάνωση ούτε προκαθορισμένη δομή. Σε αντίθεση με τα ενσύρματα δίκτυα, στα οποία οι δρομολογητές καθορίζουν την διαδικασία της δρομολόγησης, εδώ η διαδικασία προώθησης της πληροφορίας γίνεται δυναμικά με βάση τη συνδεσιμότητα του δικτύου. Πιο συγκεκριμένα, οι κόμβοι θεωρούνται ομότιμοι και

καθένας μπορεί να επικοινωνεί αυτόνομα με οποιοδήποτε άλλο κόμβο μέσω συνδέσεων και πολλαπλών βημάτων αναμετάδοσης χωρίς την μεσολάβηση κάποιου κεντρικού διαχειριστή του δικτύου. Για το λόγο αυτό, τα δύο βασικά χαρακτηριστικά κάθε ασύρματου ad hoc δίκτυου είναι ότι είναι αυτοοργανωμένο και αποκεντρωμένο.

Τα ad hoc δίκτυα είναι γνωστά και ως πολλαπλών αλμάτων ασύρματα δίκτυα (multi-hop wireless networks) για το λόγο ότι υπάρχει συνεκτικότητα, καθώς οποιαδήποτε δύο κόμβοι είναι σε θέση να επικοινωνήσουν μεταξύ τους, ακόμα και στη περίπτωση εκείνη όπου ο ένας δεν βρίσκεται εντός της εμβέλειας του άλλου, χρησιμοποιώντας τους ενδιάμεσους κόμβους ως δρομολογητές. Έτσι μπορούμε να πούμε, ότι οι κόμβοι έχουν πολλαπλούς ρόλους, είτε του αποστολέα-πηγή, είτε του παραλήπτη-αποδέκτη, είτε ενός ενδιάμεσου συστήματος με σκοπό την προώθηση των πακέτων, ανάλογα φυσικά την τοπολογία του δικτύου. Η τοπολογία αυτή διαμορφώνεται κατά τρόπο τυχαίο καθώς μία ακόμα ιδιότητα των ad hoc δικτύων είναι ότι οι κόμβοι τους έχουν κινητικότητα (mobility), δηλαδή μπορούν να εισέρχονται και να εξέρχονται από το δίκτυο απρόβλεπτα χωρίς να υπάρχει κάποιος περιορισμός. Ακόμα, τα πιο δημοφιλή ad hoc δίκτυα είναι τα mobile ad hoc δίκτυα, αλλιώς γνωστά και ως MANET (Mobile Ad hoc NETworks), όπου οι κόμβοι είναι κινητοί/ κινούμενοι. Άλλη μία κατηγορία είναι τα ασύρματα δίκτυα αισθητήρων (WSN-Wireless Sensor Networks) τα οποία χρησιμοποιούν αισθητήρες ώστε να καταγράφουν μετρήσεις, όπως για παράδειγμα μέτρηση της θερμοκρασίας, και να τις παρέχουν σε πραγματικό χρόνο. Είναι σημαντικό ότι τέτοια δίκτυα έχουν υψηλή προσαρμοστικότητα ώστε να συνδέονται τόσο κόμβοι διαφορετικού τύπου, όπως κινητές συσκευές με αισθητήρες, όσο και διαφορετικών χαρακτηριστικών, όπως διάρκεια ζωής, υπολογιστική ισχύ, εμβέλεια κτλ.



Εικόνα 5: Ασύρματο ad hoc δίκτυο [36]

2.1.2 Πρωτόκολλα επικοινωνίας

2.1.2.1 Ταξινόμηση

Μπορούμε να ταξινομήσουμε τα πρωτόκολλα δρομολόγησης (routing protocols), ή αλλιώς πρωτόκολλα επικοινωνίας, για τα ad hoc δίκτυα σύμφωνα με διάφορα κριτήρια. Είτε με βάση τον αλγόριθμο δρομολόγησης (π.χ. τον αλγόριθμο Dijkstra) που χρησιμοποιούν, είτε με την τοπολογία των κόμβων του δικτύου, είτε σε ποιο επίπεδο καθορίζεται η δρομολόγηση, είτε άλλους παράγοντες τους οποίους θέλουμε να εξετάσουμε.

Με βάση τη διαδικασία εύρεσης βέλτιστων διαδρομών ταξινομούμε τα πρωτόκολλα δρομολόγησης σε δύο κύρια είδη, τα προνοητικά (proactive) και τα αντιδραστικά (re-active) πρωτόκολλα δρομολόγησης. Τα πρώτα, προκειμένου ένας κόμβος να στείλει ένα πακέτο, εξουσιοδοτείται να ανακαλύψει τις διαδρομές σε όλους τους πιθανούς προορισμούς του δικτύου έτσι ώστε να είναι ήδη γνωστή η διαδρομή που θα ακολουθήσει. Για τη διαδικασία αυτή χρησιμοποιούνται πίνακες με διανύσματα απόστασης και την κατάσταση όλων των συνδέσεων. Τα πρωτόκολλα της δεύτερης κατηγορίας λέγονται αλλιώς on-demand και υιοθετούν μία διαφορετική προσέγγιση ώστε ένας κόμβος να βρίσκει μία διαδρομή για έναν προορισμό μόνο όταν αυτό απαιτείται. Γίνεται κατανοητό, ότι η κύρια διαφορά τους είναι ότι ένας προνοητικός αλγόριθμος για ένα πρωτόκολλο δρομολόγησης έχει ως στόχο να δημιουργήσει τις συνθήκες εκείνες που θα επιτρέψουν αργότερα μία διαδικασία δρομολόγησης, ενώ ένας αντιδραστικός αλγόριθμος επιλύει κάθε πρόβλημα την στιγμή που αυτό εμφανίζεται. Έτσι, η πρώτη κατηγορία πρωτόκολλων δρομολόγησης βασίζεται στη συνέπεια των κόμβων να ανανεώνουν και να ελέγχουν τις πληροφορίες που απαιτούνται για κάθε διαδρομή. Στην πρώτη κατηγορία, οι συνεπείς και ανανεωμένες πληροφορίες δρομολόγησης κρατούνται στους κόμβους, ενώ στη δεύτερη οι διαδρομές δρομολόγησης δημιουργούνται μόνο όταν αυτό απαιτείται από την πηγή, και διατηρούνται για όσο αυτή τις χρειάζεται.

2.1.2.2 Πρωτόκολλο DSR

Το DSR (Dynamic Sourse Routing) [37], είναι ένα on-demand πρωτόκολλο δρομολόγησης το οποίο βασίζεται στην έννοια της δρομολόγησης πηγής. Η λειτουργία του βασίζεται στο γεγονός ότι κάθε κόμβος διατηρεί κρυφές μνήμες διαδρομών (Route Cache) που περιέχουν καταχωρημένες τις διαδρομές πηγής τις οποίες γνωρίζει. Οι εγγραφές στην κρυφή μνήμη των διαδρομών ενημερώνονται συνεχώς, καθώς άλλες νέες διαδρομές ανακαλύπτονται.

Το πρωτόκολλο αποτελείται από δύο μηχανισμούς:

α) την ανακάλυψη διαδρομών (route discovery) και

β) τη συντήρηση ή διατήρηση ήδη υπαρχόντων διαδρομών (route maintenance).

Όταν ένας κόμβος θέλει να αποστείλει ένα πακέτο σε κάποιον προορισμό, ελέγχει την κρυφή μνήμη διαδρομών του για να βρει αν έχει ήδη κάποια διαδρομή για τον προορισμό αυτό. Εάν βρει ότι υπάρχει μία ισχύουσα διαδρομή για τον προορισμό θα χρησιμοποιήσει τη διαδρομή αυτή για τη μετάδοση των δεδομένων. Αν όμως δεν υπάρχει μία τέτοια διαδρομή, τότε ενεργοποιείται η διαδικασία εύρεσης διαδρομής με τη μετάδοση ενός μηνύματος αίτησης νέας διαδρομής. Αυτό το μήνυμα αίτησης διαδρομής, περιέχει τη διεύθυνση της πηγής και του κόμβου προέλευσης και ένα μοναδικό αριθμό αναγνώρισης ταυτότητας. Κάθε ενδιάμεσος κόμβος που λαμβάνει αυτό το μήνυμα, ελέγχει αν έχει αποθηκευμένη μία διαδρομή για τον συγκεκριμένο προορισμό. Αν δεν ξέρει μία τέτοια διαδρομή, προσθέτει τη διεύθυνσή του στο αρχείο διαδρομής του πακέτου και στη συνέχεια προωθεί το μήνυμα στους γειτονικούς του κόμβους. Μία απάντηση σε ένα αίτημα εύρεσης μίας διαδρομής παράγεται, είτε όταν παραληφθεί το εν λόγω μήνυμα από τον κόμβο προορισμού, είτε όταν ένας ενδιάμεσος κόμβος περιέχει στην κρυφή μνήμη του μία ισχύουσα διαδρομή προς τον προορισμό. Στο μήνυμα αποθηκεύεται όλη η αλληλουχία των κόμβων από την οποία έχει περάσει το πακέτο, έως ότου φτάσει στον κόμβο προορισμού ή σε έναν ενδιάμεσο κόμβο και δημιουργηθεί μία απάντηση για τη διαδρομή (route reply).

Ο μηχανισμός για τη διατήρηση διαδρομής επιτυγχάνεται επίσης μέσω της χρήσης δύο ειδών πακέτων: τα πακέτα λάθους διαδρομής (route error) και τις επιβεβαιώσεις (aknowledgement signal). Ένα πακέτο λάθους διαδρομής παράγεται σε ένα κόμβο όταν παρουσιαστεί ένα πρόβλημα μετάδοσης δεδομένων στο επίπεδο συνδέσεων του δικτύου. Όταν ένα τέτοιο πακέτο παραληφθεί από ένα κόμβο, η καταχωρημένη διαδρομή στην οποία παρουσιάστηκε το λάθος, καθώς και όλες οι άλλες, που περιέχουν το σύνδεσμο στον οποίο παρουσιάστηκε το πρόβλημα, αφαιρούνται από την κρυφή μνήμη των διαδρομών του. Τα πακέτα επιβεβαιώσεων χρησιμοποιούνται για να ελέγξουν τη σωστή λειτουργία των συνδέσεων των διαδρομών.

Ο δυναμικός αλγόριθμος που χρησιμοποιεί το πρωτόκολλο DSR τον καθιστά ως το χαρακτηριστικότερο πρωτόκολλο δρομολόγησης πηγής, το οποίο εύκολα μπορεί να προσαρμοστεί και να λειτουργήσει κάτω από οποιεσδήποτε συνθήκες του δικτύου.

2.1.3 Προβλήματα δρομολόγησης

Σε ένα δίκτυο χρησιμοποιούμε τον όρο διαδρομή (path) για να ορίσουμε την ακολουθία των κόμβων μέσω των οποίων θα φτάσουν τα πακέτα δεδομένων στον επιθυμητό προορισμό τους. Η διαδικασία εύρεσης τέτοιων διαδρομών ονομάζεται δρομολόγηση (routing) και είναι μία από τις ουσιώδεις λειτουργίες που υποστηρίζει κάθε δίκτυο. Η δρομολόγηση στα δίκτυα ad hoc έγινε αντικείμενο έρευνας καθώς λόγω των ιδιαίτερων χαρακτηριστικών των δικτύων τα ήδη υπάρχοντα πρωτόκολλα δρομολόγησης για τα ενσύρματα δίκτυα δεν ήταν αποδοτικά.

Όπως είναι φυσικό, επειδή τα ad hoc δίκτυα έχουν ιδιαίτερα δυναμικά μεταβαλλόμενη τοπολογία και λόγω της έλλειψης κάποιας κεντρικής διαχείρισης δεν μπορούν να χρησιμοποιήσουν τα παραδοσιακά πρωτόκολλα των ενσύρματων δικτύων σε θέματα δρομολόγησης. Είναι σημαντικό, τα πρωτόκολλα που θα χρησιμοποιηθούν να μπορούν να έχουν προσαρμοστικό χαρακτήρα και να τροποποιούν κατάλληλα τις παραμέτρους τους ανάλογα την συνολική κατάσταση του δικτύου. Για παράδειγμα, ένας αλγόριθμος δρομολόγησης για την καλύτερη απόδοση λειτουργίας του δικτύου θα πρέπει να προσαρμόζεται γρήγορα στις αλλαγές ώστε να ανακαλύπτει και να υπολογίζει τις βέλτιστες διαδρομές ανάλογα με τη θέση των κόμβων κάθε χρονική στιγμή. Γενικά, έχει παρατηρηθεί ότι το ποσοστό των πακέτων που χάνονται στα ασύρματα δίκτυα είναι αυξημένο καθώς εξαιτίας της μετακίνησης των κόμβων υπάρχει αυξημένη πιθανότητα διακοπής των συνδέσεων. Παράλληλα, τα ad hoc δίκτυα είναι ευπαθή σε επιθέσεις (attacks) και θέματα ασφάλειας (security) όχι μόνο από το γεγονός της απουσίας κεντρικής παρακολούθησης, αλλά και από το γεγονός ότι στο δίκτυο μπορεί να συνδεθεί οποιοσδήποτε κόμβος, ακόμα και κάποιος που σκοπό έχει να βλάψει το δίκτυο. Έτσι, σε σχέση με τα δίκτυα που διαθέτουν σταθερή υποδομή, η υιοθέτηση πιο πολύπλοκων πρωτόκολλων και τεχνολογιών κρίνεται αναγκαία και απαραίτητη για τη συνεργασία όλων των κόμβων.

2.2 Το Πρωτόκολλο CONFIDANT των ad hoc δικτύων

2.2.1 Σκοπός

Για την ορθή λειτουργία ενός ad hoc δικτύου, πρέπει οι κόμβοι που συμμετέχουν στη διαδικασία της δρομολόγησης να συνεργάζονται με τον αποδοτικότερο τρόπο. Για το σκοπό αυτό υιοθετήθηκαν διάφοροι μηχανισμοί υποστήριξης συνεργασίας των κόμβων πάνω στα ήδη γνωστά πρωτόκολλα δρομολόγησης. Ένα τέτοιο πρωτόκολλο ονομάζεται CONFIDANT [38] (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks) και λειτουργεί επιπρόσθετα με το πρωτόκολλο DSR που αναλύσαμε παραπάνω.

Για την αποφυγή επιθέσεων κατά τη διαδικασία της δρομολόγησης το πρωτόκολλο CONFIDANT στοχεύει στην προστασία από τους εξής τύπους ανεπιθύμητων συμπεριφορών:

- να μην προωθούν δεδομένα ή μηνύματα ελέγχου
- να στέλνουν ασυνήθιστα πολλά μηνύματα στους γειτονικούς τους για "καλές" διαδρομές δρομολόγησης ή αντίθετα να στέλνουν μόνο μη συμφέρουσες διαδρομές
- να αποστέλλουν μηνύματα για σφάλματα σε διαδρομές χωρίς όμως να υπάρχουν στην πραγματικότητα τέτοια σφάλματα ή και το αντίθετο να μην αποστέλλουν σφάλματα διαδρομών που όμως υπάρχουν
- να ανανεώνουν πολύ πιο συχνά από το αναμενόμενο τις διαδρομές δρομολόγησης
- να αλλάζουν διαδρομή δρομολόγησης με τρόπο όχι εμφανή στους υπόλοιπους κόμβους, όπως με αλλοίωση της επικεφαλίδας των μηνυμάτων ελέγχου και των πακέτων

Το πρωτόκολλο CONFIDANT έχει ως σκοπό να παρεμποδίσει τυχόν τέτοιες ανεπιθύμητες συμπεριφορές στους κόμβους που τις υιοθετούν, οι ονομαζόμενοι κακόβουλοι για το δίκτυο, με το να απομονώνονται από τους υπόλοιπους. Βασικός λοιπόν στόχος είναι η πρόληψη συμπεριφορών που δεν συμβαδίζουν με την ομαλή λειτουργία του δικτύου μέσω του εντοπισμού των κόμβων που τις δημιουργούν.

Όταν ανιχνευθεί ένας κακόβουλος κόμβος, υπάρχει ένας μηχανισμός, που θα αναλυθεί περισσότερο παρακάτω, με τον οποίο ενημερώνονται οι κόμβοι του δικτύου προκειμένου να πάρουν κάποιες αποφάσεις σχετικά με το πως θα εξελιχθεί η μετέπειτα συνεργασία τους, πάντα με τελικό στόχο την ομαλή, όπως αυτή έχει ορισθεί, λειτουργία του συστήματος. Επίσης, η βασική ιδέα πάνω στην οποία αναπτύχθηκε το πρωτόκολλο CONFIDANT είναι ότι επειδή οι κόμβοι πρέπει να συνεργάζονται μεταξύ τους αυτοί οι οποίοι με ενέργειές τους παρεμποδίζουν αυτή τη συνεργασία προτρέπουν τους υπόλοιπους να τους απομονώσουν. Η

απομόνωσή τους αυτή είναι αναγκαία καθώς αλλιώς ζημιώνουν τις υπηρεσίες του δικτύου στο σύνολό τους και το καθιστούν αναξιόπιστο. Όσο πιο γρήγορα οι κακόβουλοι κόμβοι απομακρυνθούν από το δίκτυο τόσο πιο άμεσα θα φτάσουν τα πακέτα στον προορισμό τους και άρα το δίκτυο θα λειτουργεί με το βέλτιστο τρόπο εφόσον θα περιλαμβάνει μόνο κόμβους οι οποίοι θα εξυπηρετούν το σκοπό για τον οποίο έχει δημιουργηθεί.

2.2.2 Λειτουργίες και μηχανισμοί

Συνοπτικά, οι βασικές λειτουργίες του πρωτόκολλου CONFIDANT [38] είναι τέσσερις (4):

1) Η ανίχνευση των κακόβουλων-μη ομαλά συμπεριφερόμενων κόμβων που υπάρχουν στο δίκτυο από τους γειτονικούς τους
2) Η προειδοποίηση των κόμβων του δικτύου για την ύπαρξη των κακόβουλων κόμβων
3) Η δημιουργία καλής (ή κακής) φήμης των κόμβων ανάλογα με τη δραστηριότητά τους μέσα στο δίκτυο
4) Η τελική θέσπιση ή ακόμα και διαγραφή μονοπατιών από τα οποία δρομολογούνται τα πακέτα που στέλνονται μέσω των κακόβουλων-μη ομαλά συμπεριφερόμενων κόμβων.

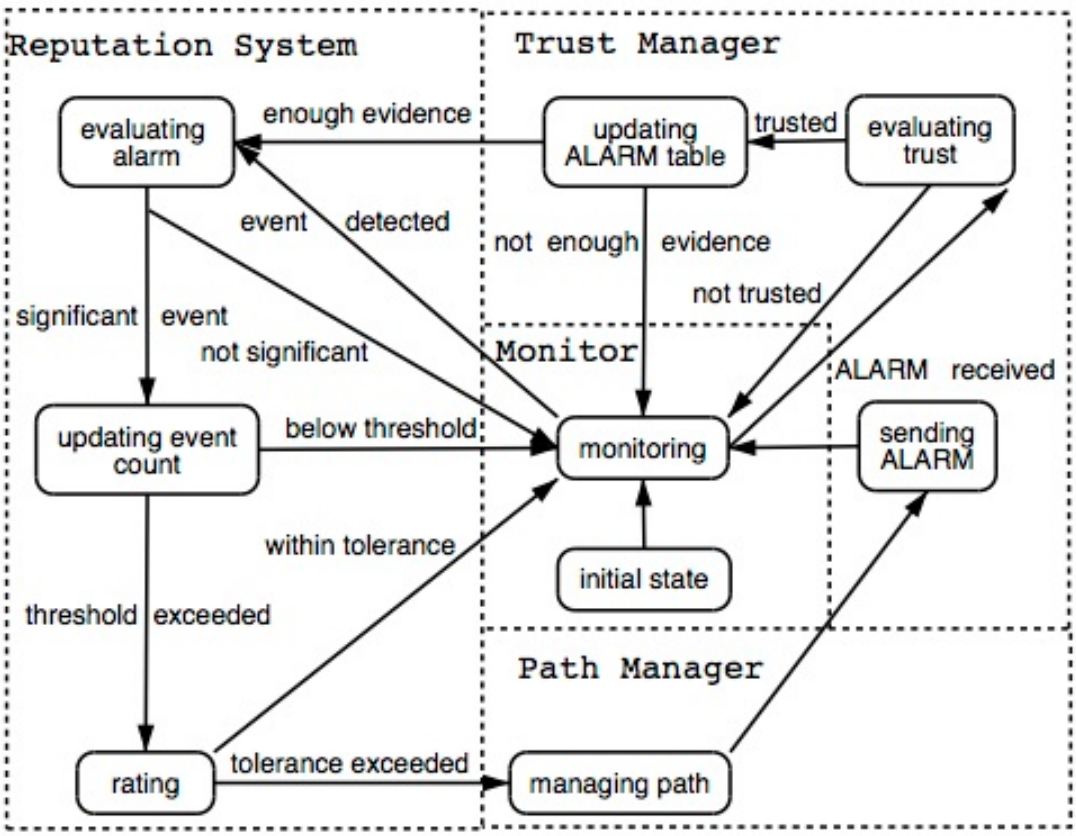
Βάσει αυτών των λειτουργιών το πρωτόκολλο CONFIDANT έχει εργαλεία που υλοποιούν κάθε μία λειτουργία:

1) Monitor – ΑΝΙΧΝΕΥΣΗ
2) Trust Manager – ΠΡΟΕΙΔΟΠΟΙΗΣΗ
3) Reputation System – ΦΗΜΗ
4) Path Manager – ΘΕΣΠΙΣΗ ΜΟΝΟΠΑΤΙΩΝ

Οι μηχανισμοί έχουν τη δυνατότητα να αλληλεπιδρούν μεταξύ τους, αλλά ενεργοποιούνται με τη σειρά που αναφέρθηκαν. Σε κάθε κόμβο του δικτύου υπάρχουν ενσωματωμένα αυτά τα εργαλεία, για να υλοποιούν τους παραπάνω μηχανισμούς, άρα κάθε κόμβος του δικτύου είναι σε θέση να συλλέγει στοιχεία σχετικά με άλλους κόμβους του δικτύου ώστε να παίρνει αποφάσεις σχετικά με τη δική του λειτουργία στο σύστημα. Παράλληλα, μπορεί να στέλνει μηνύματα στους γειτονικούς του κόμβους προκειμένου να επέμβει στη δική τους λειτουργία. Υπάρχει έτσι μία συνεχής επικοινωνία μεταξύ των κόμβων που βασίζεται στη δεδομένη κατάσταση του δικτύου όσο και στην ήδη υπάρχουσα εμπειρία και είναι μία διαδικασία που ανανεώνεται σε πραγματικό χρόνο.

Πριν αναλύσουμε τις λειτουργίες του πρωτόκολλου CONFIDANT θα γίνει μία σύντομη περιγραφή των εργαλείων.

Monitor - Trust Manager - Reputation System - Path Manager



Εικόνα 6: Αρχιτεκτονική του πρωτόκολλου CONFIDANT σε επίπεδο κόμβου [38]

Το εργαλείο Monitor χρησιμοποιείται στη λειτουργία της ανίχνευσης των κόμβων εκείνων που δεν έχουν ομαλή συμπεριφορά, όπως αυτή έχει ορισθεί από το δίκτυο. Όπως και για όλα τα εργαλεία, το Monitor υπάρχει σε κάθε κόμβο του δικτύου και μπορεί να καταγράφει οποιαδήποτε ενέργεια υλοποιεί κάθε γειτονικός του κόμβος. Το Monitor μπορεί επίσης να αλληλεπιδράσει τόσο με άλλα εργαλεία του κόμβου όσο και με το εργαλείο Trust Manager κάποιου άλλου κόμβου. Έτσι, το εργαλείο Trust Manager ενός κόμβου ενεργοποιείται μέσω του εργαλείου Monitor κάποιου άλλου κόμβου όταν λάβει κάποιο μήνυμα ειδοποίησης. Τα συστατικά στοιχεία που έχει στη διάθεση του είναι τα εξής:

- Έναν πίνακα προειδοποιήσεων με πληροφορίες από όλα τα μηνύματα ειδοποίησης που έχει λάβει ο συγκεκριμένος κόμβος
- Έναν πίνακα εμπιστοσύνης για να ελέγχεται κάθε φορά που στέλνεται ένα μήνυμα ειδοποίησης η αξιοπιστία του
- Μία λίστα των κόμβων εκείνων που μπορεί να στέλνει ο κόμβος μηνύματα ειδοποίησης, όταν αυτό κρίνεται φυσικά απαραίτητο

Ο πίνακας εμπιστοσύνης, που έχει στη διάθεσή του το εργαλείο Trust Manager, δημιουργείται από το στοιχείο Reputation System. Το Reputation System είναι υπεύθυνο για τη βαθμολόγηση των κόμβων του δικτύου με βάση τη συμμετοχή τους στη διαδικασία της δρομολόγησης. Για παράδειγμα, ένας κόμβος που προωθεί τα πακέτα όπως προβλέπεται θα έχει λάβει καλή φήμη, ενώ αντίστοιχα ένας κόμβος που μπορεί να μην προωθεί τα πακέτα ή αλλοιώνει το περιεχόμενό τους θα λάβει από τους κόμβους που έρχεται σε επικοινωνία βαθμολογία που θα τον οδηγήσει στο γεγονός να έχει κακή φήμη. Στο πρωτόκολλο CONFIDANT το εργαλείο Reputation Manager έχει στη διάθεσή του έναν πίνακα με τους κόμβους και την αντίστοιχη βαθμολογία τους. Η βαθμολογία αυτή αλλάζει μονάχα όταν υπάρχει επαρκής απόδειξη για την ύπαρξη κακόβουλης συμπεριφοράς κάποιου κόμβου. Η βαθμολογία λοιπόν ρυθμίζεται μέσω μίας συνάρτησης που λαμβάνει παραμέτρους που έχουν διαφορετικό συντελεστή βαρύτητας αναλόγως την προέλευσή τους. Οι παράμετροι αυτοί είναι ουσιαστικά οι παρατηρήσεις των κόμβων και είναι λογικό η παράμετρος σχετικά με τις παρατηρήσεις που βασίζονται στην εμπειρία του ίδιου κόμβου να έχουν μεγαλύτερη βαρύτητα σε σχέση με αυτές που ο κόμβος λαμβάνει από τους γειτονικούς του με βάση τη δική τους εμπειρία. Η βαθμολογία του κόμβου στον πίνακα εμπιστοσύνης μπορεί να αλλάζει συνεχώς και όταν “πέσει” κάτω από ένα προεπιλεγμένο επιτρεπτό όριο, τότε ο κόμβος θεωρείται κακόβουλος. Στη συνέχεια, αφού ο κόμβος χαρακτηριστεί ως κακόβουλος, καλείται το τέταρτο και τελευταίο εργαλείο ενός κόμβου, που ονομάζεται Path Manager, με σκοπό να λάβει δράση για να επιτευχθεί με το σωστότερο τρόπο η διαδικασία της δρομολόγησης.

Συνοπτικά οι λειτουργίες του είναι οι παρακάτω:

1) ανάλογα με τη φήμη κάθε κόμβου, όπως αυτή έχει ορισθεί από το στοιχείο Reputation Manager, ανακατατάσσει τη διαδρομή δρομολόγησης προκειμένου τα πακέτα να περνάνε από κόμβους με όσο το δυνατόν πιο καλή φήμη
2) τα μονοπάτια που περιέχουν κακόβολους κόμβους να διαγράφονται
3) να αγνοούνται οι αιτήσεις από κακόβολους κόμβους ή να μην υπάρχει απάντηση σε αυτές
4) σε περιπτώσεις όπου υπάρχει αίτηση για διαδρομή δρομολόγησης στην οποία υπάρχουν κόμβοι κακόβουλοι, είτε πάλι να αγνοείται είτε να αναλαμβάνει δράση ώστε να ειδοποιηθεί και η πηγή.

2.2.3 Περιγραφή λειτουργίας

Εφόσον είδαμε με αναλυτικό τρόπο τις λειτουργίες των στοιχείων του πρωτόκολλου CONFIDANT, μπορούμε να περιγράψουμε τη λειτουργία του.

- Εσωτερική διαδικασία στο επίπεδο κάθε κόμβου.

Κάθε κόμβος μέσω του Monitor είναι σε θέση να παρακολουθεί ανά πάσα στιγμή τη συμπεριφορά όλων των γειτονικών του κόμβων. Ως γειτονικοί κόμβοι προσδιορίζονται αυτοί οι οποίοι βρίσκονται ένα βήμα, αλλιώς hop, μακριά από τον κόμβο. Αν παρατηρηθεί κάποιο γεγονός από γειτονικό κόμβο που ξεφεύγει από την ομαλή συμπεριφορά του, τότε αυτό το γεγονός δίνεται ως πληροφορία στο Reputation System του ίδιου του κόμβου προκειμένου να επεξεργαστεί περαιτέρω αξιολόγηση. Αφού αξιολογηθεί και κριθεί όντως για κακόβουλη συμπεριφορά και όχι απλώς ένα τυχαίο γεγονός ή λάθος, ανανεώνεται ο πίνακας βαθμολογίας των κόμβων προκειμένου να αλλάξει η βαθμολογία των κόμβων που παρατηρήθηκε η κακόβουλη συμπεριφορά και να συγκριθεί με το επιτρεπτό όριο που έχει ορισθεί. Εάν ξεπεραστεί αυτό το όριο η πληροφορία στέλνεται και στο Path Manager ώστε να δημιουργήσει τις καταλληλότερες διαδρομές δρομολόγησης χωρίς την ύπαρξη των κόμβων εκείνων που έχουν κακή αξιολόγηση από το Reputation System. Ο κόμβος, ανεξάρτητα από το αποτέλεσμα της αξιολόγησης, συνεχίζει την διαδικασία ανίχνευσης των γειτονικών του κόμβων για την ύπαρξη κακόβουλων, πάντα μέσα από το Monitor.

- Επικοινωνία των κόμβων σε συνολικό επίπεδο, συνεργασία

Δεν πρέπει να ξεχνιέται ότι ο στόχος των κόμβων είναι η συνεργασία μεταξύ τους προκειμένου να βοηθούν ο ένας τον άλλο στη διαδικασία της δρομολόγησης. Για το λόγο αυτό, όταν ένας κόμβος ανιχνεύσει στους γειτονικούς του έναν κακόβουλο κόμβο ειδοποιεί αμέσως και το υπόλοιπο δίκτυο στέλνοντας μηνύματα ειδοποίησης (ALARM).

Κάθε μήνυμα ALARM ακολουθεί την εξής διαδρομή:

Το στοιχείο Trust Manager του κόμβου που ανίχνευσε τον κακόβουλο κόμβο αναλαμβάνει να αποστείλει ένα μήνυμα ALARM για ενημέρωση. Το περιεχόμενο του μηνύματος περιέχει όλες εκείνες τις χρήσιμες πληροφορίες σχετικά με τον χαρακτηριστικό τύπο της παραβίασης του πρωτόκολλου, τη διεύθυνση του κόμβου που γίνεται η αναφορά, τον αριθμό των περιστατικών που έχουν παρατηρηθεί στον συγκεκριμένο κόμβο, τη διεύθυνση του κόμβου που παρατήρησε την κακόβουλη συμπεριφορά κ.α. Όταν στη συνέχεια το στοιχείο Monitor ενός κόμβου-παραλήπτη λάβει κάποιο μήνυμα ALARM, από το Trust Manager του κόμβου-αποστολέα που εντόπισε τη κακόβουλη συμπεριφορά, το μεταφέρει στο Trust Manager για να αξιολογηθεί η εγκυρότητα της πηγής αυτού του μηνύματος. Αν με βάση το ιστορικό που έχει κρατήσει το Trust Manager κριθεί τελικά ότι ο κόμβος που έστειλε το μήνυμα ALARM είναι έμπιστος τότε ανανεώνεται και ο πίνακας εμπιστοσύνης.

Μετά από αρκετές παρόμοιες αναφορές για ένα κόμβο μέσω μηνυμάτων ALARM από άλλους διαφορετικούς κόμβους μπορεί να έχει αποτέλεσμα στη βαθμολόγηση που έχει αποθηκευμένο το στοιχείο Reputation System. Η βαθμολόγηση αυτή έχει πρώτα αξιολογηθεί για τη σημαντικότητά της τόσο από τον αριθμό των αναφορών όσο και από το ήδη υπάρχον ιστορικό από την εμπειρία του κόμβου. Με αυτό τον τρόπο, ένας κόμβος που έχει κατηγορηθεί για ύποπτη συμπεριφορά χαρακτηρίζεται τελικά κακόβουλος μόνο εφόσον υπάρχουν επαρκείς αποδείξεις ότι είναι όντως κακόβουλος.

2.3 Σύγκριση ad hoc και Κοινωνικών Δικτύων

Όπως ένα ad hoc δίκτυο είναι ένα σύνολο από κόμβους-δρομολογητές που συνδέονται μεταξύ τους προκειμένου την ανταλλαγή πακέτων πληροφοριών έτσι και ένα κοινωνικό δίκτυο είναι ένα σύνολο ανθρώπων-χρηστών που συνδέονται αναπτύσσοντας κοινωνικές σχέσεις όπως φιλία, συνεργασία και ανταλλάσοντας πληροφοριές και πόρους. Οι πόροι αυτοί που ανταλλάσουν οι κόμβοι ή οι χρήστες είναι το βασικό συστατικό εκείνο το οποίο τους κρατάει συνδεδεμένους μέσα στο δίκτυο. Σε ένα κοινωνικό δίκτυο κάθε είδος ανταλλασσόμενου πόρου μπορεί να το δει κάποιος σαν μια σχέση, και τα άτομα που τη διατηρούν λέγεται ότι διατηρούν έναν δεσμό. Όπως τα ad hoc δίκτυα έχουν ως βασικό συστατικό τους δρομολογητές έτσι και σε ένα κοινωνικό δίκτυο βασικό συστατικό είναι οι χρήστες. Στο κοινωνικό δίκτυο, κάθε είδος ανταλλασσόμενου πόρου θεωρείται σαν μια σχέση, και τα άτομα που διατηρούν μια τέτοια σχέση λέγεται ότι διατηρούν έναν δεσμό.

Όμως, όπως έχουμε εξηγήσει, ένα ad hoc δίκτυο είναι ένα αποκεντρωμένο δίκτυο, δηλαδή κάθε δρομολογητής είναι σε θέση να εκτελεί μόνος του εντολές, να αποθηκεύει δεδομένα και να τα διαχειρίζεται ανάλογα, προκειμένου να πάρει αποφάσεις σχετικές με τη λειτουργία κάθε δεδομένη στιγμή. Κάτι τέτοιο δεν μπορεί να συμβαίνει στους χρήστες ενός κοινωνικού δικτύου καθώς ένα κοινωνικό δίκτυο είναι ένας κεντροποιημένος τύπος δικτύου. Κάθε τέτοιο δίκτυο έχει μία κεντρική μονάδα που ελέγχει όλους τους υπόλοιπους κόμβους όπως ό,τι αποθηκεύει, τις ενέργειές τους, τις αλληλεπιδράσεις με τους άλλους χρήστες, κτλ. Παράλληλα, παρακολουθεί όλα αυτά τα δεδομένα και μπορεί να επέμβει στο τρόπο που δραστηριοποιούνται οι χρήστες στο δίκτυο. Όμως όλοι οι χρήστες είναι σε θέση να συμμετέχουν στις ίδιες λειτουργίες και καθένας τις αξιοποιεί με το δικό του μοναδικό τρόπο. Αυτό σημαίνει ότι ο κάθε χρήστης είναι ανεξάρτητος και ισότιμος ως προς τους υπόλοιπους, αλλά δεν είναι αυτόνομος σε σχέση με το δίκτυο και τους κανόνες που το διέπουν. Οι κανόνες αυτοί ορίζονται και ελέγχονται συνεχώς από την κεντρική μονάδα.

Τελικά, παρότι οι ενέργειες των κόμβων και στα δυο δίκτυα είναι ως ένα βαθμό συγκεκριμένες ανάλογα με το σκοπό λειτουργίας του δικτύου, στα μη αποκεντρωποιημένα δίκτυα, όπως τα κοινωνικά δίκτυα, είναι και ελεγχόμενες από την κεντρική μονάδα ελέγχου.

Μπορούμε να αναπαραστήσουμε και τα δύο δίκτυα με τη βοήθεια γραφημάτων, που αποτελούνται από κόμβους και ακμές και να μελετήσουμε κάποια χαρακτηριστικά τους.

Κεφάλαιο 3

3.1 Ένα προτεινόμενο Μοντέλο για την Εμπιστοσύνη στα Κοινωνικά Δίκτυα

Τα στάδια του μοντέλου εμπιστοσύνης (trust) είναι τέσσερα (4), όπως στα ad hoc δίκτυα ήταν τέσσερις οι λειτουργίες του πρωτόκολλου:

1. ΑΝΙΧΝΕΥΣΗ
2. ΠΡΟΕΙΔΟΠΟΙΗΣΗ
3. ΦΗΜΗ
4. ΤΕΛΙΚΗ ΑΝΑΠΡΟΣΑΡΜΟΓΗ

Σε κάθε στάδιο του μοντέλου θα γίνεται μία αρκετά σύντομη αναφορά στην κάθε λειτουργία του πρωτόκολλου CONFIDANT [38] όπως έχει σχεδιαστεί για τα ad hoc δίκτυα και στη συνέχεια υπάρχει αναλυτική περιγραφή της μοντελοποίησης σε ένα κοινωνικό δίκτυο.

3.1.1 Ανίχνευση

Στα ad hoc δίκτυα κάθε κόμβος γνωρίζει τη λειτουργία των γειτονικών του κόμβων κρατώντας αντίγραφο των πακέτων που στέλνει μέσω του μηχανισμού Monitor. Σκοπός είναι η καταγραφή οποιαδήποτε απόκλισης από την ομαλή και συνήθη συμπεριφορά τους για την εύκολη ανίχνευση ενός κακόβουλου κόμβου.

Η διαδικασία της ανίχνευσης στα κοινωνικά δίκτυα είναι διαφορετική αφού δεν υπάρχει κάποιος έτοιμος μηχανισμός που να μας δίνει όλες εκείνες τις πληροφορίες που να φανερώνουν αυτόματα κάποια μη ομαλή συμπεριφορά ενός χρήστη. Για το λόγο αυτό, με δεδομένες κάποιες ενέργειές του μέσα στο εκάστοτε κοινωνικό δίκτυο, πρέπει σε πρώτο στάδιο να γίνει η ανίχνευση των απομονωμένων χρηστών του, δηλαδή εκείνων που είναι πιο απομακρυσμένοι από το υπόλοιπο δίκτυο. Υπάρχουν δύο κατηγορίες απομονωμένων χρηστών (isolated users), οι ενεργοί (active) και οι ανενεργοί (inactive). Με τον όρο ενεργό απομονωμένο ορίζουμε το χρήστη εκείνον του οποίου οι φίλοι δεν έχουν μεταξύ τους άλλους κοινούς φίλους. Αυτό συμβαίνει καθώς οι χρήστες έχουν την τάση να γνωρίζουν άλλους χρήστες μέσω κοινών φίλων κάτι το οποίο αυξάνει την πιθανότητα δύο φίλοι ενός χρήστη να είναι επίσης φίλοι. Με τον όρο ανενεργό απομονωμένο ορίζουμε το χρήστη εκείνον που έχει κάποιο σημαντικό σχετικά διάστημα να χρησιμοποιήσει τον λογαριασμό του για οποιαδήποτε ενέργεια μέσα στο κοινωνικό δίκτυο. Η διαδικασία εύρεσης των χρηστών αυτών γίνεται με χρήση απλών εργαλείων που διαθέτει κάθε σύγχρονο κοινωνικό δίκτυο, αλλά και πιο πολύπλοκων που ειδικεύονται στην αναπαράσταση των κοινωνικών

δικτύων σε γραφήματα όπου κάθε χρήστης είναι ένας κόμβος με ακμές τις σχέσεις φιλίας που τον συνδέουν με τους υπόλοιπους, όπως έχει εξηγηθεί στο Κεφάλαιο 1, στην ενότητα 1.2.5 Ανάλυση των Κοινωνικών Δικτύων. Κάποιος χρήστης που ανιχνεύεται ως απομονωμένος θεωρείται σε αυτό το στάδιο ύποπτος για μη ομαλή συμπεριφορά.

Πιθανοί επίσης ύποπτοι κακόβουλοι χρήστες είναι και εκείνοι που δεν ικανοποιούν κάποιους άλλους παράγοντες που προδίδουν μία ομαλή συμπεριφορά τους στα πλαίσια της λειτουργίας κάθε κοινωνικού δικτύου. Για παράδειγμα, ένας ισχυρός δεσμός μεταξύ δύο ανθρώπων αποτελεί αναμφισβήτητα κάποια συγγένεια, άρα η εμφάνισή της στο προσωπικό προφίλ ενός χρήστη ύστερα και από βεβαίωση του χρήστη-συγγενή για την ορθότητα της είναι μία απόδειξη της πραγματικής ύπαρξης των χρηστών όπως έχουν παρουσιαστεί στο διαδικτυακό ιστότοπο.

Παρόμοιες αρκετά σημαντικές ενδείξεις μπορεί κανείς να θεωρήσει μία επαγγελματική σχέση, μία φιλική σχέση, αν δύο άτομα υπήρξαν κάποτε συμμαθητές, συμφοιτητές, αν έχουν παρευρεθεί σε ίδιες εκδηλώσεις ή έχουν συμμετάσχει σε κοινές δράσεις. Με άλλα λόγια τέτοια στοιχεία μεταξύ δύο χρηστών που μπορεί κανείς εύκολα να τα αντλήσει από έναν ιστόχωρο κοινωνικής δικτύωσης αποτελούν ενδείξεις ότι οι χρήστες γνωρίζονται και μέσω της ιστοσελίδας διατηρούν κάποια επαφή, όπως ότι σχολιάζουν και κοινοποιούν κοινές αναρτήσεις. Όταν η επαφή αυτή δεν υπάρχει μεταξύ ενός χρήστη και των φίλων του μπορεί να τον "προδώσει" για τη μη ομαλή συμπεριφορά του.

Τέλος, σημαντικό είναι να υπάρχουν ομοιότητες στις προτιμήσεις και στα κοινά ενδιαφέροντα των χρηστών που είναι φίλοι μεταξύ τους καθώς χρήστες που τους αρέσουν ίδιες δραστηριότητες είναι πιθανότερο να είναι φίλοι και στην πραγματικότητα.

Επίσης, όπως και στην πραγματική ζωή, η απόσταση (residential location distance) επηρεάζει μία φιλία, εφόσον όσο πιο κοντά βρίσκονται δύο ατόμα τόσο το πιο πιθανό είναι να γνωρίζονται στην πράξη και όχι μόνο διαδικτυακά. Άνθρωποι που μένουν σε κοινή γειτονιά είναι και πιο φυσικό ότι έχουν συναντηθεί.

Όμως αυτό το στάδιο είναι αρχικό και οι παραπάνω παράγοντες είναι ενδείξεις, όχι αποδείξεις, ότι ένας χρήστης είναι κακόβουλος οπότε μένει αυτό να εξακριβωθεί μέσω των επόμενων διαδικασιών.

Συνοπτικά λοιπόν, θα συμβολίζουμε το πως θεωρούμε ύποπτο ένα χρήστη για μη ομαλή, ή αλλιώς κακόβουλη, συμπεριφορά με βάση τις εξής δύο κατηγορίες:

α) ΑΠΟΜΟΝΩΜΕΝΟΣ χρήστης με τις υποκατηγορίες
- ενεργός
- ανενεργός

β) ΜΗ ΑΠΟΜΟΝΩΜΕΝΟΣ χρήστης που δεν έχει

- συγγενείς
- άλλους στενούς φίλους
- φίλους με κοινά ενδιαφέροντα

Οι παράγοντες που προδίδουν μη ομαλή συμπεριφορά στην περίπτωση του μη απομονωμένου χρήστη μπορούν φυσικά να αλλάξουν ανάλογα τα χαρακτηριστικά εκείνα που ορίζουν την ταυτότητα κάθε κοινωνικού δικτύου. Όπως έχουμε δει, κάθε κοινωνικό δίκτυο έχει διαφορετικό περιεχόμενο και δομή και έτσι ένας κακόβουλος χρήστης είναι σε θέση να το προσεγγίσει με διαφορετικό τρόπο. Για το λόγο αυτό, παίρνοντας για παράδειγμα το κοινωνικό δίκτυο Twitter, ένα από τα κριτήρια που θα μπορεί να ορίζουν έναν μη απομονωμένο χρήστη μπορεί να είναι η δημοσίευση κάποιων tweets με περιεχόμενο spam. Αντίστοιχα, στο LinkedIn οι σχέσεις που ορίζονται μεταξύ των χρηστών του είναι επαγγελματικής φύσεως άρα σε αυτό το βήμα χρήσιμη θα είναι η εξακρίβωση των στοιχείων που έχει δηλώσει κάθε χρήστης σχετικά με τα πτυχία και τις συναφή πιστοποιήσεις που έχει λάβει, σεμινάρια που έχει παρακολουθήσει κτλ.

3.1.2 Προειδοποίηση

Σε κάθε κόμβο του δικτύου υπάρχει το στοιχείο Trust Manager το οποίο είναι υπεύθυνο για να στέλνει προειδοποιητικά μηνύματα ALARM στους κόμβους όταν εντοπιστεί η ύπαρξη κάποιου άλλου κακόβουλου. Όποιος κόμβος λάβει ένα μήνυμα ALARM μπαίνει στη διαδικασία να εξακριβώσει την αξιοπιστία του μέσω ενός μηχανισμού φιλτραρίσματος που γίνεται μέσω μίας σειράς ελέγχων των πιστοποιήσεων που έχουν οι κόμβοι δρομολογητές. Στη συνέχεια, στα ad hoc δίκτυα όποιος κόμβος έχει λάβει το μήνυμα ALARM το στέλνει και στους κόμβους που είναι στη λίστα των γειτονικών του προκειμένου να τους προειδοποιήσει. Αυτοί με τη σειρά τους κάνουν την ίδια διαδικασία για να εξακριβώσουν την αξιοπιστία των μηνυμάτων που έλαβαν.

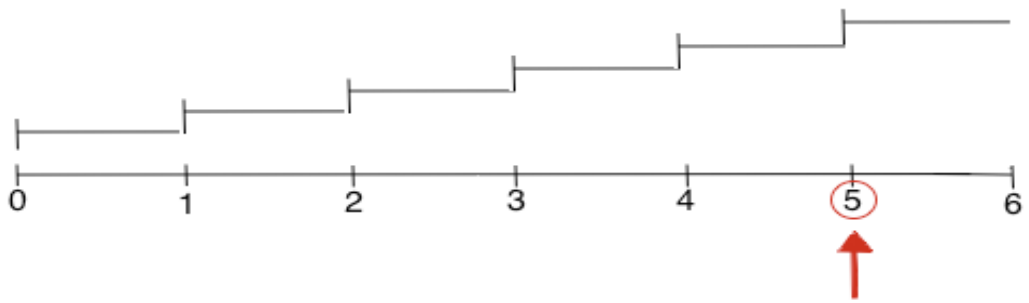
Παρόμοια διαδικασία μπορεί να εφαρμοστεί και στα κοινωνικά δίκτυα μόνο που τα μηνύματα ALARM δεν θα στέλνονται από τους χρήστες αλλά από το ίδιο το κοινωνικό δίκτυο στο χρήστη που έχει ανιχνευτεί ως ύποπτος για μη ομαλή συμπεριφορά. Έτσι, χρησιμοποιώντας την κατηγοριοποίηση που έγινε στο προηγούμενο βήμα της ανίχνευσης θα προχωρήσει η διαδικασία εξακρίβωσης των στοιχείων του χρήστη και να αποδειχθεί τελικά αν είναι όντως κακόβουλος ή όχι. Πιο συγκεκριμένα, στο χρήστη που έχει κατηγοριοποιηθεί ως απομονωμένος, είτε ενεργός είτε ανενεργός, θα στέλνεται ένα μήνυμα ALARM1 που σκοπό θα έχει να εξακριβώσει αν τελικά ο χρήστης είναι πραγματικός ή απλά κάποια μηχανή που εκτελεί προγραμματισμένες αυτόματες ενέργειες. Αν τελικά ο χρήστης δεν απαντήσει στο μήνυμα ALARM1 μετά από το πέρας ενός διαστήματος θα

σημαίνει ότι είναι μία μηχανή οπότε στο στάδιο αυτό της προειδοποίησης δεν θα του στέλνεται κάποιο άλλο προειδοποιητικό μήνυμα.

Ο μη απομονωμένος χρήστης δεν θα δέχεται ο ίδιος κάποιο μήνυμα ALARM αλλά θα το δέχονται όλοι οι φίλοι του προκειμένου να τον βαθμολογήσουν με βάση κάποια κριτήρια. Ένα ίδιο μήνυμα ALARM2 λαμβάνουν και οι φίλοι ενός απομονωμένου χρήστη που μετά το ALARM1 θεωρείται πραγματικός και όχι μηχανή.

Με τα αποτελέσματα των απαντήσεων των μηνυμάτων ALARM2 φτιάχνεται κάθε φορά μία κλίμακα για το χρήστη που έχει θεωρηθεί ύποπτος για μη ομαλή συμπεριφορά. Την κλίμακα αυτή την ονομάζουμε κλίμακα εμπιστοσύνης και δείχνει πόσο έμπιστος ή μη έμπιστος θεωρείται ένας χρήστης από ένα διαδικτυακό του φίλο στο κοινωνικό δίκτυο.
Εμάς μας ενδιαφέρει στη συγκεκριμένη μοντελοποίηση να εντοπιστούν οι μη έμπιστοι χρήστες οπότε ενδεικτικά κάποια κριτήρια που περιέχονται σε ένα μήνυμα ALARM2 και οι βαθμοί, ακέραιοι αριθμοί, για τη δημιουργία της κλίμακας εμπιστοσύνης μπορεί είναι τα εξής:

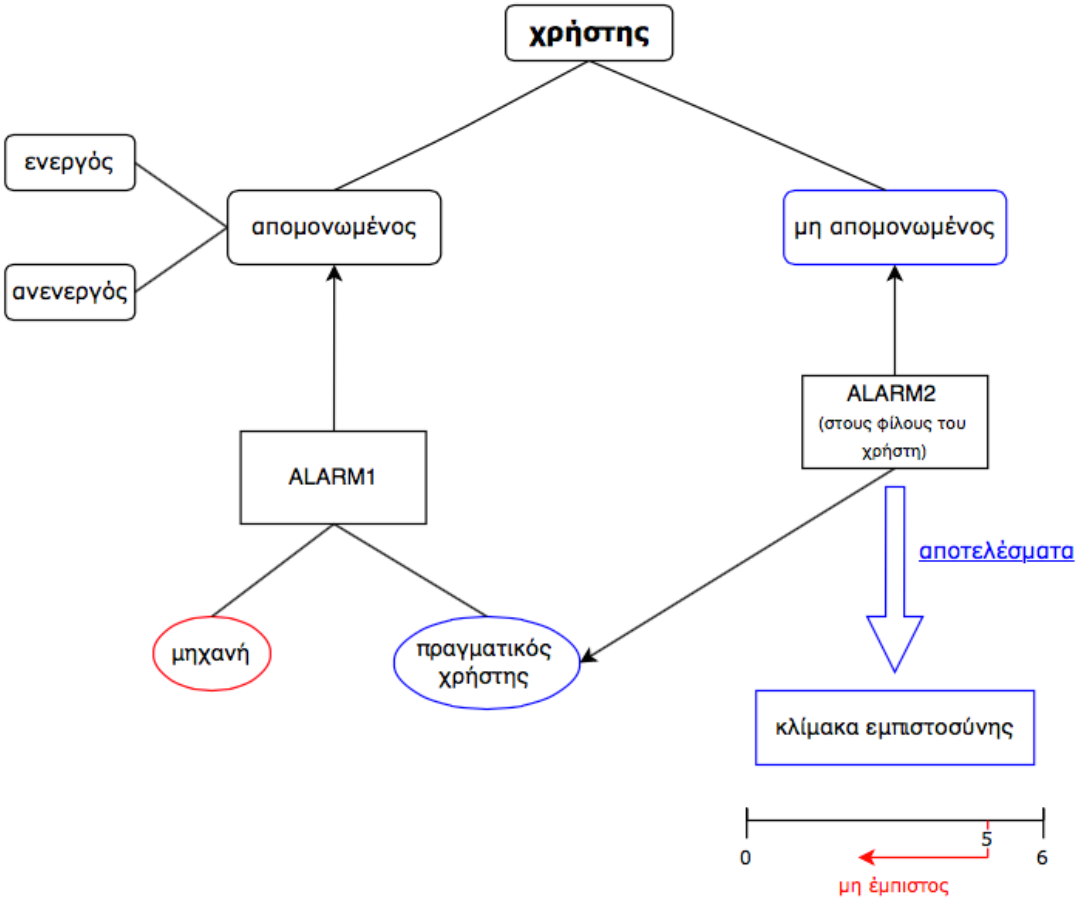
Κριτήρια	*Πιθανές απαντήσεις / βαθμοί για την κλίμακα εμπιστοσύνης*	
Αν γνωρίζουν το χρήστη μόνο διαδικτυακά	Ναι / 1	Όχι / 2
Αν η συμπεριφορά του χρήστη είναι με οποιοδήποτε τρόπο προσβλητική	Ναι / 1	Όχι / 2
Αν έχει παρατηρηθεί ότι δημοσιεύει μη αληθείς πληροφορίες	Ναι / 1	Όχι / 2



Άλλα κριτήρια μπορούν τόσο να προστεθούν όσο και να αφαιρεθούν για τη δημιουργία μίας κλίμακας εμπιστοσύνης προσαρμοσμένες στις ανάγκες κάθε δικτύου που επιθυμούμε να μοντελοποιήσουμε και πάντα γνωρίζοντας τις προτιμήσεις των χρηστών του.

Στο παράδειγμα της κλίμακας εμπιστοσύνης που μόλις δημιουργήσαμε φαίνεται πως ένας χρήστης θεωρείται έμπιστος αν έχει τιμή μεγαλύτερη του 5, δηλαδή αν η συνολική του βαθμολογία είναι 6. Αυτό συμβαίνει αν στο μήνυμα ALARM2 έχουν απαντηθεί 3 "Όχι" που το καθένα έχει 2 βαθμούς. Σε αντίθετη περίπτωση, αν η βαθμολογία είναι από 5, δύο "Όχι" και ένα "Ναι", ή κάτω από 5, όλα "Ναι", τότε ο χρήστης είναι μη έμπιστος και υπάρχει μία ένδειξη ότι η συμπεριφορά του είναι κακόβουλη. Φυσικά, η τιμή μίας μόνο κλίμακας εμπιστοσύνης δεν είναι αρκετή για να θεωρηθεί ο χρήστης κακόβουλος αλλά χρειάζεται ένα ακόμα βήμα ώστε να συνυπολογιστούν όλες οι βαθμολογίες από κάθε φίλο του χρήστη.

Μία ακόμα παρατήρηση είναι ότι δεν εξετάζεται μόνο η εμπιστοσύνη (trust) που δείχνουν οι φίλοι προς τον ίδιο το χρήστη, αλλά μπορεί να εξεταστεί και η δυσπιστία (distrust) που τους έχει δείξει εκείνος μέσα από την δραστηριότητα του στο κοινωνικό δίκτυο. Έτσι, φαίνεται πόσο κακόβουλος θεωρείται από κάθε φίλο του ο χρήστης εκείνος ο οποίος στο στάδιο της προειδοποίησης θεωρήθηκε από το δίκτυο ύποπτος για κακόβουλη συμπεριφορά.



Εικόνα 7: Σχεδιαγραμματικά τα βήματα 1 και 2 του μοντέλου

3.1.3 Φήμη

Το στοιχείο Reputation System στα ad hoc δίκτυα είναι υπεύθυνο για τη βαθμολογία των κόμβων και τη δημιουργία της καλής ή κακής τους φήμης ώστε να υπάρξει η ανάλογη δρομολόγηση και προώθηση πακέτων. Στο πρωτόκολλο CONFIDANT, το Reputation System διαχειρίζεται ένα πίνακα που περιέχει εγγραφές με τους κόμβους και τις βαθμολογίες τους από τις αναφορές που δίνει ο ίδιος κόμβος, στις παρατηρήσεις του κόμβου για τους γειτονικούς τους και στην εμπειρία που έχουν δώσει οι άλλοι κόμβοι γι' αυτόν. Η βαθμολογία κάθε κόμβου αλλάζει μόνο όταν υπάρχει επαρκής απόδειξη κακόβουλης συμπεριφοράς. Εάν η βαθμολογία αυτή πέσει κάτω από ένα επίπεδο τότε καλείται το στοιχείο Path Manager.

Παρόμοια διαδικασία θα χρησιμοποιηθεί και στα κοινωνικά δίκτυα για τη φήμη που θα έχουν οι χρήστες, όχι όλοι, αλλά μόνο αυτοί που έχουν θεωρηθεί ύποπτοι για κακόβουλη συμπεριφορά και δεν είναι μηχανές. Με άλλα λόγια, θα υπολογιστεί η κακή φήμη που έχουν αυτοί οι χρήστες. Έτσι, μία σημαντική διαφορά σε σχέση με τα ad-hoc δίκτυα είναι ότι η κακή φήμη θα είναι αποκλειστικά υπολογισμένη από τη γνώμη που έχουν σχηματίσει οι φίλοι του χρήστη. Η κακή φήμη υπολογίζεται από την ακέραιη τιμή της μέσης τιμής της βαθμολογίας της κλίμακας εμπιστοσύνης των n, σε πλήθος, φίλων του χρήστη.

Με βάση το προηγούμενο παράδειγμα αν η τιμή της φήμης είναι κάτω από 5 ο χρήστης ορίζεται κακόβουλος για το δίκτυο.

$$[\varphi\eta\mu\eta] = \frac{\sum_{i=1}^{n}\{\beta\alpha\theta\mu o\lambda o\gamma\iota\alpha_i\}}{n}$$

3.1.4 Τελική αναπροσαρμογή

Το Path Manager χρειάζεται στα ad hoc δίκτυα για τη δρομολόγηση και τις αλλαγές στην διαδρομή της δρομολόγησης κάθε πακέτου που στέλνει ένας δρομολογητής και καλείται από το Reputation System, δηλαδή όταν η βαθμολογία ενός κόμβου μειωθεί σε σημαντικό βαθμό. Έτσι, ένα πακέτο θα μεταφερθεί από τους κόμβους που έχουν συγκεντρώσει την υψηλότερη φήμη, ενώ τα μονοπάτια που έχουν κακόβουλους κόμβους θα διαγράφονται.

Γενικά, μονοπάτι από έναν κόμβο σε έναν άλλο ενός γράφου ονομάζεται μία ακολουθία κόμβων, όπου κάθε κόμβος της ακολουθίας συνδέεται με τον επόμενό του μέσω μίας ακμής. Τα κοινωνικά δίκτυα, όπως έχουμε αναφέρει και προηγουμένως, αν τα αναπαραστήσουμε ως γραφήματα τότε κόμβοι είναι οι χρήστες του δικτύου και ακμές οι σχέσεις που τους συνδέουν. Έτσι, γίνεται κατανοητό πως τα μονοπάτια έχουν νόημα σε

πολλές εφαρμογές ανάλυσης των κοινωνικών δικτύων, όμως σε αυτό το κομμάτι εμείς χρησιμοποιούμε μόνο σαν μονοπάτι μία σχέση που συνδέει δύο χρήστες.

Στα κοινωνικά δίκτυα αυτό το κομμάτι του πρωτόκολλου είναι σημαντικό καθώς θα υπάρχει μία τελική αναπροσαρμογή του δικτύου ώστε το κοινωνικό δίκτυο να είναι τελικά ένα δίκτυο εμπιστοσύνης (trust network) απαλλαγμένο από κάθε χρήστη που δεν βοηθάει στην ομαλή, έγκυρη και αξιόπιστη λειτουργία του. Ανάλογα με τη φήμη που έχει ένας χρήστης που έχει θεωρηθεί πλέον από το προηγούμενο βήμα της φήμης κακόβουλος, θα μπορούν να ακολουθηθούν δύο διαδικασίες:

1. αν η φήμη έχει την οριακή τιμή 5 να διαγράφεται το "μονοπάτι" του κακόβουλου χρήστη με όποιον χρήστη φίλο του επιθυμεί εφόσον του έχει δώσει αρνητική βαθμολογία

2. αν η φήμη έχει τιμή κάτω από 5 να διαγράφεται ο ίδιος ο κακόβουλος χρήστης επειδή ένα σημαντικό ποσοστό των φίλων του τον βαθμολογούν αρνητικά.

Η πρώτη περίπτωση σημαίνει ότι ο φίλος του χρήστη δίνοντας αυτή τη βαθμολογία δεν θέλει για φίλο του τον κακόβουλο οπότε και διαγράφεται η σχέση που τους ενώνει, όμως ο χρήστης παραμένει στο δίκτυο γιατί το σύνολο των φίλων του δεν συμμερίζονται αυτή την άποψη.

Στη δεύτερη περίπτωση ο χρήστης διαγράφεται από το κοινωνικό δίκτυο και σε αυτή τη διαδικασία συμπεριλαμβάνεται και ο απομονωμένος χρήστης που έχει θεωρηθεί μηχανή μετά από το μήνυμα ALARM1 στο στάδιο της προειδοποίησης.

3.2 Πλεονεκτήματα της μοντελοποίησης

Καταρχάς, ως προς την πολυπλοκότητα για την εύρεση ενός κακόβουλου χρήστη μπορούμε να πούμε ότι επειδή δεν ελέγχεται κάθε ένας χρήστης του δικτύου, αλλά μόνο ορισμένοι βάση κάποιων πολύ συγκεκριμένων κριτηρίων, σίγουρα είναι πιο γρήγορη χρονικά σε σχέση με οποιαδήποτε μέθοδο που καλείται να ελέγξει το σύνολο των χρηστών ενός κοινωνικού δικτύου. Η μοντελοποίηση αυτή για τα κοινωνικά δίκτυα σε δεδομένες καταστάσεις, δηλαδή ελέγχοντας ορισμένα κριτήρια και απομονώνοντας έναν αριθμό χρήστων σε σχέση με τους εκατομμύρια συνδεδεμένους λογαριασμούς, είναι απαραίτητη προκειμένου να υπάρχουν άμεσα μετρήσιμα αποτελέσματα.

Ακόμα, σημαντικό είναι ότι το συγκεκριμένο δίκτυο εμπιστοσύνης που τελικά δημιουργείται είναι διαφορετικό σε σχέση με άλλα που βασίζονται στην εμπιστοσύνη μεταξύ των χρηστών, καθώς αυτό βασίζεται στην πιθανή δυσπιστία τους. Αυτό όμως δεν γίνεται από έναν αλγόριθμο που χρησιμοποιεί το δίκτυο χωρίς την συμμετοχή των χρηστών του. Αντίθετα, οι χρήστες έχουν ενεργό ρόλο στην υλοποίηση αυτή της μοντελοποίησης και η συμμετοχή τους είναι αναγκαία. Οι ενέργειες τους μέσα στο κοινωνικό δίκτυο, όπως είναι γνωστό, καταγράφονται μέσω συνεχούς παρακολούθησης, αλλά δεν είναι αυτές που κρίνουν το αποτέλεσμα της παραπάνω διαδικασίας. Σημασία έχει το πως κάθε χρήστης επηρεάζει τις επαφές του και τη γνώμη έχουν αυτές για τον ίδιο. Συνοψίζοντας επισημαίνουμε ότι όπως και στα ad hoc δίκτυα παρατηρήθηκε ένα είδος συνεργασίας των κόμβων, έτσι και εδώ οι χρήστες συνεργάζονται, μέσω όπως της καθοδήγησης της κεντρικής μονάδας του κοινωνικού δικτύου, προκειμένου να δημιουργηθεί το δίκτυο εμπιστοσύνης. Η συμμετοχή τους στη μοντελοποίηση είναι καθοριστική και τους δίνει την ελευθερία να ορίζουν ως ένα σημείο το πλαίσιο εκείνο που οι ίδιοι επιθυμούν ή αντιλαμβάνονται την εμπιστοσύνη μεταξύ τους.

Όπως και κάθε σχέση, έτσι και μία διαδικτυακή σχέση δεν είναι αναλλοίωτη με το χρόνο, αλλά συνεχώς μεταβάλλεται και εξελίσσεται. Για το λόγο αυτό, μπορεί να υπάρξει αναβάθμιση του τελικού δικτύου ανά πάσα στιγμή χρησιμοποιώντας πάλι το μοντέλο που παρουσιάσαμε και επεμβαίνοντας εμείς σε αλλαγές όποτε αυτό κρίνεται σκόπιμο.

Παρατηρούμε τέλος ότι η μοντελοποίηση που παρουσιάστηκε δεν είναι προσαρμοσμένη σε κανένα συγκεκριμένο κοινωνικό δίκτυο, αλλά υπάρχει η δυνατότητα να χρησιμοποιηθεί και σε ένα μη διαδικτυακό περιβάλλον όπως σε μία επιχείρηση για το δίκτυο των υπαλλήλων της για την εύρεση παραδείγματος χάριν των μη αποδοτικά εργαζομένων της.

Κεφάλαιο 4

4.1 Τυπικές Μέθοδοι

Οι τυπικές μέθοδοι (formal methods) [39] είναι υπολογιστικές τεχνικές, ή αλλιώς γλώσσες, που βασίζονται στα μαθηματικά και τη λογική, και χρησιμοποιούνται για την προδιαγραφή, ανάπτυξη και επαλήθευση πολύπλοκων συστημάτων υλικού και λογισμικού. Η χρήση των τυπικών μεθόδων είναι δημοφιλής σε πολύπλοκα συστήματα, κυρίως σε συστήματα που αφορούν την ασφάλεια, για την αποφυγή λαθών καθώς έχουν διαδικασίες επαλήθευσης και ελέγχου.

Τα δύο στάδια που χρησιμοποιούνται για τη μοντελοποίηση τυπικών μεθόδων είναι τα εξής:

(1) καταγραφή των τυπικών προδιαγραφών (formal specification)

(2) τυπική επαλήθευση (formal verification)

Με την έννοια καταγραφή εννοούμε τη διαδικασία περιγραφής του συστήματος και των ιδιοτήτων που το καθορίζουν, ενώ με την έννοια επαλήθευση εννοούμε την απόδειξη της ύπαρξης των επιθυμητών ιδιοτήτων του συστήματος που περιγράψαμε.

Θα αναφέρουμε εν συντομία κάποιες κύριες τυπικές μεθόδους.

- Z

Η γλώσσα Z [40], γνωστή και ως Z notation, είναι μία τυπική γλώσσα προδιαγραφών για την περιγραφή συστημάτων υπολογισμού. Με τη χρήση μαθηματικών τύπων δεδομένων γίνεται η μοντελοποίηση τους σε ένα σύστημα. Στόχος είναι η περιγραφή του αποτελέσματος κάθε λειτουργίας του συστήματος μέσω κατηγορηματικής λογικής (predicate logic).

- SDL

Η γλώσσα SDL (Specification and Description Language) [41] ορίστηκε από τη Διεθνή Ένωση Τηλεπικοινωνιών (International Telegraph Union-Telecommunication Standardization Sector, ITU-T) και χρησιμοποιείται για την μοντελοποίηση τηλεπικοινωνιακών συστημάτων. Το βασικό μοντέλο ενός συστήματος σε SDL αποτελείται από μία επέκταση των μηχανών καταστάσεων με σκοπό την επικοινωνία.

- Αλγεβρικές γλώσσες προδιαγραφών

Οι αλγεβρικές γλώσσες προδιαγραφών (Algebraic Specification Languages) είναι τυπικές μέθοδοι που χρησιμοποιούν μαθηματική λογική για το σχεδιασμό λογισμικού και συστημάτων αξιόπιστων. Η προδιαγραφή των συστημάτων γίνεται με στόχο την περιγραφή ορισμένων ιδιοτήτων του συστήματος και στη συνέχεια τη δυνατότητα επαλήθευσης και επικύρωσης της. Αποτελούν την πια αναπτυγμένη προσέγγιση των τυπικών μεθόδων και ιδιαίτερα τα τελευταία χρόνια αναπτύσσεται ακόμα περισσότερο η έρευνα τόσο στο θεωρητικό όσο και στο πρακτικό τους κομμάτι.

4.2 Εισαγωγή στη αλγεβρική γλώσσα προδιαγραφών CafeOBJ

Η CafeOBJ [42] είναι μια εκτελέσιμη αλγεβρική γλώσσα προδιαγραφών, η οποία είναι ο μοντέρνος διάδοχος της γλώσσας OBJ, ενσωματώνοντας όμως αρκετά νέα χαρακτηριστικά. Βασίζεται στο συνδυασμό πολλαπλών προτύπων λογικής όπως άλγεβρα διατεταγμένων τύπων, άλγεβρα με κρυμμένους τύπους και τη λογική της αναγραφής. Αυτός ο συνδυασμός, όπως θα δούμε στον κύβο της CafeOBJ, βασίζεται στη θεωρία των θεσμών (theory of institutions). Η CafeOBJ επειδή είναι εκτελέσιμη γλώσσα προορίζεται για να χρησιμοποιείται κυρίως για προδιαγραφές συστημάτων, τυπική επαλήθευση των προδιαγραφών, γρήγορη κατασκευή πρωτοτύπων, απόδειξη θεωρημάτων, ακόμη και στον προγραμματισμό, κλπ.

Η CafeOBJ έχει ένα σύστημα τμημάτων προδιαγραφής (module system), όπως: πολλά είδη τέτοιων τμημάτων που μπορούν να εισαχθούν, παραμετροποιήσιμα τμήματα, και επιπλέον για κάθε τέτοιο τμήμα μπορεί κανείς να επιλέξει μεταξύ χαλαρής (loose) και σφιχτής (tight) αρχικής σημασιολογίας.

Το λογικό υπόβαθρο της CafeOBJ περιλαμβάνει [43]:

- Τη λογική με διατεταγμένους τύπους (order-sorted logic)

Ένας τύπος μπορεί να είναι υποσύνολο ενός άλλου τύπου. Για παράδειγμα, οι φυσικοί αριθμοί μπορούν να θεωρηθούν ως υποσύνολο των κλασματικών αριθμών. Αυτή η ενσωμάτωση κάνει έγκυρο τον ισχυρισμό ότι το 3 ισούται με 6/2. Επίσης, γίνεται εφικτή η κληρονομικότητα των τελεστών, με την έννοια ότι ένας τελεστής αν έχει δηλωθεί ως κλασματικός αριθμός αυτόματα δηλώνεται και ως φυσικός αριθμός. Επιπλέον, η σχέση υποτύπου δίνει τη δυνατότητα να οριστούν με απλό τρόπο οι μερικοί τελεστές (partial operations) και ο χειρισμός των εξαιρέσεων (exception handing).

- Τη λογική της αναγραφής (rewriting logic)

Εκτός από τις εξισώσεις (ισότητες), οι οποίες υπακούουν στο νόμο της συμμετρίας, είναι δυνατή η χρήση σχέσεων μετάβασης που είναι μόνο προς μια κατεύθυνση. Οι σχέσεις μεταξύ καταστάσεων τυποποιούνται με φυσικό τρόπο μέσω των σχέσεων μετάβασης οι οποίες είναι χρήσιμες και στην αναπαράσταση του ταυτοχρονισμού και της ασάφειας (indeterminacy).

- Τους κρυμμένους τύπους (hidden sorts)

Υπάρχουν δύο είδη ισοδυναμίας, η μία, η πρώτη εξισώνει τα στοιχεία αν και μόνο αν είναι τα ίδια με βάση τη δεδομένη εξισωτική θεωρία και η δεύτερη, η συμπεριφοριακή χρησιμοποιείται για τους κρυμμένους τύπους και βάσει της οποίας δύο όροι είναι

ισοδύναμοι, αν και μόνο αν συμπεριφέρονται με ίδιο τρόπο με βάση το ίδιο σύνολο των παρατηρήσεων.

4.2.1 Σημαντικά χαρακτηριστικά

Το λογικό υπόβαθρο της CafeOBJ που περιγράψαμε παραπάνω της δίνει τα εξής χαρακτηριστικά:

- Εξισωτική προδιαγραφή και προγραμματισμός (Equational Specification and Programming)

Αυτό το χαρακτηριστικό κληρονομείται από την OBJ και αποτελεί τη βάση της γλώσσας, με τα υπόλοιπα χαρακτηριστικά της να "χτίζονται" πάνω σε αυτή. Όπως και η OBJ, η CafeOBJ είναι εκτελέσιμη (μέσω της αναγραφής των όρων), το οποίο της προσδίδει ένα δηλωτικό τρόπο συναρτησιακού προγραμματισμού, τον αλγεβρικό προγραμματισμό (algebraic programming). Όπως και η OBJ, η CafeOBJ επίσης επιτρέπει τη δήλωση ιδιοτήτων στις εξισώσεις όπως τη μεταθετικότητα, προσεταιριστικότητα κ.α.

- Η συμπεριφοριακή προδιαγραφή (Behavioural Specification)

Η συμπεριφοριακή προδιαγραφή παρέχει ακόμα μια πρωτότυπη γενίκευση της παραδοσιακής αλγεβρικής προδιαγραφής. Μια συμπεριφοριακή προδιαγραφή χαρακτηρίζει πως τα αντικείμενα (και τα συστήματα) συμπεριφέρονται, όχι το πως υλοποιούνται. Ο νέος αυτός τρόπος αφαίρεσης μπορεί να είναι πολύ ισχυρός στην προδιαγραφή και την επαλήθευση συστημάτων καθώς περιλαμβάνει άλλες χρήσιμες προσεγγίσεις όπως ο ταυτοχρονισμός (concurrency), ο αντικειμενοστραφής προσανατολισμός, ο μη ντετερμινισμός (nondeterminism), κ.α. Η συμπεριφοριακή αφαίρεση πραγματοποιείται χρησιμοποιώντας προδιαγραφές με κρυμμένους τύπους και με μια συμπεριφοριακή έννοια της σχέσης ικανοποιησιμότητας που βασίζεται στην ιδέα της διάκρισης των καταστάσεων που είναι οι ίδιες όταν παρατηρηθούν, το οποίο γενικεύει επίσης την process algebra και τα συστήματα μετάβασης. Η CafeOBJ υποστηρίζει άμεσα την συμπεριφοριακή προδιαγραφή και την θεωρία αποδείξεων της, μέσω ειδικών κατασκευών της γλώσσας, όπως:

1) κρυμμένους τύπους για τις καταστάσεις του συστήματος,
2) συμπεριφοριακούς τελεστές (για τις "δράσεις" και τις "παρατηρήσεις" σε καταστάσεις των συστημάτων),
3) δηλώσεις συμπεριφοριακής συνεκτικότητας για μη-συμπεριφοριακούς τελεστές, οι οποίοι είναι δυνατό να είναι είτε έμμεσες "παρατηρήσεις" ή "κατασκευαστές" σε καταστάσεις του συστήματος, και
4) συμπεριφοριακά αξιώματα (δηλώνει ικανοποίηση συμπεριφοράς)

Η κύρια αποδεικτική μέθοδος επαγωγής για συμπεριφοριακές ιδιότητες ονομάζεται coinduction. Στην CafeOBJ, η τεχνική coinduction μπορεί να χρησιμοποιηθεί είτε με την κλασσική άλγεβρα με κρυμμένους τύπους για την απόδειξη ισοδυναμίας στη συμπεριφορά των καταστάσεων των αντικειμένων, ή για την απόδειξη αλλαγής των καταστάσεων που εμφανίζονται κατά την εφαρμογή συμπεριφορικής αφαίρεσης στη λογική της αναγραφής.
Εκτός από την κατασκευή της γλώσσας, η CafeOBJ υποστηρίζει την προδιαγραφή (specification) και την επαλήθευση (verification) μέσω διαφόρων μεθοδολογιών. Η CafeOBJ υποστηρίζει μια μεθοδολογία για ταυτόχρονη σύνθεση αντικειμένων το οποίο είναι χρήσιμο καθώς υπάρχει η δυνατότητα επαναχρησιμοποίησης όχι μόνο του κώδικα που είναι γραμμένη η προδιαγραφή αλλά και των επαληθεύσεων των συστημάτων. Μια συμπεριφοριακή προδιαγραφή στην CafeOBJ μπορεί επίσης να χρησιμοποιηθεί αποτελεσματικά ως μια εναλλακτική σε αντικείμενα ή καταστάσεις αντί για τις κλασσικές προδιαγραφές για τύπους δεδομένων. Από πειράματα φαίνεται ότι ένα αντικειμενοστραφές στυλ προδιαγραφών, ακόμα και των βασικών τύπων δεδομένων (όπως σύνολα, λίστες, κ.λπ.), μπορεί να οδηγήσει σε μεγαλύτερη απλότητα του κώδικα και δραστική απλούστευση της διαδικασίας της επαλήθευσης [44].

Η συμπεριφοριακή προδιαγραφή εντοπίζεται στο επίπεδο εκτέλεσης της προδιαγραφής από την έννοια της συμπεριφοριακής αναγραφής, η οποία ορίζει τη συνήθη αναγραφή με μια συνθήκη που επιβεβαιώνει την ορθότητα της χρήσης των συμπεριφοριακών εξισώσεων για τις αποδείξεις των αυστηρών ισοτήτων.

- Η προδιαγραφή με τη λογική της αναγραφής (Rewriting Logic Specification)

Η προδιαγραφή με τη λογική της αναγραφής (για συντομογραφία και 'RWL') στην CafeOBJ βασίζεται στην απλοποιημένη έκδοση του Meseguer [45] για ταυτόχρονα συστήματα, η οποία δίνει μια επέκταση της παραδοσιακής αλγεβρικής προδιαγραφής στην κατεύθυνση του ταυτοχρονισμού. Η λογική της αναγραφής ενσωματώνει πολλά διαφορετικά μοντέλα ταυτοχρονισμού με ένα φυσικό, απλό και ελκυστικό τρόπο δίνοντας στην CafeOBJ ένα μεγάλο εύρος εφαρμογών.
Από μεθοδολογικής άποψης, η CafeOBJ αναπτύσσει τη χρήση της λογικής των αναγραφών μέσω των μεταβάσεων για προδιαγραφή και επαλήθευση των ιδιοτήτων της δηλωτικής κωδικοποίησης των αλγορίθμων όπως επίσης και για την προδιαγραφή και την επαλήθευση συστημάτων μετάβασης.

- Σύστημα τμημάτων προδιαγραφών (Module system)

Οι αρχές του module συστήματος της CafeOBJ έχουν "κληρονομηθεί" από την γλώσσα OBJ, η οποία έχει χτιστεί σε ιδέες που πρωτοεμφανίστηκαν στη γλώσσα Clear [46] και τα στοιχεία του είναι τα εξής:

- διάφορα είδη τμημάτων
- διαμοιρασμός για πολλαπλά τμήματα
- παραμετροποιημένος προγραμματισμός ο οποίος επιτρέπει:

  - πολλαπλές παραμέτρους
  - οπτικές για παραμέτρους που ορίζονται άμεσα
  - ενοποίηση των προδιαγραφών της CafeOBJ, με εκτελέσιμο κώδικα σε χαμηλότερου επιπέδου γλώσσα
- εκφράσεις των τμημάτων των προδιαγραφών

Ωστόσο, ο συγκεκριμένος σχεδιασμός της γλώσσας αναθεωρεί την τρόπο που γινόταν η εισαγωγή λειτουργιών και παραμέτρων στην OBJ.

- Σύστημα τύπων (Type System)

Η CafeOBJ έχει ένα σύστημα τύπων το οποίο επιτρέπει τη χρήση υποτύπων με βάση την άλγεβρα με διατεταγμένους τύπους (order sorted algebra-‘OSA’). Αυτό παρέχει ένα ισχυρά μαθηματικό τρόπο για έλεγχο τύπων κατά τη διάρκεια της εκτέλεσης και διαχείρισης λαθών, δίνοντας στην CafeOBJ μια μεγαλύτερη συντακτική ευελιξία σε σχέση με γλώσσες που δεν υποστηρίζουν τύπους, ενώ παράλληλα διατηρεί όλα τα πλεονεκτήματα μια γλώσσας ισχυρά τυποποιήσιμης. Στην CafeOBJ δεν γίνονται απευθείας οι μερικές λειτουργίες αλλά τις διαχειρίζονται χρησιμοποιώντας τελεστές σφαλμάτων και τελεστές με εξισωτική λογική συμμετοχής (membership equational logic - ‘MEL’).

4.2.2 Λογικό υπόβαθρο

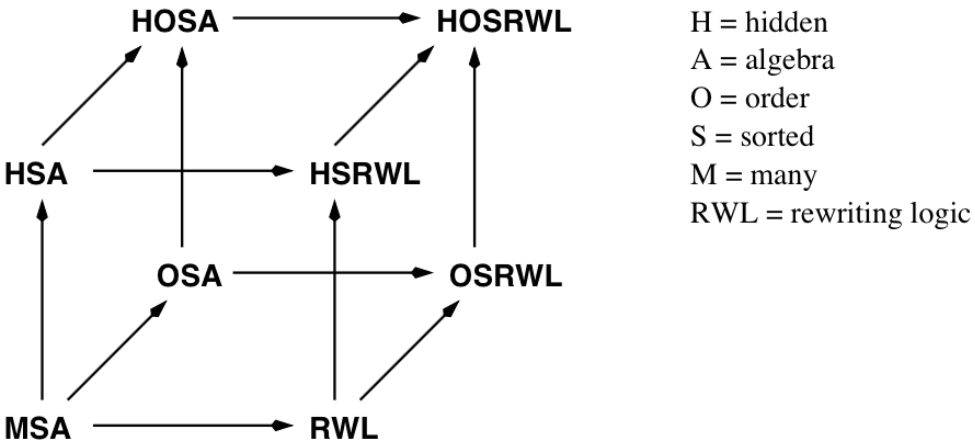
Η CafeOBJ [42] είναι μια αλγεβρική γλώσσα με αυστηρό μαθηματικό και λογικό υπόβαθρο, όπως είναι και οι υπόλοιπες γλώσσες της οικογένειας OBJ (OBJ, Eqlog, FOOPS, Maude). Η μαθηματική σημασιολογία της CafeOBJ βασίζεται στην αλγεβρική προδιαγραφή εννοιών και αποτελεσμάτων, και η οποία βασίζεται έντονα στην θεωρία κατηγοριών και τη θεωρία των θεσμών. Οι παρακάτω είναι οι αρχές που διέπουν τα λογικά και τα μαθηματικά θεμέλια της CafeOBJ:

- υπάρχει μια υποκείμενη λογική στην οποία όλες οι βασικές δομές και λειτουργίες της γλώσσας μπορεί να εξηγηθούν αυστηρά

- παρέχει μια ολοκληρωμένη, συνεκτική και ενιαία προσέγγιση για την σημασιολογία της προδιαγραφής

- αναπτύσσει όλα τα συστατικά (έννοιες, τα αποτελέσματα, κλπ.) σε υψηλό επίπεδο

Ο συνδυασμός λογικών συστημάτων στα οποία βασίζεται η CafeOBJ την κάνουν μια πολύ-λογική γλώσσα (multi-logic language) και η σχέση τους φαίνεται στο παρακάτω σχήμα, όπου απεικονίζεται ο κύβος της CafeOBJ [51]. Στον κύβο αυτό, οι κόμβοι αντιστοιχούν σε λογικές ή θεσμούς και τα βέλη δείχνουν τις σχέσεις ενσωμάτωσης μεταξύ των λογικών αυτών συστημάτων, που αντιστοιχούν σε διαφορετικούς συνδυασμούς θεσμών, με μετάβαση από τα λιγότερο στα περισσότερο πολύπλοκα συστήματα λογικής.



Εικόνα 8: Κύβος της CafeOBJ

Τα βέλη του CafeOBJ κύβου αντιστοιχούν σε συνδυασμούς θεσμών, δηλαδή M για πολλούς, S για διατεταγμένους, A για άλγεβρα, O για σειρά, H για κρυφό και RWL για τη λογική της αναγραφής.

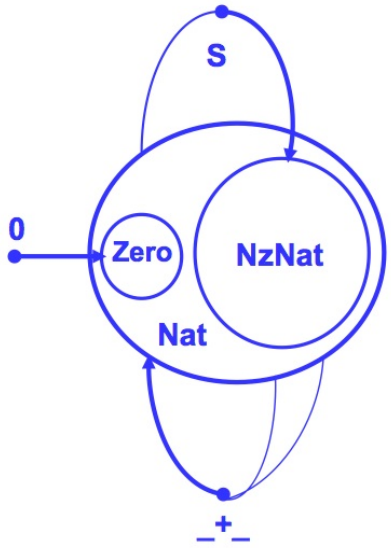
4.2.3 Βασικό συντακτικό

Οι βασικές μονάδες των προδιαγραφών της CafeOBJ είναι τα τμήματα (modules). Ένα τμήμα δηλώνεται ως εξής:

```
mod module_name {
module_element*
}

module_element = sort_declaration | operator_declaration |
variable_declaration | equation declaration
```

Το όνομα του τμήματος είναι μια αυθαίρετη συμβολοσειρά. Ένα στοιχείο ενός τμήματος μπορεί να είναι (1) δήλωση τύπου, (2) δήλωση τελεστή είτε (2α) δράσης είτε (2β) παρατήρησης, (3) δήλωση μεταβλητή ή (4) δήλωση εξίσωσης. Ένας γενικός κανόνας για τα τμήματα είναι ότι κάθε στοιχείο ενός τμήματος δεν μπορεί να χρησιμοποιηθεί εκτός αν πρώτα έχει δηλωθεί κάπου στον κώδικα. Ακόμα, κάθε στοιχείο ανήκει στο τμήμα που εμφανίζεται και δεν μπορεί να χρησιμοποιηθεί εκτός αυτού αν πριν δεν έχει εισαχθεί στο τμήμα που θέλουμε να χρησιμοποιηθεί.
Ένα παράδειγμα προδιαγραφής στην CafeOBJ για τον αφηρημένο τύπο δεδομένων Nat των φυσικών αριθμών είναι το εξής:

```
module SIMPLE-NAT {

-- declaration of sorts
[Zero NzNat < Nat]

-- declaration of operators
op 0 : -> Zero
op s : Nat -> NzNat
op _+_: Nat Nat -> Nat

-- declaration of variables
vars N N': Nat

-- declaration of equations
eq 0 + N = N .
eq s(N)+ N' = s(N+N').

}
```



Το όνομα του τμήματος δηλώνεται μετά την λέξη module (ή mod για συντομογραφία) και είναι `SIMPLE-NAT`. Οι τύποι, που το όνομά τους είναι επίσης μια αυθαίρετη ακολουθία χαρακτήρων, δηλώνονται ανάμεσα στα [ ], η διάταξή τους με το < και μια λίστα από τέτοιου ονόματα αποτελεί λίστα των αντίστοιχων τύπων που χωρίζονται από κενά. Εδώ δηλώνονται οι τύποι `Zero`, `NzNat` και `Nat` που συμβολίζουν το μηδέν, τους μη μηδενικούς φυσικούς αριθμούς και τους φυσικούς αριθμούς αντίστοιχα και οι τύποι `Zero` και `NzNat` είναι υποτύποι του `Nat`. Γενικά, υπάρχουν δύο είδη τύπων στην CafeOBJ. Οι

ορατοί (visible sorts) και οι κρυμμένοι τύποι (hidden sorts) και δηλώνονται με τον εξής τρόπο:

```
sort_declaration = hidden_sort | visible_sort
visible_sort = [list_of_sort_names{< list_of_sort_names}*]
hidden_sort = "*"[list_of_sort_names{< list_of_sort_names}*]"*"
```

Οι ορατοί τύποι που αναπαριστούν τους αφηρημένους τύπους δεδομένων της προδιαγραφής ενώ οι κρυμμένοι αναπαριστούν τις καταστάσεις των αντικειμένων ή του συστήματος γενικότερα.

Μετά το σύμβολο -- ότι ακολουθεί στην ίδια γραμμή είναι σχόλια και δεν λαμβάνονται υπόψη όταν γίνεται η μεταγλώτισση του κώδικα στην CafeOBJ.

Κάθε τελεστής, που έχει το ρόλο μιας συνάρτησης καθώς περιγράφει τον τρόπο με τον οποίο μετατρέπεται ένα στοιχείο ενός τύπου (ή των στοιχείων διαφόρων τύπων) σε έναν άλλο τύπο, δηλώνεται με τη λέξη `op` (operator). Στην περίπτωση που θέλουμε να δηλώσουμε παραπάνω από έναν τελεστή ίδιου τύπου χρησιμοποιούμε την λέξη `ops`. Οι τελεστές με όρισμα ακριβώς έναν κρυμμένο τύπο ονομάζονται συμπεριφεριακοί τελεστές, διακρίνονται σε τελεστές δράσης (transition) ή παρατήρησης (observation) και δηλώνονται με τη λέξη `bop` (behavioural operator). Μια δράση μπορεί να αλλάξει την κατάσταση του αντικειμένου ή του συστήματος, δηλαδή σαν όρισμα είναι η κατάσταση αυτή που αλλάζει και ίσως και κάποια άλλα δεδομένα και επιστρέφεται η ίδια ή μια νέα κατάσταση. Για την παρατήρηση μιας τιμής της κατάστασης του συστήματος χρησιμοποιούνται οι τελεστές παρατήρησης. Με άλλα λόγια, σαν ορίσματα εισάγονται η κατάσταση του συστήματος (ένας μόνο κρυφός τύπος) και μπορεί και κάποια δεδομένα και επιστρέφεται η παρατηρήσιμη τιμή στην κατάσταση αυτή, δηλαδή έναν ορατό τύπο.

Στο παράδειγμά μας έχουμε κάνει χρήση τριών τελεστών:

- 0 είναι μια σταθερά και συγκεκριμένα το ουδέτερο στοιχείο της πρόσθεσης, δηλαδή το μηδέν, όπως ορίζεται και στην πρώτη εξίσωση
- s είναι ο τελεστής που παίρνει σαν όρισμα ένα φυσικό αριθμό και επιστρέφει τον επόμενό του, όπως ορίζεται στην δεύτερη εξίσωση
- _+_ παίρνει δύο φυσικούς αριθμούς και επιστρέφει το άθροισμά τους

Στους τελεστές όταν χρησιμοποιούνται '_' δηλώνεται η θέση εισαγωγής των ορισμάτων, με πιο αριστερά το πρώτο όρισμα μετά το δεύτερο κ.τ.λ. Το ορίσματα του τελεστή, ή αλλιώς η λίστα από τους τύπους που μας δείχνουν και την πολλαπλότητα του τελεστή (arity), δηλώνονται πάντα πριν το → ενώ το είδος ή αλλιώς πεδίο τιμών του τελεστή (coarity) μετά, δηλαδή:

```
operator declaration = op operator: list of sort names -> sort name
```

Ο κάθε τελεστής ορίζεται μέσω μίας εξίσωσης. Οι εξισώσεις αρχίζουν με τη λέξη `eq` (equation), ενώ οι εξισώσεις που περιέχουν και κάποια συνθήκη με τη λέξη `ceq` (conditional equation). Οι όροι της εξίσωσης ενώνονται με το σύμβολο της ισότητας (=), και πάντα στο τέλος κάθε εξίσωσης υπάρχει το σύμβολο της τελείας (.).

```
equation declaration = standard eq | condition eq | behavior eq |
condibehavior eq

standard_eq = eq expression = expression .
condition_eq = ceq expression = expression if boolean .
behavior_eq = beq expression = expression .
condibehavior_eq = bceq expression = expression if boolean .
```

Οι όροι της εξίσωσης που βρίσκονται στις δύο πλευρές της πρέπει να είναι του ίδιου τύπου. Για την υπό όρους εξίσωση χρειάζεται μια boolean έκφραση, δηλαδή τύπου BOOL. Αυτό σημαίνει ότι δύο όροι είναι ίσοι αν η boolean έκφραση έχει την τιμή true. Συμπεριφορικές (`beq`) και υπό συνθήκη συμπεριφορικές (`bceq`) εξισώσεις χρησιμοποιούνται για αναφορά σε θεωρήματα, ή για να οριστούν συμπεριφοριακοί τελεστές σε όρους άλλων.

Η δήλωση των μεταβλητών γίνεται με τη λέξη `vars` (variables) και στην περίπτωση που είχαμε μόνο μια μεταβλητή τότε θα την δηλώναμε απλά με τη λέξη `var`. Κάθε μεταβλητή είναι ενός συγκεκριμένου τύπου.

Μια μεταβλητή δεν επιτρέπεται να δηλωθούν με το ίδιο όνομα μεταβλητής διάφορα είδη τύπων μεταβλητών και δηλώνεται εντός του τμήματος που εφαρμόζεται με τον εξής τρόπο

```
variable_declaration = var variable_name : sort_name
```

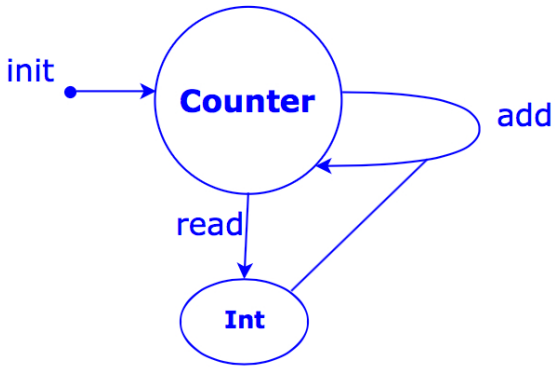
Στο τμήμα, αν δίπλα στη λέξη `module` υπάρχει το σύμβολο ! (`mod!`) τότε σημαίνει ότι το τμήμα έχει μια αρχική, σφιχτή σημασιολογία (tight semantics), δηλαδή πρόκειται για ένα μοναδικό μοντέλο της προδιαγραφής. Αν το τμήμα ορίζει μια κλάση μοντέλων, δηλαδή έχει χαλαρή σημασιολογία (loose semantics), τότε ακολουθεί το σύμβολο * (`mod*`).

module για σφιχτή σημασιολογία:
```
module! module_name {
module_element*
}
```
module για χαλαρή σημασιολογία:
```
module* module_name {
module_element*}
```

Ένα τμήμα μπορεί να εισαχθεί σε ένα άλλο τμήμα ως protecting (`pr`), extending (`ex`) ή ως using (`us`). Ένα protecting τμήμα όταν εισαχθεί μετά δεν είναι δυνατή η αλλαγή του. Χρησιμοποιείται δηλαδή όπως ακριβώς είναι, χωρίς να μπορεί να ελαττωθεί ή να επεκταθεί. Αντίθετα, σε ένα extending τμήμα είναι δυνατό να προστεθούν νέα στοιχεία αλλά δεν αλλάζουν τα στοιχεία που ήδη έχει, δηλαδή δεν μπορεί να γίνει ελάττωσή του. Τέλος, ένα using τμήμα μπορεί να μεταβληθεί τελείως, είτε να ελαττωθεί είτε να επεκταθεί. Κάθε τμήμα εξ' αρχής εισάγει το module BOOL που ορίζει τον τύπο δεδομένων boolean.

Ένα ακόμα παράδειγμα συμπεριφοριακής προδιαγραφής που βασίζεται στην άλγεβρα με κρυμμένους τύπους είναι ένας μετρητής ακεραίων αριθμών.

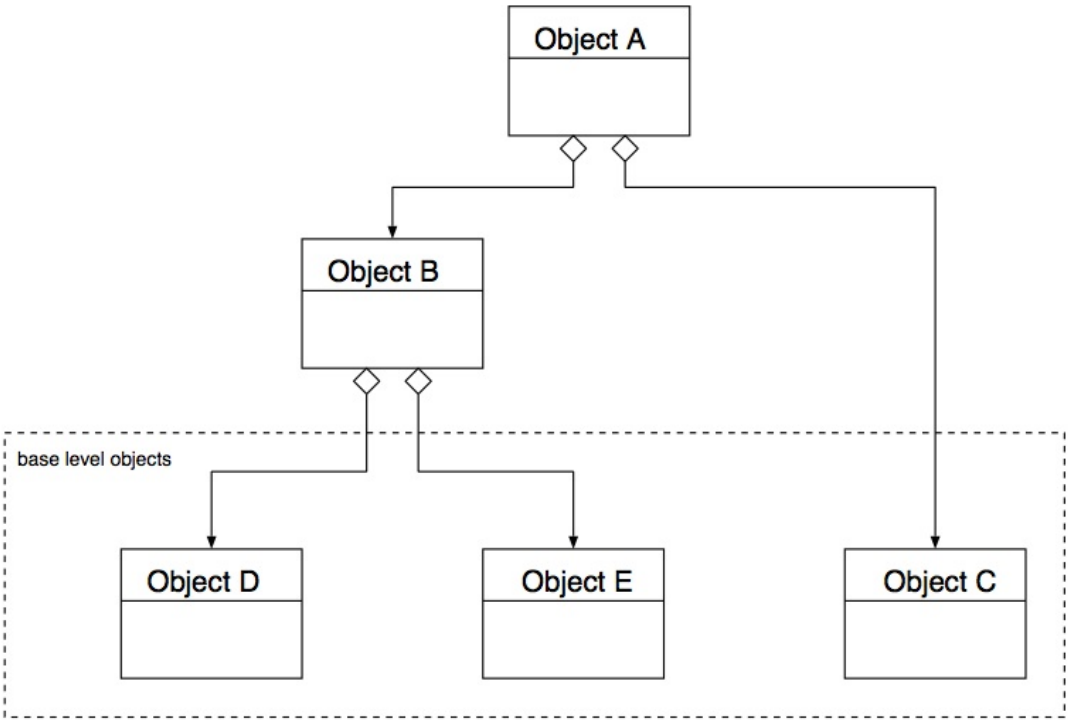
```
mod* COUNTER {
  protecting(INT)
*[ Counter ]*
-- initial state
op init : -> Counter
-- method
bop add : Int Counter ->
Counter
-- attribute
bop read : Counter -> Int
var I : Int
var C : Counter
eq read(init) = 0 .
  eq read(add(I, C)) = I +
read(C) .
}
```



Το όνομα του τμήματος είναι `COUNTER` και έχει χαλαρή σημασιολογία απεικονίζοντας μια κλάση των μοντέλων `COUNTER`. Ο κρυμμένος τύπος των μετρητών `Counter` δηλώνεται ανάμεσα στα *[ ]*. Επίσης protecting (`INT`) σημαίνει ότι το τμήμα `INT`, που υπάρχει ήδη στην CafeOBJ και περιγράφει τους ακέραιους αριθμούς έχει εισαχθεί στο τμήμα `COUNTER`.

4.3 Σύνθεση συμπεριφοριακών αντικειμένων

Γενικά, ένα από τα πιο σημαντικά θέματα των αντικειμενοστραφών μεθόδων είναι η επαναχρησιμοποίηση του πηγαίου κώδικα. Στην αντικειμενοστραφή προδιαγραφή είναι δυνατή και η επαναχρησιμοποίηση των αποδείξεων, κάτι το οποίο είναι πολύ σημαντικό στο στάδιο της επαλήθευσης. Υπάρχουν δύο τεχνικές για την επαναχρησιμοποίηση κώδικα: η σύνθεση και η κληρονομική. Εμείς θα δούμε τη σύνθεση.



Εικόνα 9: Σύνθεση συμπεριφοριακών αντικειμένων [50]

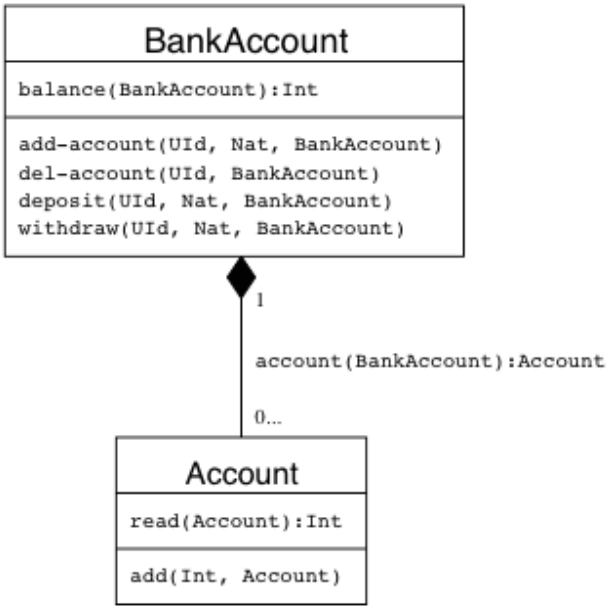
Η επαναχρησιμοποίηση των προδιαγραφών γίνεται μέσω τελεστών προβολής (projection operations). Οι τελεστές προβολής ορίζονται για κάθε αντικείμενο το οποίο παίρνει μέρος στη διαδικασία της σύνθεσης (σύνθετο αντικείμενο) ώστε να παρακολουθείται ύστερα η κατάσταση του από το συντιθέμενο ή συστατικό αντικείμενο. Όλες οι μέθοδοι του αντικειμένου που προκύπτει από τη σύνθεση σχετίζονται με τις μεθόδους των επιμέρους αντικειμένων του κάνοντας χρήση των τελεστών προβολής. Ένα μη σύνθετο αντικείμενο, δηλαδή ένα αντικείμενο βάσης (base level object), όπως τα D, E και C του παραπάνω σχήματος δεν έχουν τελεστές παρακολούθησης. Επίσης, δύο μέθοδοι ενός αντικείμενου που έχει προκύψει από σύνθεση, ανήκουν στην ίδια ομάδα μεθόδων όταν σχετίζονται με το ίδιο αντικείμενο από το οποίο έχουν συσταθεί.

Η σύνθεση αντικειμένων μπορεί να διαχωριστεί σε δύο κατηγορίες ανάλογα με το πως συνδέονται τα αντικείμενα. Έχουμε την ταυτόχρονη (ή παράλληλη) σύνδεση και την συγχρονισμένα ταυτόχρονη σύνδεση.

Η αντικειμενοστραφής μέθοδος στην CafeOBJ που υποστηρίζει την ταυτόχρονη σύνθεση συμπεριφοριακών αντικειμένων (concurrent object composition) χρησιμοποιεί τους τελεστές προβολής από τον κρυμμένο τύπο του σύνθετου αντικειμένου στους κρυμμένους τύπους των συστατικών αντικειμένων. Μπορούμε να διαχωρίσουμε δύο είδη αντικειμένων, τα στατικά και τα δυναμικά. Τα δυναμικά αντικείμενα μπορούν να δημιουργηθούν ή να διαγραφούν κατά τη διάρκεια που το σύστημα είναι σε λειτουργία, ενώ για τα στατικά αντικείμενα δεν υπάρχει αυτή η δυνατότητα.

Χρησιμοποιώντας αντικειμενοστραφή ορολογία μπορούμε να πούμε ότι τα δυναμικά αντικείμενα ορίζουν μια κλάση, ενώ τα στατικά αναπαριστούν τη συγκεκριμένη κατάσταση στην οποία η κλάση περιέχει μόνο ένα αντικείμενο.

Για παράδειγμα, ας θεωρήσουμε ένα σύστημα λογαριασμών τράπεζας το οποίο αποτελείται από ξεχωριστούς λογαριασμούς. Μέσω της απεικόνισης UML το παράδειγμα αυτό μπορεί να αναπαρασταθεί ως εξής:



Εικόνα 10: Σύνθεση αντικειμένων Account

Η προδιαγραφή των προσωπικών λογαριασμών μπορεί να θεωρηθεί ως ένα δυναμικό σύστημα από μετρητές (counters) ακεραίων αριθμών. Το επόμενο κομμάτι κώδικα ορίζει την κλάση των μετρητών, με τον μετρητή να αναγνωρίζεται εφόσον έχει παραμετροποιηθεί από το X.

```
mod* COUNTER(X :: TRIV) {
```

```
protecting(INT)
*[ Counter ]*
op init-counter : Elt -> Counter
op no-counter : -> Counter
bop add : Int Counter -> Counter
bop amount : Counter -> Nat
var ID : Elt
var I : Int
var C : Counter
beq add(I, no-counter) = no-counter .
eq amount init-counter(ID) = 0 .
ceq amount add(I, C) = I + amount(C) if I + amount(C) >=0 .
ceq amount add(I, C) = amount(C) if I + amount(C) < 0 .
}
```

Ο χώρος των καταστάσεων του μετρητή αναπαρίσταται μέσω του κρυμμένου τύπου Counter. Ο τελεστής `init-counter` ορίζει κάθε αρχική κατάσταση και είναι μια σταθερά. Ο τελεστής δράσης `add` προσθέτει ή αφαιρεί έναν ακέραιο (αφαιρεί στην περίπτωση που ένας αρνητικός αριθμός έχει περάσει ως όρισμα), αλλάζοντας έτσι την κατάστασή του. Μπορούμε να παρατηρήσουμε την κατάσταση του μετρητή χρησιμοποιώντας τον τελεστή παρατήρησης `amount`. Αυτός ο μετρητής επιτρέπει την εφαρμογή της αλλαγής κατάστασης μέσω του τελεστή `add` μόνο όταν η παρατήρηση της τελικής κατάστασης είναι θετική. Αυτό σημαίνει ότι καμία αφαίρεση που θα οδηγήσει σε ένα αρνητικό αποτέλεσμα δεν επιτρέπεται. Στην αρχική κατάσταση, όπως φαίνεται από την πρώτη εξίσωση η τιμή του μετρητή είναι ίση με μηδέν, ενώ στην επόμενη εξίσωση δίνει την τιμή του μετρητή μετά την αύξησή του κατά I.
Το τμήμα `COUNTER` μπορεί να αναπαραστηθεί ως ένα δυναμικό αντικείμενο και απαιτούμε ένα αναγνωριστικό για τον προσδιορισμό κάθε ξεχωριστού `COUNTER`. Η λειτουργία no-counter χρησιμοποιείται για καταστάσεις όπου δημιουργείται κάποιο λάθος, δηλαδή όταν κάποια άλλα αντικείμενα προσδιορίζουν ένα λανθασμένο αναγνωριστικό, όταν δηλαδή το αντικείμενο δεν υπάρχει.
Οι λογαριασμοί είναι απλά μια μετονομασία του `COUNTER` και αναφέρονται στο `USER-ID` (μέσω αναγνωριστικών).

```
mod* USER-ID
[ UId ]
mod* ACCOUNT
protecting(COUNTER(X <= view to USER-ID {sort Elt -> UId })
* {hsort Counter -> Account,
      op init-counter -> init-account,
      op no-counter -> no-account })
```

Τώρα μπορούμε να κάνουμε τη σύνθεση των αντικειμένων που είναι οι λογαριασμοί ώστε να δομούν ένα δυναμικό σύστημα λογαριασμών τράπεζας.

```
mod* ACCOUNT-SYS {
protecting(ACCOUNT)
*[ AccountSys ]*
```

```
op init-account-sys : -> AccountSys
bop add-account : UId Nat AccountSys -> AccountSys
bop del-account : UId AccountSys -> AccountSys
bop deposit : UId Nat AccountSys -> AccountSys
bop withdraw : UId Nat AccountSys -> AccountSys
bop balance : UId AccountSys -> Nat
bop account : UId AccountSys -> Account
vars U U' : UId
var A : AccountSys
var N : Nat
eq account(U, init-account-sys) = no-account .
ceq account(U, add-account(U', N, A)) = add(N, init-account(U)) if
U == U' .
ceq account(U, add-account(U', N, A)) = account(U, A) if U =/= U' .
ceq account(U, del-account(U', A)) = no-account if U == U' .
ceq account(U, del-account(U', A)) = account(U, A) if U =/= U' .
ceq account(U, deposit(U', N, A)) = add(N, account(U, A)) if U =/=
U' .
ceq account(U, deposit(U', N, A)) = account(U, A) if U =/= U' .
ceq account(U, withdraw(U', N, A)) = add(-(N), account(U, A)) if U
== U' .
ceq account(U, withdraw(U', N, A)) = account(U, A) if U =/= U' .
eq balance(U, A) = amount account(U, A) .
}
```

Παρακάτω είναι τα βήματα που εμπλέκονται στον προσδιορισμό ενός σύνθετου αντικειμένου:

1. εισάγουμε τα αντικείμενα που αποτελούν τα συστατικά στοιχεία (`ACCOUNT`),
2. ορίζουμε έναν νέο κρυμμένο τύπο για το σύνθετο αντικείμενο και ένα (συμπεριφοριακό) τελεστή προβολής στον κρυμμένο τύπο κάθε συστατικού (`account`).
3. ορίζουμε τις δράσεις του σύνθετου αντικειμένου (μέσω των αντίστοιχων τελεστών `withdraw` και `deposit`) που αντιστοιχούν στις δράσεις των σύνθετων αντικειμένων και εκφράζουν τη σχέση μεταξύ των δράσεων αυτών και τις δράσεις των συστατικών των αντικειμένων μέσα από (αυστηρές) εξισώσεις, και
4. (τελικά) ορίζουμε μερικές παρατηρήσεις για τα σύνθετα αντικείμενα ως συντομεύσεις στις παρατηρήσεις των συστατικών αντικειμένων (εξίσωση balance).

Στην αρχική κατάσταση (`init-account-sys`), το σύστημα των τραπεζικών λογαριασμών δεν περιέχει κανένα λογαριασμό. Με τη λειτουργία `add-account` γίνεται η προσθήκη των επιμέρους λογαριασμών και με την `del-account` η διαγραφή τους.

Οι τελεστές προβολής υπόκεινται σε αρκετά ακριβείς τεχνικούς όρους, οι οποίοι είναι μαθηματικά διατυπωμένοι στο [44].

Συγχρονισμός σύνθεσης

Στο παραπάνω παράδειγμα, δεν υπάρχει συγχρονισμός μεταξύ των συστατικών του συστήματος, αλλά ταυτοχρονισμός ανάμεσα στα συστατικά αντικείμενα. Αυτό σημαίνει ότι δύο μέθοδοι που είναι σε διαφορετικές ομάδες μεθόδων έχουν όλες τις καταστάσεις που περιέχουν αυτές τις μεθόδους ισοδύναμες κατά την έννοια της συμπεριφοράς (behavioural equivalent).

Παραθέτουμε πάλι την προδιαγραφή για τον μετρητή των ακεραίων αριθμών ως ένα στατικό αντικείμενο:

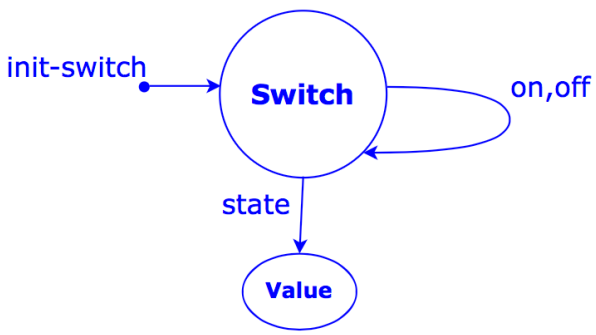
```
mod* COUNTER {
protecting(INT)
*[ Counter ]*
op init-counter : -> Counter
bop add : Int Counter -> Counter
bop amount : Counter -> Int
var I : Int
var C : Counter
eq amount(init-counter) = 0 .
eq amount(add(I, C)) = I + amount(C) .
}
```

Θεωρούμε και ένα διακόπτη, ο οποίος βρίσκεται στην κατάσταση `on` όταν η τιμή του μετρητή αυξάνεται κατά έναν ακέραιο και `off` όταν η τιμή του μειώνεται, δηλαδή η τιμή του μετρητή εξαρτάται από την τιμή του διακόπτη.

```
mod! ON-OFF {
[ Value ]
ops on off : -> Value
}
```

Στο παραπάνω τμήμα προδιαγραφών ο τύπος δεδομένων `Value` παίρνει τις τιμές `on` και `off` και χρησιμοποιείται στην προδιαγραφή του `SWITCH` μέσω της δήλωσης protecting.

```
mod* SWITCH {
protecting(ON-OFF)
*[ Switch ]*
op init-switch : -> Switch
bop on : Switch -> Switch
bop off : Switch -> Switch
bop state : Switch -> Value
var S : Switch
eq state(init-switch) = off .
eq state(on(S)) = on .
eq state(off(S)) = off .
}
```



Η προδιαγραφή SWITCH έχει μια σταθερά init για την αρχική κατάσταση, δύο τελεστές δράσεων (on, off) που εξυπηρετούν την κατάσταση του διακόπτη και μια παρατήρηση (state) η οποία επιστρέφει την τιμή της κατάστασης.
Και η τελική συγχρονισμένη σύνθεση των αντικειμένων COUNTER και SWITCH είναι:

```
mod* COUNTER-WITH-SWITCH {
protecting(SWITCH + COUNTER)
*[ Cws ]*
op init : -> Cws
bop put : Int Cws -> Cws
bop add : Cws -> Cws
bop sub : Cws -> Cws
bop amount : Cws -> Int
bop counter : Cws -> Counter
bop switch : Cws -> Switch
var N : Int
var C : Cws
eq amount(C) = amount(counter(C)) .
eq [s-1] : switch(init) = init-switch .
eq [s-2] : switch(put(N, C)) = switch(C) .
eq [s-3] : switch(add(C)) = on(switch(C)) .
eq [s-4] : switch(sub(C)) = off(switch(C)) .
eq [c-1] : counter(init) = init-counter .
cq [c-2] : counter(put(N, C)) = add(N, counter(C)) if
state(switch(C)) == on .
cq [c-3] : counter(put(N, C)) = add(-(N), counter(C)) if
state(switch(C)) == off .
eq [c-4] : counter(add(C)) = counter(C) .
eq [c-5] : counter(sub(C)) = counter(C) .
}
```

Παρατηρούμε ότι ο κρυμμένος τύπος του σύνθετου αντικειμένου COUNTER-WITH-SWITCH είναι Cws, και οι τελεστές προβολής είναι οι switch και counter. Υπάρχουν τρεις δράσεις (put, add, pub), μια σταθερά (init) για την αρχική κατάσταση, ένας τελεστής παρατήρησης (amount) και δύο τελεστές προβολής (counter, switch). Η συμπεριφοριακή ισοδυναμία COUNTER-WITH-SWITCH είναι η σύνδεση των συμπεριφοριακών ισοδυναμιών των SWITCH και COUNTER και ορίζεται έτσι:

```
mod BEQ-CWS {
protecting(COUNTER-WITH-SWITCH)
op R :CwsCws -> Bool
vars C1 C2 : Cws
eq C1 R C2 = counter(C1) =*= counter(C2) and switch(C1) =*=
switch(C2) .
}
```

Στην περίπτωση αυτή λοιπόν, έχουμε συγχρονισμό οποίος πραγματοποιείται όταν:

- η κατάσταση του αντικειμένου που προβάλλεται μέσω του τελεστή προβολής, εξαρτάται από την κατάσταση ενός διαφορετικού συστατικού αντικειμένου

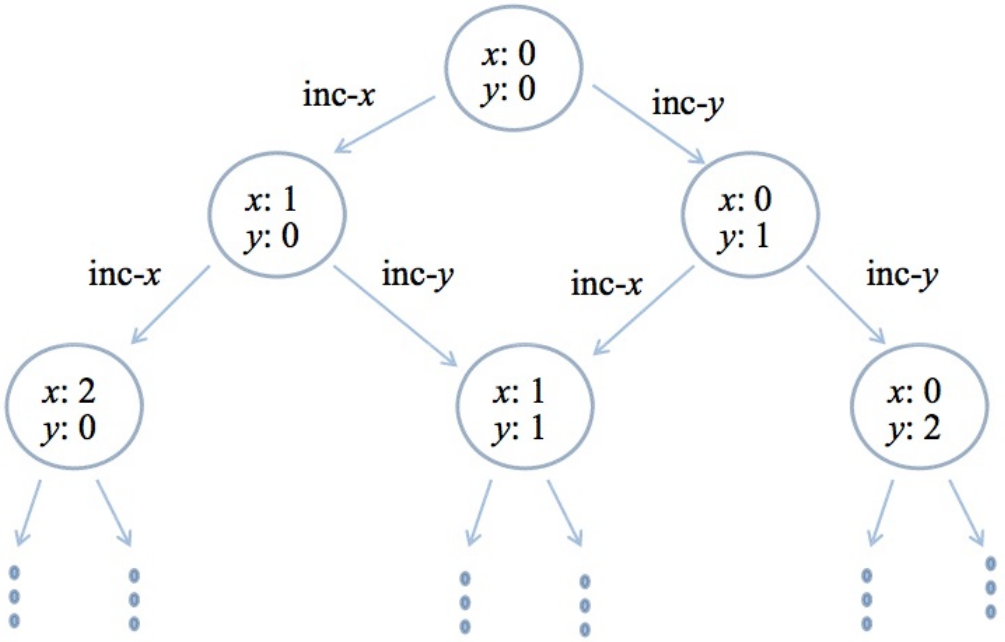
- οι μέθοδοι (δράσεις) του σύνθετου αντικειμένου αλλάζουν ταυτόχρονα την κατάσταση διαφορετικών συστατικών αντικειμένων.

4.4 Παρατηρήσιμα Συστήματα Μετάβασης

4.4.1 Μηχανή καταστάσεων

Πριν ορίσουμε τι είναι ένα ΠΣΜ-Παρατηρήσιμο Σύστημα Μετάβασης (OTS-Observational Transition System) θα δούμε ένα παράδειγμα μιας μηχανής καταστάσεων και πως αυτή μπορεί να οριστεί μέσα από την CafeOBJ.

Ας δούμε μια απλή μηχανή καταστάσεων INCXY και το διάγραμμα μετάβασης των καταστάσεών της που μπορεί να το απεικονίσουμε σχηματικά ως εξής:



Εικόνα 11: A simple State Machine [49]

Η παραπάνω μηχανή κατάστασης `INCXY` γράφεται σε κώδικα της CafeOBJ ως εξής :

```
mod* INCXY
{
pr (NAT)
*[Sys]*
op init : -> Sys
bops x y : Sys -> Nat
bops inc-x inc-y : Sys -> Sys
var S : Sys
- init
```

```
eq x(init) = 0 .
eq y(init) = 0 .
- for inc-x
eq x(inc-x(S)) = x(S) + 1
eq y(inc-y(S)) = y(S)
- for inc-y
eq x(inc-x(S)) = x(S)
eq y(inc-y(S)) = y(S) + 1
}
```

Ο κρυφός τύπος `Sys` αναπαριστά το χώρο των κατάστασεων της μηχανής, ενώ ο τύπος `Nat` αναπαριστά το σύνολο των φυσικών αριθμών και είναι ο ορατός τύπος. Η σταθερά `init` δηλώνει μια αυθαίρετη αρχική κατάσταση, τα `x` και `y` είναι συναρτήσεις παρατήρησης που παρατηρούν αντίστοιχα τις μεταβλητές x και y, ενώ τα `inc-x` και `inc-y` είναι συναρτήσεις μετάβασης.

Οι πρώτες δύο εξισώσεις ορίζουν την αυθαίρετη αρχική κατάσταση `init`, ενώ οι επόμενες ορίζουν τις μεταβάσεις `inc-x` και `inc-y`.

Οι τύποι που υπάρχουν στο παράδειγμά μας είναι δύο:
α) οι ορατοί τύποι που χρησιποιούνται για να αντιλαμβάνονται συλλογές από πληροφορίες τιμών ή τύπους δεδομένων (data values or data types) και
β) οι κρυφοί τύποι για τις καταστάσεις της μηχανής.

Οι συμπεριφοριακοί τελεστές, αυτοί δηλαδή που έχουν τελεστές με κρυμμένους τύπους, χωρίζονται επίσης σε δύο κατηγορίες:

- αυτούς που έχουν ακριβώς έναν τελεστή με κρυμμένο τύπο και το πεδίο τιμών τους είναι ένας ορατός τύπος που ονομάζεται συνάρτηση παρατήρησης. Εδώ, οι συναρτήσεις παρατήρησης της μηχανής `INCXY` είναι οι `x` και `y`.

- και αυτούς που έχουν πάλι ακριβώς έναν τελεστή με κρυμμένο τύπο και το πεδίο τιμών τους είναι ο ίδιος κρυφός τύπος που ονομάζεται συνάρτηση μετάβασης. Εδώ, οι συναρτήσεις μετάβασης της μηχανής `INCXY` είναι οι `inc-x` και `inc-y`.

Οι προδιαγραφές συμπεριφοράς (behavioral specifications) είναι αλγεβρικές προδιαγραφές συστημάτων.

- Όλος ο χώρος των καταστάσεων συμβολίζεται ως ένας κρυφός τύπος.

- Οι καταστάσεις χαρακτηρίζονται από συναρτήσεις παρατήρησης.

- Οι καταστάσεις μετάβασης αντιστοιχούν σε συναρτήσεις μετάβασης.

Οι προδιαγραφές συμπεριφοράς γενικεύουν ένα συμβατικό σύστημα παρατήρησης στο ότι ακόμα κι αν κάθε συνάρτηση παρατήρησης επιστρέφει την ίδια τιμή για δύο καταστάσεις

σε μία προδιαγραφή συμπεριφοράς, οι καταστάσεις αυτές μπορούν να είναι διαφορετικές.

4.4.2 Ορισμός: Παρατηρήσιμο Σύστημα Μετάβασης (ΠΣΜ)

Έστω ότι υπάρχει ένας καθολικός χώρος καταστάσεων που συμβολίζεται Y. Ακόμα, κάθε τύπος δεδομένων που χρησιμοποιείται στην προδιαγραφή του ΠΣΜ έχει ορισθεί προηγουμένως, καθώς και η σχέση ισοδυναμίας μεταξύ δυό τύπων δεδομένων υ_1, υ_2 είναι ορισμένη ως $\upsilon_1 = \upsilon_2$.

Ένα ΠΣΜ $S = \langle O, I, T \rangle$ αποτελείται από τα ακόλουθα [47]:

- O : Ένα σύνολο από παρατηρητές. Κάθε παρατηρητής $o \in O$ είναι μια συνάρτηση παρατήρησης $o : Y\ D_{o_1} \ldots D_{o_m} \rightarrow D_o$, όπου το D είναι ένας τύπος δεδομένων για τις παρατηρούμενες τιμές, ο οποίος μπορεί να αλλάζει από παρατηρητή σε παρατηρητή. Η σχέση ισοδυναμίας $(\upsilon_1 =_S \upsilon_2)$ ανάμεσα σε δύο καταστάσεις $\upsilon_1, \upsilon_2 \in Y$ ορίζεται $\forall\ o \in O$, ως, $o(\upsilon_1, x_1, \ldots, x_m) = o(\upsilon_2, x_1, \ldots, x_m)$, για κάθε $x_1 : D_{o_1} \ldots, x_m : D_{o_m}$.

- I : Το σύνολο των σύνολο αρχικών καταστάσεων ώστε $I \subseteq Y$.

- T : Ένα σύνολο κανόνων μετάβασης (ή αλλιώς για συντομία θα τις αναφέρουμε απλά μεταβάσεις). Κάθε $\tau \in T$ είναι μια συνάρτηση $\tau : Y/=_S D_{\tau_1} \ldots D_{\tau_m} \rightarrow Y/=_S$ ώστε $\tau(\upsilon_1, y_1, \ldots, y_m) =_S \tau(\upsilon_2, y_1, \ldots, y_m)$ για κάθε $[\upsilon] \in Y/=_S$, $\upsilon_1, \upsilon_2 \in [\upsilon]$ και $y_1 : D_{\tau_1} \ldots, y_m : D_{\tau_m}$. Η $\tau(\upsilon)$ καλείται η διάδοχη κατάσταση για κάθε $\upsilon \in Y$. Η συνθήκη c_τ για μια μετάβαση $\tau \in T$ καλείται αποτελεσματική συνθήκη της μετάβασης με $c_\tau : Y/=_S D_{\tau_1} \ldots D_{\tau_m} \rightarrow$ Bool. Η αποτελεσματική συνθήκη πρέπει να ικανοποιεί την εξής προϋπόθεση:
  Για κάθε κατάσταση $\upsilon \in Y$, αν c_τ είναι ψευδής (έχει ψευδή τιμή στην υ), τότε $\upsilon =_S \tau(\upsilon, y_1, \ldots, y_m)$.

Ένα ΠΣΜ μπορεί να περιγραφτεί μέσω της γλώσσας CafeOBJ. Οι παρατηρήσιμες τιμές ορίζονται μέσω παρατηρήσεων και οι κανόνες μετάβασης μέσω των δράσεων, σύμφωνα με τους κανόνες της CafeOBJ. Γενικά, οι παρατηρήσεις και οι κανόνες μετάβασης συμβολίζονται με $o_{i_1,\ldots,i_m}$ και $\tau_{j_1,\ldots,j_n}$ αντίστοιχα, εφόσον m, n ≥ 0 και υποθέτουμε ότι υπάρχουν τύποι δεδομένων D_k, Τέτοια ώστε $k \in D_k$ $(k = i_1, \ldots, i_m, j_1, \ldots, i_m)$.
Για παράδειγμα, σε ένα διάνυσμα ακεραίων α που γίνεται κλήση ενός στοιχείου p μπορεί να συμβολίζεται ως παρατηρήσιμη τιμή α_p, και η προσθήκη στο διάνυσμα ενός i-οστου στοιχείου μπορεί να συμβολίζεται από έναν κανόνα μετάβασης inc-$\alpha_{p,i}$.

Μια εκτέλεση του S είναι μια άπειρη ακολουθία καταστάσεων $\upsilon_0, \upsilon_1, \ldots$ που ικανοποιούν τα εξής:

- Αρχικοποίηση : $\upsilon_0 \in I$.

- Συνέπεια : Για κάθε $i \in \{0,1,...\}$, $\upsilon_{i+1} =_S \tau(\upsilon_i)$ για κάποιο $\tau \in T$.

Μια κατάσταση ονομάζεται προσιτή (reachable) για το ΠΣΜ S, αν και μόνο αν μπορεί να εμφανιστεί ως μια εκτέλεση του συστήματος S. Έστω R_S να είναι ένα σύνολο όλων των προσιτών καταστάσεων του S. Εμείς θα ασχοληθούμε μόνο με invariants, δηλαδή αμετάβλητα κατηγορήματα καταστάσεων, ή αλλιώς τις ιδιότητες που ισχύουν σε κάθε κατάσταση του S.

Ο ορισμός ενός τέτοιου κατηγορήματος αμετάβλητης κατάστασης είναι ο παρακάτω:

invariant $p = (\forall \upsilon \in I.\ p(\upsilon)) \wedge (\forall \upsilon \in R_S. \forall \tau \in T.\ (p(\upsilon) \Rightarrow p(\tau(\upsilon))))$

που σημαίνει ότι η *p* είναι αληθής για κάθε προσιτή κατάσταση του S.

4.4.3 Περιγραφή ενός ΠΣΜ στην CafeOBJ

Έστω λοιπόν το σύνολο όλων των καταστάσεων Y, το οποίο μοντελοποιείται ως ένας κρυμμένος τύπος έστω H. Ένας παρατηρητής $o_{i_1,...,i_m} \in O$ μοντελοποιείται στην CafeOBJ ως ένας τελεστής παρατήρησης. Εάν υποτεθεί ότι υπάρχουν οι τύποι δεδομένων D_k ($k = i_1, \ldots, i_m$) οι οποίοι δηλώνονται με τους ορατούς τύπους V_k ($k = i_1, \ldots, i_m$) ο τελεστής παρατήρησης της CafeOBJ που αντιστοιχεί στις παρατηρούμενες τιμές $o_{i_1,...,i_m}$ δηλώνεται ως εξής [47]:

bop o : H $V_{i_1} \ldots V_{i_m} \rightarrow V$

Κάθε κατάσταση στο σύνολο των αρχικών καταστάσεων *I*, μοντελοποιείται με μια σταθερά, δηλαδή έναν τελεστή χωρίς ορίσματα, έστω *init*, που μοντελοποιείται ως εξής:

op *init* : $\rightarrow$ H

Υποθέτοντας ότι οι αρχικές τιμές των $o_{i_1,...,i_m}$ είναι $f(i_1, \ldots, i_m)$. Αυτό εκφράζεται με την παρακάτω εξίσωση:

eq $o(init, X_{i_1}, \ldots, X_{i_m}) = f(X_{i_1}, \ldots, X_{i_m})$.

Η X_k ($k = i_1, \ldots, i_m$) είναι μια μεταβλητή στην CafeOBJ με τύπο V_k και $f(X_{i_1}, \ldots, X_{i_m})$ είναι ένας τύπος που ορίζει την $f(i_1, \ldots, i_m)$, δηλαδή την αρχική τιμή των $o_{i_1,...,i_m}$.

Μια μετάβαση $\tau_{j1,\ldots,jn} \in T$ μοντελοποιείται σαν ένας τελεστής δράσης της CafeOBJ και δηλώνεται ως εξής:

bop a : H V_{j1} . . . V_{jn} → H
όπου V_k ο ορατός τύπος δεδομένων που αντιστοιχεί στον τύπο δεδομένων D_k ($k = j_1, \ldots, j_n$).

Οι συναρτήσεις μετάβασης $\tau_{j1,\ldots,jn}$ εφαρμόζονται στις καταστάσεις του συνόλου καταστάσεων Y. Για να είναι αποτελεσματικές και να επιτύχουμε αλλαγή της τιμής των παρατηρητών $o_{i1,\ldots,im}$ πρέπει να ικανοποιούνται οι αποτελεσματικές συνθήκες τους. Αυτό, δηλώνεται στην CafeOBJ ως εξής:

ceq o(a(W, X_{j1} ,. . . , X_{jn}), X_{i1} ,. . . , X_{im}) = e-a(W, X_{j1} ,. . . , X_{jn}, X_{i1} ,. . . , X_{im}) if c-a(W, X_{j1} ,. . . , X_{jn}) .

όπου W: μια μεταβλητή της CafeOBJ για τον τύπο H και X_k μια μεταβλητή για τον τύπο δεδομένων V_k ($k = i_1, \ldots, i_m$). Ο όρος a(W, X_{j1} ,. . . , X_{jn}) είναι η επόμενη κατάσταση της W σε αντιστοιχία των X_{j1} ,. . . , X_{jn} και $\tau_{j1,\ldots,jn}$. H e-a (W, X_{j1} ,. . . , X_{jn}, X_{i1} ,. . . , X_{im}) είναι η τιμή που επιτρέφει ο παρατηρητής $o_{i1,\ldots,im}$ στην επόμενη αυτή κατάσταση που εφαρμόζεται η μετάβαση. H c-a(W, X_{j1} ,. . . , X_{jn}) είναι η αποτελεσματική συνθήκη της μετάβασης $\tau_{j1,\ldots,jn}$.

Αν η αποτελεσματική συνθήκη δεν ισχύει, τότε η τιμή του παρατηρητή δεν αλλάζει, αυτό μπορεί να γραφτεί σαν μια υπό συνθήκη εξίσωση της μορφής:

ceq a(W, X_{j1} ,. . . , X_{jn}) = W if not c-a(W, X_{j1} ,. . . , X_{jn}) .

4.4.4 Επαλήθευση ενός ΠΣΜ

Απόδειξη Αμετάβλητων Καταστάσεων

Έστω ότι θέλουμε να αποδείξουμε ότι ένα σύστημα έχει μια αμετάβλητη ιδιότητα (invariant property). Πρώτα μοντελοποιούμε το σύστημα ως ΠΣΜ στη γλώσσα CafeOBJ.
Έστω H ο κρυμμένος τύπος που δηλώνει το χώρο καταστάσεων Y και έστω $pred_1(s, \boldsymbol{x}_1)$ η αμετάβλητη κατάσταση, όπου s είναι η ελεύθερη μεταβλητή για τις καταστάσεις και $\boldsymbol{x}_1$ οι υπόλοιπες ελεύθερες μεταβλητές. Είναι συχνά αδύνατον να αποδείξουμε ότι η $pred_1(s, \boldsymbol{x}_1)$ είναι αμετάβλητη κατάσταση, ελέγχοντάς την μόνη της. Υποθέτουμε ότι είναι δυνατόν να αποδείξουμε ότι η $pred_1(s, \boldsymbol{x}_1)$ μαζί με τις n − 1 άλλες καταστάσεις είναι αμετάβλητες στο ΠΣΜ. Έτσι οι $pred_2(s, \boldsymbol{x}_2),....,pred_n(s, \boldsymbol{x}_n)$ είναι αυτές οι n-1 καταστάσεις. Τελικά, αποδεικνύουμε ότι η $(pred_1(s, \boldsymbol{x}_1) \wedge ... \wedge pred_n(s, \boldsymbol{x}_n)) = pred(s, \boldsymbol{x}_1,...,\boldsymbol{x}_n)$ είναι αμετάβλητη κατάσταση, αντί να το αποδείξουμε για την αρχική.
Παρότι μερικές φορές οι αμετάβλητες καταστάσεις μπορούν να αποδειχτούν με αναγωγή (reduction) ή και ανάλυση κατά περίπτωση (case analysis), συχνά χρειάζεται να χρησιμοποιήσουμε επαγωγή (induction), ιδιαίτερα στον αριθμό των μεταβατικών κανόνων που εφαρμόζονται ή εκτελούνται.
Ας υποθέσουμε ότι η αμετάβλητη κατάσταση $pred(s, \boldsymbol{x}_1,...,\boldsymbol{x}_n)$ αποδεικνύεται μέσω της επαγωγικής μεθόδου στον αριθμό των κανόνων μετάβασης που εφαρμόζουμε. Θεωρούμε το επαγωγικό βήμα στο οποίο ισχύει ότι κάθε μεταβατικός κανόνας που δηλώνεται από μια μεταβατική συνάρτηση σε έναν τελεστή δράσης της CafeOBJ, έστω a, διατηρεί την $pred(s, \boldsymbol{x}_1,...,\boldsymbol{x}_n)$. Επομένως τελικά πρέπει να δείξουμε ότι

$$pred(s, \boldsymbol{x}_1,...,\boldsymbol{x}_n) \Rightarrow pred(a(s, \mathbf{y}), \boldsymbol{x}_1,...,\boldsymbol{x}_n) \qquad (1)$$

για κάθε $s, \boldsymbol{x}_1,...,\boldsymbol{x}_n, \mathbf{y}$ όπου $\mathbf{y}$ είναι τα ορίσματα του τελεστή δράσης εκτός του s.
Συχνά δε μπορούμε να αποδείξουμε το παραπάνω γιατί η επαγωγική υπόθεση $pred(s, \boldsymbol{x}_1, ...,\boldsymbol{x}_n)$ είναι πολύ αδύναμη. Σ' αυτή την περίπτωση την ενδυναμώνουμε προσθέτοντας έναν τύπο, έστω *SIH* που ορίζεται ως εξής:

$pred(s, \boldsymbol{t}^1_1,...,\boldsymbol{t}^1_n) \wedge ... \wedge pred(s, \boldsymbol{t}^m_1,...,\boldsymbol{t}^m_n)$, όπου $\boldsymbol{t}^i_1,...,\boldsymbol{t}^i_n$ με $i = 1,...,m$ είναι μία λίστα όρων.

Τότε η (1) αντικαθίσταται με την εξής απόδειξη:
$(SIH \wedge pred(s, \boldsymbol{x}_1,...,\boldsymbol{x}_n)) \Rightarrow pred(a(s, \boldsymbol{y}), \boldsymbol{x}_1,...,\boldsymbol{x}_n)$.
Ο παραπάνω τύπος μπορεί να αποδεχθεί συνθέτοντας άλλες αποδείξεις. Αυτές έχουν τους ακόλουθους τύπους:
$(SIH \wedge pred(s, \boldsymbol{x}_1,...,\boldsymbol{x}_n)) \Rightarrow pred_1(a(s, \boldsymbol{y}), \boldsymbol{x}_1)$,

.
. (2)

.

$(SIH \wedge pred(s, \boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)) \Rightarrow pred_n(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_n)$.

Ή αλλιώς αρκεί να αποδείξουμε ότι:

$pred_1(s, \boldsymbol{x}_1) \Rightarrow pred_1(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_1)$,

.

. (3)

.

$pred_n(s, \boldsymbol{x}_n) \Rightarrow pred_n(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_n)$.

επειδή ο i-τύπος των εξισώσεων (2) μπορεί να εξαχθεί από τον i-τύπο των (3) με $1 \leq i \leq n$.

Ορισμένες εξισώσεις από τις (3) δεν μπορούν να αποδειχθεί ως έχουν γιατί οι επαγωγικές τους υποθέσεις είναι πολύ αδύναμες. Έστω ότι ο τύπος $pred_i(s, \boldsymbol{x}_i) \Rightarrow pred_i(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_i)$, όπου $1 \leq i \leq n$, είναι μία τέτοια επαγωγική υπόθεση. Υποθέτουμε ότι η $pred_j(s, \boldsymbol{u}_j)$ όπου $1 \leq j \leq n$ και $\boldsymbol{u}_j$ είναι $\boldsymbol{x}_j$, $\boldsymbol{t}^1_j$,…, ή $\boldsymbol{t}^m_j$ μπορούν να χρησιμοποιηθούν για να ενισχύσουν την επαγωγική υπόθεση $pred_i(s, \boldsymbol{x}_i)$ ώστε να αποδειχθεί. Η απόδειξη της εξίσωσης μπορεί να αντικατασταθεί με την απόδειξη αυτής:
$(pred_j(s, \boldsymbol{u}_j) \wedge pred_i(s, \boldsymbol{x}_i)) \Rightarrow pred_i(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_i)$,

γιατί ο i-τύπος των εξισώσεων (2) μπορεί να εξαχθεί από αυτή. Γενικά τα $pred_{j1}(s, \boldsymbol{u}_{j1}) \wedge \ldots \wedge pred_{jk}(s, \boldsymbol{u}_{jk})$, όπου $1 \leq j1 ,\ldots,jk \leq n$ και $\boldsymbol{u}_j$, $με\ j= j1,\ldots,jk$, είναι $\boldsymbol{x}_j$, $\boldsymbol{t}^1_j$,…, ή $\boldsymbol{t}^m_j$, ενισχύουν την επαγωγική υπόθεση. Έστω ότι η SIH_i είναι η εξίσωση που ενισχύει την επαγωγική υπόθεση $pred_i(s, \boldsymbol{x}_i)$. Τότε η απόδειξη του i- τύπου της (3) αντικαθίσταται με την παρακάτω απόδειξη:
$(SIH_i \wedge pred_i(s, \boldsymbol{x}_i)) \Rightarrow pred_i(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_i)$. (4)

Ακόμα έχουμε να σπάσουμε την υπόθεση σε υποπεριπτώσεις για να αποδείξουμε την (4). Υποθέτουμε ότι η υπόθεση σπάει σε l υποπεριπτώσεις. Οι l υποπεριπτώσεις δηλώνονται με l εξισώσεις $case_1^i$,….,$case_l^i$ που πρέπει να ικανοποιούν την παρακάτω:

$(case_1^i \vee \ldots \vee case_l^i) = \mathrm{true}$.

Τότε η απόδειξη της (4) αντικαθίσταται με τις αποδείξεις των ακόλουθων l εξισώσεων:

$(case_1^i \wedge SIH_i \wedge pred_i(s, \boldsymbol{x}_i)) \Rightarrow pred_i(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_i)$,

.

. (5)

.

$(case_l^i \wedge SIH_i \wedge pred_i(s, \boldsymbol{x}_i)) \Rightarrow pred_i(\mathrm{a}(s, \boldsymbol{y}), \boldsymbol{x}_i)$.

Η SIH_i μπορεί να μην είναι απαραίτητη για όλες τις υποπεριπτώσεις.
Από τα παραπάνω συμπεραίνουμε ότι οι n στο πλήθος αμετάβλητες καταστάσεις μπορούν να αποδειχθούν με σύνθετο τρόπο, ακόμα και αν εξαρτώνται η μια από την άλλη,

δεδομένου ότι η *i*-αμετάβλητη κατάσταση χρησιμοποιείται για να ενισχύσει την επαγωγική υπόθεση της j-αμετάβλητης κατάστασης και αντίστροφα. Η αρχική αμετάβλητη κατάσταση $pred_1(s, \boldsymbol{x}_1)$ που θέλουμε να αποδείξουμε μπορεί να διαιρείται σε άλλες επιμέρους αμετάβλητες καταστάσεις. Τα Proof Scores στην μέθοδο OTS/CafeOBJ βασίζονται στα παραπάνω, κυρίως στις εξισώσεις (5), και επομένως, μπορούμε να τα γράψουμε για κάθε μία από τις *n* αμετάβλητες καταστάσεις ξεχωριστά.

4.4.5 Proof Scores

Έστω ότι γράφουμε τα Proof Scores [47] για τις n αμετάβλητες καταστάσεις που περιγράφηκαν στην προηγούμενη ενότητα. Πρώτα γράφουμε ένα module, έστω `INV`, όπου η $pred_i(s, \boldsymbol{x}_i)$, $i = 1,\ldots,n$ εκφράζεται σε CafeOBJ ως εξής:

```
op inv1 : H V1 → Bool
...
op invn : H Vn → Bool

eq inv(W, X1 ) = pred1 (W, X1) .
...
eq invn (W, Xn ) = predn (W, Xn) .
```

όπου V_i ($i = 1,\ldots,n$) είναι η λίστα των visible sorts που αντιστοιχούν στα $\boldsymbol{x}_i$, W η μεταβλητή για το hidden sort H, X_i είναι η λίστα των μεταβλητών για τα V_i. Ο όρος $pred_i$ (W, X_i) ($i = 1,\ldots,n$) δηλώνει την $pred_1(s, \boldsymbol{x}_1)$.
Στο τμήμα, δηλώνουμε επίσης τις σταθερές X_i με $i = 1,\ldots,n$ για τα V_i. Στα Proof Scores, μία μεταβλητή που δεν περιορίζεται, χρησιμοποιείται για τη δήλωση ενός αυθαίρετου αντικειμένου ενός συγκεκριμένου τύπου. Για παράδειγμα, αν δηλώσουμε μια σταθερά *x* για ένα δεδομένο τύπου Nat τότε αυτός είναι ο ορατός τύπος για τους φυσικούς αριθμούς και το *x* δηλώνει έναν αυθαίρετο φυσικό αριθμό. Τέτοιες σταθερές περιορίζονται με εξισώσεις, οι οποίες μας δίνουν τη δυνατότητα να διαιρέσουμε το σύνολο καταστάσεων ή την υπόθεση. Υποθέτουμε ότι μια κατάσταση διαιρείται σε δύο μέρη: στη μία το *x* είναι ίσο με 0 και μια άλλη όπου το *x* είναι διάφορο και μεγαλύτερο από το 0. Τα παραπάνω εκφράζονται με τις εξής εξισώσεις:

```
eq x = 0 .

eq (x > 0)=true .
```

Θα περιγράψουμε κυρίως το Proof Score της i-αμετάβλητης κατάστασης. Έστω init η αρχική κατάσταση του συστήματος. Για να δείξουμε ότι η $pred_i(s, \boldsymbol{x}_i)$ ισχύει σε κάθε αρχική κατάσταση, σε κώδικα CafeOBJ γράφουμε:

```
open INV
 red inv_i (init, x_i ) .
close
```

Στη συνέχεια γράφουμε ένα τμήμα, έστω ISTEP, όπου δύο σταθερές s, s' δηλώνονται και δηλώνουν κάθε κατάσταση και κάθε επόμενη κατάσταση μετά την εφαρμογή του κανόνα μετάβασης στην κατάσταση. Τα κατηγορήματα προς απόδειξη σε κάθε επαγωγική περίπτωση, εκφράζονται σε CafeOBJ ως εξής:

```
op istep1 : V1 → Bool
...
op istepn : Vn → Bool

eq istep1 (X) = inv1 (s, X1 ) implies inv1 (s', X1) .
...
eq istepn (X) = invn(s,Xn) implies invn (s', Xn) .
```

Τα παραπάνω αντιστοιχούν στις εξισώσεις (3) της προηγούμενης ενότητας.
Κάθε επαγωγική υπόθεση συνήθως διαιρείται σε υποπεριπτώσεις με τα βασικά κατηγορήματα να δηλώνονται στις προδιαγραφές της CafeOBJ. Έστω ότι αποδεικνύουμε πως κάθε κανόνας μετάβασης που δηλώνεται στην CafeOBJ από μια τελεστή δράσης a διατηρεί την $pred_i(s, \boldsymbol{x}_i)$. Όπως παραπάνω, διαιρούμε σε l υποπεριπτώσεις $case_1^i$,….,$case_l^i$. Τότε, ο κώδικας σε CafeOBJ που δείχνει ότι ο κανόνας μετάβασης διατηρεί την $pred_i(s, \boldsymbol{x}_i)$ για την περίπτωση $case_j^i$ (j=1,...,l) είναι ο εξής:

```
open ISTEP
     Declare constants denoting arbitrary objects.
     Declare equations denoting case_j^i.
     Declare equations denoting facts if necessary.
     eq s' = a(s, y) .
     red istep_i(x_i) .
close
```

όπου y είναι η λίστα των σταθερών που χρησιμοποιούνται ως ορίσματα στον τελεστή δράσης a που δηλώνονται σε αυτόν τον κώδικα της CafeOBJ και δηλώνουν αυθαίρετα αντικείμενα για τους δηλωμένους τύπους. Αντίθετα με το y, άλλες σταθερές μπορεί να δηλωθούν στον κώδικα της CafeOBJ στη διαίρεση των περιπτώσεων. Οι εξισώσεις χρησιμοποιούνται για να εκφραστούν οι $case_j^i$. Αν χρειάζεται εξισώσεις που δηλώνουν καταστάσεις για δομές δεδομένων που έχουν χρησιμοποιηθεί μπορούν επίσης να δηλωθούν.
Αν το $istep_i(x_i)$ αναχθεί σε true, αποδεικνύεται ότι ο κανόνας μετάβασης διατηρεί το $pred_i(s, \boldsymbol{x}_i)$ στην υποπερίπτωση j, η οποία αντιστοιχεί στην απόδειξη της j-εξίσωσης της (5) από την προηγούμενη παράγραφο. Διαφορετικά, μπορούμε να κάνουμε πιο ισχυρή την επαγωγική υπόθεση με τον τρόπο που περιγράφαμε πάλι στην προηγούμενη ενότητα. Έστω ότι η SIH_i είναι η εξίσωση που ενισχύει την επαγωγική υπόθεση $pred_i(s, \boldsymbol{x}_i)$. Το $istep_i(x_i)$ ανάγεται στο παρακάτω:

(SIH_i and $inv_i(s, \boldsymbol{x}_i)$) implies $inv_i(s', \boldsymbol{x}_i)$
ή
SIH_i implies $istep_i(x_i)$.

Κεφάλαιο 5

5.1 Μοντελοποίηση Κοινωνικών Δικτύων με χρήση της CafeOBJ

Α) Μοντελοποίηση ενός χρήστη

Μέσω των τυπικών μεθόδων είναι δυνατό να περιγραφούν και να επαληθευτούν οι ιδιότητες των ταυτόχρονων διεργασιών που εκτελούνται στα κοινωνικά δίκτυα, καθώς και τα πρωτόκολλα λειτουργίας που μπορούν να εφαρμοστούν σε αυτά, όπως το πρωτόκολλο εμπιστοσύνης το οποίο έχουμε περιγράψει αναλυτικά στο Κεφάλαιο 3. Έτσι, παρουσιάζεται μια μοντελοποίηση των κοινωνικών δικτύων ως σύνθετα συμπεριφοριακά αντικείμενα ορισμένα μέσω της μεθόδου OTS/CafeOBJ.
Κάθε συμπεριφοριακό αντικείμενο του συστήματος μοντελοποιείται μέσω τελεστών, που ορίζουν τις δράσεις και τις παρατηρήσεις, και γράφουμε την προδιαγραφή του στην γλώσσα CafeOBJ.

Περιγραφή	Παρατηρητές	Κώδικας στην CafeOBJ
κάθε χρήστης έχει ένα μοναδικό χαρακτηριστικό userid	`user : Y -> D`	`bop user : User -> Userid`
ο χώρος μηνυμάτων	`inbox : Y -> D`	`bop inbox : User -> ListOfMessages`
η λίστα με τα userid των φίλων του χρήστη	`friends : Y -> D`	`bop friends : User -> ListOfFriends`

Όπου Y ο χώρος των καταστάσεων του αντικειμένου που αναπαριστά τον χρήστη στο κοινωνικό δίκτυο και D_1 κάποιος τύπος δεδομένων για τις τιμές που επιστρέφει ένα παρατηρητής.

Παρακάτω παρουσιάζουμε το κομμάτι του κώδικα, στο τμήμα με όνομα `USER`, που περιγράφει πως ένας χρήστης συνδέεται με τους φίλους του.

```
mod* USER {

*[User]*

pr(LIST)
pr(USERID)
pr(LIST1)


op init : -> User

bop user : User -> Userid
```

```
bop friends : User -> ListOfFriends

bop addFriend : User Userid -> User

bop deleteFriend : User Userid -> User

op c-addFriend : User Userid -> Bool

op c-deleteFriend : User Userid -> Bool

var U : User

var U1 : Userid

1     eq friends(init) = nilist .

2     eq c-addFriend(U, U1) = not (U1 in friends(U)) .

3     ceq friends(addFriend(U, U1)) = (U1 @ friends(U)) if c-
      addFriend(U, U1) .
4     eq c-deleteFriend(U, U1) = (U1 in friends(U)) .

5     ceq friends(deleteFriend(U, U1)) = (friends(U) out U1) if c-
      deleteFriend(U, U1) .

6     bceq addFriend(U, U1) = U if not c-addFriend(U, U1) .

7     bceq deleteFriend(U, U1) = U if not c-deleteFriend(U, U1) .

}
```

Ο χώρος καταστάσεων του χρήστη ορίζεται ως κρυμμένος τύπος `User`. Η αρχική κατάσταση είναι μια σταθερά `init`. Ο συμπεριφοριακός τελεστής παρατηρητής `friends` επιστρέφει μια λίστα με όλους τους φίλους του χρήστη. Οι συναρτήσεις μετάβασης (transitions), που είναι υπεύθυνες για την αλλαγή μιας κατάστασης του συστήματος, είναι οι `addfriend` και `deletefriend` οι οποίες συμβολίζουν την αλλαγή μιας σχέσης φιλίας μεταξύ δύο χρηστών. Συγκεκριμένα, με την πρώτη ο χρήστης προσθέτει στους φίλους του έναν άλλο χρήστη και με την άλλη διαγράφει κάποιον από τους ήδη υπάρχοντες. Η `c-addfriend` είναι η αποτελεσματική συνθήκη (effective condition) της μετάβασης `addfriend` και αντίστοιχα η `c-deletefriend` της `deletefriend`.

Τα αποτελέσματα των μεταβάσεων ορίζονται από τις εξισώσεις 1 έως 7.
Αρχικά, θεωρούμε ότι ο χρήστης δεν έχει φίλους, εξίσωση 1, ή αλλιώς ότι ο παρατηρητής `friends` επιστρέφει μια κενή λίστα. Οι εξισώσεις 2 και 4 ορίζουν τις συνθήκες που χρειάζονται για να εφαρμοστούν οι αντίστοιχες συναρτήσεις μετάβασης μέσω των εξισώσεων 3 και 5. Έτσι, στην εξίσωση 2 δηλώνεται ότι ο χρήστης U για να προσθέσει τον

χρήστη με `userid` U1 στους φίλους του θα πρέπει ο χρήστης με `userid` U1 να μην ανήκει ήδη σε αυτούς. Όταν αυτό συμβαίνει μπορεί να εφαρμοστεί η εξίσωση 3 και ο χρήστης με `userid` U1 προστίθεται στη λίστα των φίλων του χρήστη U. Όμοια, για να διαγραφεί κάποιος φίλος του χρήστη U θα πρέπει να υπάρχει στη λίστα των φίλων του, όπως φαίνεται στην εξίσωση 4. Οι τελευταίες δύο εξισώσεις δηλώνουν ότι η κατάσταση του συστήματος παραμένει αμετάβλητη εφόσον η συνθήκη μετάβασης εφαρμοστεί αλλά δεν ισχύει η αποτελεσματική της συνθήκη.

χώρος καταστάσεων: κρυμμένος τύπος User
αρχική κατάσταση: σταθερά init
παρατηρητής: friends
αλλαγή καταστάσεων μέσω των συναρτήσεων μετάβασης: addFriend, deleteFriend
αποτελεσματικές συνθήκες: c-addFriend, c-deleteFriend

Η λίστα των φίλων ως τύπος δεδομένων μοντελοποιήθηκε με την παρακάτω προδιαγραφή:

```
mod! LIST  (X :: TRIV) {

[Elt < List]

op nil : -> List

op _in_ : Elt List -> Bool

op _@_ : Elt List -> List

op _out_ : List Elt -> List

var L : List

var E : Elt

eq (E in nil) = false .

eq (E @ nil) = E .

ceq (E @ L) = L if (E in L) .

eq (nil out E) = nil .

eq (E in (L out E)) = false .

}
```

Οι βασικοί τελεστές είναι οι _in_, _@_ και _out_. Ο _in_ ως τελεστής παρατήρησης επιστρέφει αν ένα στοιχείο υπάρχει σε αυτή τη λίστα και οι _@_ και _out_ ως τελεστές δράσης προσθέτουν ή αφαιρούν αντίστοιχα ένα στοιχείο από αυτήν.

Κάθε αντικείμενο-χρήστη του συστήματος έχει και το δικό του τμήμα προδιαγραφής το οποίο περιέχει και το μοναδικό αναγνωριστικό Userid. Αυτό είναι σαν ένας μετρητής που μας δίνει τη θέση του στο συνολικό σύστημα της σύνθεσης όλων των User.

```
mod! USERID {

[Userid]

}
```

Σε αυτό το αντικείμενο (user) υπάρχουν οι εξής δύο μέθοδοι οι οποίες ορίζουν:

- τους φίλους τους χρήστη
- τα μηνύματα που λαμβάνει και στέλνει από και προς το δίκτυο.

Έχοντας περιγράψει την πρώτη μέθοδο, παρακάτω παρουσιάζουμε το κομμάτι του κώδικα, στο ίδιο φυσικά τμήμα με όνομα USER, που περιγράφει τη διαδικασία ανταλλαγής μηνυμάτων στο επίπεδο του χρήστη, δηλαδή τη δεύτερη μέθοδο.

```
mod* USER {

*[User]*

pr(LIST)
pr(USERID)
pr(MESSAGE)
pr(LIST2)

op init : -> User

bop user : User -> Userid
bop inbox : User -> ListOfMessages

bop receive : User Message -> User
bop send : User Message -> User

bop received : User Message -> Bool
bop sent : User Message -> Bool

var U : User
vars M1, M2 : Message

1     eq inbox(init) = nilist .
```

```
      eq received(init, M1) = false .

      eq sent(init, M1) = false .


2     ceq received(receive(U, M1) , M2) = true if (M1 = M2) .

3     eq inbox(receive(U, M1) = M1 @ inbox(U) .

4     ceq sent(send(U, M1) , M2) = true if (M1 = M2).

5     eq inbox(send(U, M1) = inbox(U) out M1 .

}
```

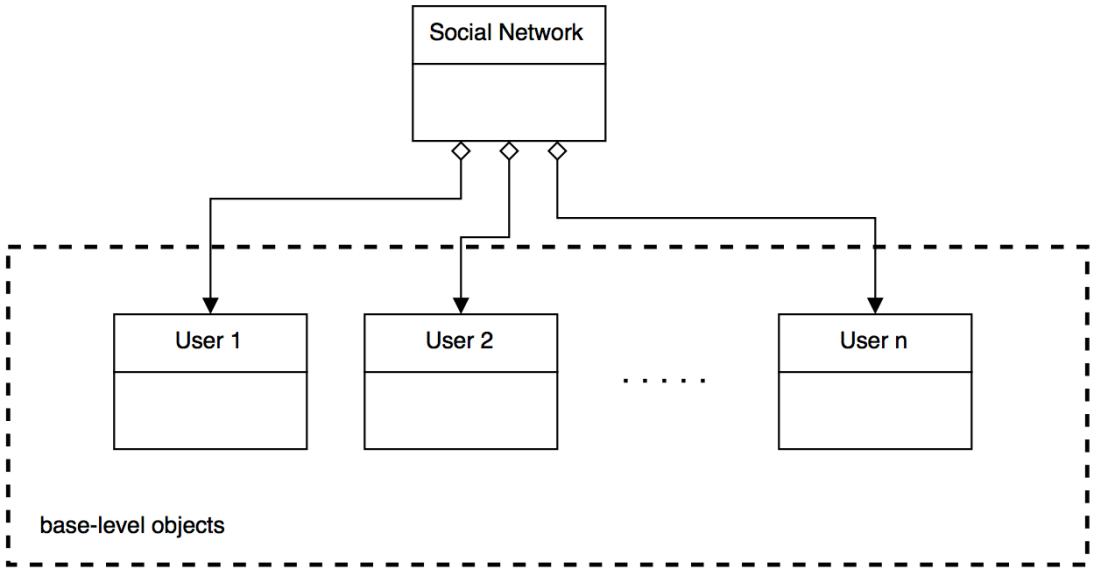
Όπως και πριν, ο χώρος καταστάσεων του χρήστη ορίζεται ως κρυμμένος τύπος `User` και η αρχική κατάσταση είναι μια σταθερά `init`. Για κάθε παρατηρητή δηλώνεται και η αντίστοιχη αρχική του κατάσταση. Ο συμπεριφοριακός τελεστής παρατηρητής `inbox` επιστρέφει μια λίστα με τα μηνύματα του χρήστη. Οι συναρτήσεις μετάβασης είναι οι `receive` και `send` οι οποίες συμβολίζουν ότι ένα μήνυμα ελήφθη στο χρήστη ή στέλνεται από τον ίδιο αντίστοιχα. Η `received` είναι η αποτελεσματική συνθήκη της μετάβασης `receive` και αντίστοιχα η `sent` της `send`. Τα μηνύματα ορίζονται μέσω του τύπου `Message`.

```
mod! MESSAGE  {
[Message]
}
```

Τα αποτελέσματα των μεταβάσεων ορίζονται από τις εξισώσεις 1 έως 5.
Αρχικά, θεωρούμε ότι ο χρήστης δεν έχει μηνύματα, εξίσωση 1, ή αλλιώς ότι ο παρατηρητής `inbox` επιστρέφει μια κενή λίστα. Ομοίως οι παρατηρητές `received` και `sent` είναι `false`. Η εξίσωση 2 δηλώνει τη συνθήκη εκείνη που ο χρήστης U μπορεί να λάβει ένα μήνυμα M1. Τότε, με την εξίσωση 3 φαίνεται ότι ο χρήστης έχει στο `inbox` του το μήνυμα M1. Όμοια, όταν ο χρήστης U στέλνει ένα μήνυμα αυτό αφαιρείται από το `inbox` του, όπως φαίνεται στην εξίσωση 5.

Β) Μοντελοποίηση του Κοινωνικού Δικτύου

Κάνοντας σύνθεση των αντικειμένων-χρηστών που ορίσαμε παραπάνω μπορούμε να ορίσουμε το συνολικό σύστημα που θα μοντελοποιεί το κοινωνικό δίκτυο που επιθυμούμε.



Εικόνα 12: Σύνθεση συμπεριφοριακών αντικειμένων προσαρμοσμένη σε ένα κοινωνικό δίκτυο

Ο τελεστής προβολής που χρησιμοποιήθηκε είναι ο εξής:

Περιγραφή	Παρατηρητές ΠΣΜ	Κώδικας στην CafeOBJ
τελεστής προβολής από την κατάσταση του κοινωνικού δικτύου στην κατάσταση του χρήστη	`usersProfile : Y' D Y`	`bop usersProfile : Sn Userid -> User`

Στο παρακάτω κομμάτι του τμήματος `SOCIALNETWORK` φαίνεται πως στο δίκτυο αυτό μπορεί να προστεθεί και διαγραφεί ένας χρήστης.

```
mod* SOCIALNETWORK {

pr(LIST3)
pr(USER)

*[Sn]*

op init : -> Sn

bop usersProfile : Sn Userid -> User

bop userids : Sn -> ListofUserids
```

```
bop addUser : Sn Userid -> Sn
op c-addUser : Sn Userid -> Bool

bop deleteUser : Sn Userid -> Sn
op c-deleteUser : Sn Userid -> Bool


vars U1 U2 : Userid
var S : Sn

eq usersProfile(init, U1) = init .

eq userids(init) = nilist .


1     eq c-addUser(S, U1)= not (U1 in userids(S)) .

2     ceq userids(adduser(S, U1)) = U1 @ userids(S) if c-addUser(S,
      U1) .

3     bceq addUser(S, U1) = S if not c-addUser(S, U1) .

4     eq c-deleteUser(S, U1) = (U1 in userids(S)) .

5     ceq userids(deleteUser(S, U1)) = U1 out userids(S) if c-
addUser(S, U1) .

6     bceq deleteUser(S, U1) = S if not c-deleteUser(S, U1) .
```

Ο χώρος των καταστάσεων του κοινωνικού δικτύου ορίζεται με τον κρυμμένο τύπο `Sn`. Ο τελεστής `init` δηλώνει την αρχική κατάσταση του συστήματος. Ο τελεστής προβολής (projection) `usersProfile` επιστρέφει την παρατήρηση της κατάστασης του χρήστη όπως έχει δηλωθεί στο τμήμα `USER`. Ο τελεστής παρατήρησης `userids` επιστρέφει μία λίστα με όλα τα `Userid` των χρηστών του κοινωνικού δικτύου. Οι συναρτήσεις μετάβασης `addUser` και `deleteUser` συμβολίζουν εάν προστέθηκε ένας χρήστης ή αντίστοιχα διαγράφτηκε από το δίκτυο. Η `c-addUser` είναι η αποτελεσματική συνθήκη της μετάβασης `addUser` και αντίστοιχα η `c-deleteUser` της `deleteUser`.

Οι εξισώσεις 1-6 δείχνουν πως εφαρμόζονται οι παραπάνω συναρτήσεις των τελεστών. Έτσι, και στην περίπτωση που προστίθενται ένας χρήστης στο κοινωνικό δίκτυο, αλλά και σε αυτή που πρέπει να διαγραφεί, ελέγχεται πρώτα αν υπάρχει ή όχι σε αυτό. Στη συνέχεια ορίζονται οι αποτελεσματικές συνθήκες (εξ. 2 και 5) καθώς και τι συμβαίνει όταν αυτές δεν ισχύουν (εξ. 3 και 6).

Αν δύο χρήστες γίνουν φίλοι αυτό στην CafeOBJ ορίζεται ως εξής:

```
mod* SOCIALNETWORK {
```

```
pr(LIST3)
pr(USER)

*[Sn]*
.
.
.
bop addFriend2 : Sn Userid Userid -> Sn

op c-addFriend2 : Sn Userid Userid -> Bool

eq c-addFriend2(S, U1, U2)=
not (U1 in friends(usersProfile(S, U2))) and (U1 in userids(S)) and
(U2 in userids(S)) .

bceq addFriend2(S, U1, U2) = S if not addFriend2(S, U1, U2) .

}
```

Γ) Μοντελοποίηση του Πρωτόκολλου Εμπιστοσύνης

Μέσα από στη μοντελοποίηση του κοινωνικού δικτύου έχουμε ορίσει και τη λειτουργία του πρωτόκολλου που περιγράφτηκε στο Κεφάλαιο 3.

Στο πρωτόκολλο υπάρχουν 2 ειδών μεταβλητές τύπου μηνύματος ALARM:

- ALARM1
- ALARM2

τα οποία είναι διαφορετικά μεταξύ τους και ορίζονται αρχικά στο τμήμα με όνομα `ALARM`.

```
mod* ALARM  {

[Alarm < Message]

op alarm1 : -> Alarm

op alarm2 : -> Alarm

op _=_ : Alarm  Alarm  -> Bool

var A : Alarm

eq (A = A) = true .
```

```
eq (alarm1 = alarm2) = false .

}
```

Αυτά μοντελοποιούνται ως εξής:

```
mod* SOCIALNETWORK {

pr(ALARM)
pr(LIST5)
pr(LIST6)
pr(LIST7)
pr(LIST8)


*[Sn]*
.
.
.

bop userokUser : Sn -> ListofuserokUsers

bop isolatedUsers : Sn -> ListofIsolatedUsers

bop friendsofrealUser : Sn -> ListofFriendsofrealUsers

bop friendsofsuspiciousUser : Sn -> ListofFriendsofsuspiciousUsers

bop alarm1 : Alarm1 Sn Userid-> Sn

op c-alarm1 : Accountid  Sn -> Bool

bop alarm2 : Alarm2 Sn ListofFriendsofsuspiciousUsers
ListofFriendsofrealUsers -> Sn

op c-alarm2 : Accountid  Sn Userid Userid -> Bool


var A1 : Alarm1

var A2 : Alarm2




eq c-alarm1(A1, S, U1) = (U1 in userids(S)) and (U1 in
isolatedUsers(S)).

ceq inbox2 (alarm1(A1, S, U1)) = A1 @ inbox(usersProfile(S, U1)) if
c-alarm1(A1, S, U1) .

bceq alarm1(A1, S, U1) = S if not c-alarm1(A1, S, U1) .
```

```
eq c-alarm2(A2, S, U1, U2) = U1 in userids(S) and U2 in userids(S)
and U1 in ListofFriendsofsuspiciousUsers(S) or U2 in
ListofFriendsofrealUsers(S) .

ceq inbox2 (alarm2(A2, S, U1, U2)) =  A2 @ inbox(U1) and A2 @
inbox(U2) if c-alarm2(A2, S, U1, U2)) .

bceq alarm2(A2, S, U1, U2) = S if not c-alarm2(A2, S, U1, U2) .
```

Οι τελεστές παρατήρησης `userokUser`, `isolatedUsers`, `friendsofrealUser`, και `friendsofsuspiciousUser` επιστρέφουν λίστες με όλα τα `Userid` των χρηστών του κοινωνικού δικτύου που έχουν τις ανάλογες ιδιότητες. Οι συναρτήσεις μετάβασης `alarm1` και `alarm2` δηλώνουν τις αλλαγές από την αρχική κατάσταση του δίκτυο και έχουν ως αποτελεσματικές συνθήκες τις `c-alarm1` και `c-alarm2`.

Παρατηρούμε ότι κάθε εφαρμογή μίας συνάρτησης μετάβασης αλλάζει την κατάσταση του συνολικού συστήματος καθώς και τις τιμές των παρατηρητών.

5.2 Προτάσεις

Παραπάνω έγινε μια προσπάθεια μοντελοποίησης ενός κοινωνικού δικτύου προκειμένου να ορισθεί η λειτουργία του μέσω ενός πρωτόκολλου βασισμένο στη συνεργασία των χρηστών του. Σκοπός ήταν το κοινωνικό δίκτυο να είναι τελικά ένα δίκτυο εμπιστοσύνης για το σύνολο των χρηστών του.

Στη συνέχεια αυτής της προσπάθειας είναι σημαντικό να γίνει έλεγχος της παραπάνω μοντελοποίησης ώστε να επαληθευτεί η σωστή λειτουργία της. Πιο συγκεκριμένα, πρέπει να γίνει επαλήθευση ορισμένων ιδιοτήτων του προτεινόμενου πρωτόκολλου μέσω μιας διαδικασίας έγκυρων αποδείξεων. Μία τέτοια διαδικασία είναι εφικτή μέσω της χρήσης της μεθόδου των Proof Scores, το θεωρητικό κομμάτι της οποίας έχουμε περιγράψει στο Κεφάλαιο 4. Μέσω τέτοιων διαδικασιών είναι δυνατόν να υπάρξει μία σειρά από πιο αυτοματοποιημένες αποδείξεις, που όπως εύκολα μπορεί κανείς να καταλάβει, είναι ιδιαίτερα χρήσιμες για τα πολύπλοκα συστήματα όπως αυτά που χρησιμοποιούμε για να περιγράψουμε τα κοινωνικά δίκτυα.

Ακόμα, περαιτέρω διερεύνηση άλλων τεχνικών επαλήθευσης θα μπορούσε να αποτελέσει έναυσμα για απόδειξη πιο σύνθετων ιδιοτήτων που μπορεί να υπάρχουν μέσα σε ένα κοινωνικό δίκτυο. Οι ιδιότητες αυτές εξαρτώνται κάθε φορά από το κοινωνικό δίκτυο στο οποίο γίνεται η μοντελοποίηση και δεν έχουν τη γενική μορφή όπως αυτές που περιγράφτηκαν στην παρούσα διπλωματική εργασία.

Παράρτημα

Κώδικας

```
mod! LIST  (X :: TRIV) {

[Elt < List]

op nil : -> List

op _in_ : Elt List -> Bool

op _@_ : Elt List -> List

op _out_ : List Elt -> List

var L : List

var E : Elt

eq (E in nil) = false .

eq (E @ nil) = E .

ceq (E @ L) = L if ( E in L ) .

eq (nil out E) = nil .

eq (E in (L out E)) = false .

}

mod! USERID  {

[Userid]

}

mod* LIST1 {

pr (LIST(USERID { sort Elt -> Userid } ) * { sort List ->
ListOfFriends , op nil -> nilist } )

}

mod! MESSAGE  {

[Message]

}
```

```
mod* LIST2 {

pr (LIST (MESSAGE { sort Elt -> Message} ) * { sort List ->
ListOfMessages, op nil -> nilist } )

}

mod* ALARM  {

[Alarm < Message]

op alarm1 : -> Alarm

op alarm2 : -> Alarm


op _=_ : Alarm  Alarm  -> Bool


var A : Alarm

eq (A = A) = true .

eq (alarm1 = alarm2) = false .


}


mod* LIST3 {

pr (LIST(USERID { sort Elt -> Userid } ) * { sort List ->
ListOfUserids , op nil -> nilist } )

}


mod* LIST4 {

pr (LIST (MESSAGE { sort Elt -> Message} ) * { sort List ->
ListOfMessages2, op nil -> nilist } )

}

mod* LIST5 {

pr (LIST(USERID { sort Elt -> Userid } ) * { sort List ->
ListofuserokUsers , op nil -> nilist } )


}
```

```
mod* LIST6 {

pr (LIST(USERID { sort Elt -> Userid } ) * { sort List ->
ListofIsolatedUsers , op nil -> nilist } )


}


mod* LIST7 {

pr (LIST(USERID { sort Elt -> Userid } ) * { sort List ->
ListofFriendsofrealUsers , op nil -> nilist } )


}



mod* LIST8 {

pr (LIST (MESSAGE { sort Elt -> Message} ) * { sort List ->
ListofFriendsofsuspiciousUsers, op nil -> nilist } )

}
```

```
mod* USER {

*[User]*

pr(LIST)
pr(USERID)
pr(LIST1)
pr(MESSAGE)
pr(LIST2)



--operators

op init : -> User

bop user : User -> Userid


bop friends : User -> ListOfFriends

bop inbox : User -> ListOfMessages


bop addFriend : User Userid -> User

bop deleteFriend : User Userid -> User


bop receive : User Message -> User

bop send : User Message -> User


op c-addFriend : User Userid -> Bool

op c-deleteFriend : User Userid -> Bool


bop received : User Message -> Bool

bop sent : User Message -> Bool


--variables

var U : User

var U1 : Userid

vars M1, M2 : Message
```

```
--equations

eq friends(init) = nilist .

eq inbox(init) = nilist .

eq received(init, M1) = false .

eq sent(init, M1) = false .


eq c-addFriend(U, U1) = not (U1 in friends(U)) .

ceq friends(addFriend(U, U1)) = (U1 @ friends(U)) if c-addFriend(U,
U1) .

eq inbox(addFriend(U, U1)) = inbox(U) .

eq received(addFriend(U, U1), M1) = received(U, M1) .

eq sent(addFriend(U, U1), M1) = sent(U, M1) .

bceq addFriend(U, U1) = U if not c-addFriend(U, U1) .


eq c-deleteFriend(U, U1) = (U1 in friends(U)) .

ceq friends(deleteFriend(U, U1)) = (friends(U) out U1) if c-
deleteFriend(U, U1) .

eq inbox(deleteFriend(U, U1)) = inbox(U) .

eq received(deleteFriend(U, U1), M1) = false .

eq sent(deleteFriend(U, U1), M1) = false .

bceq deleteFriend(U, U1) = U if not c-deleteFriend(U, U1) .


ceq received(receive(U, M1) , M2) = true if (M1 = M2) .

eq inbox(receive(U, M1) = M1 @ inbox(U) .

eq friends(receive(U, M1)) = friends(U) .


ceq sent(send(U, M1) , M2) = true if (M1 = M2) .

eq inbox(send(U, M1) = inbox(U) out M1 .

eq friends(send(U, M1)) = friends(U) .

}
```

```
mod* SOCIALNETWORK {

pr(LIST3)
pr(USER)
pr(LIST4)
pr(ALARM)
pr(LIST5)
pr(LIST6)
pr(LIST7)
pr(LIST8)


*[Sn]*

--operators

op init2 : -> Sn

bop usersProfile : Sn Userid -> User


bop userids : Sn -> ListofUserids

bop inbox2 : Sn -> ListOfMessages2


bop userokUser : Sn -> ListofuserokUsers

bop isolatedUsers : Sn -> ListofIsolatedUsers

bop friendsofrealUser : Sn -> ListofFriendsofrealUsers

bop friendsofsuspiciousUser : Sn -> ListofFriendsofsuspiciousUsers


bop addUser : Sn Userid -> Sn

op c-addUser : Sn Userid -> Bool


bop deleteUser : Sn Userid -> Sn

op c-deleteUser : Sn Userid -> Bool


bop addFriend2 : Sn Userid Userid -> Sn

op c-addFriend2 : Sn Userid Userid -> Bool


bop sendtoUser : Sn Userid Message -> Sn

op c-sendtoUser : Sn Userid Message -> Bool
```

```
bop receivefromUser : Sn Userid Message -> Sn

op c-receivefromUser : Sn Userid Message -> Bool


bop alarm1 : Alarm1 Sn Userid -> Sn

op c-alarm1 : Sn Userid -> Bool


bop alarm2 : Alarm2 Sn ListofFriendsofsuspiciousUsers
ListofFriendsofrealUsers -> Sn

op c-alarm2 : Sn Userid Userid -> Bool


--variables

vars U1 U2 U3 : Userid

var S : Sn

var M : Message

var A1 : Alarm1

var A2 : Alarm2


--equations for initial states of observers and projections


eq usersProfile(init2, U1) = init2 .

eq userids(init2) = nilist .

eq inbox2(init2) = nilist .

eq userokUser(init2) = nilist .

eq isolatedUsers(init2) = nilist .

eq friendsofrealUser(init2) = nilist .

eq friendsofsuspiciousUser(init2) = nilist .
```

```
eq c-addUser(S, U1)= not (U1 in userids(S)) .

ceq userids(addUser(S, U1))= U1 @ userids(S) if c-addUser(S, U1) .


ceq usersProfile(U1 , addUser(S, U2)) =  usersProfile(S, U1) if (U1
= U2) and c-addUser(S, U2) .

eq inbox2(addUser(S, U1)) = inbox2(S).

eq userokUser(addUser(S, U1)) = userokUser(S)  .

eq isolatedUsers(addUser(S, U1)) = isolatedUsers(S)  .

eq friendsofrealUser(addUser(S, U1)) = friendsofrealUser(S)  .

eq friendsofsuspiciousUser(addUser(S, U1)) =
friendsofsuspiciousUser(S)  .

bceq addUser(S, U1) = S if not c-addUser(S, U1) .



eq c-deleteUser(S, U1) = (U1 in userids(S)) .

ceq userids(deleteUser(S, U1)) = U1 out userids(S) if c-addUser(S,
U1) .


ceq usersProfile(U1 , deleteUser(S, U2)) =  usersProfile(S, U1) if
(U1 = U2) and c-deleteUser(S, U2) .


eq inbox2(deleteUser(S, U1)) = inbox2(S).

eq userokUser(deleteUser(S, U1)) = userokUser(S)  .

eq isolatedUsers(deleteUser(S, U1)) = isolatedUsers(S)  .

eq friendsofrealUser(deleteUser(S, U1)) = friendsofrealUser(S)  .

eq friendsofsuspiciousUser(deleteUser(S, U1)) =
friendsofsuspiciousUser(S)  .

bceq deleteUser(S, U1)= S if not c-deleteUser(S, U1) .



eq c-addFriend2(S, U1, U2) =  not (U1 in friends(usersProfile(S,
U2))) and (U1 in userids(S)) and (U2 in userids(S)) .
```

```
ceq userids(addFriend2(S, U1, U2)) = userids(S) if c-addFriend2(S,
U1, U2) .

ceq usersProfile(U1, addFriend2(S, U2, U3) ) = usersProfile(S, U1)
if c-addFriend2(S, U2, U3) .

eq inbox2(addFriend2(S, U1, U2)) = inbox2(S).

eq userokUser(addFriend2(S, U1, U2)) = userokUser(S)  .

eq isolatedUsers(addFriend2(S, U1, U2)) = isolatedUsers(S)  .

eq friendsofrealUser(addFriend2(S, U1, U2)) = friendsofrealUser(S)
.

eq friendsofsuspiciousUser(addFriend2(S, U1, U2)) =
friendsofsuspiciousUser(S)  .

bceq addFriend2(S, U1, U2) = S if not addFriend2(S, U1, U2) .



eq c-sendtoUser(S, U1, M)= true if ((U1 in userids(S) and (M in
inbox(U) ).

eq userids(sendtoUser(S, U1, M)) = userids(S).

ceq usersprofile(sendtoUser(S, U1, M), U1) =
sendtoUser(usersprofile(S, U1), M) if c-sendtoUser(S, U1, M) .

eq inbox2(sendtoUser(S, U1, M)) = inbox2(S) out M .

eq userokUser(sendtoUser(S, U1, M)) = userokUser(S)  .

eq isolatedUsers(sendtoUser(S, U1, M)) = isolatedUsers(S)  .

eq friendsofrealUser(sendtoUser(S, U1, M)) =
friendsofrealUser(S)  .

eq friendsofsuspiciousUser(sendtoUser(S, U1, M)) =
friendsofsuspiciousUser(S)  .


bceq sendtoUser(S, U1, M) = S if not c-sendtoUser(S, U1, M) .



eq c-receivefromUser(S, U1, M)= true if ((U1 in userids(S) and  (M
out inbox(U) ).

eq userids(receivefromUser(S, U1, M)) = userids(S).
```

```
ceq usersprofile(receivefromUser(S, U1, M), U1) =
receivefromUser(usersprofile(S, U1), M) if c-receivefromUser(S, U1,
M) .

ceq inbox2(receivefromUser(S, U1, M)) = M @ inbox2(S) if c-
receivefromUser(S, U1, M).

eq userokUser(receivefromUser(S, U1, M)) = userokUser(S)  .

eq isolatedUsers(receivefromUser(S, U1, M)) = isolatedUsers(S)  .

eq friendsofrealUser(receivefromUser(S, U1, M)) =
friendsofrealUser(S)  .

eq friendsofsuspiciousUser(receivefromUser(S, U1, M)) =
friendsofsuspiciousUser(S)  .


bceq receivefromUser(S, U1, M) = S if not c-receivefromUser(S, U1,
M) .



eq c-alarm1(A1, S, U1) = ( U1 in userids(S) ) and (U1 in
isolatedUsers(S)) .

ceq inbox2 (alarm1(A1, S, U1)) = A1 @ inbox(usersProfile(S, U1)) if
c-alarm1(A1, S, U1) .

eq userids(alarm1(A1, S, U1)) = userids(S) .

eq usersProfile(U1, alarm1(A1, S, U1) ) = usersProfile(S, U1) .

eq userokUser(alarm1(A1, S, U1)) = userokUser(S)  .

eq isolatedUsers(alarm1(A1, S, U1)) = isolatedUsers(S)  .

eq friendsofrealUser(alarm1(A1, S, U1)) = friendsofrealUser(S)  .

eq friendsofsuspiciousUser(alarm1(A1, S, U1)) =
friendsofsuspiciousUser(S)  .


bceq alarm1(A1, S, U1) = S if not c-alarm1(A1, S, U1) .



eq c-alarm2(A2, S, U1, U2) = ( (U1 in userids(S) ) and (U2 in
userids(S) ) ) and ( (U1 in ListofFriendsofsuspiciousUsers(S) ) or
(U2 in ListofFriendsofrealUsers(S) ) ) .
```

```
ceq inbox2 (alarm2(A2, S, U1, U2)) = ( A2 @ inbox(usersProfile(S,
U1)) and A2 @ inbox(usersProfile(S, U2)) ) if c-alarm2(A2, S, U1,
U2) .

eq userids(alarm2(A2, S, U1, U2)) = userids(S) .

eq usersProfile(U1, alarm2(A2, S, U1, U2) ) = usersProfile(S, U1).

eq userokUser(alarm2(A2, S, U1, U2)) = userokUser(S)  .

eq isolatedUsers(alarm2(A2, S, U1, U2)) = isolatedUsers(S)  .

eq friendsofrealUser(alarm1(A1, S, U1)) = friendsofrealUser(S)  .

eq friendsofsuspiciousUser(alarm1(A1, S, U1)) =
friendsofsuspiciousUser(S)  .


bceq alarm2(A2, S, U1, U2) = S if not c-alarm2(A2, S, U1, U2) .


}
```

Βιβλιογραφία

[1] Tim Berners-Lee, et al, “World-Wide Web: Information Universe”, Electronic: Research, Applications and Policy, April 1992.

[2] The World Wide Web Organisation. World Wide Web home page, http://www.w3.org

[3] Stewart W. (2000), “Mosaic - The First Global Web Browser”, The Living Internet, 2000, at http://www.livinginternet.com/w/wi_mosaic.htm

[4] O’Reilly, T. (2007), “What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. Communications & Strategies”, http://ssrn.com/abstract=1008839

[5] Boyd D., Ellison N. B. (2007), “Social network sites: Definition, history, and scholarship”, Journal of Computer-Mediated Communication, http://www.danah.org/papers/JCMCIntro.pdf

[6] Hanneman, R., Riddle, M. (2005), “Introduction to social network methods”, Riverside, CA: University of California, Riverside, http://faculty.ucr.edu/~hanneman/

[7] Classmates.com. http://www.classmates.com

[8] SixDegrees.com. http://www.sixdegrees.com

[9] Ahmad A. (2011), “A Short Description of Social Networking Websites And Its Uses”, Department of Computer Science & Engineering, Singhania University

[10] Friendster, www.friendster.com

[11] CyWorld, http://www.cyworld.com

[12] Ryze, http://www.ryze.com

[13] LinkedIn, http://www.linkedin.com

[14] MySpace, http://www.myspace.com

[15] Flickr, http://www.flickr.com

[16] YouTube, http://www.youtube.com

[17] Zoomr, http://www.zoomr.com

[18] LiveJournal, http://www.livejournal.com

[19] BlogSpot, http://www.blogspot.com

[20] Digg, http://www.digg.com

[21] Reddit, http://www.reddit.com

[22] del.icio.us, http://del.icio.us

[23] Facebook, http://www.facebook.com

[24] Twitter, www.twitter.com

[25] Google+, https://plus.google.com

[26] Cashia R. (2008), “Social Computing: Study of the Use and Impact of Online Social Networking”, EC JRC IPTS, Sevilla

[27] Within3, http://www.within3.com

[28] Care2, http://www.care2.com

[29] Ello, https://www.ello.co/

[30] Golbeck J., Hendler J., “Inferring Trust Relationships in Web-based Social Networks”

[31] Marsh S. (1994), “Formalising Trust as a Computational Concept”, Ph.D. Thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, United Kingdom

[32] Deutsch M. (1962), “Cooperation and Trust: Some Theoretical Notes”, Proceedings of the Nebraska Symposium on Motivation, Nebraska University Press, USA

[33] Dellarocas C. (2002), “The Design of Reliable Trust Management Systems for Electronic Trading Communities”, Sloan School of Management, Massachusetts Institute of Technology

[34] http://www.insna.org

[35] Tanenbaum A. S. [1996] (2000), "Δίκτυα Υπολογιστών", Τρίτη Έκδοση, Πρώτη Ελληνική Έκδοση, Εκδόσεις Παπασωτηρίου, ISBN 960-7510-70-4

[36] National Institute of Standards and Technology (NIST), “Wireless Ad Hoc Network Projects”, http://www.antd.nist.gov/wahn_bkgnd.shtml

[37] Johnson D. B., Maltz D. A., & Broch J. (2001), "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", in Ad Hoc Networking, Addison-Wesley

[38] Buchegger S., Le Boudec J.-Y. (2002), “Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks)”, MobiHoc

[39] Formal Methods Europe, website: www.fmeurope.org

[40] Spivey J. M. (1988), ‘The Z notation: A reference manual’

[41] http://www.sdl-forum.org

[42] CafeOBJ: CafeOBJ web page, http://www.ldl.jaist.ac.jp/cafeobj/

[43] Nakagawa A., Sawada T., Futatsugi K. (1999), “CafeOBJ User's Manual”, http://www.ldl.jaist.ac.jp/cafeobj/doc/

[44] Diaconescu R., Futatsugi K. (1998), “CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification”, Volume 6 of AMAST Series in Computing, World Scientific

[45] Meseguer J. (1992), “Conditional rewriting logic as a unified model of concurrency”, Theoretical Computer Science, 96(1):73–155

[46] Burstall R., Goguen J. (1980), “The semantics of Clear, a specification language”, Springer, Lecture Notes in Computer Science, Volume 86

[47] Ogata K., Futatsugi K., “Proof Scores in the OTS/CafeOBJ method”

[48] Diaconescu R., Futatsugi K., Iida S. (2000), “CafeOBJ Jewels”, In Kokichi Futatsugi, Ataru Nakagawa, and Tetsuo Tamai, editors, Cafe: An Industrial-Strength Algebraic Formal Method, Elsevier

[49] Ogata K. (JAIST), “Introduction to Specification & Verification in CafeOBJ Observational Transition Systems”, A mini course in the Graduate Program of the Univ. of Athens, Athens, Greece 2009

[50] Iida S., Matsumoto M., Diaconescu R, Futatsugi K, & Lucanu D. (1998), “Concurrent object composition in CafeOBJ”, Technical Report IS-RR-98-0009S, Japan Advanced Institute for Science and Technology

[51] http://www.theta.ro/cafeobj/cube.html