ΒΑΣΙΛΗΣ ΟΔΟΝΤΙΔΗΣ
ΑΕΜ:12291

ΕΙΔΙΚΟ ΘΕΜΑ

Έλεγχος Πρωτοκόλλων Ηλεκτρονικού Εμπορίου Με Χρωματισμένα Petri Nets: Ατομική εκτέλεση συναλλαγών

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΡΙΑ: Μ. ΓΟΥΣΙΔΟΥ ΚΟΥΤΙΤΑ
(ΣΕ ΣΥΝΕΡΓΑΣΙΑ ΜΕ ΤΟΝ Π.ΚΑΤΣΑΡΟ)

ΘΕΣΣΑΛΟΝΙΚΗ 2005

Περίληψη. Παρουσιάζουμε μια προσέγγιση ελέγχου μοντέλου για τρεις ιδιότητες ατομικότητας για το πρωτόκολλο ηλεκτρονικού εμπορίου NetBill. Επιβεβαιώνουμε ότι το πρωτόκολλο ικανοποιεί την ιδιότητα της ατομικής μεταφοράς χρημάτων, την ιδιότητα της ατομικότητας κτήσης ψηφιακών αγαθών και την ιδιότητα της εγγύησης παραλαβής των, σε όλες τις περιπτώσεις πιθανής κατάρρευσης (site failure) των συστημάτων που χρησιμοποιούν οι μετέχοντες στη συναλλαγή και σε όλες τις περιπτώσεις μονομερούς ματαίωσης (transaction abort) αυτής. Ο έλεγχος του μοντέλου γίνεται χρησιμοποιώντας τα CPN Tools, ένα εργαλείο βασισμένο στη γλώσσα προγραμματισμού ML, που διαθέτει γραφικό περιβάλλον με το οποίο σχεδιάζουμε και επεξεργαζόμαστε μοντέλα χρωματισμένων δικτύων Petri (CP-Nets). Σε περίπτωση παραβίασης κάποιας ιδιότητας, η ανάλυση αποτυχίας που προτείνουμε για το πρωτόκολλο (failure analysis), στοχεύει στην διερεύνηση όλων των πιθανών σεναρίων παραβίασης, με σκοπό να διορθώσει το σχεδιασμό του πρωτοκόλλου. Για τον έλεγχο του μοντέλου εκμεταλλευόμαστε τις συναρτήσεις ελέγχου του χώρου των καταστάσεων, που μας παρέχει το CPN Tools, καθώς και την υποστηριζόμενη βιβλιοθήκη CTL (Computation Tree like temporal logic). Από την άλλη μεριά, η ανάλυση αποτυχίας του πρωτοκόλλου πραγματοποιείται με τον έλεγχο των τελικών καταστάσεων (dead markings) και αν είναι αναγκαίο με διαδραστική προσομοίωση (interactive simulation) συγκεκριμένων σεναρίων παραβίασης ιδιοτήτων. Στο ηλεκτρονικό εμπόριο τα CP-nets έχουν χρησιμοποιηθεί για να εξετάσουν την παρουσία καταστάσεων αμοιβαίου κλειδώματος (deadlocks), την πιθανότητα κλειδώματος σε υπαρκτό κύκλο καταστάσεων (livelock) και την απουσία μη αναμενόμενων νεκρών μεταβάσεων (dead transitions). Με όσα γνωρίζουμε ως τώρα αυτή είναι η πρώτη προσπάθεια της χρήσης των χρωματισμένων δικτύων Petri για έλεγχο ατομικών ιδιοτήτων σε πρωτόκολλα ηλεκτρονικού εμπορίου καθώς και η πρώτη φορά που χρησιμοποιείται η βιβλιοθήκη CTL. Τέλος πιστεύουμε ότι η διαδικασία που περιγράφουμε μπορεί να χρησιμοποιηθεί και για έλεγχο μοντέλου άλλων λειτουργικών ιδιοτήτων οι οποίες δεν είναι εν γένει σχετικές με τις δομικές ιδιότητες του δημιουργημένου χώρου καταστάσεων σε μοντέλα πολύ πιο πολύπλοκα από αυτό που μελετάμε.

1.Ηλεκτρονικό Εμπόριο

Στις μέρες μας ο παγκόσμιος ιστός (World Wide Web) χρησιμοποιείται ευρύτατα και η ανάπτυξη του είναι ραγδαία. Η χρησιμότητα του είναι πολλαπλή ,τα πλεονεκτήματα χρήσης δελεαστικά και οι δυνατότητες του αμέτρητες. Ο τομέας ο οποίος γνωρίζει την πιο μεγάλη ανάπτυξη είναι αυτός του ηλεκτρονικού εμπορίου(electronic commerce ή εν συντομία e-commerce).

Ο παγκόσμιος ιστός έχει εξελιχθεί σ' ένα μέσο από το οποίο πληθώρα αγαθών και υπηρεσιών προσφέρεται μεταξύ εταιριών, εταιριών και ιδιωτών καθώς και ιδιωτών μεταξύ τους. Φυσικά αυτά δε παρέχονται δωρεάν και συνεπώς η ανάγκη να μπορεί ο χρήστης να πληρώνει για τις υπηρεσίες που του παρέχονται ηλεκτρονικά μέσω διαδικτύου. Αυτή λοιπόν η ανάγκη οδήγησε στη δημιουργία του ηλεκτρονικού εμπορίου, δηλαδή ενός τρόπου να αγοράζονται και να πωλούνται αγαθά μέσω του ιστού απλά και γρήγορα ενώ παράλληλα να διασφαλίζεται η ασφάλεια των συναλλαγών που πραγματοποιούνται. Το ηλεκτρονικό εμπόριο αν και δεν έχει πάνω από δύο δεκαετίες που εμφανίστηκε ολοένα και μεγαλύτερα είναι τα ποσά που διακινούνται ετησίως μέσω αυτού. Ενδεικτικά ας δούμε το παρακάτω συγκριτικό πίνακα μεταξύ του παραδοσιακού και του ηλεκτρονικού εμπορίου:

Έτος	**Παραδοσιακές μορφές εμπορίου (δισεκατομμύρια δολάρια)**	**Ηλεκτρονικό εμπόριο (δισεκατομμύρια δολάρια)**
1994	5150	245
2000	8500	1650
2005	12000	2950

Πηγή:Business Week ,Τεύχος : 12/6/1995

*Τα στοιχεία για το 2000,2005 είναι κατ' εκτίμηση

Καθώς το ηλεκτρονικό εμπόριο εξελίσσονταν άρχισαν να εμφανίζονται και διάφορες μορφές συναλλαγών τις οποίες θα παραθέσουμε συνοπτικά:

1) Συναλλαγές μέσω πιστωτικών καρτών(Credit card transactions)

Είναι ένας από τους πιο διαδεδομένους και απλούς μεθόδους συναλλαγών αν και πολλοί άνθρωποι είναι διστακτικοί λόγω του ότι φοβούνται την υποκλοπή του κωδικού της κάρτας τους. Για να μπορεί κάποιος έμπορος να δέχεται πληρωμές online μέσω πιστωτικής πρέπει να έχει κάποιο λογαριασμό εμπόρου σε κάποια αντίστοιχη τράπεζα..

2) Ηλεκτρονικά πορτοφόλια (e-wallets)

Δημιουργήθηκαν για να διευκολύνουν τις παραγγελίες μέσω πιστωτικών καρτών. Τα ηλεκτρονικά πορτοφόλια επιτρέπουν στο χρήστη να παρακολουθεί τη πίστωση του λογαριασμού του καθώς και να λαμβάνει πληροφορίες για το στάδιο στο οποίο βρίσκεται η αποστολή των εμπορευμάτων που έχει αγοράσει. Εταιρίες πιστωτικών καρτών όπως η Visa παρέχει διάφορα τέτοια πορτοφόλια ανάλογα με τις απαιτήσεις του χρήστη. Τέλος λόγω του ότι κάποιοι vendors δεν αποδεχόταν κάποια ηλεκτρονικά πορτοφόλια η Visa και η Mastercard παρουσίασαν ένα κοινό πρότυπο γλώσσας την Electronic Commerce Modeling Language (ECML) η οποία έχει γίνει αποδεκτή από τη πλειονότητα των Vendors.

3) Εναλλακτικοί τρόποι πληρωμής

Υπάρχουν έμποροι οι οποίοι αντί να δέχονται πληρωμή μέσω πιστωτικών καρτών δέχονται checks ή επιταγές μέσω ταχυδρομείου. Επίσης σπανίως δίνεται και η επιλογή της πληρωμής κατά την παράδοση. Κάποιες εταιρίες έχουν δημιουργήσει τις χρεωστικές κάρτες (debit) που είναι παρόμοιες με τις πιστωτικές με τη διαφορά ότι το ποσό της αγοράς αφαιρείται κατευθείαν από το λογαριασμό, ενώ παρέχεται η δυνατότητα οι χρήστες να μπορούν να κάνουν και ανάληψη χρημάτων από το λογαριασμό από μηχανήματα αυτόματης συναλλαγής (ATM).

4) Ψηφιακές συναλλαγματικές μονάδες(digital currency)

Υπάρχουν διάφορες μορφές όπως για παράδειγμα το ψηφιακό χρήμα (digital cash).Αποθηκεύεται ηλεκτρονικά, χρησιμοποιείται για ψηφιακές συναλλαγές ενώ μπορεί ο χρήστης να καταθέσει ή να κάνει ανάληψη χρημάτων όπως σ' ένα "κλασικό" λογαριασμό τράπεζας. Αξίζει να σημειωθεί εδώ ότι το ψηφιακό χρήμα επιτρέπει σε ανθρώπους που δεν διαθέτουν πιστωτικές κάρτες να συναλλάσσονται μέσω διαδικτύου ενώ συμφέρει και λόγω

του ότι δεν υπάρχουν χρεώσεις συναλλαγής (transaction fees) της πιστωτικής κάρτας. Μία εναλλακτική μορφή ψηφιακού χρήματος είναι το gift cash το οποίο συνηθέστερα πωλείται με τη μορφή πόντων η οποία επιτρέπει σε άτομα που δεν έχουν πιστωτική κάρτα να ψωνίσουν από το διαδίκτυο. Σε κάποιες περιπτώσεις μάλιστα παρέχονται και κάποια δώρα ανάλογα με τις αγορές του χρήστη τις επισκέψεις σε ιστοσελίδες κλπ.

5)Peer to peer Payments

Οι πληρωμές τύπου peer to peer επιτρέπουν on-line πληρωμές μεταξύ χρηστών του διαδικτύου. Υπάρχουν διάφοροι τρόποι για να γίνει αυτό και εδώ θα αναφέρουμε ενδεικτικά μερικούς:

1) Το eCash επιτρέπει τη μεταφορά ψηφιακού χρήματος μεταξύ δύο ατόμων που έχουν λογαριασμό σε τράπεζες που συνεργάζονται με το eCash μέσω email.

2) Το Paypal προσφέρει ένα σύστημα πληρωμών το X payments το όποιο επιτρέπει σ' ένα χρήστη να στέλνει χρήματα σε οποιονδήποτε θέλει αρκεί να έχει διεύθυνση ηλεκτρονικού ταχυδρομείου χωρίς να είναι απαραίτητα ήδη καταχωρημένος στο Paypal.Όποιος θέλει να το χρησιμοποιήσει πρέπει να δημιουργήσει λογαριασμό στο Paypal να καταχωρήσει το ποσό που θέλει να στείλει και αυτό χρεώνεται στη πιστωτική του κάρτα, ενώ τοποθετούνται σε λογαριασμό του Paypal τα χρήματα στο όνομα του αποδέκτη και του στέλνεται ειδοποίηση ότι έχει γίνει η χρέωση μέσω ηλεκτρονικού ταχυδρομείου. Όταν η ειδοποίηση φτάσει τότε ο αποδέκτης μπορεί απλά να κάνει register στο Paypal και να παραλάβει τα χρήματα με κατάθεση στη τράπεζά του ή με αποστολή επιταγής. Το σύστημα του Paypal είναι πολύ βολικό και εύκολο στη χρήση ενώ είναι ιδανικό για online δημοπρασίες σε πραγματικό χρόνο καθώς επιτρέπει τη πληρωμή μέσω πιστωτικής κάρτας. Αυτό πρακτικά σημαίνει τη δυνατότητα εκκίνησης επεξεργασίας της συναλλαγής με το που ξεκινήσει η δημοπρασία ώστε να αποφεύγεται η πλαστογραφία ή να έχει χρησιμοποιηθεί λογαριασμός χρήστη εν αγνοία του.

3)Ένα ακόμα αξιοσημείωτο σύστημα peer to peer συναλλαγών είναι το BillPoint που επιτρέπει στους αγοραστές να υποβάλουν ηλεκτρονικές πληρωμές στους λογαριασμούς των ατόμων από τους οποίους έχουν αγοράσει.

6) Έξυπνες κάρτες (Smart Cards)

Είναι κάρτες στις οποίες έχει τοποθετηθεί ένα μικροκύκλωμα και στις οποίες μπορεί να αποθηκευτεί πολύ μεγαλύτερη ποσότητα πληροφοριών σε σχέση με μία πιστωτική κάρτα που απλά διαθέτει μαγνητική ταινία. Οι έξυπνες κάρτες χρησιμοποιούνται π.χ. στην υγεία,

στις μεταφορές και σε τραπεζικές συναλλαγές. Μάλιστα στην ίδια κάρτα μπορούν να αποθηκεύονται παράλληλα διάφορες πληροφορίες όπως για παράδειγμα για το πρόγραμμα ιατρικής περίθαλψης καθώς και πληροφορίες για τραπεζικές συναλλαγές. Διακρίνονται σε κάρτες με επαφή (contact) και χωρίς επαφή (contactless).Για να διαβαστούν τα δεδομένα και να ενημερωθεί μια κάρτα με επαφή πρέπει να τοποθετηθεί σ' έναν οδηγό ανάγνωσης έξυπνων καρτών (smart card reader). Μια κάρτα χωρίς επαφή έχει προσαρτημένη, πέρα από το μικροκύκλωμα, μια περιελιγμένη κεραία, έτσι ώστε να μπορεί να ανταλλάσσει δεδομένα. Η κάρτα μη-επαφής επιτρέπει ταχύτερη ανταλλαγή δεδομένων σε σχέση με τη κάρτα επαφής. Οι έξυπνες κάρτες δίνουν την δυνατότητα στο χρήστη να πληκτρολογεί κάποιο κωδικό ασφαλείας δίνοντας του έτσι ένα πλεονέκτημα στο τομέα της ασφάλειας συγκριτικά με τις πιστωτικές κάρτες, καθώς υπάρχει και η δυνατότητα επιλογής τα δεδομένα της κάρτας είτε μόνο να διαβάζονται (read only) είτε να μην επιτρέπεται η πρόσβαση σ' αυτά (no access). Επίσης κάποιες εταιρίες όπως η eConnect χρησιμοποιούν παρόμοια τεχνολογία συνδυάζοντας τις έξυπνες κάρτες με συσκευές για να πετύχουν υψηλότερο επίπεδο ασφάλειας για τις συναλλαγές στο διαδίκτυο. Τέλος οικονομικοί οργανισμοί τις χρησιμοποιούν ώστε να επωφεληθούν τα μέλη τους από αυτές. Η Visa π.χ. έχει λανσάρει τη Visa Card μια έξυπνη κάρτα με την οποία ο χρήστης μπορεί να καταθέτει χρήματα και να κάνει συναλλαγές , ενώ μπορεί να τοποθετεί τη κάρτα στον οδηγό ανάγνωσης και να βλέπει πριν εκτελέσει μια συναλλαγή πόσα χρήματα θα υπάρχουν στο λογαριασμό πριν και μετά τη συναλλαγή. Ένα πλεονέκτημα αυτών των καρτών είναι ότι ο κάτοχος τους μπορεί, αν τελειώσουν τα χρήματα στη κάρτα, να μην τις ξαναχρησιμοποιήσει ή να προσθέσει και άλλα λεφτά και να πραγματοποιήσει νέες συναλλαγές.

7)Μικροπληρωμές (Micropayments)

Για κάθε συναλλαγή με πιστωτική κάρτα που πραγματοποιεί ένας έμπορος χρεώνεται με κάποιο ποσό. Αυτό μπορεί να γίνει ασύμφορο αν οι πελάτες του αγοράζουν προϊόντα μικρής αξίας καθώς υπάρχει περίπτωση η χρέωση να είναι πιο ακριβή από την αξία του προϊόντος. Για να επιτραπεί στους εμπόρους και στις εταιρίες γενικότερα να έχουν κέρδος από αυτού του είδους τις συναλλαγές δημιουργήθηκαν οι μικροπληρωμές ,δηλαδή πληρωμές οι οποίες γενικά δε ξεπερνούν τα 10 ευρώ περίπου. Για το σκοπό αυτό δημιουργ συνεργασίες μεταξύ εταιριών ώστε να είναι δυνατή η παροχή των μικροπληρωμών. Για παράδειγμα ένας λογαριασμός τηλεφώνου είναι ένα άθροισμα μικροπληρωμών οι οποίες χρεώνονται στο τέλος μιας συγκεκριμένης περιόδου ώστε να δικαιολογείται το κόστος

συναλλαγής. Ορισμένες εταιρίες πάλι συνεργαζόμενες με εταιρίες παροχής υπηρεσιών χρεώνουν τις μικροπληρωμές στους μηνιαίους λογαριασμούς των πελατών τους. Μια εναλλακτική περίπτωση είναι το eCharge το οποίο απαιτεί dial-up σύνδεση και χρεώνει το πελάτη με τη λήξη της συναλλαγής συνδέοντας τον με ένα τηλεφωνικό νούμερο 1-900 ώστε να χρεωθεί η αγορά του σαν κλήση στο τηλεφωνικό λογαριασμό του.

Όλοι οι παραπάνω τρόποι χρηματικών συναλλαγών στο διαδίκτυο για να πετύχουν εμπορικά πρέπει να διασφαλίζουν τη μη διαρροή των προσωπικών δεδομένων και την προστασία των χρημάτων των πελατών τους. Κανείς δε θα έδινε π.χ. τον αριθμό της πιστωτικής του κάρτας σε κάποιον άλλο σε μία συναλλαγή αν δεν ήταν σίγουρος ότι :

1. *Αυτός που τη λαμβάνει είναι πράγματι ο έμπορος που κάνει τη συναλλαγή*
2. *Ο έμπορος που θα λάβει το κωδικό να μην καταχραστεί χρήματα από το χρήστη*
3. *Κάποιος τρίτος δε θα μπορεί να κλέψει αυτά τα στοιχεία κ.α.*

Αυτά αλλά και περισσότερα ακόμα προβλήματα πρέπει να αντιμετωπιστούν στο ηλεκτρονικό εμπόριο και γι’ αυτό έχουν δημιουργηθεί πρωτόκολλα ασφάλειας για τις ατομικές χρηματικές συναλλαγές.

2.Πρωτόκολλα ασφαλείας στο ηλεκτρονικό εμπόριο

2.1 Εισαγωγή

Όπως αναφέραμε πιο πάνω η ανάγκη για ασφάλεια στις συναλλαγές μέσω του διαδικτύου έχει οδηγήσει στη δημιουργία πληθώρας πρωτοκόλλων ηλεκτρονικού εμπορίου. Στο Internet υπάρχουν διάφορα πρωτόκολλα κρυπτογραφίας κάθε ένα από τα οποία ειδικεύεται σε μία διαφορετική λειτουργία. Μερικά έχουν σχεδιαστεί για να προστατεύουν συγκεκριμένες μεθόδούς επικοινωνίας, όπως είναι το ηλεκτρονικό ταχυδρομείο (e-mail) και η απομακρυσμένη πρόσβαση (remote login). Άλλα παρέχουν υπηρεσίες κρυπτογράφησης σε διάφορους τομείς των επικοινωνιών. Εμείς θα περιγράψουμε εν συντομία μερικά από τα πιο βασικά πρωτόκολλα ηλεκτρονικού εμπορίου.

2.2 Digicash

Το Digicash χρησιμοποιεί ένα ανώνυμο πρωτόκολλο ψηφιακού χρήματος. Δεν ισχύει η ιδιότητα της ατομικότητας χρημάτων (αναφορά της ιδιότητας στο Κεφάλαιο 4) ενώ σε περίπτωση σφάλματος επικοινωνίας το πρωτόκολλο παύει να διατηρεί και την ανωνυμία. Επίσης χρησιμοποιούνται αρκετά πολύπλοκες κρυπτογραφικές λειτουργίες που καθιστούν τα έξοδα λειτουργίας του υψηλά. Συνεπώς το πρωτόκολλο αυτό δεν είναι κατάλληλο για μικροπληρωμές.

2.3 First Virtual

Το πρωτόκολλο First Virtual επιτρέπει στους χρήστες να αγοράζουν ελεύθερα αγαθά. Το πρωτόκολλο ενημερώνει το χρήστη από το ηλεκτρονικό του ταχυδρομείο για κάθε συναλλαγή που έχει κάνει. Το πρωτόκολλο διατηρεί τον ατομικότητα χρημάτων ενώ αποτυγχάνει στην ατομικότητα των αγαθών γιατί τι θεωρεί ασήμαντη ως ιδιότητα. Σαν παροχή ασφάλειας είναι ανώτερο από το Digicash αλλά παραμένει ευάλωτο σε απάτη ή

αποτυχίες στην ατομικότητα του. Τέλος δεν είναι ιδιαίτερα συμφέρον για μικροπληρωμές γιατί μετατρέπει κάθε ηλεκτρονική συναλλαγή σε συναλλαγή πιστωτικής κάρτας..\

2.4 SSL

Το SSL (Secure Sockets Layer) είναι ένα ευέλικτο, γενικού σκοπού σύστημα κρυπτογράφησης για την προστασία της επικοινωνίας μέσω του Web, το οποίο είναι ενσωματωμένο και στον Netscape και στον Microsoft browser.Κατά τη διάρκεια της συναλλαγής δημιουργεί ένα ασφαλές κανάλι επικοινωνίας με τη χρήση κρυπτογραφικών μεθόδων για να μεταφέρει με ασφάλεια το κωδικό της πιστωτικής κάρτας του πελάτη στον έμπορο. Αυτή η προσέγγιση των συναλλαγών παρέχει ατομικότητα χρημάτων εφόσον οι συναλλαγές μέσω πιστωτικής είναι χρηματικά ατομικές. Φυσικά ένας τέτοιος τρόπος συναλλαγής μπορεί να οδηγήσει σε απάτη από μέρος του εμπόρου αφού έχει τον κωδικό της πιστωτικής κάρτας του πελάτη. Η ατομικότητα των αγαθών δεν ισχύει σε αυτό το πρωτόκολλο. Και αυτό το πρωτόκολλο δεν συμφέρει για μικροπληρωμές.

2.5 SST/SEPP/iKP

Αυτά είναι τρία πρωτόκολλα το πρώτο της Visa/Microsoft, το δεύτερο της Mastercard και το τρίτο της IBM είναι παραδείγματα ασφαλών πρωτοκόλλων βασιζόμενων σε συναλλαγές μέσω πιστωτικής κάρτας. Αν και διαφέρουν στα σημεία έχουν ένα κοινό σημείο και αυτό είναι η ψηφιακή υπογραφή του χρήστη στην αίτηση αγοράς μαζί με τη τιμή του προϊόντος. Αυτό μετά ο χρήστης το κρυπτογραφεί χρησιμοποιώντας ένα δημόσιο κλειδί της τράπεζας. Ομοίως ο έμπορος με την ίδια διαδικασία ετοιμάζει μια αίτηση πώλησης μαζί με μία τιμή κα το στέλνει στη τράπεζα. Η τράπεζα συγκρίνει τις τιμές και αν ταιριάζουν η συναλλαγή εκτελείται. Όπως και στο SSL ισχύει η ατομικότητα χρημάτων εφόσον οι συναλλαγές μέσω πιστωτικής είναι χρηματικά ατομικές. Όμως και τα τρία αυτά πρωτόκολλα είναι καλύτερα από τα προηγούμενα γιατί αποτρέπουν την απάτη του εμπόρου. Επίσης δεν είναι και τα πλέον κατάλληλα πρωτόκολλα για μικροπληρωμές

2.6 NetBill

Το πρωτόκολλο NetBill είναι ένα πρωτόκολλο ηλεκτρονικού χρήματος και χρησιμοποιείται για την πώληση και παράδοση προϊόντων δικτύου χαμηλού κόστους. Το NetBill θα το αναπτύξουμε παρακάτω λεπτομερώς καθώς αποτελεί το πρωτόκολλο που θα χρησιμοποιήσουμε για να δείξουμε πως μπορούμε να κάνουμε έλεγχο μοντέλου.

Σε όλα τα πρωτόκολλα που αναφέραμε παραπάνω μιλήσαμε για κάποιες ιδιότητες ατομικότητας τις οποίες διαθέτουν. Παρακάτω θα περιγράψουμε τα χρωματισμένα δίκτυα Petri τα οποία θα μας βοηθήσουν να αποδείξουμε τέτοιες ιδιότητες ατομικότητας σε ένα πρωτόκολλο ηλεκτρονικού εμπορίου.

3.Χρωματισμένα δίκτυα Petri

3.1 Εισαγωγή στα Χρωματισμένα Δίκτυα Petri

Τα δίκτυα Petri (Petri Nets/PNs) είναι μια τυπική και γραφικά ελκυστική γλώσσα που είναι κατάλληλη για τη μοντελοποίηση συστημάτων με χαρακτηριστικά σύγχρονης και ασύγχρονης εκτέλεσης, κατανομής και παραλληλίας και μη ντετερμινιστικής/ στοχαστικής συμπεριφοράς. Ουσιαστικά η γλώσσα αυτή, είναι μια γενίκευση της θεωρίας αυτομάτων, τέτοια ώστε να μπορεί να εκφρασθεί η έννοια των ταυτόχρονα εκτελούμενων γεγονότων. Ως γραφικό εργαλείο, τα δίκτυα Petri μπορούν να χρησιμοποιηθούν σαν βοήθεια οπτικής μορφής παρόμοια με τα διαγράμματα ροής και τα μπλοκ διαγράμματα, ενώ ως μαθηματικό εργαλείο είναι δυνατό να κατασκευαστούν εξισώσεις κατάστασης, αλγεβρικές εξισώσεις ή άλλα μαθηματικά μοντέλα που ρυθμίζουν τη συμπεριφορά των συστημάτων.

Τα δίκτυα Petri γεννήθηκαν το 1962, στη Φυσικομαθηματική Σχολή του Τεχνικού Πανεπιστημίου του Ντάρμσταντ στη Γερμανία, από τη διδακτορική διατριβή του Carl Adam Petri, “Kommunikation mit Automaten”, που είχε ως αντικείμενο την επικοινωνία μεταξύ αυτόματων μηχανών. Χρησιμοποιώντας ένα δίκτυο, περιέγραψε την τυπική σχέση μεταξύ των γεγονότων σε ένα σύστημα υπολογιστών. Ήταν η πρώτη φορά που διατυπώθηκε μια γενική θεωρία για τα διακριτά παράλληλα συστήματα.

Τα δίκτυα Petri με το πέρασμα των χρόνων το εύρος της χρήσης των δικτύων Petri επεκτάθηκε καθώς διαπιστωνόταν σιγά σιγά η πολυτιμότητα τους. Έτσι τα δίκτυα Petri στην αρχή χρησιμοποιήθηκαν για την μοντελοποίηση και την ανάλυση συστημάτων με ταυτόχρονα γεγονότα, σε εφαρμογές μηχανικών από αναλυτές με βάσεις μηχανικών ,σε Αυτόματα Συστήματα Κατεργασιών. Τέλος ανακαλύφθηκε ότι τα δίκτυα Petri ήταν ένα ισχυρό εργαλείο στην περιγραφή συστημάτων που εξαρτώνται από γεγονότα (event driven systems). Αυτά τα συστήματα μπορεί να ήταν ασύγχρονα, να περιέχουν σειριακές ή παράλληλες εργασίες, να περιλαμβάνουν συγκρούσεις, αμοιβαίο αποκλεισμό και να μην είναι ντετερμινιστικά

3.2 Ορισμός των Χρωματισμένων Δικτύων Petri

Τα χρωματισμένα δίκτυα Petri μας παρέχουν τις πρωταρχικές έννοιες για τον ορισμό των τύπων όπως (καταχωρήσεις ονομάτων, κλπ) καθώς και τη διαχείριση των τιμών των δεδομένων τους. Έτσι ενώ διατηρούμε τη σχεδιαστική ευκολία του περιβάλλοντος μιας γλώσσας προγραμματισμού έχουμε τη δυνατότητα να επωφεληθούμε από την εκφραστικότητα και την δυνατότητα φορμαλιστικής ανάλυσης της γλώσσας μοντελοποίησης των CP-nets.

Τα δίκτυα Petri μας παρέχουν μια σαφή παράσταση των καταστάσεων και των ενεργειών ενός μοντέλου. Στα CP-nets οι καταστάσεις παριστάνονται με θέσεις (οι οποίες σχεδιάζονται ως ελλείψεις).Για λόγους συμβολισμού γράφουμε τα ονόματα των θέσεων μέσα στις ελλείψεις. Κάθε θέση έχει ένα συσχετισμένο τύπο δεδομένων ο οποίος προσδιορίζει το είδος των δεδομένων τα οποία μπορεί να περιέχει η κατάσταση (ο τύπος των πληροφοριών γράφεται με πλάγια γράμματα δίπλα από τις θέσεις).

Ο συγκεκριμένος τύπος των δηλώσεων καθορίζει τις λειτουργίες που μπορούν να γίνουν με τις τιμές των δεδομένων. Μια κατάσταση ενός CP-Net καλείται μαρκάρισμα (marking) και αποτελείται από ένα αριθμό μαρκών (tokens) τοποθετημένες σε ξεχωριστές θέσεις. Κάθε μάρκα μεταφέρει μια τιμή , η οποία ανήκει στο τύπο της αντίστοιχης θέσης.

Το μαρκάρισμα ενός CP-net είναι μια συνάρτηση η οποία απεικονίζει κάθε θέση σε ένα πολλαπλό σύνολο από μάρκες του ίδιου τύπου. Αναφερόμαστε στις τιμές των μαρκών ως χρώματα μαρκών και στους τύπους των δεδομένων τους ως σύνολα χρωμάτων. Οι τύποι ενός δεδομένου Ψ μπορεί να είναι αυθαίρετα πολύπλοκοι π.χ. μία εγγραφή όπου το

ένα πεδίο είναι πραγματικός το άλλο μια συμβολοσειρά και το τρίτο μια λίστα από ακεραίους.

Οι ενέργειες ενός CP-net αναπαρίστανται με τις μεταθέσεις (transitions) ,οι οποίες παριστάνονται ως ορθογώνια παραλληλόγραμμα. Ένα τόξο που ξεκινάει από μία θέση και πάει σε μια μετάβαση μπορεί να αφαιρέσει μάρκες από τη θέση που ξεκινάει ενώ αν ξεκινάει από μία μετάβαση και πάει προς μια θέση τότε μπορεί να προσθέσει στη θέση που καταλήγει μάρκες. Ο ακριβής αριθμός των μαρκών και οι τιμές των δεδομένων τους καθορίζονται από τις επιγραφές των τόξων οι οποίες τοποθετούνται δίπλα από τα τόξα. Οι επιγραφές των τόξων (arc inscriptions) μπορεί να περιέχουν μεταβλητές και σταθερές. Για να μιλήσουμε για εκτέλεση μιας μετάβασης πρέπει να καταχωρήσουμε στις εισερχόμενες εκφράσεις τιμές από τους αντίστοιχους τύπους. Ας υποθέσουμε ότι καταχωρούμε στην εισερχόμενη μεταβλητή v κάποιας μετάβασης T την τιμή d.Το ζευγάρι (T,<v=d>) καλείται στοιχείο καταχώρησης (binding element) και ενεργοποιείται με ένα μαρκάρισμα M , όπου υπάρχουν αρκετές μάρκες στις αρχικές του θέσεις .Σ' ένα μαρκάρισμα M είναι πιθανό να υπάρχουν ενεργοποιημένα περισσότερα του ενός στοιχεία καταχώρησης του T. Αν το στοιχείο καταχώρησης εκτελεστεί αφαιρεί μάρκες από τις θέσεις εισόδου και τις τοποθετεί στις θέσεις εξόδου. Πέρα από τις επιγραφές τόξων μπορούμε επίσης να προσθέσουμε μια μπουλιανή έκφραση με μεταβλητές σε κάθε μετάβαση. Η μπουλιανή έκφραση καλείται φύλακας και καθορίζει ότι δεχόμαστε να περάσουν μόνο συγκεκριμένα στοιχεία καταχωρήσεων για τα οποία η μπουλιανή έκφραση είναι αληθής.

Η συμπεριφορά ενός μοντέλου CP-net χαρακτηρίζεται από κάποιες συγκεκριμένες ιδιότητες.

- **Οι ιδιότητες περατότητας (Bounds-related properties)** χαρακτηρίζουν το μοντέλο όσων αφορά τον αριθμό των μαρκών που μπορεί να έχουμε σε θέσεις του μοντέλου που μας ενδιαφέρουν.

- **Οι ιδιότητες οικείας κατάστασης (Home properties)** παρέχουν πληροφορίες για τα μαρκαρίσματα η τα σύνολα μαρκαρισμάτων στα οποία είναι πάντα πιθανό να επιστρέψει το μοντέλο.
- **Οι ιδιότητες διάρκειας (Liveness properties***)* χρησιμεύουν στο να εξετάσουμε εάν ένα σύνολο από στοιχεία καταχώρησης X παραμένει ενεργό: "Για κάθε επιτρεπτή κατάσταση μαρκαρίσματος M , είναι πιθανό να βρούμε μια

πεπερασμένη ακολουθία από μαρκαρίσματα που ξεκινάνε από το M και τα οποία περιέχουν στοιχεία του X;”

- **Οι ιδιότητες δικαιοσύνης (Fairness properties)** μας παρέχουν πληροφορίες για το πόσο συχνά τα διαφορετικά στοιχεία καταχώρησης εκτελούνται.

Τα CP-nets αναλύονται με

- Τη μέθοδο της προσομοίωσης (simulation)
- Φορμαλιστικές μεθόδους ανάλυσης όπως τη κατασκευή γραφημάτων εκτέλεσης (*occurrence graphs)*, τα οποία παριστάνουν όλα τις επιτρεπτές καταστάσεις.
- Υπολογισμός και ερμηνεία των αμετάβλητων διανυσμάτων (invariants) του συστήματος (καλούνται αμετάβλητα διανύσματα θέσης και μετάβασης),
- Επίδοση μειώσεων η οποία συρρικνώνει το μοντέλο χωρίς να αλλάζει ένα συγκεκριμένο πλήθος ιδιοτήτων
- Τον έλεγχο των δομικών ιδιοτήτων (structural properties),ο οποίος εγγυάται συγκεκριμένες ιδιότητες συμπεριφοράς

3.3 CPN Tools

Στα CPN Tools, τα CP-nets σχεδιάζονται σ’ ένα μοντέρνο γραφικό περιβάλλον (GUI-based environment) το οποίο παρέχει διαδραστική ανάδραση (interactive feedback) για τη συμπεριφορά του μοντέλου μέσω της προσομοίωσης. Τα χρώματα ,οι μεταβλητές ,οι δηλώσεις συναρτήσεων και οι επιγραφές δικτύου γράφονται στη γλώσσα CPN-ML η οποία αποτελεί προέκταση της Standard ML και εξαιτίας αυτού του γεγονότος αποτελεί ένα λειτουργικό τρόπο προγραμματισμού. Στα CPN Tools εργαζόμαστε με απλά καθώς και με σύνθετα στοιχεία χρωματισμένων συνόλων όπως εγγραφές, λίστες και ενώσεις χρωματισμένων συνόλων.

Η εργαλειοθήκη (toolset) παρέχει τις απαραίτητες διευκολύνσεις για την ανάλυση απλών και δικτύων Petri με χρόνο τα οποία αποτελούνται από κάποιο αριθμό ιεραρχικά συνδεδεμένων σελίδων. Ένα τυπικό μοντέλο αποτελείται από 10-100 σελίδες με διαφοροποίηση στη πολυπλοκότητα και τις προγραμματιστικές απαιτήσεις.. Το companion *state space tool* επιτρέπει τη δημιουργία ολόκληρου ή ενός μέρους του χώρου καταστάσεως του μοντέλου (γράφημα εκτέλεσης) και την υποβολή τυποποιημένων η μη τυποποιημένων ερωτημάτων επί του μοντέλου

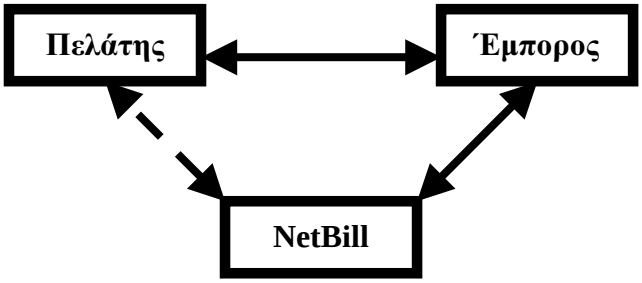
4.Έλεγχος Πρωτοκόλλων Ηλεκτρονικού Εμπορίου Με Χρωματισμένα Petri Nets

4.1 Περιγραφή του πρωτοκόλλου NetBill

Θα κάνουμε μια μη αυστηρή εισαγωγή στο τρόπο ελέγχου ενός πρωτοκόλλου ηλεκτρονικού εμπορίου χρησιμοποιώντας τα χρωματισμένα δίκτυα Petri και το CPN tools λογισμικό. Στο παράδειγμά μας θα αναφερθούμε στο πρωτόκολλο NetBill το οποίο θα δείξουμε πως μπορούμε να το προσομοιώσουμε και να ελέγξουμε αν ικανοποιεί τρεις ιδιότητες ατομικότητας οι οποίες είναι απαραίτητες για να εξασφαλιστεί η προστασία η οποία παρέχει το πρωτόκολλο. Αυτές είναι:

- **Ατομικότητα χρημάτων(Money atomicity)**Το πρωτόκολλο δεν πρέπει να δημιουργεί ή να εξαλείφει χρήματα κατά τη μεταφορά τους από το ένα άτομο που συμμετέχει στη συναλλαγή στο άλλο (party) στην άλλη.
- **Ατομικότητα αγαθών (Goods atomicity):**Το πρωτόκολλο πρέπει να εξασφαλίζει ότι εφόσον αγοράσει κάποιος ένα εμπόρευμα θα το λάβει αν και μόνο αν ο έμπορος έχει πληρωθεί.
- **Εγγυημένη παραλαβή (Certified delivery):**Το πρωτόκολλο πρέπει να εξασφαλίζει ότι ο πωλητής και αγοραστής μπορούν να αποδείξουν ποια προϊόντα έχουν στείλει ή αγοράσει αντίστοιχα.

Καταρχάς θα κάνουμε μια σύντομη αναφορά στο τρόπο λειτουργίας του πρωτοκόλλου. Μια εκτενέστερη περιγραφή μπορεί να βρεθεί στο [1]. Σε μία συναλλαγή μέσω του NetBill συμμετέχουν 3 οντότητες – πρόσωπα : ο αγοραστής , ο έμπορος και ο server του NetBill. Σχηματικά θα μπορούσε να παρασταθεί η αλληλεπίδραση μεταξύ τους ως εξής:



Μηνύματα πρωτοκόλλου συναλλαγής
Μηνύματα εκτός πρωτοκόλλου

Γενικά μια διαδικασία συναλλαγής μέσω του NetBill περιλαμβάνει τρεις φάσεις:1)διαπραγμάτευση τιμής,2)παράδοση αγαθών και 3) πληρωμή. Στις δύο πρώτες συμμετέχουν μόνο ο πελάτης και ο έμπορος ενώ στη τρίτη φάση και ο server του NetBill.Θα εξηγήσουμε τι συμβαίνει σε κάθε μία από αυτές τις φάσεις εισάγοντας και έναν αριθμό βημάτων ενδιάμεσα:

Φάση 1 – Διαπραγμάτευση τιμής

Σαν πρώτο βήμα της πρώτης φάσης ο πελάτης δίνει ένα πιστοποιητικό εισιτήριο (identifying ticket) αναγνωρίζεται ως χρήστης του NetBill ενώ με κάποια επιπλέον στοιχεία (credentials) μπορεί να τύχει και κάποιας έκπτωσης αν ανήκει σε κάποιες ειδικές περιπτώσεις. Ο πελάτης συμπληρώνει και στέλνει στον έμπορο μία Αίτηση Προϊόντος (Product Request Data) και μερικές σημαίες (flags) οι οποίες έχουν σχέση με λεπτομέρειες της συναλλαγής (όπως π.χ. οδηγίες παράδοσης του αγαθού).Επίσης μπορεί να στείλει μία προσφορά (Bid) δηλαδή μια τιμή την οποία είναι διατεθειμένος να πληρώσει για το αγαθό. Στο δεύτερο βήμα της πρώτης φάσης ο έμπορος την αποθηκεύει για να τη χρησιμοποιήσει μετέπειτα, δημιουργεί ένα νέο σύνολο από σημαίες με βάση αυτές του πελάτη ,συμφωνεί στη τιμή που του ζήτησε ο πελάτης και δημιουργεί μια ταυτότητα συναλλαγής (Transaction ID) για να υπάρχει μία αναγνώριση της συναλλαγής σε μετέπειτα στάδια γι' αυτό και δεν είναι μοναδική. Τα βήματα 1 και 2 μπορούν να επαναληφθούν μέχρι να επέλθει συμφωνία μεταξύ εμπόρου και πελάτη για τη συναλλαγή. Σε περίπτωση που έχει ξαναγίνει κάποια παλιότερη διαπραγμάτευση δε χρειάζεται να δημιουργηθεί νέα ταυτότητα συναλλαγής καθώς υπάρχει η παλαιότερη.

Φάση 2 – Η φάση παραλαβής των αγαθών

Όταν επιτευχθεί συμφωνία επί της τιμής η συναλλαγή μπαίνει στο τρίτο βήμα. Ο πελάτης κατευθύνει τον έμπορο στο να στείλει τα αγαθά στέλνοντας την ταυτότητα συναλλαγής που είχε χρησιμοποιηθεί προηγουμένως. Στο τέταρτο βήμα ο έμπορος δημιουργεί ένα μοναδικό συμμετρικά κρυπτογραφημένο κλειδί (unique symmetric cipher key), κωδικοποιεί τα αγαθά χρησιμοποιώντας το και τα στέλνει στο πελάτη μαζί μ' ένα κρυπτογραφημένο checksum. Υπολογισμένο πάνω στο αγαθό έτσι ώστε αν ο πελάτης παρατηρήσει κάποια ασυνέπεια να μη προχωρήσει τη συναλλαγή. Τέλος ο έμπορος στέλνει και μια ταυτότητα φόρμας ηλεκτρονικής παραγγελίας (Electronic Payment Order ID) μαζί με τα κρυπτογραφημένα αγαθά. Η ταυτότητα φόρμας ηλεκτρονικής παραγγελίας είναι μοναδική (globally unique) και θα χρησιμοποιηθεί από τη βάση δεδομένων του NetBill Server ώστε να αναγνωριστεί αποκλειστικά αυτή η συναλλαγή. Η ταυτότητα φόρμας ηλεκτρονικής παραγγελίας αποτελείται από τρία πεδία: στο ένα καταχωρούνται τα στοιχεία του εμπόρου, στο δεύτερο μια timestamp η οποία δηλώνει το χρόνο και τη λήξη της παραλαβής των αγαθών και τέλος ένα σειριακό αριθμό που εγγυάται τη μοναδικότητα της φόρμας. Η μοναδικότητα της ταυτότητας φόρμας ηλεκτρονικής παραγγελίας αποσκοπεί στο να αποφευχθούν παρατυπίες από εμπόρους οι οποίοι μπορεί να χρησιμοποιήσουν παλιότερες τέτοιες ταυτότητες συναλλαγών ώστε να προβούν σε πολλαπλές αιτήσεις πληρωμής χωρίς να ΄χουν στείλει κάποιο αγαθό. Έχουμε φτάσει λοιπόν στο τέλος της δεύτερης φάσης όπου ο πελάτης έχει λάβει κωδικοποιημένα τα αγαθά και δε συνεπώς δε μπορεί να τα χρησιμοποιήσει αν δε πληρώσει τον έμπορο ώστε να λάβει το κλειδί K που θα τα αποκωδικοποιήσει.

Φάση 3- Η πληρωμή

Στο πέμπτο βήμα ο χρήστης αποστέλλει στον έμπορο συμπληρωμένη την φόρμα ηλεκτρονικής παραγγελίας (EPO) υπογεγραμμένη. Αν τη στείλει μετά δε μπορεί να αναιρέσει τη διαδικασία συναλλαγής. Εν συντομία μια φόρμα ηλεκτρονικής παραγγελίας αποτελείται από δύο μέρη. Ένα το οποίο μπορεί να διαβαστεί από τον έμπορο και τον server του NetBill και ένα κρυπτογραφημένο το οποίο παρέχει οδηγίες πληρωμής και είναι αναγνώσιμο μόνο από τον server του NetBill. Στο έκτο βήμα ο έμπορος λαμβάνει τη φόρμα ηλεκτρονικής παραγγελίας και τη στέλνει στο server του NetBill.Στη φόρμα τώρα έχουν προστεθεί αυτόματα ο κωδικός λογαριασμού του εμπόρου, το πεδίο memo (memo field), το κλειδί της αποκρυπτογράφησης των αγαθών καθώς την υπογραφή του εμπόρου.

Αν στείλει ο έμπορος τη φόρμα δε μπορεί να ακυρώσει από δω και μπρος τη συναλλαγή με το πελάτη. Ο server του NetBill εκτιμάει τη φόρμα και αποφασίζει για το αν η συναλλαγή είναι πραγματοποιήσιμη η όχι και μετά εκδίδει μια απόδειξη στην οποία αναγράφεται το αποτέλεσμα της συναλλαγής ,τα στοιχεία του πελάτη και του εμπόρου , η τιμή και η περιγραφή του/των αγαθών, η φόρμα ηλεκτρονικής παραγγελίας και το κλειδί K που χρειάστηκε για να αποκρυπτογραφηθούν τα αγαθά ενώ τέλος υπογράφεται ηλεκτρονικά από το server του NetBill χρησιμοποιώντας τον αλγόριθμο DSA. Αυτή τη στέλνει στον έμπορο μαζί με μία ενημέρωση της κατάστασης των χρημάτων του λογαριασμού του πελάτη και έτσι πραγματοποιείται και το έβδομο βήμα. Τέλος στο όγδοο βήμα ο έμπορος στέλνει αυτά που έλαβε στο πελάτη. Ο πελάτης μπορεί να ελέγξει τη ταυτότητα της φόρμας ώστε να βεβαιωθεί ότι δεν έγινε κάποια απάτη στη συναλλαγή καθώς και ενημερώνεται για τη κατάσταση του λογαριασμού του.

4.2 Η μεταφορά του πρωτοκόλλου σε περιβάλλον CPN Tools

Αυτά λοιπόν γίνονται κατά τη διάρκεια μιας συναλλαγής χρησιμοποιώντας το πρωτόκολλο NetBill.Για να μπορέσουμε να φτάσουμε στο στάδιο της προσομοίωσης θα χρειαστεί να προβούμε σε κάποιες αφαιρέσεις ώστε το μοντέλο να απλοποιηθεί χωρίς όμως να χάσει τη ρεαλιστική του λειτουργία.

Ας δούμε μια απλοποιημένη μορφή της παραπάνω διαδικασίας βασιζόμενοι στα 8 βήματα που αναφέρθηκαν κατά τη περιγραφή. Συμβολίζουμε ως C τον πελάτη, M τον έμπορο και B το server του NetBill, ενώ με το συμβολισμό “X$\Rightarrow$Y” ότι ο X στέλνει το προκαθορισμένο μήνυμα στον Y.

1. $C \Rightarrow M$ Αίτηση τιμής
2. $M \Rightarrow C$ Συμφωνία τιμής
3. $C \Rightarrow M$ Αίτηση αγαθών
4. $M \Rightarrow C$ Αποστολή αγαθών κρυπτογραφημένα με κλειδί K
5. $C \Rightarrow M$ Υπογεγραμμένη φόρμα ηλεκτρονικής παραγγελίας (*EPO*)
6. $M \Rightarrow B$ Αποστολή φόρμας ηλεκτρονικής παραγγελίας (περιλαμβάνοντας το κλειδί *K*)

7. $B \Rightarrow M$ Υπογεγραμμένο αποτέλεσμα (περιλαμβάνοντας το κλειδί K σε περίπτωση επιτυχούς πληρωμής)
8. $M \Rightarrow C$ Υπογεγραμμένο αποτέλεσμα (περιλαμβάνοντας το κλειδί K σε περίπτωση επιτυχούς πληρωμής)

Αύτη τη μορφή εμείς θα μοντελοποιήσουμε. Το μοντέλο που προτείνουμε περιέχει τοπικά σφάλματα στις διαδικασίες του πελάτη και του εμπόρου και μη αξιόπιστη επικοινωνία μεταξύ των συμμετεχόντων στη συναλλαγή περιλαμβάνοντας και πιθανά προβλήματα στη μετάδοση των μηνυμάτων. Υποθέτουμε επίσης ότι ταυτόχρονα γίνεται η χρέωση στο λογαριασμό του πελάτη και η προσθήκη χρημάτων στο λογαριασμό του εμπόρου στο ίδιο μέρος (NetBill Server) και ενώ παρέχονται και στοιχειώδεις εγγυήσεις ατομικότητας της συναλλαγής. Συνεπώς παραβλέπουμε τοπικά λάθη στο server του NetBill,καθώς αυτά θα δυσκολέψουν το πρωτόκολλο μας και θα το γεμίσουν με λεπτομέρειες που δεν αποτελούν μέρος του πρωτοκόλλου NetBill αλλά έχουν σχέση με τον παρεχόμενο μηχανισμό επεξεργασίας της συναλλαγής. Αυτό συνεπάγεται με την ατομικότητα των χρημάτων, την οποία θα δείξουμε πως θα την αποδείξουμε εκμεταλλευόμενοι τις παρεχόμενες συναρτήσεις εξερεύνησης του χώρου καταστάσεων και την βιβλιοθήκη CTL (Computation Tree like temporal logic).

Στην αρχική σελίδα (σχήμα 4.2) παρουσιάζεται η γενική εικόνα του μοντέλου στην ανώτερη ιεραρχική σελίδα όπου αναπαρίστανται οι τρεις συμμετέχοντες καθώς και τα μηνύματα που ανταλλάσουν. Επίσης παρακάμπτονται τα 2 πρώτα βήματα καθώς δεν παίζουν κάποιο ρόλο στη τεκμηρίωση των ιδιοτήτων που θα κάνουμε. Το μοντέλο μας πρέπει να περιέχει όλα τα πιθανά σενάρια εκτέλεσης του πρωτοκόλλου. Θα χρησιμοποιήσουμε μια τροποποιημένη παρουσίαση όλων των περιπτώσεων απόρριψης της συναλλαγής ορίζοντας τις ως μάρκες τύπου request οι οποίες αντιστοιχούν στα παρακάτω σενάρια εκτέλεσης:

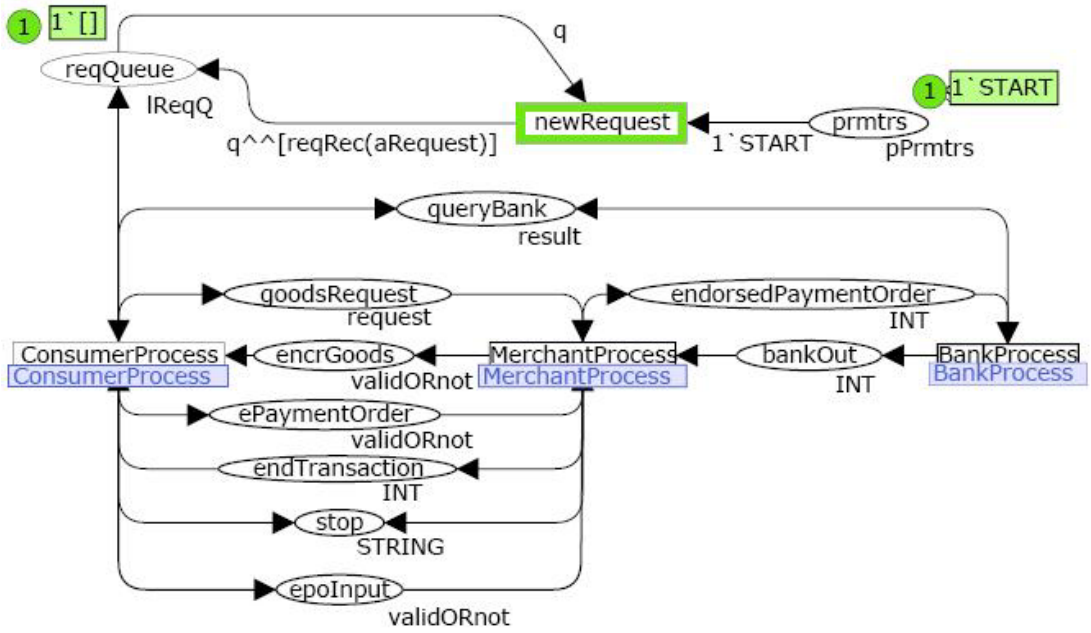
1. Ο C στέλνει στον M μια έγκυρη αίτηση αγαθών (gReq=v)
2. Ο C στέλνει στον M μια μη έγκυρη αίτηση αγαθών (gReq=v)
3. τα κρυπτογραφημένα αγαθά που λαμβάνονται από τον C είναι τα επιθυμητά (enGoods=v)
4. τα κρυπτογραφημένα αγαθά που λαμβάνονται από τον επηρεάζονται από κάποιο λάθος στη μετάδοση (enGoods=i)

5. Ο C στέλνει στον M μια έγκυρη φόρμα ηλεκτρονικής (epoReq=v)

6. Ο C στέλνει στον M μια μη έγκυρη φόρμα ηλεκτρονικής (epoReq=i)

```
colset pPrmtrs      =with START;
colset validORnot   =with v | i;
colset request      =record
                     gReq:validORnot*enGoods:validORnot*epoReq:validORnot;
colset sRequest     =union reqRec:request;
colset lReqQ        =list sRequest;
colset result       =with noFunds | paymentReceipt | noRecord;
var aRequest        :request;
var q               :lReqQ;
var valCode         :validORnot;
var intVar          :INT;
var res             :result;
```

Σχήμα 4.1 : Οι δηλώσεις των τύπων που χρησιμοποιεί το μοντέλο

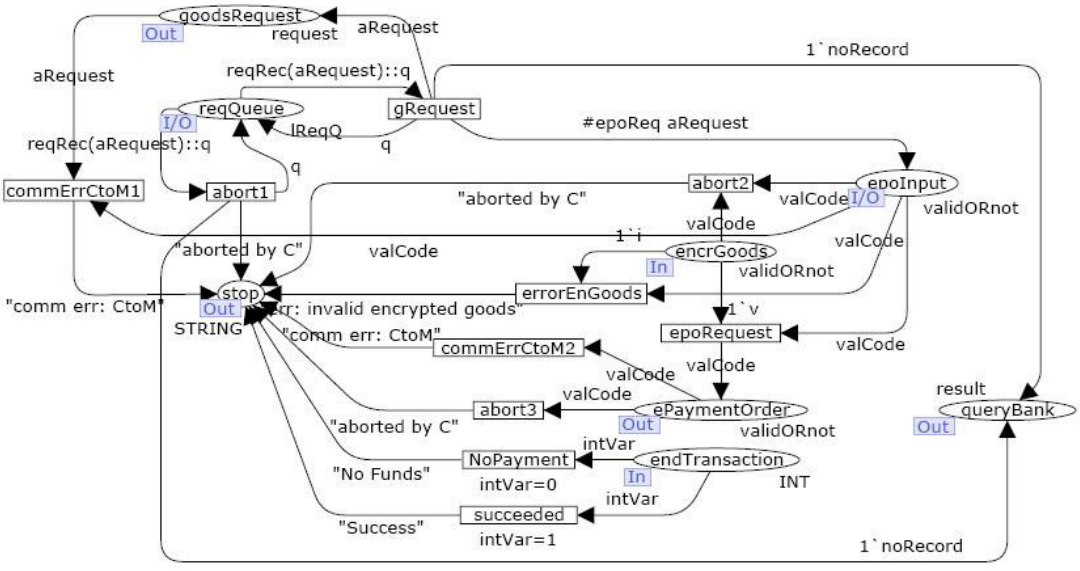


Σχήμα 4.2 Ανώτερη ιεραρχική σελίδα

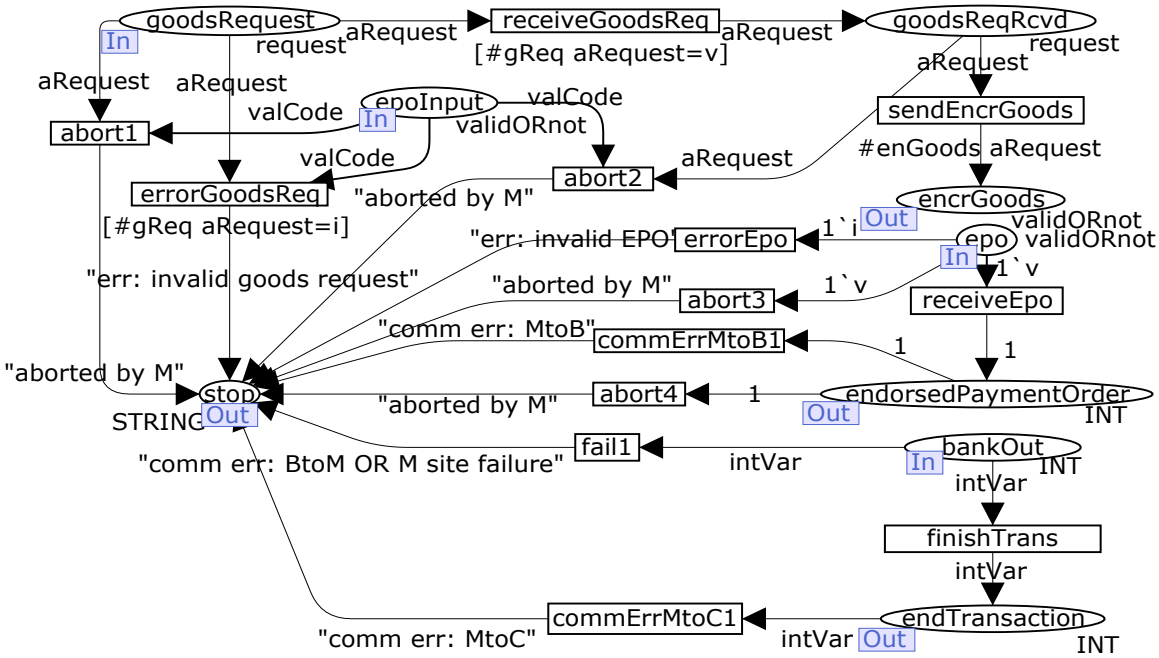
Μια φόρμα ηλεκτρονικής πληρωμής δεν είναι έγκυρη όταν δεν είναι υπογεγραμμένη ή περιέχει μη έγκυρες καταχωρήσεις ,όπως για παράδειγμα την καταχώρηση του checksum ενός προϊόντος το οποίο μπορεί να είναι διαφορετικό σε σχέση με τη κατάλληλη συμβολοσειρά αναγνώρισης η οποία υπάρχει στην θέση output stop.Μονομερείς απορρίψεις του μοντέλου επίσης αναπαρίστανται με μεταβάσεις όπως για

παράδειγμα η errorEnGoods η οποία αντιστοιχεί στις ενέργειες επικύρωσης που πραγματοποιούνται από τους συμμετέχοντες στο μοντέλο. Ο τερματισμός της ανταλλαγής μηνυμάτων μεταξύ πελάτη και εμπόρου τερματίζεται με κάποιο μήνυμα όπως π.χ. “comm err:failed to report end of transaction”, “Success” or “No Funds” το οποίο εμφανίζεται στη θέση stop,το οποίο στην ουσία είναι μια θέση στην οποία εμφανίζονται όλα τα μηνύματα που εξηγούν το λόγο αποτυχίας του μοντέλου. Σε περίπτωση τοπικής αποτυχίας, δηλαδή αποτυχία στο τρόπο που λειτουργεί το πρωτόκολλο του εμπόρου ή αποτυχίας στην επικοινωνία μεταξύ των συμμετεχόντων στη συναλλαγή ο πελάτης ενημερώνεται για το αποτέλεσμα της συναλλαγής στέλνοντας ερώτημα στο server.Το αποτέλεσμα εμφανίζεται στη θέση queryBank της top-level ιεραρχικής σελίδας.

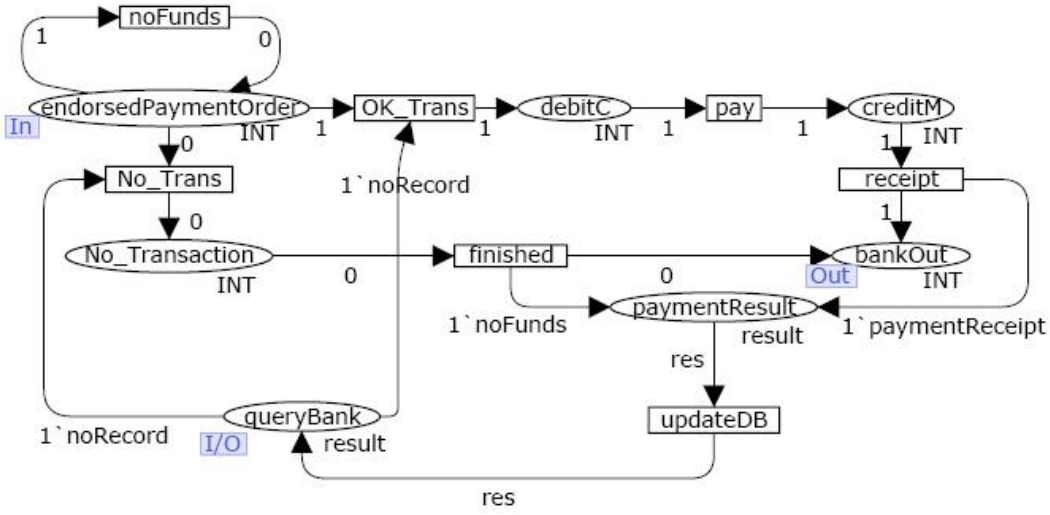
Στο μοντέλο (σχήμα 4.2) όπως προαναφέραμε υπάρχουν 3 υποσελίδες η οποίες περιγράφουν τι γίνεται ξεχωριστά σε κάθε ένα από τους τρεις συμμετέχοντες στην συναλλαγή. Στο σχήμα 4.3 παρουσιάζεται η υποσελίδα του πελάτη, δηλαδή το ConsumerProcess η οποία αντιστοιχεί στη συνώνυμη της στην ανώτερη ιεραρχική σελίδα. Κάθε θέση που είτε είναι in(put) είτε είναι out(put) εμφανίζεται και στην ανώτερη ιεραρχική σελίδα του μοντέλου. Επειδή επικοινωνούν μεταξύ τους έχουν και τα ίδια ονόματα με τις αντίστοιχες στην ανώτερη ιεραρχική σελίδα. Έτσι παραδείγματος χάριν η θέση stop στην υποσελίδα ConsumerProcess είναι ίδια με τη θέση stop στην ανώτερη ιεραρχική σελίδα του μοντέλου. Παρομοίως λειτουργούν και οι άλλες δύο υποσελίδες του μοντέλου οι οποίες απεικονίζονται στα σχήματα 4.4 , 4.5.



Σχήμα 4.3 Υποσελίδα ConsumerProcess



Σχήμα 4.4 Υποσελίδα MerchantProcess



Σχήμα 4.5 Υποσελίδα BankProcess

Στην υποσελίδα ConsumerProcess η πυροδότηση της μετάβασης gRequest τοποθετεί τη μάρκα noRecord με τύπο result στη θέση queryBank. Αυτό είναι η μοντελοποίηση της πιθανότητας ο πελάτης να υποβάλει ερωτήματα στη τράπεζα για το αποτέλεσμα της συναλλαγής. Η μάρκα aRequest με τύπο request μεταφέρεται στη θέση goodsRequest και μετά χρησιμοποιείται μη ντετερμινιστικά για να πυροδοτήσει είτε τη

μετάβαση commErrCtomM1 είτε τη μετάβαση που αντιστοιχεί στην μεταβίβαση της μάρκας στην υποσελίδα του εμπόρου.

Υπενθυμίζουμε ότι ο πελάτης μπορεί να απορρίψει τη συναλλαγή μέχρι να στείλει τη φόρμα EPO. Εν δυνάμει μονομερείς αποφάσεις ματαίωσης από τον πελάτη και τοπικές καταρρεύσεις του συστήματος στην υποσελίδα του μοντελοποιούνται από τις μεταβάσεις με όνομα abort# και τερματίζουν το πρωτόκολλο τοποθετώντας μια συμβολοσειρά στη θέση stop η οποία επεξηγεί τι έχει οδηγήσει στο τερματισμό της συναλλαγής. Μονομερείς ματαιώσεις του μοντέλου επίσης αναπαρίστανται με μεταβάσεις όπως για παράδειγμα η errorEnGoods η οποία αντιστοιχεί στις ενέργειες επικύρωσης που πραγματοποιούνται από τους συμμετέχοντες στο μοντέλο. Αυτή λοιπόν είναι μια περιγραφή του μοντέλου του NetBill.Τώρα θα περάσουμε στη δεύτερη φάση όπου θα εξετάσουμε τις ιδιότητες του μοντέλου.

4.3 Έλεγχος ιδιοτήτων του μοντέλου

Τα CPN tools παρέχουν τα κατάλληλα εργαλεία έτσι ώστε στο εξειδικευμένο γραφικό τους περιβάλλον να επιτρέπουν μια βήμα προς βήμα προσομοίωση του μοντέλου. Σε κάθε βήμα ο αναλυτής διαλέγει ποια μετάβαση να ενεργοποιήσει (παράδειγμα στο σχήμα 4.3 οι μεταβάσεις που είναι μαρκαρισμένες εσωτερικά με πράσινο χρώμα) και αυτό έχει σαν αποτέλεσμα διαφορετική κατανομή των μαρκών. Ο έλεγχος του μοντέλου βασίζεται στη δημιουργία του γραφήματος εκτέλεσης που παριστά όλες τις δυνατές καταστάσεις του μοντέλου. Αυτός ο τρόπος αντιμετώπισης οδηγεί σε μικρού μεγέθους διαστήματα καταστάσεων αλλά απαιτεί να γίνει έλεγχος μοντέλου για κάθε δυνατό σενάριο. Σε πολλές περιπτώσεις μάλιστα απαιτείται και η χρήση του γραφήματος για τα ισχυρά συνεκτικά στοιχεία (strongly connected components).Θα ξεκινήσουμε τον έλεγχο του μοντέλου χρησιμοποιώντας τα standard εργαλεία που μας παρέχει το CPN tools και τα οποία βρίσκονται στο state space του Toolbox.

```
Statistics
----------------------------------------------------------------------
  State Space
    Nodes:  59
    Arcs:   103
    Secs:   0
    Status: Full

  Scc Graph
    Nodes:  59
    Arcs:   103
    Secs:   0

Boundedness Properties
----------------------------------------------------------------------
  Best Integers Bounds                Upper       Lower
  BankProcess'No_Transaction 1        1           0
  BankProcess'creditM 1               1           0
  BankProcess'debitC 1                1           0
  BankProcess'paymentResult 1         1           0
  ConsumerProcess'epoInput 1          1           0
  MerchantProcess'goodsReqRcvd 1      1           0
  Protocol'bankOut 1                  1           0
  Protocol'ePaymentOrder 1            1           0
  Protocol'encrGoods 1                1           0
  Protocol'endTransaction 1           1           0
  Protocol'endorsedPaymentOrder 1     1           0
  Protocol'goodsRequest 1             1           0
  Protocol'prmtrs 1                   1           0
  Protocol'queryBank 1                1           0
  Protocol'reqQueue 1                 1           1
  Protocol'stop 1                     1           0

Home Properties
----------------------------------------------------------------------
  Home Markings:  None

Liveness Properties
----------------------------------------------------------------------
  Dead Markings:  13 [59,658,57,56,55,...]
  Dead Transitions Instances: None
  Live Transitions Instances: None

Fairness Properties
----------------------------------------------------------------------
  No infinite occurrence sequences.
```

Πίνακας 4.1
Occurrence graphic standard analysis report

Στο πίνακα 4.1 υπάρχουν όλα τα αποτελέσματα τα οποία προκύπτουν από τον έλεγχο μοντέλου για το χώρο καταστάσεων. Βλέπουμε στην αναφορά ότι το μοντέλο έχει πολύ μικρό χώρο καταστάσεων, η αναφορά δημιουργήθηκε σε κλάσματα του

δευτερολέπτου ενώ δεν υπάρχουν οικεία μαρκαρίσματα (home markings) διαρκείς και νεκρές μεταβάσεις (live and dead transitions) .Οι αναγνωρισμένες περιπτώσεις ως νεκρά μαρκαρίσματα δεν ενεργοποιούνται ποτέ εξαιτίας του επιλεγμένου σεναρίου που εκτελείται στο μοντέλο.

Εφόσον έχουμε τελειώσει με τον τυπικό έλεγχο του μοντέλου μπορούμε να περάσουμε στο μη τυπικό όπου εμείς μπορούμε να δημιουργήσουμε τα δικά μας ερωτήματα και να τα υποβάλουμε στο CPN βασιζόμενοι σε απλό προγραμματισμό στη γλώσσα ML.Οι συναρτήσεις που χρησιμοποιούνται στο μοντέλο εμφανίζονται στο πίνακα 4.2. Η συνάρτηση SearchNodes χρησιμοποιείται για να ανιχνεύει το μαρκάρισμα ακριβώς μετά από κάθε εκτέλεση ενός συγκεκριμένου γεγονότος, όπως για παράδειγμα την εκτέλεση της μεταφοράς χρημάτων από το λογαριασμό του πελάτη στο λογαριασμό του έμπορου.

Πρειγραφή συνάρτησης	Χρήση
`Mark.<PageName>'<PlaceName> N M`	Επιστρέφει το σύνολο των μαρκών τα οποία βρίσκονται στη θέση <PlaceName> στην N-οστή instance της σελίδας <PageName> στο μαρκάρισμα M

`SearchNodes (`
`<search area>,`
`<predicate function>,`
`<search limit>,`
`<evaluation function>,`
`<start value>,`
`<combination function>)`
	Διασχίζει τους κόμβους του τμήματος του γραφήματος καταστάσεων το οποίο καθορίζεται ως <search area>. Σε κάθε κόμβο ο υπολογισμός ο οποίος καθορίζεται από το <evaluation function> πραγματοποιείται και τα αποτελέσματα αυτών των υπολογισμών συνδυάζονται με τρόπο που ορίζουμε στο <combination function> για να πάρουμε το τελικό αποτέλεσμα. Το <predicate function> αντιστοιχεί κάθε κόμβο σε μία μπουλιανή τιμή και διαλέγει μόνο αυτούς τους κόμβους οι οποίοι υπολογίζονται ως αληθινοί. Χρησιμοποιούμε τη τιμή EntireGraph στη θέση <search area> για να δώσουμε όλο το σύνολο των κόμβων στο γράφημα και τι τιμή 1 στη θέση <start value> για να συνεχίσουμε το ψάξιμο μέχρι το πρώτο κόμβο ,για τον οποίο η συνάρτηση <predicate function> έχει τιμή αληθής.

`List.nth(l,n)`	Επιστρέφει το ν-οστό στοιχείο στη λίστα l, όπου 0 <= n < μήκος l.

Πίνακας 4.2 Συναρτήσεις ερωτημάτων για το χώρο καταστάσεων

Στο πίνακα 4.3 υπάρχουν οι εντολές της CTL που είναι ένας ακόμα τρόπος για να υποβάλουμε ερωτήματα στο μοντέλο. Αυτές οι εντολές χρειάζονται για να εκφράσουμε τις απαιτούμενες ιδιότητες σε σχέση με τα μονοπάτια (paths) σ' ένα δημιουργημένο

occurrence graph. Ένα μονοπάτι είναι μια ακολουθία από εκτελέσεις καταστάσεων και μεταβάσεων, δηλαδή ένα πέρασμα μέσα από το χώρο των καταστάσεων το οποίο κατευθύνεται από τα τόξα του μοντέλου. Ο τρόπος ελέγχου της CTL για την διαπίστωση της απαιτούμενης ιδιότητας γίνεται με τη χρήση της συνάρτησης eval_node η οποία ξεκινάει από ένα συγκεκριμένο κόμβο (node) ο οποίος ορίζεται μέσα στη συνάρτηση. Στον Πίνακα 4.4 παραθέτουμε το τρόπο με τον οποίο ενεργοποιούμε μέσα από τα CPN Tools τη CTL.

Σύνταξη state formulae	Επεξήγηση
NOT(A)	Μπουλιανή τιμή η οποία αντιστοιχεί στην άρνηση του A , όπου το A είναι μια CTL formula.
AND(A_1,A_2)	Αυτή η φόρμουλα είναι αληθής αν και το A_1 και το A_2 είναι αληθείς.
NF(<message>,<node function>)	Μια συνάρτηση η οποία χρησιμοποιείται συνήθως για να αναγνωρίζει single states ή ένα υποσύνολο του χώρου καταστάσεων. Χρησιμοποιεί μια συμβολοσειρά και μία συνάρτηση , η οποία δέχεται το χώρο καταστάσεων ενός κόμβου και επιστρέφει μία μπουλιανή τιμή. Η συμβολοσειρά χρησιμοποιείται όταν μια CTL φόρμουλα δίνει "ψευδής" στον ελεγκτλη μοντέλου.
EV(A)≡FORALL_UNTIL(TT,A)	Αυτή η φόρμουλα είναι αληθής αν το A γίνει αληθές τελικά (μέσα σε πεπερασμένο αριθμό βημάτων) ξεκινώντας από τη κατάσταση που βρισκόμαστε τώρα. Το TT δηλώνει την τιμή της σταθερά "αληθής".
ALONG(A)≡NOT(EV(NOT(A)))	Αυτή η φόρμουλα είναι αληθής αν υπάρχει ένα μονοπάτι για το οποίο η συνθήκη A ισχύει για κάθε κατάσταση. Το μονοπάτι είναι είτε άπειρο είτε τελειώνει σε μία αδιέξοδη κατάσταση (dead state).
POS(A)≡EXIST_UNTIL(TT,A)	Αυτή η φόρμουλα είναι αληθής αν είναι δυνατόν από τη κατάσταση που είμαστε τώρα να φτάσουμε σε μία κατάσταση που η συνθήκη A είναι αληθής.
EXIST_NEXT(A)	Αυτή η φόρμουλα είναι αληθής αν και μόνο αν υπάρχει μία άμεσα διαδοχική κατάσταση , από εκεί που είμαστε τώρα , στην οποία η συνθήκη A είναι αληθής.
eval_node <formula> <node>	Η συνήθης συνάρτηση ελέγχου μοντέλου με δύο συνθήκες: η CTL φόρμουλα πού πρέπει να ελεγχθεί και μία κατάσταση από την οποία ο έλεγχος μοντέλου πρέπει να ξεκινήσει.

Πίνακας 4.3 Τελεστές της CTL φόρμουλας και συναρτήσεις ελέγχου μοντέλου.

```
use (ogpath^"ASKCTL/BitArray.sml");
        use (ogpath^"ASKCTL/ASKCTL.sml");
        open ASKCTL;
```

Πίνακας 4.4 Ενεργοποιώντας την CTL

Τώρα θα παραθέσουμε τα ερωτήματα με τα οποία έχουμε αποδείξει ότι το μοντέλο μας έχει τις τρεις ιδιότητες που ζητήσαμε στην αρχή:

```
fun debitDone n = (Mark.BankProcess'debitC 1 n = [1]);
val firstDebitState = List.nth(SearchNodes (
                                        EntireGraph,
                                        fn n => (debitDone n),
                                        NoLimit,
                                        fn n => n,
                                        [],
                                        op ::),0);
fun creditDone n = (Mark.BankProcess'queryBank 1 n = [paymentReceipt]);
val noDebit = NOT(NF("Double debit!",debitDone));
val creditState = NF("No credit!",creditDone);
val moneyAtomicity = FORALL_NEXT(FORALL_UNTIL(noDebit,creditState));
eval_node moneyAtomicity firstDebitState;
```

Πίνακας 4.5. Έλεγχος ατομικότητας χρημάτων : Αληθής

```
        fun signedEPO n = (Mark.Protocol'ePaymentOrder 1 n = [v]);
val dispatchedEPOState = List.nth(SearchNodes (
                                        EntireGraph,
                                        fn n => (signedEPO n),
                                        NoLimit,
                                        fn n => n,
                                        [],
                                        op ::),0);
fun debitDone n = (Mark.BankProcess'debitC 1 n = [1]);
fun noTrans n = (Mark.Protocol'queryBank 1 n <>[paymentReceipt]);
val debitState = NF("No debit!",debitDone);
val notRegisteredDecrKey = NF("Found decryption key!",noTrans);
val noGoodsAtomicityA = ALONG(AND(EV(debitState),notRegisteredDecrKey));
eval_node noGoodsAtomicityA dispatchedEPOState;


fun sendEPO n = (Mark.Protocol'ePaymentOrder 1 n <> []);
val dispatchedEPOStates = SearchNodes (
                                        EntireGraph,
                                        fn n => (sendEPO n),
                                        NoLimit,
                                        fn n => n,
                                        [],
                                        op ::);
val dispatchedEPOState1 = List.nth(dispatchedEPOStates,0);
val dispatchedEPOState2 = List.nth(dispatchedEPOStates,1);
fun noDebitDone n = (Mark.BankProcess'debitC 1 n <> [1]);
fun succeedTrans n = (Mark.Protocol'queryBank 1 n =[paymentReceipt]);
val noDebitFound = NF("Debit found!",noDebitDone);
val registeredDecrKey = NF("Failed transaction!",succeedTrans);
val noGoodsAtomicityB = ALONG(AND(EV(registeredDecrKey),noDebitFound));
eval_node noGoodsAtomicityB dispatchedEPOState1;
eval_node noGoodsAtomicityB dispatchedEPOState2;
```

Πίνακας 4.6. Έλεγχος της μη-ατομικότητας της παράδοσης των αγαθών : Ψευδής

```
fun registerKeyState n = (Mark.Protocol'queryBank 1 n = [paymentReceipt]);
val registerKey = POS(EV(NF("No paymentReceipt!",registerKeyState)));
fun enGoodsTransferedState n = (Mark.Protocol'encrGoods 1 n = [v]);
val noGoods = NOT(POS(EV(NF("Encr goods sent!",enGoodsTransferedState))));
val nonCertifiedDelivery = EXIST_NEXT(AND(noGoods,registerKey));
eval_node nonCertifiedDelivery InitNode;
```

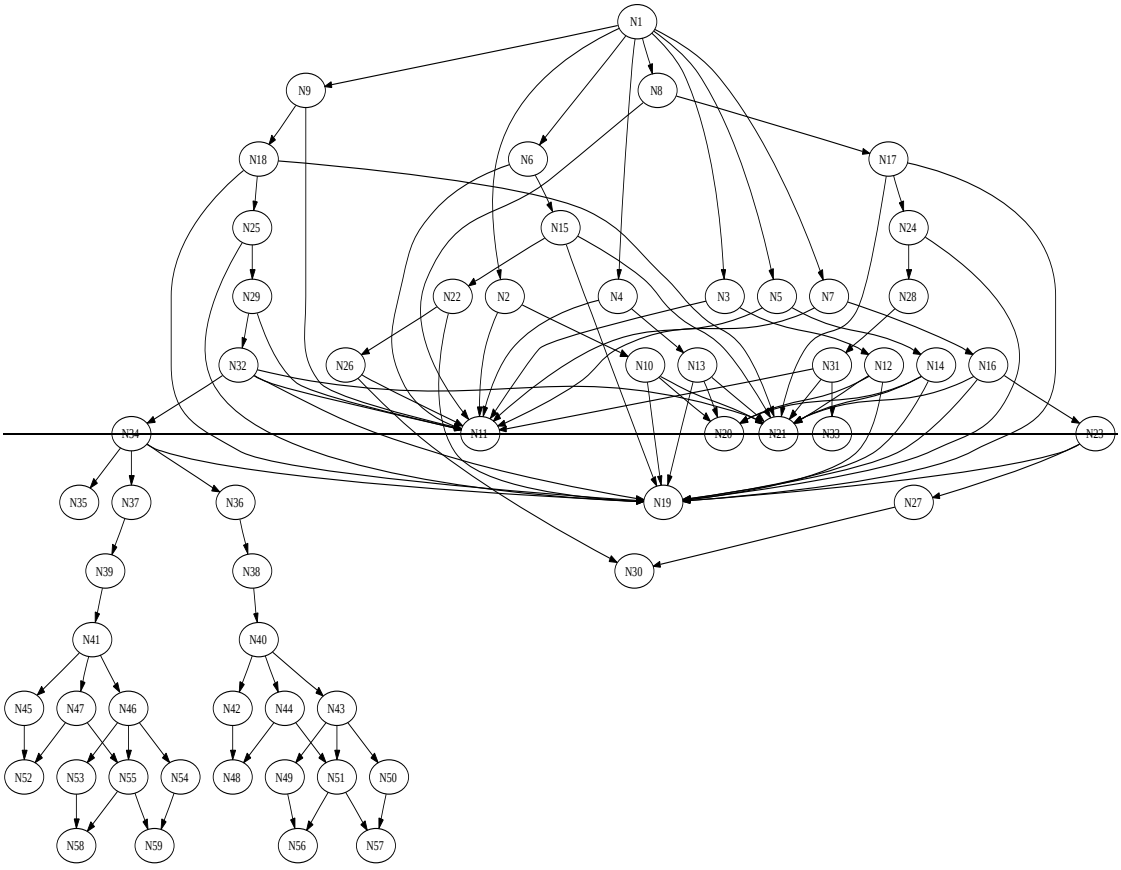
Πίνακας 4.7. Έλεγχος μη εγγυημένης παραλαβής : Ψευδής

Θα περιγράψουμε συνοπτικά τι γίνεται σε κάθε ένα από τα παραπάνω ερωτήματα. Στο πίνακα 4.5 μας ενδιαφέρει να δείξουμε ότι ανεξάρτητα από αποτυχίες στην επικοινωνία ή απόρριψη της συναλλαγής το πρωτόκολλο δεν δημιουργεί χρήματα: για όλα τα μονοπάτια ξεκινώντας από την εκτέλεση της χρέωσης του πελάτη το πρωτόκολλο βάζει τα χρήματα αυτά στον έμπορο σ' ένα πεπερασμένο αριθμό βημάτων. Στο πίνακα 4.6 μας ενδιαφέρει να δείξουμε ότι το μοντέλο τελειώνει πάντα με την ένδειξη "Success" ξεκινώντας από την εκτέλεση μιας επιτυχούς μεταφοράς χρημάτων ανεξάρτητα από αποτυχίες στην επικοινωνία ή τη μονομερής απόρριψη της συναλλαγής. Στο 4.7 τέλος μας ενδιαφέρει να δείξουμε ανεξάρτητα από αποτυχίες στην επικοινωνία ή απόρριψη της συναλλαγής ότι δεν είναι δυνατόν να τελειώσει μια επιτυχής συναλλαγή χωρίς ο πελάτης να μην έχει λάβει τη κρυπτογραφημένη μορφή των προϊόντων μαζί με ένα κρυπτογραφημένο checksum.Το αποτέλεσμα είναι ψευδές και άρα ισχύει τελικά η ιδιότητα. Σε περίπτωση που η CTL βγάλει ψευδές αποτέλεσμα μπορούμε να χρησιμοποιήσουμε τη συμβολοσειρά που περιλαμβάνεται στο πρώτο μέρος της συνάρτησης NF ώστε να μας τυπώσει ποιος είναι ο λόγος που είναι ψευδές το αποτέλεσμα.

4.4 Ανάλυση αποτυχίας πρωτοκόλλου

Η ανάλυση αποτυχίας του πρωτοκόλλου (Protocol Failure Analysis) στοχεύει στην αναγνώριση ξεχωριστών σεναρίων ματαίωσης και στη διαβεβαίωση ότι το πρωτόκολλο είναι ανθεκτικό στην ματαίωση για κάθε ένα σενάριο. Αν και στη περίπτωση μας έχουμε εισάγει στο μοντέλο περιπτώσεις ματαίωσης είτε στις επικοινωνίες είτε περιπτώσεις κατάρρευσης των επιμέρους συστημάτων συχνά χρειάζεται να επεκτείνουμε το μοντέλο με επιπρόσθετα σενάρια ματαίωσης τα οποία πιθανώς να παραβιάζουν τις ιδιότητες του μοντέλου. Η ανάλυση αποτυχίας λοιπόν στοχεύει στην διόρθωση του πρωτοκόλλου χωρίς

να παραβιάζει τις επιθυμητές του ιδιότητες. Αυτό θα το κάνουμε με έλεγχο του γραφήματος. Όπως έχουμε ήδη αναφέρει ο έλεγχος μοντέλου αυτού του είδους οδηγεί σε μικρά γραφήματα καταστάσεων τα οποία μπορούν να απεικονιστούν και σε μια σελίδα A4.Το CPN Tools εξάγει το γράφημα σε ένα αρχείο κειμένου βασισμένο στη γλώσσα DOT και το οποίο μπορεί να απεικονιστεί από το κατάλληλο πρόγραμμα. Στο σχήμα 4.6 έχουμε την απεικόνιση του γραφήματος του μοντέλου. Λόγω περιορισμένου χώρου έχουμε αντικαταστήσει κάθε ονομασία θέσης με το συμβολισμό Ni όπου i $\in$ 1,2,..,59 όσοι δηλαδή και οι κόμβοι του μοντέλου.Η ανάλυση αποτυχίας βασίζεται σε ερωτήσεις για τη κατανομή των μαρκών στα νεκρά μαρκαρίσματα (αν υπάρχουν αυτά) ή όταν το μοντέλο περιέχει διαρκείς μεταβάσεις (όπως στην περίπτωσή μας) για τα μαρκαρίσματα από τα οποία δεν είναι δυνατόν να φτάσουμε άλλα.



Σχήμα 4.6

Η συνάρτηση ListDeadMarkigs() μας δείχνει τα νεκρά μαρκαρίσματα του μοντέλου μας.

```
ListDeadMarkigs() -> val it =[59,58,57,56,55,
                              51,35,33,30,21,
                              20,19,10]:Node list
```

Τέλος ο πίνακας 4.8 δείχνει τις τιμές των μαρκών που βρίσκονται στις θέσεις Protocol`stop και Protocol`queryResult και ο οποίος δείχνει περιληπτικά τη συμπεριφορά του μοντέλου για όλα τα μαρκαρίσματα που μας ενδιαφέρουν.

marking (N)	Mark.Protocol' stop 1 N	Mark.Protocol' queryBank 1 N	Επεξήγηση
59	["No Funds"]	[noFunds]	Καμία αποτυχία.
58	["comm err: MtoC"]	[noFunds]	Ματαίωση επικοινωνίας: ο **M** αποτυγχάνει να ενημερώσει τον **C** για το αποτέλεσμα της συναλλαγής. Ο **C** ενημερώνεται για το αποτέλεσμα της συναλλαγής ρωτώντας τον **B** για το αποτέλεσμα της συναλλαγής .
57	["Success"]	[paymentReceipt]	Καμία αποτυχία.
56	["comm err: MtoC"]	[paymentReceipt]	Αποτυχία επικοινωνίας: ο **M** αποτυγχάνει να ενημερώσει τον **C** για το αποτέλεσμα της συναλλαγής. Ο **C** λαμβάνει το κλειδί αποκρυπτογράφησης του προϊόντος υποβάλλοντας ερώτημα στον **B**.
55	["comm err: BtoM or M site failure"]	[noFunds]	Αποτυχία επικοινωνίας: Ο **M** δεν ενημερώνεται για το αποτέλεσμα της συναλλαγής εξαιτίας μιας εν δυνάμει τοπικής ή επικοινωνιακής αποτυχίας. Ο **C** πληροφορείτε για το αποτέλεσμα της συναλλαγής ρωτώντας το **B**.
51	["comm err: BtoM or M site failure"]	[paymentReceipt]	Αποτυχία επικοινωνίας: Ο **M** δεν πληροφορείται για το αποτέλεσμα της συναλλαγής εξαιτίας μιας εν δυνάμει τοπικής ή επικοινωνιακής αποτυχίας. Ο **C** λαμβάνει το κλειδί αποκρυπτογράφησης του προϊόντος υποβάλλοντας ερώτημα στον **B**.
35	["comm err: MtoB"]	[noRecord]	Αποτυχία επικοινωνίας: η υπογεγραμμένη εντολή φόρμα πληρωμής δεν στέλνεται στο **B**.Ο **C** πληροφορείται ότι δεν έγινε καμία συναλλαγή υποβάλλοντας ερώτημα **B**.
33	["err: invalid EPO"]	[noRecord]	Ο **M** απορρίπτει τη συναλλαγή εξαιτίας μίας μη έγκυρης ***epo***.Ο **C** πληροφορείται ότι δεν έχει γίνει καμία συναλλαγή υποβάλλοντας ερώτημα στον **B**.
30	["err: invalid encrypted goods"]	[noRecord]	Ο **C** απορρίπτει τη συναλλαγή επειδή έχει λάβει αγαθά τα οποία είναι επηρεασμένα από κάποιο σφάλμα στην μετάδοση τους.

marking (N)	Mark.Protocol' stop 1 N	Mark.Protocol' queryBank 1 N	Επεξήγηση
21	["comm err: CtoM"]	[noRecord]	Αποτυχία επικοινωνίας: η αίτηση αγαθών ή η υπογεγραμμένη φόρμα πληρωμής δεν στέλνεται στο M. Ο C πληροφορείται ότι δεν υπάρχει καμία συναλλαγή υποβάλλοντας ερώτημα στον B.
20	["err: invalid goods request"]	[noRecord]	Ο M απορρίπτει τη συναλλαγή εξαιτίας μιας μη-έγκυρης αίτησης αγαθών.
19	["aborted by M"]	[noRecord]	Ο M απορρίπτει τη συναλλαγή εξαιτίας ενός εν δυνάμει τοπικού σφάλματος η μίας μονομερούς απόφασης απόρριψης Ο C πληροφορείται ότι δεν έχει γίνει καμία συναλλαγή υποβάλλοντας ερώτημα στον B. Ο C πληροφορείται ότι δεν έχει γίνει καμία συναλλαγή υποβάλλοντας ερώτημα στον B.
10	["aborted by C"]	[noRecord]	Ο C απορρίπτει τη συναλλαγή εξαιτίας ενός εν δυνάμει τοπικού σφάλματος η μίας μονομερούς απόφασης απόρριψης πριν δεσμευτεί στέλνοντας την υπογεγραμμένη φόρμα πληρωμής.

Πίνακας 4.8

5.Επίλογος

Έχουμε δείξει λοιπόν ότι μπορούμε να ελέγχουμε πρωτόκολλα ηλεκτρονικού εμπορίου για συγκεκριμένες ιδιότητες που θέλουμε να ικανοποιούν με τη χρήση των χρωματισμένων δικτύων Petri και μάλιστα με ένα πολύ ευέλικτο και φιλικό προς τον χρήστη λογισμικό το οποίο είναι πολύ πιο μοντέρνο και αποτελεσματικό σε σχέση με μεθόδους που χρησιμοποιούνταν ως τώρα ,όπως π.χ. τα προγράμματα FDR και Spin.H χρήση των CP-Nets πιστεύουμε ότι είναι πολύτιμη στη μελέτη πιο πολύπλοκων συστημάτων και στη διερεύνηση άλλων ιδιοτήτων που έχουν σχέση με την αξιοπιστία και την ασφάλεια ενός πρωτοκόλλου. Τέλος προτείνουμε τη χρήση ελέγχου αποτυχίας του πρωτοκόλλου για να διαπιστώσουμε τα εν δυνάμει σενάρια παραβίασης του πρωτοκόλλου και να τα αντιμετωπίσουμε. Εν κατακλείδι πιστεύουμε ότι τα CPN Tools είναι ένα πολύ πιο ελκυστικό εργαλείο έναντι των μοντέλων που ελέγχονται με τα CSP.

Βιβλιογραφία (References)

1. Tygar J.D. Atomicity in electronic commerce
2. Jensen K. An introduction to the theoretical aspects of colored Petri Nets, In: a decade of Concurrency, LNCS 803 , 1994, 230-272
3. Katsaros P. , Odontidis V., Gousidou-Koutita M. Colored Petri Net Based model checking and failure analysis for e-commerce protocols
4. Αβραμίδου Ευτέρπη , Μοντελοποίηση και Προσομοίωση με Χρωματισμένα Δίκτυα Πτυχιακή εργασία

Δικτυακοί τόποι (Sites)

www.internetindicators.com
http://www.semper.org/sirene/outsideworld/ecommerce.html