



Πανεπιστήμιο Πελοποννήσου
Σχολή Θετικών Επιστημών και Τεχνολογίας
Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών

Μεταπτυχιακή Εργασία

**Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα
πλαίσια του ηλεκτρονικού εμπορίου**

Ονοματεπώνυμο Φοιτητή : Φούντας Γεώργιος

ΑΜ : ΠΜΣ2010028

Επιβλέπων Καθηγητής: Βασιλάκης Κωνσταντίνος

Τρίπολη, Μάρτιος 2012

Copyright © 2012

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών της Σχολής Θετικών Επιστημών και Τεχνολογίας του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Περίληψη

Στην παρούσα εργασία μελετώνται τέσσερις προσεγγίσεις για την ανάπτυξη εφαρμογών με επίγνωση της περιβάλλουσας κατάστασης. Για τις μελετώμενες προσεγγίσεις, εξετάζεται ο βασικός στόχος της κάθε μίας και οι αδυναμίες που προσπαθεί να καλύψει, παρουσιάζονται οι βασικές έννοιες της προσέγγισης και περιγράφεται η μεθοδολογία ανάπτυξης των εφαρμογών που προτείνεται και τέλος δίνεται ένα παράδειγμα ανάπτυξης εφαρμογής με βάση τη μεθοδολογία.

Abstract

The present thesis reviews four approaches to the development of context-aware applications. For each of these approaches, we examine the basic goals of the approach and the issues it tries to tackle. Afterwards, we present the basic concepts used in the approach and the methodology for application development it proposes. Finally, for each approach, an application development example is given.

Περιεχόμενα

<i>Περίληψη</i>	3
<i>Abstract</i>	4
<i>Περιεχόμενα</i>	5
<i>Ευρετήριο εικόνων</i>	8
Κεφάλαιο 1 <i>Εισαγωγή</i>	10
Κεφάλαιο 2 <i>Model-driven Development of Context-Aware Web Applications</i>	11
2.1 Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει	13
2.2 Βασικές έννοιες της προσέγγισης	14
2.2.1 Λειτουργικές απαιτήσεις συστήματος	14
2.2.2 Αρχιτεκτονικές απαιτήσεις	15
2.3 Μεθοδολογία ανάπτυξης εφαρμογών σύμφωνα με την προσέγγιση	16
2.3.1 Παρουσίαση WebML	16
2.3.2 Μοντελοποίηση περιβάλλουσας κατάστασης ως δεδομένα	18
2.3.3 Δόμηση υπερκειμένου το οποίο διαθέτει επίγνωση της περιβάλλουσας κατάστασης	20
2.3.3.1 Σελίδες με επίγνωση της περιβάλλουσας κατάστασης	20
2.3.3.2 Πολιτικές προσαρμοστικότητας	21
2.3.3.3 Νέφη περιβάλλουσας κατάστασης	22
2.3.3.4 Περιέκτες με επίγνωση της περιβάλλουσας κατάστασης	23
2.3.4 Καθορισμός προσαρμοστικότητας	24
2.3.4.1 Διαχείριση δεδομένων περιβάλλουσας κατάστασης	24
2.3.4.2 Αποτίμηση των συνθηκών	25
2.3.4.3 Εκτέλεση ενεργειών προσαρμοστικότητας	25
2.4 Παράδειγμα ανάπτυξης εφαρμογής σύμφωνα με την προσέγγιση	27
2.4.1 Υπολογισμός σελίδων με επίγνωση της περιβάλλουσας κατάστασης	30
2.4.1.1 Κανόνες ειδίκευσης	31
2.4.1.2 Υπολογισμός σελίδων με επίγνωση της περιβάλλουσας κατάστασης	34
Κεφάλαιο 3 <i>Integrated Modelling of Context-Aware Adaptive Software Systems</i>	37
3.1 Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει	37
3.1.1 Αδυναμίες που προσπαθεί να καλύψει η προσέγγιση	37
3.1.2 Στόχοι της προσέγγισης	38
3.1.3 Καινοτομίες της προσέγγισης	38
3.2 Βασικές έννοιες της προσέγγισης	39
3.2.1 Context-aware Adaptive Systems Component Model (CAMP)	39

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

3.2.2	Context-aware Adaptive Software Systems Meta-Model _____	41
3.2.3	Διαχειριζόμενες οντότητες προσαρμοστικών συστημάτων με επίγνωση της περιβάλλουσας κατάστασης _____	42
3.2.4	Το μετά-μοντέλο προσαρμοστικού συστήματος με επίγνωση της περιβάλλουσας κατάστασης _____	43
3.3	Μεθοδολογία ανάπτυξης εφαρμογών σύμφωνα με την προσέγγιση __	45
3.3.1	Η κύρια οθόνη του CAST _____	45
3.3.2	Οι κλάσεις Java της εφαρμογής _____	46
3.3.3	Οι είσοδοι και έξοδοι του εργαλείου _____	49
3.3.3.1	Η XML περιγραφή του μοντέλου περιβάλλουσας κατάστασης _____	49
3.3.3.2	Η περιγραφή XML του συστήματος _____	49
3.3.3.3	Η περιγραφή XML της διαχείρισης του συστήματος _____	50
3.3.3.4	Οι έξοδοι του εργαλείου _____	51
3.4	Παράδειγμα ανάπτυξης εφαρμογής σύμφωνα με την προσέγγιση __	51
3.4.1	Τα βήματα της ανάπτυξης του προσαρμοστικού συστήματος λογισμικού επίγνωσης της περιβάλλουσας κατάστασης, χρησιμοποιώντας την προσέγγιση _____	53
3.4.2	Δημιουργία περιγραφών XML για το προσαρμοστικό σύστημα εύρεσης διαδρομής _____	54
3.4.3	Η χρήση του εργαλείου για την ανάπτυξη ενός προσαρμοστικού συστήματος σχεδίασης διαδρομής επίγνωσης της περιβάλλουσας κατάστασης _____	57
Κεφάλαιο 4 Μία προσανατολισμένη στις απόψεις προσέγγιση για την προσαρμογή των εφαρμογών κινητού ηλεκτρονικού εμπορίου _____ 60		
4.1	Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει __	60
4.1.1	Αδυναμίες που προσπαθεί να καλύψει η προσέγγιση _____	60
4.1.2	Στόχοι της προσέγγισης _____	60
4.1.3	Καινοτομίες της προσέγγισης _____	61
4.2	Βασικές έννοιες της προσέγγισης _____	61
4.3	Μεθοδολογία ανάπτυξης εφαρμογών σύμφωνα με την προσέγγιση __	62
4.3.1	Η δομή μιας διαδικτυακής εφαρμογής _____	62
4.3.2	Η μεθοδολογία σχεδιασμού μιας εφαρμογής κινητού εμπορίου _____	63
4.3.3	Τα βήματα ανάπτυξης μιας εφαρμογής κινητού εμπορίου _____	63
4.4	Παράδειγμα ανάπτυξης εφαρμογής σύμφωνα με την προσέγγιση __	65
4.4.1	Αποθετήριο πολιτικών προσαρμογής _____	67
4.4.2	The Context Consumer _____	70
4.4.3	Code Generator _____	70
4.4.4	The Adaptation Module _____	72
4.4.4.1	To Operation Adaptation Module _____	72
4.4.4.2	To Content Adaptation Module _____	72
4.4.4.3	To Presentation Adaptation Module _____	72
4.4.4.4	To Presentation Service _____	72

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

4.4.4.5	To User Interaction Module	73
Κεφάλαιο 5	Ένα πλαίσιο για τον προγραμματισμό εύρωστων εφαρμογών με επίγνωση περιβάλλουσας κατάστασης	74
5.1	Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει	75
5.1.1	Αδυναμίες που προσπαθεί να καλύψει η προσέγγιση	75
5.1.2	Στόχοι της προσέγγισης	76
5.1.3	Καινοτομίες αυτής της προσέγγισης	76
5.2	Βασικές έννοιες της προσέγγισης	76
5.2.1	Ρόλοι	77
5.2.2	Middleware (ενδιάμεσο λογισμικό)	79
5.2.3	Συμβάν περιβάλλουσας κατάστασης	80
5.2.4	Υπηρεσίες περιβάλλουσας κατάστασης	80
5.2.5	Θέματα ευρωστίας	81
5.2.5.1	Αποτυχίες στις αναδιαμορφώσεις που συμβαίνουν λόγω της περιβάλλουσας κατάστασης	81
5.2.5.2	Αποτυχίες στη σύνδεση αντικειμένων	82
5.2.5.3	Λειτουργικές αποτυχίες σε επίπεδο υπηρεσίας	83
5.2.5.4	Ακυρώσεις της περιβάλλουσας πληροφορίας	84
5.2.6	Μοντέλο ανάκαμψης από σφάλματα με χρήση συμβάντων και εξαιρέσεων	84
5.2.6.1	Μοντέλο συμβάντων στο επίπεδο του αντικειμένου και μοντέλο ανάκαμψης στο ίδιο επίπεδο	84
5.2.6.2	Μοντέλο χειρισμού εξαιρέσεων σε επίπεδο ρόλου	86
5.3	Μεθοδολογία ανάπτυξης εφαρμογών	90
5.4	Παράδειγμα ανάπτυξης εφαρμογών σύμφωνα με αυτή την προσέγγιση	90
5.4.1	Σύστημα αναπαραγωγής μουσικής	90
5.4.1.1	Σύνδεση και ανακάλυψη με βάση την περιβάλλουσα κατάσταση	91
5.4.1.2	Ρύθμιση αντικειμένου με βάση την περιβάλλουσα κατάσταση	92
5.4.2	Σύστημα πληροφοριών για ασθενείς	94
5.4.2.1	Μηχανισμός ContextGuard στο σύστημα πληροφοριών ασθενών	95
5.4.2.2	Χειρισμός εξαιρέσεων σε επίπεδο ενέργειας στο σύστημα πληροφοριών ασθενών	96
5.4.2.3	Χειρισμός εξαιρέσεων σε επίπεδο λειτουργίας στο σύστημα πληροφοριών ασθενών	96
Κεφάλαιο 6	Συμπεράσματα	98
Βιβλιογραφία		103

Ευρετήριο εικόνων

Εικόνα 1 - Διάγραμμα ροής δεδομένων της περιβάλλουσας κατάστασης.....	15
Εικόνα 2 - Σχήμα υπερκειμένου WebML	18
Εικόνα 3 - Σχεδίαση υπο-σχημάτων χρήστη, προσωποποίησης και περιβάλλουσας κατάστασης	20
Εικόνα 4 - Σχήμα υπερκειμένου WebML με δύο συμβατικές σελίδες, μία σελίδα με επίγνωση της περιβάλλουσας κατάστασης, μία περιοχή με επίγνωση της περιβάλλουσας κατάστασης, καθώς και τα σχετιζόμενα νέφη περιβάλλουσας κατάστασης	21
Εικόνα 5 - Μονάδα Get URL Parameter	25
Εικόνα 6 - Μονάδα Get Data	25
Εικόνα 7 - Μονάδα Change Site View	26
Εικόνα 8 - Μονάδα Change Style	27
Εικόνα 9 - Σχήμα δεδομένων για το σύστημα ITIS	28
Εικόνα 10 - Σχήμα υπερκειμένου για την εφαρμογή ITIS	29
Εικόνα 11 - Υπολογισμός προτύπων σελίδας επίγνωσης της περιβάλλουσας κατάστασης	33
Εικόνα 12 - Σχήμα υπερκειμένου που παρουσιάζει συμπεριφορά προσαρμοστικότητας ή απουσία προσαρμοστικότητας κατά τον χρόνο εκτέλεσης	35
Εικόνα 13 - Επικάλυψη παρωχημένων παραμέτρων συνδέσμων.....	36
Εικόνα 14 - context-aware adaptive systems component model (CAMP).....	40
Εικόνα 15 - Το μοντέλο στοιχείων, σε απλή μορφή (περίπτωση Α) και σε σύνθετη (περίπτωση Β)	41
Εικόνα 16 - Το μετά-μετά-μοντέλο μιας διαχειριζόμενης οντότητας.....	42
Εικόνα 17 - Το Μετά-Μοντέλο για προσαρμοστικά συστήματα λογισμικού με επίγνωση της περιβάλλουσας κατάστασης.....	45
Εικόνα 18 - Κύρια οθόνη του εργαλείου προσαρμοστικών συστημάτων επίγνωσης της περιβάλλουσας κατάστασης.....	46
Εικόνα 19 - Η διάταξη XML του μοντέλου περιβάλλουσας κατάστασης	49
Εικόνα 20 - Η διάταξη XML του μοντέλου περιβάλλουσας κατάστασης	50
Εικόνα 21 - Η διάταξη του μοντέλου διαχείρισης συστήματος.....	51
Εικόνα 22 - Η έξοδος του εργαλείου προσαρμοστικού συστήματος επίγνωσης της περιβάλλουσας κατάστασης.....	51
Εικόνα 23 - Διεργασία χρήσης της προσέγγισης για την ανάπτυξη προσαρμοστικών συστημάτων επίγνωσης της περιβάλλουσας κατάστασης.....	53
Εικόνα 24 - Μοντέλο στοιχείων	54

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

Εικόνα 25 - Η διάταξη XML του μοντέλου περιβάλλουσας κατάστασης	55
Εικόνα 26 - Το σχήμα XML του μοντέλου συστήματος.....	56
Εικόνα 27 - Η διάταξη XML της διαχείρισης συστήματος.....	57
Εικόνα 28 - Απεικόνιση της οθόνης του εργαλείου, μετά από τη φόρτωση των XML αρχείων	58
Εικόνα 29 - Παραγωγή του αντικειμένου του μενού του συστήματος	58
Εικόνα 30 - Ο έλεγχος της προσαρμοστικής συμπεριφοράς του συστήματος.....	59
Εικόνα 31 - Context Manager	62
Εικόνα 32 - Adaptation Manager	66
Εικόνα 33 - Request/Response Part of Configuration File.....	68
Εικόνα 34 - Μοντέλο αλληλεπίδρασης μεταξύ χρηστών και υπηρεσιών	78
Εικόνα 35 - Μοντέλα αλληλεπίδρασης	78
Εικόνα 36 - Μοντέλο συμβάντων για επίπεδο αντικειμένου	85
Εικόνα 37 - Σχέσεις μεταξύ των πεδίων ρόλου, λειτουργίας και ενέργειας	87
Εικόνα 38 - Χειρισμός εξαιρέσεων σε επίπεδο συνεδρίας.....	88
Εικόνα 39 - Δομή διεπαφής εξαίρεσης σε επίπεδο ρόλου.....	90
Εικόνα 40 - Δραστηριότητα συστήματος αναπαραγωγής μουσικής	91
Εικόνα 41 - Σύνδεση αντικειμένου MediaPlayer	92
Εικόνα 42 - Ρύθμιση αντικειμένου	93
Εικόνα 43 - Δραστηριότητα συστήματος πληροφοριών για ασθενείς.....	95
Εικόνα 44 - Παράδειγμα φρούρησης περιβάλλουσας κατάστασης.....	96
Εικόνα 45 - Παράδειγμα χειρισμού εξαιρέσεων σε επίπεδο ενεργειών.....	96
Εικόνα 46 - Χειριστής εξαιρέσεων ακύρωσης περιβάλλουσας κατάστασης	97

Κεφάλαιο 1 Εισαγωγή

Οι εξελίξεις στους τομείς της επικοινωνίας και της τεχνολογίας του Διαδικτύου αλλάζουν τον τρόπο με τον οποίο οι άνθρωποι αλληλεπιδρούν με τις εφαρμογές Διαδικτύου. Πλέον το Διαδίκτυο, με τη βοήθεια διαφορετικών τύπων κινητών συσκευών, παρέχει πρόσβαση ανά πάσα στιγμή, από οπουδήποτε και σε οποιαδήποτε υπηρεσία μέσω ενημέρωσης που προτιμούν οι χρήστες. Επιπλέον όλο και περισσότεροι χρήστες αναζητούν υπηρεσίες και εφαρμογές, προσαρμοσμένες στις ιδιαίτερες ανάγκες τους, κάτι που σχετίζεται άμεσα και με την ποικιλία νέων και ισχυρών κινητών συσκευών επικοινωνίας, ιδίως λόγω του οικονομικού ανταγωνισμού μεταξύ των εταιριών. Έτσι, έρχονται στο προσκήνιο νέες απαιτήσεις επίγνωσης της περιβάλλουσας κατάστασης και πρόσβασης πολλαπλών καναλιών σε υπηρεσίες και εφαρμογές.

Η επίγνωση της περιβάλλουσας κατάστασης θεωρείται ένα πρόσφατα αναδυόμενο ερευνητικό πεδίο της τεχνολογίας της Πληροφορικής, και ειδικότερα του τομέα του Διαδικτύου. Μπορεί να ερμηνευτεί ως φυσική εξέλιξη της προσωποποίησης (personalization), με τη σημαντική διαφορά του ότι ασχολείται πέρα από την ταυτότητα και τις προτιμήσεις του χρήστη, και με το συνολικό περιβάλλον στο οποίο λαμβάνουν χώρα οι αλληλεπιδράσεις των χρηστών. Επιπλέον, η επίγνωση της περιβάλλουσας κατάστασης στοχεύει στην ενίσχυση της χρησιμότητας των εφαρμογών, λαμβάνοντας υπόψη ένα πολύ μεγαλύτερο φάσμα ιδιοτήτων, όπως θα δείξουμε στη συνέχεια, από την προσωποποίηση.

Η ανάπτυξη εφαρμογών με επίγνωση της περιβάλλουσας πληροφορίας είναι μία ιδιαίτερα πολύπλοκη διαδικασία, καθώς –πέρα από την πολυπλοκότητα των ίδιων των υπηρεσιών που παρέχει η κάθε εφαρμογή πρέπει να αντιμετωπισθούν και τα ζητήματα που ανακύπτουν από τη χρήση της περιβάλλουσας πληροφορίας, συμπεριλαμβάνοντας τη μοντελοποίηση και την αξιοποίησή της. Προκειμένου να υποστηριχθούν οι ομάδες ανάπτυξης λογισμικού στη διαχείριση του κύκλου ζωής εφαρμογών με επίγνωση της περιβάλλουσας πληροφορίας, έχουν προταθεί σχετικές μεθοδολογίες. Στην παρούσα εργασία θα εξετάσουμε τέσσερις μεθοδολογίες, παρουσιάζοντας την εστίασή τους, τις βασικές έννοιες που εισάγουν και τα βασικά τους στάδια, ενώ θα αναδείξουμε επίσης τα ισχυρά σημεία και τις αδυναμίες τους.

Κεφάλαιο 2 Καθοδηγούμενη από μοντέλα ανάπτυξη εφαρμογών διαδικτύου με επίγνωση της περιβάλλουσας κατάστασης

Η πρόσβαση μέσω πολλαπλών καναλιών σε υπηρεσίες και εφαρμογές κερδίζει όλο και περισσότερο έδαφος μεταξύ τόσο των καταναλωτών όσο και των παρόχων ηλεκτρονικών υπηρεσιών. Για τους μεν καταναλωτές γνωρίζουμε ότι προσελκύνονται όλο και περισσότερο από φορητές συσκευές, εξοπλισμένες με έγχρωμες οθόνες υψηλής ανάλυσης, που είναι ικανές να παρέχουν δυνατότητες περιήγησης όπως ακριβώς και οι παραδοσιακοί επιτραπέζιοι υπολογιστές. Από την άλλη πλευρά επίσης, οι πάροχοι διευκολύνονται από την ανάπτυξη τυποποιημένων πρωτοκόλλων επικοινωνίας (όπως είναι το HTTP) και γλωσσών σήμανσης (όπως είναι η HTML), που υποστηρίζονται από τέτοιου είδους συσκευές και συστήματα.

Στην παρούσα προσέγγιση, η οποία παρουσιάζεται λεπτομερώς στο [16] γίνεται προσπάθεια να συνδυαστούν τα όσα αναφέρθηκαν ανωτέρω σε ένα εννοιολογικό πλαίσιο εργασίας, το οποίο θα παρέχει διευκολύνσεις για τη μοντελοποίηση εφαρμογών Διαδικτύου με επίγνωση της περιβάλλουσας κατάστασης και που είναι προσβάσιμες πολλαπλών καναλιών. Συνεπώς, θα εισαχθούν πρότυπα μοντελοποίησης ικανά να συλλάβουν τη σημασιολογία τέτοιων συμπεριφορών, και τα οποία καταδεικνύουν τον τρόπο με τον οποίο υψηλού επιπέδου δομές μπορούν να κατευθύνουν τη διαδικασία της ανάπτυξης μέσω της αυτόματης παραγωγής κώδικα.

Εν συγκρίσει με τα συμβατικά προσαρμοστικά συστήματα υπερμέσων [17-18], τα οποία αφορούν κυρίως το πρόβλημα της προσαρμοστικότητας των αποτελεσμάτων των αιτήσεων των χρηστών, στην προσέγγιση αυτή τονίζεται η σημασία της εξαρτημένης από τη περιβάλλουσα κατάσταση προσαρμογής. Η προσαρμοστικότητα αυτή ερμηνεύει τη περιβάλλουσα κατάσταση ως έναν actor που λειτουργεί ανεξάρτητα από τους χρήστες, στο ίδιο όμως υπερκείμενο που πλοηγούνται οι τελευταίοι.

Στα πλαίσια της συγκεκριμένης μεθοδολογίας, θεωρείται ότι “η περιβάλλουσα κατάσταση” αποτελεί κάθε πληροφορία που μπορεί να χρησιμοποιηθεί για τον χαρακτηρισμό της αλληλεπίδρασης ενός χρήστη με ένα λογισμικό (και αντίστροφα), καθώς και με το περιβάλλον, όπου λαμβάνει χώρα αυτή η αλληλεπίδραση. Ο ορισμός αυτός δεν επικεντρώνεται μόνο στην αλληλεπίδραση και στις ιδιότητες του περιβάλλοντος, αλλά περιλαμβάνει επίσης τον χρήστη αλλά και το λογισμικό. Επιπλέον ορίζουμε ένα σύστημα ως *σύστημα με επίγνωση της περιβάλλουσας κατάστασης* (context-aware), εάν χρησιμοποιεί την «περιβάλλουσα κατάσταση», είτε για την παροχή περιεχομένου, είτε για την εκτέλεση λειτουργιών του συστήματος, ή για να κάνει και τα δύο [19-20].

Η περιβάλλουσα κατάσταση μπορεί να χρησιμοποιείται ως εκ τούτου για την επίτευξη πιο αποτελεσματικών και αποδοτικών αλληλεπιδράσεων σε όλες εκείνες τις καταστάσεις όπου το περιεχόμενο και οι υπηρεσίες που προσφέρονται από την εφαρμογή, εξαρτώνται ειδικότερα από τις τρέχουσες καταστάσεις του περιβάλλοντος, τις ικανότητες των χρηστών, αλλά και τον πραγματικό σκοπό της εφαρμογής. Τέτοιες καταστάσεις του περιβάλλοντος είναι:

- **Προσαρμοστική προσωποποίηση** (adaptive personalization), όπου τα χαρακτηριστικά του προφίλ των χρηστών για σκοπούς προσωποποίησης

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

μπορεί να παρουσιάζουν διαφορετικές εκδόσεις εν συγκρίσει με το χρόνο. Παράδειγμα χαρακτηριστικών αποτελούν οι γρήγορες αλλαγές ιδιοτήτων, όπως για παράδειγμα η συχνότητα των παλμών, καθώς και στατικές ιδιότητες, όπως το όνομα του εκάστοτε χρήστη.

- **Λειτουργικές ανάγκες** (functional needs), όπου οι εφαρμογές μπορούν να εξαρτώνται εγγενώς από τα δεδομένα της περιβάλλουσας κατάστασης. Τέτοιες είναι εφαρμογές που κάνουν χρήση της τρέχουσας τοποθεσίας, όπως οι υπηρεσίες με χάρτες πόλης ή τα συστήματα πλοήγησης, που αντιμετωπίζουν τις πληροφορίες θέσης ως βασικό περιεχόμενο και κατά συνέπεια υποστηρίζονται κατάλληλοι μηχανισμοί εντοπισμού θέσης.
- **Χειρισμός εξαιρέσεων** (exception handling): οι προσαρμοστικοί μηχανισμοί καθώς και οι μηχανισμοί με επίγνωση περιβάλλουσας κατάστασης είναι ιδιαίτερα κατάλληλοι για τον χειρισμό καταστάσεων εξαίρεσης, αναφορικά με την αναμενόμενη συμπεριφορά της εφαρμογής. Συγκεκριμένα γεγονότα μπορεί να οδηγήσουν σε εξαιρέσεις και να διαμορφώσουν την απαίτηση για την εκτέλεση κατάλληλων αντιδράσεων. Τα υπερκείμενα που καθοδηγούνται από τη ροή εργασίας, αναπαριστούν μια κατηγορία εφαρμογών που θα μπορούσαν να επωφεληθούν από τα προσαρμοστικά χαρακτηριστικά, προκειμένου για την αντιμετώπιση των εξαιρέσεων. Επίσης, μπορούν να υποστηριχθούν μηχανισμοί δημιουργίας ειδοποιήσεων που απαιτούνται για κρίσιμες διαδικασίες παραγωγής ή για προσεγγίσεις προληπτικής συντήρησης.
- **Λειτουργίες που επιτρέπουν αλληλεπιδράσεις** (interaction-enabling functionalities), όπου η περιβάλλουσα κατάσταση μπορεί να θεωρήσει τις σωματικές αναπηρίες των χρηστών, όπως είναι τα προβλήματα στην όραση, η τύφλωση και η παράλυση, ως λειτουργίες που απαιτούν προσαρμογές στην εφαρμογή και ακολούθως να παρέχει εναλλακτικούς, κατάλληλους μηχανισμούς αλληλεπίδρασης.
- **Ενισχυμένη αποτελεσματικότητα** (enhanced effectiveness), όπου παράμετροι της περιβάλλουσας κατάστασης μπορούν να αξιοποιηθούν για την παροχή κατάλληλων χαρακτηριστικών στο πρόγραμμα, την κατάλληλη στιγμή. Για παράδειγμα, μπορούν να διαφημιστούν προσφορές που στοχεύουν σε συγκεκριμένες κατευθύνσεις, όπως είναι οι μεταβαλλόμενες συνθήκες φωτεινότητας για καλύτερη αναγνωσιμότητα βιβλίων από τους χρήστες.

Συμπερασματικά, η περιβάλλουσα κατάσταση μπορεί να περιγραφεί με βάση τις ιδιότητες και τα χαρακτηριστικά του εκάστοτε χρήστη, τις τρέχουσες δραστηριότητές του, τη θέση στην οποία χρησιμοποιείται η εφαρμογή, τις συσκευές, καθώς και κάποιες άλλες ιδιότητες του περιβάλλοντος που μπορούν να χρησιμοποιηθούν για τον προσδιορισμό της προσαρμογής της εφαρμογής που απαιτείται σε ορισμένες περιπτώσεις. Επίσης, για να είναι σε θέση η εφαρμογή να αντιδρά σε συγκεκριμένες συμπεριφορές, απαιτούνται περαιτέρω αυτοματισμοί που μπορεί να ενεργοποιηθούν από αλλαγές σε οποιοδήποτε από τις παραμέτρους που συνθέτουν την περιβάλλουσα κατάσταση.

2.1 Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει

Στον τομέα του Διαδικτύου, τα προσαρμοστικά συστήματα υπερμέσων προσπαθούν να ενσωματώσουν προηγμένους μηχανισμούς προσαρμοστικότητας και προσωποποίησης. Ένα τέτοιο παράδειγμα αποτελεί το σύστημα HyCon, το οποίο παρέχει μια γενική πλατφόρμα για την ανάπτυξη συστημάτων υπερμέσων με επίγνωση της περιβάλλουσας κατάστασης η οποία δίδει ιδιαίτερη έμφαση σε υπηρεσίες που βασίζονται στη θέση. Εφαρμογές του HyCon σχετίζονται με περιήγηση, αναζήτηση και επισήμειωση (annotation) με βάση την τρέχουσα τοποθεσία, η οποία τεκμαίρεται μέσω των συντεταγμένων του GPS. Το κύριο μειονέκτημα ωστόσο της προσέγγισης αυτής, έγκειται στο γεγονός ότι απαιτείται ένας συγκεκριμένος φυλλομετρητής Διαδικτύου, ο HyConExplorer.

Το σύστημα ΑΗΑ! βασίζεται στις τεχνολογίες του Διαδικτύου και αναπαριστά ένα εργαλείο μοντελοποίησης χρήστη και προσαρμοστικότητας, το οποίο εφαρμόστηκε αρχικά στον χώρο της ηλεκτρονικής μάθησης. Το σύστημα ΑΗΑ! διατηρεί ένα συνεχώς ενημερωμένο μοντέλο χρήστη, βάσει του οποίου εκτελείται προσαρμογή συνδέσμων υπερκειμένου και περιεχομένου. Τέλος είναι λογισμικό ανοικτού κώδικα και παρέχει μια ευέλικτη προσαρμοστική πλατφόρμα υπερμέσων για την ανάπτυξη online μαθημάτων, μουσείων, εγκυκλοπαιδειών. Επίσης, το ΑΗΑ! είναι εμπνευσμένο από το σύστημα ΑΗΑΜ, ένα μοντέλο αναφοράς που βασίζεται στο Dexter [39][40], για προσαρμοστικές εφαρμογές Διαδικτύου που αφορούν την εκπαίδευση. Έτσι, οι προσαρμοστικές εφαρμογές υπερμέσων βασίζονται σε μοντέλα τομέα (domain), χρήστη (user) και διδασκαλίας (teaching). Η προσαρμογή επιτυγχάνεται κατά το χρόνο εκτέλεσης με την προσαρμογή του περιεχομένου από το μοντέλο τομέα σύμφωνα με τους κανόνες του μοντέλου διδασκαλίας και λαμβάνοντας υπόψη τις ιδιότητες του μοντέλου του χρήστη. Δυστυχώς, το παράδειγμα διδασκαλίας-μάθησης περιορίζει τη δυνατότητα εφαρμογής του ΑΗΑΜ σε συγκεκριμένους τομείς, κι επίσης καθιστά τη μέθοδο λιγότερο κατάλληλη για τη μοντελοποίηση γενικών προσαρμοστικών συστημάτων υπερμέσων. Επιπλέον, οι μηχανισμοί προσαρμοστικότητας του ΑΗΑΜ λαμβάνουν χώρα στο επίπεδο των στιγμιότυπων κι ως εκ τούτου δεν είναι κατάλληλοι για την αντιμετώπιση της τεράστιας ποσότητας των δυναμικών περιεχομένων που απαιτούν οι σημερινές εφαρμογές του Διαδικτύου.

Η προσέγγιση μοντελοποίησης που παρουσιάζεται εδώ προσπαθεί να ξεπεράσει τις αδυναμίες προηγούμενων εργασιών και προσεγγίσεων. Αρχικά, η μέθοδος αυτή παρέχει έννοιες/θέματα, συμβολισμούς και τεχνολογίες υλοποίησης για την ανάπτυξη εφαρμογών του Διαδικτύου, χωρίς να παρουσιάζει κάποιο περιορισμό και χωρίς να απαιτεί λύσεις από την πλευρά είτε του πελάτη είτε του εξυπηρέτη. Αξιοποιεί τις δυνατότητες του βασιζόμενου σε μοντέλα υποδείγματος ανάπτυξης, το οποίο παρέχει υψηλού επιπέδου δομές για τον καθορισμό της δομής και της συμπεριφοράς της εφαρμογής. Λόγω του τυπικού ορισμού της σημασιολογίας, οι παραγόμενες προδιαγραφές μπορούν να μετασχηματιστούν σε εκτελέσιμο κώδικα, με τη βοήθεια διαφόρων ενοποιημένων τεχνικών, διευκολύνοντας έτσι τη συνολική διαδικασία ανάπτυξης. Επιπρόσθετα, η μέθοδος αυτή αντλεί στοιχεία από το βασιζόμενο στα δεδομένα υπόδειγμα ανάπτυξης (data-driven paradigm), γεγονός που ενισχύει την αναπαράσταση της προσωποποίησης και των μετα-δεδομένων της περιβάλλουσας κατάστασης στα δεδομένα της εφαρμογής. Αποτέλεσμα του

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

παραπάνω είναι η ανάπτυξη εφαρμογών του Διαδικτύου όπου η προσαρμογή του περιεχομένου και η συμπεριφορά στις αντιδράσεις της περιβάλλουσας κατάστασης επιτυγχάνονται με δυναμικό τρόπο. Τέλος, η βασική καινοτομία σε σχέση με τις άλλες προσεγγίσεις είναι η προώθηση της περιβάλλουσας κατάστασης ως actor, κάτι που οδηγεί σε εφαρμογές με δυνατότητα ανάδρασης (reactive), καθώς οι εφαρμογές αυτές μπορούν να ανταποκριθούν να αλλαγές ανίχνευσης περιβάλλουσας κατάστασης.

2.2 Βασικές έννοιες της προσέγγισης

Οι δύο βασικές έννοιες τις οποίες θα αναλύσουμε στη συνέχεια είναι η σχεδίαση δεδομένων και υπερκειμένων. Η μεν σχεδίαση δεδομένων επικεντρώνεται στην αναπαράσταση των δεδομένων της εφαρμογής (και πιο συγκεκριμένα στον καθορισμό και τη διαχείρισή τους), ενώ αντίθετα η σχεδίαση υπερκειμένων επικεντρώνεται στον καθορισμό του περιεχομένου της εφαρμογής καθώς και στις λειτουργίες του εξυπηρέτη που υλοποιούν τη λογική της εφαρμογής (ειδικότερα στην προσθήκη στα υπερκείμενα των συμπεριφορών που αντιδρούν στις αλλαγές της κατάστασης της περιβάλλουσας κατάστασης). Επίσης, η συγκεκριμένη προσέγγιση βασίζεται σε δεδομένα, αφού η συνολική διαδικασία σχεδίασης ξεκινά από τη σχεδίαση δεδομένων, και οι περισσότερες αποφάσεις που λαμβάνονται κατά τη διάρκεια των διαφόρων φάσεων εξαρτώνται από την οργάνωση των δεδομένων.

2.2.1 Λειτουργικές απαιτήσεις συστήματος

Για τη διαχείριση των δεδομένων σε εφαρμογές με επίγνωση της περιβάλλουσας κατάστασης απαιτούνται τα εξής:

- Ορισμός και αναπαράσταση μοντέλου περιβάλλουσας κατάστασης στα δεδομένα της εφαρμογής: θα πρέπει δηλαδή να αναγνωριστούν και να αναπαρασταθούν σε επίπεδο δεδομένων δεδομένα οι βασικές ιδιότητες περιβάλλουσας κατάστασης που απαιτούνται για την υποστήριξη της προσαρμοστικότητας.
- Διαχείριση μοντέλου περιβάλλουσας κατάστασης, που αποτελείται από:
 - Απόκτηση δεδομένων περιβάλλουσας κατάστασης, με μετρικές που αφορούν φυσικές ιδιότητες του πραγματικού κόσμου ιδιότητες, οι οποίες χαρακτηρίζουν το περιβάλλον χρήσης.
 - Ενημέρωση μοντέλου περιβάλλουσας κατάστασης, για τη διατήρηση της συνέπειας των δεδομένων περιβάλλουσας κατάστασης σε σχέση με τις πραγματικές συνθήκες του περιβάλλοντος.
 - Παρακολούθηση δεδομένων περιβάλλουσας κατάστασης, για ανίχνευση των διακυμάνσεων της εκάστοτε περιβάλλουσας κατάστασης που προκαλούν την εκτέλεση ενεργειών προσαρμοστικότητας. Ως εκ τούτου, οποιαδήποτε διακύμανση μπορεί να προκαλέσει αυτόματα μια συμπεριφορά προσαρμοστικότητας της εφαρμογής του Διαδικτύου.

Επιπλέον, όσον αφορά τα υπερκείμενα των εφαρμογών, οι εφαρμογές με επίγνωση της περιβάλλουσας κατάστασης πρέπει να διασφαλίζουν:

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

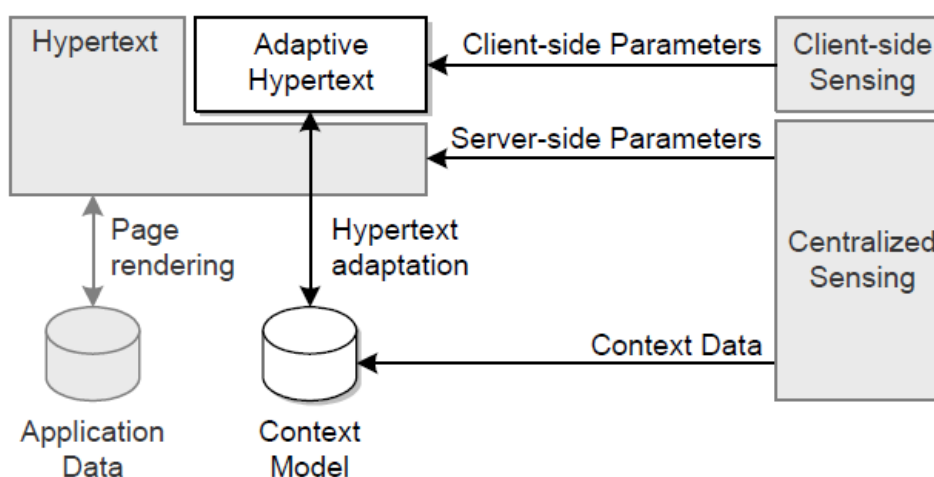
Μεταπτυχιακή εργασία

Φούντας Γεώργιος

- Την προσαρμοστικότητα των περιεχομένων και των υπηρεσιών που παρέχονται από τις σελίδες, με βάση την εκάστοτε περιβάλλουσα κατάσταση.
- Την προσαρμοστικότητα της πλοήγησης μέσω αυτόματης πλοήγησης προς άλλες σελίδες, οι οποίες είναι πιο κατάλληλες για την τρέχουσα περιβάλλουσα κατάσταση.
- Την προσαρμοστικότητα της συνολικής δομής του υπερκειμένου για να υλοποιηθεί προσαρμοστικότητα ευρείας κλίμακας π.χ. λόγω αλλαγής της συσκευής, του ρόλου και της δραστηριότητας του χρήστη, όταν αυτός ενεργεί κινούμενος εντός ενός πολυκαναλικού περιβάλλοντος.
- Την προσαρμοστικότητα των ιδιοτήτων της παρουσίασης προκειμένου να επιτραπούν λεπτομερειακού επιπέδου προσαρμογές στην εμφάνιση της εφαρμογής.

2.2.2 Αρχιτεκτονικές απαιτήσεις

Το διάγραμμα ροής δεδομένων της περιβάλλουσας κατάστασης, που φαίνεται στην εικόνα 1, είναι σχεδιασμένο για την υποστήριξη εφαρμογών υπερκειμένου με επίγνωση της περιβάλλουσας κατάστασης. Στα συνολικά δεδομένα περιλαμβάνονται τόσο τα δεδομένα της εφαρμογής (δηλαδή τα αντικείμενα που χαρακτηρίζουν όλο τον τομέα της εφαρμογής), καθώς και ορισμένα δεδομένα της περιβάλλουσας κατάστασης που προσφέρουν ανά πάσα στιγμή μια ενημερωμένη αναπαράσταση της περιβάλλουσας κατάστασης, την οποία ονομάζουμε μοντέλο της περιβάλλουσας κατάστασης. Κατά τον χρόνο εκτέλεσης, ο δυναμικός υπολογισμός του υπερκειμένου εκμεταλλεύεται κυρίως τα δεδομένα της εφαρμογής ώστε να γεμίσει τις σελίδες. Στη συνέχεια γίνεται η υπόθεση ότι σε ένα υποσύνολο των στοιχείων του υπερκειμένου, που περιλαμβάνεται στο υπερκείμενο προσαρμοστικότητας (adaptive hypertext), προστίθενται οι συμπεριφορές προσαρμοστικότητας, έτσι ώστε ο υπολογισμός τους να εκμεταλλεύεται τα δεδομένα της περιβάλλουσας κατάστασης.



Εικόνα 1 - Διάγραμμα ροής δεδομένων της περιβάλλουσας κατάστασης

Όπως φαίνεται και στην παραπάνω εικόνα, η προτεινόμενη αρχιτεκτονική λαμβάνει υπόψη μηχανισμούς αισθητήρων περιβάλλουσας κατάστασης για τη συλλογή δεδομένων της περιβάλλουσας κατάστασης από το περιβάλλον υλοποίησης

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

της εφαρμογής. Αυτό μπορεί να συνεπάγεται τη χρήση τυποποιημένου λογισμικού αισθητήρων για απόκτηση της περιβάλλουσας πληροφορίας, ενώ μπορεί επίσης να απαιτηθεί η ανάπτυξη μηχανισμών αισθητήρων που εκτελούνται στην πλευρά του εξυπηρετούμενου, όπως συμβαίνει για συσκευές που είναι εξοπλισμένες με GPS, ή η υιοθέτηση κεντρικών υποδομών συλλογής δεδομένων περιβάλλουσας κατάστασης στο περιβάλλον χρήσης και κατόπιν της επικοινωνίας τους με την εφαρμογή.

Ωστόσο, σε εννοιολογικό επίπεδο, οι μηχανισμοί αίσθησης περιβάλλουσας κατάστασης δεν επηρεάζουν ιδιαίτερα τον καθορισμό του υπερκειμένου της εφαρμογής: το μόνο στοιχείο που επηρεάζεται είναι ο τρόπος επικοινωνίας με τους αισθητήρες αλλά και η διαχείριση των σχετικών δεδομένων από την εφαρμογή. Συνεπώς, στη μέθοδο μοντελοποίησης γίνεται η υπόθεση ότι κάποιες λύσεις επιτρέπουν ανίχνευση τέτοιων δεδομένων, τα οποία γίνονται διαθέσιμα στην εφαρμογή με τους ακόλουθους τρόπους:

- Ορισμένες παράμετροι της περιβάλλουσας κατάστασης που δημιουργήθηκαν/ανακτήθηκαν στην πλευρά της συσκευής του χρήστη στέλνονται πίσω στην εφαρμογή. Τέτοιο παράδειγμα αποτελούν οι τιμές που προσαρτώνται στη διεύθυνση URL των επιθυμητών σελίδων ή μέσω μηνυμάτων SOAP.
- Οι τιμές κάποιων παραμέτρων επιπέδου συνόδου HTTP που διατηρούνται από την πλευρά του εξυπηρετή τίθενται σύμφωνα με τα πρόσφατα δεδομένα της περιβάλλουσας κατάστασης.
- Οι μηχανισμοί αίσθησης πληροφορίας ενημερώνουν το μοντέλο της περιβάλλουσας κατάστασης στο επίπεδο των δεδομένων, για παράδειγμα μέσω ασύγχρονων υπηρεσιών.

Οι δύο πρώτοι τρόποι απόκτησης δρουν στο επίπεδο του υπερκειμένου κι επομένως δεν ενεργούν άμεσα στο επίπεδο δεδομένων. Ωστόσο, ορισμένες λειτουργίες στο υπερκείμενο πρέπει να παρέχουν υποστήριξη για την αποθήκευση των νέων τιμών, διατηρώντας έτσι ενημερωμένο το μοντέλο της περιβάλλουσας κατάστασης στο επίπεδο δεδομένων, όσον αφορά την περιβάλλουσα κατάσταση. Ο τελευταίος μηχανισμός απόκτησης, ο οποίος λειτουργεί άμεσα στα δεδομένα, είναι κατάλληλος για τα συστήματα προσαρμοστικότητας που βασίζονται σε κεντρικούς μηχανισμούς αίσθησης περιβάλλουσας κατάστασης, όπως είναι το RFID.

2.3 Μεθοδολογία ανάπτυξης εφαρμογών σύμφωνα με την προσέγγιση

Η μεθοδολογία ανάπτυξης εφαρμογών θα παρουσιαστεί μέσω της εφαρμογής της με βάση το μοντέλο δόμησης εφαρμογών υπερκειμένου *WebML*, το οποίο περιέχει τις αφαιρετικές έννοιες που παρουσιάστηκαν στα προηγούμενα εδάφια. Στη συνέχεια αναλύεται το κάθε ένα επιμέρους στοιχείο της συγκεκριμένης εφαρμογής κάνοντας χρήση και του αντίστοιχου σχήματος.

2.3.1 Παρουσίαση WebML

Η *WebML* αποτελεί μια οπτική γλώσσα για τον καθορισμό της δομής των περιεχομένων των εφαρμογών του Διαδικτύου αλλά και της οργάνωσης και παρουσίασης των περιεχομένων σε ένα ή περισσότερα υπερκείμενα.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

Η διαδικασία σχεδιασμού ξεκινά με τον καθορισμό ενός σχήματος δεδομένων, εκφράζοντας την οργάνωση των περιεχομένων της εφαρμογής, χρησιμοποιώντας κλασικά και καθιερωμένα μοντέλα δεδομένων, όπως είναι το μοντέλο Οντοτήτων-Συσχετίσεων ή το διάγραμμα κλάσεων της UML.

Εν συνέχεια, το **μοντέλο υπερκειμένου της WebML** (WebML Hypertext Model) περιγράφει τον τρόπο με τον οποίο τα περιεχόμενα, τα οποία καθορίστηκαν προηγουμένως στο σχήμα των δεδομένων, δημοσιεύονται μέσω του υπερκειμένου της εφαρμογής. Η συνολική δομή του υπερκειμένου ορίζεται με βάση τις **όψεις ιστοχώρου**, τις **περιοχές**, τις **σελίδες** και τις **μονάδες περιεχομένου**. Μια *όψη ιστοχώρου* είναι ένα υπερκείμενο, σχεδιασμένο για την ικανοποίηση ενός συγκεκριμένου συνόλου απαιτήσεων. Πολλές όψεις ιστοχώρου μπορούν να οριστούν στο ίδιο σχήμα δεδομένων, για την εξυπηρέτηση αναγκών διαφορετικών κοινοτήτων χρηστών, είτε για να μπορέσουν οι σελίδες να ανταποκριθούν στις απαιτήσεις διαφόρων συσκευών όπως τα PDAs, τα smart phones και παρόμοιες συσκευές.

Μια όψη ιστοχώρου αποτελείται από περιοχές, που είναι οι κύριοι τομείς του υπερκειμένου, και αυτές με τη σειρά τους περιλαμβάνουν αναδρομικά άλλες υποπεριοχές ή σελίδες. Οι σελίδες είναι αυτές που περιέχουν τις πληροφορίες που παρέχονται στον χρήστη. Είναι κατασκευασμένες από *μονάδες περιεχομένου*, οι οποίες αποτελούν τα στοιχειώδη κομμάτια πληροφορίας που εξάγονται από τις πηγές των δεδομένων μέσω των ερωτημάτων του χρήστη και δημοσιεύονται στις σελίδες. Ειδικότερα, οι μονάδες περιεχομένου δηλώνουν εναλλακτικούς τρόπους για εμφάνιση ενός ή περισσοτέρων στιγμιότυπων. Ο καθορισμός των μονάδων απαιτεί τον ορισμό μιας πηγής (source) και μιας οντότητας-επιλογέα (selector). Από την πηγή εξάγεται το περιεχόμενο της μονάδας, ενώ ο επιλογέας είναι μια κατάσταση που χρησιμοποιείται για την ανάκτηση των πραγματικών αντικειμένων της πηγής, δηλαδή του περιεχομένου της μονάδας.

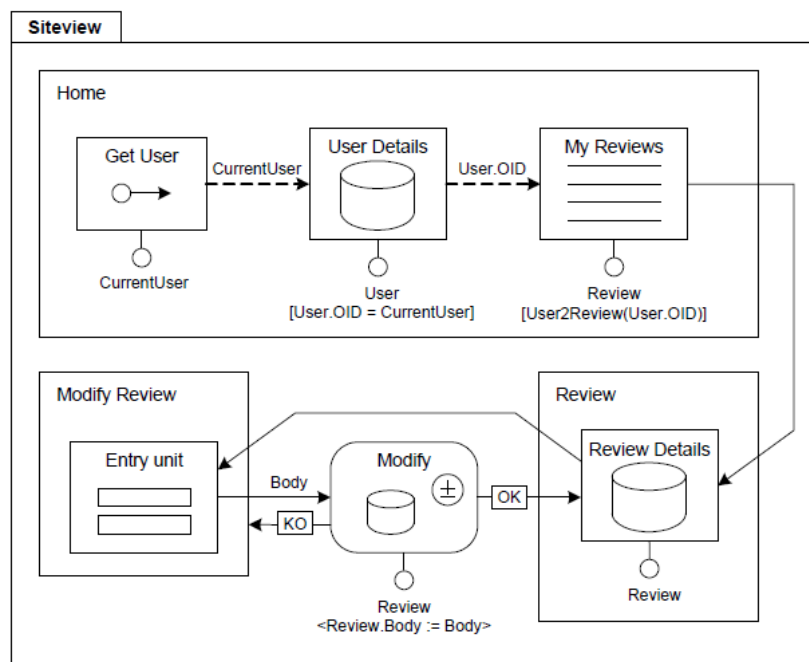
Οι μονάδες περιεχομένου και οι σελίδες διασυνδέονται με *συνδέσμους* για τη συγκρότηση των όψεων. Οι σύνδεσμοι μπορούν να συνδέσουν τις μονάδες με μια ποικιλία διαμορφώσεων, οδηγώντας με αυτόν τον τρόπο σε σύνθετους μηχανισμούς πλοήγησης. Εκτός από την αναπαράσταση της πλοήγησης του χρήστη, οι σύνδεσμοι μεταξύ των μονάδων προσδιορίζουν επίσης τη μεταφορά ορισμένων παραμέτρων, που χρησιμοποιεί η μονάδα προορισμού στη συνθήκη του επιλογέα για την εξαγωγή των στιγμιότυπων των δεδομένων που θα εμφανιστούν.

Ορισμένες μονάδες WebML υποστηρίζουν επιπλέον τον καθορισμό λειτουργιών διαχείρισης περιεχομένου. Δηλαδή επιτρέπουν τη δημιουργία, διαγραφή ή τροποποίηση ενός στιγμιότυπου μιας οντότητας (αντίστοιχα μέσω των μονάδων create, delete και modify), ή την προσθήκη είτε κατάργηση μιας σχέσης μεταξύ δύο στιγμιότυπων (αντίστοιχα μέσω των μονάδων connect και disconnect). Η WebML παρέχει επίσης μονάδες για τον ορισμό των *παραμέτρων συνεδρίας* (session parameters). Οι παράμετροι μπορούν να ρυθμιστούν μέσω της μονάδας *Set* και καταναλώνονται μέσα σε μια σελίδα μέσω της μονάδας *Get*.

Στην εικόνα 2 παρουσιάζεται ένα παράδειγμα ενός μοντέλου υπερκειμένων WebML. Πιο συγκεκριμένα, φαίνεται μια όψη ιστοχώρου με τρεις σελίδες, που επιτρέπουν στους χρήστες να δουν τις κριτικές τους σχετικά με τα έργα τέχνης ενός μουσείου, που είχαν υποβληθεί στο παρελθόν μέσα από μια εφαρμογή Διαδικτύου, κι ακολούθως να τις τροποποιήσουν. Στη σελίδα Home, μια μονάδα Get διαβάζει το αναγνωριστικό του εκάστοτε χρήστη από την παράμετρο *CurrentUser* και το παρέχει

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

ως είσοδο στην ακόλουθη μονάδα *Data*, η οποία δημοσιεύει ακολούθως τα δεδομένα του προσωπικού προφίλ του χρήστη. Τα στοιχεία που θα δημοσιευτούν, επιλέγονται από μία συνθήκη επιλογέα, η οποία εμφανίζεται κάτω από μονάδα. Το αναγνωριστικό του χρήστη μεταδίδεται περαιτέρω στην μονάδα καταλόγου (index unit) *My Reviews*, μέσω ενός *συνδέσμου μεταφοράς* (transport link). Στη σελίδα *Review*, μια μονάδα δεδομένων παρουσιάζει μια κριτική που επιλέχτηκε προηγουμένως μέσω του *My Reviews*. Επίσης, ένας σύνδεσμος οδηγεί το χρήστη στη σελίδα *Modify Review*, η οποία περιλαμβάνει μια *μονάδα καταχώρησης* (Entry) για την εισαγωγή νέου κειμένου για τροποποίηση της κριτικής. Ο σύνδεσμος υποβολής της φόρμας ενεργοποιεί μια λειτουργία *Modify*, η οποία ενημερώνει τις κριτικές στα δεδομένα της εφαρμογής.



Εικόνα 2 - Σχήμα υπερκειμένου WebML

2.3.2 Μοντελοποίηση περιβάλλουσας κατάστασης ως δεδομένα

Στην παράγραφο αυτή εμπλουτίζουμε τα δεδομένων της εφαρμογής μ' ένα μοντέλο περιβάλλουσας πληροφορίας, ώστε να διαμορφώσουμε μια συνεπή και συνεχώς ενημερωνόμενη αναπαράσταση των μετα-δεδομένων που απαιτούνται για την υποστήριξη της προσαρμοστικότητας.

Τα δεδομένα της περιβάλλουσας κατάστασης μπορούν να αντλούνται από διάφορες πηγές που περιλαμβάνουν αισθητήρες, πληροφορίες που παρέχονται από τον χρήστη και συναγόμενα δεδομένα. Η διαφορά τους είναι ότι τα μεν παρεχόμενα στοιχεία του χρήστη είναι γενικώς αξιόπιστα και τείνουν να είναι στατικά, τα δεδομένα που προέρχονται από αισθητήρες είναι ιδιαίτερος δυναμικά και μπορεί να γίνουν αναξιόπιστα λόγω πιθανού θορύβου και σφαλμάτων στους αισθητήρες.

Επιπλέον αναγνωρίζεται η διάκριση μεταξύ φυσικών και λογικών δεδομένων της περιβάλλουσας κατάστασης, όπου τα λογικά δεδομένα παράγονται κατόπιν μετασχηματισμού των φυσικών, παρέχοντας αφαιρετικές έννοιες με πλουσιότερη σημασιολογία όσον αφορά την εφαρμογή. Τέτοια μετατροπή μπορεί να επιτευχθεί μέσω εφαρμογής κατάλληλων φίλτρων στα ακατέργαστα (raw) δεδομένα της

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

περιβάλλουσας κατάστασης. Στην προσέγγιση που μελετούμε στην παρούσα ενότητα, θεωρείται ότι τα δεδομένα που τροφοδοτούν το μοντέλο της περιβάλλουσας κατάστασης, κανονικοποιούνται μ' ένα τέτοιο φίλτράρισμα.

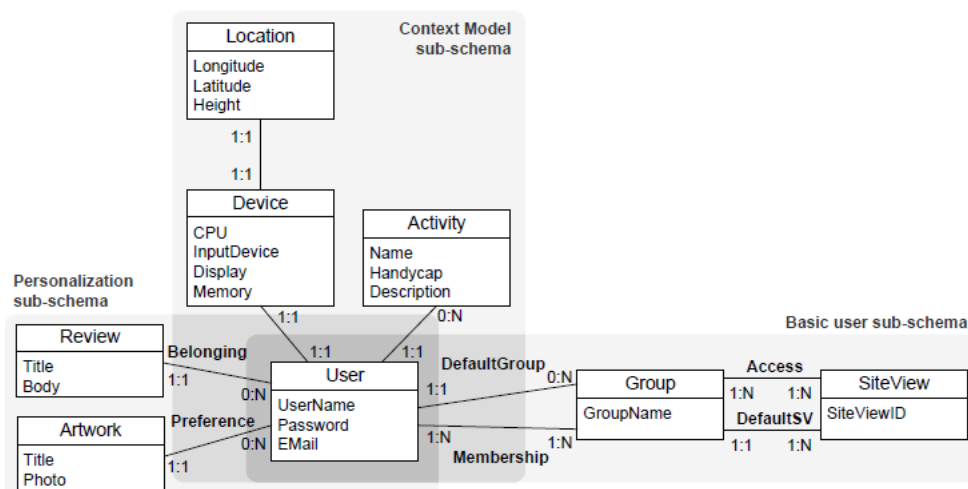
Επίσης, γίνεται η υπόθεση ότι το πραγματικό μοντέλο της περιβάλλουσας κατάστασης μιας πραγματικής εφαρμογής μπορεί να ποικίλει ανάλογα με τους στόχους της προσαρμοστικότητας που πρέπει να επιτευχθούν. Ακόμα κι αν υπάρχουν πολλές ιδιότητες, που συνήθως θεωρούνται ως *χαρακτηριστικά της περιβάλλουσας κατάστασης* (context attributes) (όπως είναι για παράδειγμα, η τοποθεσία, ο χρόνος ή τα χαρακτηριστικά της συσκευής), δεν υπάρχει κανένα καθολικό μοντέλο της περιβάλλουσας κατάστασης που εφαρμόζεται σε όλα τα είδη των εφαρμογών.

Στην εικόνα 3 παρουσιάζεται ένα διάγραμμα Οντοτήτων-Συσχετίσεων που αντιστοιχεί στο μοντέλο δεδομένων μιας εφαρμογής, συμπεριλαμβάνοντας ένα πιθανό μοντέλο περιβάλλουσας κατάστασης. Το μοντέλο περιέχει 3 υπο-σχήματα, που το καθένα επιτελεί συγκεκριμένους σκοπούς:

- **Υπο-σχήμα προφίλ χρήστη** (user profile sub-schema), όπου οι χρήστες, οι ομάδες και οι όψεις του ιστοχώρου αναπαριστούνται ως βασικές οντότητες στα δεδομένα της εφαρμογής. Η οντότητα *User* παρέχει ένα βασικό προφίλ των χρηστών της εφαρμογής, η οντότητα *Group* επιτρέπει τη διαχείριση των δικαιωμάτων πρόσβασης και η οντότητα *Site View* επιτρέπει την αναπαράσταση των όψεων και την αντιστοίχιση σε αυτές δεδομένων της εφαρμογής κατά τέτοιο τρόπο ώστε οι όψεις να είναι προσαρμοσμένες στις ανάγκες της εκάστοτε ομάδας χρηστών. Η συσχέτιση *Membership*, η οποία έχει πληθικότητα πολλά-προς-πολλά, εκφράζει το ότι οι χρήστες μπορούν να ανήκουν σε πολλές διαφορετικές ομάδες, οι οποίες με τη σειρά τους περιλαμβάνουν πολλούς χρήστες. Η συσχέτιση *DefaultGroup* συνδέει ένα χρήστη με τον προεπιλεγμένο ρόλο του. Όταν ο χρήστης συνδέεται στην εφαρμογή, τότε μπορεί μέσω της συσχέτισης *DefaultSV* να μεταφερθεί στην προεπιλεγμένη όψη ιστοχώρου της προεπιλεγμένης ομάδας του. Η συσχέτιση *Access*, η οποία έχει πληθικότητα πολλά-προς-πολλά, ορίζει τις όψεις ιστοχώρου στις οποίες μπορεί να έχει πρόσβαση μια συγκεκριμένη ομάδα. Η συσχέτιση αυτή απαιτείται μόνο στην περίπτωση εφαρμογών προσαρμοστικότητας που ενδέχεται να απαιτούν διαφορετικές δομές αλληλεπίδρασης και πλοήγησης για την ίδια ομάδα.
- **Υπο-σχήμα προσωποποίησης** (personalization sub-schema), όπου στην εικόνα 3 φαίνεται ένα πιθανό σχήμα προσωποποίησης που αποτελείται από κάποιες συσχετίσεις που συνδέουν διάφορες οντότητες της εφαρμογής με την οντότητα *User*. Για παράδειγμα, η οντότητα *Comment* συνδέεται με την οντότητα *User* μέσω της συσχέτισης *Belonging*, η οποία με τη σειρά της αναφέρει τους χρήστες στους οποίους ανήκουν σχόλια που δημοσιεύτηκαν στην εφαρμογή. Ομοίως, η σχέση *Preference* συνδέει ένα χρήστη με τα έργα που του αρέσουν περισσότερο. Σε γενικές γραμμές, η οι συσχετίσεις προσωποποίησης μεταξύ της οντότητας *User* και κάποιων άλλων οντοτήτων, σημαίνει ότι ο χρήστης είναι ο δημιουργός ενός συγκεκριμένου αντικειμένου ή ότι έχει προτιμήσεις στα αντικείμενα μιας συγκεκριμένης οντότητας. Τα σχήματα της προσωποποίησης μπορούν να ποικίλλουν στο τομέα της πολυπλοκότητας,

ανάλογα με την ποσότητα του περιεχομένου που πρέπει να προσαρμόζεται σε μεμονωμένους χρήστες.

- Υπο-σχήμα μοντέλου περιβάλλουσας κατάστασης** (context model sub-schema), όπου στην εικόνα 3 προτείνεται μια πιθανή ρύθμιση των μεταδεδομένων της περιβάλλουσας κατάστασης, όπως θα μπορούσαν να εφαρμοστούν, για παράδειγμα, σε κινητές και πολυκαναλικές εφαρμογές. Οι οντότητες *Device*, *Location* και *Activity* περιγράφουν τις συγκεκριμένες ιδιότητες της περιβάλλουσας κατάστασης που παρέχεται από την εφαρμογή. Δηλαδή στο σχήμα της προσωποποίησης, οι οντότητες περιβάλλουσας κατάστασης συνδέονται με την οντότητα *User* για την ανάθεση του κάθε χρήστη στο προσωπική του περιβάλλουσα κατάσταση.



Εικόνα 3 - Σχεδίαση υπο-σημάτων χρήση, προσωποποίησης και περιβάλλουσας κατάστασης

2.3.3 Δόμηση υπερκειμένου το οποίο διαθέτει επίγνωση της περιβάλλουσας κατάστασης

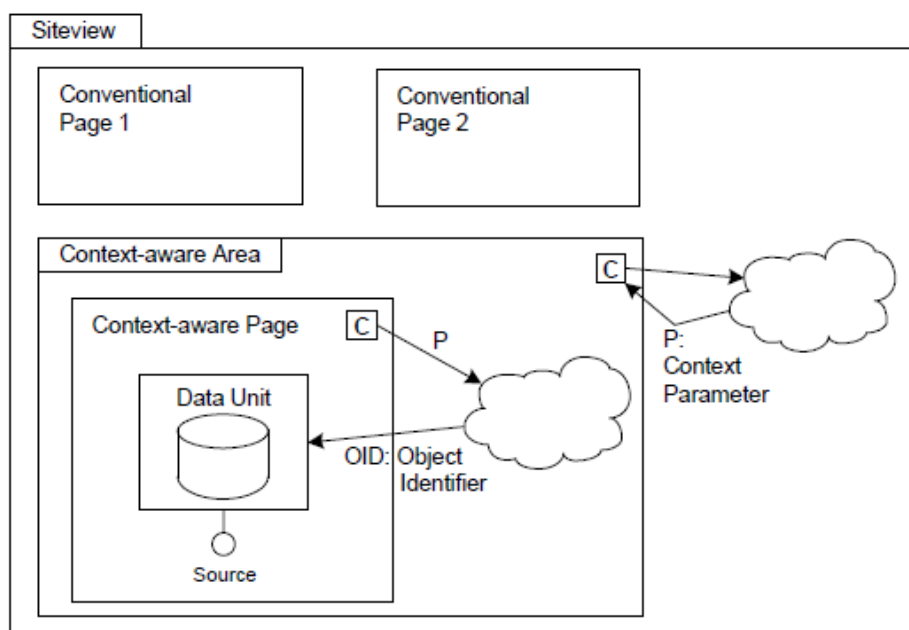
Εκτός από τη συμπλήρωση στα δεδομένα της εφαρμογής, των δεδομένων της περιβάλλουσας κατάστασης, η προσέγγιση μοντελοποίησης εισάγει επίσης ορισμένες νέες κατασκευές στο επίπεδο υπερκειμένου, που απαιτούνται για τον καθορισμό συμπεριφορών προσαρμοστικότητας στην εφαρμογή.

2.3.3.1 Σελίδες με επίγνωση της περιβάλλουσας κατάστασης

Η εικόνα 4 απεικονίζει το όραμα της προσέγγισης αυτής, για τα υπερκείμενα με επίγνωση της περιβάλλουσας κατάστασης. Η βασική υπόθεση είναι ότι η επίγνωση της περιβάλλουσας κατάστασης είναι μια ιδιότητα που σχετίζεται μόνο με ορισμένες σελίδες μιας εφαρμογής κι όχι απαραίτητα με την ίδια την εφαρμογή στο σύνολό της. Οι εφαρμογές με επίγνωση της τοποθεσίας, για παράδειγμα, προσαρμόζουν τα βασικά περιεχόμενα της ίδιας της εφαρμογής στη θέση/τοποθεσία του χρήστη, αλλά οι τυπικές “σελίδες πρόσβασης” (συμπεριλαμβανομένων των συνδέσεων στις κύριες περιοχές της εφαρμογής) μπορεί να μην επηρεάζονται από τη περιβάλλουσα κατάσταση χρήσης της θέσης.

Όπως φαίνεται στην εικόνα 4, οι προσαρμοστικές σελίδες επισημαίνονται με μία ετικέτα «C», έτσι ώστε να διακρίνονται εύκολα από τις συμβατικές σελίδες. Αυτή η ετικέτα φανερώνει το ότι κάποιες ενέργειες προσαρμοστικότητας συνδέονται με τη σελίδα. Έτσι, κατά την εκτέλεση της εφαρμογής, αυτές οι ενέργειες πρέπει να αποτιμώνται πριν από τον υπολογισμό της σελίδας, διότι μπορεί να χρησιμεύουν στην προσαρμογή του περιεχομένου της σελίδας ή στην τροποποίηση της προκαθορισμένης ροής πλοήγησης.

Προκειμένου να μπορεί μία σελίδα χαρακτηρισμένη με ετικέτα «C» να υλοποιήσει συμπεριφορά που να περιλαμβάνει επίγνωση της περιβάλλουσας κατάστασης, πρέπει να είναι εφοδιασμένη με τη δυνατότητα παρακολούθησης της περιβάλλουσας κατάστασης και με βάση αυτή, να προκαλεί αντίστοιχες ενέργειες προσαρμοστικότητας. Αυτό μπορεί εν μέρει να επιτευχθεί με την αποτίμηση των ενεργειών προσαρμοστικότητας κάθε φορά που η σελίδα προσπελάζεται και πιο συγκεκριμένα πριν τον υπολογισμό της σελίδας αυτής. Μια διαφορετική πολιτική μπορεί να απαιτήσει την αποτίμηση και την εκτέλεση των ενεργειών προσαρμοστικότητας αφότου προσπελάσει ο χρήστης τη σελίδα. Αφού προσπελαστεί, η σελίδα ακολούθως παρουσιάζεται σύμφωνα με τις επιλογές του χρήστη, και το περιεχόμενό της προσαρμόζεται με βάση την τρέχουσα περιβάλλουσα κατάσταση, μέσω μίας ενέργειας ανανέωσης περιεχομένου.



Εικόνα 4 - Σχήμα υπερκειμένου WebML με δύο συμβατικές σελίδες, μία σελίδα με επίγνωση της περιβάλλουσας κατάστασης, μία περιοχή με επίγνωση της περιβάλλουσας κατάστασης, καθώς και τα σχετιζόμενα νέφη περιβάλλουσας κατάστασης

2.3.3.2 Πολιτικές προσαρμοστικότητας

Όπως αναφέραμε προηγουμένως, υπάρχουν διάφορες ενέργειες προσαρμοστικότητας όσον αφορά τις σελίδες που έχουν χαρακτηριστεί με ετικέτες «C». Αυτές οι ενέργειες μπορούν να αποτιμηθούν σύμφωνα με δύο διαφορετικές πολιτικές:

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

- **Αναβαλλόμενη προσαρμοστικότητα**, όπου ο χρήστης αποκτά την υψηλότερη προτεραιότητα. Αφού ο χρήστης έχει προσπελάσει τη σελίδα και αυτή έχει εμφανισθεί σύμφωνα με τις επιλογές του χρήστη, δημιουργούνται με αυτόματο τρόπο περιοδικές αιτήσεις σελίδας που προκαλούν την αποτίμηση των ενεργειών προσαρμοστικότητας.
- **Άμεση προσαρμοστικότητα**, όπου η περιβάλλουσα κατάσταση αποκτά την υψηλότερη προτεραιότητα. Οι ενέργειες προσαρμοστικότητας αποτιμώνται κάθε φορά που η σελίδα προσπελαύνεται, πριν από τον υπολογισμό της σελίδας αυτής. Αυτό σημαίνει ότι οι ενέργειες προσαρμοστικότητας αξιολογούνται τόσο όταν προσπελαύνεται για πρώτη φορά η σελίδα από χρήστες όσο και σε κάθε αυτόματο έλεγχο αλλαγής.

Στην προσέγγισή που μελετούμε στο παρόν εδάφιο, η αναβαλλόμενη προσαρμοστικότητα θεωρείται ως η προεπιλεγμένη πολιτική. Η προσαρμοστικότητα ξεκινά μόνο με τις αυτόματες ανανεώσεις που πραγματοποιούνται αφότου ο χρήστης έχει προσπελάσει τη σελίδα. Όταν ένας χρήστης ζητά μια συγκεκριμένη σελίδα, η πρώτη παραγόμενη απάντηση έχει πάντα τα αναμενόμενα αποτελέσματα με βάση την επιλογή του χρήστη. Μόνο μετά από αυτή την πρώτη σελίδα, θα μπορούσε κάποια σελίδα να γίνει αντικείμενο προσαρμοστικότητας, σύμφωνα με τη ισχύουσα περιβάλλουσα κατάσταση. Η επιλογή αυτή αποσκοπεί στην ελαχιστοποίηση των συμπεριφορών της εφαρμογής που θα μπορούσαν να εκληφθούν ως ενοχλητικές από τους χρήστες, κάτι το οποίο αποτελεί το πιο φυσικό τρόπο προσαρμοστικότητας της μοντελοποίησης.

Ωστόσο, η άμεση προσαρμοστικότητα θα μπορούσε να είναι χρήσιμη για την αντιμετώπιση εξαιρετικών καταστάσεων, όπως είναι στις περιπτώσεις που η έγκαιρη αντίδραση στις μεταβολές της περιβάλλουσας κατάστασης θα μπορούσε να είναι πιο σημαντική από τις επιλογές του χρήστη.

2.3.3.3 Νέφη περιβάλλουσας κατάστασης

Ως **νέφος περιβάλλουσας κατάστασης** ορίζουμε το σύνολο των ενεργειών προσαρμοστικότητας που σχετίζονται με μια σελίδα. Όπως φαίνεται στην εικόνα 4, το **νέφος** είναι εξωτερικό ως προς τη σελίδα και η αλυσίδα των ενεργειών προσαρμοστικότητας διαχωρίζεται από τις προδιαγραφές της σελίδας. Ο στόχος είναι να αναδειχτούν οι 2 διαφορετικές λογικές που απορρέουν από τον ρόλο που διαδραματίζουν οι σελίδες και τα νέφη περιβάλλουσας κατάστασης. Ενώ η πρώτη λογική δρα ως πάροχος περιεχομένων και υπηρεσιών, αντίθετα η δεύτερη λογική δρα ως εργαλείο τροποποίησης των εν λόγω περιεχομένων και υπηρεσιών.

Το νέφος περιβάλλουσας κατάστασης συνδέεται με την σελίδα μέσω μιας κατευθυνόμενης ακμής, δηλαδή μέσω ενός συνδέσμου που εξέρχεται από την ετικέτα «C». Αυτός ο σύνδεσμος εξασφαλίζει την επικοινωνία μεταξύ της λογικής της σελίδας και της λογικής του νέφους αφού μπορεί να μεταφέρει τις παραμέτρους που προκύπτουν από το περιεχόμενο της σελίδας, το οποίο με τη σειρά του μπορεί να είναι χρήσιμο για τον υπολογισμό ενεργειών που καθορίζονται στο νέφος. Επίσης, ένας σύνδεσμος από το νέφος προς τη σελίδα μπορεί να μεταφέρει παραμέτρους της περιβάλλουσας κατάστασης ή γενικότερα, τιμές υπολογισμένες από τις ενέργειες προσαρμοστικότητας, οι οποίες με τη σειρά τους ενδέχεται να επηρεάσουν την

προσαρμοστικότητα των περιεχομένων της σελίδας σε σχέση με τη προσφάτως ανανεωμένη περιβάλλουσα κατάσταση.

2.3.3.4 Περιέκτες με επίγνωση της περιβάλλουσας κατάστασης

Το κεντρικό στοιχείο επίγνωσης της περιβάλλουσας κατάστασης είναι η σελίδα. Ωστόσο, όπως απεικονίζεται στην εικόνα 4, προτείνεται επίσης ο καθορισμός περιεκτών (containers) με επίγνωση της περιβάλλουσας κατάστασης (δηλαδή όψεις ιστοχώρου και περιοχές, με βάση την ορολογία της WebML) ως δομές ομαδοποίησης καταστάσεων. Αυτές οι καταστάσεις επιτρέπουν να καθοριστούν μόνο μία φορά ενέργειες προσαρμοστικότητας που πρέπει να εκτελεστούν για κάθε που έχει χαρακτηριστεί με ετικέτα «C» και περιλαμβάνεται στον περιέκτη.

Η εικόνα 4 απεικονίζει επίσης τη δυνατότητα της ιεραρχικής μεταφοράς παραμέτρων από ένα εξωτερικό νέφος σε ένα εσωτερικό. Πιο συγκεκριμένα, εάν η αποτίμηση του εξωτερικού νέφους παράγει αποτελέσματα που επαναχρησιμοποιούνται σ' ένα εσωτερικό επίπεδο, όπως συμβαίνει σε ορισμένες παραμέτρους της περιβάλλουσας κατάστασης, τότε επιστρέφει αυτές τις τιμές στην ετικέτα «C» που ενεργοποίησε τον υπολογισμό του νέφους. Στο εσωτερικό επίπεδο, αυτές οι παράμετροι μπορούν εν συνεχεία να «καταναλώνονται» στα νέφη περιβάλλουσας κατάστασης. Η μεταφορά παραμέτρων από εξωτερικούς περιέκτες στο τρέχον νέφος περιβάλλουσας κατάστασης πραγματοποιείται μέσω του συνδέσμου ενεργοποίησης νέφους. Στο τέλος της αλυσίδας ενεργειών προσαρμοστικότητας, οι σύνδεσμοι που εξέρχονται από το τελευταίο νέφος που αποτιμήθηκε, μπορεί να μεταφέρουν τιμές παραμέτρων για τον υπολογισμό των μονάδων της σελίδας.

Οι τυπικές ενέργειες που πρέπει να καθοριστούν στο επίπεδο των περιεκτών είναι η απόκτηση νέων δεδομένων της περιβάλλουσας κατάστασης και η συνεπαγόμενη ενημέρωση του μοντέλου περιβάλλουσας κατάστασης. Προτείνονται συνεπώς, δύο επίπεδα για τον προσδιορισμό των ενεργειών προσαρμοστικότητας της περιβάλλουσας κατάστασης:

- **Ενέργειες για τη διαχείριση του μοντέλου περιβάλλουσας κατάστασης**, με σκοπό τον καθορισμό των πράξεων της απόκτησης δεδομένων περιβάλλουσας κατάστασης και της συνεπαγόμενης ενημέρωσης του μοντέλου περιβάλλουσας κατάστασης. Αυτές οι ενέργειες προσαρμοστικότητας πρέπει να εκτελούνται πριν από την εκτέλεση οποιασδήποτε άλλης ενέργειας μέσα στο νέφος περιβάλλουσας κατάστασης, για τη συγκέντρωση μιας ενημερωμένης εικόνας της τρέχουσας περιβάλλουσας κατάστασης. Ως εκ τούτου, μπορούν να συσχετίζονται με τους περισσότερους εξωτερικούς περιέκτες (όψεις ή περιοχές), και κληρονομούνται από όλους τους εσωτερικούς περιέκτες (περιοχές ή σελίδες).
- **Ενέργειες για την προσαρμοστικότητα του υπερκειμένου**, με σκοπό τον ορισμό των κανόνων προσαρμογής σελίδας και προσαρμογής πλοήγησης που σχετίζονται με σελίδες που έχουν χαρακτηριστεί με ετικέτα «C».

2.3.4 Καθορισμός προσαρμοστικότητας

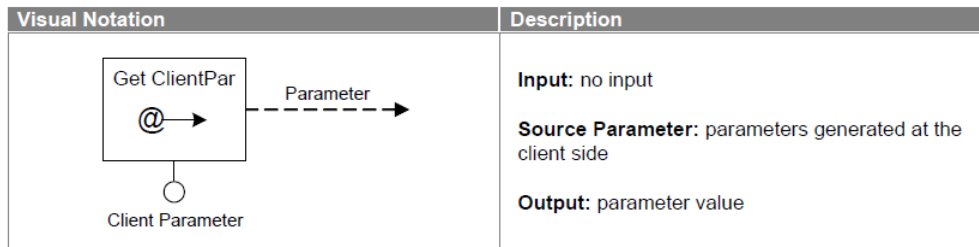
Η μελετούμενη προσέγγιση εισάγει την ομαδοποίηση εντός των νεφών περιβάλλουσας κατάστασης του προσδιορισμού των ενεργειών. Στη συνέχεια παρουσιάζονται νέες δομές μοντελοποίησης WebML, οι οποίες εξασφαλίζουν πλήρη κάλυψη του καθορισμού της διαχείρισης του μοντέλου περιβάλλουσας κατάστασης καθώς και των ενεργειών προσαρμοστικότητας του υπερκειμένου. Ο καθορισμός των ενεργειών αυτών επιτρέπει τον προσδιορισμό ενεργειών για την απόκτηση και ενημέρωση των δεδομένων περιβάλλουσας κατάστασης, καθώς και κανόνων *συμβάν-συνθήκη-ενέργεια* (Event-Condition-Action – ECA), όπου το συμβάν αναφέρεται το αίτημα της σελίδας, η συνθήκη για ενεργοποίηση του κανόνα αναφέρεται στην αποτίμηση των παραμέτρων περιβάλλουσας κατάστασης που αποκτήθηκαν προηγουμένως, και η ενέργεια συνίσταται στην προσαρμογή του υπερκειμένου.

2.3.4.1 Διαχείριση δεδομένων περιβάλλουσας κατάστασης

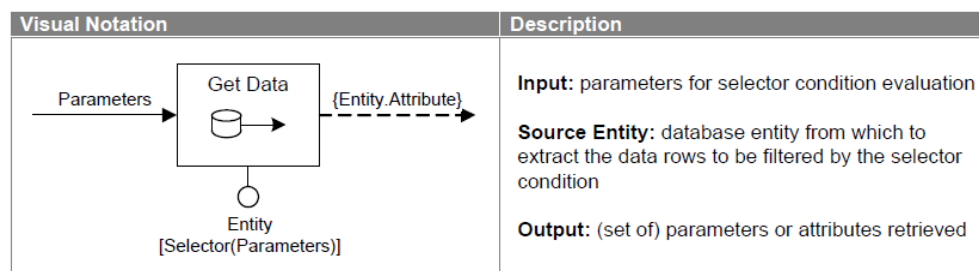
Για την πραγματοποίηση της προσαρμοστικότητας σε σχέση με την τρέχουσα περιβάλλουσα κατάσταση, η εφαρμογή πρέπει να είναι σε θέση να αποκτήσει και να διαχειριστεί τις παραμέτρους περιβάλλουσας κατάστασης, σύμφωνα με τους μηχανισμούς που αναφέρθηκαν στην ενότητα των λειτουργικών απαιτήσεων συστήματος. Για παράδειγμα, δεδομένα περιβάλλουσας κατάστασης που παρέχονται ως παράμετροι επιπέδου συνεδρίας HTTP, σε επίπεδο WebML μπορούν να αναπαρασταθούν από μονάδες τύπου Get. Ωστόσο, απαιτούνται νέες βασικές έννοιες και μηχανισμοί για:

- **Καθορισμό της απόκτησης νέων δεδομένων περιβάλλουσας κατάστασης που ανιχνεύεται από την πλευρά του πελάτη.** Μια νέα μονάδα Get *ClientPar* (όπως φαίνεται στην εικόνα 5) είναι αναγκαία για την υποστήριξη της ανάκτησης των παραμέτρων που παράγονται από την πλευρά του πελάτη και την αποστολή τους στην εφαρμογή.
- **Καθορισμός της απόκτησης δεδομένων περιβάλλουσας κατάστασης από το μοντέλο περιβάλλουσας κατάστασης.** Στη WebML υπάρχει ο τύπος μονάδας *Get Data* (όπως φαίνεται στην εικόνα 6) για την ανάκτηση των τιμών από τις πηγές δεδομένων, σύμφωνα με την κατάσταση ενός επιλογέα. Η σημασιολογία της είναι παρόμοια με εκείνη των μονάδων δημοσίευσης περιεχομένου, με τη διαφορά του ότι τα δεδομένα που ανακτηθήκαν, δημοσιεύονται σε μια σελίδα, αλλά χρησιμοποιούνται μόνο ως είσοδος σε ακόλουθες μονάδες ή λειτουργίες.
- **Ενημέρωση του μοντέλου περιβάλλουσας κατάστασης.** Μόλις ανακτηθούν οι νέες παράμετροι περιβάλλουσας κατάστασης, θα πρέπει να χρησιμοποιηθούν για την ενημέρωση του μοντέλου περιβάλλουσας κατάστασης στο επίπεδο των δεδομένων. Η ενέργεια αυτή συνίσταται στην τροποποίηση των τιμών που έχουν ήδη αποθηκευτεί στο αρχείο προέλευσης των δεδομένων. Όσον αφορά τη WebML, οι προδιαγραφές του χρησιμοποιούν μονάδες λειτουργίας (όπως αναφέρθηκε στο υποκεφάλαιο της παρουσίασης της WebML) που παρέχουν υποστήριξη για τις πιο κοινές λειτουργίες διαχείρισης βάσεων δεδομένων, δηλαδή τροποποίηση, εισαγωγή, διαγραφή (*modify, insert, delete*).

- **Παρακολούθηση του μοντέλου περιβάλλουσας κατάστασης.** Αυτή η λειτουργία επιτυγχάνεται με χρήση ενός περιοδικού μηχανισμού ελέγχου όπου ερωτάται το μοντέλο περιβάλλουσας κατάστασης και εντοπίζονται οι μεταβολές της περιβάλλουσας κατάστασης της σελίδας που συνεπάγονται εκτέλεση ενεργειών προσαρμοστικότητας.



Εικόνα 5 - Μονάδα Get URL Parameter



Εικόνα 6 - Μονάδα Get Data

2.3.4.2 Αποτίμηση των συνθηκών

Η εκτέλεση των ενεργειών προσαρμοστικότητας μπορεί να εξαρτάται από την αποτίμηση ορισμένων συνθηκών. Το συχνότερα επαναλαμβανόμενο πρότυπο αποτελείται από την αποτίμηση του εάν η περιβάλλουσα κατάσταση έχει αλλάξει, κι ως εκ τούτου απαιτείται η εκτέλεση ενεργειών προσαρμοστικότητας. Η αποτίμηση συνθηκών συνηθέστατα χρησιμοποιείται στα πλαίσια εντολών if και switch.

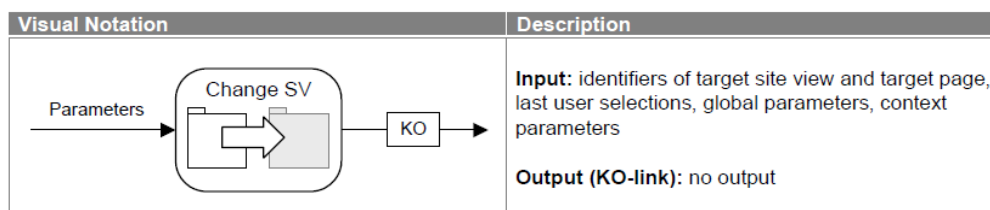
2.3.4.3 Εκτέλεση ενεργειών προσαρμοστικότητας

Μόλις καθοριστεί η τρέχουσα περιβάλλουσα κατάσταση και αποτιμηθούν οι πιθανές συνθήκες, τότε μπορούν να ενεργοποιηθούν ενέργειες προσαρμοστικότητας για την προσαρμογή των περιεχομένων της σελίδας, της πλοήγησης, της τρέχουσας όψης του ιστοχώρου καθώς και του στυλ παρουσίασης, όπως παρουσιάζονται παρακάτω:

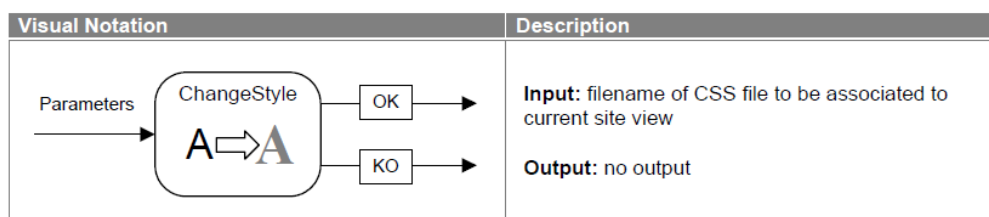
- **Προσαρμογή περιεχομένων σελίδας.** Τα περιεχόμενα της σελίδας προσαρμόζονται μέσω κατάλληλων επιλογών δεδομένων, των οποίων ο ορισμός βασίζεται σε παραμέτρους περιβάλλουσας κατάστασης που είτε ανακτώνται από το μοντέλο περιβάλλουσας κατάστασης είτε θα υπολογιστούν εντός του νέφους περιβάλλουσας κατάστασης της σελίδας. Η χρήση των επιλογών επιτρέπει επίσης το φιλτράρισμα των δεδομένων, όσον αφορά την τρέχουσα περιβάλλουσα κατάσταση.
- **Προσαρμογή πλοήγησης.** Σε ορισμένες περιπτώσεις, το αποτέλεσμα της αποτίμησης της συνθήκης στο νέφος περιβάλλουσας κατάστασης μπορεί

να είναι μια αυτόματη ενέργεια πλοήγησης που προκαλεί την ανακατεύθυνση σε διαφορετική σελίδα. Ο καθορισμός των αυτόματων πλοηγήσεων που ενεργοποιούνται από τη περιβάλλουσα κατάσταση, απαιτεί τη σύνδεση ενός από τους συνδέσμους που εξέρχονται από το νέφος περιβάλλουσας κατάστασης, σε μια αυθαίρετα επιλεγόμενη σελίδα-προορισμό του υπερκειμένου, για την ανακατεύθυνση του χρήστη σ' αυτή τη σελίδα. Ως εκ τούτου, οι σύνδεσμοι που εξέρχονται από το νέφος περιβάλλουσας κατάστασης και κατευθύνονται σε άλλες σελίδες, αντιπροσωπεύουν τις καλούμενες αυτόματες ενέργειες πλοήγησης.

- Προσαρμογή όψης ιστοχώρου.** Οι αλλαγές στη περιβάλλουσα κατάσταση αλληλεπίδρασης μπορεί στην πραγματικότητα να απαιτούν μια συνολική αναδιάρθρωση του συνόλου του υπερκειμένου. Τέτοιες περιπτώσεις απαντώνται είτε επειδή έχει αλλάξει η συσκευή του χρήστη, είτε επειδή ο χρήστης μετατοπίζεται σε μια διαφορετική δραστηριότητα. Συνεπώς, εισάγεται μια μονάδα *Change Site View* (όπως φαίνεται στην εικόνα 7), η οποία δέχεται ως είσοδο τα αναγνωριστικά μιας όψης ιστοχώρου καθώς και μίας σελίδας. Επιπλέον, ο εισερχόμενος σύνδεσμος μεταφέρει τις παραμέτρους που χαρακτηρίζουν την τρέχουσα κατάσταση της αλληλεπίδρασης, δηλαδή:
 - Τις παραμέτρους εισόδου της σελίδας, που αντιπροσωπεύουν τις τελευταίες επιλογές του χρήστη.
 - Γενικές παραμέτρους, που αντιπροσωπεύουν τα δεδομένα μιας συνεδρίας, καθώς και κάποιες παλαιότερες επιλογές του χρήστη.
 - Παραμέτρους που χαρακτηρίζουν τη παρούσα περιβάλλουσα κατάσταση: οι παράμετροι αυτές ανακτήθηκαν στον τελευταίο κύκλο απόκτησης των δεδομένων που πραγματοποιήθηκε και δεν έχουν ακόμη εντελώς αποθηκευτεί στο μοντέλο περιβάλλουσας κατάστασης.
- Προσαρμογή στυλ παρουσίασης.** Μερικές φορές, οι αλλαγές περιβάλλουσας κατάστασης μπορεί να απαιτούν λεπτομερείς προσαρμογές των ιδιοτήτων μιας παρουσίασης (όπως συμβαίνει για παράδειγμα λόγω των διαφορετικών συνθηκών φωτεινότητας) κι όχι μια πλήρη αναδιάρθρωση του συνολικού υπερκειμένου. Στη παρούσα προσέγγιση εισάγεται η μονάδα *Change Style* για τη δυναμική επιλογή ιδιοτήτων στυλ παρουσίασης (όπως φαίνεται στην εικόνα 8). Οι ιδιότητες στυλ συλλέγονται σε κατάλληλα αρχεία .css (Cascading Style Sheet), και η μονάδα επιτρέπει κατά τη διάρκεια του χρόνου εκτέλεσης, την αλλαγή του αρχείου .css που σχετίζεται με την εφαρμογή.



Εικόνα 7 - Μονάδα Change Site View



Εικόνα 8 - Μονάδα Change Style

2.4 Παράδειγμα ανάπτυξης εφαρμογής σύμφωνα με την προσέγγιση

Το πλαίσιο εργασίας που περιγράφηκε ανωτέρω σχετικά με τη μοντελοποίηση έχει εφαρμοστεί στο Ιταλικό ερευνητικό έργο MAIS (Multichannel Adaptive Information Systems). Για την επικύρωση των αποτελεσμάτων του έργου, έχει θεωρηθεί ως μελέτη περίπτωσης ένα ολοκληρωμένο πληροφοριακό σύστημα τουρισμού (Integrated Tourist Information System, ITIS). Πιο συγκεκριμένα, το σύστημα αυτό παρέχει στους τουρίστες ενημερωμένες, εξατομικευμένες και με επίγνωση τοποθεσίας, πληροφορίες. Στη συνέχεια παρουσιάζεται μια απλοποιημένη έκδοση του συνολικού σχεδιασμού του συστήματος.

Με τον συνδυασμό μηχανισμών εντοπισμού τοποθεσίας εξωτερικού και εσωτερικού χώρου, το ITIS επιτρέπει την υποστήριξη σε πληροφοριακό υλικό τόσο σε αδρή όσο και σε λεπτομερική διακρίτοτητα. Όσον αφορά τους μηχανισμούς εξωτερικού χώρου, το ITIS παρέχει κυρίως χάρτες της πόλης, περιγραφές αξιοθέατων καθώς και πιθανές διαφημίσεις κοντινών εστιατορίων. Όσον αφορά τους μηχανισμούς εσωτερικού χώρου, τα διαθέσιμα περιεχόμενα ορίζονται αυτόνομα από τους διαχειριστές των αντίστοιχων δομών και διαφέρουν από κτίριο σε κτίριο. Για παράδειγμα, το μουσείο μιας πόλης παρέχει στους επισκέπτες του περιγραφές της τοποθεσίας των έργων τέχνης και των καλλιτεχνών.

Για τον εντοπισμό της θέσης των τουριστών στον εξωτερικό χώρο, το σύστημα χρησιμοποιεί τις συντεταγμένες που αντλεί από το GPS (Global Positioning System), ενώ ο εντοπισμός σε εσωτερικό χώρο επιτυγχάνεται με τη βοήθεια ενεργών ετικετών RFID. Κάθε τουρίστας έχει ένα PDA, ένα GPS και μια προσωπική ετικέτα RFID, τα οποία είναι ενδεχομένως ενσωματωμένα στο PDA. Οι αιτήσεις HTTP που κατευθύνονται προς το ITIS, εμπλουτίζονται με τις συντεταγμένες του GPS μέσω ενός μικρού εξυπηρέτη αντιπροσώπευσης (ο οποίος είναι εγκατεστημένος στο PDA του τουρίστα), ο οποίος με τη σειρά του διαχειρίζεται την επικοινωνία μεταξύ του PDA και της μονάδας GPS. Οι εσωτερικές θέσεις που έχουν ανιχνευθεί εισάγονται απευθείας στο μοντέλο περιβάλλουσας κατάστασης της εφαρμογής από τις υποδομές ανίχνευσης.

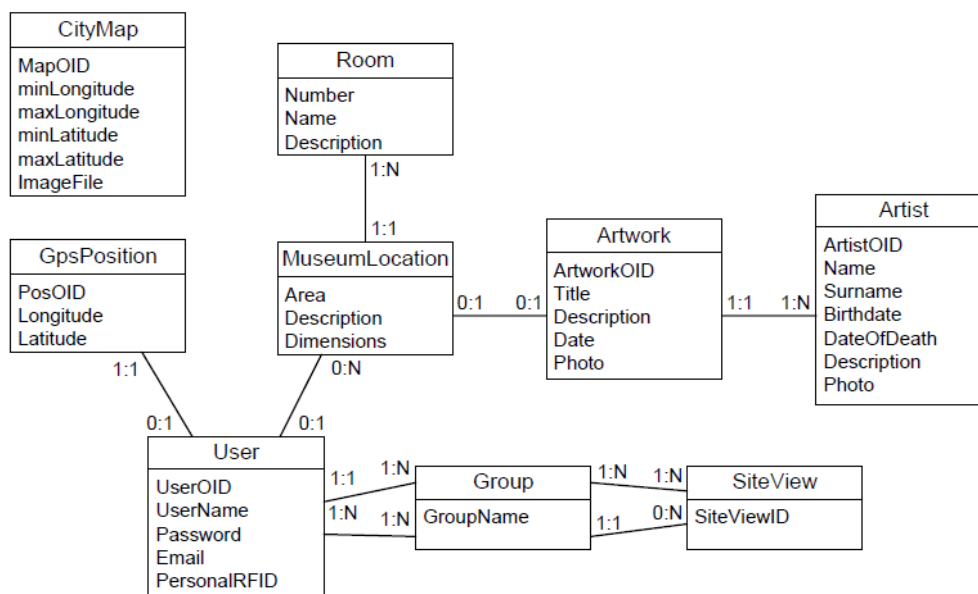
Η εικόνα 9 απεικονίζει το μοντέλο δεδομένων που χρησιμοποιείται από το ITIS. Εκτός από τα δεδομένα της εφαρμογής (δηλαδή τις οντότητες CityMap, Room, Artwork και Artist), υπάρχουν και οι οντότητες GpsPosition και MuseumLocation που δημιουργούν το μοντέλο περιβάλλουσας κατάστασης της εφαρμογής. Ενώ οι συντεταγμένες του GPS που συνδέονται με τους χρήστες, ενημερώνονται μέσω των κατάλληλων πράξεων στο επίπεδο του υπερκειμένου (όπως φαίνεται και στην εικόνα 10), η αντιστοίχιση των τοποθεσιών με τους χρήστες μέσα στο μουσείο συντηρείται από τις εσωτερικές υποδομές ανίχνευσης, με βάση το χαρακτηριστικό PersonalRFID

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

του κάθε χρήστη. Πιο συγκεκριμένα, όταν εισέρχεται ένας χρήστης σε μία από τις περιοχές του μουσείου, του αποκαλύπτεται το PersonalRFID και ενημερώνεται η αντιστοίχιση περιοχή-χρήστη.



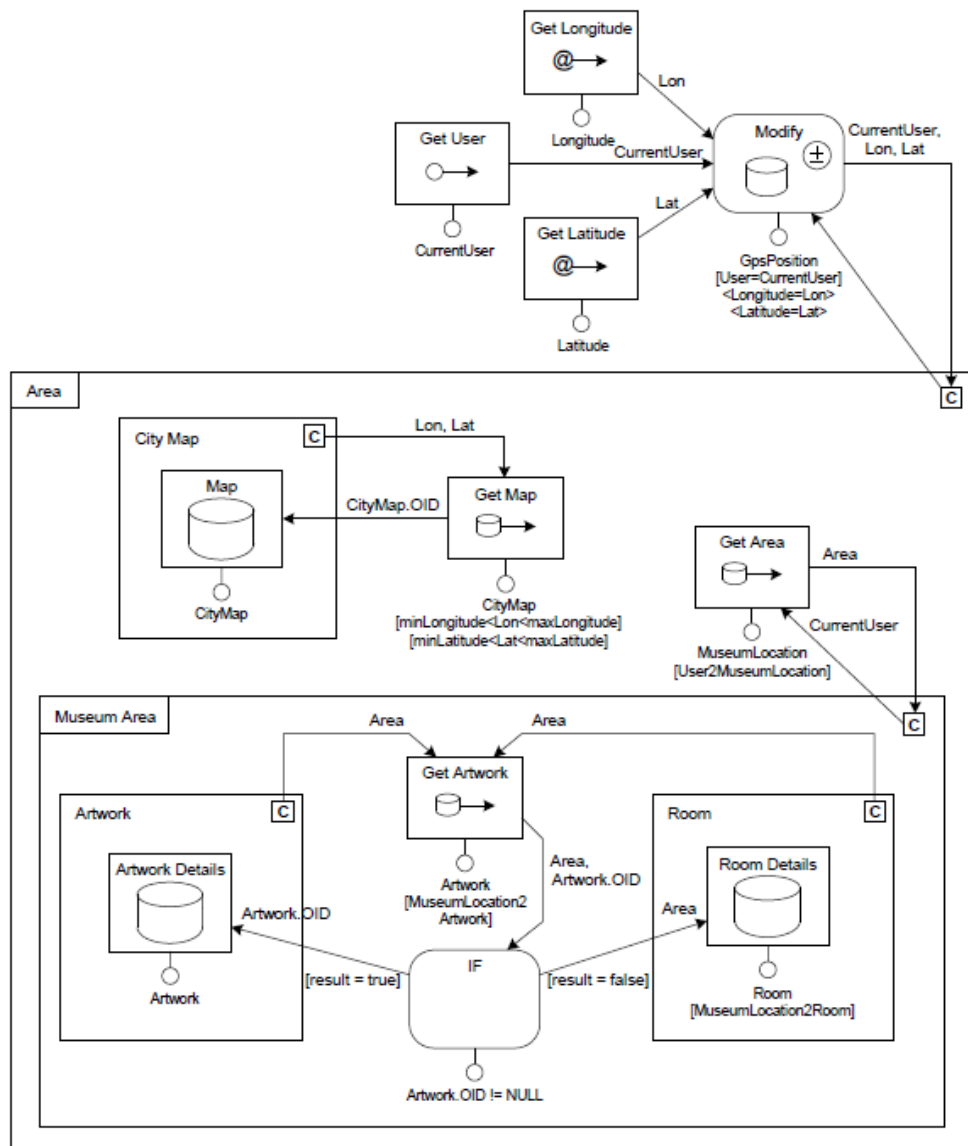
Εικόνα 9 - Σχήμα δεδομένων για το σύστημα ITIS

Το μοντέλο υπερκειμένου της εφαρμογής Διαδικτύου ITIS που απεικονίζεται στην εικόνα 10, είναι κατασκευασμένο με βάση το μοντέλο δεδομένων της εικόνας 9. Το σχήμα περιλαμβάνει τόσο σελίδες με ετικέτα «C» όσο και περιοχές με ετικέτα «C», και δείχνει τις εναλλακτικές πλοηγήσεις που εκκινούνται από την περιβάλλουσα κατάσταση. Κάθε περιέκτης έχει τη δική του αλυσίδα ενεργειών προσαρμοστικότητας, και ορισμένες παράμετροι περιβάλλουσας κατάστασης μεταβιβάζονται ιεραρχικά από τα πιο εξωτερικά νέφη περιβάλλουσας κατάστασης στα πιο εσωτερικά. Για παράδειγμα, οι παράμετροι γεωγραφικού μήκους και πλάτους (*Lon* και *Lat*), όπως και η παράμετρος *CurrentUser* παράγονται στην πιο εξωτερική περιοχή και καταναλώνονται από το νέφος της σελίδας City Map. Επίσης, η παράμετρος *Area* παράγεται από τις ενέργειες προσαρμοστικότητας που συσχετίζονται με το *Museum Area*, και καταναλώνεται για την προσαρμογή των σελίδων *Artwork* και *Room*.

Η πιο εξωτερική περιοχή συσχετίζεται με τις ενέργειες για την απόκτηση νέων δεδομένων περιβάλλουσας κατάστασης και για την ενημέρωση του μοντέλου περιβάλλουσας κατάστασης. Πιο συγκεκριμένα, η διαχείριση της τρέχουσας εξωτερικής τοποθεσίας των χρηστών πραγματοποιείται σύμφωνα με τα ακόλουθα βήματα:

- Η τιμή της παραμέτρου συνεδρίας *CurrentUser*, η οποία φυλάσσει το αναγνωριστικό του τρέχοντος χρήστη, ανακτάται μέσω της μονάδας *Get User*.
- Η τιμή των παραμέτρων πλευράς εξυπηρετούμενου, που αντιπροσωπεύουν το τρέχον γεωγραφικό μήκος και πλάτος, ανακτώνται μέσω των μονάδων *Get Longitude* και *Get Latitude*.

- Η νέα τοποθεσία GPS συσχετίζεται με τον προσδιορισθέντα χρήστη, τροποποιώντας την αντίστοιχη οντότητα δεδομένων (μέσω της μονάδας *Modify*).



Εικόνα 10 - Σχήμα υπερκειμένου για την εφαρμογή ITIS

Αυτά τα βήματα εκτελούνται κάθε φορά που αποτιμώνται τα νέφη περιβάλλουσας κατάστασης που συσχετίζονται με μία από τις εσωτερικές σελίδες με ετικέτα «C». Προκειμένου να εξασφαλισθεί η συνοχή του μηχανισμού μεταβίβασης των παραμέτρων περιβάλλουσας κατάστασης, τα βήματα εκτελούνται *πριν* από την αποτίμηση οποιουδήποτε εσωτερικού νέφους περιβάλλουσας κατάστασης. Η τιμή της ιδιότητας *Adaptivity_Policy* τόσο για την περιοχή όσο και για τις σελίδες της, έχει τεθεί στην τιμή *αναβαλλόμενη (deferred)*. Ως εκ τούτου τα νέφη περιβάλλουσας κατάστασης, από τα πιο εξωτερικά ως τα εσωτερικά, αποτιμώνται μόνο όταν δημιουργούνται αυτόματες ανανεώσεις που ακολουθούν τις αιτήσεις των χρηστών. Με άλλα λόγια, όταν ένας χρήστης βλέπει τη σελίδα *City Map*, πριν από την εκτέλεση των ενεργειών προσαρμοστικότητας στην ανανέωση της σελίδας, εκτελείται το εξωτερικό νέφος και τίθενται οι κατάλληλες τιμές στις παραμέτρους

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

CurrentUser, *Lon* και *Lat*. Μόνο όταν έχει ολοκληρωθεί αυτό το βήμα μπορούν να εκτελεστούν οι ενέργειες προσαρμοστικότητας της σελίδας που απαιτούν τις παραμέτρους *Lon* και *Lat* ως είσοδο, και στη συνέχεια επιλέγεται ένας κατάλληλος χάρτης της πόλης για να προβληθεί η απάντηση HTTP.

Από την άλλη πλευρά, οι σελίδες που περιέχονται στην περιοχή *Museum Area*, κατασκευάζονται με βάση τα δεδομένα τοποθεσίας εσωτερικού χώρου, που ανακτώνται μέσω του νέφους περιβάλλουσας κατάστασης, το οποίο με τη σειρά του συσχετίζεται με το *Museum Area*. Επίσης, για την περιοχή αυτή, η τιμή της ιδιότητας *Adaptivity_Policy* ορίζεται ξανά ως *αναβαλλόμενη*. Η μοναδική λειτουργία του νέφους καταναλώνει στην είσοδο την παράμετρο *CurrentUser* και παράγει στην έξοδο την παράμετρο *Area*, η οποία τελικά μεταβιβάζεται σε εσωτερικές σελίδες.

Η σελίδα *Artwork* για παράδειγμα, χρησιμοποιεί την παράμετρο *Area* για την αποτίμηση του κανόνα προσαρμοστικότητάς της, η οποία παράμετρος ανακτά την ταυτότητα του έργου τέχνης που συνδέεται με την τρέχουσα τοποθεσία του χρήστη. Επιπλέον, η παράμετρος αυτή ελέγχει αν η ταυτότητα δεν είναι μηδενική (το οποίο θα σήμαινε ότι δεν υπάρχουν έργα τέχνης σε αυτή τη τοποθεσία) κι ενημερώνει το περιεχόμενο της σελίδας *Artwork* εάν η ανακτώμενη ταυτότητα περιέχει ένα έγκυρο αναγνωριστικό. Διαφορετικά, μη μπορώντας να παράσχει ουσιαστικές λεπτομέρειες για κάποιο έργο τέχνης, ο κανόνας ανακατευθύνει το χρήστη στη σελίδα *Room*, με σκοπό τη δημοσίευση λεπτομερειών όσον αφορά τον τρέχοντα εκθεσιακό χώρο. Οι σελίδες *Room* και *Artwork* μοιράζονται τις ίδιες ενέργειες προσαρμοστικότητας, οι οποίες ενεργοποιούν αυτόματες πλοηγήσεις μεταξύ των δύο σελίδων σε συνάρτηση με τη διαθεσιμότητα των κατάλληλων δεδομένων έργων τέχνης.

2.4.1 Υπολογισμός σελίδων με επίγνωση της περιβάλλουσας κατάστασης

Ο αλγόριθμος υπολογισμού για τις συμβατικές σελίδες WebML βασίζεται στον υπολογισμό των εσωτερικών μονάδων της σελίδας. Ξεκινά από τον υπολογισμό όλων των μονάδων που δεν λαμβάνουν κανένα σύνδεσμο στην είσοδό τους, κι ως εκ τούτου δεν απαιτείται καμία παράμετρος για τον υπολογισμό τους. Στη συνέχεια, προχωράει στις εξωτερικές μονάδες για τις οποίες υπάρχουν επαρκείς τιμές εισόδου στις παραμέτρους που μεταβιβάζονται στη σελίδα. Ως εκ τούτου, μέχρις ότου υπολογιστούν όλες οι δυνατές μονάδες μέσα στη σελίδα, ο αλγόριθμος επιλέγει επαναληπτικά τη μονάδα που πρέπει να υπολογιστεί στη συνέχεια, με βάση τις ακόλουθες συνθήκες:

- Όλες οι υποχρεωτικές παράμετροι εισόδου της μονάδας πρέπει να έχουν μία τιμή.
- Όλες οι μονάδες που θα μπορούσαν να παρέχουν μια τιμή σε μια παράμετρο εισόδου της μονάδας, πρέπει να έχει ήδη υπολογιστεί.

Στον υπολογισμό των σχημάτων υπερκειμένου με υποστήριξη προσαρμοστικότητας θα πρέπει να προσαρμοστεί η λογική της σελίδας για την κάλυψη της αυτόματης εκτέλεσης ενεργειών προσαρμοστικότητας που συσχετίζονται τόσο με τις σελίδες όσο και με τους περιέκτες τους. Συνεπώς, ο υπολογισμός των σελίδων προσαρμοστικότητας δεν πρέπει να αντιμετωπίσει μόνο το πρόβλημα του τρόπου υπολογισμού των μονάδων που περιέχονται στη σελίδα, αλλά πρέπει επίσης να διασφαλίσει τη σωστή ενεργοποίηση κι εκτέλεση των συσχετιζόμενων νεφών

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

περιβάλλουσας κατάστασης. Όπως παρουσιάστηκε στην εικόνα 4, αυτό προϋποθέτει την αξιολόγηση κάθε νέφους περιβάλλουσας κατάστασης πριν από τον υπολογισμό της πραγματικής σελίδας, αρχίζοντας από το πιο εξωτερικό νέφος και προχωρώντας προς το πιο εσωτερικό, κ.λπ. Έτσι, η λογική υπολογισμού απλών σελίδων διατηρεί την εγκυρότητά της, αλλά επιπλέον χειρίζεται κατάλληλα πράξεις προσαρμοστικότητας που ενδεχομένως υπάρχουν, εξετάζοντας τα νέφη περιβάλλουσας κατάστασης που ορίζονται για τις σελίδες και τους περιέκτες τους. Αυτή η συμπεριφορά μπορεί να συνοψιστεί από την ακόλουθη λειτουργία *codeGen*, όπου οι λειτουργίες *buildContext* και *buildPage* εκτελούν τον υπολογισμό του νέφους περιβάλλουσας κατάστασης και της σελίδας, αντίστοιχα:

```

FUNCTION codeGen(C:Container)
BEGIN
  IF (contextAware(C) THEN {
    IF (included(C,C') AND contextAware(C')) THEN
      codeGen(C');
    newPage = buildContext(C);
    IF (newPage != null) THEN
      codeGen(newPage);
  }
  IF (isPage(C)) THEN
    buildPage(C);
END

```

Ειδικότερα, η λειτουργία *buildContext* επιστρέφει μια τιμή η οποία, όταν η αποτίμηση του νέφους περιβάλλουσας κατάστασης ενεργοποιεί μια αυτόματη πλοήγηση σε μια διαφορετική σελίδα *newPage*, αποτελεί ένα δείκτη προς τη σελίδα προορισμού. Αυτή η τιμή είναι null όταν δεν απαιτείται αυτόματη πλοήγηση. Επίσης, το αν η ενεργοποίηση του υπολογισμού του νέφους περιβάλλουσας κατάστασης εντός της λειτουργίας *buildContext* θα είναι η αναβαλλόμενη ή άμεση εξαρτάται από την πολιτική προσαρμοστικότητας που σχετίζεται με τον περιέκτη με χαρακτηρισμό C που αποτιμάται.

2.4.1.1 Κανόνες ειδίκευσης

Σε ορισμένες ρυθμίσεις μιας σελίδας μπορεί μια μονάδα να έχει πολλαπλούς εισερχόμενους συνδέσμους, οι οποίοι να αναθέτουν τιμές στην ίδια παράμετρο. Δεδομένου ότι μπορεί να κρατηθεί μόνο μία τιμή, αυτό έχει ως αποτέλεσμα ο υπολογισμός της μονάδας να είναι αμφίσημος. Έτσι, είναι απαραίτητοι μερικοί κανόνες ειδίκευσης για να αποφασισθεί ποιά από τις εισερχόμενες τιμές θα χρησιμοποιηθεί.

Για απλές σελίδες, η ειδίκευση των παραμέτρων εισόδου αποτιμάται σύμφωνα με τις ακόλουθες αρχές:

- Οι τιμές που απορρέουν από τις επιλογές του τρέχοντος χρήστη, που με τη σειρά τους εκφράζονται από το τελευταίο συμβάν πλοήγησης, είναι οι πιο συγκεκριμένες.
- Οι τιμές που εξαρτώνται από τις παλαιότερες επιλογές του χρήστη ή προέρχονται από καθολικές παραμέτρους που ανακτώνται μέσω μονάδων Get, είναι οι δεύτερες πιο συγκεκριμένες.

- Οι τιμές που απορρέουν μέσω ευρεστικών από το περιεχόμενο άλλων μονάδων, είναι οι λιγότερο συγκεκριμένες.

Για την περίπτωση των σελίδων με επίγνωση της περιβάλλουσας κατάστασης, οι κανόνες ειδίκευσης επεκτείνονται μέσω μιας νέας συνθήκης που πρέπει να αποτιμηθεί πριν από οποιονδήποτε άλλο κανόνα. Ο εν λόγω κανόνας ορίζει ότι οι τιμές που παράγονται από τον υπολογισμό του νέφους περιβάλλουσας κατάστασης της σελίδας (αν αυτό αποτιμηθεί) είναι οι πιο συγκεκριμένες. Ο νέος κανόνας ειδίκευσης θεωρεί τη περιβάλλουσα κατάσταση ως ένα νέο actor που μπορεί να προκαλέσει ενέργειες πλοήγησης ή την προσαρμοστικότητα της σελίδας.

Κατά συνεπή τρόπο σε σχέση με τις παραπάνω επιλογές, στη συνέχεια παρατίθεται μία κατηγοριοποίηση τριών πιθανών καταστάσεις που μπορεί να προκύψουν κατά την πρόσβαση σε σελίδες:

- **Σελίδες χωρίς επίγνωση της περιβάλλουσας κατάστασης**, όπου ισχύουν οι συνήθεις κανόνες ειδίκευσης, χωρίς να λαμβάνονται υπόψη τις τυχόν ενέργειες προσαρμοστικότητας, και οι μονάδες υπολογίζονται με το συνήθη τρόπο.
- **Πρόσβαση σε σελίδες με ετικέτες «C» και αναβαλλόμενη προσαρμοστικότητα**, όπου οι ενέργειες προσαρμοστικότητας που έχουν ενδεχομένως οριστεί για τέτοιου είδους σελίδες αγνοούνται όταν προσπελαίνεται για πρώτη φορά η σελίδα, προκειμένου να δοθεί στον χρήστη η υψηλότερη προτεραιότητα. Οι πιθανές ενέργειες προσαρμοστικότητας που έχουν καθοριστεί για τη σελίδα και τις εξωτερικές περιοχές της αποτιμώνται όταν λαμβάνουν χώρα οι αυτόματες ανανεώσεις, που παράγονται περιοδικά μετά την πρώτη προσπέλαση των χρηστών στη σελίδα. Αυτό μπορεί να οδηγήσει στην τροποποίηση των προτιμήσεων του χρήστη που έχουν καταγραφεί.
- **Πρόσβαση σε σελίδες με ετικέτες «C» και άμεση προσαρμοστικότητα**, όπου οι ενέργειες προσαρμοστικότητας αποτιμώνται σε κάθε αίτηση για τη σελίδα, συμπεριλαμβανομένης επίσης της πρώτης προσπέλασης της σελίδα από τον χρήστη.

Στην εικόνα 11 παρουσιάζονται τα βήματα που απαιτούνται κατά τον χρόνο εκτέλεσης, για τον υπολογισμό δυναμικών προτύπων σελίδας, και τονίζονται οι επιπρόσθετοι υπολογισμοί που απαιτούνται για τις σελίδες με επίγνωση της περιβάλλουσας κατάστασης. Η χρήση μιας παραμέτρου με το όνομα *automatic*, που επισυνάπτεται στην αίτηση HTTP για ανάκτηση της σελίδας, καθίσταται αναγκαία για τον προσδιορισμό του ότι η αίτηση για τη σελίδα έχει γίνει από τον αυτόματο μηχανισμό ανανέωσης κι όχι από τον ίδιο το χρήστη. Όπως φαίνεται και στην εικόνα, όταν ο εξυπηρέτης διαδικτύου λάβει ένα αίτημα νέας σελίδας, αποκωδικοποιεί τις εισερχόμενες παραμέτρους του αιτήματος και μόνο όσον αφορά τις σελίδες «C», επαληθεύει το αν είναι απαραίτητη μια προσαρμοστικότητα. Ο έλεγχος αυτός αποτελείται από:

- a) την αποτίμηση της τιμής που αποδόθηκε στην ιδιότητα της σελίδας *Adaptivity_Policy* και
- b) την επαλήθευση της ύπαρξης της παραμέτρου *automatic* στο URL του αιτήματος.

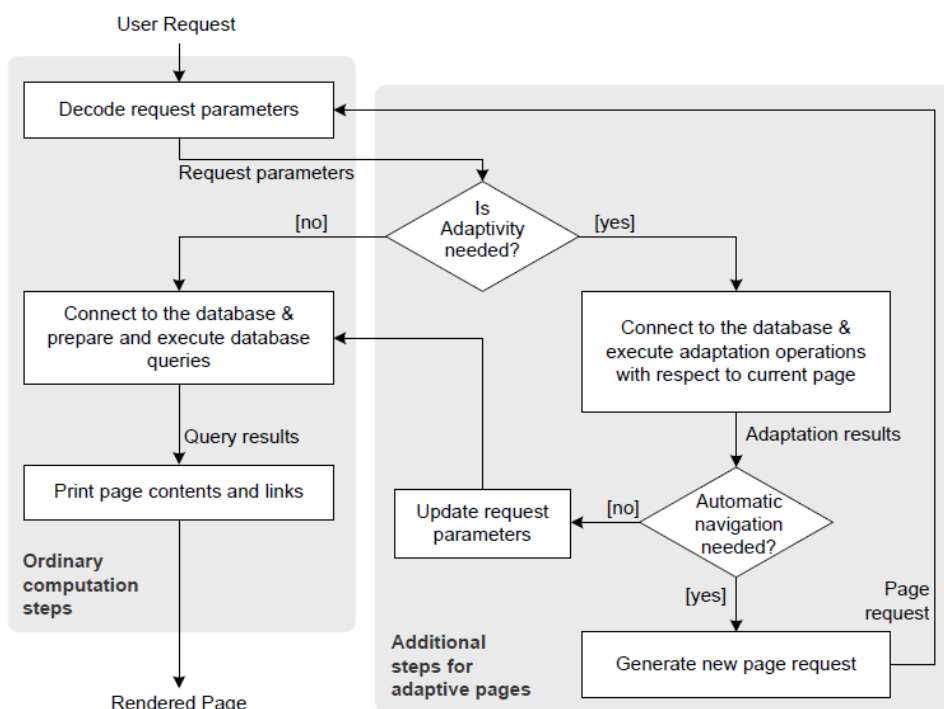
Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

Αναλόγως, ο υπολογισμός της σελίδας συνεχίζεται ως εξής:

- Αν δεν απαιτείται προσαρμοστικότητα (όπως είναι για συμβατικές σελίδες ή για την πρώτη προσπέλαση του χρήστη σε σελίδες με ετικέτα «C» και αναβαλλόμενη πολιτική), ο υπολογισμός συνεχίζεται όπως φαίνεται στην αριστερή πλευρά της εικόνας 11.
- Αν απαιτείται προσαρμοστικότητα, όπως είναι για παράδειγμα όταν η τιμή της ιδιότητας *Adaptivity_Policy* είναι «άμεσα» ή όταν η τιμή της ιδιότητας *Adaptivity_Policy* είναι «αναβαλλόμενη» και υπάρχει η παράμετρος *automatic*, τότε ο υπολογισμός συνεχίζεται όπως φαίνεται στη δεξιά πλευρά της εικόνας 11. Δύο διαφορετικές ενέργειες προσαρμοστικότητας μπορούν να ληφθούν:
 - *Προσαρμοστικότητα σελίδας*, όπου η βάση δεδομένων προσπελάζεται για την ανάγνωση της τρέχουσας περιβάλλουσας κατάστασης και ορισμένες παράμετροι ενημερώνονται αναλόγως. Έτσι ο υπολογισμός συνεχίζεται όπως και στις συνήθεις σελίδες. Η νέα τιμή των παραμέτρων περιβάλλουσας κατάστασης θα προκαλέσει την προσαρμογή του περιεχομένου ή του στυλ της σελίδας.
 - *Πλοήγηση προς μια διαφορετική σελίδα*, εντός ή εκτός της τρέχουσας όψης ιστοχώρου, όπου στην περίπτωση αυτή, η διαδικασία του υπολογισμού δημιουργεί ένα αίτημα νέας σελίδας, και η διαδικασία υπολογισμού της σελίδας με τη σειρά της, ξεκινά εκ νέου, χωρίς την χρήση της παραμέτρου *automatic*.



Εικόνα 11 - Υπολογισμός προτύπων σελίδας επίγνωσης της περιβάλλουσας κατάστασης

Αξίζει να σημειωθεί ότι οι ατέρμονες βρόχοι (infinite loops) με αποτιμήσεις της περιβάλλουσας κατάστασης που δεν ολοκληρώνονται θα μπορούσαν να προκύψουν κατά την εκτέλεση των αλυσίδων των νεφών περιβάλλουσας

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

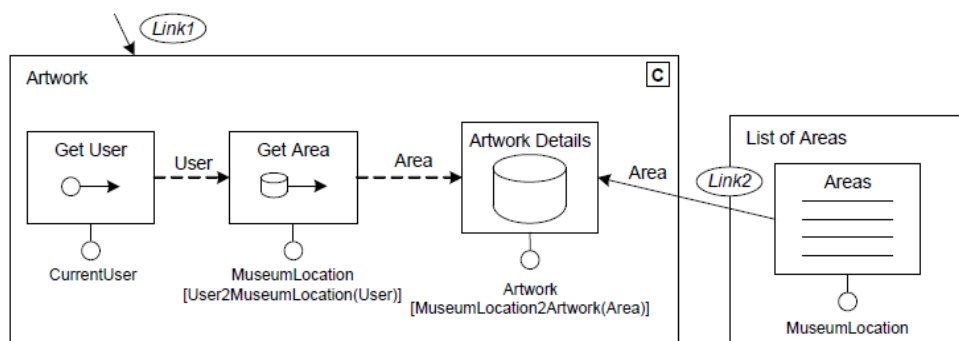
κατάστασης. Αυτό μπορεί να συμβεί όταν έχουμε αυτόματη πλοήγηση η οποία (α) ξεκινάει από μία σελίδα με ετικέτα «C» και «άμεση» προσαρμογή (β) καταλήγει σε μία σελίδα με ετικέτα «C» και «άμεση» προσαρμογή και (γ) το νέφος της σελίδας προορισμού βρίσκεται σε σύγκρουση με τις ενέργειες προσαρμογής που καθορίζονται για την αρχική σελίδα. Σε αυτή την περίπτωση θα μπορούσε να ανακατευθύνεται ο χρήστης πίσω στην αρχική σελίδα, κατόπιν στη σελίδα προορισμού (λόγω της άμεσης πολιτικής), και ούτω καθεξής.

Μία σχεδιαστική οδηγία για την πρόληψη ατέρμωνων βρόχων είναι η αποφυγή των κύκλων αυτόματης πλοήγησης όπου τόσο η αρχική σελίδα όσο και η σελίδα προορισμού έχουν άμεση προσαρμοστικότητα. Όταν πρέπει να καθοριστούν οι κύκλοι, συνιστάται να χρησιμοποιείται αναβαλλόμενη πολιτική για τις εμπλεκόμενες σελίδες. Αυτό εξασφαλίζει ότι η σελίδα προορισμού προσφέρεται στον χρήστη πριν από την εισαγωγή επιπρόσθετων ενεργειών προσαρμοστικότητας. Αυτό έχει ως αποτέλεσμα, ο χρήστης να μπορεί να διακόψει τον (πιθανό) κύκλο, απενεργοποιώντας τη λειτουργία με επίγνωση της περιβάλλουσας κατάστασης ή επιλέγοντας να πλοηγηθεί σε άλλη σελίδα.

2.4.1.2 Υπολογισμός σελίδων με επίγνωση της περιβάλλουσας κατάστασης

Στην υποενότητα αυτή παρουσιάζονται μερικά παραδείγματα υπολογισμού σελίδας με στόχο την αποσαφήνιση της παρεμβολής και των αλληλεπιδράσεων μεταξύ των ενεργειών πλοήγησης του χρήστη, των αυτόματα παραγόμενων αιτημάτων σελίδας καθώς και των ενεργειών προσαρμοστικότητας. Το παράδειγμα δείχνει ότι πρέπει να έχει κατανοηθεί πλήρως η λογική μοντελοποίησης της σελίδας όταν οι σελίδες περιλαμβάνουν επίγνωση της περιβάλλουσας κατάστασης.

Θεωρούμε τη σελίδα στην εικόνα 12, και υποθέτουμε ότι δεν διαθέτει επίγνωση της περιβάλλουσας κατάστασης, υπό την έννοια ότι δεν αντικατοπτρίζει καμία αλλαγή της περιβάλλουσας κατάστασης (ως εκ τούτου θα πρέπει να θεωρηθεί ότι η εικόνα δεν περιέχει την ετικέτα C). Παρ' όλα αυτά, διαβάζει μετα-δεδομένα που έχουν καθοριστεί για την περιβάλλουσα κατάσταση, όπως φαίνεται από τις μονάδες Get User και Get Area, οι οποίες αντίστοιχα ανακτούν τον τρέχοντα χρήστη και την τρέχουσα περιοχή του μουσείου, που αναπαρίσταται από τη συσχέτιση User2MuseumLocation. Η σελίδα μπορεί να έχει ανακτηθεί μέσω δύο συνδέσμων. Ο σύνδεσμος Link1 δεν εφοδιάζει με παραμέτρους τη σελίδα, ενώ ο σύνδεσμος Link2 εφοδιάζει τη σελίδα με μία παράμετρο που αναπαριστά μία περιοχή, η οποία επιλέγεται από το χρήστη μέσω ενός καταλόγου περιοχών που εμφανίζεται στη σελίδα *List of Areas*. Η λογική της σελίδας, που υπολογίζεται σύμφωνα με τους τυπικούς κανόνες ειδικεύσης (τους οποίους περιγράψαμε προηγουμένως), είναι να παρουσιάσει τις λεπτομέρειες των έργων τέχνης της περιοχής που επέλεξε ο χρήστης, αν η σελίδα έχει προσπελαστεί μέσω του συνδέσμου Link2, ειδικά να δείξει τις λεπτομέρειες των έργων τέχνης της τρέχουσας περιοχής του χρήστη. Ειδικότερα, όταν χρησιμοποιείται ο σύνδεσμος Link2, η ειδικεύση της επιλεγμένης περιοχής του χρήστη υπερισχύει στον υπολογισμό της μονάδας *Artwork Details*, και η σελίδα δεν προσαρμόζει το περιεχόμενό της με βάση την περιοχή όπου βρίσκεται ο χρήστης.



Εικόνα 12 - Σχήμα υπερκειμένου που παρουσιάζει συμπεριφορά προσαρμοστικότητας ή απουσία προσαρμοστικότητας κατά τον χρόνο εκτέλεσης

Στη συνέχεια τίθεται το ερώτημα του τι θα γίνει αν η σελίδα συμπεριφερθεί ως σελίδα με επίγνωση της περιβάλλουσας κατάστασης, όπως ακριβώς αναπαριστάται από την ετικέτα «C» στην εικόνα. Η σελίδα έχει την ακόλουθη συμπεριφορά (ανεξάρτητα από την επιλεγείσα πολιτική προσαρμοστικότητας):

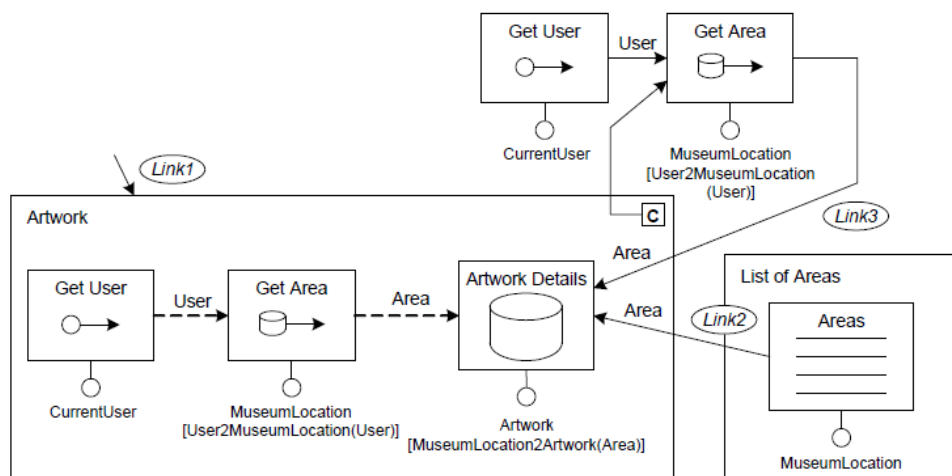
- Αν η σελίδα έχει προσπελαστεί μέσω του συνδέσμου *Link1*, τότε αντανακλώνται οι αλλαγές στην τοποθεσία του χρήστη μέσω της αλλαγής των στοιχείων των έργων τέχνης που εμφανίζονται. Η σελίδα προσαρμόζει σωστά το περιεχόμενο στη τρέχουσα περιοχή, χάρη στις μονάδες *Get User* και *Get Area*, οι οποίες μπορούν να διαβάσουν τα δεδομένα περιβάλλουσας κατάστασης. Σε κάθε ανανέωση που ακολουθεί την πρώτη προσπέλαση των χρηστών, η σελίδα ενημερώνεται όσον αφορά την τρέχουσα περιβάλλουσα κατάσταση.
- Ωστόσο, αν η σελίδα έχει προσπελαστεί μέσω του συνδέσμου *Link2*, τότε υπερισχύει η τιμή που επέλεξε ο χρήστης (η πιο συγκεκριμένη), και η σελίδα δεν προσαρμόζει το περιεχόμενό της. Η τιμή αυτή επικρατεί επίσης και σε κάθε ανανέωση.

Σύμφωνα με τη δεύτερη συμπεριφορά, το περιεχόμενο της σελίδας διατηρείται αμετάβλητο όταν η σελίδα προσπελαύνεται αρχικά από έναν σύνδεσμο που μεταφέρει την επιλογή ενός χρήστη. Αντίθετα ο σχεδιαστής θα μπορεί να επιλέξει μια ενιαία προσαρμοστικότητα ανεξάρτητα από το σύνδεσμο πλοήγησης. Αυτή η ομοιόμορφη συμπεριφορά μπορεί να επιτευχθεί με τον επανασχεδιασμό της σελίδας, έτσι ώστε να καταστήσει την ανάκτηση της τρέχουσας περιοχής, μια ρητή λειτουργία που εξαρτάται από την περιβάλλουσα κατάσταση. Η αποτίμηση αυτή θα πρέπει επομένως να αποτελέσει μέρος ενός νέφους περιβάλλουσας κατάστασης, όπως φαίνεται και στην εικόνα 13. Με τον τρόπο αυτό, δίνεται μεγαλύτερη προτεραιότητα στις ενέργειες προσαρμοστικότητας σε σχέση με τις παλαιότερες επιλογές του χρήστη. Υποθέτοντας μια αναβαλλόμενη πολιτική προσαρμοστικότητας, καταλήγουμε στο ότι η σελίδα υπολογίζεται με τον ακόλουθο τρόπο:

- *Προσπέλαση σελίδας μέσω του συνδέσμου Link1.* Η σελίδα *Artwork* προσπελαύνεται μέσω ενός συνδέσμου που δεν φέρει καμία παράμετρο, και η σελίδα εμφανίζει τις λεπτομέρειες των μνημείων στην περιοχή όπου βρίσκεται ο χρήστης. Δεδομένου ότι πρόκειται για την πρώτη πρόσβαση στη σελίδα, ο σύνδεσμος πλοήγησης του χρήστη δεν ενεργοποιεί ενέργειες προσαρμοστικότητας στο νέφος της περιβάλλουσας

κατάστασης, τα δεδομένα περιβάλλουσας κατάστασης ωστόσο ανακτώνται μέσω της μονάδας *Get Data* εντός της σελίδας.

- **Προσπέλαση σελίδας μέσω του συνδέσμου *Link2*.** Η σελίδα *Artwork* προσπελαύνεται μέσω ενός συνδέσμου που παρέχει την περιοχή που επιλέγεται από το χρήστη. Όπως και στην προηγούμενη περίπτωση, αυτός ο σύνδεσμος δεν ενεργοποιεί οποιαδήποτε ενέργεια στο νέφος της περιβάλλουσας κατάστασης. Η σελίδα δεν προσαρμόζει το περιεχόμενό της, διότι η παράμετρος του συνδέσμου *Link2* υπερισχύει της παραμέτρου *Area* που παράγεται από τη μονάδα *Get Data*.
- **Προσπέλαση σελίδας μέσω ανανέωσης.** Εκτελούνται οι ενέργειες εντός του νέφους περιβάλλουσας κατάστασης. Ως αποτέλεσμα, η τρέχουσα περιοχή του χρήστη περνά στην είσοδο της μονάδας *Artwork Details* μέσω του συνδέσμου *Link3*. Για τη μονάδα δεδομένων που πρέπει να υπολογιστεί, είναι πλέον διαθέσιμες τρεις τιμές της παραμέτρου *Area*: η ταυτότητα αντικειμένου (OID) για την τελευταία περιοχή που επέλεξε ο χρήστης (η οποία παρέχεται μέσω του συνδέσμου *Link2*), η ταυτότητα αντικειμένου (OID) για μία περιοχή που ανακτάται από μία μονάδα *Get Data* που βρίσκεται εντός της σελίδας και η ταυτότητα αντικειμένου (OID) για την περιοχή που ανακτάται από το νέφος της περιβάλλουσας κατάστασης και παρέχεται μέσω του συνδέσμου *Link3*. Σύμφωνα με τους κανόνες ειδίκευσης, οι τιμές που παράγονται στο νέφος περιβάλλουσας κατάστασης υπερισχύουν των τιμών που παράγονται από το χρήστη. Έτσι ο σύνδεσμος *Link3* «αντικαθιστά» τον σύνδεσμο *Link2*, ανεξάρτητα από την αρχική πρόσβαση στη σελίδα.



Εικόνα 13 - Επικάλυψη παρωχημένων παραμέτρων συνδέσμων

Τα τρία παραδείγματα που αναφέρθηκαν σ' αυτή την ενότητα, παρουσιάζουν τρεις πιθανούς τύπους επίγνωσης της περιβάλλουσας κατάστασης: *στατικό* (για συμβατικές σελίδες, όπου γίνεται χρήση των μονάδων *Get* για την ανάκτηση των τρεχουσών τιμών των δεδομένων περιβάλλουσας κατάστασης), *δυναμικό με διακριτή σημασιολογία ανανέωσης που βασίζεται στην αρχική πρόσβαση*, όπως φαίνεται στην εικόνα 12, και *δυναμικό με ενιαία σημασιολογία ανανέωσης*, όπως φαίνεται στην εικόνα 13.

Κεφάλαιο 3 Ολοκληρωμένη μοντελοποίηση Προσαρμοστικών Συστημάτων Λογισμικού με επίγνωση της περιβάλλουσας κατάστασης

Η προσέγγιση που θα παρουσιαστεί στο παρόν κεφάλαιο αναλύεται στο [1]. Η ανάγκη δημιουργίας ενός τέτοιου μοντέλου, έγκειται στην αυξανόμενη ζήτηση για συστήματα λογισμικού που προσαρμόζουν δυναμικά τη συμπεριφορά τους κατά τον χρόνο εκτέλεσης ανάλογα με τις μεταβολές στις προτιμήσεις και απαιτήσεις των χρηστών τους, καθώς και ανάλογα με το λειτουργικό περιβάλλον και την υφιστάμενη υποδομή της περιβάλλουσας κατάστασης και του συστήματος [2, 3].

Η έρευνα που διεξάγεται, μέχρι στιγμής, σε συστήματα που αυτόνομα προσαρμόζονται στις απαιτήσεις των αλλαγών της περιβάλλουσας κατάστασης, χωρίζεται σε δύο κυρίως αντιλήψεις οι οποίες δίνουν αντίστοιχα έμφαση στις εξής δύο μεθοδολογίες: την αυτό-προσαρμοστικότητα [2, 4] και την επίγνωση της περιβάλλουσας κατάστασης [5, 6]. Από τη μια πλευρά, η έρευνα στην επίγνωση της περιβάλλουσας κατάστασης εστιάζει κυρίως στο πώς θα αναπαραστήσει, επεξεργαστεί και οργανώσει τις πληροφορίες της περιβάλλουσας κατάστασης. Ωστόσο, η μεθοδολογία αυτή περιορίζεται μόνο στον τρόπο με τον οποίο ένα σύστημα προσαρμόζεται από μόνο του, ώστε να αντιμετωπίσει τις προβλεπόμενες αλλαγές στις πληροφορίες της περιβάλλουσας κατάστασης. Από την άλλη πλευρά, η έρευνα στην αυτό-προσαρμοστικότητα αφορά κυρίως το πώς θα προσαρμοστεί η δομή και / ή συμπεριφορά του συστήματος, ώστε να αντιμετωπίσει τις αλλαγές της περιβάλλουσας κατάστασης, εστιάζοντας λιγότερο στο πώς η περιβάλλουσα κατάσταση, παρουσιάζεται, επεξεργάζεται και υπόκειται σε διαχείριση, και αυτό αυξάνει την πολυπλοκότητα του συστήματος, καθώς αναμιγνύει τη διαχείριση των λειτουργιών της περιβάλλουσας κατάστασης, μαζί με τη λειτουργική και προσαρμοστική λογική του συστήματος. Επιπροσθέτως, η υλοποίηση των συστημάτων που μελετούν την έμμεση διαχείριση πληροφοριών της περιβάλλουσας κατάστασης, είναι επιρρεπής σε σφάλματα και δύσκολη στην εφαρμογή σε μεγάλης κλίμακας συστημάτων λογισμικού. Ως εκ τούτου, χρειάζεται μια προσέγγιση που να λαμβάνει υπόψη άμεσα τις συσχετίσεις του συστήματος-περιβάλλουσας κατάστασης κατά τον σχεδιασμό του συστήματος, την υλοποίηση και την μετέπειτα εκτέλεση για την αντιμετώπιση των αναμενόμενων αλλαγών της περιβάλλουσας κατάστασης, καθώς το σύστημα είναι σε λειτουργία.

3.1 Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει

3.1.1 Αδυναμίες που προσπαθεί να καλύψει η προσέγγιση

Οι μέχρι τώρα αδυναμίες των υπάρχοντων μοντέλων, που προσπαθεί η προσέγγιση να καλύψει είναι:

- *Η δυσκολία σχεδιασμού του συστήματος* (αφορά κυρίως την σχεδίαση συστημάτων μεγάλης κλίμακας). Τα παραδοσιακά μοντέλα στοιχείων του συστήματος, ορίζουν μόνο άμεσα τις απαιτούμενες και παρεχόμενες λειτουργίες [7, 8]. Επιπροσθέτως, τα μοντέλα που είναι βασισμένα στην κατάσταση [9] ή σε βοηθητικές λειτουργίες [10], είναι δύσκολο να

εφαρμοστούν σε συστήματα μεγάλης κλίμακας, ενώ δυσχέρειες παρουσιάζουν και στην κωδικοποίηση των κανόνων που αναπαριστούν την προσαρμοστική συμπεριφορά του συστήματος.

- *Η πιθανότητα πρόκλησης σφαλμάτων*, τα οποία είναι πολυάριθμα στις περιπτώσεις που η περιβάλλουσα κατάσταση λαμβάνεται υπ' όψιν μόνο έμμεσα κατά την υλοποίηση του συστήματος. Οι περισσότερες από τις πρόσφατες προσεγγίσεις λαμβάνουν υπόψη τη σχέση συστήματος-περιβάλλουσας κατάστασης έμμεσα [9, 11, 12] και όσες προσεγγίσεις λαμβάνουν υπόψη αυτή τη σχέση άμεσα δεν υπολογίζουν όλους τους τύπους των πληροφοριών περιβάλλουσας κατάστασης [13].
- *Η επιβάρυνση της επίβλεψης*. Οι πρόσφατες προσεγγίσεις παρέχουν ένα προσαρμοσμένο μοντέλο περιβάλλουσας κατάστασης, αλλά δεν παρέχουν μια μέθοδο για τον έλεγχο του μοντέλου περιβάλλουσας κατάστασης, αφήνοντας αυτή την εργασία για τον σχεδιαστή του συστήματος [14, 15].
- *Η πολυπλοκότητα του σχεδιασμού και της υλοποίησης του συστήματος*. Οι έως τώρα προταθείσες τεχνικές είτε (α) διαχωρίζουν τη λειτουργικότητα του συστήματος από την λογική προσαρμογής ενώ παράλληλα θεωρούν τη διαχείριση περιβάλλουσας κατάστασης έμμεσα είτε [2] (β) θεωρούν τη διαχείριση της περιβάλλουσας πληροφορίας άμεσα, αλλά εισάγουν άμεσα κώδικα στο λειτουργικό μέρος του συστήματος για την υλοποίηση της λογικής της προσαρμογής [5]. Σε κάθε περίπτωση, υπάρχει ανάγκη για τον διαχωρισμό των τριών στοιχείων ώστε να μειωθεί η πολυπλοκότητα.

3.1.2 Στόχοι της προσέγγισης

Ο στόχος της παρούσας προσέγγισης είναι, η ανάπτυξη ενός συστήματος λογισμικού το οποίο θα μπορεί να ανταπεξέρχεται στις αναμενόμενες αλλαγές της περιβάλλουσας κατάστασης. Για να το πετύχει αυτό, θα πρέπει να έχει μια αναπαράσταση της περιβάλλουσας κατάστασης του, καθώς και την ικανότητα να προσαρμόζεται από μόνο του κατά τον χρόνο εκτέλεσης, ώστε να αντεπεξέλθει στις αλλαγές των πληροφοριών της περιβάλλουσας κατάστασης. Πέραν των προηγούμενων δύο πτυχών, στόχος της προσέγγισης είναι να λαμβάνει υπόψη άμεσα τις συσχετίσεις περιβάλλουσας κατάστασης-συστήματος από την σχεδίαση στην υλοποίηση.

3.1.3 Καινοτομίες της προσέγγισης

Η συνεισφορά της προσέγγισης, συνοψίζεται στα ακόλουθα:

1. Απλοποίηση της σχεδίασης και υλοποίησης του συστήματος μέσω:
 - a. ενός νέου μοντέλου προσαρμοστικού συστήματος που να περιλαμβάνει στοιχεία επίγνωσης της περιβάλλουσας κατάστασης και
 - b. ενός μετά-μοντέλου της εκτέλεσης του λογισμικού, το οποίο κάνει δυνατή την αυτοματοποίηση της υλοποίησης του συστήματος.
2. Η μείωση των πιθανών σφαλμάτων κατά την ανάπτυξη του συστήματος, λαμβάνοντας υπόψη κατά τη σχεδίαση τις σχέσεις συστήματος-περιβάλλουσας κατάστασης.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

3. Αναπαράσταση των σχέσεων συστήματος- περιβάλλουσας κατάστασης κατά τον χρόνο εκτέλεσης, ώστε να καθίσταται δυνατή η προσαρμογή και εξέλιξή τους κατά τον χρόνο εκτέλεσης.
4. Δυνατότητα ελέγχου και εντοπισμού των παραλειπόμενων/λανθασμένων κανόνων προσαρμογής με οπτικό τρόπο.
5. Παροχή ενός προσαρμόσιμου μοντέλου επίγνωσης της περιβάλλουσας κατάστασης για τη μείωση της αναγκαιότητας για επίβλεψη.
6. Μείωση της πολυπλοκότητας του συστήματος, μέσω του διαχωρισμού της διαχείρισης της περιβάλλουσας κατάστασης από το τμήμα του συστήματος που υλοποιεί τις λειτουργικές προδιαγραφές, καθώς και από τη διαχείριση του τελευταίου.

3.2 Βασικές έννοιες της προσέγγισης

Σε αυτή την ενότητα, θα αναλυθούν οι βασικές έννοιες της προσέγγισης. Αυτό θα γίνει σε δύο μέρη. Αρχικά, θα περιγραφεί το μοντέλο CAMP, το οποίο προσδοκά να απλοποιήσει το σχεδιασμό του συστήματος και να μειώσει την πολυπλοκότητα του συστήματος και των λαθών σχεδιασμού, μέσω της αναπαράστασης των σχέσεων συστήματος-περιβάλλουσας κατάστασης άμεσα στο επίπεδο της σχεδίασης. Στη συνέχεια, περιγράφεται το προσαρμοσμένο μετά-μοντέλο επίγνωσης της περιβάλλουσας κατάστασης στο οποίο στηρίζεται η παρούσα προσέγγιση καθώς και το εργαλείο ανάπτυξης (CAST) το οποίο θα χρησιμοποιηθεί για την ανάπτυξη και υλοποίηση της εφαρμογής.

3.2.1 Context-aware Adaptive Systems Component Model (CAMP)

Τα συστήματα λογισμικού με επίγνωση της περιβάλλουσας κατάστασης πρέπει να λαμβάνουν υπόψη τις πληροφορίες περιβάλλουσας κατάστασης που επιδρούν στην λειτουργία τους και να προσαρμόζονται στις αλλαγές περιβάλλουσας κατάστασης, προσθέτοντας/αφαιρώντας/αντικαθιστώντας στοιχεία του συστήματος. Ως εκ τούτου, τα στοιχεία του συστήματος πρέπει να είναι σε θέση, αυτόνομα να διατηρούν αυτές τις δύο προϋποθέσεις άμεσα, ορίζοντας (α) ποιά είναι η πληροφορία περιβάλλουσας κατάστασης που τα επηρεάζει ή χρειάζεται από αυτά για να ολοκληρώσουν τις διεργασίες τους, και (β) ποιές είναι οι πιθανές ενέργειες προσαρμογής που μπορούν να εφαρμοστούν σε αυτά. Επιπλέον, για τη μοντελοποίηση της προσαρμοστικότητας του συστήματος, το στοιχείο που την υλοποιεί πρέπει να οριστεί είτε ως υποχρεωτικό (να χρειάζεται δηλαδή σε κάθε περίπτωση περιβάλλουσας πληροφορίας), είτε ως προαιρετικό (να μπορεί δηλαδή, να προστεθεί ή να αφαιρεθεί βάση της δεδομένης κατάστασης της περιβάλλουσας κατάστασης). Στην εικόνα 14 απεικονίζεται το μοντέλο στοιχείων μας το οποίο λαμβάνει υπόψη τις ανωτέρω υποθέσεις, ενώ στις ακόλουθες παραγράφους τα στοιχεία του περιγράφονται συνοπτικά.

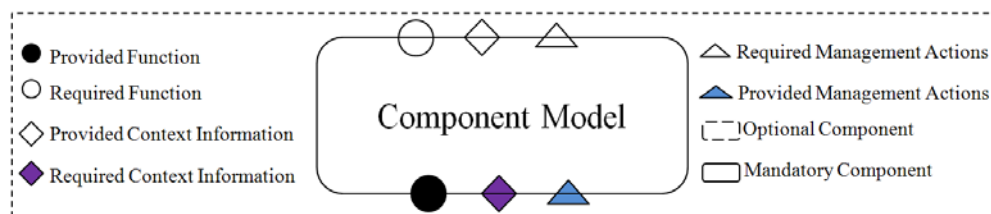
Απαιτούμενες και παρεχόμενες λειτουργίες: Το σύστημα λογισμικού αποτελείται από ένα σύνολο στοιχείων που αφορούν τη λειτουργία του. Τα στοιχεία αυτά είναι συνδεδεμένα μεταξύ τους μέσω των απαιτούμενων και παρεχόμενων *θυρών λειτουργίας*, δηλ. των σημείων όπου δύο στοιχεία συνδέονται αν το ένα χρειάζεται μια λειτουργία που παρέχει το άλλο. Επιπροσθέτως, η χρήση των θυρών αυτών εξυπηρετεί την αναπαράσταση του στοιχείου όπου έχει διαφορετικές εκδοχές

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

(π.χ. ο αλγόριθμος εύρεσης διαδρομής μπορεί να έχει διαφορετικές εκδοχές ανάλογα με την περιβάλλουσα κατάσταση). Στην περίπτωση αυτή, η θύρα του στοιχείου που απαιτεί μία λειτουργικότητα, είναι συνδεδεμένη με πολλές θύρες στοιχείων που την παρέχουν και κατά τη διάρκεια εκτέλεσης του συστήματος μια από αυτές επιλέγεται (πχ η ενέργεια προσαρμογής αντικατάστασης στοιχείου).



Εικόνα 14 - context-aware adaptive systems component model (CAMP)

Απαιτούμενη και παρεχόμενη πληροφορία περιβάλλουσας κατάστασης:

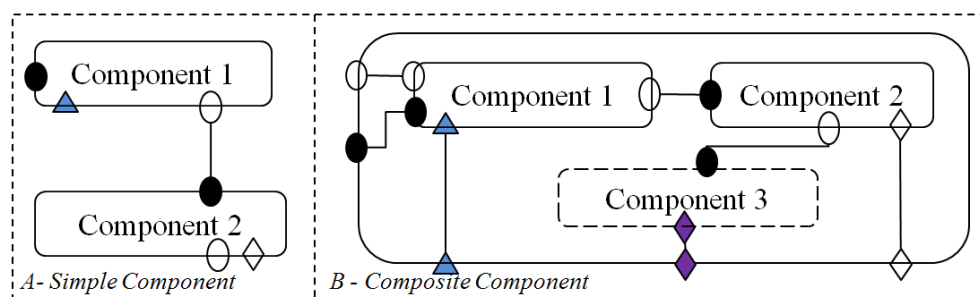
Προκειμένου το σύστημα λογισμικού να συνεχίσει τη λειτουργία του, χρειάζεται τη πληροφορία περιβάλλουσας κατάστασης. Επιπροσθέτως, η πληροφορία περιβάλλουσας κατάστασης μπορεί να προκαλέσει στο σύστημα τη προσαρμογή από μια κατάσταση σε μια άλλη. Ως εκ τούτου, κάθε στοιχείο πρέπει να καθορίζει άμεσα ποιά είναι εκείνη η πληροφορία περιβάλλουσας κατάστασης που χρειάζεται. Αυτή η πληροφορία περιβάλλουσας κατάστασης παρέχεται από άλλα στοιχεία, τα οποία καλούνται *πάροχοι περιβάλλουσας κατάστασης*. Επίσης, η διαχείριση του συστήματος πρέπει να γνωρίζει τη δεδομένη κατάσταση του συστήματος, και κάθε στοιχείο πρέπει να παρέχει τη κατάστασή του, στη διαχείριση του συστήματος.

Απαιτούμενες και παρεχόμενες δράσεις διαχείρισης: Το σύστημα πρέπει να προσαρμοστεί από μια διαμόρφωση σε μια άλλη και στη συνέχεια πρέπει να εκτελέσει μια δέσμη ενεργειών προσαρμογής όπως η πρόσθεση/αφαίρεση/αντικατάσταση ενός στοιχείου (η αντικατάσταση ενός στοιχείου μπορεί να επιτευχθεί με την εισαγωγή ενός νέου στοιχείου ή με την μετατροπή του υπάρχοντος στοιχείου που είναι σε χρήση). Ως εκ τούτου, η περιγραφή του στοιχείου του συστήματος λογισμικού μπορεί να καθορίσει άμεσα ποιές είναι οι ενέργειες προσαρμογής που υποστηρίζει. Επιπλέον, η λογική προσαρμογής του συστήματος χρησιμοποιείται για να καθορίζει τις απαιτούμενες ενέργειες προσαρμογής, και έπειτα τα στοιχεία λογικής προσαρμογής να καθορίζουν άμεσα τις απαιτούμενες ενέργειες προσαρμογής. Έπειτα, οι παρεχόμενες ενέργειες προσαρμογής στο επίπεδο του συστήματος λογισμικού, συνδέονται με τις απαιτούμενες ενέργειες προσαρμογής στο επίπεδο της διαχείρισης του συστήματος.

Προαιρετικά και υποχρεωτικά στοιχεία: Το σύστημα έχει το βασικό επίπεδο, το οποίο στην ουσία είναι το ελάχιστο σύνολο στοιχείων που χρειάζονται από το σύστημα λογισμικού για να ξεκινήσει η διαδικασία εκτέλεσής του. Αυτά τα στοιχεία καλούνται υποχρεωτικά στοιχεία. Επιπροσθέτως, υπάρχει μια δέσμη στοιχείων που μπορεί να προστεθεί και να αφαιρεθεί ανάλογα με τις απαιτήσεις του συστήματος και τις διάφορες μεταβολές στην περιβάλλουσα κατάσταση: αυτά καλούνται προαιρετικά στοιχεία. Σε μεγάλης κλίμακας συστήματα λογισμικού, υπάρχει η ανάγκη για ένα υψηλότερο επίπεδο αφαίρεσης ώστε να απλοποιηθεί ο σχεδιασμός του συστήματος. Έτσι το μοντέλο στοιχείων μας μπορεί να είναι είτε ένα απλό στοιχείο είτε ένα σύνθετο στοιχείο, το οποίο κατά βάση είναι ένα σύνολο στοιχείων που αλληλεπιδρούν μεταξύ τους για να παράγουν μια λειτουργία. Στην εικόνα 15-A, απεικονίζεται ένα σύστημα το οποίο αποτελείται από δύο απλά στοιχεία

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

τα οποία είναι συνδεδεμένα μεταξύ τους από τις απαιτούμενες και παρεχόμενες θύρες. Στην εικόνα 15-B, απεικονίζεται ένα σύνθετο στοιχείο το οποίο αποτελείται από τρία απλά στοιχεία τα οποία αλληλεπιδρούν μεταξύ τους. Οι απαιτούμενες/παρεχόμενες θύρες των εσωτερικών στοιχείων προβάλλονται ως οι απαιτούμενες/παρεχόμενες θύρες του σύνθετου στοιχείου.



Εικόνα 15 - Το μοντέλο στοιχείων, σε απλή μορφή (περίπτωση A) και σε σύνθετη (περίπτωση B)

3.2.2 Context-aware Adaptive Software Systems Meta-Model

Υπάρχουν σημαντικές διαφορές όσον αφορά τον σχεδιασμό προσαρμοστικών συστημάτων λογισμικού με επίγνωση της περιβάλλουσας κατάστασης, από τη σχεδίαση των παραδοσιακών συστημάτων. Οι διαφορές αυτές κυρίως έγκεινται στο γεγονός ότι χρειάζεται να σχεδιάσουμε ένα δυναμικό σύστημα το οποίο έχει τη δυνατότητα μεταβολής κατά τον χρόνο εκτέλεσης. Ο σχεδιασμός των παραδοσιακών συστημάτων, πραγματοποιείται με τον καθορισμό του συστήματος ως ένα σύνολο στατικών στοιχείων. Για παράδειγμα, η δομή ενός συστήματος αποτελείται από ένα σύνολο στοιχείων, καθώς και ένα σύνολο συνδέσμων για την καταγραφή των αλληλεπιδράσεων των στοιχείων. Ωστόσο, για την σχεδίαση προσαρμοστικών συστημάτων με επίγνωση της περιβάλλουσας κατάστασης, υπάρχει η ανάγκη της παρουσίασης του στατικού μέρους του συστήματος μαζί με τις απόψεις (aspects) του συστήματος. Τα εκτελέσιμα μέρη του συστήματος είναι δύο. Το πρώτο μέρος, αφορά τις καταστάσεις του συστήματος, που καθορίζουν τις πιθανές ρυθμίσεις και/ή συμπεριφορές κατά την εκτέλεση του συστήματος και το δεύτερο μέρος, αφορά τις διαδρομές εκτέλεσης του συστήματος που καθορίζουν τους κανόνες, σύμφωνα με το οποίους το σύστημα αλλάζει κάθε φορά κατάσταση. Ως εκ τούτου, το σύστημα χρειάζεται να σχεδιαστεί ως προς τρεις παραμέτρους:

- i) το στατικό μέρος,
- ii) τις καταστάσεις και
- iii) τις διαδρομές εκτέλεσης του συστήματος.

Στα προσαρμοστικά συστήματα με επίγνωση της περιβάλλουσας κατάστασης, χρειάζεται να σχεδιαστεί το περιβάλλον, η λειτουργικότητα και η διαχείριση του συστήματος. Αυτά τα μοντέλα έχουν τα ίδια χαρακτηριστικά με τα μοντέλα εκτέλεσης, όπου το καθένα μπορεί να έχει έναν ορισμό, καταστάσεις κατά τον χρόνο εκτέλεσης, και εκτέλεση. Στην εικόνα 16 απεικονίζεται ένα μετά-μετά-μοντέλο των ανωτέρω τριών απόψεων. Επιπλέον, προστίθεται η υλοποίηση του συστήματος ως ξεχωριστό μέρος από τον ορισμό και την κατάσταση ώστε να επιτευχθεί η προσαρμοστικότητα με τη μεταβολή της υλοποίησης, χωρίς να επηρεαστούν οι αλληλεπιδράσεις των στοιχείων όπου αυτές οι αλληλεπιδράσεις λαμβάνουν χώρα στο

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

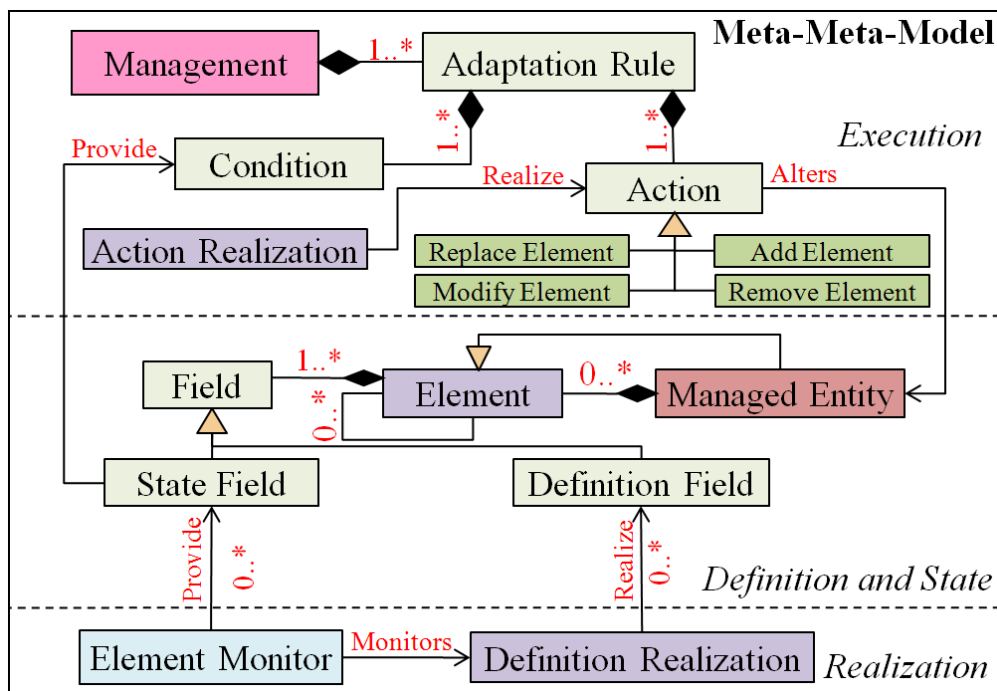
Φούντας Γεώργιος

επίπεδο αναπαράστασης του συστήματος. Το προτεινόμενο μετά-μετά-μοντέλο αποτελείται από:

Διαδρομές εκτέλεσης: Είναι το σύνολο των κανόνων, που καθορίζουν τη μετάπτωση μεταξύ των καταστάσεων για τις οντοτήτων που διαχειρίζεται το σύστημα. Οι κανόνες, βασίζονται σε μια συνθήκη για τη μεταβολή της διαχειριζόμενης οντότητας, μέσω της ενεργοποίησης μιας δέσμης ενεργειών προσαρμογής. Οι ενέργειες προσαρμογής που χρησιμοποιούμε εδώ είναι γενικές και χρησιμοποιούνται για την πρόσθεση/αφαίρεση/αντικατάσταση ή μετατροπή ενός στοιχείου. Ως εκ τούτου μπορούν να εφαρμοστούν σε οποιαδήποτε διαχειριζόμενη οντότητα.

Κατάσταση και ορισμός: Σε αυτό το σημείο, η διαχειριζόμενη οντότητα αναπαριστάται ως ένα σύνολο στοιχείων. Το κάθε στοιχείο, έχει τον ορισμό και την κατάσταση που επιδρά στην κατάσταση του χρόνου εκτέλεσης του στοιχείου.

Υλοποίηση: Αυτό το μέρος χρησιμεύει για να επιτευχθεί η ευελιξία μέσω του διαχωρισμού της κατάστασης ή του ορισμού του στοιχείου από την υλοποίησή τους, και έπειτα κατά το χρόνο εκτέλεσης ο πάροχος κατάστασης ή η εφαρμογή του στοιχείου μπορεί να αλλάξει χωρίς να επηρεάσει τη συνολική λειτουργικότητα του συστήματος.



Εικόνα 16 - Το μετά-μετά-μοντέλο μιας διαχειριζόμενης οντότητας

3.2.3 Διαχειριζόμενες οντότητες προσαρμοστικών συστημάτων με επίγνωση της περιβάλλουσας κατάστασης

Τα προσαρμοστικά συστήματα με επίγνωση της περιβάλλουσας κατάστασης έχουν τρεις διαχειριζόμενες οντότητες, οι οποίες αναλύονται ακολούθως:

Το μοντέλο περιβάλλουσας κατάστασης, όπου η δομή του μοντέλου επίγνωσης της περιβάλλουσας κατάστασης έχει τη δυνατότητα μεταβολής κατά τον χρόνο εκτέλεσης, έτσι ώστε να μειώσει την απαίτηση για επίβλεψη ή για να

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

ενσωματώσει ένα νέο στοιχείο περιβάλλουσας κατάστασης κατά τη διάρκεια εκτέλεσης του συστήματος.

Το λειτουργικό σύστημα (functional system), το οποίο ανταποκρινόμενο στις αλλαγές της περιβάλλουσας κατάστασης χρειάζεται να προσαρμόσει τη δομή του, ώστε να ανταπεξέλθει στις αλλαγές του περιβάλλοντος.

Η διαχείριση του συστήματος, όπου μια νέα προσαρμοσμένη συμπεριφορά προστίθεται κατά τη διάρκεια εκτέλεσης, για να ανταπεξέλθει στις προβλεπόμενες αλλαγές της περιβάλλουσας κατάστασης ή να προσαρμόσει τη λογική, ώστε να βελτιώσει την απόδοση της λήψεως απόφασης.

3.2.4 Το μετά-μοντέλο προσαρμοστικού συστήματος με επίγνωση της περιβάλλουσας κατάστασης

Έπειτα από το μετά-μετά-μοντέλο που παρουσιάστηκε στην εικόνα 16, ένα μετά-μοντέλο για προσαρμοστικά συστήματα με επίγνωση της περιβάλλουσας κατάστασης, το οποίο εστιάζει στο μοντέλο περιβάλλουσας κατάστασης και το σύστημα λογισμικού ως διαχειριζόμενες οντότητες απεικονίζεται στην εικόνα 17. Αυτό το μετα-μοντέλο περιλαμβάνει τις εξής απόψεις:

Διαχείριση αλλαγών: Είναι υπεύθυνη για την επιλογή των απαιτούμενων ενεργειών προσαρμογής, που εφαρμόζονται μέσω των ενεργειών υλοποίησης. Ακολουθεί το κομμάτι της εκτέλεσης που απεικονίζεται στην εικόνα 16. Χάριν απλότητας, χρησιμοποιούνται εδώ οι ίδιες ενέργειες του μετά-μετά-μοντέλου, οι οποίες είναι γενικές και μπορούν να χρησιμοποιηθούν με το μοντέλο περιβάλλουσας κατάστασης και την προσαρμογή του λειτουργικού συστήματος (functional system). Επιπροσθέτως, μπορούν να επεκταθούν ώστε να συμπεριλάβουν συγκεκριμένες ενέργειες όπως, η πρόσθεση οντότητας περιβάλλουσας κατάστασης, η αφαίρεση οντότητας περιβάλλουσας κατάστασης, κ.τ.λ. Ακόμη, οι κανόνες προσαρμογής συλλαμβάνουν τη σχέση συστήματος-περιβάλλουσας κατάστασης από τη σκοπιά της διαχείρισης, όπου οι αλλαγές της περιβάλλουσας κατάστασης και/ή του συστήματος χρησιμοποιούνται για να ενεργοποιήσουν/προκαλέσουν την αποτίμηση της λογικής προσαρμογής και έπειτα οι κατάλληλοι κανόνες πυροδοτούνται, ώστε ακολούθως μια δέσμη ενεργειών προσαρμογής να καθοριστεί για να εφαρμοστεί στο λειτουργικό σύστημα ή/και στο μοντέλο επίγνωσης της περιβάλλουσας κατάστασης.

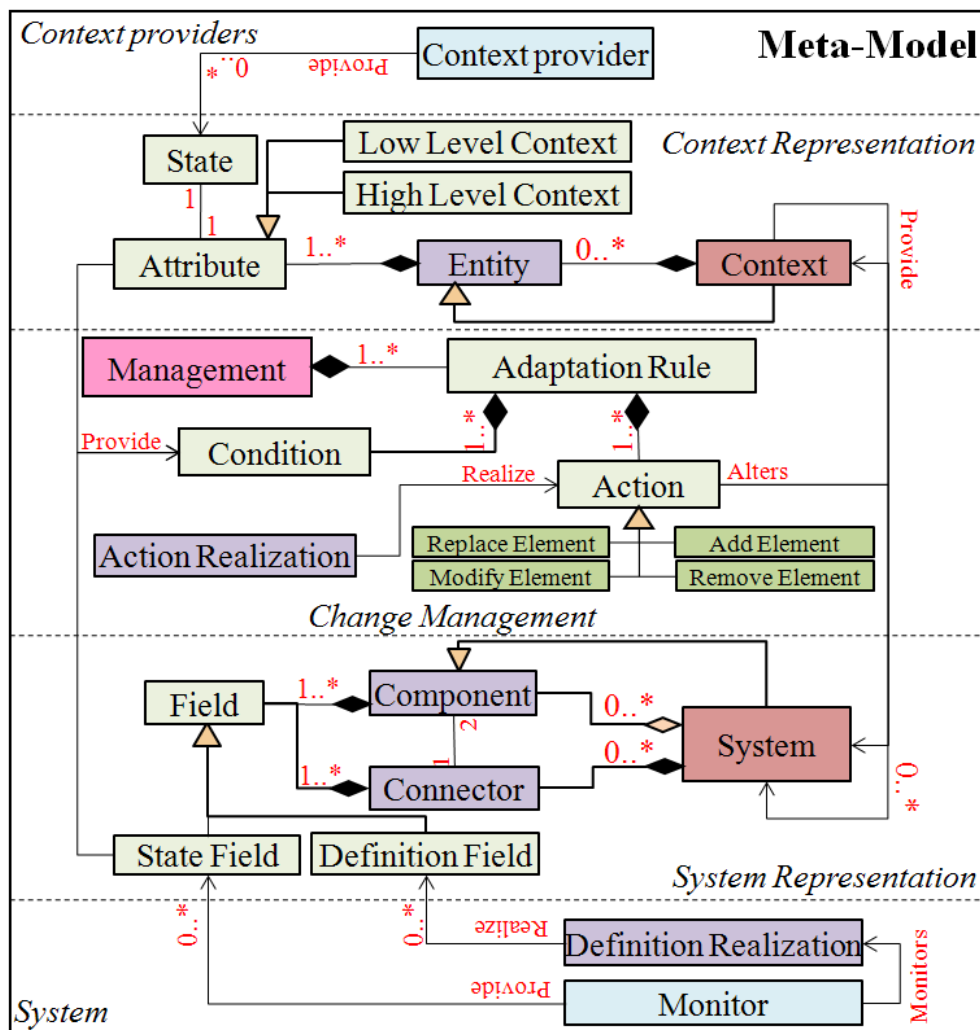
Αναπαράσταση περιβάλλουσας κατάστασης: Η περιβάλλουσα κατάσταση είναι οι περιβάλλουσες οντότητες που είναι σχετικές με το σύστημα και χρειάζονται είτε (α) για τη συνέχιση/ολοκλήρωση της εφαρμογής, είτε (β) για την ενεργοποίηση της προσαρμογής του συστήματος. Ορίζουμε το μοντέλο περιβάλλουσας κατάστασης ως ένα σύνολο οντοτήτων. Η κάθε οντότητα μπορεί να έχει πολλές ιδιότητες. Οι ιδιότητες μπορεί να περιβάλλουσα κατάσταση χαμηλού επιπέδου ή υψηλού επιπέδου. Το σύστημα πρέπει να γνωρίζει την κατάσταση του περιβάλλοντος και καθορίζεται επίσης η κατάσταση της κάθε ιδιότητας-μεταβλητής, η οποία παρέχεται μέσω ενός ή περισσότερων παρόχων περιβάλλουσας κατάστασης. Επιπροσθέτως, η σύνθεση της περιβάλλουσας κατάστασης είναι από μόνη της μια οντότητα και έπειτα η περιβάλλουσα κατάσταση μπορεί να διασπαστεί σε επίπεδα ιεράρχησης ώστε αφενός να έχει καλύτερη οργάνωση και αφετέρου να είναι εύκολα προσβάσιμη. Έτσι, η σύνθεση της περιβάλλουσας κατάστασης κάνει δυνατή την αλλαγή στο μοντέλο περιβάλλουσας κατάστασης με το να προσθέτει/αφαιρεί/αντικαθιστά μια οντότητα περιβάλλουσας κατάστασης και/ή μια ιδιότητα-μεταβλητή.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Πάροχοι περιβάλλουσας κατάστασης: Η πληροφορία περιβάλλουσας κατάστασης παρέχεται με τη χρήση των παρόχων περιβάλλουσας κατάστασης, οι οποίοι μπορεί να είναι υλικοί αισθητήρες ή/και άυλοι αισθητήρες. Το όφελος του διαχωρισμού της αναπαράστασης περιβάλλουσας κατάστασης από τους παρόχους περιβάλλουσας κατάστασης είναι η δυνατότητα της επιλογής κατά τον χρόνο εκτέλεσης των παρόχων περιβάλλουσας κατάστασης βάσει του κατά περίπτωση απαιτούμενου επιπέδου ποιότητας της περιβάλλουσας κατάστασης, της προσβασιμότητας των παρόχων κ.λπ.

Αναπαράσταση συστήματος: Η διαχείριση των αλλαγών, είναι αναγκαίο να γνωρίζει την τρέχουσα κατάσταση του συστήματος, ώστε να επιλέξει τις κατάλληλες ενέργειες προσαρμογής (εφόσον χρειάζεται), και επίσης πρέπει να συντηρείται η κατάσταση λειτουργίας του συστήματος. Επιπροσθέτως, το λειτουργικό σύστημα (functional system) αποτελείται από ένα σύνολο στοιχείων που επιτρέπουν τη λειτουργικότητα του πυρήνα του συστήματος, και από ένα σύνολο συνδέσμων, όπου αναπαριστούν τις αλληλεπιδράσεις μεταξύ των στοιχείων του συστήματος. Έτσι κάθε στοιχείο στο μοντέλο του λειτουργικού μας συστήματος, περιλαμβάνει τα πεδία κατάστασης και ορισμού, που ακολουθούν το μετά-μετά-μοντέλο που παρουσιάστηκε στην εικόνα 16. Το πεδίο κατάστασης χρησιμεύει στο να παρέχει την κατάσταση του εκτελούμενου λειτουργικού συστήματος και το πεδίο ορισμού χρησιμεύει στο να παράγει την υλοποίηση του συστήματος (αυτό είναι διαφορετικό από το μοντέλο συστήματος όπου δεν χρειαζόμαστε μόνο να γνωρίζουμε την κατάσταση του συστήματος, αλλά πρέπει να παραχθεί και η υλοποίησή του). Παρομοίως με το μοντέλο περιβάλλουσας κατάστασης, το σύστημα, είναι ένα στοιχείο από μόνο του ικανό να καθορίσει την ιεραρχία σαν πολλαπλά επίπεδα αφαίρεσης και επίσης, μπορεί να προσαρμοστεί προσθέτοντας/αφαιρώντας/αντικαθιστώντας τα στοιχεία και τις συνδέσεις τους.

Υλοποίηση συστήματος: Η αναπαράσταση του συστήματος ουσιαστικά είναι ένα πλαίσιο που θα συμπληρωθεί με τη δομή του συστήματος, και παράλληλα υπάρχει ένα σύνολο εποπτών (monitors) που χρησιμοποιούνται για να παρέχουν την κατάσταση του συστήματος και που αντιστοιχούν στους παρόχους περιβάλλουσας κατάστασης στο κομμάτι του μοντέλου περιβάλλουσας κατάστασης. Επιπροσθέτως, τα στοιχεία του συστήματος παρέχονται μέσω της υλοποίησης διαφορετικών στοιχείων και η κάθε μια από αυτές τις εκδοχές μπορεί να χρησιμοποιηθεί κατά το χρόνο εκτέλεσης, βάσει της διαθεσιμότητας τους ή των απαιτούμενων μη-λειτουργικών απαιτήσεων του συστήματος (δηλ. των απαιτήσεων προσαρμοστικότητας).



Εικόνα 17 - Το Μετά-Μοντέλο για προσαρμοστικά συστήματα λογισμικού με επίγνωση της περιβάλλουσας κατάστασης

3.3 Μεθοδολογία ανάπτυξης εφαρμογών σύμφωνα με την προσέγγιση

Σε αυτή την ενότητα αναλύεται η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη της εφαρμογής του ανωτέρω μετά-μοντέλου. Η παρουσίαση θα γίνει μέσω του εργαλείου *Context-aware Adaptive Systems Development Tool (CAST)*, το οποίο μπορεί να χρησιμοποιηθεί για την ανάπτυξη προσαρμοστικών συστημάτων λογισμικού με επίγνωση της περιβάλλουσας κατάστασης. Ακολουθώς παρουσιάζονται τα κύρια μέρη του εργαλείου ανάπτυξης.

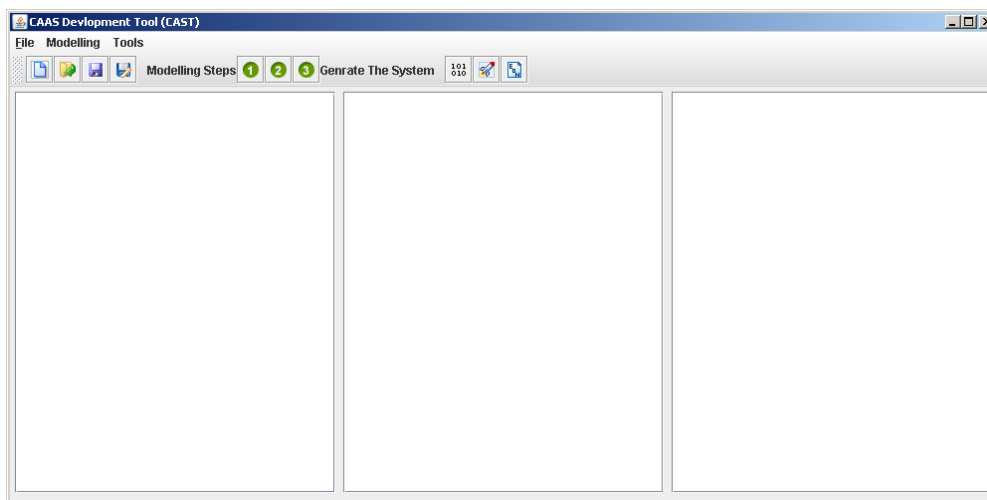
3.3.1 Η κύρια οθόνη του CAST

Η κύρια οθόνη του εργαλείου απεικονίζεται στην εικόνα 18. Το εργαλείο έχει τρία μενού και μια γραμμή εργαλείων, όπου κάθε αντικείμενο του μενού έχει το αντίστοιχο κουμπί στη γραμμή εργαλείων.

1. Το μενού "Αρχείο" περιλαμβάνει τις λειτουργίες για τη δημιουργία ενός νέου έργου, το άνοιγμα ενός υπάρχοντος έργου, την αποθήκευση του υπό

επεξεργασία έργου και την αποθήκευση του υπό επεξεργασία έργου με νέο όνομα.

2. Το μενού "Σχεδίαση" περιλαμβάνει τρεις λειτουργίες που αφορούν, τον σχεδιασμό της περιβάλλουσας κατάστασης (βήμα 1), του συστήματος (βήμα 2), και της διαχείρισης του συστήματος/περιβάλλουσας κατάστασης (βήμα 3). Αυτή τη στιγμή δεν παρέχεται το γραφικό περιβάλλον χρήστη (GUI) ώστε να δώσει τη δυνατότητα στον χρήστη να σχεδιάσει το σύστημα, τη περιβάλλουσα κατάσταση και τις σχέσεις συστήματος- περιβάλλουσας κατάστασης μέσα από το εργαλείο, αλλά τα μοντέλα φορτώνονται από αρχεία XML όπου περιγράφουν το γραφικό μοντέλο το οποίο στηρίζεται πάνω στο παρόν μοντέλο συνιστωσών και το μετά-μοντέλο.
3. Το μενού "Εργαλεία" παρέχει ένα σύνολο λειτουργιών στον σχεδιαστή όπως, η παραγωγή της υλοποίησης του συστήματος, η εκτέλεση του ελέγχου προσαρμοστικής συμπεριφοράς του συστήματος ώστε να αξιολογηθεί η προσαρμοστική συμπεριφορά του συστήματος οπτικά, και ο μετασχηματισμός των κανόνων προσαρμογής σε ένα τυπικό μοντέλο (π.χ. Petri-Nets) ώστε να ενεργοποιήσει σε σχέση με τους κανόνες προσαρμογής (α) την επικύρωση μέσω της εκτέλεσης των Petri-Nets ή (β) τις επαληθεύσεις, ελέγχοντας ένα σύνολο ιδιοτήτων πάνω στη συμπεριφορά του μοντέλου.



Εικόνα 18 - Κύρια οθόνη του εργαλείου προσαρμοστικών συστημάτων επίγνωσης της περιβάλλουσας κατάστασης

3.3.2 Οι κλάσεις Java της εφαρμογής

Στο σημείο αυτό παρουσιάζονται, οι κλάσεις της JAVA που χρησιμοποιούνται για την εφαρμογή:

CAAS: Αυτή η κλάση χρησιμοποιείται ώστε να παράγει την κύρια οθόνη του εργαλείου που απεικονίζεται στην εικόνα 18 και να συνδέσει τις ενέργειες που έχουν επιλεγεί από τον σχεδιαστή με τις αντίστοιχες κλάσεις (πχ το κουμπί που παράγει την υλοποίηση της εφαρμογής με την κλάση GenerateImp).

CASystem: Χρησιμοποιείται ως πλαίσιο που θα γεμίσει με την περιγραφή του λειτουργικού συστήματος. Περιγράφει το σύστημα ως ένα σύνολο συνιστωσών (Component class), και αυτά τα στοιχεία είναι συνδεδεμένα με τις παρεχόμενες και απαιτούμενες θύρες. Επιπροσθέτως, κάθε στοιχείο έχει διαφορετική υλοποίηση σε διαφορετική περιβάλλουσα κατάσταση (*Component implementation class*). Κάθε υλοποίηση στοιχείου έχει έναν ορισμό του τι είναι η απαιτούμενη περιβάλλουσα κατάσταση, η οποία χρειάζεται από το στοιχείο για να συνεχίσει τη λειτουργία του (*Required context class*). Η προσαρμογή του συστήματος δεν εφαρμόζεται μόνο για να αντιμετωπίσει τις αλλαγές της περιβάλλουσας κατάστασης, αλλά επίσης για να αντιμετωπίσει τις αλλαγές στην κατάσταση του συστήματος. Κάθε υλοποίηση στοιχείου έχει τη δικιά της παρεχόμενη πληροφορία περιβάλλουσας κατάστασης (*Provided context class*).

Syscontext: Η περιβάλλουσα κατάσταση του συστήματος περιγράφεται από αυτή τη κλάση, όπου αποτελείται από ένα σύνολο οντοτήτων περιβάλλουσας κατάστασης (*Entity class*). Κάθε οντότητα περιβάλλουσας κατάστασης έχει ένα σύνολο ιδιοτήτων περιβάλλουσας κατάστασης (*Attribute class*). Επιπροσθέτως, κάθε ιδιότητα έχει έναν ή περισσότερους παρόχους περιβάλλουσας κατάστασης (*Context provider class*).

Management: Αυτή η κλάση αντιστοιχεί στην περιγραφή της προσαρμοστικής συμπεριφοράς του συστήματος. Αποτελείται από ένα σύνολο κανόνων προσαρμογής (*Adaptation Rule class*), όπου κάθε κανόνας περιλαμβάνει μια ή περισσότερες συνθήκες (*Condition class*) και μια ή περισσότερες ενέργειες (*Action class*).

Readcontext: Αυτή η κλάση χρησιμοποιείται για να αναγνώσει την περιγραφή XML του στοιχείου, και προσδιορίζει ποιές είναι οι οντότητες στοιχείου, οι ιδιότητές τους και οι πάροχοι ιδιοτήτων στοιχείου.

Readsystem: Παρόμοια με τη κλάση Readcontext, αυτή η κλάση χρησιμοποιείται για να αναλύσει το αρχείο XML του συστήματος και να καθορίσει ποιά είναι τα στοιχεία του συστήματος και οι σχέσεις μεταξύ τους και με τη περιβάλλουσα κατάσταση.

Readmanagement: Αυτή η κλάση έχει την ίδια λειτουργικότητα με τις δύο προηγούμενες, αλλά διαβάζει το αρχείο XML της διαχείρισης συστήματος (π.χ. τους κανόνες προσαρμογής του συστήματος).

Genclass: Για τη δημιουργία της αντίστοιχης εφαρμογής βάσει των σχεδιασμένων μοντέλων, αυτή η κλάση δέχεται την περιγραφή ενός στοιχείου του μοντέλου ως είσοδο και παράγει τον αντίστοιχο κώδικα σε JAVA ως έξοδο. Έχει διάφορες υποκλάσεις που χρησιμοποιούνται όπως, η κλάση **Field** που χρησιμοποιείται για να ορίσει τις ιδιότητες της κλάσης, η κλάση **Method**, η οποία χρησιμοποιείται για να διατηρήσει την περιγραφή των μεθόδων κλάσης, η κλάση **Import class**, η οποία περιγράφει την βιβλιοθήκη που πρέπει να εισαχθεί, και η κλάση **Constructor** που αναπαριστά τις μεθόδους κατασκευής των κλάσεων.

GenerateImp: Αυτή η κλάση είναι η ραχοκοκαλιά του εργαλείου. Είναι υπεύθυνη για την παραγωγή του κώδικα που αντιστοιχεί στην περιγραφή XML των μοντέλων. Επίσης, χρησιμοποιείται για τη δημιουργία ενός στιγμιότυπου του συστήματος με απλό GUI για τον έλεγχο της προσαρμοστικής συμπεριφοράς του συστήματος. Αυτή η κλάση έχει τις ακόλουθες λειτουργίες:

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

- **Παράγει τον κώδικα για την περιβάλλουσα κατάσταση:** Χρησιμοποιείται για να παράγει τον κώδικα που αναπαριστά τη περιβάλλουσα κατάσταση του συστήματος. Το μέρος που πρέπει να ολοκληρωθεί από τον σχεδιαστή είναι η λειτουργία που βρίσκεται μέσα στον κώδικα του παρόχου περιβάλλουσας κατάστασης, που καλείται “Provide_ Context Attribute Name”. Αυτή η λειτουργία συνδέει την παραγόμενη εφαρμογή μας με το εξωτερικό περιβάλλον μέσω του παρόχου περιβάλλουσας κατάστασης (πχ υλικοί ή άυλοι αισθητήρες).
- **Παράγει τον κώδικα του συστήματος:** Ο κώδικας που παρέχει τη λειτουργικότητα του συστήματος παράγεται χρησιμοποιώντας αυτή τη μέθοδο. Επιπροσθέτως, οι σχέσεις του συστήματος λογισμικού με την περιβάλλουσα κατάσταση, εκτελούνται βάσει της απαιτούμενης πληροφορίας της περιβάλλουσα κατάστασης στην περιγραφή της υλοποίησης του στοιχείου, και με τα άλλα στοιχεία του συστήματος βάσει των απαιτούμενων και παρεχόμενων θυρών λειτουργίας. Το μέρος που χρειάζεται να ολοκληρωθεί από τον προγραμματιστή είναι κάποιες λειτουργίες που καλούνται “Provides_ Provided Function Name” και “Provides_ State”, οι οποίες είναι μέσα στην κλάση υλοποίησης στοιχείου.
- **Παράγει τον κώδικα διαχείρισης:** Η προσαρμοστική συμπεριφορά του συστήματος παράγεται σαν ένα σύνολο κανόνων που καθοδηγούν την προσαρμοστική συμπεριφορά του συστήματος. Η προσαρμογή εφαρμόζεται βάσει των αλλαγών της κατάστασης του περιβάλλοντος και της κατάστασης του συστήματος, και στη συνέχεια εφαρμόζονται οι σχέσεις ανάμεσα στην διαχείριση του συστήματος, τη περιβάλλουσα κατάσταση του περιβάλλοντος, και τη περιβάλλουσα κατάσταση του συστήματος λογισμικού. Σε αυτή τη κλάση, οι προγραμματιστές δεν χρειάζονται να προσθέσουν πρόσθετο κώδικα.
- **Παράγει κώδικα για τον έλεγχο της προσαρμοστικής συμπεριφοράς του συστήματος,** επιτρέποντας να διαφανεί το πώς δημιουργούνται στιγμιότυπα της παραγόμενης υλοποίησης και πώς οι αλλαγές περιβάλλουσας κατάστασης επηρεάζουν το σύστημα λογισμικού. Αυτή η κλάση παράγεται και έχει τις ακόλουθες λειτουργίες:
 - **Αρχικοποίηση του συστήματος:** Δημιουργείται ένα στιγμιότυπο της εφαρμογής και στη συνέχεια εφαρμόζονται οι συνδέσεις μεταξύ της περιβάλλουσας κατάστασης, του λειτουργικού συστήματος, και της διαχείρισης συστήματος.
 - **Παραγωγή ενός βασικού GUI για τον έλεγχο:** Ένα απλό GUI παράγεται αυτόματα, το οποίο παρέχει μία οπτική αναπαράσταση του συστήματος και των στοιχείων περιβάλλουσας κατάστασης. Αυτό το GUI δίνει τη δυνατότητα του ελέγχου της προσαρμοστικής συμπεριφοράς μέσω της επιλογής ρυθμίσεων μιας περιβάλλουσας κατάστασης και στη συνέχεια πατώντας το κουμπί “προσαρμογή συστήματος” μας δίνει τη δυνατότητα να δούμε την επίδραση των αλλαγών περιβάλλουσας κατάστασης στο λειτουργικό σύστημα βάσει των κανόνων που καθορίστηκαν στη διαχείριση συστήματος.

- **Προσαρμογή συστήματος:** Είναι υπεύθυνο να συλλέξει τις ρυθμίσεις περιβάλλουσας κατάστασης από το παραγόμενο GUI και να τις αποθηκεύσει σε στιγμιότυπο της περιβάλλουσας κατάστασης. Ακολούθως, το στιγμιότυπο της διαχείρισης του συστήματος καλείται για να ορίσει τις απαιτούμενες ενέργειες και κατόπιν οι υφιστάμενες ρυθμίσεις του συστήματος οπτικοποιούνται μέσω του GUI για να δείξουν τις επιδράσεις των ενεργειών προσαρμογής στο λειτουργικό σύστημα.

3.3.3 Οι είσοδοι και έξοδοι του εργαλείου

Το εργαλείο απαιτεί τρία αρχεία XML ως είσοδο που περιγράφουν την περιβάλλουσα κατάσταση, το λειτουργικό σύστημα και τη διαχείριση συστήματος. Τα αρχεία αυτά, αντιστοιχούν στο σύστημά μας και στο ολοκληρωμένο μοντέλο περιβάλλουσας κατάστασης του. Η έξοδος του εργαλείου είναι ο κώδικας Java που αντιστοιχεί στην περιγραφή των μοντέλων.

3.3.3.1 Η XML περιγραφή του μοντέλου περιβάλλουσας κατάστασης

Το μοντέλο περιβάλλουσας κατάστασης αποτελείται από ένα σύνολο οντοτήτων περιβάλλουσας κατάστασης. Κάθε οντότητα περιβάλλουσας κατάστασης έχει ένα σύνολο ιδιοτήτων που μπορεί να είναι οντότητες περιβάλλουσας κατάστασης υψηλού ή χαμηλού επιπέδου. Τέλος, κάθε ιδιότητα οντότητας περιβάλλουσας κατάστασης έχει όνομα, τύπο που περιγράφει τον τύπο της πληροφορίας περιβάλλουσας κατάστασης όπως ένα αλφαριθμητικό ή ένας αριθμός, και τέλος πάροχο/ους περιβάλλουσας κατάστασης. Η εικόνα 19 δείχνει το σχήμα XML του αρχείου περιβάλλουσας κατάστασης.

```
<xsd:complexType name="Context">
  <xsd:sequence>
    <xsd:element name="Entity" type="Entity Info" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Entity Info">
  <xsd:sequence>
    <xsd:element name=" EName" type="xsd:string"/>
    <xsd:element name="Attributes" type="Attribute Des" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Attribute Des">
  <xsd:sequence>
    <xsd:element name=" AName" type="xsd:string"/>
    <xsd:element name="Type" type="xsd:string"/>
    <xsd:element name="Providers" type="CP Info"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CP Info">
  <xsd:sequence>
    <xsd:element name="Provider" type="xsd:string" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
```

Εικόνα 19 - Η διάταξη XML του μοντέλου περιβάλλουσας κατάστασης

3.3.3.2 Η περιγραφή XML του συστήματος

Το μοντέλο του συστήματος αντιπροσωπεύει τη λειτουργικότητα του πυρήνα του συστήματος. Αυτό το μοντέλο αποτελείται από ένα σύνολο στοιχείων, όπου κάθε στοιχείο έχει (α) τις απαιτούμενες λειτουργίες που αντιπροσωπεύουν τις λειτουργίες

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

που χρειάζονται οι συνιστώσες από άλλες συνιστώσες για επιτελεστεί ο σκοπός του, (β) τις παρεχόμενες λειτουργίες που αντιπροσωπεύουν τις λειτουργίες που παρέχονται από αυτή τη συνιστώσα, (γ) τις υλοποιήσεις συνιστωσών του στοιχείου που αντιπροσωπεύουν τις διαφορετικές υλοποιήσεις του στοιχείου. Κάθε υλοποίηση έχει (γ.1) την απαιτούμενη περιβάλλουσα πληροφορία, που αντιπροσωπεύει την πληροφορία περιβάλλοντος που χρειάζεται η συγκεκριμένη συνιστώσα, (γ.2) την παρεχόμενη περιβάλλουσα πληροφορία, που αντιπροσωπεύει την κατάσταση του στοιχείου που απαιτείται από τη διαχείριση του συστήματος. Η εικόνα 20 δείχνει σε γενικές γραμμές τη περιγραφή των στοιχείων του συστήματος.

```

<xsd:complexType name="System">
  <xsd:sequence>
    <xsd:element name="Component" type="Component Des" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Component Des">
  <xsd:sequence>
    <xsd:element name="CName" type="xsd:string"/>
    <xsd:element name="ReqFunction" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ProFunction" type="xsd:string" minOccurs="1"/>
    <xsd:element name="Complmp" type="Complmp Des" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Component Implementation Des">
  <xsd:sequence>
    <xsd:element name="CMName" type="xsd:string"/>
    <xsd:element name="ReqContext" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ProContext" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

```

Εικόνα 20 - Η διάταξη XML του μοντέλου περιβάλλουσας κατάστασης

3.3.3.3 Η περιγραφή XML της διαχείρισης του συστήματος

Η διαχείριση του συστήματος αποτελείται από ένα σύνολο κανόνων που περιγράφουν τη προσαρμοστική συμπεριφορά του συστήματος. Οι κανόνες αποτελούνται από συνθήκες, οι οποίες αφορούν την κατάσταση του συστήματος ή/και του περιβάλλοντος (πχ $X=1$ και $Y=2$). Η συνθήκη του κανόνα δομείται σύμφωνα με τα εξής:

- την ιδιότητα-μεταβλητή (πχ X και Y) που αφορά η συνθήκη (π.χ. μεταβλητή του συστήματος ή στοιχείο περιβάλλουσας κατάστασης),
- έναν τελεστή σύγκρισης (π.χ. ίσον, μεγαλύτερο από, μικρότερο από, κ.τ.λ.),
- την τιμή με την οποία συγκρίνεται η μεταβλητή (πχ. 1 και 2) και
- προαιρετικά, κάποιο λογικό τελεστή που συνδέει πολλές συνθήκες.

Επίσης οι κανόνες περιέχουν ενέργειες, που καθορίζουν την αντίδραση του συστήματος σε συγκεκριμένη κατάσταση και κάθε ενέργεια έχει ένα τύπο και ένα στοιχείο που εφαρμόζεται σε αυτή. Η εικόνα 21 δείχνει πως δομείται το XML αρχείο που περιγράφει τη διαχείριση του συστήματος.

```

<xsd:complexType name="Management">
  <xsd:sequence>
    <xsd:element name="AdaptationRule" type="AdaptationRule Des" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AdaptationRule Des">
  <xsd:sequence>
    <xsd:element name="Condition" type="Condition Des" minOccurs="1"/>
    <xsd:element name="Action" type="Action Des" minOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Condition Des">
  <xsd:sequence>
    <xsd:element name="Att" type="xsd:string"/>
    <xsd:element name="Aop" type="xsd:string"/>
    <xsd:element name="Val" type="xsd:string"/>
    <xsd:element name="Lop" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Action Des">
  <xsd:sequence>
    <xsd:element name="Type" type="xsd:string"/>
    <xsd:element name="cname" type="xsd:string"/>
    <xsd:element name="ciname" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

```

Εικόνα 21 - Η διάταξη του μοντέλου διαχείρισης συστήματος

3.3.3.4 Οι έξοδοι του εργαλείου

Στην εικόνα 22 απεικονίζεται η έξοδος του εργαλείου, όπου φαίνονται οι αντίστοιχες κλάσεις του στοιχείου (22-1), του συστήματος (22-2), της διαχείρισης (22-3) και τέλος η κλάση που θα χρησιμοποιηθεί για να ελέγξει τη προσαρμοστική συμπεριφορά του συστήματος (22-4).

Preferences.java	1 KB	Java Source File	30/09/2010 8:57 AM	IMP1_ProvidedContext_ComponentState.java	1 KB	Java Source File	30/09/2010 8:57 AM
Preferences_Route.java	1 KB	Java Source File	30/09/2010 8:57 AM	IMP1_RequiredContext_BulkR.java	1 KB	Java Source File	30/09/2010 8:57 AM
Preferences_Route_PCIA.java	1 KB	Java Source File	30/09/2010 8:57 AM	IMP1_RequiredContext_Cong.java	1 KB	Java Source File	30/09/2010 8:57 AM
RoutePlanning_Context.java	1 KB	Java Source File	30/09/2010 8:57 AM	IMP1_RequiredContext_Route.java	1 KB	Java Source File	30/09/2010 8:57 AM
Traffic.java	1 KB	Java Source File	30/09/2010 8:57 AM	IMP2_ProvidedContext_ComponentState.java	1 KB	Java Source File	30/09/2010 8:57 AM
Traffic_BulkR.java	2 KB	Java Source File	30/09/2010 8:57 AM	IMP2_RequiredContext_BulkR.java	1 KB	Java Source File	30/09/2010 8:57 AM
Traffic_BulkR_RSU.java	1 KB	Java Source File	30/09/2010 8:57 AM	IMP2_RequiredContext_Cong.java	1 KB	Java Source File	30/09/2010 8:57 AM
Traffic_BulkR_TSP.java	1 KB	Java Source File	30/09/2010 8:57 AM	RoutePlanning_Implementation_IMP1.java	3 KB	Java Source File	30/09/2010 8:57 AM
Traffic_Cong.java	2 KB	Java Source File	30/09/2010 8:57 AM	RoutePlanning_Implementation_IMP2.java	1 KB	Java Source File	30/09/2010 8:57 AM
Traffic_Cong_RSU.java	1 KB	Java Source File	30/09/2010 8:57 AM	RoutePlanning_Implementation_IMP3.java	2 KB	Java Source File	30/09/2010 8:57 AM
Traffic_Cong_TSP.java	1 KB	Java Source File	30/09/2010 8:57 AM	RoutePlanning_Requires_I1.java	1 KB	Java Source File	30/09/2010 8:57 AM
VehicleLocation.java	1 KB	Java Source File	30/09/2010 8:57 AM	RoutePlanning_Requires_I2.java	1 KB	Java Source File	30/09/2010 8:57 AM
VehicleLocation_LocationBy.java	1 KB	Java Source File	30/09/2010 8:57 AM	RoutePlanning_System.java	1 KB	Java Source File	30/09/2010 8:57 AM
VehicleLocation_LocationBy_GPS.java	1 KB	Java Source File	30/09/2010 8:57 AM				
RoutePlanning_Management.java	1 KB	Java Source File	30/09/2010 8:57 AM				
Rule_1.java	1 KB	Java Source File	30/09/2010 8:57 AM				
Rule_2.java	2 KB	Java Source File	30/09/2010 8:57 AM				
Rule_3.java	2 KB	Java Source File	30/09/2010 8:57 AM				
Use_RoutePlanning_Implementation_IMP2.java	2 KB	Java Source File	30/09/2010 8:57 AM				
Use_RoutePlanning_Implementation_IMP3.java	2 KB	Java Source File	30/09/2010 8:57 AM				
Use_Traffic_Cong_TSP.java	1 KB	Java Source File	30/09/2010 8:57 AM				
				Test_Adaptive_Behaviour.java	29 KB	Java Source File	30/09/2010 8:57 AM

Εικόνα 22 - Η έξοδος του εργαλείου προσαρμοστικού συστήματος επίγνωσης της περιβάλλουσας κατάστασης

3.4 Παράδειγμα ανάπτυξης εφαρμογής σύμφωνα με την προσέγγιση

Σε αυτή την ενότητα, περιγράφεται η χρήση του μοντέλου και το εργαλείο ανάπτυξης, στα συστήματα πλοήγησης οχημάτων.

Το παράδειγμα που θα χρησιμοποιηθεί στη παρούσα ενότητα, αφορά τα συστήματα πλοήγησης των οχημάτων. Τα συστήματα πλοήγησης, μέσω του συστήματος εύρεσης διαδρομής του οχήματος, δίνουν τη δυνατότητα στον οδηγό να εκτελέσει το προγραμματισμένο ταξίδι του. Η βασική λειτουργία του σχεδιασμού διαδρομής, έγκειται στο να λάβεις από τον οδηγό την παρούσα θέση του και τον προορισμό του, και έπειτα να παρέχεις όλες τις κατάλληλες διαδρομές μεταξύ των δύο σημείων. Παρακάτω, παρατίθενται κάποιες περιπτώσεις όπου αυτή η εφαρμογή

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

χρειάζεται να λάβει υπόψη τις πληροφορίες περιβάλλουσας κατάστασης και να προσαρμοστεί πάνω σε αυτές:

Αλγόριθμοι σχεδιασμού διαδρομής: Το πλεονέκτημα της τεχνολογίας αυτής, είναι ότι δίνει τη δυνατότητα στα συστήματα των οχημάτων να ενσωματώνουν πληροφορίες περιβάλλουσας κατάστασης από διαφορετικές πηγές, οι οποίες μπορούν να χρησιμοποιηθούν για να ενισχύσουν τη λειτουργία του σχεδιασμού διαδρομής. Τέτοιου είδους πληροφορίες είναι, πρώτον η πληροφόρηση της κυκλοφοριακής κατάστασης από μια μονάδα στο δρόμο ή από έναν πάροχο υπηρεσιών πληροφοριών κυκλοφοριακής κατάστασης, και δεύτερον, οι προτιμήσεις του οδηγού από το κινητό του ή το PDA. Επιπροσθέτως, τα αυτοκίνητα παράγονται σε διαφορετικά μοντέλα και με διαφορετικές τεχνολογικές δυνατότητες και έτσι η κατασκευή μιας εφαρμογής σχεδίασης διαδρομών που να καλύπτει τις ανάγκες πολλών αυτοκινήτων είναι μια πρόκληση, τη στιγμή που υπάρχει διακύμανση μεταξύ των αυτοκινήτων που χρειάζονται να χειρίζονται (πχ κάποια αυτοκίνητα δεν έχουν τη δυνατότητα να επικοινωνούν με το PAD του οδηγού ώστε να έχουν τις προτιμήσεις του). Επιπλέον, το αυτοκίνητο μπορεί να ενισχυθεί με την πρόσθεση ενός στοιχείου, το οποίο παρέχει τον τύπο των πληροφοριών περιβάλλουσας κατάστασης οι οποίες συνυπολογίζονται κατά τη διάρκεια που αλγόριθμος σχεδίασης διαδρομών σχεδιάζει το χρόνο (π.χ. η εγκατάσταση μια ασύρματης σύνδεσης δίνει τη δυνατότητα επικοινωνίας μεταξύ του αυτοκινήτου και της μονάδας στο δρόμο). Ως εκ τούτου, το σύστημα σχεδίασης διαδρομών πρέπει να έχει διαφορετικούς αλγόριθμους για να υπολογίζει τις πιθανές διαδρομές βασιζόμενο στη διαθέσιμη περιβάλλουσα κατάσταση πληροφοριών.

Συνεργατική αυτοκινητοπομπή: Έστω, ότι δύο φίλοι φεύγουν για ένα ταξίδι και θέλουν να πάρουν την ίδια διαδρομή. Έτσι, τα οχήματά τους επικοινωνούν μεταξύ τους και το ένα από αυτά έχει το ρόλο του οδηγού (πχ. Το όχημα οδηγός-ηγέτης σχεδιάζει το ταξίδι) ενώ το άλλο ακολουθεί τη διαδρομή που διάλεξε το πρώτο όχημα. Σε αυτή τη περίπτωση, η διαδρομή του οχήματος που ακολουθεί, αποκτάται απευθείας από το όχημα-οδηγός. Ως εκ τούτου, το σύστημα σχεδίασης διαδρομών πρέπει να υποστηρίζει δύο λειτουργίες, την *αυτόνομη λειτουργία*, όπου το όχημα σχεδιάζει τη διαδρομή από μόνο του χρησιμοποιώντας του αλγόριθμους σχεδίασης διαδρομών, και την *λειτουργία αυτοκινητοπομπής*, όπου το όχημα αποκτά το σχέδιο διαδρομής από το όχημα-οδηγό.

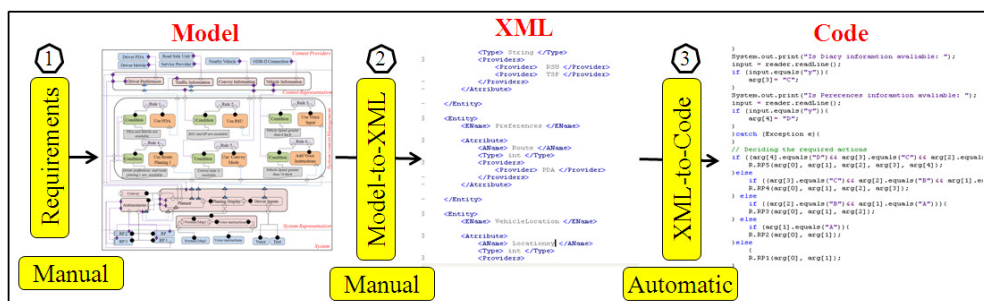
Φωνητικές οδηγίες διαδρομής: Όταν το όχημα τρέχει με υψηλή ταχύτητα, είναι δύσκολο για τον οδηγό να συγκεντρωθεί ταυτόχρονα και στο δρόμο και στον χάρτη. Γι' αυτό, σε τέτοιες καταστάσεις η εφαρμογή πρέπει να παρέχει όχι μόνο τη δυνατότητα απεικόνισης του χάρτη με τη διαδρομή του ταξιδιού, αλλά και τη παροχή φωνητικών οδηγιών ώστε ο οδηγός να μπορεί να είναι συγκεντρωμένος μόνο στο δρόμο. Έτσι η εμφάνιση σχεδίασης διαδρομών πρέπει να υποστηρίζει δύο λειτουργίες (κανονικής και υψηλής ταχύτητας λειτουργίες).

Είσοδοι δεδομένων οδηγού: Για τη βελτίωση της ασφάλειας του οδηγού καθοδόν, το σύστημα πλοήγησης έχει δυο μεθόδους για την εισαγωγή των δεδομένων του οδηγού. Η πρώτη είναι χρησιμοποιώντας τη λειτουργία κειμένου, όταν ο οδηγός σχεδιάζει το ταξίδι του πριν ξεκινήσει, και η δεύτερη είναι η λειτουργία φωνητικής αναγνώρισης, όταν το όχημα βρίσκεται εν κινήσει.

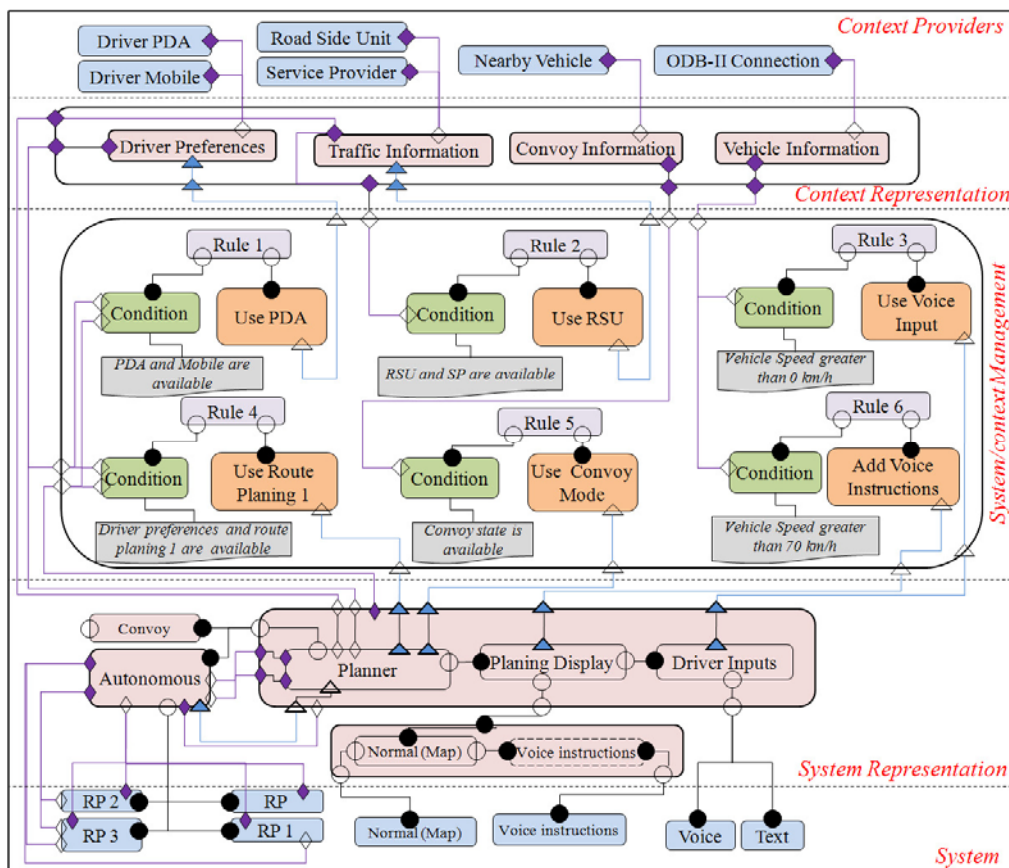
3.4.1 Τα βήματα της ανάπτυξης του προσαρμοστικού συστήματος λογισμικού επίγνωσης της περιβάλλουσας κατάστασης, χρησιμοποιώντας την προσέγγιση

Τα βήματα της χρήσης του μοντέλου και του εργαλείου που αναπτύχθηκαν αναλύονται ακολούθως και απεικονίζονται στην εικόνα 23:

1. Οι απαιτήσεις του συστήματος χρησιμοποιούνται για να καθορίσουν ένα ολοκληρωμένο μοντέλο του συστήματος, της περιβάλλουσας κατάστασης, και των σχέσεων τους. Το μοντέλο αυτό διαμορφώνεται στο μετά-μοντέλο μας και βασίζεται στο μοντέλο στοιχείων μας (εικόνα 24).
2. Στα σχεδιασμένα μοντέλα μετασχηματίζονται χειροκίνητα στις αντίστοιχες περιγραφές XML.
3. Οι περιγραφές XML που δημιουργήθηκαν στο 2^ο βήμα χρησιμοποιούνται για να παράγουν την υλοποίηση του συστήματος.



Εικόνα 23 - Διεργασία χρήσης της προσέγγισης για την ανάπτυξη προσαρμοστικών συστημάτων επίγνωσης της περιβάλλουσας κατάστασης



Εικόνα 24 - Μοντέλο στοιχείων

3.4.2 Δημιουργία περιγραφών XML για το προσαρμοστικό σύστημα εύρεσης διαδρομής

Βάσει του σχήματος XML που αναλύθηκε ανωτέρω, σε αυτή την υπο-ενότητα αναλύονται τα αρχεία XML τα οποία αντιστοιχούν στο μοντέλο του συστήματος που απεικονίζεται στην εικόνα 24.

Αρχικά, όσον αφορά το μοντέλο περιβάλλουσας κατάστασης, υπάρχουν δύο τύποι πληροφοριών του στοιχείου που χρειάζεται να ληφθούν υπόψη, η πρώτη αφορά το στοιχείο περιβάλλοντος, που είναι η πληροφορία περιβάλλοντος που επηρεάζει τη λειτουργία της σχεδίασης διαδρομής, όπως οι προτιμήσεις του οδηγού και η πληροφορία κυκλοφοριακής συμφόρησης, και η δεύτερη είναι το η περιβάλλουσα κατάσταση συστήματος, το οποίο διαχωρίζεται σε δύο τύπους (α) τα εκτελούμενα στοιχεία του συστήματος και (β) η κατάσταση των μη ενεργών στοιχείων του συστήματος, που αντιπροσωπεύουν το στοιχείο που μπορεί να χρησιμοποιηθεί από το σύστημα όταν αυτό προσαρμόζεται από μια ρύθμιση σε μια άλλη. Ο δεύτερος τύπος περιβάλλουσας κατάστασης θα μοντελοποιηθεί μαζί με το λειτουργικό σύστημα (functional system), και εδώ εστιάζουμε μόνο στο μοντέλο περιβάλλουσας κατάστασης του περιβάλλοντος. Βάση του σχήματος XML του στοιχείου που αναφέρθηκε στην εικόνα 19, η εικόνα 25 παρουσιάζει το μοντέλο περιβάλλουσας κατάστασης του συστήματος πλοήγησης.

```

<Context>
  <Entity>
    <EName> TrafficInformaiton </EName>
    <Attribute>
      <AName> Congestions </AName>
      <Type> String </Type>
      <Providers>
        <Provider> RSU </Provider>
        <Provider> TISP </Provider>
      </Providers>
    </Attribute>
    <Attribute>
      <AName> BlockedRoads </AName>
      <Type> String </Type>
      <Providers>
        <Provider> RSU </Provider>
        <Provider> TISP </Provider>
      </Providers>
    </Attribute>
  </Entity>

  <Entity>
    <EName> DriverPreferences </EName>

    <Attribute>
      <AName> RoutePreference </AName>
      <Type> String </Type>
      <Providers>
        <Provider> DriverPDA </Provider>
        <Provider> DriverMobile </Provider>
      </Providers>
    </Attribute>
  </Entity>

  <Entity>
    <EName> VehicleInformation </EName>
    <Attribute>
      <AName> VehicleSpeed </AName>
      <Type> String </Type>
      <Providers>
        <Provider> OBD-IIScanner </Provider>
      </Providers>
    </Attribute>
  </Entity>

  <Entity>
    <EName> Cooperative_Convoy </EName>

    <Attribute>
      <AName> ConvoyState </AName>
      <Type> String </Type>
      <Providers>
        <Provider> NearbyVehicle </Provider>
      </Providers>
    </Attribute>
  </Entity>
</Context>

```

Εικόνα 25 - Η διάταξη XML του μοντέλου περιβάλλουσας κατάστασης

Δεύτερον, το μοντέλο του λειτουργικού συστήματος: Το ανωτέρω παράδειγμα σύνθεσης του λειτουργικού συστήματος αποτελείται από τρία στοιχεία: (α) τον σχεδιαστή διαδρομής, που έχει δύο μοντέλα (π.χ. αυτόνομο και αυτοκινητοπομπή), (β) την εμφάνιση της διαδρομής, η οποία μπορεί να είναι μόνο ένας χάρτης ή ένας χάρτης με φωνητική καθοδήγηση, (γ) την είσοδο του οδηγού, που μπορεί να είναι σε μορφή κειμένου ή φωνητικών οδηγιών, βάσει της ταχύτητας του οχήματος. Με βάση το σχήμα XML του συστήματος που περιγράφεται στην εικόνα 20, η εικόνα 26 περιγράφει το σύστημα λογισμικού.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

```

<System>
  <Component>
    <CName> RoutePlanner </CName>
    <ProFunction> RoutePlaning </ProFunction>
    <ComImp>
      <CMName> RoutePlaningConvoy </CMName>
    </ComImp>
  </Component>
  <Component>
    <CMName> RoutePlaningAutonomous </CMName>
  </Component>
  <Component>
    <CName> RoutePlaningConvoy </CName>
    <ProFunction> RoutePlaning </ProFunction>
    <ComImp>
      <CMName> NBV_Routing </CMName>
    </ComImp>
  </Component>
  <Component>
    <CName> RoutePlaningAutonomous </CName>
    <ProFunction> RoutePlaning </ProFunction>
    <ComImp>
      <CMName> IMP3 </CMName>
      <ReqContext> Congestions </ReqContext>
      <ReqContext> BlockedRoads </ReqContext>
      <ReqContext> RoutePreference </ReqContext>
    </ComImp>
    <ComImp>
      <CMName> IMP2 </CMName>
      <ReqContext> Congestions </ReqContext>
      <ReqContext> BlockedRoads </ReqContext>
    </ComImp>
    <ComImp>
      <CMName> IMP1 </CMName>
      <ReqContext> RoutePreference </ReqContext>
    </ComImp>
    <ComImp>
      <CMName> IMP </CMName>
    </ComImp>
  </Component>
  <Component>
    <CName> RoutePlaningDisply </CName>
    <ProFunction> RoutsDisplay </ProFunction>
    <ComImp>
      <CMName> Map </CMName>
    </ComImp>
    <ComImp>
      <CMName> MapwithVoice </CMName>
    </ComImp>
  </Component>
  <Component>
    <CName> DriverInput </CName>
    <ProFunction> DriverInputs </ProFunction>
    <ComImp>
      <CMName> VoiceMode </CMName>
    </ComImp>
    <ComImp>
      <CMName> TextMode </CMName>
    </ComImp>
  </Component>
</System>

```

Εικόνα 26 - Το σχήμα XML του μοντέλου συστήματος

Τέλος, το μοντέλο διαχείρισης του συστήματος: Στο ανωτέρω παράδειγμα, υπάρχει ένα σύνολο κανόνων που κατευθύνει τη προσαρμογή του συστήματος, βάσει της τρέχουσας κατάστασης του συστήματος (πχ η περιβάλλουσα κατάσταση συστήματος) και της κατάστασης του περιβάλλοντος (πχ η περιβάλλουσα κατάσταση προβλήματος), και το διαθέσιμο στοιχείο που μπορεί να χρησιμοποιηθεί από το σύστημα (πχ περιβάλλουσα κατάσταση λύσης). Με βάση το σχήμα XML που περιγράφεται στην εικόνα 21, η εικόνα 27 περιγράφει μερικούς από τους κανόνες προσαρμογής του συστήματος.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος


```

<Management>
  <AdaptationRule>
    <Condition>
      <Att> RoutePreference </Att>
      <Aop> = </Aop>
      <Val> active </Val>
      <Lop> null </Lop>
    </Condition>
    <Action>
      <type> Use </type>
      <cname> RoutePlaning </cname>
      <ciname> IMP1 </ciname>
    </Action>
  </AdaptationRule>
  <AdaptationRule>
    <Condition>
      <Att> RoutePreference </Att>
      <Aop> = </Aop>
      <Val> active </Val>
      <Lop> and </Lop>
    </Condition>

    //other conditions

    <Action>
      <type> Use </type>
      <cname> RoutePlaning </cname>
      <ciname> IMP3 </ciname>
    </Action>
  </AdaptationRule>
  <AdaptationRule>
    <Condition>
      <Att> RoutePreference.DriverPDA </Att>
      <Aop> = </Aop>
      <Val> active </Val>
      <Lop> and </Lop>
    </Condition>
    <Condition>
      <Att> RoutePreference.DriverMobile </Att>
      <Aop> = </Aop>
      <Val> active </Val>
      <Lop> null </Lop>
    </Condition>
    <Action>
      <type> Use </type>
      <cname> RoutePreference </cname>
      <ciname> DriverMobile </ciname>
    </Action>
  </AdaptationRule>
  //Other rules
</Management>

```

Εικόνα 27 - Η διάταξη XML της διαχείρισης συστήματος

3.4.3 Η χρήση του εργαλείου για την ανάπτυξη ενός προσαρμοστικού συστήματος σχεδίασης διαδρομής επίγνωσης της περιβάλλουσας κατάστασης

1^ο Βήμα: Άνοιγμα του φακέλου του εργαλείου και επιλογή του αρχείου **Run.bat**, η κύρια οθόνη του εργαλείου μας θα εμφανιστεί όπως φαίνεται στην εικόνα 18.

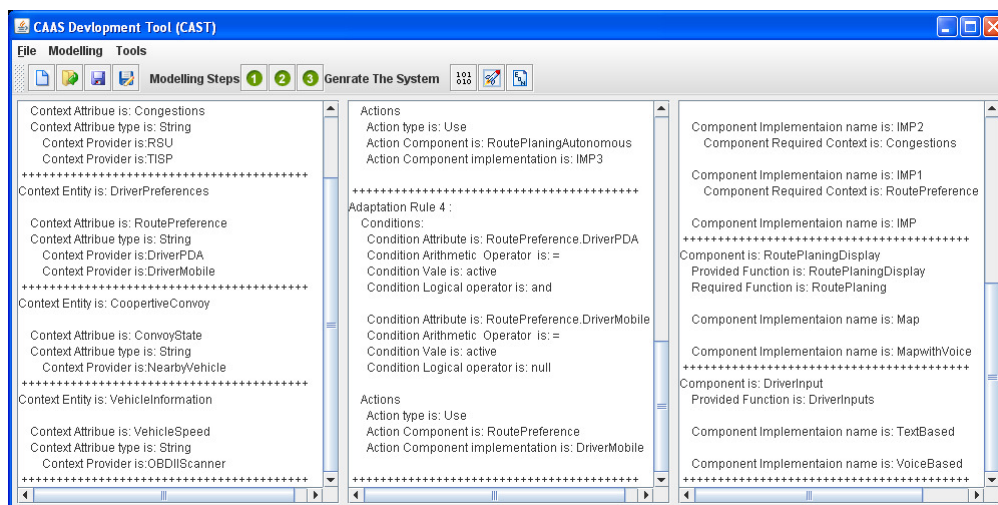
2^ο Βήμα: Δημιουργία ενός νέου έργου από το μενού "Αρχείο" ή επιλέγοντας το αντίστοιχο κουμπί από την εργαλειοθήκη, ένα νέου παράθυρο διαλόγου θα εμφανιστεί για την επιλογή της ονομασίας του έργου και της τοποθεσίας που θα αποθηκευθεί.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

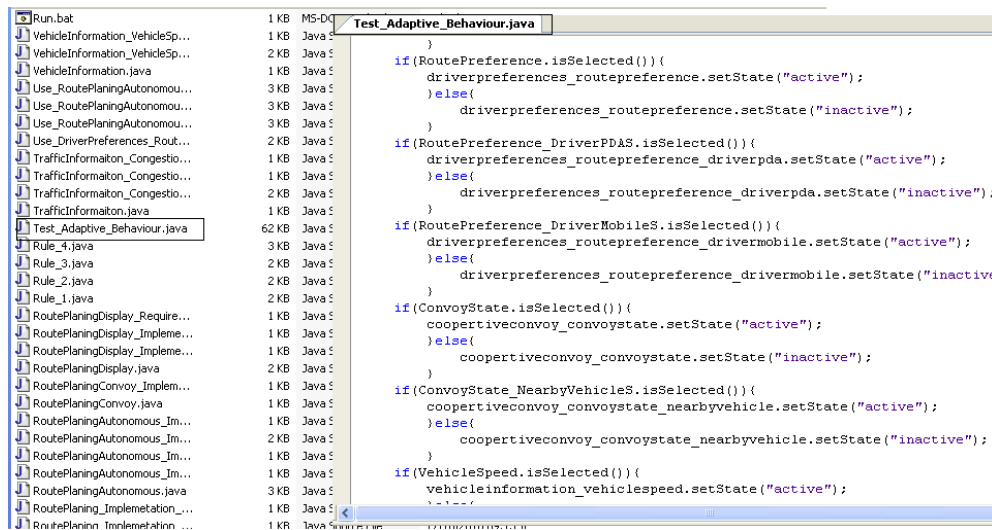
Φούντας Γεώργιος

3^ο Βήμα: Φόρτωση των αρχείων XML που περιγράφηκαν στις εικόνες 25, 26 και 27, από το μενού "Σχεδίαση" (βήματα 1, 2 και 3) ή από την εργαλειοθήκη (οι αριθμοί 1, 2 και 3). Το εργαλείο θα προσπελάσει τα τρία αρχεία και θα φανερώσει τις πληροφορίες τους όπως φαίνεται στην εικόνα 28.



Εικόνα 28 - Απεικόνιση της οθόνης του εργαλείου, μετά από τη φόρτωση των XML αρχείων

4^ο Βήμα: Δημιουργία της υλοποίησης του συστήματος επιλέγοντας το μενού "Εργαλεία" και ακολούθως παραγωγή του αντικειμένου του μενού του συστήματος. Τα αρχεία που αντιστοιχούν στην περιγραφή της περίπτωσης θα δημιουργηθούν όπως απεικονίζονται στην εικόνα 29.



Εικόνα 29 - Παραγωγή του αντικειμένου του μενού του συστήματος

5^ο Βήμα: Εκτέλεση ενός απλού ελέγχου της προσαρμοστικής συμπεριφοράς του συστήματος πατώντας το κουμπί εκτέλεση από την εργαλειοθήκη ή επιλέγοντας την αντίστοιχη εντολή από το μενού "εργαλεία". Αυτό το κουμπί καλεί ένα από τα παραγόμενα αρχεία μας, το οποίο είναι μια κλάση για την οπτικοποίηση των στοιχείων της περιβάλλουσας κατάστασης και του συστήματος. Επιπροσθέτως, ο σχεδιαστής μπορεί να μεταβάλλει την κατάσταση της περιβάλλουσας κατάστασης επιλέγοντας ή από-επιλέγοντας τα οπτικοποιημένα στοιχεία. Έπειτα πατώντας το

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

κουμπί "Προσάρμοσε το σύστημα", το σύστημα προσαρμόζεται στη τρέχουσα κατάσταση της περιβάλλουσας κατάστασης. Η εικόνα 30-A δείχνει την αρχική φόρτωση του GUI μας, όπου ορίζουμε τις προκαθορισμένες τιμές της περιβάλλουσας κατάστασης και έπειτα το σύστημα προσαρμόζεται σε αυτό (οι αρχικές ρυθμίσεις του συστήματος λογισμικού απεικονίζονται στην εικόνα 30-A. Η εικόνα 30-B δείχνει τις ρυθμίσεις του συστήματος, όταν η περιβάλλουσα κατάσταση (πληροφόρηση κυκλοφοριακής συμφόρησης) είναι διαθέσιμη. Η κατάσταση αντιστοιχεί στον κανόνα 2, και έπειτα η υλοποίηση σχεδίασης διαδρομής 2 επιλέγεται.

Εικόνα 30 - Ο έλεγχος της προσαρμοστικής συμπεριφοράς του συστήματος

Κεφάλαιο 4 Μία προσανατολισμένη στις απόψεις προσέγγιση για την προσαρμογή των εφαρμογών κινητού ηλεκτρονικού εμπορίου

Στη παρούσα ενότητα παρουσιάζεται μια προσέγγιση η οποία εστιάζει στο πότε, πού και πώς μια προσαρμογή πρέπει να ενσωματωθεί σε μια εφαρμογή κινητού εμπορίου. Για το σκοπό αυτό υιοθετείται μια προσέγγιση δύο σταδίων.

Στο πρώτο στάδιο η εφαρμογή σχεδιάζεται χωρίς να λαμβάνονται υπόψη τα χαρακτηριστικά προσαρμογής της εφαρμογής, ωστόσο η διαδικασία σχεδίασης έχει διαμορφωθεί έτσι ώστε να παράγει μια «έτοιμη προς προσαρμογή» εφαρμογή, στην οποία η προσαρμοστικότητα θα μπορεί εύκολα να ενσωματωθεί.

Στο δεύτερο στάδιο περιγράφεται μια αρχιτεκτονική λογισμικού η οποία αξιοποιεί τις δομές που παρέχονται από τον προσανατολισμένο στις απόψεις προγραμματισμό, έτσι ώστε να κάνει συνυφάνει τη προσαρμογή στη μη-προσαρμοσμένη εφαρμογή που σχεδιάστηκε στο πρώτο στάδιο.

4.1 Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει

4.1.1 Αδυναμίες που προσπαθεί να καλύψει η προσέγγιση

Η βασική αδυναμία που προσπαθεί να καλύψει η προσέγγιση είναι η ανάμειξης του κώδικα προσαρμογής με τον κώδικα του πυρήνα της εφαρμογής. Στα μειονεκτήματα της ανάμειξης του κώδικα προσαρμογής με τον κώδικα του πυρήνα της εφαρμογής περιλαμβάνονται τα εξής:

- 1) Η αύξηση του βαθμού ένωσης [22] στον κώδικα εφαρμογής, από τη στιγμή που η κύρια λογική εφαρμογής και η λογική προσαρμογής αναμειγνύονται.
- 2) Η αυξανόμενη δυσκολία συντήρησης, από τη στιγμή που οι μετατροπές στη λογική εφαρμογής θα δημιουργήσουν πιθανώς αλλαγές στην λογική προσαρμογής και το αντίστροφο.
- 3) Μείωση της δυνατότητας για την επαναχρησιμοποίηση των ενοτήτων είτε από την κύρια εφαρμογή, είτε από τον κώδικα προσαρμογής, από τη στιγμή που αναμειγνύονται και προσαρμόζονται για τις ανάγκες της συγκεκριμένης εφαρμογής.

Επιπροσθέτως, πολλές από τις τεχνικές που χρησιμοποιούνται, σε πολλές περιπτώσεις διαχειρίζονται μόνο ένα μικρό υποσύνολο της περιβάλλουσας κατάστασης, χωρίς να παρέχουν επαρκές υπόβαθρο για τη διεύρυνση της περιβάλλουσας κατάστασης που λαμβάνεται υπόψη. Σε άλλες περιπτώσεις, δεν γίνεται χρήση των αρχείων προσαρμοστικής πολιτικής, τα οποία παρέχουν πιο ευέλικτη προσαρμογή.

4.1.2 Στόχοι της προσέγγισης

Αφού αναλυθεί η δομή μιας εφαρμογής κινητού εμπορίου και περιγραφεί ο τρόπος με τον οποίο θα πρέπει να δομηθεί ώστε να επιδέχεται προσαρμογή, θα

προταθεί μια προσέγγιση βασισμένη στις απόψεις προκειμένου να συνυφανθεί η προσαρμοσμένη συμπεριφορά της εφαρμογής στη βασική εφαρμογή. Με αυτόν τον τρόπο επιτυγχάνεται η ενθυλάκωση της λογικής για την προσαρμογή της συμπεριφοράς κατά το χρόνο εκτέλεσης μιας εφαρμογής σε μία ενότητα λογισμικού. Η ενότητα αυτή (Context Consumer) είναι η μηχανή (engine) για την πυροδότηση (firing) των κανόνων προσαρμογής, καθένας από τους οποίους αποτελείται από μια συνθήκη που όταν ικανοποιηθεί θα λάβει χώρα μια προσαρμογή και μια δράση η οποία υποδεικνύει την κατάλληλη προσαρμοσμένη αντίδραση.

4.1.3 Καινοτομίες της προσέγγισης

Το βασικό πλεονέκτημα της προσέγγισης που μελετάται στη παρούσα ενότητα (separation of concerns/aspect-oriented paradigm), είναι ο μεγάλος βαθμός ανεξαρτησίας, των εμπλεκόμενων της κύριας εφαρμογής και των εμπλεκόμενων της εφαρμογής προσαρμογής. Σημαντικό επίσης πλεονέκτημα είναι η δυνατότητα εύκολης αλλαγής του κώδικα των κλάσεων της κύριας εφαρμογής, καθώς και η δυνατότητα επαναχρησιμοποίησης τους, όπως επίσης και η ευκολία συντήρησης και ελέγχου του κώδικα προσαρμογής.

4.2 Βασικές έννοιες της προσέγγισης

Για το σχεδιασμό συστημάτων πληροφορίας περιβάλλουσας κατάστασης, έχουν προταθεί πολλοί ορισμοί και ποικίλες τεχνικές. Ειδικότερα στο πεδίο εφαρμογών κινητού εμπορίου, οι Μπένου και Βασιλάκης διατυπώνουν τον ακόλουθο ορισμό σε σχέση με την πληροφορία περιβάλλουσας κατάστασης:

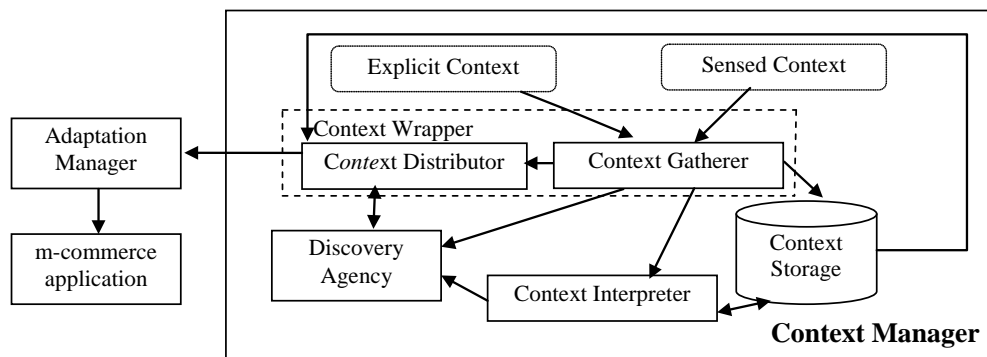
«Η πληροφορία περιβάλλουσας κατάστασης μιας εφαρμογής κινητού εμπορίου, είναι το κάθε μέρος πληροφορίας που μπορεί να χρησιμοποιηθεί ώστε να χαρακτηρίσει την κατάσταση μιας οντότητας, όπου μπορεί να θεωρηθεί σχετική με την αλληλεπίδραση του χρήστη με την συγκεκριμένη εφαρμογή. Η κατάσταση της οντότητας μπορεί να αλλάζει είτε δυναμικά είτε στατικά, όπου η συσχέτιση της οντότητας με την αλληλεπίδραση της εφαρμογής του χρήστη, μπορεί να εξαχθεί από τη δυνατότητα που υπάρχει να αξιοποιηθεί η πληροφορία που περιγράφει τη κατάσταση της οντότητας, ώστε να βελτιστοποιείται αυτή η αλληλεπίδραση, έτσι ώστε να μεγιστοποιηθεί η εμπορική αξία της εφαρμογής».

Με τον όρο «οντότητα», εδώ καλύπτονται τόσο οι «οντότητες», όσο και οι «συσχετίσεις» ενός μοντέλου οντοτήτων-συσχετίσεων (ER-Model). Σύμφωνα με τον προαναφερθέντα ορισμό προκύπτει ότι, η σχετιζόμενη πληροφορία περιβάλλουσας κατάστασης, η οποία χρησιμοποιείται κατά την εμπορική συναλλαγή που πραγματοποιείται μέσω μιας κινητής συσκευής ενός χρήστη, μπορεί να οργανωθεί ως μια συλλογή αλληλένδετων οντοτήτων. Οι ιδιότητες αυτών των οντοτήτων αναπαριστούν τα στοιχεία περιβάλλουσας κατάστασης, τα οποία θα χρησιμοποιηθούν από τις υπηρεσίες υλοποίησης της προσαρμογής της κινητής εφαρμογής.

Σχετικά, με τη συλλογή, τη διαχείριση, την αποθήκευση και τη κατανομή της περιβάλλουσας κατάστασης, πλήθος ερευνητών πρότεινε συστήματα λογισμικού με διάφορες τεχνικές για τη διαχείριση πληροφοριών περιβάλλουσας κατάστασης. Η πιο κοινά αποδεκτή αρχιτεκτονική είναι αυτή που περιλαμβάνει ένα ή περισσότερα κύρια στοιχεία για τη διαχείριση της πληροφορίας περιβάλλουσας κατάστασης, και μερικά καταναμημένα στοιχεία για συλλογή της πληροφορίας περιβάλλουσας κατάστασης. Αυτή η προσέγγιση, προτάθηκε από ένα πλήθος ερευνητών και έργων.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Οι Μπένου και Βασιλάκης, παρουσίασαν μια αρχιτεκτονική για τη διαχείριση περιβάλλουσας κατάστασης (που καλείται Context Manager), προσαρμοσμένη στις ιδιαιτερότητες των εφαρμογών κινητού εμπορίου, με διακριτά στοιχεία συλλογής, επεξεργασίας και διανομής περιβάλλουσας κατάστασης. Η προσέγγιση του “Context Manager” απεικονίζεται στην εικόνα 31.



Εικόνα 31 - Context Manager

4.3 Μεθοδολογία ανάπτυξης εφαρμογών σύμφωνα με την προσέγγιση

4.3.1 Η δομή μιας διαδικτυακής εφαρμογής

Προκειμένου να είναι εφικτή η διαδικασία της προσαρμογής, η οποία θα επιτελεστεί στα πλαίσια του Διαχειριστή προσαρμογής (Adaptation Manager), θα πρέπει ο πυρήνας της εφαρμογής να είναι σχεδιασμένος κατάλληλα ώστε να μπορούν να εκτελούνται οι λειτουργίες της προσαρμογής. Πριν προχωρήσουμε στην παρουσίαση του σχεδιασμού του πυρήνα της εφαρμογής, κατά τρόπο που διευκολύνει την εκτέλεση της προσαρμογής, θα αναλυθεί η δομή μιας διαδικτυακής εφαρμογής κινητού εμπορίου (web-based m-commerce application). Η επιλογή των διαδικτυακών εφαρμογών κινητού εμπορίου οφείλεται στο γεγονός ότι η πλειοψηφία των εφαρμογών κινητού εμπορίου υλοποιούνται πλέον με τεχνολογίες διαδικτύου και έτσι η εμβάθυνση στη λειτουργία της διαδικτυακής εφαρμογής θα συνεισφέρει στην πληρέστερη κατανόηση της διαδικασίας προσαρμογής που περιγράφεται παρακάτω.

Το μοντέλο υπερκειμένου (hypertext model) ορίστηκε στη γλώσσα WebML για να περιγράψει τον τρόπο που οργανώνεται το περιεχόμενο (content) μιας εφαρμογής διαδικτύου (web application) σε δομή υπερκειμένου και συνακόλουθα τον τρόπο με τον οποίο πλοηγείται ο χρήστης εντός του περιεχομένου αυτού. Μια διαδικτυακή εφαρμογή περιλαμβάνει ένα βασικό μοντέλο δεδομένων (που συνήθιστα αποτυπώνεται με κάποια μεθοδολογία, π.χ. UML ή ER), μία βασική επιχειρηματική λογική και ένα μοντέλο πλοήγησης (navigation model), που τελικά συνδιαμορφώνουν μια συλλογή σελίδων (pages) [23]. Οι σελίδες δεδομένων φέρουν την πληροφορία του μοντέλου δεδομένων μετασχηματισμένη μέσω της επιχειρηματικής λογικής, και οργανώνονται μέσω του μοντέλου πλοήγησης ώστε να διανέμουν αυτή την πληροφορία στους χρήστες με εύχρηστο τρόπο. Μια σελίδα διαδικτύου συνίσταται από ένα σύνολο από ενότητες περιεχομένου (content units) οι οποίες είναι υποδοχείς στοιχειωδών τμημάτων πληροφορίας, τα οποία ανακτώνται από τις πηγές δεδομένων (data sources). Κάθε ενότητα περιεχομένου έχει μια σειρά

από ιδιότητες παρουσίασης (*presentation properties*) και συγκεκριμένη λειτουργικότητα (*functionality*).

Η βασική εφαρμογή αποτελείται από μία συνεπή *διεπαφή χρήστη* (consistent user interface [24]) και από ένα αριθμό από *υπηρεσίες* (services), τα οποία υλοποιούν τη βασική και μη context-aware λειτουργικότητα του συστήματος. Η *διεπαφή χρήστη* αποτελείται από τις σελίδες (web pages) οι οποίες με τη σειρά τους συντίθενται από τις *μονάδες περιεχομένου*. Οι *υπηρεσίες* είναι εκείνες οι συνιστώσες της εφαρμογής οι οποίες τροφοδοτούν με δεδομένα και λειτουργικότητα τις ενότητες περιεχομένου. Οι υπηρεσίες δεν είναι πληροφορίες περιβάλλουσας κατάστασης και η συμπεριφορά τους παραμένει η ίδια και ανεξάρτητη από τις μεταβολές της περιβάλλουσας πληροφορίας. Επιπρόσθετα, αυτές οι υπηρεσίες καθιστούν διαθέσιμη τη λειτουργικότητά τους μέσα από μια σειρά από διακριτές εναλλακτικές επιλογές καλούμενες *διεπαφές υπηρεσιών* (*service interfaces*) (π.χ. πολυμεσική εκδοχή – multimedia- έναντι εκδοχής απλού κειμένου) προκειμένου να υποστηρίξουν τις διαφορετικές προτιμήσεις κατά τη διαδικασία της προσαρμογής (adaptation process). Παρόμοιες δομές χρησιμοποιούνται και σε άλλες μεθοδολογίες σχεδιασμού διαδικτυακών εφαρμογών UWE [27, 28] και Hera [25, 26].

4.3.2 Η μεθοδολογία σχεδιασμού μιας εφαρμογής κινητού εμπορίου

Για τον σχεδιασμό μιας εφαρμογής κινητού εμπορίου υιοθετείται η μέθοδος UWE. Η μέθοδος UWE είναι μια αντικειμενοστρεφής μέθοδος μηχανικής εφαρμογών ιστού που στηρίζεται στη UML. Έτσι προκειμένου να σχεδιάσουμε εφαρμογή κινητού εμπορίου, δεδομένων των διαγραμμάτων UML και του μοντέλου περιεχομένου (UML class diagrams), πρώτα δημιουργούμε το μοντέλο πλοήγησης (navigation model). Στο μοντέλο πλοήγησης αποτυπώνονται τόσο η λειτουργικότητα του συστήματος (μέσω process class, index, query, navigation class κ.λ.π.) όσο και οι σύνδεσμοι πλοήγησης (navigation links).

Στη συνέχεια δημιουργείται το μοντέλο παρουσίασης (presentation model) κάθε σελίδας, στο οποίο αποτυπώνονται ποιές πλοηγήσεις και ποιές λειτουργικότητες (process class, index, query, navigation class), περιλαμβάνονται σε κάθε σελίδα, με αναλυτική αναφορά στα δεδομένα της σελίδας και στους ελέγχους που θα χρησιμοποιηθούν. Τέλος δημιουργούμε το μοντέλο της διαδικασίας (process model) στο οποίο αποτυπώνονται οι διαφορετικές κλάσεις διαδικασίας κάθε σελίδας και οι σχέσεις μεταξύ τους, καθώς και οι αναλυτικές δραστηριότητες που είναι σχετικές με κάθε κλάση διαδικασίας. Αναλύοντας το μοντέλο παρουσίασης μιας σελίδας στα επιμέρους συστατικά του μέρη (process class, index, query, navigation class) προκύπτουν οι ενότητες περιεχομένου που ορίστηκαν νωρίτερα.

4.3.3 Τα βήματα ανάπτυξης μιας εφαρμογής κινητού εμπορίου

Προκειμένου να σχεδιαστεί μια σελίδα μιας εφαρμογής κινητού εμπορίου θα πρέπει τελικά να σχεδιαστούν τα συστατικά της μέρη, δηλ. οι ενότητες περιεχομένου (content units) που την απαρτίζουν. Ο σχεδιασμός μιας ενότητας περιεχομένου ανάγεται στον καθορισμό-σχεδιασμό των συνιστωσών της ενότητας ήτοι: i) της *διεπαφής* της, δηλαδή των στοιχείων ελέγχου που περιλαμβάνει (controls), ii) τις *ιδιότητες παρουσίασης* της διεπαφής, iii) τα δεδομένα με τα οποία θα τροφοδοτηθεί και iv) η επιπρόσθετη λειτουργικότητα της π.χ. δυνατότητα εισαγωγής μιας νέας

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

εγγραφής, ή η δυνατότητα αναλυτικότερης πληροφόρησης για μια εγγραφή (π.χ. μέσω μετάβασης σε μια νέα σελίδα) ή η δυνατότητα για πλοήγηση σε μια νέα σελίδα.

Ο καθορισμός/σχεδιασμός των παραπάνω συστατικών μερών μιας ενότητας περιεχομένου, μπορεί να γίνει μέσα από αντίστοιχες μεθόδους κατάλληλων υπηρεσιών σε μια service-oriented προσέγγιση. Ο τεμαχισμός του σχεδιασμού μιας ενότητας περιεχομένου, στον σχεδιασμό των επιμέρους μεθόδων για τον καθορισμό των ιδιοτήτων παρουσίασης, των δεδομένων και της λειτουργικότητας της κάθε ενότητας περιεχομένου, θα επιτρέψει την επίτευξη της διαδικασίας της προσαρμογής με ένα πιο ευέλικτο τρόπο, όπως θα διαφανεί στις επόμενες παραγράφους.

Επομένως η βασική εφαρμογή θα πρέπει να σχεδιαστεί με κατάλληλο τρόπο, (ενότητες περιεχομένου με τις αντίστοιχες μεθόδους τους), ώστε να δύναται να προσαρμοστεί. Ένας σχεδιασμός της βασικής εφαρμογής που επιτρέπει τη διαδικασία της προσαρμογής, μπορεί να πραγματοποιηθεί μέσα από τα ακόλουθα βήματα, τα οποία ακολουθούν τη φιλοσοφία σχεδιασμού της WebML [23]:

Βήμα 1: Κατά τη διάρκεια του υψηλού επιπέδου σχεδιασμού της εφαρμογής (high-level application design), θα πρέπει να προσδιοριστούν οι διαφορετικές σελίδες της εφαρμογής, και οι ενότητες περιεχομένου που μπορεί να περιλαμβάνει η κάθε σελίδα.

Βήμα 2 : Στον λεπτομερή σχεδιασμό της εφαρμογής (detailed application design), για κάθε ενότητα περιεχομένου ορίζεται η υπηρεσία που θα παρέχει το περιεχόμενο για την ενότητα, προσδιορίζοντας i) τις εναλλακτικές διεπαφές της υπηρεσίας (π.χ. για text-based ή multimedia πληροφορία), ii) την παραμετροποίηση των ερωτήσεων που υποβάλλει η υπηρεσία στη βάση δεδομένων της εφαρμογής προκειμένου να αντληθεί το περιεχόμενο της ενότητας.

Βήμα 3: Για κάθε υπηρεσία που ορίστηκε στο βήμα 2, ορίζονται οι κατάλληλες μέθοδοι που υλοποιούν τις σχετικές προδιαγραφές και πιο αναλυτικά:

Βήμα 3α: Για κάθε ενότητα περιεχομένου θα οριστούν οι μέθοδοι που παρέχουν την επιπρόσθετη λειτουργικότητά του (πέραν της άντλησης των δεδομένων, για παράδειγμα την εισαγωγή των στοιχείων κάποιας συναλλαγής).

Βήμα 3β: Για κάθε ενότητα περιεχομένου θα οριστεί η μέθοδος που θα παρέχει τον σχεδιασμό της διεπαφής της ενότητας περιεχομένου.

Βήμα 3γ: Για κάθε ενότητα περιεχομένου θα οριστεί η μέθοδος που θα ορίζει τις ιδιότητες παρουσίασης της ενότητας περιεχομένου.

Βήμα 3δ: Για κάθε ενότητα περιεχομένου θα οριστούν οι μέθοδοι οι οποίες θα δημιουργήσουν τον κώδικα πλευράς εξυπηρετούμενου (client-side code) που θα υποστηρίξει την αλληλεπίδραση της σελίδας με το χρήστη (χωρίς να απαιτείται η αποστολή των δεδομένων της σελίδας στο web server π.χ. έλεγχοι εγκυρότητας δεδομένων, μηνύματα κ.λπ.).

Βήμα 4: Στη συνέχεια θα πρέπει να οριστεί το βασικό (default) περιεχόμενο της σελίδας χωρίς να ληφθεί υπόψη η περιβάλλουσα πληροφορία, (π.χ. για περιπτώσεις που για οποιαδήποτε λόγο ο Context Manager αδυνατεί να παρέχει περιβάλλουσα πληροφορία), ή για τις περιπτώσεις που ο χρήστης της εφαρμογής επιλέγει να μην συνδεθεί στο σύστημα [οπότε να είναι δυνατή η χρήση πληροφοριών από το προφίλ του] ή να μην έχει παράσχει εξουσιοδότηση για χρήση της

περιβάλλουσας πληροφορίας που τον αφορά). Ο ορισμός του προκαθορισμένου περιεχομένου συνίσταται στον καθορισμό της πληροφορίας και λειτουργικότητας κάθε σελίδας μέσα από επιλογή συγκεκριμένων ενοτήτων περιεχομένου και πιο συγκεκριμένα, γίνονται οι εξής επιλογές:

- επιλέγονται οι μέθοδοι που θα δημιουργήσουν τη διεπαφή αυτών των ενοτήτων περιεχομένου. Οι μέθοδοι αυτές έχουν οριστεί στο βήμα 3β, ανωτέρω.
- επιλέγονται οι μέθοδοι που θα κληθούν για να γεμίσουν τις ενότητες περιεχομένου με δεδομένα. Οι μέθοδοι αυτές έχουν οριστεί στο βήμα 3α, ανωτέρω.
- επιλέγονται οι μέθοδοι που θα κληθούν για να ορίσουν τις ιδιότητες παρουσίασης των στοιχείων ελέγχου κάθε ενότητας περιεχομένου. Οι μέθοδοι αυτές έχουν οριστεί στο βήμα 3γ, ανωτέρω.
- επιλέγονται οι μέθοδοι που θα κληθούν για να δημιουργήσουν τον κώδικα για την υποστήριξη της αλληλεπίδρασης χρήστη-εφαρμογής. Οι μέθοδοι αυτές έχουν οριστεί στο βήμα 3δ, ανωτέρω.
- τέλος, επιλέγονται οι μέθοδοι που θα κληθούν για να ενσωματώσουν στη σελίδα την επιπρόσθετη λειτουργικότητα της ενότητας περιεχομένου. Οι μέθοδοι αυτές έχουν οριστεί στο βήμα 3α, ανωτέρω.

4.4 Παράδειγμα ανάπτυξης εφαρμογής σύμφωνα με την προσέγγιση

Η διαδικασία της προσαρμογής πραγματοποιείται εξ ολοκλήρου μέσα στα πλαίσια του *διαχειριστή προσαρμογής* μέσα από συγκεκριμένες λειτουργίες οι οποίες επιτελούνται από τα δομικά συστατικά του μέρη (εικόνα 32). Ειδικότερα η *συνιστώσα προσαρμογής παρουσίασης* (Presentation Adaptation Module) εκτελεί την προσαρμογή της παρουσίασης, η *συνιστώσα προσαρμογής περιεχομένου* (Content Adaptation Module) εκτελεί την προσαρμογή του περιεχομένου και η *συνιστώσα προσαρμογής λειτουργιών* (Operation Adaptation Module) εκτελεί την προσαρμογή των λειτουργιών -ουσιαστικά δηλαδή αυτές οι συνιστώσες υλοποιούν την προσαρμογή.

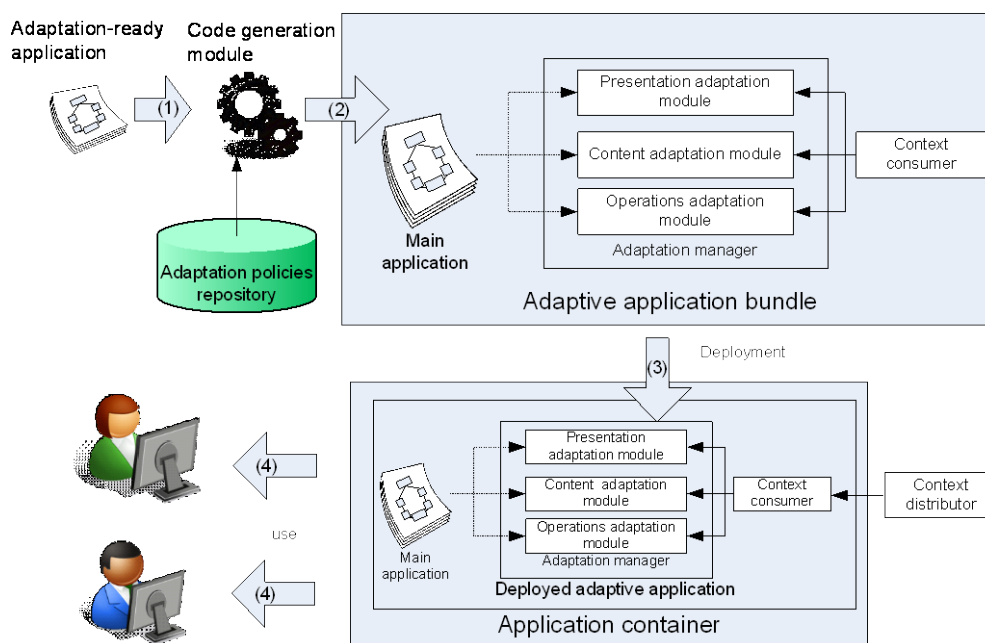
Η δημιουργία των παραπάνω συνιστωσών γίνεται με γένεση κώδικα είτε κατά το χρόνο εκτέλεσης είτε κατά το χρόνο ανάπτυξης της εφαρμογής. Ο *καταναλωτής περιβάλλουσας πληροφορίας* (Context Consumer) τροφοδοτεί τις τρεις προηγούμενες συνιστώσες προσαρμογής με περιβάλλουσα πληροφορία. Το *αποθετήριο πολιτικών προσαρμογής* (Adaptation Policies Repository) διατηρεί τους κανόνες βάσει των οποίων πραγματοποιείται η προσαρμογή. Η γένεση του κώδικα για τη δημιουργία των συνιστωσών προσαρμογής περιεχομένου, λειτουργιών και παρουσίασης γίνεται λαμβάνοντας υπόψη τους εκάστοτε κανόνες προσαρμογής, όπως αυτοί είναι ορισμένοι στο *αποθετήριο πολιτικών προσαρμογής*, και πραγματοποιείται από τη *συνιστώσα παραγωγής κώδικα* (Code Generator Module), κατά τη διαδικασία ανάπτυξης της εφαρμογής. Η επιλογή αυτή είναι σύμφωνη με τη φιλοσοφία του aspect-oriented programming, όπου το «*aspect weaving*» γίνεται πριν την εκτέλεση της εφαρμογής, είτε σε επίπεδο bytecode με εκτέλεση κάποιας εντολής, είτε αυτόματα κατά το φόρτωμα των κλάσεων. Προφανώς, η αλλαγή των κανόνων της

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

πολιτικής προσαρμογής καθιστά αναγκαία την επανάληψη της διαδικασίας ανάπτυξης της εφαρμογής ώστε να ενσωματωθούν οι νέοι κανόνες. Ωστόσο η υποστήριξη για εκ νέου θέση σε λειτουργία (redployment) σε παραγωγικά περιβάλλοντα που έχει ενσωματωθεί σε εξυπηρετές εφαρμογών μπορεί να εγγυηθεί ότι μία τέτοια ενέργεια εκ νέου θέσης σε λειτουργία δεν θα επηρεάσει την εξυπηρέτηση των πελατών που χρησιμοποιούν την εφαρμογή.



Εικόνα 32 - Adaptation Manager

Το σχήμα που παρουσιάζεται, διαχωρίζει περισσότερο τις φάσεις της εφαρμογής - δημιουργία πρόσθετων συνιστωσών, ανάπτυξης και χρήση. Τα διάστικτα βελάκια (από main application προς adaptation manager) αναπαριστούν ροή ελέγχου, ενώ τα πλήρη βελάκια ροή δεδομένων. Ένα θέμα που τίθεται είναι το πώς ο γεννήτορας κώδικα (code generator) καταλαβαίνει ποιες είναι οι μέθοδοι στις οποίες πρέπει να ενσωματώσει προσαρμογή.

Οι πιθανές λύσεις σε αυτό είναι οι εξής:

- να ορίζει ο σχεδιαστής ποιες είναι αυτές οι μέθοδοι είτε γράφοντας μόνος του τα pointcuts και τα advices είτε χρησιμοποιώντας μία απλούστερη σημειογραφία (π.χ. “adaptable content WebPage.shopping_cart_populate()”, που σημαίνει ότι η μέθοδος WebPage.page_shopping_cart_populate(); αποφασίζει ποια είναι τα content items μιας σελίδας [εν προκειμένω της shopping_cart] και έτσι δημιουργεί τις σχετικές μεθόδους.
- να χρησιμοποιείται μία σύμβαση στην ονοματολογία, π.χ. οι μέθοδοι που ξεκινάνε με create_operations, create_content ή create_presentation να είναι υπεύθυνες για τις λειτουργίες, το περιεχόμενο και την παρουσίαση αντίστοιχα.

Δεδομένου του ότι μια σελίδα αποτελείται τελικά από μια συλλογή από ενότητες περιεχομένου, τη διαδικασία της προσαρμογής μιας σελίδας ιστού μπορούμε να την υλοποιήσουμε i) με την προσθήκη ή αφαίρεση των ενοτήτων περιεχομένου ή λειτουργικότητας αυτών, ii) τον επιλεκτικό καθορισμό των δεδομένων (text data,

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

images, videos, specific columns) που θα εμφανιστούν σε αυτά και των χαρακτηριστικών τους και iii) τον κατά περίπτωση προσδιορισμό των ιδιοτήτων παρουσίας τους.

Η υιοθέτηση της αρχής του “διαχωρισμού των ενδιαφερόντων - separation of concerns” για τον διαχωρισμό του πυρήνα λειτουργικότητας της εφαρμογής από τη λογική προσαρμογής, μας οδηγεί στην εισαγωγή μίας νέας ενότητας για κάθε σελίδα, της ενότητας *προσαρμογής* της σελίδας, μέσα στα όρια του οποίου θα πραγματοποιηθεί η διαδικασία της προσαρμογής, δηλαδή την προσαρμογή των *λειτουργιών*, του *περιεχομένου* και της *παρουσίας*. Ουσιαστικά για κάθε σελίδα κατά την ανάπτυξη της εφαρμογής δημιουργείται κατάλληλος κώδικας για να χειριστεί την προσαρμογή των *λειτουργιών*, του *περιεχομένου* και της *παρουσίας* της σελίδας. Ο κώδικας δημιουργείται με «generative programming» σε αυτή τη φάση, ενώ κατά την παραγωγική λειτουργία της εφαρμογής αντλεί στοιχεία της περιβάλλουσας κατάστασης και εκτελεί τις κατάλληλες προσαρμογές, όπως θα δούμε στις επόμενες παραγράφους. Ο κώδικας που δημιουργεί τη σελίδα περιστοιχίζεται από τις κατάλληλες δομές του aspect-oriented programming ώστε να καλείται ο κατάλληλος κώδικας για την προσαρμογή της.

Το *adaptation module* έχει τη δυνατότητα να εκτελέσει δύο μοντέλα προσαρμογής: i) Το *Request/Response* μοντέλο και ii) το *Push* μοντέλο. Στο *Request/Response* μοντέλο η προσαρμογή –αν απαιτείται- γίνεται κατά τη διάρκεια της διαμόρφωσης της απάντησης (response) σε μια αίτηση (request) του πελάτη (client). Στο *Push* μοντέλο η προσαρμογή –αν απαιτείται- γίνεται όταν, με πρωτοβουλία του server, διαμορφώνεται η σελίδα που θα σταλεί στον πελάτη. Το *module προσαρμογής* κάθε σελίδας θα είναι ένα *aspect*, ο κώδικας του οποίου θα υλοποιεί διαδοχικά τη *λειτουργία*, το *περιεχόμενο* και τη *προσαρμογή παρουσίασης*

Συνεχίζουμε με μια πιο λεπτομερή περιγραφή των δομικών συστατικών του *Adaptation Manager*.

4.4.1 Αποθετήριο πολιτικών προσαρμογής

Το *Αποθετήριο πολιτικών προσαρμογής* (*Adaptation Policies Repository*) περιέχει του κανόνες προσαρμογής (adaptation rules) σύμφωνα με τους οποίους λαμβάνει χώρα η διαδικασία της προσαρμογής. (e.g. *If condition x is met then action y is triggered*). Για κάθε σελίδα ιστού της εφαρμογής υπάρχει ένα *αρχείο ρυθμίσεων*, στο *Αποθετήριο πολιτικών προσαρμογής*, το οποίο έχει δύο τμήματα: i) το *Request/Response μέρος* και ii) το *Push μέρος*.

Το μοντέλο που ακολουθούμε για την περιγραφή του *Request/Response* μέρους του αρχείου ρυθμίσεων είναι το *CA (Condition, Action)*. Σύμφωνα με αυτό το μοντέλο θα πρέπει να οριστούν οι συνθήκες που θα πρέπει να ικανοποιούν οι τιμές των μεταβλητών της πληροφορίας περιβάλλουσας κατάστασης και η λειτουργία προσαρμογής η οποία θα εκτελεστεί - όταν ικανοποιούνται αυτές οι συνθήκες.

Το μοντέλο που ακολουθούμε για την περιγραφή του *Push* μέρους του αρχείου ρυθμίσεων είναι το *ECA (Event, Condition, Action)*. Σύμφωνα με αυτό το μοντέλο θα πρέπει να οριστεί το συμβάν (event), το οποίο ενδεχομένως θα πυροδοτήσει μια προσαρμογή, καθώς και οι συνθήκες που θα πρέπει να ικανοποιούν οι τιμές των μεταβλητών της πληροφορίας περιβάλλουσας κατάστασης, καθώς και η λειτουργία

προσαρμογής η οποία θα εκτελεστεί όταν συμβεί το συμβάν και ταυτόχρονα πληρούνται και οι συνθήκες.

<pre> Scenario 1 If (User_Group = g1 and Location = Shopping_Mall1 and intention = shopping) Then <u>Operation Subsection</u> CU1={Add}, CU2={Hide}, CU3={Replace CU4} ... CUn={Add} <u>Content Subsection</u> CU1={Select id, description, price From products Where (price>10) Order By (price desc), Param1=English, Param2=pdf, Paramn=10dpi} CU3=..... ... CUn=..... <u>Presentation Subsection</u> CU1={Text(Color: Black, Size: 8pt), Link(Style: Underline, Color: Black), Image(Size: 500, Color: false) } CU3=..... ... CUn=..... </pre>
<pre> Page "BestOffers" Context_Params (Ctx_UserGroup, Ctx_Location, Ctx_Intention) <u>Operation Subsection</u> CU1={Add} If (Ctx_UserGroup == "g1" Or Ctx_UserGroup == "g2") CU2={Hide} Else CU2={"Add"} Endif If (Ctx_Intention == "CasualShopping") CU3={Replace CU4} Else if (Ctx_Intention == "OfferHunting") CU3={Replace CU5} Endif ... CUn={Add} <u>Content Subsection</u> CU1={DataSource = "Select id, description, price, discount From products, shops, malls Where malls.name = " + Ctx_Location + "and shops.hostedIn = malls.Id and products.SoldBy = shops.id /* select items sold in current mall */ Order By discount desc" Lang=English, ReportType=pdf, Resolution=200dpi} CU3=..... ... CUn=..... <u>Presentation Subsection</u> CU1={Text(Color: Black, Size: 8pt), Link(Style: Underline, Color: Black), Image(Size: 500, Color: false) } CU3=..... ... CUn=..... </pre>

Εικόνα 33 - Request/Response Part of Configuration File

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

Κάθε ένα από τα μέρη *Request/Response* και *Push* του αρχείου ρυθμίσεων περιέχει τόσα τμήματα όσα τα διαφορετικά σενάρια στα πλαίσια των οποίων δύναται να εκτελεστεί η εφαρμογή. Κάθε σενάριο ορίζεται σαν μια λογική πρόταση που περιλαμβάνει στοιχεία περιβάλλουσας κατάστασης και τελεστές *And*, *Or* και *Not*. Ένα τμήμα για κάποιο σενάριο το καλούμε *τμήμα σεναρίου*. Σε κάθε *τμήμα σεναρίου* -εκτός του κανόνα που ορίζει το σενάριο- ορίζονται οι εργασίες που θα εκτελεστούν για: i) την *προσαρμογή του περιεχομένου*, ii) τη *προσαρμογή λειτουργικότητας*, και iii) την *προσαρμογή παρουσίασης* στα *υπό-τμήματα λειτουργικότητας, περιεχομένου και παρουσίασης* αντίστοιχα (εικόνα 32).

Τα σενάρια, οι κανόνες (συνθήκες) και οι εργασίες της προσαρμογής που θα καταχωρηθούν στα εκάστοτε υπό-τμήματα του αρχείου ρυθμίσεων για κάθε σελίδα προκύπτουν κατά τη φάση της ανάλυσης των απαιτήσεων για τη περιβάλλουσα κατάσταση. Κατά τη διαδικασία σύνταξης των διαγραμμάτων UML πληροφορίας περιβάλλουσας κατάστασης, για κάθε επιμέρους λειτουργία της εφαρμογής, ορίζονται τόσο οι καταστάσεις κάτω από τις οποίες θα πραγματοποιηθεί η προσαρμογή όσο και η ίδια λειτουργία/εργασία της προσαρμογής. Οι καταστάσεις κάτω από τις οποίες θα πραγματοποιηθεί η προσαρμογή, αφού αναλυθούν στα συστατικά τους μέρη δηλαδή στα στοιχεία περιβάλλουσας κατάστασης που συμμετέχουν, θα δώσουν τελικά τις συνθήκες δηλαδή τον κανόνα του κάθε σεναρίου. Δηλαδή τα διαγράμματα UML πληροφορίας περιβάλλουσας κατάστασης, τα οποία είναι το προϊόν της ανάλυσης των απαιτήσεων που αφορούν την περιβάλλουσα κατάσταση μιας εφαρμογής κινητού εμπορίου, περιέχουν ουσιαστικά τα σενάρια και τις εργασίες της - προσαρμογής που εμφανίζεται για μια ενότητα περιεχομένου στις *υπό-ενότητες λειτουργίας*, την *υπό-ενότητα περιεχομένου* και την *υπό-ενότητα παρουσίασης* κάθε σεναρίου.

Η κατασκευή του αρχείου ρυθμίσεων για κάθε σελίδα, που ουσιαστικά καθορίζει τις λειτουργίες της προσαρμογής, μπορεί να αποδειχθεί μια εξαιρετικά χρονοβόρα και επίπονη διαδικασία, ειδικά σε περιπτώσεις εφαρμογών με μεγάλο αριθμό σελίδων και μεγάλο πλήθος σεναρίων. Για το λόγο αυτό η κατασκευή και η χρήση ενός βοηθητικού προγράμματος (*utility program*) το οποίο θα διευκολύνει την παραγωγή αυτών των αρχείων ρυθμίσεων κρίνεται εξαιρετικά χρήσιμη. Το συγκεκριμένο βοηθητικό πρόγραμμα θα μπορεί να παρέχει με έναν εύκολο τρόπο τα συστατικά μέρη κάθε σελίδας (μέσω *reflection*), ώστε ο κατασκευαστής του εκάστοτε αρχείου ρυθμίσεων να μπορεί εύκολα να δημιουργήσει τις σχετικές ενότητες του αρχείου ρυθμίσεων (*operation, content, presentation*) για κάθε σελίδα.

Επίσης αυτό το πρόγραμμα μπορεί να εμπλουτιστεί με δυνατότητες αυτοματισμού όπως τον καθορισμό γενικών συνθηκών, (οι οποίες θα αποτελέσουν επιμέρους συνθήκες σεναρίων), με την αντίστοιχη επιθυμητή «προσαρμογή» τους (*operation, content, presentation adaptation*). Όταν οι συγκεκριμένες γενικές συνθήκες αποδίδονται σε συγκεκριμένα σενάρια, τότε η λειτουργία «προσαρμογής» της γενικής συνθήκης θα αποδίδεται στο σενάριο. Σε περιπτώσεις όπου έχουμε σενάρια στα οποία αποδίδονται περισσότερες από μία γενικές συνθήκες και ταυτόχρονα υπάρχει κάποια σύγκρουση (*conflict*) π.χ. μια συνθήκη απαιτεί *font=10pt* και η άλλη *font=12pt*, αυτές οι αντικρουόμενες απαιτήσεις θα πρέπει είτε να επιλύονται αυτόματα (με κάποιο αλγόριθμο, αν αυτό είναι εφικτό) ή να παρουσιάζονται στο χρήστη για να αποφασίσει.

Επίσης το συγκεκριμένο πρόγραμμα θα μπορεί να παρέχει και αυτοματισμό για καθορισμό “προσαρμογής”, ανεξάρτητα από συγκεκριμένη (γενική) συνθήκη και

η οποία “προσαρμογή” θα μπορεί να αποδίδεται σε όλα τα σενάρια ή ακόμα και “προσαρμογή” που θα μπορεί να αποδίδεται σε όλα τα σενάρια όλων των αρχείων ρυθμίσεων όλων των σελίδων (π.χ. presentation properties), ώστε να απαλλάσσεται ο χρήστης από ένα μέρος του φόρτου της εργασίας του. Η αναλυτική λειτουργικότητα ενός τέτοιου βοηθητικού προγράμματος θα πρέπει να καθοριστεί μέσα από περαιτέρω ανάλυση.

4.4.2 The Context Consumer

Ο *Context Consumer* είναι το module εκείνο που θα ανιχνεύσει το τρέχον σενάριο και θα φέρει τη πληροφορία περιβάλλουσας κατάστασης στον *Adaptation Manager*, και ειδικότερα στο *Operation Adaptation Module*, στο *Content Adaptation Module* και στο *Presentation Adaptation Module*, τόσο στην υλοποίηση του *Request/Response* μοντέλου όσο και στην υλοποίηση του *Push* μοντέλου. Επιπρόσθετα στην υλοποίηση του *Push* μοντέλου είναι εκείνο το module το οποίο θα ανιχνεύσει το «συμβάν» και θα πυροδοτήσει την διαδικασία της προσαρμογής (την οποία θα εκτελέσουν το *Operation Adaptation Module*, το *Content Adaptation Module* και το *Presentation Adaptation Module*). Δηλαδή σε αυτήν την περίπτωση η διαδικασία προσαρμογής θα γίνει με πρωτοβουλία του Context Consumer.

Προκειμένου να υλοποιηθεί το *request/response* μοντέλο της προσαρμογής, κάθε φορά που φορτώνεται μία σελίδα ο *Context Consumer* φορτώνει το *Request/Response* μέρος του αρχείου ρυθμίσεων της σελίδας, προσδιορίζει το κάθε σενάριο και το αναλύει για να εξάγει τις απαιτήσεις της σελίδας σε πληροφορία περιβάλλουσας κατάστασης. Κάνοντας κλήση στον Discovery Agency του Context Manager προσδιορίζει τους context wrappers από τους οποίους θα αντλήσει την πληροφορία περιβάλλουσας κατάστασης που απαιτείται και καλεί τις κατάλληλες μεθόδους οι οποίες θα του παρέχουν αυτήν την πληροφορία. Στη συνέχεια προωθεί τις μεταβλητές περιβάλλουσας κατάστασης με τις αντίστοιχες τιμές τους στο *Operation Adaptation Module*, στο *Content Adaptation Module* και στο *Presentation Adaptation Module*, τα οποία καλούνται διαδοχικά προκειμένου να διεκπεραιώσουν τις διαδικασίες προσαρμογής.

Προκειμένου να υλοποιηθεί το *push* μοντέλο της προσαρμογής, είναι απαραίτητο για την κάθε σελίδα ο *Context Consumer* να φορτώνει και να αναλύει το *Push Part* του αρχείου ρυθμίσεων της σελίδας. Για κάθε έναν από αυτούς τους κανόνες ECA απομονώνει τα συμβάντα για τα οποία θέλει να ειδοποιηθεί και αφού συμβουλευθεί το Discovery Agency του Context Manager εγγράφεται για ειδοποίηση (notification) στους κατάλληλους context wrappers. Μόλις ειδοποιηθεί για κάποιο «συμβάν» από κάποιον context wrapper, στη συνέχεια προσδιορίζει κατά πόσο ικανοποιούνται οι συνθήκες του κανόνα ECA. Αυτό γίνεται αφού αντλήσει τις τιμές των μεταβλητών περιβάλλουσας κατάστασης με τον query μηχανισμό που παρέχουν οι context wrappers. Στη συνέχεια προωθεί τις μεταβλητές περιβάλλουσας κατάστασης με τις αντίστοιχες τιμές τους στο *Operation Adaptation Module*, στο *Content Adaptation Module* και στο *Presentation Adaptation Module*, τα οποία καλούνται διαδοχικά προκειμένου να διεκπεραιώσουν τις διαδικασίες προσαρμογής.

4.4.3 Code Generator

Ο *Code Generator* ανακτά από το *Adaptation Rules Repository* διαδοχικά το *Request/Response* και το *Push* μέρος του αρχείου ρυθμίσεων της σελίδας και

σχηματίζει τον κώδικα για το *aspect* που θα αποτελέσει το *module προσαρμογής* της σελίδας.

Αυτός ο κώδικας αρχικά καλεί τον *Context Consumer* ο οποίος θα προσδιορίσει το τρέχον σενάριο και θα τροφοδοτήσει με πληροφορία περιβάλλουσας κατάστασης το *aspect*, - μέσω της δημιουργίας ενός *instance* της κλάσης του *Context Consumer*, του οποίου οι ιδιότητες έχουν τιμές για τα στοιχεία περιβάλλουσας κατάστασης του συγκεκριμένου σεναρίου.

Στη συνέχεια αναλύει τη *λειτουργία*, το *περιεχόμενο*, και την *υπό-ενότητα παρουσίασης* του κάθε σεναρίου, πληροφορίες τις οποίες θα χρησιμοποιήσει για να σχηματίσει τα *pointcuts* του κάθε σεναρίου και τα αντίστοιχά τους *advices*.

Στη συνέχεια για κάθε σενάριο ορίζει το *pointcut*, στο *advice* του οποίου θα γίνει ο καθορισμός της επιλογής των ενοτήτων περιεχομένου. Στη συνέχεια δημιουργεί αυτό το *advice* το οποίο θα εκτελεστεί *πριν* από τη φόρτωση κάθε σελίδας και το οποίο ουσιαστικά αποτελείται από τον κώδικα ο οποίος θα αποκρύψει, θα προσθέσει ή θα αντικαταστήσει κάποιες ενότητες περιεχομένου στον προκαθορισμένο ορισμό της σελίδας (σύμφωνα με τις τιμές που αντλήθηκαν από την *υπό-ενότητα λειτουργίας* του συγκεκριμένου σεναρίου του αρχείου ρυθμίσεων). Επιπρόσθετα, για τις ενότητες περιεχομένου που προστέθηκαν (πέραν αυτών που είχαν οριστεί στη προκαθορισμένη συμπεριφορά της σελίδας) σε αυτό το *advice* θα i) οριστούν οι παράμετροι που θα χρησιμοποιηθούν για να αντλήσουν δεδομένα από τη βάση του πυρήνα εφαρμογής (με βάση τις τιμές που αντλήθηκαν από την *υπό-ενότητα περιεχομένου* του συγκεκριμένου σεναρίου του αρχείου ρυθμίσεων), ii) θα κληθούν οι μέθοδοι για να “γεμίσουν” με δεδομένα τις ενότητες περιεχομένου, iii) οι μέθοδοι που θα προσδιορίσουν τις ιδιότητες παρουσίασης των ενοτήτων περιεχομένου (με βάση τις τιμές που αντλήθηκαν από την *υπό-ενότητα παρουσίασης* του συγκεκριμένου σεναρίου του αρχείου ρυθμίσεων), iv) οι μέθοδοι που θα ορίσουν την επιπρόσθετη λειτουργικότητά τους, καθώς και v) οι μέθοδοι που θα δημιουργήσουν τον *client-side* κώδικα της κάθε ενότητας περιεχομένου (π.χ. *control validations*).

Στη συνέχεια, για τις ενότητες περιεχομένου που αποκρύφθηκαν (είναι κάποια από αυτά που ορίστηκαν στην προκαθορισμένη συμπεριφορά της σελίδας), ορίζει *pointcuts* με τις μεθόδους που i) δημιούργησαν τη διεπαφή της ενότητας περιεχομένου, ii) το τροφοδότησαν με δεδομένα, iii) όρισαν τις ιδιότητες παρουσίασης του, iv) δημιούργησαν την επιπρόσθετη λειτουργικότητά του και v) δημιούργησαν τον *client-side* κώδικα, και ορίζει *advice* το οποίο θα τα αντικαταστήσει (*around advice*) με “*null*” συμπεριφορά (δηλαδή θα τα απενεργοποιήσει). Στη συνέχεια, για τις ενότητες περιεχομένου (τα προκαθορισμένα στη ροή της σελίδας) για τα οποία απαιτείται επιπρόσθετη λειτουργικότητα, ενσωματώνει τον κώδικα για τις μεθόδους κάθε ενότητας περιεχομένου οι οποίες ορίζουν την επιπρόσθετη λειτουργικότητα του (εκτός της τροφοδότησής του με *data* π.χ. *insert*, *update*, *delete row*) και τις μεθόδους για τον κώδικα που θα εκτελεστεί σαν απάντηση στα συμβάντα της συγκεκριμένης ενότητας περιεχομένου.

Τόσο στην περίπτωση της προσθήκης νέων ενοτήτων περιεχομένου στη σελίδα, όσο και στην περίπτωση της προσθήκης λειτουργικότητας σε μια ενότητα περιεχομένου που ήδη ανήκει στη προκαθορισμένη ροή της σελίδας, η προσθήκη μεθόδων στην κλάση που υλοποιεί τη σελίδα θα γίνει χρησιμοποιώντας την δυνατότητα προσθήκης μεθόδων σε μια κλάση του πυρήνα εφαρμογής μέσα σε ένα *advice*, με την τεχνική των *intertype-declarations* του *Aspect Oriented Programming*.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Κώδικας σχετικός με την υλοποίηση της λειτουργικότητας της ενότητας περιεχομένου που θα πρέπει να εκτελεστεί σε συμβάντα της σελίδας, (δηλαδή όχι σε επίπεδο ενότητας περιεχομένου), συμπεριλαμβάνεται με τον καθορισμό *pointcuts* για τις μεθόδους αυτών των συμβάντων της σελίδας και την σύνδεσή τους με ένα *around advice*, όπου θα καλούνται οι σχετικές με τις ενότητες περιεχομένου μέθοδοι.

Η κλήση του *Code Generator*, που ουσιαστικά δημιουργεί τον κώδικα προσαρμογής κάθε σελίδας, μπορεί να γίνει είτε κατά το χρόνο εκτέλεσης της εφαρμογής είτε κατά το χρόνο ανάπτυξης. Προφανώς η γένεση κώδικα κατά το χρόνο εκτέλεσης προσθέτει ευελιξία σε περιπτώσεις που έχουμε συχνές αλλαγές στους κανόνες προσαρμογής όμως ταυτόχρονα αυξάνει και τον χρόνο γένεσης, συλλογής και φόρτωσης του κώδικα. Η γένεση κώδικα κατά τον χρόνο ανάπτυξης ουσιαστικά έχει το μειονέκτημα ότι σε κάθε αλλαγή των κανόνων προσαρμογής θα πρέπει να γίνει εκ νέου η γένεση και συλλογή του κώδικα.

4.4.4 The Adaptation Module

4.4.4.1 To Operation Adaptation Module

Το *Operation Adaptation Module* υλοποιεί τη *προσαρμογή λειτουργίας* και αποτελείται από τμήματα του *aspect*, (δηλαδή του *module προσαρμογής* της σελίδας), που ορίζουν οι ενότητες περιεχομένου που θα παρέχει η σελίδα. Επίσης περιλαμβάνει την κλήση των υπηρεσιών που τροφοδοτούν οι ενότητες περιεχομένου με δεδομένα και πρόσθετη λειτουργικότητα. Η επιλογή των ενοτήτων περιεχομένου ισοδυναμεί τελικά με τη σύνθεση των υπηρεσιών που θα παρέχει η εφαρμογή. Αυτά οι υπηρεσίες επιλέχθηκαν με βάση τους κανόνες προσαρμογής της σελίδας όπως αυτές ορίστηκαν στο *Application Policies Repository*. Τα δεδομένα περιβάλλουσας κατάστασης τροφοδοτήθηκαν σαν παράμετροι εισόδου και θα χρησιμοποιηθούν στον κώδικα των *advices* του *aspect*.

4.4.4.2 To Content Adaptation Module

Το *Content Adaptation Module* υλοποιεί τη *προσαρμογή περιεχομένου* που ακολουθεί τη *προσαρμογή λειτουργίας* και αποτελείται από τα τμήματα του *aspect* που ορίζουν τις παραμέτρους της επιλογής κατάστασης της υπηρεσίας που τροφοδοτεί με δεδομένα τις ενότητες περιεχομένου, και από παραμέτρους των διαφόρων αντικειμένων που θα φέρουν την πληροφορία. Αυτό το *module* τελικά επιλέγει το κατάλληλο περιεχόμενο, τη γλώσσα, το είδος συμπίεσης των δεδομένων κ.λπ., ώστε να ικανοποιούνται οι απαιτήσεις για τα χαρακτηριστικά παρουσίασης του εξυπηρετούμενου, τις δυνατότητες του δικτύου, τις προτιμήσεις του χρήστη κ.ο.κ.

4.4.4.3 To Presentation Adaptation Module

Το *Presentation Adaptation Module* διενεργεί την *προσαρμογή παρουσίασης* η οποία λαμβάνει χώρα μετά τη *προσαρμογή περιεχομένου*. Αποτελείται από τα τμήματα του *aspect* όπου πραγματοποιούνται οι μεταβολές που σχετίζονται με την παρουσίαση των αντικειμένων του *interface* του χρήστη.

4.4.4.4 To Presentation Service

Το *Presentation Service* είναι ένα συγκεκριμένο *instance* μιας συγκεκριμένης λειτουργίας προσαρμογής, το οποίο προωθείται στο *User Interaction Module*.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

4.4.4.5 To User Interaction Module

To User Interaction Module εμφανίζει τα δεδομένα και τις διάφορες επιλογές λειτουργικότητας στον τελικό χρήστη. Η προσαρμοσμένη σελίδα είναι το αποτέλεσμα της διαδικασίας προσαρμογής, αφού ελήφθησαν υπόψη οι ιδιαιτερότητες του χρήστη καθώς και οι μεταβαλλόμενες συνθήκες του περιβάλλοντος λειτουργίας της εφαρμογής.

Κεφάλαιο 5 Ένα πλαίσιο για τον προγραμματισμό εύρωστων εφαρμογών με επίγνωση περιβάλλουσας κατάστασης

Η επίγνωση της περιβάλλουσας κατάστασης αποτελεί μια βασική πτυχή των εφαρμογών διάχυτου υπολογισμού, και χαρακτηρίζει την ικανότητά τους να προσαρμόζονται και να εκτελούν καθήκοντα κι εργασίες με βάση τις διαφορετικές συνθήκες του περιβάλλοντος. Με τον όρο “περιβάλλουσα κατάσταση” αναφερόμαστε σ’ ένα σύνολο συνθηκών φυσικού ή εικονικού κόσμου που μπορεί να χρησιμοποιηθεί από μια εφαρμογή με σκοπό την δυναμική προσαρμογή της λειτουργικότητάς της. Τυπικά παραδείγματα της περιβάλλουσας κατάστασης είναι η τοποθεσία στην οποία βρίσκεται ένα άτομο, η απόσταση στην οποία βρίσκεται ένα άτομο από ένα άλλο άτομο (ή μια συσκευή ή ένα αντικείμενο), οι συσκευές που χρησιμοποιούνται από ένα άτομο, η δραστηριότητα με την οποία απασχολείται ένα άτομο, κ.λπ. Το υπολογιστικό περιβάλλον στο οποίο αναπτύσσονται οι εφαρμογές επίγνωσης της περιβάλλουσας κατάστασης ονομάζεται ενεργός χώρος (active space) [29]. Ένας τέτοιος χώρος μπορεί να εκτείνεται σε πολλές διαφορετικές φυσικές περιοχές και είναι δυνατόν να παρέχει υπηρεσίες υποδομής που απαιτούνται για την υλοποίηση και τη λειτουργία εφαρμογών επίγνωσης περιβάλλουσας κατάστασης. Εδώ περιλαμβάνονται υπηρεσίες για την ανίχνευση περιβάλλουσας κατάστασης αθροίσματος δεδομένων από αισθητήρες, παραγωγής γεγονότων περιβάλλουσας κατάστασης, ανακάλυψης πόρων υπηρεσιών, ονομασίας υπηρεσιών, καθώς και υπηρεσιών για τον έλεγχο και την άδεια πρόσβασης [29], [30].

Μέχρι σήμερα, αρκετοί ερευνητές έχουν δημιουργήσει διάφορες εφαρμογές επίγνωσης περιβάλλουσας κατάστασης, όπως είναι οι οδηγοί για τουρίστες [31], τα έξυπνα δωμάτια [32], εφαρμογές για μουσεία [33], καθώς και εφαρμογές υποβοηθούμενης διαβίωσης [34]. Οι ερευνητές έχουν αναπτύξει πλαίσια εργασίας για τέτοιου είδους εφαρμογές [23], [30], [35]. Αυτά τα πλαίσια επιτρέπουν το σχεδιασμό εφαρμογών επίγνωσης περιβάλλουσας κατάστασης που υποστηρίζουν διάφορα χαρακτηριστικά, όπως την ανακάλυψη δυναμικών υπηρεσιών με βάση την περιβάλλουσα κατάσταση, τον έλεγχο πρόσβασης και το συντονισμό πολλαπλών χρηστών με βάση την περιβάλλουσα κατάσταση, τις ενέργειες που πυροδοτούνται από την περιβάλλουσα κατάσταση, καθώς και την πρόσβαση σε πληροφορίες με βάση την περιβάλλουσα κατάσταση.

Στην προσέγγιση που παρουσιάζεται στην παρούσα ενότητα, ο σχεδιασμός μιας εφαρμογής με επίγνωση της περιβάλλουσας κατάστασης προσδιορίζεται με χρήση ενός προγραμματιστικού μοντέλου με καθορισμένο τομέα, το οποίο εκφράζεται σε γλώσσα XML. Η αρχική σχεδίαση αυτού του προγραμματιστικού πλαισίου εργασίας παρουσιάστηκε στην εργασία [35]. Αυτό το πλαίσιο δεν περιελάμβανε μηχανισμούς για ανάκαμψη από προγραμματιστικά λάθη. Στο μοντέλο σχεδιασμού που υποστηριζόταν από αυτό το πλαίσιο, μια εφαρμογή με επίγνωση της περιβάλλουσας κατάστασης, προγραμματιζόταν με χρήση της έννοιας της *δραστηριότητας* (activity), όπου μια δραστηριότητα ορίζει ένα κοινό χώρο αντικειμένων κι ένα σύνολο από ρόλους του χρήστη. Επιπλέον, μια δραστηριότητα μπορεί να επεκταθεί σε πολλές διαφορετικές φυσικές περιοχές και μπορεί να περιλαμβάνει πολλαπλούς χρήστες που επιτελούν σε συνεργασία διάφορες εργασίες και καθήκοντα. Διάφοροι πόροι και υπηρεσίες που απαιτούνται από την εφαρμογή,

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

προσπελαύνονται μέσω αντικειμένων (objects) που ορίζονται στη δραστηριότητα. Ένα αντικείμενο συνδέεται με διαφορετικές υπηρεσίες και πόρους υπό διαφορετικές συνθήκες περιβάλλουσας κατάστασης. Επίσης, μια δραστηριότητα παρέχει ρόλους (roles) για την αναπαράσταση των δικαιωμάτων των χρηστών ώστε να εκτελούν διάφορες εργασίες της εφαρμογής. Οι ρόλοι επιπρόσθετα, αποτελούν το κύριο μηχανισμό για τον έλεγχο πρόσβασης σε μια δραστηριότητα. Ο μηχανισμός λειτουργίας (operation) παρέχεται με το ρόλο, μέσω του οποίου οι χρήστες μπορούν να εκτελούν εργασίες της εφαρμογής. Αποτελεί επίσης το μηχανισμό μέσω του οποίου οι χρήστες μπορούν να έχουν πρόσβαση σε υπηρεσίες ενεργού χώρου μέσα από μια αλληλεπιδραστική συνεδρία.

5.1 Στόχοι της προσέγγισης και αδυναμίες που προσπαθεί να καλύψει

5.1.1 Αδυναμίες που προσπαθεί να καλύψει η προσέγγιση

Διάφορα ζητήματα προκύπτουν λόγω της δυναμικής φύσης αυτών των εφαρμογών και επίσης λόγω της δυναμικής φύσης των περιβαλλόντων στα οποία τέτοιες εφαρμογές εκτελούνται. Συγκεκριμένα, προκύπτουν διάφορα ζητήματα ευρωστίας όπως:

- *Αποτυχίες ανακάλυψης και αναδιαμόρφωσης υπηρεσιών (Service discovery and reconfiguration failures)*: Μια εφαρμογή μπορεί να μη λειτουργήσει σωστά λόγω των αποτυχιών στην εξεύρεση των απαιτούμενων πόρων και υπηρεσιών κατά τη διάρκεια των αναδιαμορφώσεων που πραγματοποιούνται από αλλαγές στην περιβάλλουσα κατάσταση. Αυτό μπορεί να συμβεί εάν αποτύχει η εύρεση μιας υπηρεσίας ή εάν δεν υπάρχει καμία εγγεγραμμένη υπηρεσία που να ταιριάζει με την υπηρεσία που αναζητείται.
- *Αποτυχίες σύνδεσης σε επίπεδο υπηρεσίας (Service-level binding failures)*: Η σύνδεση μιας εφαρμογής με μια υπηρεσία ενεργού χώρου μπορεί να αποτύχει, εμποδίζοντας κατά συνέπεια οποιοσδήποτε προσπάθειες επικοινωνίας χρηστών με τη συγκεκριμένη υπηρεσία. Η αιτία για αυτές τις αποτυχίες σύνδεσης μπορεί να αποτελούν τυχόν δικτυακά προβλήματα, καταρρεύσεις υπηρεσιών ή ανακλήσεις εξουσιοδοτήσεων πρόσβασης από τις υπηρεσίες.
- *Εξαιρέσεις σε επίπεδο υπηρεσίας (Service-level exceptions)*: Οι αλληλεπιδραστικές συνεδρίες των χρηστών μπορεί να παρουσιάσουν εξαιρέσεις (exceptions) που εγείρονται από την υπηρεσία με την οποία είναι υπό εξέλιξη η επικοινωνία. Τέτοιες εξαιρέσεις μπορεί να προκύψουν λόγω ασυμβατότητας πρωτοκόλλων για πρόσβαση σε πόρους, είτε λόγω ανεπαρκών προνομίων ασφάλειας της εφαρμογής, ή λόγω αποτυχιών στην εκτέλεση των λειτουργιών υπηρεσιών.
- *Ακυρώσεις περιβάλλουσας πληροφορίας (Context invalidations)*: Μια εφαρμογή που απαιτεί να αληθεύει μία συνθήκη κατά τη διάρκεια της εκτέλεσης ορισμένων εργασιών, η οποία συνθήκη περιλαμβάνει στοιχεία περιβάλλουσας πληροφορίας, είναι επιρρεπής σε αποτυχίες όταν παραβιάζεται αυτή η συνθήκη. Αυτό το πρόβλημα αναφέρεται κι ως *πρόβλημα ακύρωσης περιβάλλουσας πληροφορίας (context invalidation problem)*. Για τη σωστή

εκτέλεση τέτοιων ενεργειών, απαιτούνται μηχανισμοί ώστε να εντοπίζονται περιπτώσεις όπου έχουμε ακυρώσεις περιβάλλουσας πληροφορίας.

- *Σφάλματα κατά τη συνεργασία χρηστών:* Είναι προβλήματα που δημιουργούνται όταν κάποιοι χρήστες, που εργάζονται για ένα κοινό στόχο, μπορεί να αποτύχουν να εκτελέσουν ορισμένες υποχρεωτικές εργασίες. Αυτή η αποτυχία μπορεί να επηρεάσει τους υπόλοιπους χρήστες που συνεργάζονται για την πραγματοποίηση του κοινού αυτού στόχου.
- *Σφάλματα περιβάλλοντος:* Αυτά μπορεί να προκύψουν από γεγονότα που συμβαίνουν στον πραγματικό κόσμο και επηρεάζουν την εφαρμογή. Αυτά τα γεγονότα μπορεί να επηρεάσουν τις υποθέσεις της εφαρμογής για την κατάσταση του εξωτερικού κόσμου.

5.1.2 Στόχοι της προσέγγισης

Στόχος της συγκεκριμένης προσέγγισης είναι η δημιουργία ενός προγραμματιστικού πλαισίου εργασίας για την ανάπτυξη εφαρμογών με επίγνωση της περιβάλλουσας κατάστασης, το οποίο εισάγει ένα μοντέλο ανάκαμψης και επιτρέπει τον ορισμό ενεργειών ανάκαμψης σε επίπεδο εφαρμογής για την αντιμετώπιση αποτυχιών. Αυτές οι ενέργειες ανάκαμψης εκμεταλλεύονται τον σχεδιασμό του πλαισίου που εισάγει τις έννοιες της δραστηριότητας, των ρόλων, του αντικειμένου και των αλληλεπιδράσεων ώστε να ανακαλύψει άμεσα κάποια σφάλματα και μέσω αυτού του διαχωρισμού να αντιληφθεί σε ποιο επίπεδο δημιουργείται το πρόβλημα και να το αντιμετωπίσει χρησιμοποιώντας το κατάλληλο μοντέλο ανάκαμψης.

5.1.3 Καινοτομίες αυτής της προσέγγισης

Το κύριο καινοτόμο χαρακτηριστικό αυτού του μοντέλου είναι η ενσωμάτωση του χειρισμού γεγονότων σε επίπεδο αντικειμένου με ένα χειριστή εξαιρέσεων σε επίπεδο ρόλου, ώστε να δημιουργηθούν εύρωστες εφαρμογές με επίγνωση της περιβάλλουσας κατάστασης. Επίσης μια ακόμα καινοτομία είναι η δημιουργία ενός νέου μηχανισμού με τη μορφή διεπαφής εξαιρέσεων για τους ρόλους που παρέχει τη δυνατότητα στους χρήστες να χειριστούν εξαιρέσεις.

5.2 Βασικές έννοιες της προσέγγισης

Σκοπός της παρούσας ενότητας είναι να παρουσιαστούν και να αναλυθούν οι κύριες έννοιες που παρουσιάστηκαν στην εισαγωγή όπως είναι αυτές της δραστηριότητας, των ρόλων, των αντιδράσεων, των γεγονότων καθώς και του ενδιαμέσου λογισμικού. Ακόμα σ' αυτήν την ενότητα θα αναφερθούν εκτενώς τα προβλήματα ευρωστίας που προκύπτουν ώστε να γίνει αντιληπτό γιατί είναι απαραίτητο το μοντέλο ανάκαμψης που θα παρουσιαστεί στο ίδιο κεφάλαιο.

Το προγραμματιστικό πλαίσιο εργασίας που παρουσιάζεται, βασίζεται το υπόδειγμα του generative programming. Σ' αυτό το προγραμματιστικό πλαίσιο εργασίας, οι εφαρμογές με επίγνωση της περιβάλλουσας κατάστασης έχουν σχεδιαστεί με χρήση ενός μοντέλου σχεδιασμού. Αυτό το μοντέλο σχεδιασμού παρέχει τρόπους και μεθόδους για την ενσωμάτωση των χρηστών, των στοιχείων της εφαρμογής, καθώς και των υπηρεσιών υποδομής για την κατασκευή του περιβάλλοντος εκτέλεσης μιας εφαρμογής με επίγνωση της περιβάλλουσας κατάστασης.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

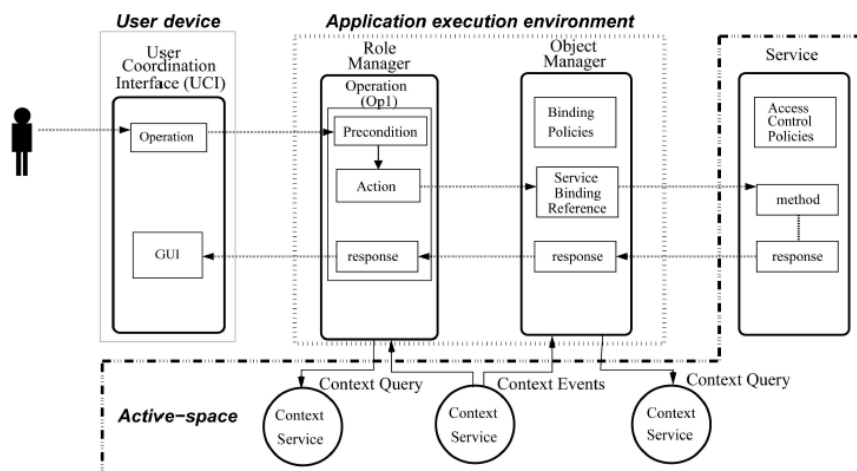
Μεταπτυχιακή εργασία

Φούντας Γεώργιος

Μια εφαρμογή με επίγνωση της περιβάλλουσας κατάστασης προγραμματίζεται με χρήση της έννοιας της *δραστηριότητας*. Μια δραστηριότητα καθορίζει ένα χώρο ονομάτων για τα αντικείμενα (objects), τους ρόλους (roles) και τις αντιδράσεις (reactions). Το αντικείμενο παρέχεται στη δραστηριότητα για την πρόσβαση σε διάφορους πόρους και υπηρεσίες που απαιτούνται από την εφαρμογή. Αυτό περιλαμβάνει υπηρεσίες περιβάλλουσας κατάστασης από τις οποίες μπορεί μια εφαρμογή να αποκτήσει τις απαραίτητες πληροφορίες περιβάλλουσας κατάστασης. Ένα αντικείμενο μπορεί να προγραμματιστεί ώστε να συνδεθεί με διάφορες υπηρεσίες ενεργού χώρου κάτω από διαφορετικές συνθήκες περιβάλλουσας κατάστασης. Τα αντικείμενα που ορίζονται στο χώρο των ονομάτων μιας δραστηριότητας διαμοιράζονται από όλους τους ρόλους που ορίζονται στη συγκεκριμένη δραστηριότητα. Κάθε ρόλος ορίζει επίσης ένα χώρο ονομάτων για τα αντικείμενα και τις λειτουργίες του ρόλου. Αντικείμενα που ορίζονται στα πλαίσια ενός ρόλου είναι ιδιωτικά, όσον αφορά αυτό τον ρόλο. Έτσι, ένα ξεχωριστό στιγμιότυπο ενός τέτοιου αντικειμένου δημιουργείται για κάθε μέλος ενός ρόλου. Τέτοιου είδους αντικείμενα απαιτούνται δεδομένου του ότι, σε μια εφαρμογή με πολλούς χρήστες, μπορεί να θέλουμε διαφορετικά μέλη ενός ρόλου να έχουν πρόσβαση σε διαφορετικά στιγμιότυπα ενός είδους υπηρεσίας με βάση την ατομική τους περιβάλλουσα κατάσταση.

5.2.1 Ρόλοι

Ένας ρόλος περιέχει μία ή περισσότερες λειτουργίες. Μια λειτουργία ρόλου αναπαριστά μια εργασία/καθήκον που καλείται ρητά από τα μέλη του ρόλου. Η εικόνα 34 παρουσιάζει το μοντέλο αλληλεπίδρασης μεταξύ χρηστών και υπηρεσιών με βάση τους ρόλους στο συγκεκριμένο πλαίσιο εργασίας. Οι χρήστες εκτελούν τις λειτουργίες του ρόλου μέσω μιας διεπαφής χρήστη (User Coordination Interface, UCI). Μια λειτουργία ρόλου αποτελείται από δύο μέρη: μία *προϋπόθεση* (precondition) και μια *ακολουθία* από μία ή περισσότερες ενέργειες. Ο διαχειριστής του ρόλου αποτιμά την προϋπόθεση που σχετίζεται με την καλούμενη λειτουργία. Εάν η προϋπόθεση αληθεύει, τότε ο διαχειριστής του ρόλου εκτελεί τις ενέργειες που προβλέπονται για την εν λόγω λειτουργία. Μια ενέργεια μπορεί να περιλαμβάνει την κλήση μιας μεθόδου σ' ένα αντικείμενο. Ο διαχειριστής αντικειμένου καλεί την εν λόγω μέθοδο της υπηρεσίας με την οποία συνδέεται αυτή τη στιγμή. Ο διαχειριστής ρόλου μπορεί επίσης να εξετάζει τις υπηρεσίες παροχής περιβάλλουσας κατάστασης που είναι διαθέσιμες στον ενεργό χώρο για την αποτίμηση των προϋποθέσεων που χρησιμοποιούν πληροφορίες περιβάλλουσας κατάστασης.

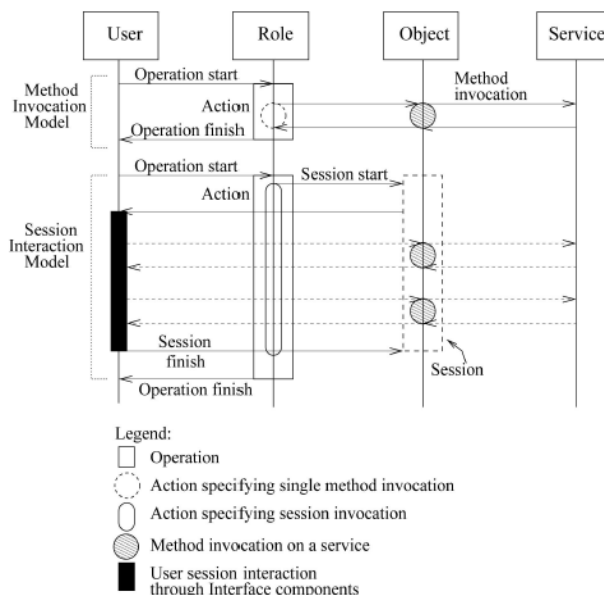


Εικόνα 34 - Μοντέλο αλληλεπίδρασης μεταξύ χρηστών και υπηρεσιών

Όπως φαίνεται στην εικόνα 2, υποστηρίζονται δύο μοντέλα για την εκτέλεση των ενεργειών στο πλαίσιο μιας λειτουργίας ενός ρόλου:

1. Στο μοντέλο κλήσης μεθόδου (method invocation model), μια ενέργεια περιλαμβάνει μόνο μία κλήση μιας μεθόδου σ' ένα αντικείμενο.
2. Στο μοντέλο αλληλεπίδρασης συνεδρίας (session interaction model), περιλαμβάνεται η κλήση των αποτελεσμάτων μιας ενέργειας στην έναρξη μιας αλληλεπιδραστικής συνεδρίας μεταξύ του χρήστη κι ενός αντικειμένου. Μέσα από μια τέτοια συνεδρία, ο χρήστης μπορεί να εκτελέσει μια οποιαδήποτε ακολουθία καθορισμένων μεθόδων του αντικειμένου.

Η εικόνα 35 δείχνει την κλήση δύο λειτουργιών ρόλου. Η πρώτη λειτουργία αφορά κλήση μίας μεθόδου. Η δεύτερη λειτουργία οδηγεί σε μια αλληλεπιδραστική συνεδρία μεταξύ του στοιχείου διεπαφής του χρήστη και του αντικειμένου.



Εικόνα 35 - Μοντέλα αλληλεπίδρασης

Είναι πιθανό ορισμένες υπηρεσίες του περιβάλλοντος να μπορούν να υποστηρίξουν τις μεθόδους συναλλαγής (transactional methods). Για παράδειγμα, μία

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

υπηρεσία που βασίζεται στη γλώσσα Java, μπορεί να υλοποιεί τη διεπαφή TransactionManager που ορίζεται ως μέρος του Java Transaction API. Είναι πιθανό επίσης, μια λειτουργία να περιλαμβάνει μια σειρά ενεργειών που αφορούν την εκτέλεση τέτοιων μεθόδων συναλλαγής σε διαφορετικά αντικείμενα. Ακόμα κι αν κάποιες από τις μεθόδους υπηρεσιών μπορούν να παρέχουν εγγυήσεις συναλλαγών, εντούτοις, δεν αντιμετωπίζεται η εκτέλεση μιας λειτουργίας ρόλου ως μια συναλλαγή. Αυτό συμβαίνει για τους εξής δύο λόγους:

1. Όταν κάποια ενέργεια ορίζεται να εκτελείται όταν αποτύχει μια λειτουργία, η εκτέλεση της ενέργειας μπορεί να μην είναι δυνατόν να επαναφέρει εν συνόλω τις επιπτώσεις όλων των προηγούμενων επιτυχών ενεργειών. Για παράδειγμα, μπορεί να μην είναι δυνατή η επαναφορά των ενεργειών που επηρεάζουν τις συνθήκες του περιβάλλοντος. Αυτό είναι παρόμοιο με τις καταστάσεις που συμβαίνουν σε ορισμένα συστήματα ροής εργασίας.
2. Οι εν μέρει εκτελεσθείσες λειτουργίες μπορούν να φανούν χρήσιμες σε ορισμένες περιπτώσεις.

Οι ενέργειες που πυροδοτούνται από την περιβάλλουσα κατάσταση προγραμματίζονται με χρήση του μηχανισμού αντίδρασης (reaction) που παρέχεται από το προγραμματιστικό μοντέλο. Ο μηχανισμός αυτός ακολουθεί το μοντέλο εκτέλεσης Event-Condition-Action. Μια αντίδραση είναι παρόμοια με τη λειτουργία ενός ρόλου (αποτελείται δηλαδή από μια προϋπόθεση και μια ή περισσότερες ενέργειες), με τη μόνη διαφορά να έγκειται στο ότι η εκτέλεσή της ενεργοποιείται από ένα συμβάν κι όχι από ένα χρήστη. Εξαιτίας αυτού του γεγονότος, μια αντίδραση υποστηρίζει μόνο το μοντέλο κλήσης μεθόδου, αλλά όχι το μοντέλο αλληλεπίδρασης συνεδρίας.

Ο μηχανισμός αντίδρασης χρησιμοποιείται για δύο σκοπούς στο προτεινόμενο προγραμματιστικό πλαίσιο εργασίας. Πρώτον, χρησιμοποιείται για την αυτοματοποιημένη εκτέλεση καθηκόντων κι εργασιών που δεν περιλαμβάνουν συμμετοχή των χρηστών. Δεύτερον, χρησιμοποιείται για τον προγραμματισμό των ενεργειών πυροδότησης της περιβάλλουσας κατάστασης ενός αντικειμένου.

5.2.2 Middleware (ενδιάμεσο λογισμικό)

Το πλαίσιο που παρουσιάζουμε περιλαμβάνει μία συνιστώσα ενδιάμεσου λογισμικού (middleware), η οποία παρέχει τρία γενικά στοιχεία-διαχειριστές: έναν διαχειριστή δραστηριοτήτων (activity manager), έναν διαχειριστή ρόλων (role manager) κι έναν διαχειριστή αντικειμένων (object manager). Το περιβάλλον εκτέλεσης μιας εφαρμογής κατασκευάζεται εξειδικεύοντας αυτούς τους διαχειριστές με βάση τις προδιαγραφές μιας συγκεκριμένης δραστηριότητας. Για κάθε αντικείμενο δημιουργείται ένας διαχειριστής αντικειμένου. Κάθε τέτοιος διαχειριστής διατηρεί μια αναφορά στην υπηρεσία με την οποία συνδέεται κάθε χρονική στιγμή. Οι αντιδράσεις που πυροδοτούνται από την περιβάλλουσα κατάσταση και που έχουν οριστεί για δυναμική σύνδεση ενός αντικειμένου, εκτελούνται από το διαχειριστή του αντικειμένου. Κάθε διαχειριστής καταγράφει συμβάντα περιβάλλουσας κατάστασης που απαιτούνται από τις διαφορετικές πολιτικές περιβάλλουσας κατάστασης που με τη σειρά τους καθορίζονται για το συγκεκριμένο διαχειριστή.

5.2.3 Συμβάν περιβάλλουσας κατάστασης

Ένα συμβάν περιβάλλουσας κατάστασης μπορεί να καταγραφεί από τους διαχειριστές αντικειμένου, τους διαχειριστές ρόλου, και τις αντιδράσεις στο επίπεδο της δραστηριότητας. Ένα συμβάν αποστέλλεται πρώτα σε διάφορους διαχειριστές αντικειμένου, κατόπιν στους διαχειριστές ρόλου, και τέλος σε όλες τις αντιδράσεις στο επίπεδο της δραστηριότητας. Η λογική πίσω από αυτή την ακολουθία είναι ότι οι ενέργειες που ορίζονται στις λειτουργίες του ρόλου ή οι αντιδράσεις στο επίπεδο της δραστηριότητας, μπορεί να περιλαμβάνουν κλήσεις μεθόδων σε ορισμένα αντικείμενα, και είναι σημαντικό αυτές οι κλήσεις μεθόδων να πραγματοποιούνται μόνο μετά τη σύνδεση των αντικειμένων με κατάλληλους πόρους/υπηρεσίες σύμφωνα με την τρέχουσα περιβάλλουσα κατάσταση.

Πολλαπλοί διαχειριστές αντικειμένων μπορούν να εμπλέκονται σ' ένα συμβάν περιβάλλουσας κατάστασης. Η σειρά με την οποία αποστέλλεται το συμβάν στους διάφορους διαχειριστές αντικειμένου μπορεί επίσης να είναι ζωτικής σημασίας για τη σωστή συμπεριφορά της εφαρμογής [36]. Για παράδειγμα, είναι σημαντικό η σύνδεση των αντικειμένων που αναφέρονται σε υπηρεσίες περιβάλλουσας κατάστασης να γίνεται πριν από τη σύνδεση άλλων αντικειμένων, ειδικά, μπορεί να οδηγηθούμε στη χρήση εσφαλμένων πληροφοριών περιβάλλουσας κατάστασης από τα άλλα αντικείμενα. Στο προγραμματιστικό μοντέλο παρέχεται μια δομή μέσω της οποίας μπορεί να προσδιοριστεί μια σειρά σύνδεσης για ένα σύνολο αντικειμένων. Επίσης, σε ορισμένες περιπτώσεις, η ταυτόχρονη επεξεργασία των συμβάντων περιβάλλουσας κατάστασης από διαφορετικές αντιδράσεις, θα μπορούσε να οδηγήσει σε λανθασμένες συμπεριφορές της εφαρμογής [36]. Για να αποφευχθούν τέτοιου είδους περιπτώσεις, το ενδιάμεσο λογισμικό εξασφαλίζει την ατομικότητα στο χειρισμό των συμβάντων περιβάλλουσας κατάστασης απαιτώντας από τους διαχειριστές αντικειμένου να χειριστούν διαδοχικά τα συμβάντα περιβάλλουσας κατάστασης.

5.2.4 Υπηρεσίες περιβάλλουσας κατάστασης

Οι υπηρεσίες περιβάλλουσας κατάστασης αποστέλλουν συμβάντα περιβάλλουσας κατάστασης σε μια δραστηριότητα με τη σειρά με την οποία αυτά ανιχνεύονται. Επιπλέον, αποτελεί ευθύνη της υπηρεσίας περιβάλλουσας κατάστασης η εξασφάλιση της παράδοσης των συμβάντων περιβάλλουσας κατάστασης, εφόσον απαιτείται από μια εφαρμογή. Για τον λόγο αυτό, οι υπηρεσίες περιβάλλουσας κατάστασης μπορεί να χρειαστεί να επικοινωνήσουν μεταξύ τους. Για παράδειγμα, αν ένας χρήστης ξεκινά από ένα δωμάτιο και μπαίνει σε ένα άλλο δωμάτιο, τότε είναι πιθανό το συμβάν της άφιξης να ανιχνευτεί και να εκτελεστεί πριν από το συμβάν της αναχώρησης. Αυτό όμως θα συμβεί εφόσον τα δυο αυτά συμβάντα ανιχνευτούν από δύο διαφορετικές υπηρεσίες περιβάλλουσας κατάστασης που δεν επικοινωνούν μεταξύ τους.

Για την υποστήριξη της ανακάλυψης των πόρων του περιβάλλοντος και των υπηρεσιών για τη δυναμική σύνδεση, αναπτύχθηκε στην παρουσιαζόμενη προσέγγιση μια υπηρεσία ανακάλυψης πόρων γι' αυτό το πλαίσιο εργασίας. Μία ή περισσότερες από αυτές τις υπηρεσίες μπορεί να αναπτυχθούν σ' ένα ενεργό χώρο. Οι εφαρμογές ερωτούν την υπηρεσία ανακάλυψης ώστε να εντοπισθεί η απαιτούμενη υπηρεσία που ταιριάζει με ένα σύνολο κριτηρίων ανακάλυψης. Για την περιγραφή μιας υπηρεσίας, αναπτύχθηκε ένα σχήμα XML που ονομάζεται Resource Description Definition

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

(RDD). Το RDD καθορίζει τρία στοιχεία: μια λίστα από ζεύγη χαρακτηριστικών για μια υπηρεσία, διεπαφές υπηρεσιών, καθώς και συμβάντα που μπορούν να δημιουργηθούν από μια υπηρεσία και τα οποία μπορεί να καταγραφούν από μια εφαρμογή με επίγνωση της περιβάλλουσας κατάστασης. Μερικά από τα χαρακτηριστικά ενός σχήματος RDD αντιμετωπίζονται ως χαρακτηριστικά-παράμετροι. Οι τιμές τους καθορίζονται κατά το χρόνο εκτέλεσης, με βάση την περιβάλλουσα κατάσταση.

Για την επαλήθευση της λειτουργικότητας του σχήματος που παρουσιάζεται, αναπτύχθηκαν και υλοποιήθηκαν διάφορες υπηρεσίες περιβάλλουσας κατάστασης για τη συγκέντρωση των δεδομένων από αισθητήρες και την παροχή πληροφοριών περιβάλλουσας κατάστασης. Για παράδειγμα, αναπτύχθηκε στο σύστημα, μια υπηρεσία τοποθεσίας για τη διατήρηση της πληροφορίας της εσωτερικής τοποθεσίας κάθε χρήστη. Η υπηρεσία αυτή ανιχνεύει και παράγει γεγονότα που αναπαριστούν αφίξεις ή αναχωρήσεις του χρήστη, προς ή από ένα δωμάτιο. Υποστηρίζει, επίσης, πληροφορίες κατάστασης διαφόρων δωματίων, όπως τον αριθμό των χρηστών του βρίσκονται σ' ένα δωμάτιο.

5.2.5 Θέματα ευρωστίας

Τα θέματα ευρωστίας κατατάσσονται σε δύο κατηγορίες. Η πρώτη περιλαμβάνει τα θέματα που βρίσκονται σε επίπεδο συστήματος και η δεύτερη αυτά που βρίσκονται σε επίπεδο εφαρμογής. Τα θέματα επιπέδου συστήματος προκύπτουν λόγω της ταυτόχρονης επεξεργασίας γεγονότων περιβάλλουσας κατάστασης. Σε αυτήν την υποενότητα θα μελετηθούν θέματα ευρωστίας που αφορούν στην ανακάλυψη και σύνδεση των αντικειμένων με τις υπηρεσίες ενεργού χώρου, τη διακοπή σύνδεσης ενός αντικειμένου με τις υπηρεσίες ενεργού χώρου, τις εξαιρέσεις υπηρεσιών καθώς επίσης και με ακυρώσεις της περιβάλλουσας πληροφορίας.

5.2.5.1 Αποτυχίες στις αναδιαμορφώσεις που συμβαίνουν λόγω της περιβάλλουσας κατάστασης

Η οδηγούμενη από την περιβάλλουσα κατάσταση αναδιαμόρφωση απαιτείται για την υποστήριξη της ικανότητας μίας εφαρμογής με επίγνωση περιβάλλουσας κατάστασης να συνδεθεί με διαφορετικές υπηρεσίες ενεργού χώρου κάτω από διαφορετικές συνθήκες περιβάλλουσας κατάστασης. Γι' αυτό τον λόγο, μια εφαρμογή μπορεί να χρειαστεί να ανακαλύψει και να συνδεθεί δυναμικά με τον απαραίτητο τύπο υπηρεσιών στον ενεργό χώρο.

Πολλά θέματα ευρωστίας προκύπτουν από αυτή την αναγκαιότητα. Πρώτον, οι ενέργειες ανακάλυψης μπορεί να αποτύχουν, αν το απαιτούμενο είδος υπηρεσίας δεν είναι πια διαθέσιμο στον ενεργό χώρο. Για παράδειγμα, όπως θα δείξουμε και στο παράδειγμα της εφαρμογής αναπαραγωγής ήχου, απαιτείται σε κάθε δωμάτιο που εισέρχεται ο χρήστης να υπάρχει συσκευή αναπαραγωγής ήχου. Έτσι, εφόσον δεν υπάρχει σ' ένα δωμάτιο μια τέτοια συσκευή, τότε αποτυγχάνει η ανακάλυψη. Δεύτερον μια εφαρμογή μπορεί να μην είναι στη σωστή κατάσταση για εκτέλεση ενεργειών αναδιαμόρφωσης. Για παράδειγμα, ορισμένες εφαρμογές μπορεί να απαιτήσουν ότι ορισμένες ενέργειες σύνδεσης και ανακάλυψης υπηρεσιών, σε επίπεδο αντικειμένου, δεν θα πρέπει να εκτελεστούν, εφόσον είναι σε εξέλιξη μια αλληλεπιδραστική συνεδρία ενός χρήστη μ' ένα αντικείμενο.

Για να αντιμετωπιστούν αυτά τα θέματα, χρειάζονται οι ακόλουθοι μηχανισμοί στο μοντέλο ανάκαμψης:

1. Το μοντέλο χρειάζεται να υποστηρίξει τη δυνατότητα ενεργοποίησης ενός συνόλου εναλλακτικών εργασιών, οι οποίες δεν εξαρτώνται από τις υπηρεσίες που δεν μπορούν να ανακαλυφθούν. Αυτές οι εργασίες ωστόσο, μπορεί να χρειάζεται να χρησιμοποιήσουν κάποιες άλλες υπηρεσίες, οι οποίες με τη σειρά τους απαιτούν τη σύνδεση με αυτές τις υπηρεσίες.
2. Να αντιμετωπίσει θέματα σχετικά με το αν θα πρέπει ή όχι να επανασυνδεθεί ένα αντικείμενο, ενώ μια συνεδρία ενός χρήστη είναι ενεργή. Αυτές οι αποφάσεις συνήθως λαμβάνονται από τον σχεδιαστή του συστήματος. Σε αυτό το προγραμματιστικό μοντέλο χρειάζεται να παρέχονται μηχανισμοί μέσω των οποίων ο σχεδιαστής να μπορεί να περιορίσει τις επιτρεπόμενες αλλαγές κατά τη σύνδεση ενός αντικειμένου.

5.2.5.2 Αποτυχίες στη σύνδεση αντικειμένων

Η σύνδεση ενός αντικειμένου με μία υπηρεσία αποτυγχάνει όταν η υπηρεσία στην οποία έχει συνδεθεί το αντικείμενο καθίσταται μη προσπελάσιμη. Αυτό μπορεί να συμβεί είτε λόγω κατάρρευσης μιας υπηρεσίας είτε λόγω ύπαρξης διακοπών στις συνδέσεις δικτύου, ή ακόμα κι όταν μια υπηρεσία ανακαλεί την πρόσβαση μιας εφαρμογής σε αυτή. Τέτοιες αποτυχίες μπορούν να επηρεάσουν την τρέχουσα ενεργή αλληλεπίδραση συνεδρίας μ' αυτό το αντικείμενο ή άλλα αντικείμενα στη δραστηριότητα. Για παράδειγμα, όταν αποτύχει μια σύνδεση ενός αντικειμένου, τότε οι πληροφορίες διαμόρφωσης σ' ένα άλλο αντικείμενο της ίδιας συνεδρίας, είναι πιθανό να πρέπει να τροποποιηθούν.

Η αποτυχία σύνδεσης ενός αντικειμένου μπορεί να απαιτήσει ενέργειες ανάκαμψης που θα εκτελεστούν από άλλα αντικείμενα και ρόλους. Σε ορισμένες εφαρμογές οι αποτυχίες σύνδεσης μπορεί επίσης να απαιτήσουν την εκτέλεση ενεργειών αναδιαμόρφωσης σε επίπεδο δραστηριότητας. Για τον λόγο αυτό, οι αποτυχίες σύνδεσης αντικειμένου μπορεί να χρειαστεί να κοινοποιηθούν στις αντιδράσεις στο επίπεδο δραστηριότητας.

Το μοντέλο ανάκαμψης είναι απαραίτητο να υποστηρίζει την ανίχνευση τέτοιων αποτυχιών στη σύνδεση αντικειμένων. Οι ενέργειες ανάκαμψης για το χειρισμό ορισμένων ειδών αποτυχιών, μπορούν να προγραμματιστούν από έναν διαχειριστή αντικειμένου. Για παράδειγμα, οι αποτυχίες σύνδεσης εξαιτίας των αποσυνδέσεων του δικτύου, μπορούν να αντιμετωπισθούν μέσω της προσπάθειας σύνδεσης του αντικειμένου με την ίδια υπηρεσία ξανά, εφόσον βέβαια είναι πιθανό οι αποσυνδέσεις δικτύου να αποκατασταθούν γρήγορα. Αν η αιτία της αποτυχίας σύνδεσης προέρχεται από την ανάκληση μιας υπηρεσίας, ο διαχειριστής αντικειμένου μπορεί να προσπαθήσει να ανακαλύψει και να συνδέσει το αντικείμενο σε άλλη υπηρεσία του ίδιου τύπου. Επίσης, όσον αφορά την ανάκαμψη, το προγραμματιστικό μοντέλο χρειάζεται να υποστηρίξει την επικοινωνία των συμβάντων που σχετίζονται με αλλαγές στην κατάσταση σύνδεσης ενός αντικειμένου, με άλλες οντότητες στην δραστηριότητα.

5.2.5.3 Λειτουργικές αποτυχίες σε επίπεδο υπηρεσίας

Οι αποτυχίες αυτού του είδους οφείλονται σε εξαιρέσεις που εγείρονται κατά την κλήση κάποιας μεθόδου υπηρεσίας στα πλαίσια μιας ενέργειας σε μια λειτουργία ρόλου ή μιας αντίδρασης. Μια τέτοια ενέργεια μπορεί να ανήκει στο μοντέλο κλήσης μεθόδου ή στο μοντέλο αλληλεπίδρασης συνεδρίας. Αρκετές αποφάσεις σχεδιασμού πρέπει να ληφθούν σε σχέση με τον χειρισμό εξαιρέσεων που προκαλούν αυτές οι αλληλεπιδράσεις.

Το πρώτο ερώτημα που προκύπτει είναι το που πρέπει να τοποθετηθούν οι χειριστές εξαιρέσεων. Στη τρέχουσα προσέγγιση, θεωρήθηκε ότι η υποστήριξη χειρισμού εξαιρέσεων πρέπει να γίνει από κοινού στο επίπεδο ενεργειών και λειτουργίας, ώστε να δίνεται μεγαλύτερη ευελιξία στο χειρισμό εξαιρέσεων. Αυτό επιτρέπει το χειρισμό εξαίρεσης πρώτα στο πεδίο της ενέργειας κι ακολούθως, αν απαιτείται, στο πεδίο της λειτουργίας. Ο περιορισμός του χειρισμού εξαιρέσεων μόνο στο πεδίο των λειτουργιών δεν προτείνεται για τους εξής λόγους:

1. Στο πεδίο της λειτουργίας ρόλου, μόνο περιορισμένες ενέργειες ανάκαμψης μπορούν να εκτελεστούν, δεδομένου ότι η ακριβής περιβάλλουσα κατάσταση της εξαίρεσης δεν είναι διαθέσιμη.
2. Έπειτα, εάν αποτύχει μια ενέργεια, το σύνολο των λειτουργιών τερματίζεται, με αποτέλεσμα να μην εκτελεστούν οι ακόλουθες ενέργειες. Η υποστήριξη της ανάκαμψης μέσω του χειρισμού εξαίρεσης σε επίπεδο ενέργειας, παρέχει τη δυνατότητα να εκτελεστούν οι ακόλουθες ενέργειες.

Το δεύτερο ερώτημα που προκύπτει είναι αν θα είναι ορατές οι εξαιρέσεις στο επίπεδο των υπηρεσιών, για να είναι δυνατός ο χειρισμός τους από τους χειριστές εξαιρέσεων. Αυτό πράγματι απαιτείται σε ορισμένες περιπτώσεις όπως θα παρουσιαστεί παρακάτω. Σε αρκετές εφαρμογές, φαίνεται ότι μερικές φορές είναι απαραίτητο να γνωρίζει κάποιος την ακριβή αιτία μιας αποτυχίας στο επίπεδο υπηρεσίας, ώστε να την χειριστεί σωστά. Αυτό είναι ιδιαίτερα σημαντικό κατά την εκτέλεση ανάκαμψης σε επίπεδο ενεργειών. Για παράδειγμα σε ένα σύστημα αναπαραγωγής ήχου, θεωρούμε τη λειτουργία Play η οποία έχει αντίστοιχα μια μέθοδο playMusic. Η μέθοδος αυτή μπορεί να παρουσιάσει κάποια εξαίρεση αν η συσκευή του χρήστη δεν υποστηρίζει το τύπο του μουσικού αρχείου που θα επιλέξει ο χρήστης. Στο πεδίο της μηχανικής λογισμικού με χρήση συνιστωσών, έχει προσδιοριστεί η ανάγκη του χειρισμού εξαιρέσεων επιπέδου συνιστώσας εντός των περιτυλιγμάτων ενθουλάκωσης. Αυτή η ανάγκη είναι παρόμοια με την προαναφερόμενη απαίτηση της ορατότητας των εξαιρέσεων σε επίπεδο υπηρεσιών στο πεδίο της ενέργειας.

Η τρίτη απόφαση αφορά τον χειρισμό εξαιρέσεων που περιλαμβάνονται σε μια αλληλεπίδραση συνεδρίας. Μια τέτοια αλληλεπίδραση εκτελείται μέσω ενός συστατικού διεπαφής. Γίνεται εύκολα αντιληπτό ότι ένα τέτοιο συστατικό διεπαφής, το οποίο είναι συνήθως προγραμματισμένο σε γλώσσες (όπως Java, C++ κ.ά.), μπορεί να ορίσει τους δικούς της χειριστές εξαιρέσεων. Τέτοια συστατικά σχεδιάζονται και εκτελούνται ξεχωριστά από τις προδιαγραφές δραστηριότητας. Οι εξαιρέσεις που προκύπτουν από μία αλληλεπίδραση θα πρέπει να διαδίδονται πρώτα στο συστατικό διεπαφής. Είναι πιθανό το συστατικό διεπαφής να εκτελεί κάποιες ενέργειες ανάκαμψης ως μέρος του γενικού χειρισμού εξαιρέσεων. Στην περίπτωση που δεν οριστούν χειριστές εξαιρέσεων στο συστατικό διεπαφής ή αν αποτύχει η ανάκαμψη,

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

απαιτείται από το συστατικό διεπαφής να κοινοποιήσει ένα γεγονός τερματισμού αλληλεπίδρασης στο διαχειριστή αντικειμένων. Τότε ο διαχειριστής αντικειμένου θα πρέπει να τερματίσει την αλληλεπίδραση και να δημιουργήσει μια εξαίρεση για την αλληλεπίδραση που ματαιώθηκε στο πεδίο της ενέργειας (session-aborted exception).

5.2.5.4 Ακυρώσεις της περιβάλλουσας πληροφορίας

Ορισμένες πολιτικές ελέγχου πρόσβασης που βασίζονται στην περιβάλλουσα κατάσταση μπορεί να απαιτήσουν την εκτέλεση ορισμένων λειτουργιών ανάλογα με τα προνόμια, που ορίζονται σε σχέση με τις συνθήκες περιβάλλουσας κατάστασης. Τέτοιες λειτουργίες μπορεί να περιλαμβάνουν μια αλληλεπιδραστική συνεδρία με κάποιες υπηρεσίες. Για τη σωστή αντιμετώπιση τέτοιων πολιτικών, είναι ζωτικής σημασίας να ανακληθεί το προνόμιο εκτέλεσης λειτουργίας, εφόσον η συνθήκη που περιλαμβάνει την περιβάλλουσα πληροφορία καταστεί ψευδής. Η έννοια της ακύρωσης της περιβάλλουσας πληροφορίας περιγράφεται στην εργασία [37]. Οι μηχανισμοί προϋποθέσεων είναι ανεπαρκείς για τέτοιες περιπτώσεις. Αυτό συμβαίνει γιατί χρειάζεται συνεχής έλεγχος των συνθηκών περιβάλλουσας πληροφορίας ενώ ο χρήστης εκτελεί μια λειτουργία, καθώς είναι δυνατόν να πρέπει να ανακληθεί το προνόμιο εφ' όσον η συνθήκη παύει να ισχύει.

Για τον έλεγχο των συνθηκών περιβάλλουσας κατάστασης μίας λειτουργίας, το σχεδιαστικό μοντέλο παρέχει έναν μηχανισμό που καλείται ContextGuard, ο οποίος μπορεί να συνεργάζεται με μια λειτουργία ρόλου. Κάθε ContextGuard αποτελείται από δύο μέρη. Το πρώτο είναι η προδιαγραφή της συνθήκης που περιλαμβάνει στοιχεία της περιβάλλουσας κατάστασης. Δεύτερον είναι η προδιαγραφή των γεγονότων που μπορεί να συνδέονται με την παραβίαση μιας συνθήκης ContextGuard, δηλαδή των γεγονότων που μπορεί όταν εμφανιστούν, να καταστήσουν ψευδή τη συνθήκη του πρώτου μέρους. Ο διαχειριστής ρόλου αξιολογεί την συνθήκη φρούρησης όταν συμβαίνουν τα καθορισμένα γεγονότα. Αν αυτή η συνθήκη δεν παραμένει αληθής, ο διαχειριστής ρόλου τερματίζει την αλληλεπίδραση ανακαλώντας έτσι την πρόσβαση του χρήστη.

5.2.6 Μοντέλο ανάκαμψης από σφάλματα με χρήση συμβάντων και εξαιρέσεων

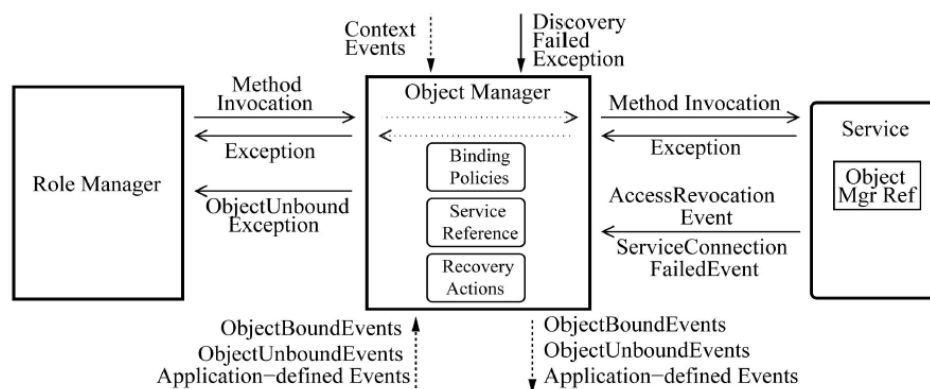
Στη συγκεκριμένη προσέγγιση έχουν μελετηθεί κυρίως θέματα ευρωστίας που αφορούν μηχανισμούς για ασύγχρονο χειρισμό γεγονότων και σύγχρονο χειρισμό εξαιρέσεων σε επίπεδο εφαρμογής. Επιπλέον, τα γεγονότα αντιπροσωπεύουν τη δημιουργία μιας κατάστασης (επιτυχίας ή αποτυχίας) που σχετίζεται με την εφαρμογή με επίγνωση της περιβάλλουσας κατάστασης. Οι εξαιρέσεις αντιπροσωπεύουν μια υποκατηγορία των γεγονότων που προκύπτουν με σύγχρονο τρόπο κατά την εκτέλεση μίας ενέργειας μέσα σε μία λειτουργία ρόλου ή σε μια αντίδραση. Οι εξαιρέσεις συμπεριλαμβάνουν επίσης πληροφορίες που είναι σχετικές με την αιτία δημιουργίας του σφάλματος το οποίο είναι ουσιαστικό για τους χειριστές εξαιρέσεων.

5.2.6.1 Μοντέλο συμβάντων στο επίπεδο του αντικειμένου και μοντέλο ανάκαμψης στο ίδιο επίπεδο

Στην εικόνα 36 παρουσιάζεται το μοντέλο συμβάντων στο επίπεδο του αντικειμένου. Το μοντέλο περιέχει τρεις κατηγορίες συμβάντων: συμβάντα που δείχνουν αποτυχίες συνθηκών, συμβάντα αλλαγής της κατάστασης σύνδεσης του

αντικειμένου και συμβάντα περιβάλλουσας κατάστασης. Τα συμβάντα που δείχνουν αποτυχίες συνθηκών διανέμονται σ' ένα διαχειριστή αντικειμένου μέσω του middleware. Τα συμβάντα, αλλαγής της κατάστασης σύνδεσης του αντικειμένου, παράγονται από τον διαχειριστή αντικειμένου. Ένα συμβάν σύνδεσης αντικειμένου (ObjectBoundEvent) παράγεται όταν το αντικείμενο έχει επιτυχώς συνδεθεί με μια υπηρεσία. Επίσης παράγεται ένα συμβάν αποσύνδεσης αντικειμένου (ObjectUnboundEvent) κάθε φορά που το αντικείμενο αποσυνδέεται. Τα συμβάντα περιβάλλουσας κατάστασης διανέμονται στους διαχειριστές αντικειμένων από τις υπηρεσίες περιβάλλουσας κατάστασης.

Δύο είδη συνθηκών αποτυχίας μπορεί να προκύψουν από έναν διαχειριστή αντικειμένου. Το πρώτο είδος σχετίζεται με την σύνδεση ενός αντικειμένου με μία υπηρεσία ενεργού χώρου. Οι συγκεκριμένες καταστάσεις αποτυχίας μπορεί να προκύπτουν λόγω αποτυχίας στην ανακάλυψη υπηρεσιών, είτε λόγω αποτυχίας στη σύνδεση με μια υπηρεσία. Μια αποτυχία ανακάλυψης αποστέλλεται στον διαχειριστή αντικειμένου από μια υπηρεσία ανακάλυψης, δημιουργώντας μια εξαίρεση αποτυχίας ανακάλυψης (discovery failed exception). Αυτή η εξαίρεση προκύπτει μέσα από την ενέργεια σύνδεσης του αντικειμένου που βασίζεται στην ανακάλυψη. Ένας χειριστής γι' αυτό τον τύπο εξαίρεσης, μπορεί να οριστεί για αυτή την ενέργεια και για την εκτέλεση άλλων εναλλακτικών ενεργειών.



Εικόνα 36 - Μοντέλο συμβάντων για επίπεδο αντικειμένου

Οι αποτυχίες σύνδεσης (καταρρεύσεις υπηρεσιών και ανακλήσεις προσβάσεων) από την άλλη πλευρά, αποστέλλονται ως ασύγχρονα συμβάντα στο διαχειριστή αντικειμένου από το middleware. Η προσέγγιση αυτή διαχωρίζει τις αποτυχίες σύνδεσης που προκαλούνται εξαιτίας καταρρεύσεων υπηρεσιών και εκείνων που δημιουργούνται εξαιτίας ανακλήσεων πρόσβασης. Αυτό γίνεται γιατί το καθένα χρειάζεται διαφορετικό τρόπο ανάκαμψης για να επιλυθούν τα θέματα σφαλμάτων.

Επίσης, οι ενέργειες ανάκαμψης μπορούν να προγραμματιστούν για να χειριστούν εξαιρέσεις αποτυχίας στη σύνδεση ενός αντικειμένου, μέσω μηχανισμών αντιδράσεων. Με τη σειρά τους, οι αντιδράσεις ορίζονται στις προδιαγραφές αντικειμένου. Ένα αντικείμενο ακόμα μπορεί να περιγράψει γεγονότα αλλαγής κατάστασης σύνδεσης άλλων αντικειμένων που ορίζονται στην δραστηριότητα. Στο προγραμματιστικό μοντέλο είναι επίσης δυνατόν να αποφευχθεί η αλλαγή σύνδεσης ενός αντικειμένου, ενώ μια συνεδρία είναι ενεργή με αυτό το αντικείμενο. Αυτές οι πολιτικές μπορεί να προγραμματιστούν με χρήση του μηχανισμού των προϋποθέσεων που παρέχεται στο μοντέλο αντίδρασης.

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

Το δεύτερο είδος συνθηκών αποτυχίας σ' ένα αντικείμενο αποτελούνται από εκείνες που προκύπτουν κατά τη διάρκεια μιας αλληλεπίδρασης χρήστη με την υπηρεσία ενεργού χώρου στην οποία το αντικείμενο είναι συνδεδεμένο. Όπως παρουσιάζεται στην εικόνα 35, δύο μοντέλα αλληλεπίδρασεων υποστηρίζονται στο προγραμματιστικό πλαίσιο της προσέγγισης. Και τα δυο αυτά μοντέλα αντιμετωπίζουν τα ακόλουθα είδη αποτυχιών:

1. Πρώτον η υπηρεσία στην οποία είναι συνδεδεμένο το αντικείμενο, μπορεί να εγείρει εξαιρέσεις κατά τη διάρκεια κλήσης μιας μεθόδου. Στο μοντέλο κλήσης μεθόδου, οι εξαιρέσεις που εγείρονται από μια υπηρεσία προωθούνται στον διαχειριστή ρόλων, ενώ στο μοντέλο αλληλεπίδρασης συνεδρίας, προωθούνται στα συστατικά διεπαφής.
2. Δεύτερον, η σύνδεση ενός αντικειμένου με μια υπηρεσία μπορεί να αποτύχει. Τότε ο διαχειριστής αντικειμένου δημιουργεί μια εξαίρεση αποσύνδεσης αντικειμένου (`ObjectUnboundException`) στην περιβάλλουσα κατάσταση εκτέλεσης όλων των υπό εξέλιξη αλληλεπιδραστικών συνεδριών, μ' αυτό το αντικείμενο. Είναι πιθανόν μια μέθοδος να κληθεί στο αντικείμενο όταν αυτό δεν είναι συνδεδεμένο. Σ' αυτή την περίπτωση επίσης, ο διαχειριστής αντικειμένου δημιουργεί εξαίρεση `ObjectUnboundException`.

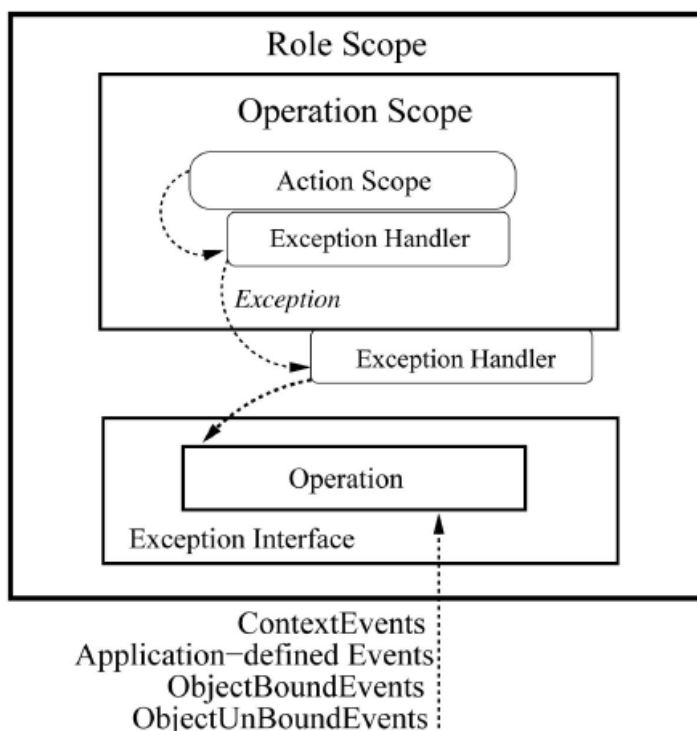
5.2.6.2 Μοντέλο χειρισμού εξαίρεσης σε επίπεδο ρόλου

Ο έννοια του ρόλου περιέχει τρία πεδία εκτέλεσης, οι οποίες φαίνονται στην εικόνα 37. Αυτά αντιστοιχούν στο πεδίο του ρόλου, στο πεδίο της λειτουργίας ρόλου και στο πεδίο της ενέργειας. Το πεδίο του ρόλου ενθυλακώνει το πεδίο της λειτουργίας ρόλου, που με τη σειρά του ενθυλακώνει το πεδίο της ενέργειας.

Τρεις κατηγορίες εξαιρέσεων μπορεί να προκύψουν ως μέρος των εκτελέσεων λειτουργιών ρόλου. Αυτές περιλαμβάνουν:

1. Εξαιρέσεις που προέρχονται από μια υπηρεσία κατά τη διάρκεια εκτέλεσης μιας μεθόδου σ' αυτήν,
2. Εξαιρέσεις που προκύπτουν εξαιτίας της διακοπής σύνδεσης ενός αντικειμένου και
3. Ακυρώσεις της περιβάλλουσας πληροφορίας κατά τη διάρκεια εκτέλεσης μίας λειτουργίας.

Τα δύο πρώτα είδη εμφανίζονται στο πεδίο της ενέργειας, ενώ το τρίτο είδος εμφανίζεται στο πεδίο της λειτουργίας.



Εικόνα 37 - Σχέσεις μεταξύ των πεδίων ρόλου, λειτουργίας και ενέργειας

Ένας χειριστής εξαιρέσεων μπορεί να προσαρτηθεί τόσο σε μία ενέργεια όσο και σε μια λειτουργία. Επίσης μπορεί να περιέχει μία ή περισσότερες ενέργειες. Ο χειρισμός εξαιρέσεων βασίζεται στο μοντέλο τερματισμού. Η εκτέλεση της ενέργειας που περιέχει μια εξαίρεση, τερματίζεται και αναζητείται χειριστής για αυτή την εξαίρεση στο πεδίο της ενέργειας. Αν έχει καθοριστεί χειριστής κι αν η ενέργεια χειρισμού της εξαίρεσης είναι επιτυχής, τότε συνεχίζεται η εκτέλεση των μετέπειτα ενεργειών, που καθορίζονται στη λειτουργία. Αν δεν έχει καθοριστεί χειριστής, η εξαίρεση διαδίδεται στο πεδίο του ρόλου. Στο πεδίο του ρόλου παρέχεται μια διεπαφή εξαίρεσης (exception interface). Αυτή περιέχει ένα σύνολο λειτουργιών που μπορούν να εκτελεστούν από τα μέλη του ρόλου ώστε να εκτελεστούν δράσεις ανάκαμψης.

i) Χειρισμός εξαιρέσεων σε επίπεδο ενέργειας

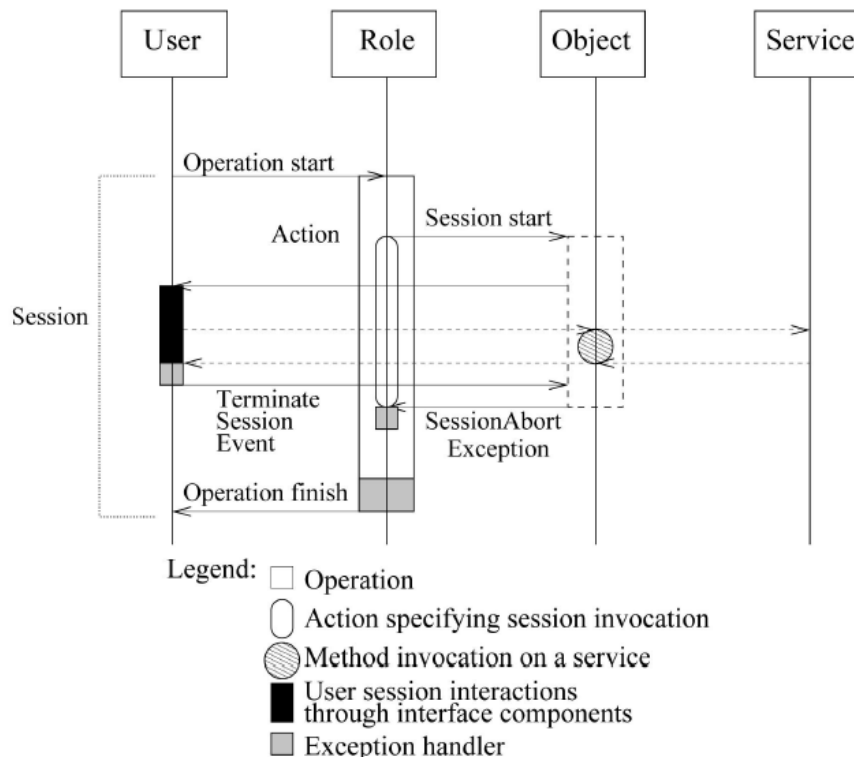
Σε παρακάτω υποκεφάλαιο θα παρουσιαστεί ο χειρισμός των συγκεκριμένων εξαιρέσεων. Συγκεκριμένα, θα αναφερθεί στο σύστημα πληροφοριών ασθενών. Γενικά, για το χειρισμό τέτοιων καταστάσεων, μπορεί κάποιος να προγραμματίσει ένα χειριστή εξαιρέσεων που καλεί μια μέθοδο για την εφεδρική υπηρεσία αυτής που αντιμετώπισε πρόβλημα.

ii) Χειρισμός εξαιρέσεων σε επίπεδο συνεδρίας

Οι εξαιρέσεις σε επίπεδο υπηρεσίας που εμφανίζονται σε μία αλληλεπιδραστική συνεδρία, διαδίδονται στα στοιχεία της διεπαφής, όπου και θα γίνει ο σχετικός χειρισμός. Στη προσέγγιση που παρουσιάζεται, απαιτείται ο τερματισμός της συνεδρίας, εφόσον η εξαίρεση δεν τύχει κατάλληλου χειρισμού από ένα στοιχείο της διεπαφής. Στο μοντέλο αυτό, παρέχεται το συμβάν `TerminateSessionEvent`, το οποίο μπορεί να αποσταλεί από το στοιχείο διεπαφής στον διαχειριστή αντικειμένου ώστε να δηλωθεί αυτός ο τερματισμός. Ο διαχειριστής

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

αντικειμένου τερματίζει τη συνεδρία κι εγείρει την εξαίρεση `SessionAbortException` στο πεδίο της ενέργειας απ' όπου ξεκίνησε η συνεδρία. Το μοντέλο παρουσιάζεται στην εικόνα 38.



Εικόνα 38 - Χειρισμός εξαιρέσεων σε επίπεδο συνεδρίας

Ο δεύτερος τύπος αποτυχίας που συναντάται από μια αλληλεπιδραστική συνεδρία, αντιστοιχεί στις διαταραχές των συνεδριών συναλλαγών όταν αποτύχει η σύνδεση ενός αντικειμένου. Αυτό έχει ως αποτέλεσμα τον τερματισμό της συνεδρίας κι ακολούθως τίθεται σε λειτουργία η εξαίρεση `SessionAbortException` στην ενέργεια από την οποία ξεκίνησε η συνεδρία. Ο σχεδιαστής μπορεί να θέλει να διαχειριστεί τις εν λόγω συναλλαγές με τον ακόλουθο τρόπο: Όταν ένας διαχειριστής αντικειμένου επανασυνδεθεί με την υπηρεσία από την οποία ξεκίνησε η συναλλαγή, μπορεί είτε να συνεχιστεί, είτε να ματαιωθεί, η συναλλαγή που έχει διακοπεί.

iii) Χειρισμός εξαιρέσεων σε επίπεδο λειτουργίας

Οι εξαιρέσεις που έχουν απομείνει χωρίς να τύχουν χειρισμού στο πεδίο της ενέργειας, διαδίδονται στο πεδίο της λειτουργίας ρόλου. Επιπρόσθετα, στο πεδίο της λειτουργίας μπορεί να εμφανιστούν αποτυχίες λόγω ακυρώσεων περιβάλλουσας πληροφορίας. Το μοντέλο παρέχει την εξαίρεση `ContextInvalidationException` για την αναπαράσταση τέτοιων σφαλμάτων. Όπως και προηγουμένως, παρακάτω θα παρουσιαστεί ο χειρισμός των συγκεκριμένων εξαιρέσεων και πιο συγκεκριμένα, θα αναφερθεί στο σύστημα πληροφοριών ασθενών.

iv) Διεπαφή Εξαιρέσεων σε επίπεδο ρόλου

Η συμμετοχή του χρήστη στις ενέργειες ανάκαμψης υποστηρίζεται μέσω μηχανισμών διεπαφής εξαιρέσεων σε επίπεδο ρόλου. Στις εφαρμογές με επίβλεψη της

περιβάλλουσας κατάστασης, αυτός ο τύπος ανάκαμψης απαιτείται για τους ακόλουθους λόγους:

- Σε συγκεκριμένες καταστάσεις, μπορεί να μην είναι δυνατό να γίνει αυτόματος χειρισμός των εξαιρέσεων, που εγείρονται στο πεδίο της λειτουργίας ρόλου.
- Σε εφαρμογές όπου συνεργάζονται πολλοί χρήστες, προκύπτουν καταστάσεις όπου ορισμένες ενέργειες ανάκαμψης μπορεί να χρειαστεί να εκτελεστούν από ένα μέλος του ρόλου, ώστε να βοηθήσουν στην ανάκαμψη από σφάλματα που περιέχονται σε κάποιους άλλους ρόλους.
- Στο περιβάλλον μπορούν να προκύψουν μη ομαλές καταστάσεις, όπου μπορεί να απαιτηθεί να γίνουν οι κατάλληλες ενέργειες από ένα χρήστη που έχει έναν συγκεκριμένο ρόλο. Αυτού του είδους οι εξαιρέσεις απαντώνται και σε προσανατολισμένα σε πράκτορες συστήματα, όπου θεωρούνται ως εξαιρέσεις του περιβάλλοντος.
- Είναι πιθανό να απαιτηθεί η εκτέλεση ορισμένων εναλλακτικών ενεργειών, όταν εγερθούν μέσα σε μία δραστηριότητα σφάλματα σύνδεσης αντικειμένου.

Στην εικόνα 39 παρουσιάζεται μία δομή της διεπαφής εξαίρεσης σε επίπεδο ρόλου. Μια λειτουργία της διεπαφής εξαίρεσης μπορεί να ενεργοποιηθεί από τα ακόλουθα συμβάντα:

1. από μία εξαίρεση που δεν έχει τύχει χειρισμού σε μια από τις λειτουργίες του ρόλου,
2. από ένα συμβάν που έχει οριστεί στην εφαρμογή,
3. από συμβάντα σύνδεσης ή αποσύνδεσης αντικειμένου και
4. από συμβάντα περιβάλλουσας κατάστασης.

Κάθε τέτοιο συμβάν, δίνει τη δυνατότητα εκτέλεσης της λειτουργίας διεπαφής εξαίρεσης, σε ένα ή περισσότερα μέλη του ρόλου. Ένα μέλος ρόλου μπορεί να εκτελέσει μία λειτουργία διεπαφής εξαίρεσης μόνο όταν υπάρχει στην ουρά της διεπαφής εξαίρεσης ένα τέτοιου είδους συμβάν για τη συγκεκριμένη λειτουργία κι εφόσον η προϋπόθεση γι' αυτή τη λειτουργία, είναι αληθής. Ένα συμβάν παρέχει την περιβάλλουσα κατάσταση για την εκτέλεση των σχετιζομένων λειτουργιών διεπαφής εξαίρεσης. Η διεπαφή εξαίρεσης υποστηρίζει ένα μοντέλο ουράς για τη διανομή και τον χειρισμό των συμβάντων. Μια λειτουργία διεπαφής εξαίρεσης ενεργοποιείται όταν ένα συμβάν παραδίδεται στην ουρά μιας λειτουργίας διεπαφής εξαίρεσης.

Role	roleId
Operation	Operation-Definition
Exception_Interface	
When Event	(roleId.opId.Exception Application defined events Binding events Context events)
EnableFor	(Invoker ANY ALL)
Operation	Operation-Definition

Εικόνα 39 - Δομή διεπαφής εξαίρεσης σε επίπεδο ρόλου

Είναι πιθανό να υπάρχουν πολλοί χρήστες σ' ένα ρόλο. Επομένως προκύπτει ένα ερώτημα σχετικά με το ποιο μέλος του ρόλου θα πάρει το προνόμιο για εκτέλεση της λειτουργίας διεπαφής εξαίρεσης. Το σχεδιαστικό μοντέλο παρέχει τρεις qualifiers για τον καθορισμό αυτό, τα Invoker, ALL και ANY, που μπορούν να συσχετιστούν με μια λειτουργία διεπαφής εξαίρεσης. Τα συμβάντα αφαιρούνται από την ουρά, όταν η λειτουργία διεπαφής εξαίρεσης έχει εκτελεστεί σύμφωνα με τα παραπάνω qualifiers. Όταν δύο μέλη του ρόλου προσπαθούν να εκτελέσουν μια λειτουργία διεπαφής εξαίρεσης που καθορίζεται από το qualifier ALL, τότε μόνο ένα μέλος θα την εκτελέσει, ακολουθώντας έτσι ένα μη αιτιοκρατικό μοντέλο. Ο qualifier ALL απαιτεί από κάθε μέλος του ρόλου να εκτελέσει τη λειτουργία διεπαφής εξαίρεσης πριν το συμβάν αφαιρεθεί από την ουρά της εξαίρεσης διεπαφής ρόλου.

5.3 Μεθοδολογία ανάπτυξης εφαρμογών

Για να γίνει επίλυση τέτοιων προβλημάτων με τη συγκεκριμένη προσέγγιση είναι απαραίτητο να κατανοηθούν οι λειτουργικές απαιτήσεις του προβλήματος. Αφού γίνει αυτό, πρέπει να ακολουθήσει μια διαγραμματική παρουσίαση της δραστηριότητας. Μέσα σε αυτή πρέπει να οριστούν σαφώς οι ρόλοι που μπορεί να υπάρχουν στην εφαρμογή, καθώς επίσης και τα αντικείμενα που είναι ιδιωτικά γι' αυτούς τους ρόλους, εφόσον αυτά υπάρχουν στην εφαρμογή. Τα υπόλοιπα αντικείμενα θα πρέπει να οριστούν στον χώρο ονομάτων της δραστηριότητας και θα πρέπει να συνδέσουν την δραστηριότητα με τις διάφορες υπηρεσίες ενεργού χώρου. Αφού γίνει ο διαχωρισμός των κύριων συστατικών της εφαρμογής πρέπει να σχεδιαστεί το μοντέλο ανάκαμψης.

Στην επόμενη ενότητα θα ακολουθήσει ένα παράδειγμα εφαρμογής ώστε να γίνουν πλήρως κατανοητές οι έννοιες που προαναφέρθηκαν καθώς και ο τρόπος με τον οποίο επιλύονται.

5.4 Παράδειγμα ανάπτυξης εφαρμογών σύμφωνα με αυτή την προσέγγιση

Στη συνέχεια παρουσιάζονται δύο εφαρμογές για την απεικόνιση των παραπάνω θεμάτων. Αυτές περιλαμβάνουν ένα σύστημα αναπαραγωγής μουσικής με επίγνωση της περιβάλλουσας κατάστασης κι ένα σύστημα πληροφοριών για ασθενείς με επίγνωση της περιβάλλουσας κατάστασης.

5.4.1 Σύστημα αναπαραγωγής μουσικής

Η εφαρμογή αυτή τρέχει σε μια προσωπική κινητή συσκευή χρήστη, όπως είναι το PDA, και υποστηρίζει τις ακόλουθες απαιτήσεις με βάση την περιβάλλουσα

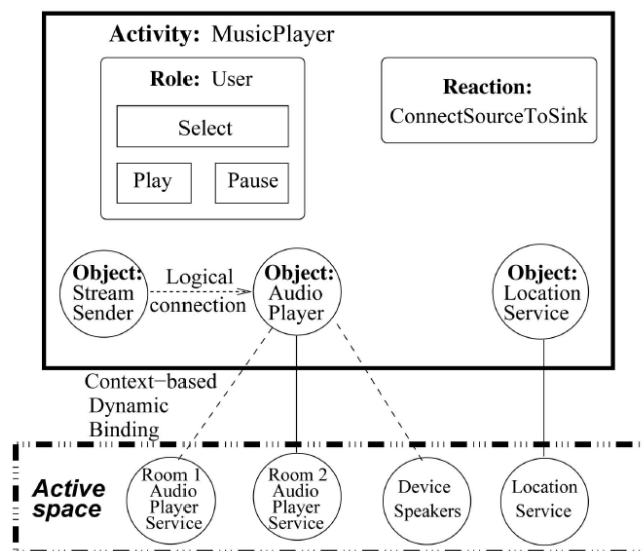
Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

κατάσταση: όταν ο χρήστης τρέξει την εφαρμογή αυτή, θα πρέπει να αρχίσει η ροή μουσικής στην υπηρεσία αναπαραγωγής ήχου της προσωπικής συσκευής του χρήστη. Επίσης, όταν ο χρήστης μπει σ' ένα δωμάτιο, η εφαρμογή θα πρέπει να ανακαλύψει και να συνδεθεί με την υπηρεσία αναπαραγωγής ήχου του δωματίου και να αρχίσει να μεταδίδει μουσική εφόσον βέβαια δεν είναι παρόν στο δωμάτιο κάποιο άλλο πρόσωπο. Επιπλέον, όταν ο χρήστης βγει από το δωμάτιο ή όταν κάποιο άλλο πρόσωπο εισέλθει στο δωμάτιο, τότε η εφαρμογή θα πρέπει να συνδεθεί με την υπηρεσία αναπαραγωγής ήχου της συσκευής του χρήστη και να συνεχίσει να μεταδίδει μουσική στον ίδιο.

Στην εικόνα 40 παρουσιάζεται η σχηματική αναπαράσταση της δραστηριότητας αναπαραγωγής μουσικής. Ορίζουμε ο ρόλος User και παρέχονται λειτουργίες για τις επιλογές Select, Play και Pause. Ύστερα, ορίζονται τρία αντικείμενα στη συγκεκριμένη δραστηριότητα: StreamSender, AudioPlayer και LocationService. Το αντικείμενο StreamSender αναπαριστά την πηγή του ήχου. Το αντικείμενο AudioPlayer αντιπροσωπεύει το μέσο με το οποίο αναπαράγεται ο ήχος και το οποίο μπορεί να συνδέεται με την υπηρεσία αναπαραγωγής ήχου στη συσκευή του χρήστη ή στη συσκευή που βρίσκεται στο δωμάτιο στο οποίο ο χρήστης μπορεί να είναι παρόν. Το αντικείμενο LocationService συνδέεται με την υπηρεσία περιβάλλουσας κατάστασης που παρέχει πληροφορίες όσον αφορά την τοποθεσία των χρηστών. Η δραστηριότητα εισάγει τα συμβάντα UserArrivalEvent, UserDepartureEvent και RoomStatusChangeEvent μέσω της υπηρεσίας αυτής. Τέλος, χρησιμοποιείται η αντίδραση ConnectSourceToSink για τη ρύθμιση των αντικειμένων StreamSender και AudioPlayer όποτε αλλάξει η σύνδεση του αντικειμένου AudioPlayer.



Εικόνα 40 - Δραστηριότητα συστήματος αναπαραγωγής μουσικής

5.4.1.1 Σύνδεση και ανακάλυψη με βάση την περιβάλλουσα κατάσταση

Για την παρουσίαση των μηχανισμών σύνδεσης και ανακάλυψης, χρησιμοποιείται ως παράδειγμα το αντικείμενο AudioPlayer. Οι προδιαγραφές για αυτό το αντικείμενο φαίνεται στην εικόνα 41. Χρησιμοποιείται ένας ψευδοκώδικας για την παρουσίαση των προδιαγραφών. Πιο συγκεκριμένα, οι όροι με έντονα

γράμματα αναπαριστούν XML ετικέτες. Ορίζονται τρεις αντιδράσεις για αυτό το αντικείμενο:

- Η αντίδραση `BindOnUserArrival` (γραμμές 2-7) ενεργοποιείται από το συμβάν `UserArrivalEvent` που αντιστοιχεί στη μεταβλητή `thisUser`. Η μεταβλητή `thisUser` με τη σειρά της, όταν βρίσκεται σε μια έκφραση, αντιπροσωπεύει την ταυτότητα του χρήστη στον οποίο αντιστοιχεί ένα συμβάν περιβάλλουσας κατάστασης που δημιουργείται. Η αντίδραση αυτή συνδέεται με το αντικείμενο της υπηρεσίας αναπαραγωγής ήχου στο δωμάτιο όπου ανιχνεύεται η παρουσία του χρήστη, με την προϋπόθεση ότι κανένα άλλο πρόσωπο δεν είναι παρόν στο δωμάτιο αυτό. Η `AudioPlayerRDD` που σχετίζεται με το αντικείμενο `AudioPlayer` χρησιμοποιείται με σκοπό την ανακάλυψη. Ένα από τα χαρακτηριστικά που ορίζονται στο εν λόγω RDD είναι η τοποθεσία (`LOCATION`). Αυτό το χαρακτηριστικό είναι ένα χαρακτηριστικό-παράμετρος, όπου η τιμή του προσδιορίζεται στο χρόνο εκτέλεσης από την επερώτηση της θέσης του χρήστη από την υπηρεσία εντοπισμού.
- Η αντίδραση `BindOnUserDeparture` (γραμμές 8-10) ενεργοποιείται από το συμβάν `UserDepartureEvent`. Όταν ο χρήστης βγει από ένα δωμάτιο, η αντίδραση αυτή συνδέεται με το αντικείμενο της υπηρεσίας αναπαραγωγής ήχου που εκτελείται στη συσκευή του χρήστη.
- Η αντίδραση `BindOnRoomStatusChange` (γραμμές 11-15) ενεργοποιείται από το συμβάν `RoomStatusChangeEvent`. Η αντίδραση αυτή συνδέεται με το αντικείμενο της υπηρεσίας αναπαραγωγής ήχου στη συσκευή του χρήστη, εάν ο αριθμός των ατόμων στο δωμάτιο όπου αυτός ο χρήστης είναι παρόν, είναι μεγαλύτερος από ένα.

```

1.Object AudioPlayer RDD (//AudioPlayerRDD.xml) {
2. Reaction BindOnUserArrival
3.   When Event UserArrivalEvent(userId=thisUser)
4.   Precondition LocationService.userCount(
5.     LocationService.getLoc(thisUser)) == 1
6.   Action Bind Discover
7.     (LOCATION=LocationService.getLoc(thisUser))
8. Reaction BindOnUserDeparture
9.   When Event UserDepartureEvent
10.  Action Bind Direct (//DeviceAudioPlayerURL)
11.Reaction BindOnRoomStatusChange
12.  When Event RoomStatusChangeEvent
13.  Precondition LocationService.userCount(
14.    LocationService.getLoc(thisUser)) > 1
15.  Action Bind Direct (//DeviceAudioPlayerURL)
16.}

```

Εικόνα 41 - Σύνδεση αντικειμένου AudioPlayer

5.4.1.2 Ρύθμιση αντικειμένου με βάση την περιβάλλουσα κατάσταση

Η εικόνα 42 παρουσιάζει τις προδιαγραφές δραστηριότητας της αναπαραγωγής ήχου. Ορίζεται η λειτουργία `Play` (γραμμές 9-15) στο ρόλο `User` για τη ρύθμιση του αντικειμένου `StreamSender` ώστε να συνδεθεί με το αντικείμενο

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

AudioPlayer. Η πρώτη ενέργεια αυτής της λειτουργίας καθορίζει την διεύθυνση της πηγής μέσα στο αντικείμενο AudioPlayer (γραμμές 11-12). Η δεύτερη ενέργεια (γραμμές 13-14) προσθέτει στη συνέχεια την υπηρεσία αναπαραγωγής ήχου ως το στόχο του αντικειμένου του αποστολέα. Και οι δύο αυτές ενέργειες ακολουθούν το μοντέλο κλήσης μεθόδου που καθορίζεται μέσω του InvokeMethod. Η τρίτη ενέργεια (γραμμή 15) ανοίγει μια συνεδρία με το αντικείμενο StreamSender μέσω του οποίου ενεργοποιείται η αναπαραγωγή της μουσικής από τον χρήστη. Η ενέργεια αυτή χρησιμοποιεί το μοντέλο αλληλεπίδρασης συνεδρίας που ορίζεται μέσω του InvokeSession.

Παρέχεται επίσης η αντίδραση ConnectSourceToSink σ' αυτή τη δραστηριότητα (γραμμές 19-26), όπου ο σκοπός της είναι η αποκατάσταση της σύνδεσης μεταξύ των αντικειμένων StreamSender και AudioPlayer, για οποιοδήποτε αλλαγές σύνδεσης του AudioPlayer. Η αντίδραση αυτή ορίζεται για την ενεργοποίηση από το ObjectBoundEvent που αντιστοιχεί στο αντικείμενο AudioPlayer. Το middleware δημιουργεί αυτό το συμβάν κάθε φορά που ένα αντικείμενο συνδέεται με μια υπηρεσία. Η αρχική ρύθμιση μεταξύ των αντικειμένων StreamSender και AudioPlayer πραγματοποιείται με χρήση της λειτουργίας Play. Οι επόμενες αναδιαμορφώσεις εκτελούνται μέσω αυτής της αντίδρασης. Η προϋπόθεση διασφαλίζει ότι η αντίδραση θα εκτελεστεί μόνο όταν πραγματοποιηθεί η αρχική ρύθμιση από το χρήστη μέσω της λειτουργίας Play. Στο προγραμματιστικό μοντέλο, η λειτουργία # αναπαριστά την καταμέτρηση ενός συμβάντος, εφόσον εφαρμόζεται σ' ένα συμβάν. Για κάθε λειτουργία, το middleware παράγει το συμβάν start όταν η λειτουργία ξεκινά την εκτέλεσή της και το συμβάν finish, όταν η λειτουργία τελειώσει την εκτέλεση της.

```

1. Activity MusicPlayer {
2.   Object LocationService {
3.     When Event ActivityCreationEvent
4.     Action Bind Direct (//LocationServiceURL)
5.     ImportEvent UserArrivalEvent
6.     UserDepartureEvent, RoomStatusChangeEvent
7.   }
8.   Role User {
9.     Operation Play
10.    Precondition AudioPlayer.isBound()
11.    Action InvokeMethod AudioPlayer.setSender(
12.      StreamSender.getAddress())
13.    Action InvokeMethod StreamSender.addTarget(
14.      AudioPlayer.getAddress())
15.    Action StreamSender InvokeSession play
16.    Operation SelectAudioFile { ... }
17.    Operation Pause { ... }
18.  }
19. Reaction ConnectSourceToSink {
20.   When Event AudioPlayer.ObjectBoundEvent
21.   Precondition #Play.start - #Play.finish > 0
22.   Action InvokeMethod AudioPlayer.setSender(
23.     StreamSender.getAddress())
24.   Action InvokeMethod StreamSender.addTarget(
25.     AudioPlayer.getAddress())
26. }
27. }

```

Εικόνα 42 - Ρύθμιση αντικειμένου

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

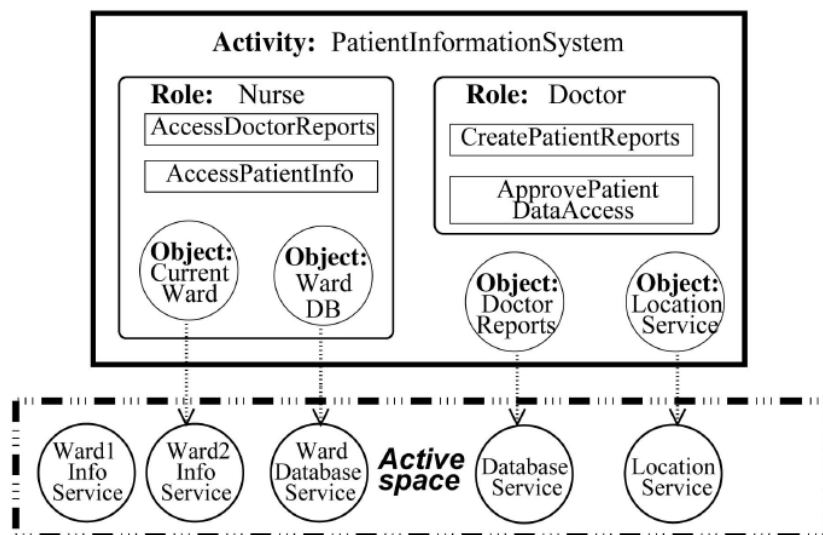
Φούντας Γεώργιος

5.4.2 Σύστημα πληροφοριών για ασθενείς

Αυτή είναι μια εφαρμογή όπου συνεργάζονται πολλοί χρήστες και η οποία μιμείται ένα σύστημα πληροφοριών για ασθενείς που μπορεί να προσπελαστεί από νοσηλευτές και γιατρούς. Υποστηρίζει μια σειρά από απαιτήσεις [38]: η εφαρμογή επιτρέπει στους γιατρούς να δημιουργήσουν διαφορετικά είδη εκθέσεων για τους ασθενείς. Έπειτα, η εφαρμογή υποστηρίζει διάφορα είδη στοιχείων κι εκθέσεων για τους ασθενείς που εισάγονται σε διαφορετικούς θαλάμους του νοσοκομείου. Για μια νοσοκόμα, η πρόσβαση σε εκθέσεις γιατρών επιτρέπεται μόνο εφόσον κάποιος γιατρός είναι παρόν στο θάλαμο όπου η νοσοκόμα βρίσκεται κι επίσης αν ο γιατρός έχει δώσει εκ των προτέρων έγκριση για μια τέτοια πρόσβαση. Μια νοσοκόμα μπορεί να προσπελάσει και να ενημερώσει τα στοιχεία του ασθενούς μόνο όταν είναι παρούσα στο θάλαμο του ασθενούς.

Στην εικόνα 43, παρουσιάζεται η σχηματική αναπαράσταση της δραστηριότητας πληροφοριών για ασθενείς. Στο χώρο αυτού αντικειμένου της δραστηριότητας, ορίζονται δύο αντικείμενα: DoctorReports και LocationService. Το αντικείμενο DoctorReports αναφέρεται στην υπηρεσία βάσης δεδομένων που κρατά τα στοιχεία των ασθενών. Το αντικείμενο LocationService αναφέρεται στην υπηρεσία που παρέχει πληροφορίες για την τοποθεσία των νοσοκόμων και των γιατρών στο νοσοκομείο. Ύστερα, ορίζονται σ' αυτή τη δραστηριότητα, δύο ρόλοι, η νοσοκόμα (Nurse) κι ο γιατρός (Doctor). Όπου στο ρόλο Nurse, ορίζονται δύο ιδιωτικά αντικείμενα: CurrentWard και WardDB. Ένα ξεχωριστό στιγμιότυπο αυτών των αντικειμένων δημιουργείται για κάθε μέλος του ρόλου Nurse. Το αντικείμενο CurrentWard αναφέρεται στην υπηρεσία της περιβάλλουσας κατάστασης του θαλάμου, στον οποίο είναι παρόν μια νοσοκόμα. Το αντικείμενο WardDB αναφέρεται σε μια υπηρεσία που παρέχει πρόσβαση στη βάση δεδομένων, η οποία με τη σειρά της κρατά πληροφορίες για τους ασθενείς ενός συγκεκριμένου θαλάμου. Οι συνδέσεις των δύο αυτών αντικειμένων αλλάζουν καθώς η νοσοκόμα μετακινείται από ένα θάλαμο σε άλλο. Έτσι, ορίζονται οι ακόλουθες λειτουργίες του ρόλου Nurse. Η λειτουργία AccessDoctorReports επιτρέπει σε μια νοσοκόμα να προσπελαύνει τις εκθέσεις/αναφορές των γιατρών. Η λειτουργία AccessPatientInfo επιτρέπει σε μια νοσοκόμα να προσπελαύνει τα στοιχεία των ασθενών που εισάγονται σ' ένα συγκεκριμένο θάλαμο.

Όσον αφορά το ρόλο Doctor, ορίζονται οι ακόλουθες λειτουργίες: η λειτουργία CreatePatientReports επιτρέπει στο γιατρό να δημιουργεί διάφορες εκθέσεις σχετικά με τους ασθενείς. Ακόμα, η λειτουργία ApprovePatientDataAccess επιτρέπει στον γιατρό να δώσει την έγκριση σε μια συγκεκριμένη νοσοκόμα, για πρόσβαση σε ιατρικές εκθέσεις.



Εικόνα 43 - Δραστηριότητα συστήματος πληροφοριών για ασθενείς

5.4.2.1 Μηχανισμός ContextGuard στο σύστημα πληροφοριών ασθενών

Μία από τις απαιτήσεις αυτής της εφαρμογής είναι ότι μια νοσοκόμα θα πρέπει να έχει πρόσβαση στις εκθέσεις των γιατρών, μόνο όταν η ίδια και κάποιος γιατρός είναι παρόντες στον ίδιο θάλαμο. Όμως, κατά τη διάρκεια που η νοσοκόμα έχει πρόσβαση στις εκθέσεις των γιατρών, μπορεί να φύγει από το θάλαμο ή ακόμα και να φύγει ο ίδιος ο γιατρός από το θάλαμο. Σ' αυτή την περίπτωση, η πρόσβαση της νοσοκόμας στις εκθέσεις των γιατρών, θα πρέπει να ανακληθεί. Η απαίτηση αυτή έχει προγραμματιστεί με το μηχανισμό ContextGuard, όπως φαίνεται στην παρακάτω εικόνα 44. Μια φρούρηση περιβάλλουσας κατάστασης έχει καθοριστεί με τη λειτουργία AccessDoctorReports. Σ' αυτό το παράδειγμα, η εκτέλεση της φρούρησης ενεργοποιείται κάθε φορά που η RoomStatusChangeEvent κοινοποιείται από τον χειριστή ρόλου. Η συνθήκη φρούρησης ελέγχει ότι κάποιος γιατρός είναι παρόν στο θάλαμο όπου βρίσκεται η νοσοκόμα. Αυτό γίνεται με την επερώτηση του αντικείμενου CurrentWard με τη λίστα των χρηστών που βρίσκονται στο ρόλο Doctor. Το προγραμματιστικό μοντέλο παρέχει τη λειτουργία members(RoleName), η οποία με τη σειρά της επιστρέφει τη λίστα των χρηστών του ρόλου RoleName. Εάν δεν τηρείται η συνθήκη φρούρησης, η συνεδρία τερματίζεται, κι έτσι ανακαλείται η πρόσβαση της νοσοκόμας.

```

Role Nurse {
  Operation AccessDoctorReports {
    Precondition
      CurrentWard.isPresent(thisUser) ^
      CurrentWard.isPresent(members(Doctor))
    Action DoctorReports InvokeSession accessReports
  }
  ContextGuard {
    When Event RoomStatusChangeEvent
    GuardCondition
      CurrentWard.isPresent(thisUser) ^
      CurrentWard.isPresent(members(Doctor))
  }
}

```

Εικόνα 44 - Παράδειγμα φρούρησης περιβάλλουσας κατάστασης

5.4.2.2 Χειρισμός εξαιρέσεων σε επίπεδο ενέργειας στο σύστημα πληροφοριών ασθενών

Στην παρακάτω εικόνα 45 παρουσιάζεται ένα παράδειγμα χειρισμού εξαιρέσεων σε επίπεδο ενέργειας. Η λειτουργία AccessDoctorReports του ρόλου Nurse μπορεί να αντιμετωπίσει μια εξαίρεση ServiceInteractionException κατά τη διάρκεια των αλληλεπιδράσεών της με την υπηρεσία βάσης δεδομένων των ασθενών. Για το χειρισμό τέτοιων καταστάσεων, όπως είπαμε και παραπάνω, μπορεί κανείς να προγραμματίσει ένα χειριστή εξαιρέσεων που καλεί τη συγκεκριμένη μέθοδο για την εφεδρική υπηρεσία της βάσης δεδομένων.

```

Role Nurse {
  Operation AccessDoctorReports
  Action InvokeMethod DoctorReports.accessReports()
  OnException ServiceInteractionException
  Action InvokeMethod
    DoctorReportsBackup.accessReports()
}

```

Εικόνα 45 - Παράδειγμα χειρισμού εξαιρέσεων σε επίπεδο ενεργειών

5.4.2.3 Χειρισμός εξαιρέσεων σε επίπεδο λειτουργίας στο σύστημα πληροφοριών ασθενών

Όπως είδαμε προηγουμένως, η απαίτηση της πρόσβασης μιας νοσοκόμας στις εκθέσεις των γιατρών είχε προγραμματιστεί με χρήση του μηχανισμού Context guard. Για την ανάκτηση από μια τέτοια αποτυχία, προσθέτουμε ένα χειριστή εξαίρεσης ContextInvalidationException για αυτή τη λειτουργία, όπως φαίνεται στην εικόνα 46. Ένας τρόπος για το χειρισμό αυτής της απαίτησης είναι να επιτραπεί στη νοσοκόμα η συνέχιση της πρόσβασης στις εκθέσεις των γιατρών αφότου πάρει έγκριση από κάποιο γιατρό. Συγκεκριμένα, όσον αφορά τη λειτουργία AccessDoctorReports, δημιουργήθηκε το συμβάν RequestPermissionEvent. Αυτό το γεγονός μπορεί να πραγματοποιηθεί μέσω του ρόλου Doctor, όπου μπορεί να χρησιμοποιηθεί για να

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

επιτρέψει την εκτέλεση μιας λειτουργίας μέσω της οποίας ένας γιατρός μπορεί να χορηγήσει έγκριση στην αρμόδια νοσοκόμα.

```

Role Nurse {
  Operation AccessDoctorReports {
    Precondition
      CurrentWard.isPresent(thisUser) ^
      CurrentWard.isPresent(members(Doctor))
    Action DoctorReports InvokeSession accessReports
  }
  ContextGuard {
    When Event RoomStatusChangeEvent
    GuardCondition
      CurrentWard.isPresent(thisUser) ^
      CurrentWard.isPresent(members(Doctor))
  }
  OnException ContextInvalidationException {
    Action NotifyEvent RequestPermissionEvent(requestor=thisUser)
  }
}

```

Εικόνα 46 - Χειριστής εξαιρέσεων ακύρωσης περιβάλλουσας κατάστασης

Κεφάλαιο 6 Συμπεράσματα

Στη παρούσα εργασία περιγράφηκαν τέσσερις προσεγγίσεις, οι οποίες έδωσαν έμφαση στην περιβάλλουσα κατάσταση και ιδιαίτερα στην προσαρμοστικότητα της. Ακολούθως, περιγράφονται τα συμπεράσματα που εξήχθησαν, από τη μελέτη των μεθοδολογιών που αναλύθηκαν νωρίτερα.

Όσον αφορά τη πρώτη προσέγγιση, η εστίασή της βρίσκεται στην προσαρμοστικότητα με βάση την περιβάλλουσα κατάσταση και αναδείχτηκε ο τρόπος με τον οποίο απαιτείται αύξηση της εκφραστικής δύναμης των μοντέλων εφαρμογών του Διαδικτύου, έτσι ώστε να ενσωματώνονται οι αλλαγές στη λογική παραγωγής σελίδας που εξαρτώνται από τη περιβάλλουσα κατάσταση. Αρχικά, γίνεται μια αναφορά στη πρόσβαση μέσω πολλαπλών καναλιών σε υπηρεσίες και εφαρμογές, η οποία κερδίζει όλο και περισσότερο έδαφος μεταξύ τόσο των καταναλωτών και των παρόχων. Οι μεν καταναλωτές προσελκύονται όλο και περισσότερο από φορητές συσκευές, εξοπλισμένες με έγχρωμες οθόνες υψηλής ανάλυσης, που είναι ικανές να παρέχουν δυνατότητες περιήγησης όπως ακριβώς και οι παραδοσιακοί επιτραπέζιοι υπολογιστές. Από την άλλη πλευρά, οι πάροχοι διευκολύνονται από την ανάπτυξη τυπικών πρωτοκόλλων επικοινωνίας (όπως είναι το HTTP) και γλωσσών σήμανσης (όπως είναι η HTML), που υποστηρίζονται από τέτοιου είδους συσκευές και συστήματα. Επομένως, στην προσέγγιση αυτή γίνεται προσπάθεια να συνδυαστεί η περιβάλλουσα κατάσταση με τη πρόσβαση πολλαπλών καναλιών σε υπηρεσίες και εφαρμογές, σ' ένα εννοιολογικό πλαίσιο εργασίας, το οποίο θα παρέχει διευκολύνσεις για τη μοντελοποίηση εφαρμογών Διαδικτύου γνώσης όσον αφορά τη περιβάλλουσα κατάσταση και τα πολλαπλά κανάλια.

Επιπρόσθετα, είδαμε ότι η περιβάλλουσα κατάσταση μπορεί να χρησιμοποιείται για την επίτευξη πιο αποτελεσματικών αλληλεπιδράσεων σε όλες εκείνες τις καταστάσεις όπου το περιεχόμενο και οι υπηρεσίες που προσφέρονται από την εφαρμογή, εξαρτώνται από τις τρέχουσες καταστάσεις του περιβάλλοντος. Τέτοιες καταστάσεις του περιβάλλοντος είναι η προσαρμοστική προσωποποίηση, οι λειτουργικές ανάγκες, ο χειρισμός εξαιρέσεων, οι λειτουργίες που επιτρέπουν αλληλεπιδράσεις, καθώς και η ενισχυμένη αποτελεσματικότητα.

Πιο αναλυτικά, μπορούμε να αναφέρουμε ότι η μέθοδος αυτή παρέχει έννοιες, συμβολισμούς και τεχνολογίες υλοποίησης για την ανάπτυξη εφαρμογών του Διαδικτύου, χωρίς να παρουσιάζει κάποιο περιορισμό και χωρίς να απαιτεί λύσεις από τη μεριά είτε του πελάτη είτε του εξυπηρετή. Εκμεταλλεύεται το βασισμένο σε μοντέλα υπόδειγμα ανάπτυξης, το οποίο παρέχει υψηλού επιπέδου κατασκευές για τον καθορισμό της δομής και της συμπεριφοράς της εφαρμογής. Λόγω του τυπικού ορισμού της σημασιολογίας, οι παραγόμενες προδιαγραφές μπορούν να μετασχηματιστούν σε εκτελέσιμο κώδικα, με τη βοήθεια διαφόρων ενοποιημένων τεχνικών, διευκολύνοντας έτσι τη συνολική διαδικασία ανάπτυξης. Επιπρόσθετα, η προσέγγιση αυτή αντλεί στοιχεία από το υπόδειγμα ανάπτυξης με επίκεντρο τα δεδομένα, γεγονός που ενισχύει την αναπαράσταση της προσωποποίησης και των μετά-δεδομένων του πλαισίου στα δεδομένα της εφαρμογής. Αποτέλεσμα του παραπάνω είναι η ανάπτυξη εφαρμογών του Διαδικτύου όπου η προσαρμογή του περιεχομένου και η συμπεριφορά στις αντιδράσεις του πλαισίου επιτυγχάνονται με δυναμικό τρόπο.

Η πρώτη προσέγγιση βασίζεται στο μοντέλο της WebML, επιτρέποντας μια πλήρως αυτόματη παραγωγή εφαρμογών προσαρμοστικότητας. Παρ' όλα αυτά, οι προτεινόμενες βασικές έννοιες της σχεδίασης είναι γενικής φύσεως, κι έτσι η μοντελοποίηση περιβάλλουσας κατάστασης, οι αλλαγές προσαρμοστικότητας, η μεταβίβαση παραμέτρων, καθώς και η χρήση της ανανέωσης, μπορούν να κωδικοποιηθούν από τους προγραμματιστές των εφαρμογών διαδικτύου.

Ως συνέχεια της παραπάνω προσέγγισης, ορίστηκε ένα τυπικό μοντέλο για την ανάλυση των πιθανών συγκρούσεων και αντιφάσεων που μπορεί να προκληθούν από την ενεργοποίηση των ενεργειών προσαρμοστικότητας. Μελετήθηκε, επίσης, ο τρόπος με τον οποίο οι εφαρμογές διαδικτύου για καθορισμένες συσκευές, θα μπορούσαν να δημοσιευθούν (με τη χρήση της νέας έννοιας υπηρεσίας του WebML), έτσι ώστε να καταστεί δυνατός ο καθορισμός της διαχείρισης των μετά-δεδομένων για τα δεδομένα περιβάλλουσας κατάστασης. Επιπλέον, μελετήθηκαν οι τρόποι προσαρμοστικότητας, οι οποίοι είτε εξαρτώνται από τον χρόνο (δηλαδή η καθυστέρηση ανανέωσής τους μπορεί να μοντελοποιηθεί σύμφωνα με τις δοθείσες πολιτικές) ή είναι πραγματικού χρόνου (δηλαδή η ενεργοποίηση της αντίδρασής τους είναι άμεση).

Στη δεύτερη προσέγγιση περιγράφηκε μια προσέγγιση για τον σχεδιασμό και την υλοποίηση προσαρμοστικών λειτουργικών συστημάτων πληροφορίας περιβάλλουσας κατάστασης, λαμβάνοντας υπόψη τις σχέσεις συστήματος - περιβάλλουσας κατάστασης άμεσα, από το σχεδιασμό του συστήματος στην υλοποίηση, και έπειτα στην εκτέλεση του συστήματος. Εν αντιθέσει με τις πρόσφατες προσεγγίσεις που αντιμετωπίζουν ξεχωριστά τη πληροφορία περιβάλλουσας κατάστασης και την αυτό-προσαρμοστικότητα, η συγκεκριμένη προσέγγιση τις μελετάει σε συνδυασμό.

Αρχικά, συζητήθηκαν τα κίνητρα για τη διεξαγωγή της συγκεκριμένης έρευνας. Στη συνέχεια περιγράφηκε το μοντέλο της προσέγγισης και περιγράφηκε το εργαλείο που χρησιμοποιήθηκε στην ανάπτυξη προσαρμοστικών συστημάτων πληροφορίας περιβάλλουσας κατάστασης. Τέλος, αναλύθηκε ένα παράδειγμα εφαρμογής με τη χρήση της προσέγγισης, και συγκεκριμένα τα στάδια της σχεδίασης και υλοποίησης προσαρμοστικών συστημάτων πλοήγησης οχημάτων, δίνοντας έμφαση στη πληροφορία περιβάλλουσας κατάστασης.

Η συνεισφορά της συγκεκριμένης μεθόδου, όπως αναλύθηκε στη σχετική ενότητα είναι:

- Απλοποίηση της σχεδίασης και υλοποίησης του συστήματος μέσω ενός νέου προσαρμοστικού συστήματος μοντέλου στοιχείων περιβάλλουσας κατάστασης και ενός μετά-μοντέλου λογισμικού εκτέλεσης που κάνει δυνατή την παραγωγή της υλοποίησης του συστήματος.
- Η μείωση των πιθανών σφαλμάτων κατά την ανάπτυξη του συστήματος, λαμβάνοντας υπόψη τις σχέσεις συστήματος-περιβάλλουσας κατάστασης στο επίπεδο σχεδίασης.
- Αναπαράσταση του χρόνου εκτέλεσης των σχέσεων συστήματος-περιβάλλουσας κατάστασης, ώστε να κάνει δυνατή την προσαρμογή και εξέλιξή τους κατά το χρόνο εκτέλεσης.

- Δυνατότητα οπτικού ελέγχου και εντοπισμού των παραλειπόμενων-λανθασμένων κανόνων προσαρμογής.
- Παροχή ενός προσαρμόσιμου μοντέλου περιβάλλουσας κατάστασης για την μείωση της εκ των υστέρων επίβλεψης.
- Μείωση της πολυπλοκότητας του συστήματος, μέσω του διαχωρισμού της διαχείρισης περιβάλλουσας κατάστασης από το λειτουργικό σύστημα και τη διαχείρισή του.

Τέλος, αξίζει να αναφερθεί ότι η συγκεκριμένη μεθοδολογία μπορεί να ενισχυθεί ποικιλοτρόπως. Καταρχήν με την επέκταση της προσέγγισης με μια μέθοδο που να εξασφαλίζει την ορθότητα του συστήματος. Ακολουθώς, τα λειτουργικά συστήματα (functional systems) συνήθως αναπτύσσονται σε περιβάλλον που δεν είναι απολύτως προβλεπόμενο κατά το χρόνο σχεδίασης του συστήματος, άρα είναι πολύ σημαντικό να αναδειχτεί το πώς θα ενσωματώσουν τις απρόβλεπτες αλλαγές της περιβάλλουσας κατάστασης. Τέλος, είναι καθοριστικής σημασίας η προσέγγιση να επεκταθεί κατά τέτοιον τρόπο, ώστε να έχει τη δυνατότητα να αντεπεξέρχεται στις απαιτούμενες αλλαγές όταν το σύστημα είναι σε λειτουργία.

Στην τρίτη προσέγγιση παρουσιάστηκε ο σχεδιασμός του Adaptation Manager στα πλαίσια του οποίου θα πραγματοποιηθεί η προσαρμογή μιας εφαρμογής κινητού εμπορίου. Η μεθοδολογία που αναπτύχθηκε, θέλησε να αντιμετωπίσει τα προβλήματα που δημιουργούνται κατά την ανάμειξη του κώδικα προσαρμογής με τον κώδικα του πυρήνα εφαρμογής. Πιο συγκεκριμένα, η προσέγγιση θέλησε να καλύψει τα κενά που δημιουργούνται από την αύξηση του βαθμού ένωσης στον κώδικα εφαρμογής, από τη στιγμή που η κύρια λογική εφαρμογής και η λογική προσαρμογής αναμειγνύονται. Ένα άλλο πρόβλημα που θέλησε να αντιμετωπίσει είναι, η αυξανόμενη δυσκολία διατήρησης των ενεργειών, από τη στιγμή που οι μετατροπές στη λογική εφαρμογής θα δημιουργήσουν πιθανώς αλλαγές στην λογική προσαρμογής και το αντίστροφο. Ένα ακόμα επιδιωκόμενο αποτέλεσμα ήταν η δυνατότητα για την επαναχρησιμοποίηση των συνιστωσών είτε από την κύρια εφαρμογή, είτε από τον κώδικα προσαρμογής, από τη στιγμή που αναμειγνύονται και προσαρμόζονται για το συγκεκριμένο. Επιπροσθέτως, πολλές από τις τεχνικές που χρησιμοποιούνται, σε πολλές περιπτώσεις διαχειρίζονται μόνο ένα μικρό υποσύνολο της περιβάλλουσας κατάστασης, χωρίς να παρέχουν επαρκή ερείσματα για την διεύρυνση του παράγοντα περιβάλλουσας κατάστασης που λαμβάνεται υπόψη. Σε άλλες περιπτώσεις, δεν γίνεται χρήση των αρχείων προσαρμοστικής πολιτικής, τα οποία παρέχουν πιο ευέλικτη προσαρμογή.

Έτσι λοιπόν, αφού αναλύθηκε η δομή μιας εφαρμογής κινητού εμπορίου και περιγράφηκε ο τρόπος με τον οποίο θα πρέπει να δομηθεί ώστε να επιδέχεται προσαρμογή, προτάθηκε μια aspect-oriented προσέγγιση προκειμένου να συνυφανθεί ο κώδικας που υλοποιεί την προσαρμοστικότητα στη βασική εφαρμογή. Με αυτόν τον τρόπο επετεύχθη η ενθυλάκωση της λογικής για την προσαρμογή της συμπεριφοράς κατά το χρόνο εκτέλεσης μιας εφαρμογής σε μία συνιστώσα. Η συνιστώσα αυτή είναι η μηχανή για την πυροδότηση των κανόνων προσαρμογής, καθένας από τους οποίους αποτελείται από μια συνθήκη που όταν ικανοποιηθεί θα λάβει χώρα μια προσαρμογή και μια δράση η οποία υποδεικνύει την κατάλληλη αντίδραση προσαρμογής.

Το πλεονέκτημα της προσέγγισης που αναλύθηκε, είναι ο μεγάλος βαθμός ανεξαρτησίας, των εμπλεκομένων της κύριας εφαρμογής και των εμπλεκομένων της εφαρμογής προσαρμογής, η δυνατότητα εύκολης αλλαγής του κώδικα των κλάσεων της κύριας εφαρμογής και η δυνατότητα επαναχρησιμοποίησης των καθώς και η ευκολία συντήρησης και ελέγχου του κώδικα προσαρμογής.

Όσον αφορά την τέταρτη προσέγγιση, παρουσιάστηκε η δημιουργία ενός προγραμματιστικού πλαισίου εργασίας για την ανάπτυξη εφαρμογών με επίγνωση της περιβάλλουσας κατάστασης, το οποίο εισάγει ένα μοντέλο ανάκαμψης και επιτρέπει τον ορισμό ενεργειών ανάκαμψης σε επίπεδο εφαρμογής για την αντιμετώπιση αποτυχιών. Όπως παρατηρήθηκε, το κύριο χαρακτηριστικό της προσέγγισης αυτής είναι η ενσωμάτωση του χειρισμού γεγονότων σε επίπεδο αντικειμένου με ένα χειριστή εξαιρέσεων σε επίπεδο ρόλου, ώστε να δημιουργηθούν εύρωστες εφαρμογές με επίγνωση της περιβάλλουσας κατάστασης. Επιπροσθέτως δημιουργήθηκε ένας νέος μηχανισμός (με τη μορφή διεπαφής εξαιρέσεων για τους ρόλους) που παρέχει τη δυνατότητα στους χρήστες να χειριστούν τις εξαιρέσεις.

Πιο συγκεκριμένα, έχουμε τον σχεδιασμό μιας εφαρμογής με επίγνωση της περιβάλλουσας κατάστασης που προσδιορίζεται με χρήση ενός προγραμματιστικού μοντέλου με καθορισμένο τομέα, το οποίο εκφράζεται σε γλώσσα XML. Στο μοντέλο σχεδιασμού, μια εφαρμογή με επίγνωση της περιβάλλουσας κατάστασης, προγραμματίζεται με χρήση μια δραστηριότητας, όπου με τη σειρά της, μια δραστηριότητα ορίζει ένα κοινό χώρο αντικειμένων κι ένα σύνολο από ρόλους του χρήστη. Επιπλέον, μια δραστηριότητα μπορεί να επεκταθεί σε πολλές διαφορετικές φυσικές περιοχές και μπορεί να περιλαμβάνει πολλαπλούς χρήστες που επιτελούν σε συνεργασία διάφορες εργασίες και καθήκοντα. Διάφοροι πόροι και υπηρεσίες που απαιτούνται από την εφαρμογή, προσπελούνται μέσω αντικειμένων που ορίζονται στη δραστηριότητα. Ένα αντικείμενο με τη σειρά του, συνδέεται με διαφορετικές υπηρεσίες και πόρους υπό διαφορετικές συνθήκες περιβάλλουσας κατάστασης. Επίσης, μια δραστηριότητα παρέχει ρόλους για την αναπαράσταση των δικαιωμάτων των χρηστών ώστε να εκτελούν διάφορες εργασίες της εφαρμογής. Οι ρόλοι επιπρόσθετα, αποτελούν το κύριο μηχανισμό για τον έλεγχο πρόσβασης σε μια δραστηριότητα. Ο μηχανισμός λειτουργίας παρέχεται με το ρόλο, μέσω του οποίου οι χρήστες μπορούν να εκτελούν εργασίες της εφαρμογής. Αποτελεί επίσης το μηχανισμό μέσω του οποίου οι χρήστες μπορούν να έχουν πρόσβαση σε υπηρεσίες ενεργού χώρου μέσα από μια αλληλεπιδραστική συνεδρία.

Επιπλέον, υποστηρίζονται δύο μοντέλα για την εκτέλεση των ενεργειών στο πλαίσιο μιας λειτουργίας ενός ρόλου, το μοντέλο κλήσης μεθόδου -όπου μια ενέργεια περιλαμβάνει μόνο μία κλήση μιας μεθόδου σ' ένα αντικείμενο- και το μοντέλο αλληλεπίδρασης συνεδρίας -όπου περιλαμβάνεται η κλήση των αποτελεσμάτων μιας ενέργειας στην έναρξη μιας αλληλεπιδραστικής συνεδρίας μεταξύ του χρήστη κι ενός αντικειμένου. Παράλληλα, εισάγεται ένα επίπεδο ενδιάμεσου λογισμικού και το οποίο παρέχει τρία γενικά στοιχεία-διαχειριστές: ένα διαχειριστή δραστηριότητας, ένα διαχειριστή ρόλου κι ένα διαχειριστή αντικειμένου. Το περιβάλλον εκτέλεσης μιας εφαρμογής κατασκευάζεται εξειδικεύοντας αυτούς τους διαχειριστές με βάση τις προδιαγραφές μιας συγκεκριμένης δραστηριότητας. Για κάθε αντικείμενο δημιουργείται ένας διαχειριστής αντικειμένου και κάθε τέτοιος διαχειριστής διατηρεί μια αναφορά της υπηρεσίας με την οποία συνδέεται αυτή τη στιγμή.

Οι εφαρμογές που παρουσιάστηκαν για την απεικόνιση των θεμάτων αυτής της προσέγγισης περιλαμβάνουν ένα σύστημα αναπαραγωγής μουσικής με επίγνωση της περιβάλλουσας κατάστασης κι ένα σύστημα πληροφοριών για ασθενείς με επίγνωση της περιβάλλουσας κατάστασης.

Βιβλιογραφία

1. M. Hussein, J. Han and A. Colman, *TR: C3-516_02: Integrated Modeling of Context-Aware Adaptive Software Systems*, Oct. 2010, Swinburne University of Technology, p. 1-25.
2. Salehie, M. and L. Tahvildari, *Self-adaptive software: Landscape and research challenges*. ACM Trans. Auton. Adapt. Syst., 2009. 4(2): p. 1-42.
3. Cheng, B.H., et al., *Software Engineering for Self-Adaptive Systems: A Research Roadmap*, in *Software Engineering for Self-Adaptive Systems*. 2009, Springer-Verlag. p. 1-26.
4. Zhang, J. and B.H.C. Cheng, *Model-based development of dynamically adaptive software*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai,China. p. 371-380.
5. Baldauf, M., S. Dustdar, and F. Rosenberg, *A survey on context-aware systems*. Int. J. Ad Hoc Ubiquitous Comput., 2007. 2(4): p. 263-277.
6. Bettini, C., et al., *A survey of context modelling and reasoning techniques*. Pervasive Mob. Comput., 2010. 6(2): p. 161-180.
7. Garlan, D., R. Monroe, and D. Wile, *Acme: an architecture description interchange language*, in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*.1997, IBM Press: Toronto, Ontario, Canada. p. 7.
8. Magee, J., et al., *Specifying distributed software architectures*. Software Engineering-ESEC'95, 1995: p. 137-153.
9. Heaven, W., et al., *A Case Study in Goal-Driven Architectural Adaptation*, in *Software Engineering for Self-Adaptive Systems*. 2009, Springer-Verlag. p. 109-127.
10. Floch, J., et al., *Using Architecture Models for Runtime Adaptability*. IEEE Softw., 2006. 23(2): p. 62-70.
11. Morin, B., et al., *Taming Dynamically Adaptive Systems using models and aspects*, in *Proceedings of the 31st International Conference on Software Engineering*. 2009, IEEE Computer Society. p. 122-132.
12. Garlan, D., et al., *Rainbow: architecture-based self-adaptation with reusable infrastructure*. Computer, 2004. 37(10): p. 46-54.
13. Sheng, Q.Z. and B. Benatallah, *ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development*, in *Proceedings of the International Conference on Mobile Business*. 2005, IEEE Computer Society. p. 206-212.
14. Rouvoy, R., et al., *MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments*, in *Software Engineering for Self-Adaptive Systems*, B. Cheng, et al., Editors. 2009, Springer Berlin / Heidelberg. p. 164-182.
15. Taconet, C., et al., *CA3M: A Runtime Model and a Middleware for Dynamic Context Management*, in *Proceedings of the Confederated International*

Προσεγγίσεις στην υλοποίηση προσαρμοστικών υπηρεσιών στα πλαίσια του ηλεκτρονικού εμπορίου

Μεταπτυχιακή εργασία

Φούντας Γεώργιος

- Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part I.* 2009, Springer-Verlag: Vilamoura, Portugal. p. 513-530.
16. Stefano Ceri, Florian Daniel, Maristella Matera, Federico M. Facca, *Model-driven Development of Context-Aware Web Applications*, ACM Transactions on Internet Technology, Volume 7 Issue 1, February 2007.
 17. Ceri, S., Fraternali, P., and Matera, M. 2002. *Conceptual Modeling of Data-Intensive Web Applications*. IEEE Internet Computing 6, 4 (July-August), 20-30.
 18. Schwabe, D., Guimaraes, R., and Rossi, G. 2002. *Cohesive Design of Personalized Web Applications*. IEEE Internet Computing 6, 2 (March-April), 34-43.
 19. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., and Matera, M. 2002. *Designing Data-Intensive Web Applications*. Morgan Kaufmann.
 20. Ceri, S., Fraternali, P., Bongio, A., Butti, S., Acerbis, R., Tagliasacchi, M., Toffetti, G., Conserva, C., Elli, R., Ciapessoni, F., and Greppi, C. 2003. *Architectural Issues and Solutions in the Development of Data-Intensive Web Applications*. In Proceedings of CIDR 2003, January 2003, Asilomar, CA, USA.
 21. L. Capra, "Reflective Mobile Middleware for Context-Aware Applications" in Department of Computer Science: University of London, 2003.
 22. Pressman, R. S. *Software Engineering - A Practitioner's Approach - Seventh Edition*. McGraw-Hill, 2009. ISBN 0073375977
 23. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai and M. Matera. *Designing Data-Intensive Web Applications*. Morgan Kaufmann Publishers, 2003, ISBN: 1-55860-843-5.
 24. G. L. Lohse, P. Spiller. *Internet retail store design: How the user interface influences traffic and sales*. Journal of Computer-Mediated Communication, Volume 5, Issue 2, December 1999.
 25. R. Hennicker and N. Koch. A UML-based Methodology for Hypermedia Design. Proceedings of the Unified Modeling Language Conference (UML'2000), LNCS 1939, Springer Verlag, 2000.
 26. J. C. Preciado, M. Linaje, F. Sanchez-Figueroa, G. Zhang, C. Kroiß and N. Koch. *Designing Rich Internet Applications Combining UWE and RUX-Method*. Proceedings of the International Conference on Web Engineering (ICWE 2008), IEEE, New York, USA, 2008.
 27. R. Vdovjak, F. Frasinjar, G.-J. Houben, and P. Barna. *Engineering Semantic Web Information Systems in Hera*. Journal of Web Engineering (2003) Volume: 2, Issue: 1-2, Publisher: Rinton Press, Pages: 3-26
 28. K. van der Sluijs, G.-J. Houben, J. Broekstra and S. Casteleyn. *Hera-S - Web Design Using Sesame*. Proceedings of the 6th International Conference on Web Engineering (ICWE 2006), 11-14 July 2006, Palo Alto, California, USA, pp. 337 - 345, ACM.

29. M. Roman, C.K. Hess, R. Cerqueira, A. Ranganathan, R.H. Campbell, and K. Nahrstedt, "Gaia: A Middleware Platform for Active Spaces," *Mobile Computing and Comm. Rev.*, vol. 6, no. 4, pp. 65-67, 2002.
30. D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project Aura: Toward Distraction-Free Pervasive Computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22-31, Apr.-June 2002.
31. N. Davies, K. Cheverst, K. Mitchell, and A. Efrat, "Using and Determining Location in a Context-Sensitive Tour Guide," *Computer*, vol. 34, no. 8, pp. 35-41, Aug. 2001.
32. B. Johanson, A. Fox, and T. Winograd, "The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms," *IEEE Pervasive Computing Magazine*, vol. 1, no. 2, pp. 67-74, Apr.- June 2002.
33. M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rajani, and M. Spasojevic, "From Informing to Remembering: Ubiquitous Systems in Interactive Museums," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 13-21, Apr. 2002.
34. S. Consolvo, P. Roessler, B.E. Shelton, A. LaMarca, B. Schilit, and S. Bly, "Technology for Care Networks of Elders," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 22-29, Apr.-June 2004.
35. A. Tripathi, D. Kulkarni, and T. Ahmed, "A Specification Model for Context-Based Collaborative Applications," *Elsevier J. Pervasive and Mobile Computing*, vol. 1, no. 1, pp. 21-42, May/June 2005.
36. D. Kulkarni and A. Tripathi, "Application-Level Recovery Mechanisms for Context-Aware Pervasive Computing," *Proc. IEEE Symp. Reliable Distributed Systems*, pp. 13-22, 2008.
37. D. Kulkarni and A. Tripathi, "Context-Aware Role-Based Access Control in Pervasive Computing Systems," *Proc. 13th ACM Symp. Access Control Models and Technologies*, pp. 113-122, 2008.
38. M. Evered and S. Bogeholz, "A Case Study in Access Control Requirements for a Health Information System," *Proc. Second Workshop Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation*, pp. 53-61, 2004.
39. F. Halasz and M. Schwartz. "The Dexter Reference Model". In *Proc. NIST Hypertext Standardization Workshop*, pp. 95-133, 1990.
40. F. Halasz and M. Schwartz. "The Dexter Hypertext Reference Model". *Communications of the ACM*, Vol. 37, nr. 2, pp. 30-39, 1994.