

Σχεδιασμός και υλοποίηση
μηχανισμών διαλειτουργικότητας
τουριστικών υπηρεσιών σύμφωνα με
τις προδιαγραφές της ebXML και του
Open Travel Alliance

Αλεξόπουλος Δημήτρης

Μία εργασία που παρουσιάστηκε στο Πολυτεχνείο Κρήτης σε
εκπλήρωση των απαιτήσεων για την απόκτηση Διπλώματος στο Τμήμα
Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών

Πολυτεχνείο Κρήτης

Χανιά, Ελλάδα 2005

Περίληψη

Η μεγάλη διάδοση της τεχνολογίας στον χώρο του τουρισμού οδήγησε σταδιακά στην εμφάνιση πολύ μεγάλου αριθμού διαφορετικών συστημάτων, γεγονός που καθιστά δύσκολη την μεταξύ τους συνεργασία. Η επίτευξη της διαλειτουργικότητας μεταξύ όλων αυτών των διαφορετικών συστημάτων είναι μία από τις μεγαλύτερες προκλήσεις που καλείται να αντιμετωπίσει η σύγχρονη τεχνολογία. Η επίτευξη του στόχου αυτού θα δώσει μεγάλη ώθηση στην τουριστική βιομηχανία, δίνοντας την δυνατότητα ανάπτυξης νέων καινοτόμων υπηρεσιών που θα αυξήσουν τον βαθμό ικανοποίησης του πελάτη. Επίσης θα δώσει τη δυνατότητα στις τουριστικές επιχειρήσεις να βελτιώσουν τις επιχειρησιακές διαδικασίες τους, να μειώσουν τα λειτουργικά τους έξοδα και να ανακαλύψουν νέες πηγές εσόδων.

Τόσο οι επιχειρήσεις που δραστηριοποιούνται στον χώρο του τουρισμού, όσο και ο ακαδημαϊκός κόσμος, καταβάλλουν συνεχώς προσπάθειες για να επιτύχουν τον δύσκολο στόχο της διαλειτουργικότητας των τουριστικών πληροφοριακών συστημάτων. Κάποιες πολλά υποσχόμενες νέες τεχνολογίες όπως η XML και οι υπηρεσίες διαδικτύου (Web Services) φαίνεται να ανοίγουν τον δρόμο προς την σωστή κατεύθυνση. Το γεγονός ότι πρόκειται για ανοιχτά πρότυπα και μπορεί να χρησιμοποιηθούν ανεξαρτήτως πλατφόρμας υλικού και λογισμικού, οδήγησαν στην ευρεία διάδοση τους από το σύνολο σχεδόν της βιομηχανίας λογισμικού.

Η ebXML είναι ένα προτεινόμενο διεθνές πρότυπο στην κατηγορία των πρωτοκόλλων για παροχή υπηρεσιών στο διαδίκτυο. Ανήκει στην κατηγορία των πρωτοκόλλων που αποσκοπούν στο να περιγράψουν την σειρά της ανταλλαγής μηνυμάτων που απαιτούνται για την ολοκλήρωση μιας συνεργασίας ή μιας υπηρεσίας, που συχνά αναφέρονται σαν πρωτόκολλα συγχρονισμού (coordination protocols). Η ebXML είναι ένα κάθετο πρωτόκολλο (vertical protocol) γιατί απευθύνεται σε με μια συγκεκριμένη περιοχή εφαρμογών (ηλεκτρονικό εμπόριο). Παρόλο που η ebXML είναι κάθετο πρωτόκολλο δεν είναι αρκετά εξειδικευμένη για να περιγράψει τη σημαντική των όρων που χρησιμοποιούνται σε μια συγκεκριμένη περιοχή του ηλεκτρονικού εμπορίου, όπως για παράδειγμα το ηλεκτρονικό εμπόριο στην περιοχή του τουρισμού. Η διαλειτουργικότητα των υπηρεσιών στην περιοχή του

τουρισμού θα αυξανόταν σημαντικά εάν τα μηνύματα που ανταλλάσσονται με τη χρήση της ebXML χρησιμοποιούσαν όρους με αυστηρά καθορισμένη σημαντική για την περιοχή του τουρισμού. Αυτό υποδεικνύει την ενσωμάτωση οντολογιών πεδίου (domain ontologies) με τα πρότυπα υπηρεσιών του διαδικτύου, ώστε τα μηνύματα που ανταλλάσσονται να αποδίδουν συντακτική και σημαντική γνώση στον παραλήπτη.

Ο Open Travel Alliance (OTA) είναι ένας αυτοχρηματοδοτούμενος μη κερδοσκοπικός οργανισμός που αναπτύσσει προδιαγραφές διαλειτουργικότητας για τον τουριστικό τομέα. Χρησιμοποιεί την XML Schema σαν γλώσσα περιγραφής των XML εγγράφων που μεταδίδονται μεταξύ διαφορετικών τουριστικών πληροφοριακών συστημάτων. Αποτελείται από τις μεγαλύτερες επιχειρήσεις και οργανισμούς στον χώρο της τουριστικής βιομηχανίας, συσσωρεύοντας έτσι σημαντικό όγκο γνώσης των αναγκών και απαιτήσεων της συγκεκριμένης αγοράς (αποτελώντας ουσιαστικά μια οντολογία για τον τουρισμό), και θέτει τις βάσεις ώστε να καθιερωθεί ως ένα καθολικά αποδεκτό πρότυπο διαλειτουργικότητας στην περιοχή του τουρισμού.

Με βάση τα παραπάνω, είναι ιδιαίτερα κρίσιμο για τα υπάρχοντα τουριστικά πληροφοριακά συστήματα να προσαρμοστούν στα σύγχρονα πρότυπα, ακολουθώντας προδιαγραφές διαλειτουργικότητας όπως αυτές της ebXML και του OTA. Η προσαρμογή αυτή μπορεί να αποτελέσει μια αρκετά δύσκολη διαδικασία. Είναι σημαντικό να αναζητηθούν μέθοδοι και τεχνικές με τις οποίες να τεθούν οι βάσεις για πιο ομαλή μετάβαση των υπαρχόντων συστημάτων από την σημερινή τους μορφή στο σύγχρονο διαλειτουργικό περιβάλλον που διαμορφώνεται. Αυτός είναι και ο στόχος της παρούσας διπλωματικής εργασίας:

- η μελέτη προτύπων συνεργασίας, διαλειτουργικότητας και παροχής υπηρεσιών στο διαδίκτυο βασισμένων σε διεθνή πρότυπα συνεργασιών (coordination protocols) όπως η ebXML,
- η μελέτη προδιαγραφών διαλειτουργικότητας του τουριστικού τομέα (OTA) που μπορούν να χρησιμοποιηθούν για την απόδοση συντακτικής και σημασιολογικής πληροφορίας (domain ontology) στις υπηρεσίες διαδικτύου για τον τουρισμό,
- Ο συνδυασμός των διεθνών προτύπων διαλειτουργικότητας με οντολογίες πεδίου γνώσης (domain ontologies) και συγκεκριμένα πρωτόκολλα του πεδίου

- γνώσης (όπως αυτά που παραδοσιακά παρείχε το EDI), ώστε η επικοινωνία, η συνεργασία και η διαλειτουργικότητα να είναι βασισμένη στη γνώση του πεδίου (domain knowledge) και όχι στη σύνταξη των XML εγγράφων,
- Η αξιολόγηση του συνολικού αποτελέσματος του συνδυασμού των διεθνών προτύπων υπηρεσιών με οντολογίες πεδίου γνώσης για αποτελεσματικότερη παροχή υπηρεσιών στο διαδίκτυο,
 - η μελέτη της προσαρμογής υπαρχόντων τουριστικών πληροφοριακών συστημάτων στις προδιαγραφές και μηχανισμούς διαλειτουργικότητας,
 - η ανάδειξη των σημαντικών χαρακτηριστικών των προδιαγραφών για την υλοποίηση σύνθετης λειτουργικότητας και
 - η ανάπτυξη μεθόδων και τεχνικών που θα διευκολύνει και θα επιταχύνει την διαδικασία της προσαρμογής υπαρχόντων συστημάτων, καθώς και την ανάπτυξη νέων συστημάτων.

Αναλυτικότερα, στην παρούσα εργασία χρησιμοποιήθηκαν οι προδιαγραφές του OTA για την υλοποίηση ebXML υπηρεσιών διαδικτύου σε ένα υπάρχον τουριστικό πληροφοριακό σύστημα. Η βασική λειτουργικότητα που υποστηρίζεται από το υπάρχον σύστημα και υλοποιήθηκε σαν υπηρεσία διαδικτύου (Web Service) είναι η αναζήτηση ξενοδοχείων. Επίσης χρησιμοποιήθηκαν οι προδιαγραφές του OTA για τον σχεδιασμό και την ανάπτυξη σχεσιακών βάσεων δεδομένων που να υποστηρίζουν τις λειτουργίες αναζήτησης διαθεσιμότητας και κράτησης δωματίων ξενοδοχείων. Επιπλέον αναπτύχθηκε σχεσιακή βάση δεδομένων για το προφίλ πελάτη που περιγράφεται στις προδιαγραφές του OTA, και χρησιμοποιήθηκε στην υπηρεσία αναζήτησης διαθεσιμότητας για την εξατομίκευση των αποτελεσμάτων.

Το σύστημα παροχής υπηρεσιών διαδικτύου που υλοποιήθηκε, ονομάζεται ebXML OTA Server. Η υποδομή του στηρίχτηκε στις προδιαγραφές της ebXML, σύμφωνα με τον τρόπο που προτείνεται από τις προδιαγραφές του OTA. Συγκεκριμένα χρησιμοποιήθηκε η υπηρεσία μηνυμάτων της ebXML (ebMS) για την ανταλλαγή των XML μηνυμάτων μεταξύ των συνεργατών. Η διαπραγμάτευση της συνεργασίας πραγματοποιείται με μια εφαρμογή κατασκευής CPA που αναπτύχθηκε στα πλαίσια της εργασίας. Το CPA περιγράφει τους κανόνες που διέπουν μια συγκεκριμένη συνεργασία.

Επίσης σχεδιάστηκε και αναπτύχθηκε μια εφαρμογή που αυτοματοποιεί την διαδικασία της κατασκευής σχεσιακών βάσεων δεδομένων από ένα XML Schema, και του απαραίτητου συνοδευτικού λογισμικού διαχείρισης της βάσης, παρέχοντας λειτουργίες όπως εισαγωγή, ανάκτηση και διαγραφή στοιχείων από την βάση. Ο ebXML OTA Server υλοποιεί ένα μεγάλο μέρος των προδιαγραφών του Open Travel Alliance για παροχή υπηρεσιών στην περιοχή του τουρισμού και χρησιμοποιεί τις οντολογίες που προτείνει ο OTA.

Τέλος αναπτύχθηκε μια πρότυπη διαδικτυακή εφαρμογή για την χρήση όλων των παρεχόμενων υπηρεσιών. Παρέχει ένα σύνολο από σελίδες για την αναζήτηση ξενοδοχείων και διαθεσιμότητας και την εμφάνιση των αποτελεσμάτων, για την πραγματοποίηση κρατήσεων και την διαχείριση τους, καθώς και για την διαχείριση των προφίλ πελατών. Η εφαρμογή συνδέεται με τον ebXML OTA Server μέσω της ebMS χρησιμοποιώντας ένα CPA.

Ο πλήρης σχεδιασμός των βάσεων δεδομένων του συστήματος δίνεται στα παραρτήματα Β και Γ και το λογισμικό υλοποίησης μαζί με την εφαρμογή επίδειξης είναι διαθέσιμα μέσω του Πολυτεχνείου Κρήτης.

Στην παρούσα εργασία δεν έγινε πλήρης υλοποίηση των προδιαγραφών του OTA. Συγκεκριμένα υλοποιήθηκε ένα τμήμα των προδιαγραφών που αφορούν στα ξενοδοχεία και στο προφίλ πελάτη. Το σύνολο των προδιαγραφών του OTA είναι πάρα πολύ μεγάλο για να μπορέσει να υλοποιηθεί στα πλαίσια μιας διπλωματικής εργασίας. Εντούτοις, ο σχεδιασμός και η υλοποίηση του ebXML OTA Server και των υπόλοιπων επιμέρους τμημάτων της εργασίας οδήγησε σε χρήσιμα συμπεράσματα. Το σύνολο της προσπάθειας και τα συμπεράσματα που προέκυψαν αποτελούν την συνεισφορά της παρούσας εργασίας και συνοψίζονται στα παρακάτω:

- Η ενδελεχής μελέτη των προδιαγραφών της ebXML και του OTA, καταγράφοντας τους τρόπους που συνδυάζονται για την παροχή τουριστικών υπηρεσιών, καθώς και τα πλεονεκτήματα και τα μειονεκτήματα τους κατά την υλοποίησή τους.
- Η ανάπτυξη καινοτόμων υπηρεσιών όπως η εξατομίκευση των αποτελεσμάτων αναζήτησης διαθεσιμότητας μέσω των προτιμήσεων που περιέχονται στο προφίλ πελάτη, καθώς και η παροχή εναλλακτικών

αποτελεσμάτων στην περίπτωση που η αρχική αναζήτηση δεν επιστρέφει αποτελέσματα.

- Η μεθοδολογία για την αυτόματη κατασκευή σχεσιακών βάσεων δεδομένων από XML schemata και την ανάπτυξη λογισμικού για τη διαχείριση των βάσεων δεδομένων του συστήματος.
- Η γνώση και εμπειρία που αποκτήθηκε και η αξιολόγηση τόσο των προδιαγραφών όσο και των μεθόδων και των εργαλείων που χρησιμοποιήθηκαν.
- Η εφαρμοσιμότητα των τεχνικών που χρησιμοποιήθηκαν και των μεθόδων που αναπτύχθηκαν σε παρόμοια συστήματα μεγαλύτερης κλίμακας.
- Η αξιολόγηση της μεγάλης προσπάθειας που απαιτείται για την παροχή υπαρχόντων υπηρεσιών τουρισμού με πρότυπα υπηρεσιών ebXML και OTA
- Η αξιολόγηση τεχνικών γνώσεων και τεχνολογιών που απαιτούνται

ΕΥΧΑΡΙΣΤΗΡΙΑ

Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στον καθηγητή κ. Σταύρο Χριστοδουλάκη για την επίβλεψη και την καθοδήγησή του στην ολοκλήρωση της διπλωματικής εργασίας και για τις πολύτιμες εμπειρίες που μου προσέφερε στα πλαίσια της εργασίας μου στο Εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών του Πολυτεχνείου Κρήτης.

Θα ήθελα, επίσης, να ευχαριστήσω τους καθηγητές κκ. Κουμπαράκη Μανώλη και Πετράκη Ευρυπίδη για τον χρόνο που διέθεσαν για την ανάγνωση του κειμένου και τις εποικοδομητικές παρατηρήσεις τους.

Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον κ. Μουμουτζή Νεκτάριο για την επίβλεψη, την συμπαράσταση και την συμβολή του σε όλα τα στάδια της ανάπτυξης της διπλωματικής αυτής εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω ολόκληρη την ομάδα του Εργαστηρίου Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών για την συμπαράσταση και τις συμβουλές τους κατά την εκπόνηση της διπλωματικής αυτής εργασίας.

Αλεξόπουλος Δημήτρης
Πολυτεχνείο Κρήτης
Ιούλιος, 2005

Αφιέρωση

Στην οικογένεια μου ...

Περιεχόμενα

Κεφάλαιο 1	1
Εισαγωγή	1
1.1 Αναγκαιότητα	1
1.2 Ο Οργανισμός Open Travel Alliance (OTA)	4
1.3 electronic business XML (ebXML)	6
1.4 Στόχοι της διπλωματικής εργασίας	8
1.5 Δομή της εργασίας	12
Κεφάλαιο 2	15
Επισκόπηση Σχετικής Έρευνας	15
2.1 Εισαγωγή	15
2.2 Ορισμός των Web Services	15
2.3 Υιοθέτηση των Web Services από την Τουριστική Βιομηχανία	17
2.4 Ερευνητικά Προγράμματα	18
2.5 Λειτουργικότητα Τουριστικών Πληροφοριακών Συστημάτων	21
2.6 Ανακεφαλαίωση	23
Κεφάλαιο 3	25
Περιγραφή των προδιαγραφών της ebXML και του OTA	25
3.1 Εισαγωγή στην ebXML	25
3.2 Αρχιτεκτονική της ebXML	26
3.3 Επισκόπηση των στοιχείων που απαρτίζουν την ebXML	28
3.3 Περιγραφή της ebMS	29
3.3.1 Ανατομία ενός ebXML μηνύματος	31
3.4 Περιγραφή των Collaboration Protocol Profile και Agreement (CPP/A)	32
3.5 Εισαγωγή στο OTA	33
3.6 Επισκόπηση της Αρχιτεκτονικής του OTA	34
3.6.1 Μοντέλο Αναφοράς	35
3.7 Δομή και συσκευασία μηνυμάτων	36
3.8 Γενικά μηνύματα και το πρότυπο Υπηρεσία / Δράση	37
3.9 Οι Υπηρεσίες που παρέχει ο OTA	38
3.10 Περιγραφή των προδιαγραφών του OTA για το προφίλ πελάτη	41
3.11 Περιεχόμενο του προφίλ	41
3.12 Η ροή των μηνυμάτων και των λειτουργιών του προφίλ	43
3.12.1 Μηνύματα σχετικά με το προφίλ	43
3.12.2 Ιδιότητες διαφύλαξης προσωπικών δεδομένων	44
3.12.3 Πληροφορίες για τον πελάτη	45
3.12.3.1 Απαιτούμενα δεδομένα πελάτη	45
3.12.4 Προτιμήσεις του πελάτη	45
3.12.4.1 Προτιμήσεις Ξενοδοχείων	47
3.13 Ανακεφαλαίωση	49
Κεφάλαιο 4	51
Επισκόπηση της αρχιτεκτονικής του ebXML OTA Server	51
4.1 Εισαγωγή	51
4.2 Περιγραφή του ebXML OTA Server	51
4.2.1 Η επικοινωνία του συστήματος με το διαδίκτυο (Application Interface)	52
4.2.2 Οι Διαχειριστές Υπηρεσιών (Application Logic)	53
4.2.3 Η σύνδεση με τις Βάσεις Δεδομένων	55
4.3 Οι Βάσεις Δεδομένων του συστήματος	55
4.3.1 Η Βάση Δεδομένων για τις κρατήσεις δωματίων	56
4.3.2 Η Βάση Δεδομένων για το Προφίλ Χρήστη	63
4.3.3 Η Βάση Δεδομένων για τα OTA Sessions	63
4.4 Υλοποίηση της Υπηρεσίας διαχείρισης των OTA Sessions	65
4.4.1 Περιγραφή της διαδικασίας δημιουργίας OTA Session	65
4.4.2 Περιγραφή της διαδικασίας τερματισμού ενός OTA Session	68
4.5 Υλοποίηση της Υπηρεσίας Αναζήτησης Ξενοδοχείων	68
4.5.1 Περιγραφή του XML Schema του OTA για την αναζήτηση ξενοδοχείων	69
4.5.2 Τα κριτήρια αναζήτησης ξενοδοχείων που παρέχει το υπάρχον σύστημα	70
4.5.3 Τα κριτήρια αναζήτησης ξενοδοχείων που παρέχουν οι προδιαγραφές του OTA (OTA_HotelSearchRQ/RS)	71

4.5.4 Σύγκριση των δύο μοντέλων	72
4.5.5 Διαδικασία Αναζήτησης ξενοδοχείων	73
4.5.6 Περιγραφή του XML Schema του OTA για τα αποτελέσματα της αναζήτησης ξενοδοχείων	76
4.5.7 Παραδείγματα XML εγγράφων	78
4.6 Υλοποίηση της Υπηρεσίας Αναζήτησης Διαθεσιμότητας	80
4.6.1 Περιγραφή των XML Σχημάτων του OTA για την αναζήτηση διαθεσιμότητας (OTA_HotelAvailRQ / OTA_HotelAvailRS)	80
4.6.2 Περιγραφή του στοιχείου AvailRequestSegment	81
4.6.3 Η διαδικασία της αναζήτησης διαθεσιμότητας	85
4.6.4 Η διαδικασία της αναζήτησης εναλλακτικής διαθεσιμότητας	89
4.6.5 Η χρήση των προτιμήσεων του προφίλ πελάτη στην αναζήτηση διαθεσιμότητας και εναλλακτικής διαθεσιμότητας	91
4.6.5.1 Η μέθοδος ExactMatch	91
4.6.5.2 Η μέθοδος WeightsOnly	92
4.6.6 Παραδείγματα XML εγγράφων	92
4.7 Υλοποίηση της Διαδικασίας Κράτησης Ξενοδοχείων	94
4.7.1 Περιγραφή των XML σχημάτων του OTA για την διαχείριση κρατήσεων ξενοδοχείων (OTA_HotelResRQ/OTA_HotelResRS)	95
4.7.2 Επεξεργασία κρατήσεων	97
4.7.2.1 Περιγραφή της διαδικασίας δημιουργίας καινούργιας κράτησης	98
4.7.2.2 Περιγραφή της διαδικασίας τροποποίησης υπάρχουσας κράτησης	99
4.7.2.3 Περιγραφή της διαδικασίας ακύρωσης υπάρχουσας κράτησης	101
4.7.2.4 Περιγραφή της διαδικασίας ανάκτησης υπάρχουσας κράτησης	103
4.8 Υλοποίηση της υπηρεσίας διαχείρισης του προφίλ	105
4.8.1 Δημιουργία νέου προφίλ	106
4.8.2 Ανάκτηση προφίλ από την βάση δεδομένων του ebXML OTA Server	107
4.8.3 Διαγραφή προφίλ από την βάση δεδομένων του ebXML OTA Server	108
4.8.4 Τροποποίηση προφίλ	109
4.9 Ανακεφαλαίωση	111
Κεφάλαιο 5	113
Σχεδιασμός και Υλοποίηση του CPA Editor	113
5.1 Εισαγωγή	113
5.2 Περιγραφή των προδιαγραφών της ebXML για το CPA	114
5.2.1 Περιγραφή του στοιχείου PartyInfo	116
5.2.2 Περιγραφή του στοιχείου CollaborationRole	119
5.3 Περιγραφή της λειτουργίας και του γραφικού περιβάλλοντος του CPA Editor	121
5.4 Ανακεφαλαίωση	123
Κεφάλαιο 6	125
Περιγραφή του Demo Client	125
6.1 Εισαγωγή	125
6.2 Επισκόπηση της εφαρμογής	126
6.3 Ο Τρόπος λειτουργίας της εφαρμογής και η χρήση της ebMS	139
6.4 Η Βάση Δεδομένων του Demo Client	140
6.5 Ανακεφαλαίωση	141
Κεφάλαιο 7	143
Περιγραφή της εφαρμογής XML → DB+Middleware	143
7.1 Εισαγωγή	143
7.2 Επισκόπηση της διαδικασίας που απαιτείται πριν τη χρήση της εφαρμογής	144
7.3 Επισκόπηση της λειτουργίας της εφαρμογής	144
7.4 Περιγραφή της κλάσης ObjectCopier	154
7.5 Ανακεφαλαίωση	155
Κεφάλαιο 8	157
Ανακεφαλαίωση και Μελλοντικές Επεκτάσεις	157
8.1 Ανακεφαλαίωση	157
8.2 Συμπεράσματα	160
8.4 Μελλοντικές Επεκτάσεις	163
8.3.1 Επεκτάσεις των αλγορίθμων αναζήτησης και εξατομίκευσης	164
8.3.2 Πληρέστερη υλοποίηση των OTA υπηρεσιών	164
8.3.3 Υποστήριξη διαλειτουργικότητας μέσω οντολογιών	164

Βιβλιογραφία.....	167
Παράρτημα Α.....	169
Παράρτημα Β.....	183
Παράρτημα Γ.....	232

Πίνακας Σχημάτων

Εικόνα 1 Γενική επισκόπηση της χρήσης των επιμέρους τμημάτων που υλοποιήθηκαν	12
Εικόνα 2 Αρχιτεκτονική της ebXML	26
Εικόνα 3 Παράδειγμα χρήσης της ebXML.....	27
Εικόνα 4 Τα τμήματα του ebXML Message Handler.....	30
Εικόνα 5 Γενική δομή ενός ebXML μηνύματος.....	32
Εικόνα 6 Μοντέλα Αναφοράς της αρχιτεκτονικής του OTA	35
Εικόνα 7 Μοντέλα Αναφοράς της αρχιτεκτονικής του OTA	36
Εικόνα 8 Η γενική δομή ενός συμβατού με ebXML μηνύματος του OTA	37
Εικόνα 9 Το XML Schema του OTA για το προφίλ χρήστη.....	43
Εικόνα 10 Το element του XML Schema του προφίλ χρήστη που αναφέρεται στις προτιμήσεις του χρήστη	46
Εικόνα 11 Η αρχιτεκτονική του ebXML OTA Server	52
Εικόνα 12 Το XML Schema του OTA για το RoomType.....	57
Εικόνα 13 Το XML Schema του OTA για το RatePlan	58
Εικόνα 14 Το XML Schema του OTA για το RoomRate.....	59
Εικόνα 15 Οι βασικοί πίνακες της βάσης δεδομένων για τις κρατήσεις ξενοδοχείων	59
Εικόνα 16 Οι πίνακες της βάσης δεδομένων για τα OTA Sessions.....	64
Εικόνα 17 Το XML Schema του OTA που παρέχει πληροφορίες για την δημιουργία ενός OTA Session	65
Εικόνα 18 Το XML Schema που περιγράφει την απόκριση στην αίτηση για δημιουργία ενός OTA Session.....	67
Εικόνα 19 Το XML Schema του στοιχείου ServicesSupported	67
Εικόνα 20 Το XML Schema του OTA για την αναζήτηση ξενοδοχείων	69
Εικόνα 21 Οι Πίνακες Κριτηρίων Αναζήτησης ξενοδοχείων.....	73
Εικόνα 22 Οι πίνακες «HotelSearchResultsForAlternateAvailability» και «MessageInfo»	75
Εικόνα 23 Το διάγραμμα δραστηριοτήτων της διαδικασίας αναζήτησης ξενοδοχείων	76
Εικόνα 24 Το XML Schema του OTA για την απόκριση στην αίτηση αναζήτησης ξενοδοχείων.....	77
Εικόνα 25 Το XML Schema του OTA για την αναζήτηση διαθεσιμότητας	81
Εικόνα 26 Το XML Schema του στοιχείου RatePlanCandidates	82
Εικόνα 27 Το XML Schema του στοιχείου Profiles.....	82
Εικόνα 28 Το XML Schema του στοιχείου RoomStayCandidates.....	83
Εικόνα 29 Το XML Schema του OTA για τα μηνύματα που επιστρέφουν τα αποτελέσματα της αναζήτησης διαθεσιμότητας.....	84
Εικόνα 30 Το γενικό διάγραμμα δραστηριοτήτων της διαδικασίας αναζήτησης διαθεσιμότητας	85
Εικόνα 31 Οι πίνακες της βάσης δεδομένων όπου κρατούνται τα κριτήρια της αναζήτησης διαθεσιμότητας	86
Εικόνα 32 Το Διάγραμμα δραστηριοτήτων της διαδικασίας αναζήτησης διαθεσιμότητας.....	88
Εικόνα 33 Το XML Schema που σχεδιάστηκε για την υποστήριξη της χρήσης του προφίλ χρήστη και προστέθηκε στο βασικό XML Schema του OTA.....	90
Εικόνα 34 Το XML Schema του OTA για την αίτηση κράτησης δωματίων σε ξενοδοχεία	95
Εικόνα 35 Το XML Schema του OTA για την απόκριση στην αίτηση κράτησης ξενοδοχείων	96
Εικόνα 36 Το διάγραμμα δραστηριοτήτων για την διαδικασία κράτησης δωματίων.....	99
Εικόνα 37 Το διάγραμμα δραστηριοτήτων της διαδικασίας τροποποίησης υπάρχουσας κράτησης	101
Εικόνα 38 Το XML Schema του OTA για την αίτηση ακύρωσης κράτησης.....	102
Εικόνα 39 Το διάγραμμα δραστηριοτήτων της διαδικασίας ακύρωσης μιας υπάρχουσας κράτησης ..	103
Εικόνα 40 Το XML Schema του OTA για το γενικό μήνυμα ανάγνωσης	104
Εικόνα 41 Το XML Schema του OTA που περιγράφει τις κρατήσεις ξενοδοχείων	104
Εικόνα 42 Το XML Schema του OTA για τα μηνύματα δημιουργίας νέου προφίλ.....	106
Εικόνα 43 Το XML Schema του OTA για την απόκριση στην αίτηση για δημιουργία νέου προφίλ ..	107
Εικόνα 44 Το XML Schema του OTA για την αίτηση ανάγνωσης.....	107
Εικόνα 45 Το XML Schema του OTA για το ProfileInfo	108
Εικόνα 46 Το XML Schema του OTA για την αίτηση διαγραφής	108
Εικόνα 47 Το XML Schema του OTA για την απόκριση στην αίτηση διαγραφής	109
Εικόνα 48 Το XML Schema του OTA για την αίτηση τροποποίησης	110
Εικόνα 49 Το XML Schema του OTA για την απόκριση στην αίτηση τροποποίησης	110

Εικόνα 50 Το γραφικό περιβάλλον της εφαρμογής OTA CPA Editor	122
Εικόνα 51 Η βασική σελίδα της εφαρμογής Demo Client όπου γίνεται η επιλογή του CPA.....	127
Εικόνα 52 Το βασικό μενού της εφαρμογής Demo Client όπου ο χρήστης επιλέγει τη λειτουργία που επιθυμεί να εκτελέσει	127
Εικόνα 53 Η σελίδα εισαγωγής των κριτηρίων αναζήτησης ξενοδοχείων	128
Εικόνα 54 Η σελίδα που εμφανίζει τα αποτελέσματα της αναζήτησης ξενοδοχείων.....	130
Εικόνα 55 Η σελίδα εισαγωγής των κριτηρίων αναζήτησης διαθεσιμότητας	131
Εικόνα 56 Η σελίδα που εμφανίζει τα αποτελέσματα της αναζήτησης διαθεσιμότητας.....	133
Εικόνα 57 Η σελίδα για την προετοιμασία και εκτέλεση κράτησης ξενοδοχείων.....	134
Εικόνα 58 Η σελίδα όπου γίνεται η επιλογή ενός υπάρχοντος προφίλ.....	134
Εικόνα 59 Η σελίδα όπου δημιουργείται ένα νέο προφίλ που θα συσχετιστεί με την νέα κράτηση	135
Εικόνα 60 Η σελίδα όπου γίνεται η διαχείριση μίας κράτησης ξενοδοχείων.....	136
Εικόνα 61 Η σελίδα για την δημιουργία του προφίλ χρήστη	137
Εικόνα 62 Η σελίδα για την ενημέρωση του προφίλ χρήστη	138
Εικόνα 63 Οι πίνακες της βάσης δεδομένων του Demo Client.....	140
Εικόνα 64 Η αρχιτεκτονική της εφαρμογής XML → DB+Middleware.....	147
Εικόνα 65 Επισκόπηση της χρήσης της εφαρμογής για την διαχείριση XML εγγράφων στην βάση δεδομένων.	148
Εικόνα 66 Παραδείγματα του κώδικα SQL για την κατασκευή των πινάκων της βάσης δεδομένων. .	151
Εικόνα 67 Παραδείγματα του κώδικα SQL για την κατασκευή των διαδικασιών εισαγωγής, ανάκτησης και διαγραφής από τη βάση δεδομένων (stored procedures).....	152
Εικόνα 68 Παράδειγμα του κώδικα Java για την κατασκευή των διαδικασιών εισαγωγής, ανάκτησης και διαγραφής από τη βάση δεδομένων (Java API).	153
Εικόνα 69 Στιγμιότυπο του γραφικού περιβάλλοντος για το προφίλ χρήστη που κατασκευάστηκε αυτόματα από την εφαρμογή XML → DB+Middleware	154

Πίνακας Συντομεύσεων

CRS	: Computer Reservation System
DMS	: Destination Management System
ebBPSS	: electronic business Business Process Specification Schema
ebCPA	: electronic business Collaboration Protocol Agreement
ebCPP	: electronic business Collaboration Protocol Profile
ebMS	: electronic business Message Service
ebXML	: electronic business XML
EDI	: Electronic Data Interchange
GDS	: Global Distribution System
OTA	: Open Travel Alliance
SOAP	: Simple Object Access Protocol
UDDI	: Universal Description, Discovery and Integration
WSDL	: Web Services Description Language
XML	: eXtensible Markup Language

Κεφάλαιο 1

Εισαγωγή

1.1 Αναγκαιότητα

Η ραγδαία εξέλιξη της τεχνολογίας στον χώρο των υπολογιστών και των τηλεπικοινωνιών και η παράλληλη μείωση του κόστους πρόσβασης σε αυτά τα νέα μέσα, έδωσαν μια νέα διάσταση στον τρόπο λειτουργίας των επιχειρήσεων. Η τουριστική βιομηχανία είναι από τους πρωτοπόρους στην αξιοποίηση των νέων τεχνολογιών. Από τη δεκαετία του 60 μέχρι και σήμερα αποτελεί κλασσικό πεδίο εφαρμογής της πληροφορικής και των τηλεπικοινωνιών, όπως στα ηλεκτρονικά συστήματα κρατήσεων (Computerized Reservation Systems-CRS), που παρέχουν λειτουργικότητα για κρατήσεις και αγορά αεροπορικών εισιτηρίων, ενοικιάσεις αυτοκινήτων, διαμονή σε ξενοδοχεία και άλλα τουριστικά προϊόντα συμπεριλαμβανομένης και πρόσβασης σε τιμές, δρομολόγια και διαθεσιμότητα των σχετικών προϊόντων, καθώς και στα παγκόσμια συστήματα διανομής¹(Global Distribution Systems - GDS), που παρέχουν τις παραπάνω δυνατότητες για πολλές διαφορετικές τουριστικές επιχειρήσεις και σε πολλές χώρες.

Η εκτεταμένη εφαρμογή των νέων τεχνολογιών στην τουριστική βιομηχανία, λόγω της απουσίας κοινώς αποδεκτών προτύπων, είχε σαν αποτέλεσμα να υπάρχει σήμερα πληθώρα διαφορετικών συστημάτων, με διαφορετικές δυνατότητες, τεχνολογίες και αρχιτεκτονικές. Υπάρχει λοιπόν πρόβλημα διαλειτουργικότητας, το οποίο αποτελεί σημαντικό εμπόδιο στην ανάπτυξη του ηλεκτρονικού εμπορίου, που είναι ιδιαίτερα σημαντικό για τις μικρές επιχειρήσεις.

Μέχρι και σήμερα, η ανταλλαγή επιχειρησιακών δεδομένων πραγματοποιείται μέσω του πρωτοκόλλου Ηλεκτρονικής Ανταλλαγής Δεδομένων (Electronic Data

¹ Τα τέσσερα μεγαλύτερα GDSs σήμερα είναι το Sabre [27], το Galileo [14], το Amadeus [1] και το Worldspan [37]. Όλες οι αεροπορικές εταιρίες, πολλές αλυσίδες ξενοδοχείων και εταιρίες ενοικίασης αυτοκινήτων κοινοποιούν τα προϊόντα τους μέσω GDSs. Υπάρχουν επίσης εξειδικευμένα GDSs, όπως για παράδειγμα η Pegasus Solutions [25] και η Utell [34], οι οποίες βασικά πωλούν δωμάτια ξενοδοχείων.

Interchange - EDI) [7]. Το EDI όμως έχει το μειονέκτημα ότι είναι κλειστό πρότυπο και έχει πολύ υψηλό κόστος στην εγκατάσταση και συντήρηση του. Καθίσταται έτσι πολύ δύσκολη έως απαγορευτική η χρήση του από μικρομεσαίες επιχειρήσεις, αναγκάζοντας τις να εξαρτώνται από εξειδικευμένες επιχειρήσεις, όπως τα GDSs και τα switches² στην τουριστική βιομηχανία.

Μια σειρά από εξελίξεις στην τεχνολογία και το διαδίκτυο υπόσχονται να ανατρέψουν την υπάρχουσα κατάσταση. Η σημαντικότερη ίσως εξέλιξη είναι η εμφάνιση της XML [39]. Η XML είναι το ακρωνύμιο των λέξεων eXtensible Markup Language, και είναι απόγονος της SGML [29]. Η XML είναι από τη φύση της επεκτάσιμη, επιτρέπει την επικοινωνία μεταξύ διαφορετικών εφαρμογών ανεξάρτητα από την πλατφόρμα υλικού και λογισμικού που χρησιμοποιείται, και αποτελεί ανοιχτό πρότυπο. Τα παραπάνω οδήγησαν στην ευρεία αποδοχή της από τη βιομηχανία λογισμικού, και στην διαρκώς αυξανόμενη χρήση της στην ανταλλαγή δεδομένων μέσω του διαδικτύου.

Μια άλλη ανερχόμενη και πολλά υποσχόμενη τεχνολογία, που βασίζεται στην XML, είναι οι Web Services [35]. Πρόκειται για μια τεχνολογία που επιτρέπει την πρόσβαση τμημάτων λογισμικού μέσω του διαδικτύου, δίνοντας τη δυνατότητα σε διαφορετικές εφαρμογές να επικοινωνούν μεταξύ τους και να ανταλλάσσουν δεδομένα. Στηρίζεται σε ανοιχτά πρότυπα και δεν δεσμεύεται από συγκεκριμένο υλικό ή λογισμικό.

Η ευρεία διάδοση της XML και των Web Services οδήγησε στη δημιουργία οργανισμών για τη σχεδίαση προδιαγραφών διαλειτουργικότητας για πολλούς τομείς της βιομηχανίας, όπως είναι ο οργανισμός Open Travel Alliance [22] για τον τουρισμό. Περαιτέρω, αναπτύχθηκαν προδιαγραφές για την ενίσχυση και καθιέρωση του ηλεκτρονικού επιχειρείν γενικότερα, όπως είναι η electronic business XML (ebXML) [8].

² Τα switches είναι επιχειρήσεις που παρέχουν σύνδεση με περισσότερα από ένα GDS. Έτσι οι τουριστικές επιχειρήσεις μπορούν να χρησιμοποιούν μία μόνο σύνδεση – με το switch - αντί για πολλές συνδέσεις με τα διάφορα GDSs.

Η ebXML είναι ένα προτεινόμενο διεθνές πρότυπο στην κατηγορία των πρωτοκόλλων για παροχή υπηρεσιών στο διαδίκτυο. Ανήκει στην κατηγορία των πρωτοκόλλων που αποσκοπούν στο να περιγράψουν την σειρά της ανταλλαγής μηνυμάτων που απαιτούνται για την ολοκλήρωση μιας συνεργασίας ή μιας υπηρεσίας, που συχνά αναφέρονται σαν πρωτόκολλα συγχρονισμού (coordination protocols). Η ebXML είναι ένα κάθετο πρωτόκολλο (vertical protocol) γιατί απευθύνεται σε με μια συγκεκριμένη περιοχή εφαρμογών (ηλεκτρονικό εμπόριο). Παρόλο που η ebXML είναι κάθετο πρωτόκολλο δεν είναι αρκετά εξειδικευμένη για να περιγράψει τη σημαντική των όρων που χρησιμοποιούνται σε μια συγκεκριμένη περιοχή του ηλεκτρονικού εμπορίου, όπως για παράδειγμα το ηλεκτρονικό εμπόριο στην περιοχή του τουρισμού. Η διαλειτουργικότητα των υπηρεσιών στην περιοχή του τουρισμού θα αυξανόταν σημαντικά εάν τα μηνύματα που ανταλλάσσονται με τη χρήση της ebXML χρησιμοποιούσαν όρους με αυστηρά καθορισμένη σημαντική για την περιοχή του τουρισμού. Αυτό υποδεικνύει την ενσωμάτωση οντολογιών πεδίου (domain ontologies) με τα πρότυπα υπηρεσιών του διαδικτύου, ώστε τα μηνύματα που ανταλλάσσονται να αποδίδουν συντακτική και σημαντική γνώση στον παραλήπτη.

Ο Open Travel Alliance (OTA) είναι ένας αυτοχρηματοδοτούμενος μη κερδοσκοπικός οργανισμός που αναπτύσσει προδιαγραφές διαλειτουργικότητας για τον τουριστικό τομέα. Χρησιμοποιεί την XML Schema σαν γλώσσα περιγραφής των XML εγγράφων που μεταδίδονται μεταξύ διαφορετικών τουριστικών πληροφοριακών συστημάτων. Αποτελείται από τις μεγαλύτερες επιχειρήσεις και οργανισμούς στον χώρο της τουριστικής βιομηχανίας, συσσωρεύοντας έτσι σημαντικό όγκο γνώσης των αναγκών και απαιτήσεων της συγκεκριμένης αγοράς (αποτελώντας ουσιαστικά μια οντολογία για τον τουρισμό), και θέτει τις βάσεις ώστε να καθιερωθεί ως ένα καθολικά αποδεκτό πρότυπο διαλειτουργικότητας στην περιοχή του τουρισμού.

Είναι ιδιαίτερα κρίσιμο για τα υπάρχοντα τουριστικά πληροφοριακά συστήματα να προσαρμοστούν στα σύγχρονα πρότυπα, ακολουθώντας προδιαγραφές διαλειτουργικότητας όπως αυτές της ebXML και του OTA. Η προσαρμογή αυτή μπορεί να αποτελέσει μια αρκετά δύσκολη διαδικασία. Είναι σημαντικό να αναζητηθούν μέθοδοι και τεχνικές με τις οποίες να τεθούν οι βάσεις για πιο ομαλή μετάβαση των υπάρχοντων συστημάτων από την σημερινή τους μορφή στο σύγχρονο

διαλειτουργικό περιβάλλον που διαμορφώνεται. Αυτός είναι και ο στόχος της παρούσας διπλωματικής εργασίας, και συνοψίζεται στις παρακάτω τρεις γενικές κατευθύνσεις:

- τη μελέτη και υλοποίηση της προσαρμογής υπαρχόντων τουριστικών πληροφοριακών συστημάτων στις προδιαγραφές της ebXML και του OTA,
- την ανάδειξη των σημαντικών χαρακτηριστικών των προδιαγραφών για την υλοποίηση σύνθετης λειτουργικότητας και
- την ανάπτυξη μεθόδων και τεχνικών που θα διευκολύνει και θα επιταχύνει την διαδικασία της προσαρμογής.

Μια αναλυτικότερη εικόνα των στόχων της εργασίας δίνεται στην ενότητα 1.4 του παρόντος κεφαλαίου, όπου παρατίθενται και τα επιμέρους τμήματα που υλοποιήθηκαν στα πλαίσια της εργασίας.

Στη συνέχεια δίνεται μια περιγραφή των προδιαγραφών του OTA καθώς και της ebXML, πάνω στις οποίες στηρίζεται η εργασία.

1.2 Ο Οργανισμός Open Travel Alliance (OTA)

Ο OTA (Open Travel Alliance) είναι ένας αυτοχρηματοδοτούμενος μη κερδοσκοπικός οργανισμός που σχεδιάζει και αναπτύσσει ανοιχτές προδιαγραφές διαλειτουργικότητας, με στόχο την εδραίωση του ηλεκτρονικού εμπορίου και την ενθάρρυνση της δημιουργίας νέων υπηρεσιών και προϊόντων σε όλο το φάσμα του τουριστικού τομέα.

Αποτελείται από τις μεγαλύτερες επιχειρήσεις και οργανισμούς στο χώρο της τουριστικής βιομηχανίας, με αποτέλεσμα τη συσσώρευση σημαντικού όγκου γνώσης και εμπειρίας, που αντανακλώνται στο μέγεθος και τη λεπτομέρεια των προδιαγραφών που αναπτύσσει. Οι προδιαγραφές αυτές, δίνονται σε μορφή **XML Schema** [40], περιλαμβάνουν τη μορφή και το περιεχόμενο της πληροφορίας που μπορεί να μεταδίδεται μεταξύ των τουριστικών επιχειρήσεων, και αφορούν σε όλους τους κλάδους της τουριστικής βιομηχανίας – αεροπορικές εταιρίες, ξενοδοχεία, εταιρίες ενοικίασης αυτοκινήτων κ.α.

Ο OTA προσδιορίζει την αρχιτεκτονική με την οποία επιτυγχάνεται η διαλειτουργικότητα μεταξύ διαφορετικών πληροφοριακών συστημάτων για τον τουρισμό, η οποία στηρίζεται στην ανταλλαγή μηνυμάτων με την μορφή XML εγγράφων μεταξύ των συνεργαζόμενων τουριστικών επιχειρήσεων. Τα μηνύματα ακολουθούν τη λογική αίτησης - απόκρισης. Για παράδειγμα, η αναζήτηση ξενοδοχείων εκτελείται στέλνοντας ένα μήνυμα αίτησης για αναζήτηση, που περιέχει τα κριτήρια της αναζήτησης και το αποτέλεσμα της αναζήτησης επιστρέφεται με το αντίστοιχο μήνυμα απόκρισης. Αυτό το μοντέλο αίτησης - απόκρισης είναι όμοιο εννοιολογικά με το μοντέλο των Web Services³. Μπορούμε λοιπόν να αποκαλούμε την ανταλλαγή των μηνυμάτων του OTA σαν **OTA Web Services**.

Συνοπτικά, τα μηνύματα που ορίζουν οι προδιαγραφές διακρίνονται στις παρακάτω ομάδες:

- Γενικά μηνύματα: Αν και πολλά μηνύματα αναφέρονται σε συγκεκριμένο τομέα της τουριστικής βιομηχανίας όπως είναι τα ξενοδοχεία, μερικά μηνύματα είναι γενικά και μπορούν να εφαρμοστούν ευρύτερα, όπως είναι τα μηνύματα ακύρωσης μιας κράτησης και διαγραφής ενός προφίλ.
- Μηνύματα για τα αεροπορικά ταξίδια, όπως είναι η αίτηση για διαθεσιμότητα και η αίτηση για την αναζήτηση των χαμηλότερων τιμών.
- Μηνύματα για την ενοικίαση αυτοκινήτων, όπως είναι η αναζήτηση αυτοκινήτων.
- Μηνύματα για τα ξενοδοχεία, όπως είναι η αναζήτηση ξενοδοχείων και η αναζήτηση διαθέσιμων δωματίων.
- Μηνύματα για πληροφορίες σχετικά με τρένα, όπως είναι η αναζήτηση διαθεσιμότητας.
- Μηνύματα για την κράτηση χρόνου για ένα παιχνίδι γκολφ ή την αναζήτηση διαθέσιμου μαθήματος γκολφ.
- Μηνύματα σχετικά με την ασφάλεια του ταξιδιού, όπως η αναζήτηση ασφαλιστικού πλάνου.
- Μηνύματα για πακέτα διακοπών: ένα πακέτο διακοπών μπορεί να περιέχει πολλά τμήματα, όπως μεταφορά, διαμονή, ενοικίαση αυτοκινήτου, επιπλέον

³ Οι Web Services αναλύονται στο κεφάλαιο 2 όπου γίνεται επισκόπηση της σχετικής έρευνας

προϊόντα και υπηρεσίες. Παράδειγμα τέτοιου μηνύματος είναι η αναζήτηση διαθέσιμου πακέτου διακοπών.

- Μηνύματα για τη διαχείριση του δρομολογίου του ταξιδιού, όπως είναι η αίτηση για ανάγνωση του δρομολογίου.
- Μηνύματα σχετικά με την αξιοπιστία και την φερεγγυότητα του πελάτη.
- Μηνύματα για τη διαχείριση του προφίλ χρήστη, όπως είναι η αίτηση για δημιουργία προφίλ.

Όσον αφορά τον τρόπο μετάδοσης των μηνυμάτων, για την εξασφάλιση της εγγυημένης μετάδοσης, της αξιοπιστίας και της ασφάλειας, ο ΟΤΑ συστήνει τη χρήση της Υπηρεσίας Μηνυμάτων της **ebXML** (**ebMS**).

1.3 electronic business XML (ebXML)

Η **ebXML** (electronic business XML) είναι ένα αρθρωτό πλαίσιο ανοιχτών προδιαγραφών για το ηλεκτρονικό εμπόριο. Αποτελεί το αποτέλεσμα της κοινής προσπάθειας δύο διεθνών οργανισμών, των **UN/CEFACT**⁴ [32] και **OASIS**⁵ [21], με στόχο να αναπτυχθεί «μια ενιαία παγκόσμια ηλεκτρονική αγορά βασισμένη σε μια ανοικτή δημόσια υποδομή με βάση την XML, που να επιτρέπει την παγκόσμια χρήση των ηλεκτρονικών επιχειρησιακών πληροφοριών κατά τρόπο διαλειτουργικό, ασφαλή και συνεπή από όλα τα συμβαλλόμενα μέρη». Η ebXML ανήκει στην κατηγορία των πρωτοκόλλων συγχρονισμού (coordination protocol⁶) και επιπλέον αποτελεί κάθετο πρωτόκολλο καθώς απευθύνεται στο ηλεκτρονικό εμπόριο. Η ebXML προδιαγράφει τον τρόπο με τον οποίο θα πρέπει να ορίζονται, να τεκμηριώνονται και να διεξάγονται οι ανταλλαγές ηλεκτρονικού εμπορίου. Από την άποψη αυτή, ο ορισμός των υπηρεσιών διαδικτύου από την ebXML είναι περισσότερο ολοκληρωμένος και λεπτομερής από τον ορισμό που δίνουν άλλες προδιαγραφές, και δίνει μια καλύτερη

⁴ Το Κέντρο των Ηνωμένων Εθνών για την Διευκόλυνση των Συναλλαγών και του Ηλεκτρονικού Επιχειρείν. Είναι ο κατασκευαστής του EDIFACT και είναι ένας από τους τέσσερις οργανισμούς παγκοσμίως που μπορούν να θέσουν de jure πρότυπα

⁵ Ο μεγαλύτερος, ανεξάρτητος, μη – κερδοσκοπικός οργανισμός, που ασχολείται με την προτυποποίηση των εφαρμογών της XML, με περισσότερες από 200 εταιρίες μέλη

⁶ Ο όρος «coordination protocol» είναι δανεισμένος από το βιβλίο Web Services – Concepts, Architectures and Applications [1]

εικόνα του τι θα πρέπει να περιλαμβάνεται όταν χρησιμοποιούνται οι υπηρεσίες διαδικτύου. Περιγράφει δηλαδή το τι λείπει από τις άλλες προδιαγραφές (κυρίως το UDDI, το SOAP και το WSDL). Κοινή προσδοκία είναι η σταδιακή σύγκλιση των προδιαγραφών.

Συνοπτικά, οι προδιαγραφές που απαρτίζουν την ebXML είναι οι παρακάτω

- **Business Process Specification Schema (BPSS) [9] – Σχήμα Περιγραφής Επιχειρησιακών Διαδικασιών:**

Η BPSS είναι μια γλώσσα προδιαγραφών βασισμένη στην XML η οποία περιγράφει τις «δημόσιες» επιχειρησιακές διαδικασίες που υποστηρίζει μια επιχείρηση.

- **Core Components (CC)- Βασικά Συστατικά**

Είναι τα βασικά συστατικά που παρέχουν την επιχειρησιακή πληροφορία που κωδικοποιείται στα επιχειρησιακά έγγραφα τα οποία ανταλλάσσονται μεταξύ των εμπορικών συνεργατών.

- **Registry / Repository (Reg / Rep) [12] – Μητρώο / Αποθήκη**

Είναι η περιγραφή του τρόπου αποθήκευσης και καταχώρισης των επιχειρησιακών διαδικασιών, των προφίλ των επιχειρήσεων και όλων των απαραίτητων στοιχείων για την επίτευξη μιας ηλεκτρονικής επιχειρησιακής συνεργασίας.

- **Collaboration Protocol Profile / Agreement (CPP/A) [10] – Προφίλ Πρωτοκόλλου Συνεργασίας / Συμφωνία Πρωτοκόλλου Συνεργασίας**

Είναι XML έγγραφα που περιγράφουν τις ικανότητες ηλεκτρονικής συνεργασίας μιας επιχείρησης και τις συμφωνίες ηλεκτρονικής συνεργασίας δύο επιχειρήσεων αντίστοιχα.

- **Message Service (ebMS) [11] – Υπηρεσία Ανταλλαγής Μηνυμάτων**

Η Υπηρεσία Μηνυμάτων της ebXML παρέχει έναν κομψό, γενικού σκοπού μηχανισμό για την ανταλλαγή μηνυμάτων, που στηρίζεται στο SOAP⁷ με συνημμένα, και μπορεί να μεταφέρει περιεχόμενο οποιουδήποτε τύπου.

Οι προδιαγραφές αυτές μπορούν να χρησιμοποιηθούν είτε αυτόνομα, σύμφωνα με τις εκάστοτε ανάγκες αυτού που τις υιοθετεί, ή σε συνδυασμό, οπότε και αποτελούν ένα ολοκληρωμένο πλαίσιο ηλεκτρονικού επιχειρείν. Αυτή την ευελιξία της ebXML αξιοποίησε ο ΟΤΑ, υιοθετώντας την Υπηρεσία Μηνυμάτων (ebMS), ως μια ισχυρή υποδομή για την ανταλλαγή των μηνυμάτων του.

⁷ Απλό πρωτόκολλο μετάδοσης δεδομένων βασισμένο στην XML. Το όνομα του προέρχεται από τα αρχικά των λέξεων Simple Object Access Protocol [30].

1.4 Στόχοι της διπλωματικής εργασίας

Όπως προαναφέρθηκε, σήμερα υπάρχει μεγάλη πληθώρα διαφορετικών πληροφοριακών συστημάτων για τον τουρισμό. Η εμπέδωση κοινώς αποδεκτών προδιαγραφών διαλειτουργικότητας, όπως είναι οι προδιαγραφές της ebXML και του OTA, και η υιοθέτηση τους από τα υπάρχοντα και τα αναπτυσσόμενα ή μελλοντικά τουριστικά πληροφοριακά συστήματα, μπορεί να βοηθήσει σημαντικά στην επίτευξη της διαλειτουργικότητας.

Η επέκταση των υπάρχοντων συστημάτων ώστε να υποστηρίζουν τα σύγχρονα διεθνή πρότυπα αποτελεί επιτακτική ανάγκη, προκειμένου τα συστήματα αυτά να εξελιχθούν και να διατηρήσουν την ανταγωνιστικότητά τους. Ταυτόχρονα όμως αποτελεί μια δύσκολη και χρονοβόρα διαδικασία. Επιπλέον η υποστήριξη νέων λειτουργιών, ενσωματώνοντας καινούργιες έννοιες και σύνθετα μοντέλα δεδομένων, όπως είναι το ολοκληρωμένο προφίλ χρήστη που ορίζει ο OTA, εισάγει υψηλό βαθμό πολυπλοκότητας στην διαδικασία. Δημιουργείται έτσι η ανάγκη για την εμφάνιση μεθοδολογιών που θα βοηθήσουν στην υλοποίηση των παραπάνω στόχων.

Η δημιουργία μιας γενικής μεθοδολογίας για την ανάπτυξη λογισμικού υποστήριξης των προδιαγραφών διαλειτουργικότητας της ebXML και του OTA αποτελεί τον κύριο στόχο της παρούσας εργασίας. Συγκεκριμένα, στα πλαίσια της εργασίας εφαρμόστηκαν οι προδιαγραφές της ebXML και του OTA για την αναζήτηση ξενοδοχείων σε ένα πληροφοριακό σύστημα για τις τουριστικές επιχειρήσεις της Κρήτης, και επιπλέον σχεδιάστηκε και υλοποιήθηκε ένα πρότυπο σύστημα διαχείρισης κρατήσεων ξενοδοχείων και προφίλ χρηστών, καθώς και παροχής των αντίστοιχων υπηρεσιών διαδικτύου (Web Services).

Το σύστημα που αναπτύχθηκε ονομάζεται ebXML OTA Server (μιας και παρέχει Web Services βασισμένες στις προδιαγραφές της ebXML και του OTA) και περιλαμβάνει τις παρακάτω υπηρεσίες:

- Αναζήτηση ξενοδοχείων (με εφαρμογή για την περιοχή της Κρήτης) σύμφωνα με τα κριτήρια αναζήτησης ξενοδοχείων που ορίζονται στο αντίστοιχο XML Schema του OTA (OTA_HotelSearchRQ.xsd).

- Αναζήτηση διαθέσιμων δωματίων ξενοδοχείων χρησιμοποιώντας τα κριτήρια που ορίζονται στις προδιαγραφές του OTA (OTA_HotelAvailRQ.xsd) και / ή τις προτιμήσεις που ορίζονται στο προφίλ του χρήστη.
- Κράτηση ενός ή περισσότερων δωματίων σε ένα ή περισσότερα ξενοδοχεία.
- Διαχείριση υπάρχουσας κράτησης (ανάγνωση, τροποποίηση, ακύρωση)⁸.
- Δημιουργία και διαχείριση του προφίλ χρήστη.

Η πρόσβαση στις παραπάνω υπηρεσίες επιτυγχάνεται με την αποστολή και λήψη κατάλληλων XML μηνυμάτων. Για παράδειγμα, κάποιος που επιθυμεί να εξετάσει ποια ξενοδοχεία ικανοποιούν συγκεκριμένα κριτήρια της αρεσκείας του, συνθέτει ένα XML μήνυμα που περιέχει τα κριτήρια αυτά, το οποίο και αποστέλλει στο σύστημα. Το μήνυμα αναλύεται και εκτελείται η αναζήτηση ξενοδοχείων με τα κριτήρια που δόθηκαν. Στη συνέχεια συντίθεται ένα καινούργιο XML μήνυμα από το σύστημα που περιέχει τα αποτελέσματα της αναζήτησης και αποστέλλεται στο χρήστη. Τα μηνύματα αυτά υπακούουν στα αντίστοιχα XML Schemata του OTA.

Η υποδομή του συστήματος στηρίζεται στην ebXML. Συγκεκριμένα, η μετάδοση των OTA XML μηνυμάτων πραγματοποιείται μέσω μιας υλοποίησης ανοιχτού λογισμικού της υπηρεσίας μηνυμάτων της ebXML. Έτσι το XML έγγραφο αίτησης μιας υπηρεσίας που κατασκευάζει ο χρήστης, αποστέλλεται συνημμένο σε ένα ebXML μήνυμα προς το σύστημα. Το XML έγγραφο απόκρισης στην αίτηση, αποστέλλεται επίσης συνημμένο σε ένα ebXML μήνυμα προς τον χρήστη. Η επικοινωνία των μηνυμάτων είναι σύγχρονη.

Για να επιτραπεί η χρήση των υπηρεσιών του συστήματος απαιτείται η ύπαρξη ή η κατασκευή ενός καινούργιου CPA. Για την κατασκευή του CPA χρησιμοποιείται μια πρότυπη εφαρμογή κατασκευής CPA. Ο χρήστης εισάγει ορισμένες παραμέτρους, όπως το URI στο οποίο δέχεται ebXML μηνύματα η δική του ebMS υλοποίηση, τις υπηρεσίες τις οποίες επιθυμεί να χρησιμοποιήσει, επιλέγοντας από ένα σύνολο προεπιλεγμένων τιμών, και το όνομα που επιθυμεί να δώσει στο συγκεκριμένο CPA.

⁸ Στην παρούσα εργασία δεν υλοποιήθηκαν compensations, όπως η επιβολή ποινών ακύρωσης κράτησης ή η προκαταβολή χρηματικού ποσού για την κράτηση. Ωστόσο, οι σχεσιακές βάσεις δεδομένων του συστήματος σχεδιάστηκαν έτσι ώστε να παρέχουν υποστήριξη για μελλοντική υλοποίηση τέτοιων λειτουργιών, ακολουθώντας τις έννοιες που ορίζονται στις προδιαγραφές του OTA για τις κρατήσεις ξενοδοχείων.

Για την υποστήριξη των επιμέρους υπηρεσιών σχεδιάστηκαν και αναπτύχθηκαν σχεσιακές βάσεις δεδομένων, παρέχοντας έτσι τη δυνατότητα υποστήριξης σύνθετων ερωτημάτων, όπως για παράδειγμα στην περίπτωση αναζήτησης ξενοδοχείων και διαθεσιμότητας, καθώς και η δυνατότητα αποδοτικότερης και ταχύτερης διαχείρισης των πληροφοριών που σχετίζονται με τις υπηρεσίες του συστήματος. Επίσης υλοποιήθηκε και η απαιτούμενη λειτουργικότητα για την αλληλεπίδραση των σχεσιακών βάσεων δεδομένων με τα XML έγγραφα. Η μετατροπή δηλαδή των πληροφοριών που περιέχονται στα XML έγγραφα σε σχεσιακή μορφή και η αποθήκευση τους στις σχεσιακές βάσεις δεδομένων, καθώς και η εξαγωγή των πληροφοριών που βρίσκονται αποθηκευμένες στις σχεσιακές βάσεις δεδομένων σε μορφή XML εγγράφων. Η διαδικασία αυτή της αμφίδρομης μετατροπής των πληροφοριών των XML εγγράφων σε σχεσιακή μορφή περιλαμβάνει ένα ενδιάμεσο στάδιο, όπου τα XML έγγραφα αναπαρίστανται σε ιεραρχίες αντικειμένων. Η διαδικασία αυτή ονομάζεται Data Binding και χρησιμοποιείται σαν ενδιάμεσο στάδιο και στην αποθήκευση των XML εγγράφων στις σχεσιακές βάσεις δεδομένων και στην εξαγωγή των πληροφοριών των σχεσιακών βάσεων δεδομένων σε XML έγγραφα.

Η μεγάλη λεπτομέρεια των XML σχημάτων του OTA δίνει τη δυνατότητα να χρησιμοποιηθούν για την κατασκευή των σχεσιακών βάσεων δεδομένων που χρειάζεται το σύστημα, όπως είναι η βάση δεδομένων για τις κρατήσεις ξενοδοχείων και η βάση δεδομένων για το προφίλ των χρηστών. Η ιδιόχειρη κατασκευή των σχεσιακών βάσεων δεδομένων, ώστε να καλύπτουν τα XML Schemata σε όλη τους την έκταση, είναι μια εξαιρετικά χρονοβόρα διαδικασία, και ενέχει τον κίνδυνο να παρουσιαστούν σφάλματα. Στην περίπτωση επίσης που γίνουν αλλαγές στα XML Schemata, όπως και συνέβη στη συγκεκριμένη περίπτωση με τις συνεχείς ανανεώσεις στις εκδόσεις των προδιαγραφών του OTA, τότε η κατάσταση περιπλέκεται περισσότερο και απαιτείται η εύρεση μιας καλύτερης λύσης. Για την αντιμετώπιση αυτού του προβλήματος σχεδιάστηκε και υλοποιήθηκε μια εφαρμογή που επιτρέπει την αυτόματη κατασκευή σχεσιακής βάσης δεδομένων από ένα XML Schema, καθώς και την αυτόματη κατασκευή stored procedures και Java κλάσεων για τις λειτουργίες

της εισαγωγής, ανάκτησης και διαγραφής δεδομένων από τη σχεσιακή βάση δεδομένων⁹.

Για τον έλεγχο και την επίδειξη της λειτουργίας των υπηρεσιών που παρέχει το σύστημα, υλοποιήθηκε μια εφαρμογή πελάτης που χρησιμοποιεί τις παραπάνω υπηρεσίες. Η εφαρμογή αποτελείται από ένα σύνολο ιστοσελίδων (JSPs) για την εισαγωγή δεδομένων και την εμφάνιση των αποτελεσμάτων των υπηρεσιών, την σύνδεση με την υπηρεσία μηνυμάτων της ebXML (ebMS) για την επικοινωνία με το σύστημα μέσω ebXML μηνυμάτων, και από μια βάση δεδομένων όπου αποθηκεύονται πληροφορίες σχετικά με την χρήση των υπηρεσιών (τα IDs των κρατήσεων που έχουν πραγματοποιηθεί και των προφίλ που έχουν κατασκευαστεί).

Συνοπτικά, τα επιμέρους τμήματα που υλοποιήθηκαν στα πλαίσια της εργασίας είναι:

1. ebXML OTA Server

a. Υπηρεσίες που υποστηρίζονται

- Αναζήτηση ξενοδοχείων
- Έλεγχος διαθεσιμότητας
- Κράτηση δωματίων
- Διαχείριση κρατήσεων
- Διαχείριση Προφίλ
- Διαχείριση Sessions

b. ebXML Messaging

- Προσαρμογή της ebMS για τις ανάγκες ανταλλαγής μηνυμάτων του συστήματος

c. Databases

- Κατασκευή των απαιτούμενων σχεσιακών βάσεων δεδομένων με αυτόματο και μη αυτόματο τρόπο
 - Διαθεσιμότητα, Κρατήσεις ξενοδοχείων
 - Προφίλ
 - OTA Sessions

2. CPA Editor

- Πρότυπη εφαρμογή για την κατασκευή CPA, το οποίο απαιτείται για την επικοινωνία του Server με τους Clients

⁹ Η εφαρμογή στηρίζεται σε απλούς καθιερωμένους κανόνες μετατροπής XML Schema σε σχεσιακό σχήμα. Οι κανόνες αυτοί καθώς και όλη η διαδικασία που ακολουθείται για την μετατροπή αναλύονται στο κεφάλαιο 7, όπου περιγράφεται η σχετική εφαρμογή.

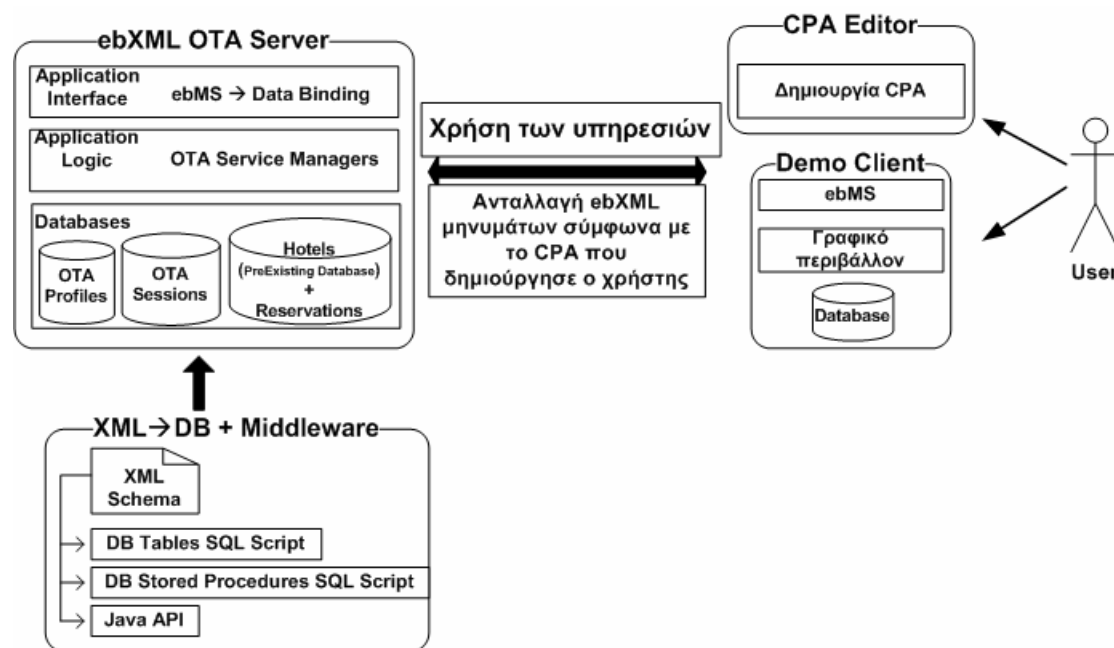
3. Demo Client

- Πρότυπη εφαρμογή για την επίδειξη και τον έλεγχο λειτουργίας του συστήματος μέσω γραφικού περιβάλλοντος.

4. XML → DB + Middleware

- Κατασκευή προγραμματιστικού εργαλείου για την αυτοματοποίηση της διαδικασίας μετατροπής XML Schema σε σχεσιακή βάση δεδομένων.
 - SQL script που περιέχει τους πίνακες της βάσης.
 - SQL script που περιέχει τα stored procedures για εισαγωγή, ανάγνωση και διαγραφή από τη βάση.
 - Java API που χρησιμοποιεί τα stored procedures.

Η χρήση των επιμέρους τμημάτων δίνεται γραφικά στο παρακάτω σχήμα (Εικόνα 1):



Εικόνα 1 Γενική επισκόπηση της χρήσης των επιμέρους τμημάτων που υλοποιήθηκαν

1.5 Δομή της εργασίας

Στο 2^ο κεφάλαιο της εργασίας γίνεται επισκόπηση της σχετικής έρευνας; Περιγράφονται οι Web Services, γίνεται κατηγοριοποίηση και σύγκριση τους, και γίνονται αναφορές σε προγράμματα που έχουν υλοποιηθεί ή βρίσκονται σε εξέλιξη και σχετίζονται με τον τουριστικό τομέα. Περιγράφεται επίσης το πληροφοριακό

σύστημα για τις τουριστικές επιχειρήσεις της Κρήτης, το οποίο χρησιμοποιήθηκε σαν βασική υποδομή για την ανάπτυξη των υπηρεσιών που υλοποιήθηκαν στην παρούσα εργασία.

Στο 3^ο κεφάλαιο περιγράφονται οι προδιαγραφές της ebXML και του OTA. Περιγράφονται ιδιαίτερα η υπηρεσία μηνυμάτων (ebMS) και το προφίλ και η συμφωνία στο πρωτόκολλο συνεργασίας (CPP/A), τα οποία και χρησιμοποιούνται στην εργασία. Γίνεται επίσης περιγραφή των προδιαγραφών του OTA, δίνοντας έμφαση στις προδιαγραφές που αναφέρονται στον τομέα των ξενοδοχείων, καθώς και στην αρχιτεκτονική που προτείνει ο OTA ως υποδομή για την υλοποίηση των προδιαγραφών του. Περιγράφεται επίσης το προφίλ χρήστη που ορίζεται στις προδιαγραφές του OTA.

Στο 4^ο κεφάλαιο περιγράφεται ο σχεδιασμός και η υλοποίηση του συστήματος παροχής OTA Web Services. Περιγράφεται η αρχιτεκτονική του συστήματος, η επεξεργασία και ο τρόπος μετάδοσης των XML εγγράφων. Περιγράφεται επίσης ο σχεδιασμός και η υλοποίηση των βάσεων δεδομένων του συστήματος. Παρουσιάζονται οι βασικές έννοιες στις οποίες στηρίζεται η βάση δεδομένων για τις κρατήσεις ξενοδοχείων καθώς και στιγμιότυπα από το σχήμα της βάσης δεδομένων. Στο ίδιο κεφάλαιο περιγράφονται οι υπηρεσίες που υλοποιήθηκαν:

- Παρουσιάζεται η διαδικασία δημιουργίας και τερματισμού των OTA Sessions.
- Περιγράφεται η αντιστοίχιση των προδιαγραφών του OTA για την αναζήτηση ξενοδοχείων με τα κριτήρια που παρέχονται από το τουριστικό πληροφοριακό σύστημα στο οποίο εφαρμόζονται οι προδιαγραφές, καθώς και τα αντίστοιχα XML Schemata και η υλοποίηση της διαδικασίας αναζήτησης ξενοδοχείων.
- Περιγράφεται η υλοποίηση της υπηρεσίας αναζήτησης διαθέσιμων δωματίων. Περιγράφεται επίσης η χρήση των προτιμήσεων που περιέχονται στο προφίλ χρήστη και η διαδικασία της εναλλακτικής διαθεσιμότητας.
- Περιγράφεται η υλοποίηση της υπηρεσίας κράτησης δωματίων ξενοδοχείων. Περιγράφονται επίσης οι λειτουργίες διαχείρισης κρατήσεων – ανάκτηση, τροποποίηση και ακύρωση.

- Τέλος περιγράφεται η υλοποίηση της υπηρεσίας διαχείρισης του προφίλ χρήστη. Δίνονται τα XML Schemata των μηνυμάτων για δημιουργία, ανάκτηση, τροποποίηση και διαγραφή του προφίλ.

Το 5^ο κεφάλαιο περιέχει μια αναλυτικότερη περιγραφή των προδιαγραφών της ebXML για το CPA. Παρουσιάζονται τα κύρια σημεία του δίνοντας παραδείγματα από ένα CPA που χρησιμοποιήθηκε στην εργασία. Επίσης περιγράφεται ο τρόπος λειτουργίας και το γραφικό περιβάλλον της εφαρμογής CPA Editor που αναπτύχθηκε στα πλαίσια της εργασίας για την κατασκευή CPA.

Το 6^ο κεφάλαιο αναφέρεται στην εφαρμογή πελάτη (Demo Client) που αναπτύχθηκε για την χρήση των υπηρεσιών που διατίθενται από το σύστημα. Γίνεται αναλυτική περιγραφή όλων των σελίδων του γραφικού περιβάλλοντος της εφαρμογής, καθώς και της λειτουργίας της εφαρμογής και της διασύνδεσης της με τον ebXML OTA Server μέσω της υπηρεσίας μηνυμάτων της ebXML (ebMS).

Στο 7^ο κεφάλαιο περιγράφεται η εφαρμογή XML → DB + Middleware, η οποία αυτοματοποιεί τη διαδικασία μετατροπής ενός XML Schema σε σχεσιακό σχήμα, καθώς και την κατασκευή των κατάλληλων προγραμματιστικών εργαλείων για την διαχείριση της σχεσιακής βάσης δεδομένων – εισαγωγή, ανάκτηση και διαγραφή XML εγγράφων από τη βάση δεδομένων. Περιγράφεται επίσης η κλάση ObjectCopier που επιτρέπει την αντιγραφή αντικειμένων μιας ιεραρχίας κλάσεων σε αντικείμενα μιας άλλης ιεραρχίας κλάσεων.

Τέλος στο 8^ο κεφάλαιο δίνεται μια σύντομη ανακεφαλαίωση της παρούσας εργασίας και προτείνονται μελλοντικές επεκτάσεις του συστήματος.

Κεφάλαιο 2

Επισκόπηση Σχετικής Έρευνας

2.1 Εισαγωγή

Σ' αυτό το κεφάλαιο γίνεται μια επισκόπηση της τεχνολογίας των Web Services και παραθέτονται παραδείγματα υλοποίησης τους σε συνδυασμό με τις προδιαγραφές του OTA και της ebXML σε εμπορικές εφαρμογές και ερευνητικά προγράμματα για τον τουριστικό τομέα.

2.2 Ορισμός των Web Services

Ένας γενικός ορισμός των Web Services είναι: *«υπηρεσίες προσβάσιμες μέσω του διαδικτύου»*. Αυτός ο ορισμός όμως δεν είναι δόκιμος, καθώς δεν καταδεικνύει το υπόβαθρο και την δυναμική των Web Services, όπως επίσης προκαλεί σύγχυση με προϋπάρχουσες τεχνολογίες που ικανοποιούν τον παραπάνω ορισμό, όπως το RPC (Remote Procedure Call – Απομακρυσμένη Κλήση Μεθόδων). Ένας πιο δόκιμος σύντομος ορισμός είναι ότι πρόκειται για ένα *«σύνολο ανοιχτών προτύπων που επιτρέπουν την επικοινωνία εφαρμογών μέσω του διαδικτύου»* [26].

Ο θεμέλιος λίθος των Web Services είναι η XML. Τα δεδομένα δηλαδή που ανταλλάσσονται μεταξύ των εφαρμογών κωδικοποιούνται σε μορφή XML εγγράφων. Από εκεί και μετά μπορούμε να διαχωρίσουμε τις Web Services σε δύο κατηγορίες: τις **βασικές** ή **απλές** και τις **συνεργατικές** Web Services. Οι **απλές** Web Services λειτουργούν σαν RPC (ακολουθούν δηλαδή το μοντέλο αίτησης / απόκρισης) και έχει επικρατήσει να αποκαλούνται αυτές μόνο σαν Web Services.

Οι απλές Web Services είναι ουσιαστικά το σύνολο των παρακάτω ανοιχτών προτύπων:

- WSDL– Web Services Description Language [36]. Καθορίζει τη γλώσσα περιγραφής των Web Services.

- SOAP – Simple Object Access Protocol [30]. Καθορίζει τον τρόπο με τον οποίο μεταδίδονται τα μηνύματα.
- UDDI – Universal Description, Discovery and Integration [33]. Καθορίζει τον τρόπο καταχώρησης, αποθήκευσης και αναζήτησης των Web Services.

Με τον όρο **συνεργατικές** Web Services περιγράφεται η λειτουργικότητα που επιτρέπει την ολοκληρωμένη διασύνδεση μεταξύ εφαρμογών διαφορετικών επιχειρήσεων (B2B Integration), καθώς και ενδοεταιρικών εφαρμογών (Enterprise Application Integration - EAI). Τέτοιου είδους διασυνδέσεις απαιτούν προηγμένα χαρακτηριστικά, όπως ασφαλή και εγγυημένη ανταλλαγή μηνυμάτων, μοντελοποίηση και δημοσιοποίηση επιχειρησιακών διαδικασιών, διαπραγμάτευση των παραμέτρων συνεργασίας με τυποποιημένο και αυτοματοποιημένο τρόπο. Χαρακτηριστικός εκπρόσωπος των συνεργατικών Web Services είναι η ebXML.

Οι **απλές** και οι **συνεργατικές** Web Services παρουσιάζουν αρκετά κοινά χαρακτηριστικά. Τα σημαντικότερα είναι τα παρακάτω:

- Coarse-grained – Οι Web services συνήθως είναι το εξωτερικό περίβλημα του συνδυασμού πολλών μικρότερων εξειδικευμένων υπηρεσιών.
- Loosely coupled – Οι εφαρμογές που επικοινωνούν μέσω Web services δεν γνωρίζουν τις λεπτομέρειες της υλοποίησης τους. Κι αυτό γιατί οι Web services είναι ανεξάρτητες από το εκάστοτε hardware και software.
- Self-describing – Όλες οι πληροφορίες που απαιτούνται για την πρόσβαση στις υπηρεσίες αυτές, καθώς και χαρακτηριστικά αξιοπιστίας και παράμετροι ασφάλειας, περιγράφονται με τυποποιημένο τρόπο.

Από την άλλη πλευρά, υπάρχουν ορισμένες διαφορές ανάμεσα τους, οι οποίες δίνονται συνοπτικά στον παρακάτω πίνακα:

Πίνακας 1: Διαφορές των Web Services

Απλές Web Services	Συνεργατικές Web Services
Παρέχουν ένα απλοϊκό μοντέλο αλληλεπίδρασης, με απλές αιτήσεις και αποκρίσεις.	Παρέχουν σύνθετο μοντέλο αλληλεπίδρασης, με αρκετές ανταλλαγές μηνυμάτων.
Έχουν περιορισμένες απαιτήσεις ασφάλειας.	Έχουν υψηλές απαιτήσεις ασφάλειας.
Δεν υποστηρίζουν επιχειρησιακές συνεργασίες.	Υποστηρίζουν επιχειρησιακές συνεργασίες.

Είναι σύγχρονες και μη αξιόπιστες.	Είναι ασύγχρονες και αξιόπιστες.
------------------------------------	----------------------------------

Σημαντικό όμως είναι να τονίσουμε ότι οι προδιαγραφές των Web Services εξελίσσονται συνεχώς με στόχο την σύγκλιση και την μετεξέλιξή τους σε ένα ολοκληρωμένο σύνολο προδιαγραφών που να καλύπτει τις αυξανόμενες απαιτήσεις του αναπτυσσόμενου ψηφιακού επιχειρηματικού τοπίου.

2.3 Υιοθέτηση των Web Services από την Τουριστική Βιομηχανία

Πολλές από τις μεγαλύτερες επιχειρήσεις στο χώρο του τουρισμού, αναγνωρίζοντας τα πλεονεκτήματα που προσφέρουν οι Web Services, προχώρησαν στην υλοποίηση των δικών τους λύσεων. Η Sabre και η Datalex είναι από τους πρώτους που ανέπτυξαν OTA Web Services.

Η **Sabre**, ένα από τα μεγαλύτερα GDS, ανέπτυξε Web Services βασισμένες στις προδιαγραφές του OTA, και παρέχει όλη τη λειτουργικότητα που απαιτείται για την πώληση ενός τουριστικού προϊόντος - κράτηση πτήσεων, ενοικίαση οχημάτων, κράτηση ξενοδοχείων, πληροφορίες πελάτη. Παρέχει έτσι τη δυνατότητα στα τουριστικά πρακτορεία να μειώσουν το χρόνο και το κόστος ανάπτυξης νέων και βελτιωμένων εφαρμογών[28].

Ένα άλλο μεγάλο GDS, το Galileo, προχώρησε στην ανάπτυξη Web Services σε παγκόσμια κλίμακα, ελαττώνοντας, όπως ισχυρίζεται, τον χρόνο ανάπτυξης κατά 80% [15]. Το Galileo παρέχει ένα σύνολο από έτοιμες υπηρεσίες που απευθύνονται στις πιο κοινά χρησιμοποιούμενες υπηρεσίες, όπως είναι η κράτηση και η τροποποίηση κρατήσεων. Συγκεκριμένα, οι υπηρεσίες που παρέχει το Galileo είναι:

- Δημιουργία κράτησης – συνδυασμός λειτουργιών κρατήσεων σε μία κλήση προς το GDS.
- Σχεδιασμός ταξιδιού – ολοκληρωμένη διαδικασία για μεταφορά, ενοικίαση οχήματος και διαμονής σε ξενοδοχείο με μια κλήση.
- Πληροφορίες πτήσεων σε πραγματικό χρόνο.
- Επισκόπηση δρομολογίου – επιτρέπει την ανάκτηση και επισκόπηση λεπτομερών πληροφοριών για το δρομολόγιο.

2.4 Ερευνητικά Προγράμματα

Αρκετές ερευνητικές προσπάθειες επικεντρώνονται στον τουριστικό τομέα, επιδιώκοντας να αντιμετωπίσουν τα προβλήματα διαλειτουργικότητας που υφίστανται έως σήμερα. Οι προσπάθειες που έχουν γίνει ή βρίσκονται σε εξέλιξη, αφορούν είτε στην υιοθέτηση νέων τεχνολογιών, όπως οι Web Services, ή στην χρήση οντολογιών και την συνένωση διαφορετικών τεχνολογιών σε σημασιολογικό επίπεδο, ή στο συνδυασμό των δύο προηγούμενων κατευθύνσεων.

- **Το πρόγραμμα Harmonise**

Το Harmonise [16] είναι ένα πρόγραμμα της Ευρωπαϊκής Ένωσης για τον τουρισμό που έχει ολοκληρωθεί. Οι στόχοι του προγράμματος ήταν

- ♦ Η υποστήριξη μιας διαδικασίας κοινά αποδεκτής από τους κυριότερους συντελεστές του τουριστικού τομέα στην Ευρώπη, με σκοπό να αποτελέσει μια μεγάλη δύναμη στα τουριστικά πρότυπα και την διαλειτουργικότητα.
- ♦ Ο καθορισμός ενός ελάχιστου συνόλου εννοιών σαν αφετηρία για ένα πρότυπο «μεσολάβησης» για τον τουρισμό.
- ♦ Ο καθορισμός μιας οντολογίας που να καλύπτει αυτό το σύνολο εννοιών, και η υποστήριξη αυτής της διαδικασίας με τεχνικά μέσα χρησιμοποιώντας εργαλεία κατασκευής και διαχείρισης οντολογιών.
- ♦ Η κατασκευή μιας βάσης γνώσης που να περιγράφει πρότυπα και έννοιες σαν σημείο αναφοράς για τον τουριστικό τομέα.
- ♦ Η εκκίνηση μιας διαδικασίας συμβατότητας που να επιτρέπει σε διαφορετικούς συνεργάτες να διατηρήσουν τη δική τους οντολογία και δομή δεδομένων επιτυγχάνοντας ταυτόχρονα την μεταξύ τους συνεργασία.
- ♦ Η ανάπτυξη ενός βιώσιμου επιχειρησιακού μοντέλου, που να επιτρέπει την είσοδο νέων χρηστών / συνεργατών, βάσει ελάχιστων κανόνων. [19]

- **Το πρόγραμμα FETISH**

Το FETISH (Federated European Tourism Information Systems Integration) [13] είναι ένα ακόμη πρόγραμμα της Ευρωπαϊκής Ένωσης που έχει ολοκληρωθεί. Η κύρια πρόταση του FETISH ήταν η σύνδεση πληροφοριακών συστημάτων και υπηρεσιών προστιθέμενης αξίας που στηρίζονται στο διαδίκτυο σε μια κοινότητα, όπου μια κρίσιμη μάζα Ευρωπαϊκών πηγών δεδομένων μπορούν να διαμοιραστούν σε ένα ευρύ γεωγραφικά δίκτυο.

Οι κύριοι τεχνικοί στόχοι του FETISH ήταν:

- ◆ Η δημιουργία της απαιτούμενης υποδομής για την υποστήριξη του δικτύου τουριστικών υπηρεσιών.
- ◆ Να δίνει τη δυνατότητα σε αυτούς που εισέρχονται στο δίκτυο να μπορούν να:
 - Προσφέρουν υπηρεσίες.
 - Έχουν πρόσβαση και να χρησιμοποιούν υπηρεσίες, και
 - Μοιράζονται υπηρεσίες
- ◆ Η υλοποίηση της υποδομής ώστε να υποστηρίζει την ολοκλήρωση των υφιστάμενων τουριστικών υπηρεσιών, παρέχοντας οδηγίες και εργαλεία για την προσαρμογή τους στο περιβάλλον του FETISH. [20]

• Το πρόγραμμα SATINE

Το SATINE [6] ήταν μια από τις πρώτες προσπάθειες για την ανάπτυξη Web Services βασισμένων στο OTA. Επιπλέον το SATINE προτείνει οντολογίες βασισμένες στο OTA, για τον σημασιολογικό εμπλουτισμό των Web Services για τον τουρισμό, καθώς επίσης και τον εμπλουτισμό των μητρώων των Web Services με σημασιολογική πληροφορία.

Στα πλαίσια του SATINE αναπτύχθηκε μια οντολογία για τα μηνύματα του Amadeus¹⁰, κατασκευάστηκε μια ολοκληρωμένη οντολογία για το OTA και χρησιμοποιήθηκαν αντιστοιχίσεις σε επίπεδο οντολογίας για την πρόσβαση Web Services του Amadeus μέσω συμβατών με το OTA μηνυμάτων. Τα παραπάνω έχουν σαν αποτέλεσμα να είναι δυνατή η πρόσβαση στο σύστημα του Amadeus

¹⁰ Το Amadeus, όπως αναφέρθηκε και στην εισαγωγή της εργασίας είναι ένα από τα τέσσερα μεγαλύτερα GDSs.

χρησιμοποιώντας Web Services, χωρίς να είναι απαραίτητη η γνώση της δομής των μηνυμάτων του Amadeus. Προτάθηκε επίσης η ανάπτυξη ενός φίλτρου ερωτήσεων συμβατό με OWL-QL [24] για τα μητρώα της ebXML [12].

- **Το πρόγραμμα COTRIN**

Το COTRIN (Comprehensive Ontology for the Travel Industry) [5] ήταν ένα πρόγραμμα που αποσκοπούσε στη δημιουργία μιας οντολογίας για τον τουρισμό χρησιμοποιώντας τις προδιαγραφές του OTA.

Ο στόχος του COTRIN ήταν η οντολογία να αναπτυχθεί σε βάθος, να είναι σύνθετη, να περιλαμβάνει πλούσιες σημασιολογικές σχέσεις μεταξύ των κόμβων των ταξινομιών της, οι οποίες προκύπτουν από τα δεδομένα και την ορολογία που βρίσκεται στις XML προδιαγραφές του OTA. Το COTRIN θα αναπτυχθεί στη Γλώσσα Οντολογιών του Διαδικτύου (OWL) [23].

Όπως αναφέρθηκε, όλες οι παραπάνω ερευνητικές προσπάθειες είχαν ως στόχο την επίτευξη διαλειτουργικότητας στα πληροφοριακά συστήματα του τουριστικού τομέα. Η χρήση οντολογιών, για την σύνδεση διαφορετικών συστημάτων σε σημασιολογικό επίπεδο, εμφανίζεται να είναι ο κύριος άξονας στον οποίο κινούνται οι περισσότερες προσπάθειες. Οι οντολογίες που αναπτύχθηκαν στηρίχθηκαν είτε σε αυτόνομες προσπάθειες, με βάση την εμπειρία πολλών μελών της τουριστικής βιομηχανίας, είτε στις πληροφορίες που συσσωρεύονται στις προδιαγραφές του OTA, οι οποίες αντανακλούν με οργανωμένο και δομημένο τρόπο τις ανάγκες και τις απαιτήσεις των τουριστικών επιχειρήσεων. Επίσης οι Web Services που αναπτύχθηκαν σε συνδυασμό με τις προδιαγραφές του OTA (SATINE) στηρίχθηκαν στο απλό μοντέλο Web Services (SOAP, WSDL), που αναφέρθηκε στην αρχή του παρόντος κεφαλαίου, και όχι σε σύνθετα μοντέλα Web Services όπως η ebXML που προτείνεται και από τις προδιαγραφές του OTA.

Σε αντίθεση με τα παραπάνω ερευνητικά προγράμματα, η παρούσα εργασία έχει ως κύριο στόχο την εφαρμογή των προδιαγραφών του OTA στην υλοποίηση πρότυπων Web Services, καθώς και την υλοποίηση της απαραίτητης υποδομής για την

μελλοντική υποστήριξη περισσότερο εξεζητημένων υπηρεσιών και λειτουργιών. Αναλυτικότερα οι κύριοι στόχοι της εργασίας είναι:

- Ανάπτυξη Web Services ακολουθώντας τις οδηγίες των προδιαγραφών του OTA.
- Χρήση των προδιαγραφών της ebXML (ebMS και ebCPP/A) όπως προτείνεται στις προδιαγραφές του OTA.
- Υλοποίηση υποδομής ικανής να υποστηρίξει μελλοντικές επεκτάσεις και νέες υπηρεσίες.
- Υλοποίηση του προφίλ χρήστη που περιγράφει ο OTA, τόσο σε επίπεδο υποδομής (ανάπτυξη σχεσιακής βάσης δεδομένων), όσο και σε επίπεδο λειτουργικότητας (ανάπτυξη υπηρεσιών διαχείρισης του προφίλ και χρήση του για εξατομικευμένη αναζήτηση διαθεσιμότητας).
- Καταγραφή των τεχνικών προβλημάτων που εμφανίζονται στην υλοποίηση, επίλυση των προβλημάτων αυτών και σχεδίαση και ανάπτυξη μεθόδων και λογισμικού για την εύκολη και ταχεία ανάπτυξη παρόμοιων συστημάτων.
- Αξιολόγηση των μεθόδων και εργαλείων που χρησιμοποιήθηκαν στην υλοποίηση της εργασίας.

Η διαδικασία υλοποίησης σύνθετων προδιαγραφών διαλειτουργικότητας, όπως αυτές του OTA, είναι ιδιαίτερα σημαντική τόσο στην προσπάθεια προσαρμογής υπάρχοντων συστημάτων σε σύγχρονες προδιαγραφές, όσο και στην σχεδίαση νέων συστημάτων. Η γνώση και η εμπειρία που αποκτάται μέσα από αυτή την διαδικασία, μπορεί να βοηθήσει τόσο στην ταχύτερη, ευκολότερη και ορθότερη υλοποίηση των ίδιων ή ανάλογων προδιαγραφών, όσο και να συμβάλλει στην βελτίωση των ίδιων των προδιαγραφών και των μεθόδων και εργαλείων που χρησιμοποιούνται στην υλοποίηση.

2.5 Λειτουργικότητα Τουριστικών Πληροφοριακών Συστημάτων

Ο τουρισμός είναι από τους βασικότερους τομείς της ελληνικής οικονομίας. Η γεωγραφική θέση και οι κλιματολογικές συνθήκες της Ελλάδας, σε συνδυασμό με την πλούσια ιστορία και πολιτιστική κληρονομιά της, την καθιστούν έναν από τους πιο αγαπημένους τουριστικούς προορισμούς. Για αυτό τον λόγο υπάρχει ένας τεράστιος

αριθμός ξενοδοχείων και γενικά επιχειρήσεων διαμονής. Ειδικότερα η Κρήτη, που είναι και το μεγαλύτερο και νοτιότερο νησί της Ελλάδας, φιλοξενεί πολύ μεγάλο αριθμό τουριστών κάθε χρόνο.

Για την πληροφοριακή υποστήριξη και προώθηση του τουρισμού στην Ελλάδα λειτουργούν αρκετά τουριστικά πληροφοριακά συστήματα. Για να καταδειχθεί η εφαρμοσιμότητα των μεθοδολογιών και των σχεδιασμών που προτείνονται στην εργασία, κρίθηκε αναγκαία η επικοινωνία με ένα υπαρκτό σύστημα τουριστικών πληροφοριών που υλοποιεί μια πλούσια λειτουργικότητα.

Το πληροφοριακό σύστημα που χρησιμοποιήθηκε στην παρούσα εργασία, είναι ένα Destination Management System που καλύπτει την περιοχή της Κρήτης. Είναι δηλαδή ένα σύστημα που αποθηκεύει εκτενείς πληροφορίες σχετικά με το τουριστικό προϊόν της Κρήτης, όπως ξενοδοχεία, αξιοθέατα, εκδηλώσεις κ.α. Οι πληροφορίες αυτές χρησιμοποιούνται για την προώθηση του τουριστικού προϊόντος της Κρήτης και την προσέλκυση τουριστών.

Συνοπτικά το σύστημα περιλαμβάνει τις παρακάτω δυνατότητες

- Παρέχει έναν πλήρη τουριστικό οδηγό για την Κρήτη
- Περιέχει πληροφορίες για
 - Γεωγραφία
 - Μυθολογία
 - Ιστορία
 - Κουλτούρα
 - Χλωρίδα και πανίδα και πολλά άλλα
- Παρέχει εικονική πλοήγηση στην Κρήτη μέσα από πλούσιους διαδραστικούς χάρτες
- Δίνει τη δυνατότητα για αναζήτηση, εξέταση και επικοινωνία με κάθε είδους τουριστικές επιχειρήσεις της Κρήτης
- Παρέχει πληθώρα κριτηρίων για την αναζήτηση επιχειρήσεων

Κύρια χαρακτηριστικά του συστήματος είναι:

- η ολοκληρωμένη παροχή ηλεκτρονικών υπηρεσιών (διαφήμιση, πληροφόρηση και ηλεκτρονικό εμπόριο) σχετικά με το τουριστικό προϊόν της Κρήτης.

- πολύγλωσση πληροφόρηση (Ελληνικά, Αγγλικά, Γερμανικά, Ιταλικά, Γαλλικά και Νορβηγικά)
- πλήρης πληροφόρηση (περιοχές και αξιοθέατα του νησιού, αγκυροβόλια, επιχειρήσεις διαμονής, τουριστικά πρακτορεία, επιχειρήσεις ενοικίασης αυτοκινήτων και δικύκλων, επιχειρήσεις μεταφορών, εστιατόρια και επιχειρήσεις ψυχαγωγίας, επιχειρήσεις παραγωγής παραδοσιακών Κρητικών προϊόντων, κ.α.)
- εκτενής / πλούσια πληροφόρηση (πλούσιο φωτογραφικό υλικό, ψηφιακοί χάρτες, εκτενή κείμενα, κ.α.)

2.6 Ανακεφαλαίωση

Στο κεφάλαιο αυτό έγινε μια εισαγωγή στην τεχνολογία των Web Services, περιγράφοντας τις δύο διαφορετικές κατηγορίες που συναντώνται σήμερα, και εντοπίστηκαν τα κοινά τους σημεία καθώς και οι διαφορές τους. Επίσης έγινε αναφορά στην υιοθέτηση των Web Services από την τουριστική βιομηχανία παρουσιάζοντας συγκεκριμένα παραδείγματα. Στη συνέχεια έγινε μια σύντομη περιγραφή ερευνητικών προγραμμάτων που εξετάζουν την αντιμετώπιση του προβλήματος της έλλειψης διαλειτουργικότητας στον τουριστικό τομέα. Τέλος έγινε συνοπτική παρουσίαση ενός υπάρχοντος συστήματος διαχείρισης πληροφοριών διαμονής για την περιφέρεια της Κρήτης, το οποίο χρησιμοποιήθηκε στην εργασία για την εφαρμογή των OTA Web Services.

Όπως αναφέρθηκε, πολλές μεγάλες επιχειρήσεις της τουριστικής βιομηχανίας ανέπτυξαν Web Services για τα προϊόντα που ήδη παρείχαν. Οι υλοποιήσεις των τουριστικών επιχειρήσεων άλλοτε είναι συμβατές με τις προδιαγραφές του OTA και άλλοτε όχι. Από την άλλη πλευρά τα ερευνητικά προγράμματα που αναπτύχθηκαν ή βρίσκονται υπό ανάπτυξη, είχαν ως κύριο άξονα το γεφύρωμα του χάσματος μεταξύ των διαφορετικών τουριστικών επιχειρήσεων σε σημασιολογικό επίπεδο. Σε κάποιες περιπτώσεις, η σημασιολογική πληροφορία που χρησιμοποιήθηκε προήλθε από τις προδιαγραφές του OTA, αναγνωρίζοντας έτσι την πληρότητα και την έκταση των προδιαγραφών του.

Από την άλλη πλευρά, η παρούσα εργασία έχει ως αντικείμενο αφενός την υποστήριξη των προδιαγραφών του OTA σε ένα υπάρχον τουριστικό πληροφοριακό σύστημα, αφετέρου την χρησιμοποίηση των προδιαγραφών αυτών σαν μοντέλο δεδομένων για την κατασκευή σχεσιακών βάσεων δεδομένων. Αυτές οι βάσεις δεδομένων χρησιμοποιούνται για την επέκταση της λειτουργικότητας του υπάρχοντος τουριστικού πληροφοριακού συστήματος, ώστε να υποστηρίζεται η διαχείριση κρατήσεων ξενοδοχείων και προφίλ χρηστών, και της διάθεσης OTA Web Services για τη χρήση αυτής της λειτουργικότητας με τρόπο ανοικτό και διαλειτουργικό.

Στο επόμενο κεφάλαιο γίνεται μια σύντομη περιγραφή των προδιαγραφών της ebXML και ιδιαίτερα εκείνων που χρησιμοποιεί ο OTA.

Κεφάλαιο 3

Περιγραφή των προδιαγραφών της ebXML και του ΟΤΑ

3.1 Εισαγωγή στην ebXML

Η ebXML (Electronic Business eXtensible Markup Language), είναι ένα σύνολο προδιαγραφών για επιχειρηματικές συνεργασίες. “Δίνει τη δυνατότητα σε επιχειρήσεις οποιουδήποτε μεγέθους και σε οποιαδήποτε γεωγραφική θέση να συνεργαστούν μέσω του διαδικτύου. Χρησιμοποιώντας την ebXML, οι επιχειρήσεις μπορούν με τυποποιημένο τρόπο να ανταλλάσσουν επιχειρησιακά μηνύματα, να διεξάγουν εμπορικές συναλλαγές, να επικοινωνήσουν δεδομένα με κοινούς όρους και να καθορίσουν και να καταχωρήσουν επιχειρησιακές διαδικασίες.” [8]

Η ebXML είναι το αποτέλεσμα της κοινής πρωτοβουλίας των UN /CEFACT και OASIS με βασικό στόχο να αναπτυχθεί μια ενιαία παγκόσμια ηλεκτρονική αγορά βασισμένη σε μια ανοικτή δημόσια υποδομή, με βάση την XML, που να επιτρέπει τη χρήση των ηλεκτρονικών επιχειρησιακών πληροφοριών κατά τρόπο διαλειτουργικό, ασφαλή και συνεπή από όλα τα συμβαλλόμενα μέρη. Ο στόχος αυτός επιτυγχάνεται με την ανάπτυξη ενός ενιαίου συνόλου από διεθνώς συμφωνημένες προδιαγραφές, οι οποίες αναπτύσσονται έχοντας κατά νου τα παρακάτω κριτήρια:

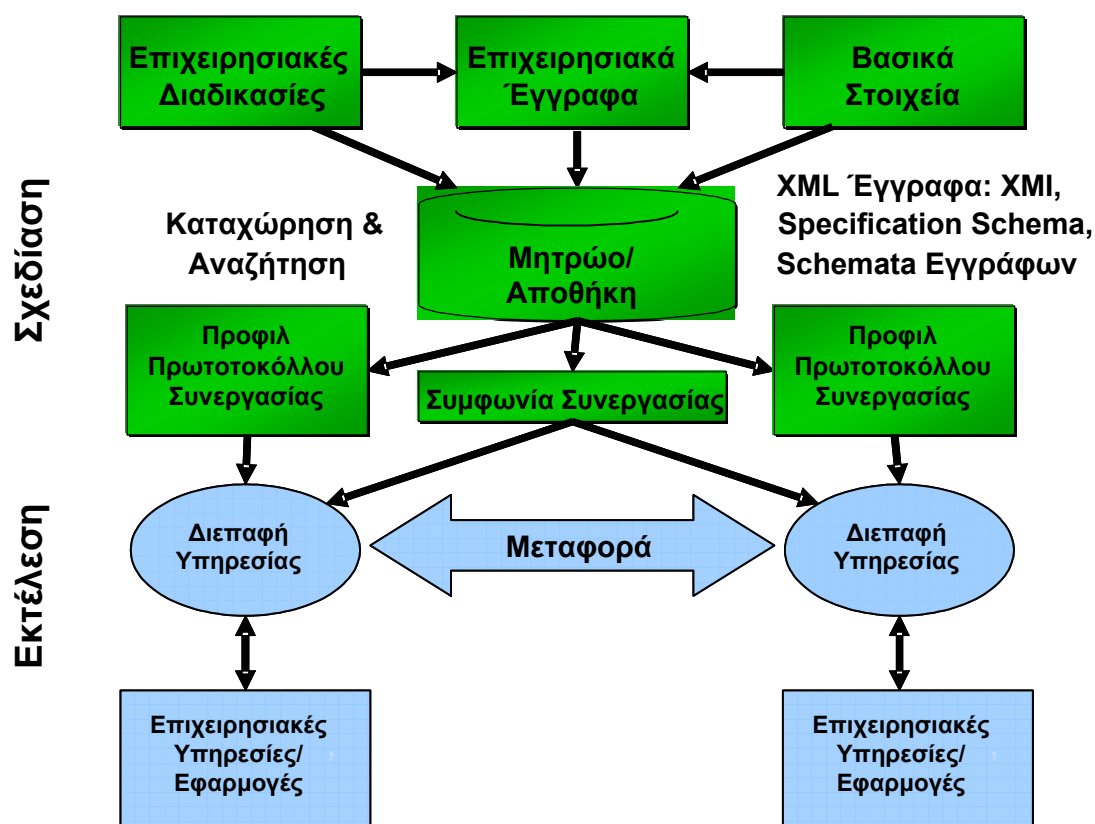
- Να επιτρέπουν το απλό, εύκολο και «πανταχού παρών» ηλεκτρονικό επιχειρείν μέσω της χρήσης της XML.
- Την ελαχιστοποίηση του κόστους του ηλεκτρονικού επιχειρείν
- Τη διατήρηση πλήρους συμβατότητας με τις τεχνικές προδιαγραφές της XML όπως αυτές ορίζονται από το W3C [38].
- Την παροχή πολυγλωσσικής υποστήριξης
- Την μεγιστοποίηση της διαδραστικότητας και αποδοτικότητας παρέχοντας παράλληλα ένα τρόπο μετάβασης από το EDI (Electronic Data Interchange) [7].

3.2 Αρχιτεκτονική της ebXML

Η αρχιτεκτονική της ebXML μπορεί να χωριστεί σε δύο τμήματα: το τμήμα της σχεδίασης και το τμήμα της εκτέλεσης (Εικόνα 2).

Στο τμήμα της σχεδίασης περιλαμβάνονται οι επιχειρησιακές διαδικασίες που υποστηρίζουν οι επιχειρήσεις, τα επιχειρησιακά έγγραφα που ανταλλάσσουν και τα προφίλ συνεργασίας των επιχειρήσεων μαζί με τυχόν υπάρχουσες συμφωνίες συνεργασίας. Αυτά αποθηκεύονται και καταχωρούνται στο μητρώο και χρησιμοποιούνται για την αναζήτηση και διαπραγμάτευση μεταξύ των επιχειρήσεων.

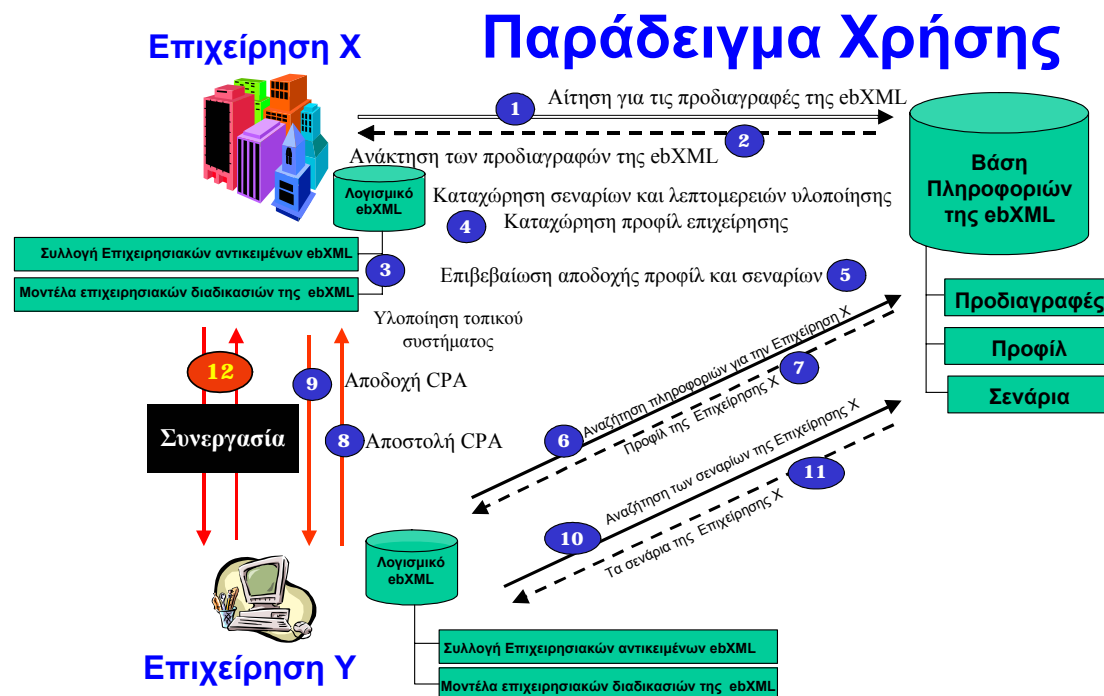
Στο τμήμα της εκτέλεσης, οι επιχειρήσεις που έχουν ήδη συμφωνήσει στον τρόπο συνεργασίας, έχουν δηλαδή κατασκευάσει σε ένα CPA, συνεργάζονται ανταλλάσσοντας μηνύματα μέσω της υπηρεσίας μηνυμάτων (ebMS).



Εικόνα 2 Αρχιτεκτονική της ebXML

Σενάριο Χρήσης

Στην παρακάτω εικόνα δίνεται ένα παράδειγμα χρήσης της ebXML (Εικόνα 3).



Εικόνα 3 Παράδειγμα χρήσης της ebXML

Αρχικά, μια επιχείρηση «X» αναλύει μια επιχειρηματική διαδικασία με στόχο να κατασκευάσει ηλεκτρονικές συνδιαλλαγές που υλοποιούν αυτή τη διαδικασία. Αυτό μπορεί να το επιτύχει χρησιμοποιώντας τη μεθοδολογία μοντελοποίησης του UN/CEFACT (Unified Modeling Methodology - UMM) ή κάποια άλλη μεθοδολογία. Το αποτέλεσμα αυτής της προσπάθειας είναι μια τυπική περιγραφή της επιχειρηματικής διαδικασίας σύμφωνα με το σχήμα καθορισμού επιχειρηματικών διαδικασιών της ebXML (ebXML Business Process Specification Schema - BPSS). Ένα τμήμα αυτής της εργασίας μπορεί να είναι ο καθορισμός των τύπων των επιχειρηματικών πληροφοριών που θα χρειαστεί να ανταλλαχθούν. Σ' αυτό το τμήμα μπορεί να χρησιμοποιηθούν τα ebXML Core Components ή άλλοι τύποι περιεχομένου. Τα BPSS που κατασκευάζονται καθώς και τεχνικές λεπτομέρειες του συστήματος ηλεκτρονικού εμπορίου της επιχείρησης χρησιμοποιούνται για τον καθορισμό του προφίλ του πρωτοκόλλου συνεργασίας της επιχείρησης (ebXML

Collaboration Protocol Profile - CPP). Τα BPSS και τα CPPs που κατασκευάστηκαν αποθηκεύονται στο ebXML Repository και καταχωρούνται στο ebXML Registry.

Μια άλλη επιχείρηση «Υ» που επιθυμεί να επεκτείνει τις ηλεκτρονικές της συνεργασίες, μπορεί να χρησιμοποιήσει το Μητρώο της ebXML (ebXML Registry) για να αναζητήσει επιχειρηματικές διαδικασίες στις οποίες μπορεί να παίξει συγκεκριμένο ρόλο – για παράδειγμα αγοραστής ή πωλητής ενός προϊόντος. Καταχωρεί επίσης στο Μητρώο το δικό της προφίλ συνεργασίας (CPP) ώστε να μπορεί κάποια άλλη ενδιαφερόμενη επιχείρηση να την ανακαλύψει. Εξετάζοντας το προφίλ της επιχείρησης «Χ» αποφασίζει να διαπραγματευτεί μαζί της μια συμφωνία συνεργασίας. Όταν και οι δύο επιχειρήσεις συμφωνήσουν στους όρους συνεργασίας κατασκευάζουν μια συμφωνία πρωτοκόλλου συνεργασίας (ebXML Collaboration Protocol Agreement - CPA). Το CPA στο οποίο κατέληξαν χρησιμοποιείται για να καθορίζει τους κανόνες στην ανταλλαγή μηνυμάτων μέσω της Υπηρεσίας Μηνυμάτων (ebXML Message Service - ebMS).

3.3 Επισκόπηση των στοιχείων που απαρτίζουν την ebXML

- **Business Process Specification Schema (BPSS) – Σχήμα Περιγραφής Επιχειρησιακών Διαδικασιών:**

Η BPSS είναι μια γλώσσα προδιαγραφών βασισμένη στην XML η οποία περιγράφει τις «δημόσιες» επιχειρησιακές διαδικασίες. Εστιάζεται στην συνεργασία των εμπορικών συνεργατών, στις δυαδικές συνεργασίες στις οποίες οι δύο αυτοί συνεργάτες εμπλέκονται (δηλαδή σε ζεύγη) και στις δραστηριότητες των επιχειρησιακών συνδιαλλαγών που εκτελούν στα πλαίσια αυτών των συνεργασιών. Η BPSS επηρεάζεται σημαντικά από την UMM, μια μεθοδολογία μοντελοποίησης που αναπτύχθηκε από το UN /CEFACT, αλλά δεν την απαιτεί.

- **Core Components (CC)- Βασικά Συστατικά**

Είναι τα βασικά συστατικά που παρέχουν την επιχειρησιακή πληροφορία που κωδικοποιείται στα επιχειρησιακά έγγραφα τα οποία ανταλλάσσονται μεταξύ των εμπορικών συνεργατών. Μπορούν να συντεθούν σε δομές από δημόσια ή ιδιωτικά

μητρώα επιτυγχάνοντας έτσι την επαναχρησιμοποίηση κοινών επιχειρησιακών δομών.

- **Registry / Repository (Reg / Rep) – Μητρώο / Αποθήκη**

Είναι η περιγραφή του τρόπου αποθήκευσης και καταχώρισης των επιχειρησιακών διαδικασιών, των προφίλ των επιχειρήσεων και όλων των απαραίτητων στοιχείων για την επίτευξη μιας ηλεκτρονικής επιχειρησιακής συνεργασίας.

- **Collaboration Protocol Profile / Agreement (CPP/A) – Προφίλ Πρωτοκόλλου Συνεργασίας / Συμφωνία Πρωτοκόλλου Συνεργασίας**

Είναι XML έγγραφα που περιγράφουν τις ικανότητες ηλεκτρονικής συνεργασίας μιας επιχείρησης και τις συμφωνίες ηλεκτρονικής συνεργασίας δύο επιχειρήσεων αντίστοιχα. Είναι στενά συνδεδεμένα με την BPSS. Σε συνδυασμό με την ebMS παρέχουν πληροφορίες σχετικές με την ηλεκτρονική σύνδεση. Σε συνδυασμό με το Μητρώο υποστηρίζουν την αναζήτηση επιχειρήσεων και τη διαδικασία εγκατάστασης ηλεκτρονικών συνεργασιών.

- **Message Service (ebMS) – Υπηρεσία Ανταλλαγής Μηνυμάτων**

Η Υπηρεσία Μηνυμάτων της ebXML παρέχει έναν κομψό, γενικού σκοπού μηχανισμό για την ανταλλαγή μηνυμάτων. Είναι αρκετά «ώριμη» υπηρεσία, και ουσιαστικά απαιτείται η χρήση της από πολλά άλλα τμήματα της ebXML, όπως για παράδειγμα για την πρόσβαση στο Μητρώο. Στηρίζεται στο SOAP with Attachments και μπορεί να μεταφέρει περιεχόμενο οποιουδήποτε τύπου.

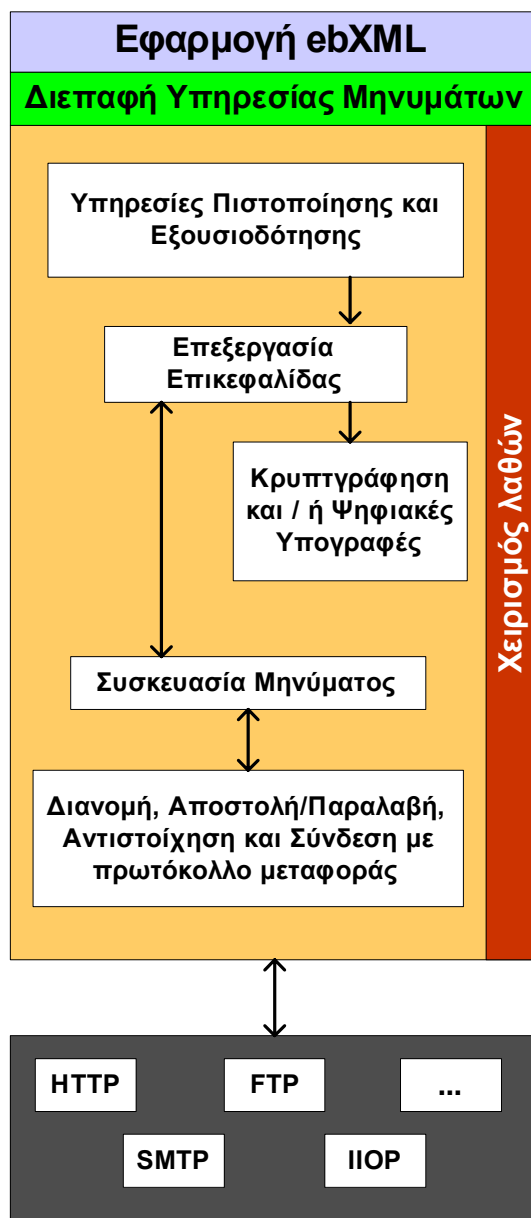
3.3 Περιγραφή της ebMS

Η Υπηρεσία μηνυμάτων της ebXML (ebMS) καθορίζει τη μορφή των μηνυμάτων, καθώς και την συμπεριφορά της εφαρμογής που ανταλλάσσει τα ebXML μηνύματα. Η εφαρμογή μπορεί να υλοποιηθεί σαν αυτόνομο λογισμικό μετάδοσης μηνυμάτων, ή να αποτελεί τμήμα της λειτουργικότητας ενός ολοκληρωμένου συστήματος ηλεκτρονικού εμπορίου ή εξυπηρετητή εφαρμογών (Application Server).

Μπορούμε να θεωρήσουμε ότι αποτελείται από τρία επίπεδα:

- Μια αφαιρετική διεπαφή υπηρεσιών, την οποία χρησιμοποιούν οι εφαρμογές για να έχουν πρόσβαση στην υπηρεσία μηνυμάτων
- Τις λειτουργίες της υπηρεσίας μηνυμάτων και
- Την αντιστοίχιση στα πρωτόκολλα μετάδοσης (π.χ. HTTP ή SMTP)

Το παρακάτω διάγραμμα δίνει μια επισκόπηση των διάφορων λειτουργιών που παρέχονται από την υπηρεσία μηνυμάτων (Εικόνα 4).



Εικόνα 4 Τα τμήματα του ebXML Message Handler

Οι λειτουργικές απαιτήσεις της υπηρεσίας μηνυμάτων περιλαμβάνουν:

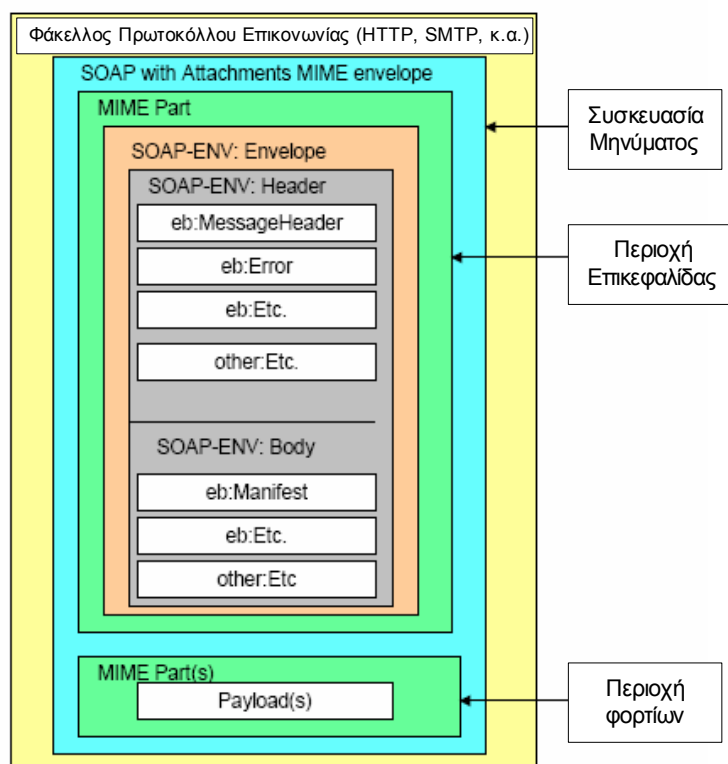
- Επεξεργασία επικεφαλίδας (κατασκευή της επικεφαλίδας από παραμέτρους που περνά η εφαρμογή που καλεί την υπηρεσία)
- Ανάλυση της επικεφαλίδας (το αντίθετο της επεξεργασίας)
- Υπηρεσίες ασφάλειας (δημιουργία και επαλήθευση ψηφιακών υπογραφών, πιστοποίηση και εξουσιοδότηση)
- Αξιόπιστη μετάδοση μηνυμάτων (αποθήκευση μηνυμάτων, επαναμετάδοση, ειδοποίηση σφάλματος, αναφορά παραλαβής)
- Συσκευασία των δεδομένων προς αποστολή σαν συνημμένο σε ένα μήνυμα SOAP with Attachments και τέλος
- Χειρισμός σφαλμάτων

Η ebMS καθορίζει τη σύνδεση με τα πρωτόκολλα HTTP και SMTP αλλά επιτρέπει την χρησιμοποίηση και άλλων πρωτοκόλλων. Επίσης δεν θέτει περιορισμούς στον τύπο των δεδομένων που θα μεταφερθούν, επιτρέποντας έτσι τη χρήση της σε μια ευρεία γκάμα εφαρμογών.

3.3.1 Ανατομία ενός ebXML μηνύματος

Η Υπηρεσία Μηνυμάτων της ebXML τοποθετείται πάνω από το SOAP με συνημμένα. Το SOAP με συνημμένα ενσωματώνει ένα SOAP φάκελο σαν πρώτο τμήμα ενός MIME container, το οποίο περιέχει όλη την απαραίτητη πληροφορία για την δρομολόγηση και γι' αυτό επιτυγχάνεται πολύ καλή απόδοση στην επεξεργασία του μηνύματος. Άλλο περιεχόμενο μπορεί να επισυναφθεί σαν πρόσθετο τμήμα MIME.

Στο παρακάτω διάγραμμα δίνεται η γενική δομή ενός ebXML μηνύματος (Εικόνα 5)



Εικόνα 5 Γενική δομή ενός ebXML μηνύματος

3.4 Περιγραφή των Collaboration Protocol Profile και Agreement (CPP/A)

Το CPP είναι ένα XML έγγραφο που περιγράφει, με τρόπο κατανοητό για τον υπολογιστή, τα χαρακτηριστικά που διαθέτει μια επιχείρηση όσον αφορά την εμπλοκή της σε μια ηλεκτρονική συνδιαλλαγή μέσω ebXML. Συγκεκριμένα το CPP περιέχει:

- Το όνομα της, και στοιχεία για την επικοινωνία με την επιχείρηση
- Τα πρωτόκολλα μετάδοσης και ασφάλειας που υποστηρίζει
- Τα πρωτόκολλα μετάδοσης μηνυμάτων που υποστηρίζει
- Τις επιχειρησιακές διαδικασίες που υποστηρίζει
- Κάθε είδους τεχνικά χαρακτηριστικά όπως URLs, πιστοποιητικά και οτιδήποτε άλλο είναι απαραίτητο για την διεξαγωγή ηλεκτρονικών συνδιαλλαγών μέσω της ebXML και περιγράφονται στις προδιαγραφές [10].

Πρόκειται λοιπόν για την περιγραφή μιας επιχείρησης στα πλαίσια της ebXML. Μια επιχείρηση μπορεί να έχει πολλά διαφορετικά CPPs, π.χ. για να περιγράψει διαφορετικές λειτουργίες που υποστηρίζει.

Όταν μια επιχείρηση επιθυμεί να συνεργαστεί ηλεκτρονικά μέσω ebXML με μια άλλη επιχείρηση, χρησιμοποιεί το δικό της CPP και το CPP της άλλης επιχείρησης για να κατασκευάσει και να διαπραγματευτεί την τελική συμφωνία μεταξύ τους, η οποία θα εκφραστεί με ένα CPA.

Το CPA περιγράφει τα χαρακτηριστικά μιας συγκεκριμένης συνεργασίας μεταξύ δύο επιχειρήσεων. Περιέχει τις πληροφορίες που περιέχονται στα CPPs και των δύο επιχειρήσεων και περιγράφει τον τρόπο με τον οποίο συνεργάζονται. Το CPA χρησιμοποιείται από τα ebXML συστήματα και των δύο συνεργατών για την ρύθμιση τους, ώστε τα συστήματα τους να επιβάλλουν τους κανόνες που ορίζονται στο CPA όταν «τρέχει» η μεταξύ τους συνεργασία. Περισσότερες λεπτομέρειες για το CPA και την δομή του δίνονται στο κεφάλαιο 5 όπου περιγράφεται η εφαρμογή CPA Editor η οποία χρησιμοποιείται για την κατασκευή CPA.

3.5 Εισαγωγή στο OTA

Ο Open Travel Alliance ιδρύθηκε τον Μάιο του 1999 και είναι μια κοινοπραξία από προμηθευτές σε όλους τους τομείς της τουριστικής βιομηχανίας, συμπεριλαμβανομένων αεροπορικών εταιριών, εταιριών ενοικίασης αυτοκινήτων, ξενοδοχείων, ταξιδιωτικών πρακτορείων, και χειριστών ταξιδιών, καθώς και σχετιζόμενων εταιριών που παρέχουν την υποστήριξη διανομής και τεχνολογίας στην τουριστική βιομηχανία. Η κοινοπραξία έχει τώρα πάνω από 125 μέλη που αντιπροσωπεύουν σημαντικά ονόματα σε όλους τους τομείς της τουριστικής βιομηχανίας.

Ο βασικός στόχος του OTA είναι να σχεδιάσει βιομηχανικές προδιαγραφές ικανές να εκμεταλλευτούν τις επικοινωνίες συστημάτων που είναι διαθέσιμες με την διαδικτυακή συνδεσιμότητα. Για να επιτύχει αυτούς τους στόχους, ο OTA έχει σχεδιάσει προδιαγραφές έχοντας ως γνώμονα τα ακόλουθα κριτήρια:

- **Ανοιχτότητα:** Οι προδιαγραφές του OTA είναι δημόσια διαθέσιμες.
- **Ευελιξία:** Παρέχουν την ελάχιστη λειτουργικότητα που απαιτείται για την αξιόπιστη επικοινωνία μεταξύ των τουριστικών συστημάτων, δίνοντας ταυτόχρονα την ευελιξία στις επιχειρήσεις να αναπτύξουν νέες βελτιωμένες υπηρεσίες.

- **Ανεξαρτησία πλατφόρμας:** Οι προδιαγραφές αναπτύχθηκαν με γνώμονα τη χρήση τους από οποιαδήποτε πλατφόρμα υλικού και λογισμικού που μπορεί υποστηρίζει τα κοινά πρότυπα που χρησιμοποιούνται στις προδιαγραφές.
- **Ασφάλεια:** Δίνεται μεγάλη σημασία στην προστασία των πληροφοριών από μη εξουσιοδοτημένη χρήση, καθώς και στην ανάγκη να δοθεί στον πελάτη ο έλεγχος της δημιουργίας, ανανέωσης και ανταλλαγής των δεδομένων με άλλους συνεργάτες.
- **Επεκτασιμότητα:** Οι προδιαγραφές επεκτείνονται συνεχώς, με τρόπο όμως που να ελαχιστοποιεί την ασυμβατότητα με προηγούμενες εκδόσεις.
- **Διεθνής εμβέλεια:** Οι αρχικές προδιαγραφές γράφηκαν στα αγγλικά, εντούτοις ο OTA σκοπεύει να επεκτείνει τις νεώτερες εκδόσεις ώστε να παρέχουν αναπαράσταση σε σύνολα χαρακτήρων που υποστηρίζουν το πρότυπο Unicode.

3.6 Επισκόπηση της Αρχιτεκτονικής του OTA

Η αρχιτεκτονική υποδομής του που προτείνει ο OTA δανείζεται κυρίως από την Τεχνική Αρχιτεκτονική [ebTA] και την Υπηρεσία Μηνυμάτων [ebMS] της ebXML. Η Υπηρεσία Μηνυμάτων της ebXML, η οποία βασίζεται στο SOAP έκδοση 1.1 και στο SOAP με συνημμένα, παρέχει τη λειτουργικότητα που απαιτείται ώστε δύο ή περισσότεροι συνεργάτες να εμπλακούν σε μια «ηλεκτρονική επιχειρηματική συνδιαλλαγή».

Οι επιχειρηματικές εφαρμογές διασυνδέονται με ένα ebMS μέσω μιας «διεπαφής υπηρεσιών», η οποία είναι μοναδική σε κάθε προϊόν ebMS. Η «διεπαφή υπηρεσιών» επιτρέπει στις εφαρμογές να κάνουν αίτηση για υπηρεσίες και να λαμβάνουν ειδοποιήσεις από ένα προϊόν ebMS.

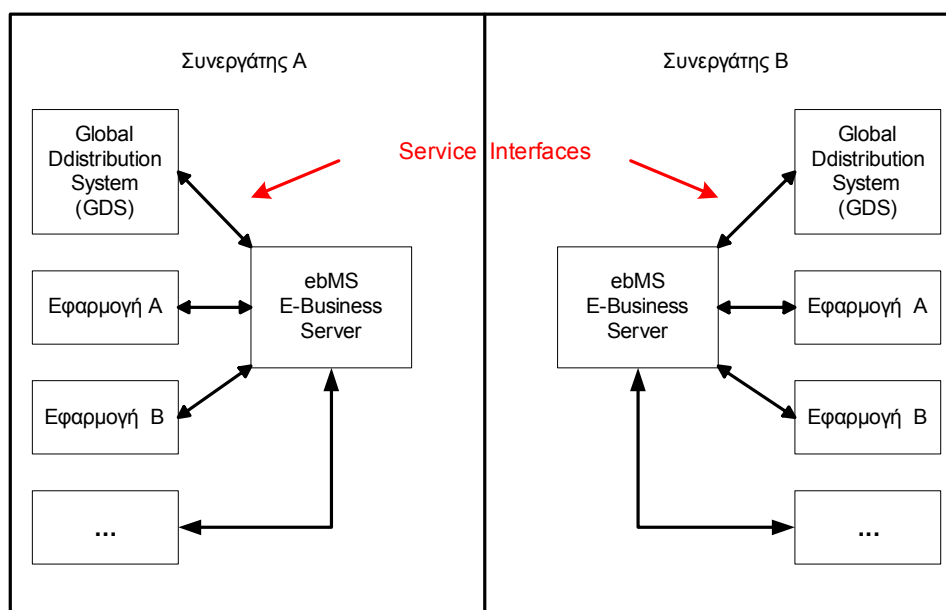
Όταν μια εφαρμογή χρειάζεται να εμπλακεί σε μια ηλεκτρονική συνδιαλλαγή με έναν συνεργάτη, πρέπει να καλέσει την τοπική ebMS και να εγκαταστήσει ένα ασφαλές και αξιόπιστο session με το σύστημα ebMS/OTA του συνεργάτη. Από τη στιγμή που εγκατασταθεί ένα session τα ebMS/OTA συστήματα ανταλλάσσουν πληροφοριακό υλικό του τύπου «service/action» που θα ζητηθεί. Ένα session μπορεί να

περιλαμβάνει πολλαπλές ανταλλαγές μηνυμάτων μεταξύ των ebMS/OTA συστημάτων δύο ή περισσότερων συνεργατών.

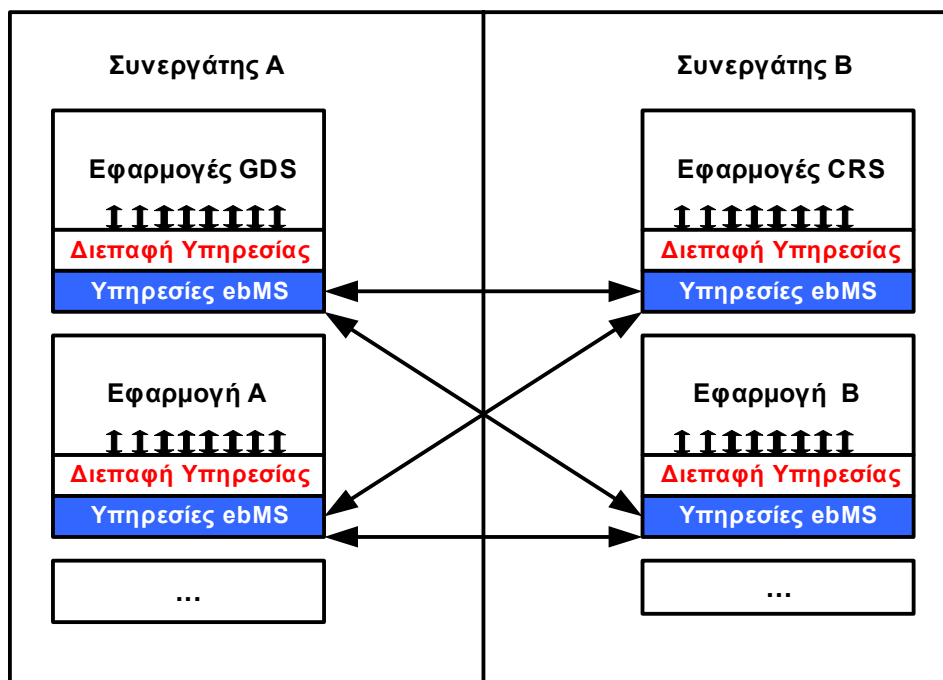
3.6.1 Μοντέλο Αναφοράς

Ένα ebMS μπορεί να χρησιμοποιηθεί σαν ένας κεντρικός “E-Business Server” και να διαχειρίζεται όλες τις ηλεκτρονικές συνδιαλλαγές μιας επιχείρησης με τους συνεργάτες της. Εναλλακτικά, η λειτουργικότητα του ebMS μπορεί ενσωματωθεί σε κάθε εφαρμογή που εμπλέκεται σε ηλεκτρονικές επιχειρησιακές συνδιαλλαγές.

Οι δύο παραπάνω περιπτώσεις αναπαρίστανται σχηματικά στις παρακάτω εικόνες (Εικόνα 6) και (Εικόνα 7)



Εικόνα 6 Μοντέλα Αναφοράς της αρχιτεκτονικής του OTA



Εικόνα 7 Μοντέλο Αναφοράς της αρχιτεκτονικής του OTA

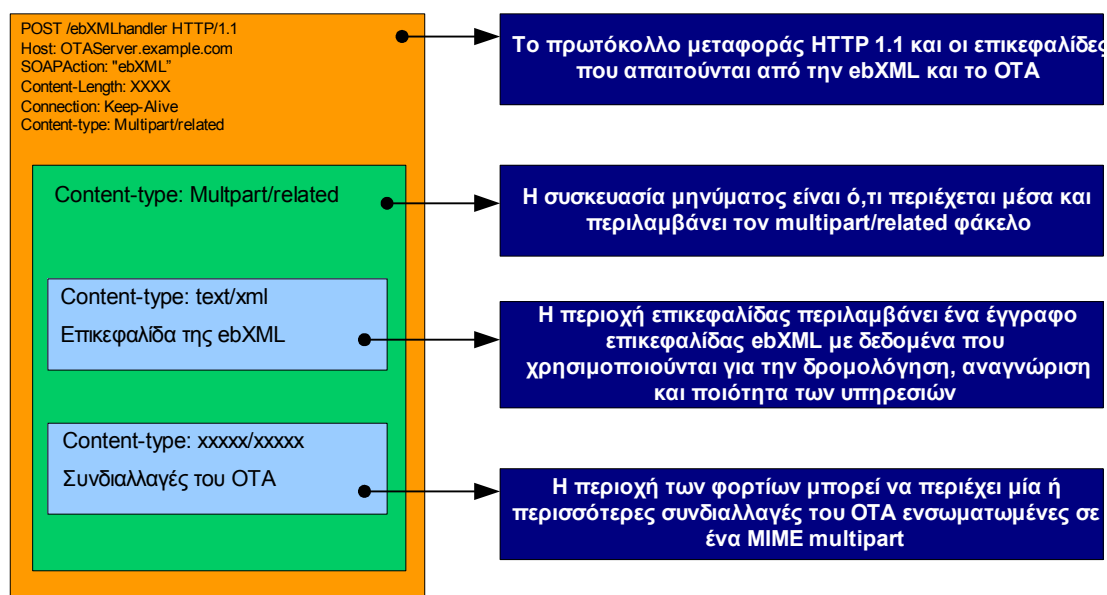
3.7 Δομή και συσκευασία μηνυμάτων

Ένα μήνυμα ebXML, ακολουθώντας τις συμβάσεις του OTA, ΠΡΕΠΕΙ να περιέχει έναν χώρο επικεφαλίδας και μηδέν ή ένα χώρο για το φορτίο. Και οι δύο χώροι τυλίγονται από ένα ενιαίο MIME/Multipart φάκελο και ολόκληρη η συσκευασία αναφέρεται ως *συσκευασία μηνυμάτων*.

Οι δύο χώροι μέσα σε μια *συσκευασία μηνυμάτων* περιγράφονται παρακάτω:

- Το πρώτο μέρος MIME, που αναφέρεται ως *Χώρος Ετικέτας*, περιέχει ένα έγγραφο επικεφαλίδας ebXML. Το έγγραφο επικεφαλίδας περιλαμβάνει μόνο εκείνα τα στοιχεία και τις ιδιότητες που απαιτούνται από το OTA, οι λεπτομέρειες του οποίου διευκρινίζονται παρακάτω.
- Το προαιρετικό δεύτερο μέρος MIME, που αναφέρεται ως *χώρος φορτίου*, περιέχει φορτία επιπέδου εφαρμογής που περιέχουν υλικό επιχειρησιακών δεδομένων για την υπηρεσία / δράση που προσδιορίζεται στην επικεφαλίδα ebXML. Ο χώρος φορτίων μπορεί να περιέχει μια ή περισσότερες επιχειρησιακές συνδιαλλαγές OTA.

Η γενική δομή και η σύνθεση ενός μηνύματος OTA συμβατού με ebXML περιγράφονται στην ακόλουθη εικόνα (Εικόνα 8).



Εικόνα 8 Η γενική δομή ενός συμβατού με ebXML μηνύματος του OTA

Όταν μια εφαρμογή χρειάζεται να εμπλακεί σε μια ηλεκτρονική επιχειρηματική συνδιαλλαγή με έναν συνεργάτη, πρέπει να καλέσει την τοπική ebMS υπηρεσία, μέσω της διεπαφής υπηρεσιών, ώστε να εγκατασταθεί μια ασφαλής και αξιόπιστη συνεδρία με το ebMS/OTA σύστημα του συνεργάτη. Από τη στιγμή που εγκαθίσταται μια συνεδρία, τα ebMS/OTA συστήματα ανταλλάσσουν πληροφορία σχετική με τον τύπο της «υπηρεσίας / δράσης» που ζητήθηκε. Μια συνεδρία μπορεί να περιλαμβάνει πολλαπλές ανταλλαγές μηνυμάτων μεταξύ των ebMS/OTA συστημάτων από δύο ή περισσότερους συνεργάτες.

3.8 Γενικά μηνύματα και το πρότυπο Υπηρεσία / Δράση

Αν και πολλά μηνύματα αφορούν συγκεκριμένο τομέα της τουριστικής βιομηχανίας (π.χ. ξενοδοχεία) άλλα μηνύματα είναι γενικά εφαρμόσιμα και μπορεί να χρησιμοποιηθούν ευρύτερα. Τα περισσότερα μηνύματα του OTA ορίζονται σε ζεύγη αίτησης / απάντησης (αν και η υποδομή υποστηρίζει την αξιόπιστη διανομή μονόδρομων μηνυμάτων τύπου ειδοποίησης).

Ο OTA καθορίζει ένα απλό μοντέλο Υπηρεσία / Δράση (Service / Action) που περικλείει όλα τα καθορισμένα μηνύματα. Το μοντέλο Υπηρεσία / Δράση είναι εννοιολογικά παρόμοιο με το πρότυπο των Web Services, όπου η εφαρμογή παρέχει

μια ή περισσότερες υψηλού επιπέδου υπηρεσίες, κάθε μια από τις οποίες έχει μια ή περισσότερες εφαρμόσιμες ενέργειες (αυτές ισούνται με τις μεθόδους που είναι διαθέσιμες στις υπηρεσίες). Όταν χρησιμοποιούνται από κοινού με τα ζεύγη μηνυμάτων αίτησης / απάντησης, το μοντέλο είναι εννοιολογικά όμοιο με το RPC (Remote Procedure Call), αν και στην πραγματικότητα το υπόστρωμα μηνυμάτων δεν χρειάζεται να στηρίζεται σε RPCs.

Η ebXML παρέχει τα πεδία Service και Action για την μετάδοση πληροφοριών σχετικά με την προοριζόμενη επεξεργασία ενός μηνύματος ebXML. Από την άποψη της υπηρεσίας μηνυμάτων της ebXML αυτές οι τιμές απλά και μόνο μεταφέρονται και παρέχονται για να επιτρέψουν στα συμβαλλόμενα μέρη να κατευθύνουν τα μηνύματά τους για την επεξεργασία ή τη δράση στις συγκεκριμένες υπηρεσίες μέσα στα συστήματά τους.

3.9 Οι Υπηρεσίες που παρέχει ο OTA

Με βάση και όσα προαναφέρθηκαν, τα XML Schemata του OTA κατηγοριοποιούνται με βάση τον τομέα στον οποίο αναφέρονται. Για παράδειγμα η Υπηρεσία HotelBooking που αναφέρεται σε ξενοδοχεία περιλαμβάνει τις ενέργειες:

- OTA_HotelSearchRQ - για την αναζήτηση ξενοδοχείων
- OTA_HotelAvailRQ - για την αναζήτηση διαθεσιμότητας
- OTA_HotelResRQ - για την κράτηση ξενοδοχείων

Στον παρακάτω πίνακα συνοψίζονται οι Υπηρεσίες (Services) που παρέχει ο OTA

OTA Service	Περιγραφή
Profile	Αυτή η υπηρεσία παρέχει λειτουργίες δημιουργίας και διαχείρισης (ανάκτηση, τροποποίηση και διαγραφή) στα προφίλ των πελατών
VehicleBooking	Υπηρεσία για τον έλεγχο διαθεσιμότητας, κράτηση, τροποποίηση ή / και ακύρωση ενοικίασης οχήματος
AirBooking	Υπηρεσία για τον έλεγχο διαθεσιμότητας και κρατήσεων αεροπορικών πτήσεων
TravelInsurance	Υπηρεσία σχετική με την τουριστική ασφάλεια.
HotelBooking	Υπηρεσία για την αναζήτηση και εύρεση ξενοδοχείων, την εύρεση διαθεσιμότητας, την κράτηση, τροποποίηση ή / και ακύρωση μιας

	διαμονής σε ξενοδοχείο, καθώς και για άλλες λειτουργίες που σχετίζονται με τα ξενοδοχεία.
HotelResNotification	Υπηρεσία για την διανομή ξενοδοχειακών κρατήσεων μεταξύ συστημάτων.
HotelPropertyInformation	Υπηρεσίες που αφορούν σε στατιστικά στοιχεία, όπως είναι στοιχεία διαμονής, που χρειάζονται να μεταδοθούν σε άλλα συστήματα.
MeetingProfile	Υπηρεσία για την δημιουργία / τροποποίηση προφίλ συνάντησης, π.χ. για ομάδες ανθρώπων ή συνέδρια.
PackageBooking	Υπηρεσία για διαθεσιμότητα , κράτηση, τροποποίηση , ακύρωση τουριστικών πακέτων.
Session	Υπηρεσία υποδομής του OTA για την εγκατάσταση και τερματισμό των Sessions.
GolfTeeTimes	Υπηρεσία για την κράτηση χρόνου για ένα παιχνίδι γκολφ ή την αναζήτηση διαθέσιμου μαθήματος γκολφ

Πίνακας 2: Οι Υπηρεσίες που παρέχει ο OTA

Στην παρούσα εργασία υλοποιήθηκαν οι υπηρεσίες Profile, HotelBooking και Session που σχετίζονται με την διαχείριση του προφίλ χρήστη, την αναζήτηση ξενοδοχείων, διαθεσιμότητας και διαχείρισης κρατήσεων και την διαχείριση των OTA Sessions αντίστοιχα. Οι ενέργειες (Actions) που μπορούν να εκτελεστούν ως μέθοδοι των υπηρεσιών Profile, HotelBooking¹¹ και Session περιγράφονται στους παρακάτω πίνακες.

Profile Action	Αναμενόμενη Απόκριση	Σε απόκριση	Περιγραφή
OTA_CreateProfileRQ	OTA_CreateProfileRS	-	Αίτηση για δημιουργία νέου προφίλ
OTA_CreateProfileRS	-	OTA_CreateProfileRQ	Απόκριση στην αίτηση για δημιουργία προφίλ
OTA_ReadRQ*	OTA_ReadProfileRS	-	Αίτηση για ανάγνωση ενός προφίλ
OTA_ReadProfileRS	-	OTA_ReadRQ	Απόκριση στην αίτηση για ανάγνωση ενός προφίλ
OTA_UpdateRQ*	OTA_UpdateRS	-	Αίτηση για ανανέωση των περιεχομένων ενός προφίλ

¹¹ Ο OTA περιγράφει και άλλες ενέργειες που μπορούν να εκτελεστούν στην υπηρεσία HotelBooking, όμως στην παρούσα εργασία υλοποιήθηκαν αυτές που σχετίζονται με την αναζήτηση ξενοδοχείων, διαθεσιμότητας και διαχείρισης κρατήσεων.

OTA_UpdateRS	-	OTA_UpdateRQ	Απόκριση στην αίτηση για ανανέωση των περιεχομένων ενός προφίλ
OTA_DeleteRQ*	OTA_DeleteRS	-	Αίτηση για διαγραφή ενός προφίλ
OTA_DeleteRS	-	OTA_DeleteRQ	Απόκριση στην αίτηση για διαγραφή ενός προφίλ

HotelBooking Action	Αναμενόμενη Απόκριση	Σε απόκριση	Περιγραφή
OTA_HotelSearchRQ	OTA_HotelSearchRS	-	Αίτηση για αναζήτηση ξενοδοχείων
OTA_HotelSearchRS	-	OTA_HotelSearchRQ	Απόκριση στην αίτηση για αναζήτηση ξενοδοχείων (τα αποτελέσματα της αναζήτησης)
OTA_HotelAvailRQ	OTA_HotelAvailRS	-	Αίτηση για αναζήτηση διαθεσιμότητας
OTA_HotelAvailRS	-	OTA_HotelAvailRQ	Απόκριση στην αίτηση για αναζήτηση διαθεσιμότητας (τα αποτελέσματα της αναζήτησης διαθεσιμότητας)
OTA_ResRQ	OTA_ResRS	-	Αίτηση για κράτηση ξενοδοχείων
OTA_ResRS	-	OTA_ResRQ	Απόκριση στην αίτηση για κράτηση ξενοδοχείων
OTA_CancelRQ*	OTA_CancelRS	-	Αίτηση για ακύρωση μιας κράτησης
OTA_CancelRS	-	OTA_CancelRQ	Απόκριση στην αίτηση για ακύρωση μιας κράτησης (επιτυχία ή αποτυχία της ακύρωσης)

Session Action	Αναμενόμενη Απόκριση	Σε απόκριση	Περιγραφή
OTA_CreateRQ	OTA_CreateRS	-	Αίτηση για δημιουργία ενός OTA Session
OTA_CreateRS	-	OTA_CreateRQ	Απόκριση στην αίτηση για δημιουργία ενός OTA Session
OTA_CloseRQ	OTA_CloseRS	-	Αίτηση για κλείσιμο ενός

			OTA Session
OTA_CloseRS	-	OTA_CloseRQ	Απόκριση στην αίτηση για κλείσιμο ενός OTA Session

*Τα συγκεκριμένα μηνύματα είναι γενικά και χρησιμοποιούνται σε όλους τους τομείς της τουριστικής βιομηχανίας που καλύπτονται από τις προδιαγραφές του OTA.

Πιο λεπτομερής περιγραφή των υπηρεσιών και των ενεργειών τους δίνεται στα αντίστοιχα κεφάλαια όπου μελετάται η υλοποίησή τους.

3.10 Περιγραφή των προδιαγραφών του OTA για το προφίλ πελάτη

Οι προδιαγραφές του OTA για το Προφίλ Πελάτη καθορίζουν με λεπτομέρεια το περιεχόμενο του προφίλ πελάτη από την άποψη της τουριστικής βιομηχανίας. Οι προδιαγραφές παρέχουν ένα σύνολο κοινών μηνυμάτων για την εισαγωγή και μετάδοση των δεδομένων του προφίλ, και για την ανταλλαγή πληροφοριών των προφίλ μεταξύ τουριστικών υπηρεσιών της βιομηχανίας.

3.11 Περιεχόμενο του προφίλ

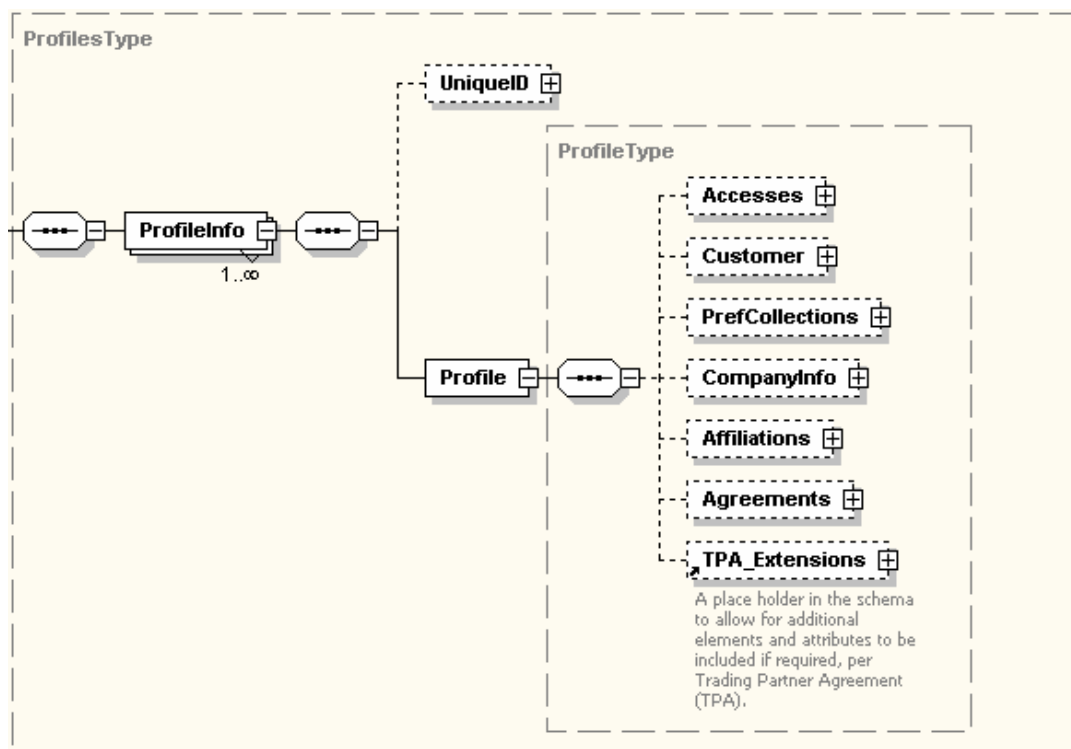
Το προφίλ περιλαμβάνει βασικές πληροφορίες σχετικά με έναν πελάτη ή μια επιχείρηση, καθώς επίσης και για οικονομικές συνδιαλλαγές και επαφές του. Το προφίλ ορίζει επίσης συλλογές από προτιμήσεις για συγκεκριμένους τύπους ταξιδιού συμπεριλαμβάνοντας σημαντικές τουριστικές υπηρεσίες υποστήριξης, όπως είναι ταξιδιωτικά πρακτορεία και ασφάλεια. Τα προφίλ περιέχουν επίσης πληροφορίες σχετικά με ένταξη σε οργανισμούς και αναγνώριση πιστοποιήσεων και συμμαχιών που κατέχουν οι επιχειρήσεις στις επιχειρηματικές τους σχέσεις. Δεν περιλαμβάνονται πληροφορίες κοστολόγησης ούτε δεδομένα για πολιτικές σε σχέση με τον τουρισμό ή απαιτήσεις ενός οργανισμού.

Συνοπτικά, τα επιμέρους τμήματα που απαρτίζουν το προφίλ είναι τα εξής:

- **UniqueID** – Μοναδικό αναγνωριστικό του προφίλ.

- **Προσβάσεις** – μια συλλογή από στοιχεία που περιέχουν το χρονικό των πρόσφατων προσβάσεων στο προφίλ, σημειώνοντας την ημερομηνία και την ώρα της δημιουργίας, και ανανεώσεις ή επόμενες ενέργειες που εκτελέστηκαν στην εγγραφή του προφίλ.
- **Πελάτης** – πληροφορίες σχετικά με τον ταξιδιώτη που είναι απαραίτητες για τον καθορισμό και την τεκμηρίωση της ταυτότητας ενός ατόμου, μέσα επικοινωνίας, τρόπους πληρωμής, και βασικές ανάγκες και ενδιαφέροντα για τουριστικές υπηρεσίες.
- **Προτιμήσεις** – γενικές και ειδικές συνθήκες για την ικανοποίηση των τουριστικών αναγκών ενός πελάτη ή μιας επιχείρησης για συγκεκριμένο σκοπό όπως είναι επαγγελματικά ταξίδια, οικογενειακές διακοπές, εκδρομές για γκολφ ή διεθνή ταξίδια. Το προφίλ μπορεί επίσης να χρησιμοποιηθεί για να καταδείξει προτιμήσεις για συγκεκριμένου τύπου τουριστικές υπηρεσίες όπως είναι αεροπορικά ταξίδια, διαμονή σε ξενοδοχείο, ενοικιάσεις αυτοκινήτων και άλλους τύπους τουριστικών υπηρεσιών.
- **Εντάξεις (Affiliations)** – οργανισμοί με τους οποίους ο πελάτης ή η επιχείρηση διατηρεί μια σχέση αποκομίζοντας έτσι πλεονεκτήματα ή προνόμια, όπως π.χ. εργαζόμενος, ομάδες ενδιαφέροντος, συνδρομές σε οργανισμούς, travel arrangers, ασφαλιστικές εταιρίες και χρηματοδοτούμενες από χορηγούς τουριστικές λέσχες όπως εκείνες που παρέχονται από τις αεροπορικές εταιρίες. Αυτό το τμήμα περιλαμβάνει μόνο δεδομένα σχετικά με τη σχέση με αυτούς τους οργανισμούς (π.χ. αναγνωριστικός αριθμός εργαζόμενου) και όχι πολιτικές για τις ίδιες τις εταιρίες.
- **Πληροφορίες για την Επιχείρηση** – πληροφορίες σχετικά με μια επιχείρηση ή μια επιχειρηματική οντότητα, όπως διευθύνσεις, τηλέφωνα και άλλες μορφές επικοινωνίας, τρόπους πληρωμής, loyalty programs κ.α.
- **Συμφωνίες** – πληροφορίες σχετικά με πιστοποιήσεις και συμμαχίες τις οποίες μπορεί να διατηρεί μια επιχείρηση.
- **Συμφωνημένες Επεκτάσεις (TPA_Extensions)** – λόγω της ιδιωτικής φύσης των συμφωνιών στις επιχειρηματικές σχέσεις, το στοιχείο **TPA_Extensions** περιλαμβάνεται σ' αυτό το τμήμα του Προφίλ και επιτρέπει στις επιχειρήσεις να προσθέσουν επιλεγμένες πληροφορίες σύμφωνα με τις ανάγκες τους.

Στη συνέχεια δίνεται η γενική δομή του XML Schema του OTA για το προφίλ (Εικόνα 9)



Εικόνα 9 Το XML Schema του OTA για το προφίλ χρήστη

3.12 Η ροή των μηνυμάτων και των λειτουργιών του προφίλ

Τα μηνύματα που υποστηρίζονται από τις προδιαγραφές περιλαμβάνουν τις βασικές ενέργειες της δημιουργίας, ανάκτησης, ενημέρωσης ενός τμήματος του προφίλ ή της αντικατάστασης ολόκληρου του προφίλ, και της διαγραφής προφίλ.

Αυτά τα τέσσερα ζεύγη αίτησης / απόκρισης καθορίζουν το ρόλο και τη λειτουργία των ανταλλαγών μηνυμάτων μεταξύ των εμπορικών συνεργατών για τα προφίλ πελατών. Οι ενέργειες «Δημιουργία», «Ανάγνωση», «Αντικατάσταση» και «Διαγραφή» αναφέρονται σε ολόκληρες τις εγγραφές των προφίλ, ενώ η ενέργεια «Ενημέρωση» αναφέρεται σε τμήματα του προφίλ.

3.12.1 Μηνύματα σχετικά με το προφίλ

Παρακάτω δίνονται οι δηλώσεις ενέργειας (actions) που χρησιμοποιούνται σαν αρχικά στοιχεία ενός εγγράφου που χρησιμοποιείται για την ανταλλαγή δεδομένων προφίλ:

- **OTA_CreateProfileRQ** – δηλώνει την αρχική δημιουργία ενός προφίλ

- **OTA_CreateProfileRS** – δηλώνει την έκβαση (επιτυχία ή μη) της δημιουργίας ενός προφίλ
- **OTA_ReadRQ** – δηλώνει την αίτηση για ανάγνωση ενός προφίλ όταν συνδυάζεται με UniqueId τύπου «Profile»
- **OTA_ReadProfileRS** – Επιστρέφει το προφίλ πελάτη σε απάντηση μιας αίτησης «Ανάγνωσης» τύπου «Profile»
- **OTA_UpdateRQ** – δηλώνει την αίτηση για ενημέρωση της εγγραφής ενός προφίλ πελάτη όταν συνδυάζεται με UniqueId τύπου «Profile». Η συγκεκριμένη δράση(εις) που πρόκειται να εφαρμοστεί στην εγγραφή του προφίλ καθορίζεται από τα «παιδιά» του στοιχείου <Position>.
- **OTA_UpdateRS** – δηλώνει την έκβαση (επιτυχία ή μη) της ενημέρωσης ενός προφίλ
- **OTA_DeleteRQ** – δηλώνει την αίτηση για διαγραφή ενός προφίλ χρήστη όταν συνδυάζεται με UniqueId τύπου «Profile».
- **OTA_DeleteRS** - δηλώνει την έκβαση (επιτυχία ή μη) της διαγραφής ενός προφίλ.

3.12.2 Ιδιότητες διαφύλαξης προσωπικών δεδομένων

Δύο σύνολα ιδιοτήτων επιτρέπουν τους πελάτες και τις επιχειρήσεις να προσδιορίσουν ποια στοιχεία θα ήθελαν για να μοιραστούν για το συγχρονισμό των προφίλ τους σε άλλες θέσεις και για τη λήψη διαφημιστικών πληροφοριών. Η προδιαγραφή απαιτεί τις ιδιότητες ShareAllSynchInd και ShareAllMarketInd στο στοιχείο «Profile» για την άδεια να μοιραστούν όλα τα στοιχεία του προφίλ για συγχρονισμό και διαφημιστικούς σκοπούς αντίστοιχα. Αυτές οι ιδιότητες πρέπει να έχουν προκαθορισμένη την τιμή «No» και να απαιτούν ρητή έγκριση από τον πελάτη για να αλλαχτούν σε «Yes».

Η προδιαγραφή δίνει επίσης στον πελάτη ή την επιχείρηση τον έλεγχο της διαμοίρασης σημαντικών στοιχείων μέσα στο προφίλ. Οι ιδιότητες ShareSynchInd και ShareMarketInd στα σημαντικότερα στοιχεία επιτρέπουν στον ιδιοκτήτη του προφίλ να καθορίσει την άδεια να διαμοιραστούν τα δεδομένα σε αυτά τα στοιχεία για συγχρονισμό και διαφημιστικούς σκοπούς. Αυτές οι ιδιότητες πρέπει να έχουν

προκαθορισμένη την τιμή «Inherit», η οποία δείχνει ότι το στοιχείο κληρονομεί την τιμή που ορίζεται στο αμέσως υψηλότερο επίπεδο. Εντούτοις, μια τιμή εκτός από «Inherit» πρέπει να αγνοήσει την άδεια που χορηγείται στα πιο υψηλά επίπεδα στην ιεραρχία για εκείνο το δεδομένο στοιχείο. Για παράδειγμα, μια τιμή «No» σε ShareAllMarketInd στο προφίλ θα ισχύει για ολόκληρο το προφίλ, εκτός αν έχει αλλαχτεί σε «Yes» στα ShareMarketInd πεδία συγκεκριμένων στοιχείων στο προφίλ.

3.12.3 Πληροφορίες για τον πελάτη

Το στοιχείο **Customer** καλύπτει τα βασικά δεδομένα σχετικά με τον πελάτη, περιλαμβάνοντας όνομα και διεύθυνση, τρόπους επικοινωνίας, ταξιδιωτικά κείμενα, τρόπους πληρωμής και αναφορές σε σχετιζόμενους ταξιδιώτες, όπως είναι μέλη της οικογένειας ή επιχειρηματικοί συνεργάτες.

3.12.3.1 Απαιτούμενα δεδομένα πελάτη

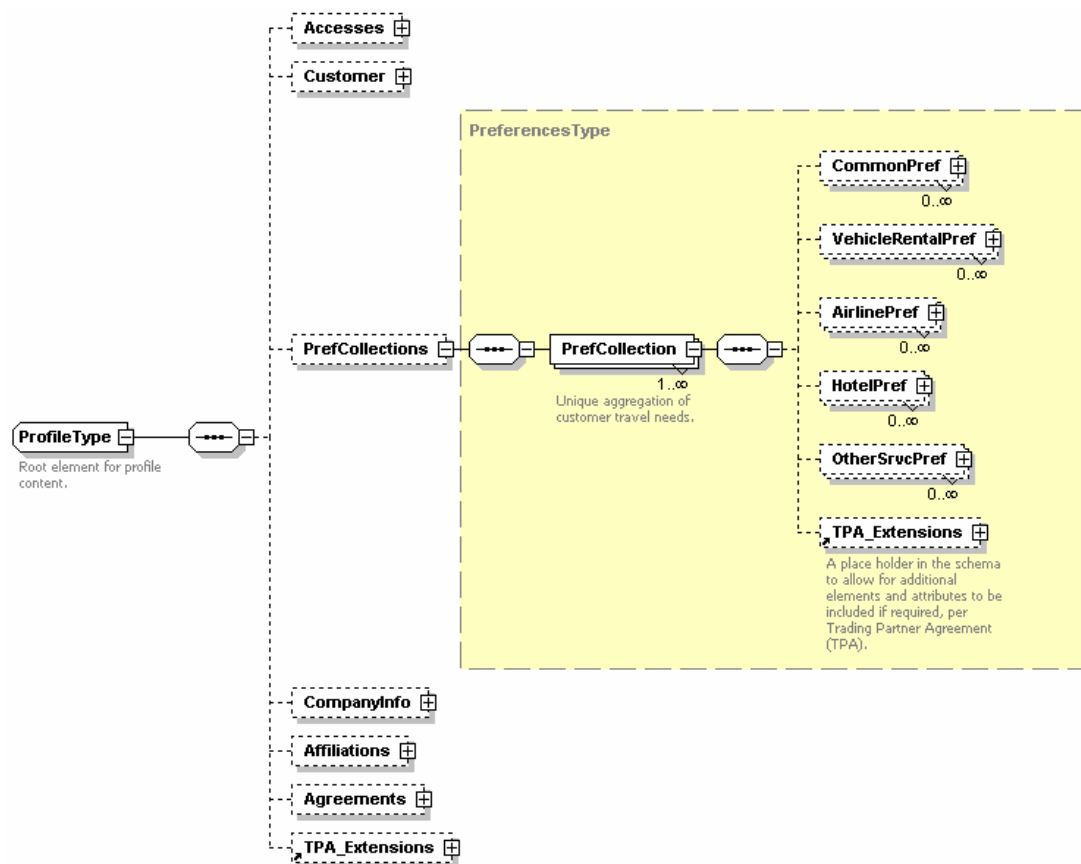
Η προδιαγραφή καθορίζει ένα ελάχιστο σύνολο δεδομένων που απαιτούνται για ένα προφίλ, με τα δεδομένα αυτά να τοποθετούνται στο τμήμα **Customer**. Κάθε προφίλ πρέπει να έχει δεδομένα για τουλάχιστον ένα από τα παρακάτω στοιχεία:

- Όνομα, ετικέτα <**PersonName**>
- Τηλέφωνο, ετικέτα <**Telephone**>
- Email, ετικέτα <**Email**>

3.12.4 Προτιμήσεις του πελάτη

Το τμήμα των Προτιμήσεων καταγράφει τις ανάγκες των ταξιδιωτών σε διάφορες μορφές. Επιτρέπει σε έναν πελάτη ή μια επιχείρηση να συγκεντρώσει αυτές τις απαιτήσεις σε συγκεκριμένες συλλογές προτιμήσεων και να σχηματίσει ένα προκαθορισμένο σύνολο, καθώς και να ονοματίσει τις ομάδες διάφορων τουριστικών εμπειριών π.χ. επιχειρηματικές, διεθνείς, οικογενειακές διακοπές κτλ. Το τμήμα των προτιμήσεων παρέχει επίσης τη δυνατότητα να καθοριστούν προτιμήσεις για ξεχωριστές τουριστικές υπηρεσίες (ξενοδοχεία, πρακτορεία ενοικίασης αυτοκινήτων κ.α.). Η κατηγορία «OtherSrvcPref», επιτρέπει την επέκταση των υπηρεσιών σε περιοχές που δεν έχουν ακόμα καθοριστεί. Στο επόμενο σχήμα (Εικόνα 10) δίνεται

Το element του XML Schema του προφίλ χρήστη που αναφέρεται στις προτιμήσεις του χρήστη.



Εικόνα 10 Το element του XML Schema του προφίλ χρήστη που αναφέρεται στις προτιμήσεις του χρήστη

Το στοιχείο <PrefCollection> επιτρέπει την αναγνώριση κοινών προτιμήσεων καθώς και προτιμήσεων που συνδέονται με συγκεκριμένες τουριστικές υπηρεσίες. Τα προφίλ ΜΠΟΠΕΙ να έχουν μια ή περισσότερες συλλογές. Εάν ένα προφίλ περιλαμβάνει τουριστικές προτιμήσεις ΘΑ ΠΡΕΠΕΙ να έχει υπηρεσίες από μία ή περισσότερες από τις παρακάτω ομάδες στοιχείων: <CommonPref>, <AirlinesPref>, <VehicleRentalPref>, <HotelPref>, <OtherSrcvPref> (άλλες υπηρεσίες) ή <TPA_Extensions>.

Η χρήση των συλλογών προτιμήσεων είναι ένα χαρακτηριστικό που επιτρέπει να καθοριστούν τουριστικές υπηρεσίες για κάθε είδος τουρισμού. Για παράδειγμα, ένας ταξιδιώτης μπορεί να έχει απαιτήσεις για υπηρεσίες για μεγάλες υπερατλαντικές πτήσεις τις οποίες δεν θα χρειαζόταν μικρότερες πτήσεις εσωτερικού. Σ' αυτό το σενάριο, οι συλλογές θα μπορούσαν να καθοριστούν για διεθνείς και εσωτερικές

πτήσεις, ορίζοντας διαφορετικές αεροπορικές υπηρεσίες σε κάθε συλλογή. Ταξίδια σε διαφορετικούς προορισμούς μπορεί να απαιτούν τον καθορισμό διαφορετικών συλλογών.

Παρακάτω δίνεται μια περιγραφή των προτιμήσεων για τα ξενοδοχεία. Αυτές οι προτιμήσεις χρησιμοποιούνται στη συνέχεια της εργασίας για την αναζήτηση διαθεσιμότητας με χρήση του προφίλ πελάτη.

3.12.4.1 Προτιμήσεις Ξενοδοχείων

Στο τμήμα των προτιμήσεων ξενοδοχείων, οι πελάτες ή οι επιχειρήσεις μπορούν να επιλέξουν τις προτιμήσεις τους για διάφορες διευκολύνσεις των ξενοδοχείων και των δωματίων. Οι ταξιδιώτες μπορούν να δηλώσουν προτιμώμενες αλυσίδες ξενοδοχείων, τοποθεσία, τύπο ξενοδοχείου, κατηγορία ξενοδοχείου, τοποθεσία δωματίου και άλλα. Μπορούν να καθορίσουν τις προτιμήσεις τους για υπηρεσίες που παρέχονται από ξενοδοχεία, όπως ψυχαγωγικές και επιχειρηματικές υπηρεσίες, καθώς και διευκολύνσεις δωματίων όπως τύπο κρεβατιού. Μπορούν επίσης να καθορίσουν διαιτητικούς περιορισμούς και γαστρονομικές προτιμήσεις, καθώς και να καθορίσουν υπηρεσίες φαγητού που παρέχονται από τα ξενοδοχεία. Η προδιαγραφή επιτρέπει τον καθορισμό προτιμήσεων διασκέδασης, κατά τη διάρκεια της διαμονής τους στο ξενοδοχείο, καθώς και προσωπικές υπηρεσίες και άλλες ειδικές απαιτήσεις. Ακόμα μπορούν να καθορίσουν προτιμήσεις για την ασφάλεια και απαιτήσεις για τις ανάγκες των ατόμων με ειδικές ανάγκες.

Η ιδιότητα **PreferLevel**, που εμπεριέχεται σε όλα τα στοιχεία, δείχνει το επίπεδο προτίμησης που επιλέγεται για κάθε στοιχείο. Οι τιμές που μπορεί να πάρει είναι: **Only**, **Unacceptable** και **Preferred** με προεπιλεγμένη την τιμή **Preferred**.

- Η τιμή **Only** δηλώνει ότι μόνο αυτή η προτίμηση πρέπει να χρησιμοποιηθεί στη συγκεκριμένη περίπτωση
- Η τιμή **Unacceptable** δηλώνει ότι ΔΕΝ πρέπει να χρησιμοποιηθεί αυτή η προτίμηση στη συγκεκριμένη περίπτωση

Η ιδιότητα **RatePlanCode** δηλώνει το προτιμώμενο οικονομικό πακέτο για τη συγκεκριμένη περίπτωση.

Στον παρακάτω πίνακα περιγράφονται συνοπτικά οι ιδιότητες του στοιχείου **HotelPref**, καθώς και τα στοιχεία παιδιά του με τις ιδιότητες τους.

Όνομα στοιχείου	Περιεχόμενο	Ιδιότητες	Περιγραφή
HotelPref	Element	PreferLevel, RatePlanCode, SmokingInd, HotelGuestType	Προτιμήσεις ξενοδοχείων
LoyaltyPref	Element	PreferLevel	Προγράμματα αξιοπιστίας
PaymentFormPref	Element	PreferLevel, RPH	Τρόποι πληρωμής
HotelChainPref	Element	PreferLevel	Αλυσίδα ξενοδοχείων
PropertyNamePref	Element	PreferLevel	Όνομα ξενοδοχείου
PropertyLocationPref	Element	PreferLevel, PropertyLocationType	Τοποθεσία ξενοδοχείου
PropertyTypePref	Element	PreferLevel, PropertyType	Τύπος ξενοδοχείου
PropertyClassType	Element	PreferLevel, PropertyClassType	Κατηγορία ξενοδοχείου
PropertyAmenityPref	Element	PreferLevel, PropertyAmenityType	Διευκολύνσεις ξενοδοχείου
RoomAmenityPref	Element	PreferLevel, RoomAmenityType	Διευκολύνσεις δωματίων
RoomLocationPref	Element	PreferLevel, RoomLocationType	Τοποθεσία δωματίου
BedTypePref	Element	PreferLevel, BedType	Τύπος κρεβατιού
FoodSrcvPref	Element	PreferLevel, FoodSrcvType	Προτιμήσεις φαγητών και ποτών
MealPref	Element	PreferLevel, MealType, FavoriteFood, Beverage	Προτιμήσεις εστίασης
MediaEntertainPref	Element	PreferLevel	προτιμήσεις διασκέδασης
PetInfoPref	Element	PreferLevel	Πληροφορίες για κατοικίδια
RecreationSrcvPref	Element	PreferLevel, RecreationSrcvType	Υπηρεσίες αναψυχής
BusinessSrcvPref	Element	PreferLevel, BusinessSrcvType	Επιχειρηματικές υπηρεσίες
PersonalSrcvPref	Element	PreferLevel, PersonalSrcvType	Προσωπικές υπηρεσίες
SecurityFeaturePref	Element	PreferLevel	Ζητήματα ασφάλειας
PhysChallFeaturePref	Element	PreferLevel, PhysChallFeature	Διευκολύνσεις για άτομα με ειδικές ανάγκες
SpecRequestPref	Element	PreferLevel	Ειδικές απαιτήσεις
Extensions	Element		Το στοιχείο αυτό επιτρέπει την προσθήκη επιπλέον πληροφοριών σύμφωνα με τις ανάγκες των επιχειρήσεων

Πίνακας 1: Οι ιδιότητες του στοιχείου **HotelPref** του XML Schema του OTA για το προφίλ χρήστη, όπως δίνεται στο συνοδευτικό κείμενο των προδιαγραφών του OTA

3.13 Ανακεφαλαίωση

Στο κεφάλαιο αυτό έγινε μια περιγραφή των προδιαγραφών της ebXML και του OTA.

Έγινε μια συνοπτική περιγραφή των προδιαγραφών της ebXML. Παρουσιάστηκε μια γενική εικόνα της αρχιτεκτονικής στην οποία στηρίζεται και δόθηκε ένα ολοκληρωμένο παράδειγμα της χρήσης της. Δόθηκε ιδιαίτερα έμφαση στην ebMS και στο CPP/A μιας και αυτές χρησιμοποιούνται στις προδιαγραφές του OTA.

Έγινε περιγραφή των προδιαγραφών του OTA, κυρίως όσον αφορά την αρχιτεκτονική του, τις υπηρεσίες που ορίζει και τα μηνύματα που ανταλλάσσονται για συγκεκριμένες υπηρεσίες. Δεν έγινε αναλυτική περιγραφή του περιεχομένου όλων των μηνυμάτων που έχει αναπτύξει, καθώς ξεφεύγει από τα πλαίσια της εργασίας. Θα περιγραφούν αναλυτικά τα μηνύματα που σχετίζονται με την αναζήτηση ξενοδοχείων, διαθεσιμότητας και κρατήσεων ξενοδοχείων στο επόμενο κεφάλαιο, όπου περιγράφεται η αρχιτεκτονική του συστήματος ebXML OTA Server και της υλοποίησης όλων των OTA Web Services που υποστηρίζονται.

Κεφάλαιο 4

Επισκόπηση της αρχιτεκτονικής του ebXML OTA Server

4.1 Εισαγωγή

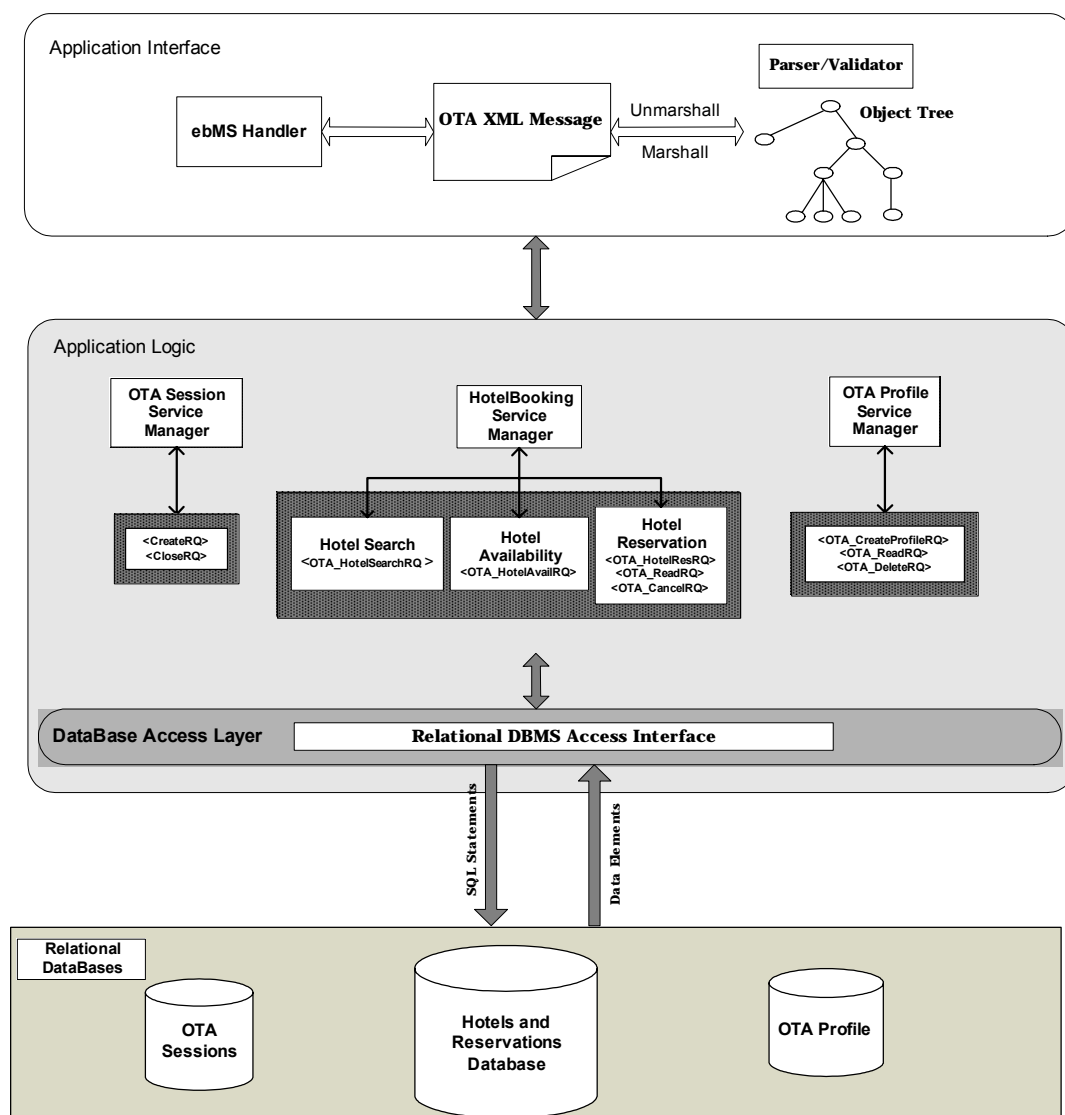
Στο κεφάλαιο αυτό περιγράφεται η αρχιτεκτονική του συστήματος που φιλοξενεί τις OTA Web Services, περιγράφονται τα επιμέρους τμήματα που απαρτίζουν τον ebXML OTA Server, όπως είναι η επικοινωνία με το διαδίκτυο, οι διαχειριστές υπηρεσιών και οι βάσεις δεδομένων. Περιγράφεται επίσης η υλοποίηση των OTA Web Services που υποστηρίζονται, απεικονίζονται τα σχετικά XML Schemata, καθώς και στιγμιότυπα χρήσης τους από XML έγγραφα.

4.2 Περιγραφή του ebXML OTA Server

Ο OTA Server αποτελείται από τρία τμήματα:

- Το τμήμα που αναλαμβάνει την επικοινωνία του συστήματος με το διαδίκτυο.
- Το τμήμα όπου εδράζονται οι βασικές λειτουργίες των υπηρεσιών.
- Το τμήμα των σχεσιακών βάσεων δεδομένων που διατηρούν όλες τις απαιτούμενες πληροφορίες για την λειτουργία του συστήματος.

Στο παρακάτω σχήμα (Εικόνα 11) δίνεται μια συνοπτική γραφική απεικόνιση της αρχιτεκτονικής του ebXML OTA Server, όπου απεικονίζονται τα επιμέρους τμήματα που απαρτίζουν το σύστημα:



Εικόνα 11 Η αρχιτεκτονική του ebXML OTA Server

Στη συνέχεια περιγράφονται τα τρία παραπάνω τμήματα.

4.2.1 Η επικοινωνία του συστήματος με το διαδίκτυο (Application Interface)

Το πρώτο τμήμα είναι υπεύθυνο για την λήψη και αποστολή των XML μηνυμάτων, καθώς και για την δρομολόγηση των μηνυμάτων στα κατάλληλα τμήματα που υλοποιούν τις υπηρεσίες για περαιτέρω επεξεργασία.. Ο ebXML OTA Server λαμβάνει τα εισερχόμενα ebXML μηνύματα μέσω της ebMS, ανακτά τα OTA XML μηνύματα που μεταφέρονται σαν συνημμένα στα ebXML μηνύματα, ελέγχει την τιμή

του πεδίου Service¹² και δρομολογεί τα OTA XML μηνύματα στα κατάλληλα τμήματα της εφαρμογής για περαιτέρω επεξεργασία. Τα τμήματα αυτά ονομάζονται Διαχειριστές Υπηρεσιών (Service Managers).

Για την υλοποίηση του συστήματος ώστε να υποστηρίζει την επιθυμητή λειτουργικότητα, ακολουθήθηκαν οι κατευθύνσεις που δίνονται στις προδιαγραφές του OTA που αναφέρονται στην αρχιτεκτονική. Πιο συγκεκριμένα, η υπηρεσία μηνυμάτων της ebXML (ebMS) επικοινωνεί με το τμήμα που υλοποιεί τις υπηρεσίες του OTA, μεταφέροντας τα XML μηνύματα των πελατών από και προς τον ebXML OTA Server. Ο ebXML OTA Server είναι Web εφαρμογή και φιλοξενείται σε έναν Web Server (Apache Tomcat 4.03 [17]).

Η ebMS που χρησιμοποιήθηκε είναι μια υλοποίηση ανοιχτού λογισμικού την οποία παρέχει η εταιρία Sybase [31]. Χρησιμοποιεί ένα Servlet για να δέχεται και να στέλνει μηνύματα μέσω του πρωτοκόλλου HTTP. Υποστηρίζει αξιόπιστη μετάδοση μηνυμάτων (Reliable Messaging) και την ύπαρξη στιγμιότυπων CPA (στη δευτερεύουσα μνήμη). Παρέχει διεπαφές (interfaces) για την υλοποίηση τόσο σύγχρονης όσο και ασύγχρονης επεξεργασίας μηνυμάτων. Στη συγκεκριμένη εφαρμογή υλοποιήθηκε η σύγχρονη μετάδοση και επεξεργασία των μηνυμάτων.

4.2.2 Οι Διαχειριστές Υπηρεσιών (Application Logic)

Στο δεύτερο τμήμα του συστήματος, τα OTA XML μηνύματα που εισέρχονται στο σύστημα λαμβάνονται από του Διαχειριστές Υπηρεσιών (Service Managers), όπου γίνεται η κυρίως επεξεργασία των μηνυμάτων και εκτέλεση των υπηρεσιών. Οι Διαχειριστές Υπηρεσιών του συστήματος είναι οι εξής:

- OTA Session Service Manager: διαχειρίζεται τα μηνύματα που σχετίζονται με τη λειτουργία των OTA Sessions (δημιουργία ενός καινούργιου Session, τερματισμός ενός υπάρχοντος Session). Επίσης αναλαμβάνει να ελέγξει εάν τα εισερχόμενα μηνύματα (αυτά που δεν σχετίζονται με τη λειτουργία των OTA

¹² Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, το πεδίο Service αναφέρεται στην υπηρεσία που ζητείται. Στην παρούσα εργασία οι υπηρεσίες που υλοποιήθηκαν είναι οι: Profile (για την διαχείριση του προφίλ), HotelBooking (για αναζήτηση ξενοδοχείων, διαθεσιμότητας και διαχείριση κρατήσεων) και Session (για την διαχείριση των OTA Sessions).

Sessions) αναφέρονται σε κάποιο έγκυρο OTA Session. Τα Sessions που δημιουργούνται αποθηκεύονται στη βάση δεδομένων του συστήματος.

- Hotel Booking Service Manager: διαχειρίζεται τα μηνύματα που αναφέρονται στην αναζήτηση και κράτηση ξενοδοχείων. Ελέγχει την τιμή του πεδίου Action¹³, η οποία πρέπει να παίρνει μια από τις παρακάτω τιμές:
 - OTA_HotelSearchRQ για την εκτέλεση της αναζήτησης ξενοδοχείων
 - OTA_HotelAvailRQ για την αναζήτηση διαθεσιμότητας
 - OTA_HotelResRQ για την δημιουργία κράτησης
 - OTA_ReadRQ για την ανάκτηση μιας κράτησης από τη βάση δεδομένων
 - OTA_CancelRQ για την ακύρωση μιας υπάρχουσας κράτησης
- OTA Profile Service Manager: διαχειρίζεται τα μηνύματα που αφορούν τις λειτουργίες του προφίλ (δημιουργία, ανάγνωση, ανανέωση και διαγραφή). Ελέγχει την τιμή του πεδίου Action, η οποία πρέπει να παίρνει μια από τις παρακάτω τιμές:
 - OTA_CreateProfileRQ για τη δημιουργία ενός καινούργιου προφίλ και την αποθήκευση του στη βάση δεδομένων
 - OTA_ReadRQ για την ανάκτηση ενός προφίλ από τη βάση δεδομένων
 - OTA_UpdateRQ για την ανανέωση ενός προφίλ στη βάση δεδομένων
 - OTA_DeleteRQ για την διαγραφή ενός προφίλ από τη βάση δεδομένων

Τα XML έγγραφα που περιέχονται στα ebXML μηνύματα και αφορούν στις υπηρεσίες του ebXML OTA Server, μετατρέπονται σε ιεραρχίες από αντικείμενα Java. Τα αντικείμενα αυτά είναι στιγμιότυπα της ιεραρχίας Java κλάσεων που κατασκευάζεται με την μέθοδο Data Binding από τα XML Schemata του OTA. Η μέθοδος αυτή έχει ως στόχο την εύκολη διαχείριση των XML εγγράφων χρησιμοποιώντας μια αντικειμενοστραφή γλώσσα προγραμματισμού. Το αποτέλεσμα της εφαρμογής αυτής της μεθόδου είναι μια ιεραρχία από κλάσεις που αντιστοιχούν στα στοιχεία (Elements) και στους σύνθετους τύπου (ComplexTypes) που ορίζονται σε ένα XML Schema. Η μέθοδος που εκτελεί τη δημιουργία αντικειμένων για αυτές

¹³ Το πεδίο Action δηλώνει την ενέργεια που θα εκτελεστεί στην συγκεκριμένη υπηρεσία (Service). Για παράδειγμα για την κατασκευή ενός καινούργιου προφίλ, τα πεδία αυτά θα πρέπει να έχουν τις τιμές «Profile» (για το πεδίο Service) και «OTA_CreateProfileRQ» (για το πεδίο Action). Τα δύο αυτά πεδία (Service και Action) παρέχονται από τις προδιαγραφές της ebXML σαν προαιρετικά πεδία που μπορεί να υπάρχουν σε ένα ebXML μήνυμα, αλλά η ύπαρξή τους σε κάθε ebXML μήνυμα απαιτείται από τις προδιαγραφές του OTA.

τις κλάσεις, παίρνοντας σαν είσοδο ένα αρχείο ή ένα String που περιέχει το XML έγγραφο, ονομάζεται Unmarshal. Η αντίστροφη μέθοδος όπου γίνεται η μετατροπή ενός XML εγγράφου που βρίσκεται σε μορφή ιεραρχίας αντικειμένων, σε ένα String ή η αποθήκευσή του σε αρχείο ονομάζεται Marshal. Το πρόγραμμα που χρησιμοποιήθηκε για την μετατροπή αυτή είναι το Breeze XML Binder [4].

Οι Διαχειριστές Υπηρεσιών διατρέχουν τα αντικείμενα που παράγονται από τη διαδικασία Data Binding για να ανακτήσουν την πληροφορία που εμπεριέχεται στα XML έγγραφα. Η πληροφορία αυτή χρησιμοποιείται στη συνέχεια για την εκτέλεση των λειτουργιών των υπηρεσιών.

4.2.3 Η σύνδεση με τις Βάσεις Δεδομένων

Το τρίτο τμήμα του συστήματος είναι οι βάσεις δεδομένων, οι οποίες αποθηκεύουν όλες τις πληροφορίες που απαιτούνται για την εκτέλεση των OTA Web Services. Η επικοινωνία των βάσεων δεδομένων με το κυρίως σύστημα επιτυγχάνεται με τη χρήση της κατάλληλης διεπαφής πρόσβασης σχεσιακών βάσεων δεδομένων (JDBC). Η αλληλεπίδραση με τη βάση δεδομένων γίνεται με την εκτέλεση αποθηκευμένων στη βάση δεδομένων λειτουργιών (stored procedures) και την ανάκτηση των αποτελεσμάτων από τη βάση δεδομένων μέσω της διεπαφής. Οι βάσεις δεδομένων του ebXML OTA Server περιγράφονται αναλυτικότερα στην επόμενη ενότητα.

4.3 Οι Βάσεις Δεδομένων του συστήματος

Η υλοποίηση των υπηρεσιών αναζήτησης ξενοδοχείων και κράτησης δωματίων, καθώς και της διαχείρισης του προφίλ απαιτεί την ύπαρξη βάσεων δεδομένων προκειμένου να προκύψουν αποτελέσματα και να μπορεί να ελεγχθεί η σωστή λειτουργία των υπηρεσιών. Για την αναζήτηση ξενοδοχείων χρησιμοποιήθηκε ένα υπάρχον σύστημα που περιλαμβάνει βάση δεδομένων για τα ξενοδοχεία όλης της Κρήτης. Για την υλοποίηση όμως των υπηρεσιών αναζήτησης διαθεσιμότητας και κράτησης δωματίων, καθώς και της διαχείρισης του προφίλ χρήστη κατασκευάστηκαν εξ αρχής βάσεις δεδομένων, οι οποίες στηρίζονται στις προδιαγραφές του OTA. Αυτές οι βάσεις δεδομένων, επειδή ακριβώς στηρίζονται στις προδιαγραφές του OTA, δεν περιορίζονται για χρήση μόνο στην συγκεκριμένη

εφαρμογή αλλά μπορούν να εφαρμοστούν γενικότερα. Αποτελούν δηλαδή ένα γενικότερο πλαίσιο για την προσαρμογή υπάρχοντων συστημάτων στις προδιαγραφές διαλειτουργικότητας του OTA, καθώς και την επέκταση τέτοιων συστημάτων με την δυνατότητα διαχείρισης κρατήσεων και προφίλ χρηστών.

4.3.1 Η Βάση Δεδομένων για τις κρατήσεις δωματίων

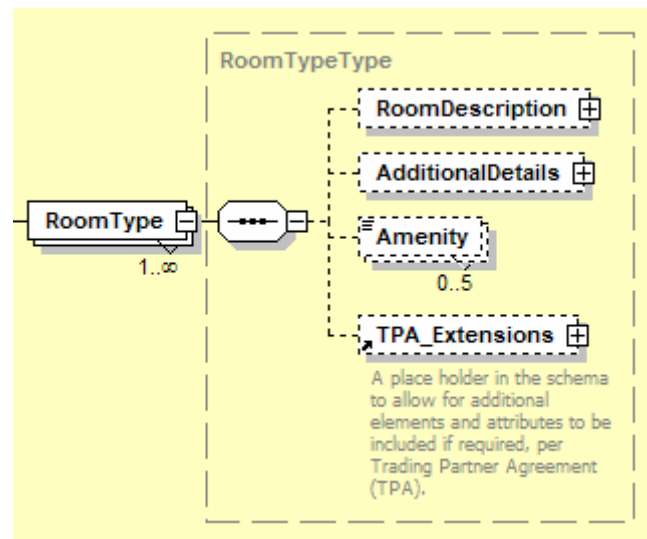
Για τον σχεδιασμό της Βάσης Δεδομένων που θα αποθηκεύει τις κρατήσεις δωματίων ξενοδοχείων χρησιμοποιήθηκαν οι προδιαγραφές του OTA που αναφέρονται στις κρατήσεις ξενοδοχείων, και πιο συγκεκριμένα το XML Schema «*OTA_HotelReservation.xsd*». Η πληθώρα των πληροφοριών που περιέχουν τα XML Schemata του OTA για τον τουριστικό τομέα τα καθιστούν ιδανική πηγή γνώσης, και δίνουν τη δυνατότητα για την κατασκευή λεπτομερέστατων βάσεων δεδομένων, παρόμοιων με αυτές που χρησιμοποιούν οι μεγαλύτερες τουριστικές επιχειρήσεις. Εξάλλου, από την πολύχρονη γνώση και εμπειρία αυτών των επιχειρήσεων προέκυψαν αυτές οι προδιαγραφές.

Το XML Schema «*OTA_HotelReservation.xsd*» από μόνο του δεν αρκεί για την κατασκευή της βάσης δεδομένων, καθώς ο στόχος του περιορίζεται στο να περιγράψει τα μηνύματα που μεταφέρουν τις πληροφορίες για μια κράτηση. Ο πυρήνας της βάσης δεδομένων σχεδιάστηκε από την αρχή με στόχο να ικανοποιεί τα παρακάτω κριτήρια:

- Να παρέχει τη δυνατότητα για την κράτηση περισσότερων από ένα δωματίων
- Να υπάρχει σύνδεση της κράτησης με κάποιο προφίλ χρήστη
- Να μπορούν να συσχετιστούν περισσότερες από μία κρατήσεις με κάποιο προφίλ χρήστη

Οι βασικές έννοιες στο σχεδιασμό της βάσης δεδομένων, που προκύπτουν από τη μελέτη του XML Schema «*OTA_HotelReservation.xsd*», πάνω στις οποίες στηρίζεται και ο σχεδιασμός της βάσης δεδομένων, είναι οι παρακάτω:

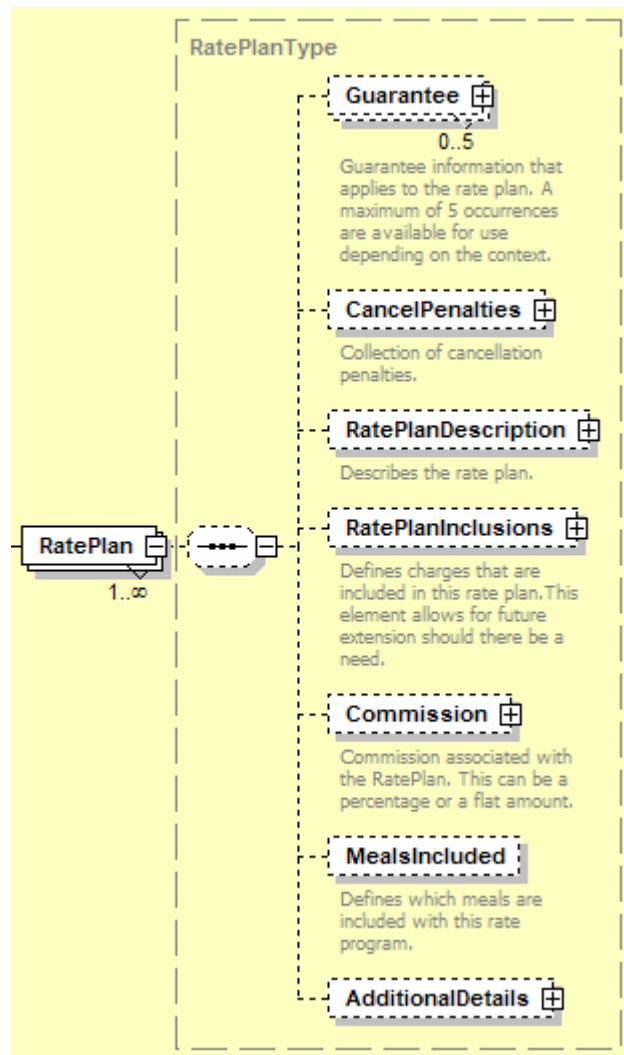
- **RoomType**: είναι ένα στοιχείο (Element) του XML Schema που παρέχει πληροφορίες για τον τύπο του δωματίου, τις διευκολύνσεις που παρέχει, σύντομη περιγραφή του, καθώς και επιπλέον λεπτομέρειες που ο ιδιοκτήτης του θέλει να παρέχει (Εικόνα 12).



Εικόνα 12 Το XML Schema του OTA για το RoomType

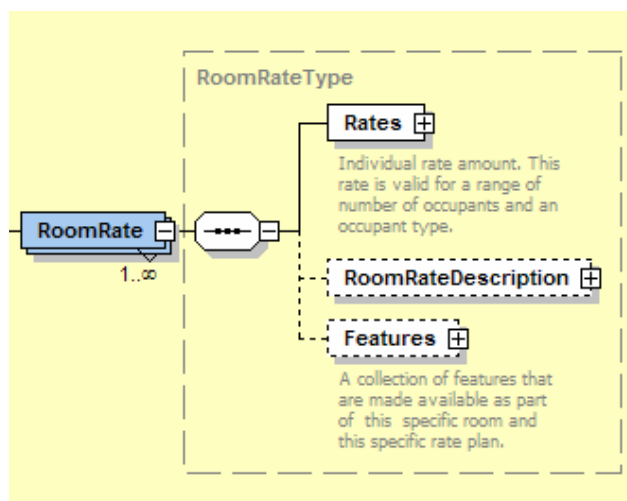
- ο **RatePlan**: είναι ένα στοιχείο (Element) του XML Schema που παρέχει πληροφορίες οικονομικού χαρακτήρα που μπορεί να συνοδεύουν μια κράτηση. Αποτελεί ουσιαστικά το οικονομικό πακέτο στο οποίο παρέχονται τα δωμάτια, και το οποίο μπορεί να περιλαμβάνει:
 - ο Την χρονική περίοδο για την οποία ισχύει το συγκεκριμένο οικονομικό πακέτο.
 - ο Τυχόν προκαταβολή που μπορεί να απαιτείται ως εγγύηση.
 - ο Τις ποινές που μπορεί να επιβάλλονται σε περίπτωση ακύρωσης μιας κράτησης.
 - ο Διάφορες χρεώσεις που μπορεί να περιλαμβάνονται στο οικονομικό πακέτο.
 - ο Τα γεύματα που προσφέρονται στα πλαίσια του συγκεκριμένου οικονομικού πακέτου.
 - ο Πρόσθετες πληροφορίες που αφορούν το συγκεκριμένο οικονομικό πακέτο.

Στο επόμενο σχήμα (Εικόνα 13) δίνεται ένα στιγμιότυπο από το XML Schema του RatePlan



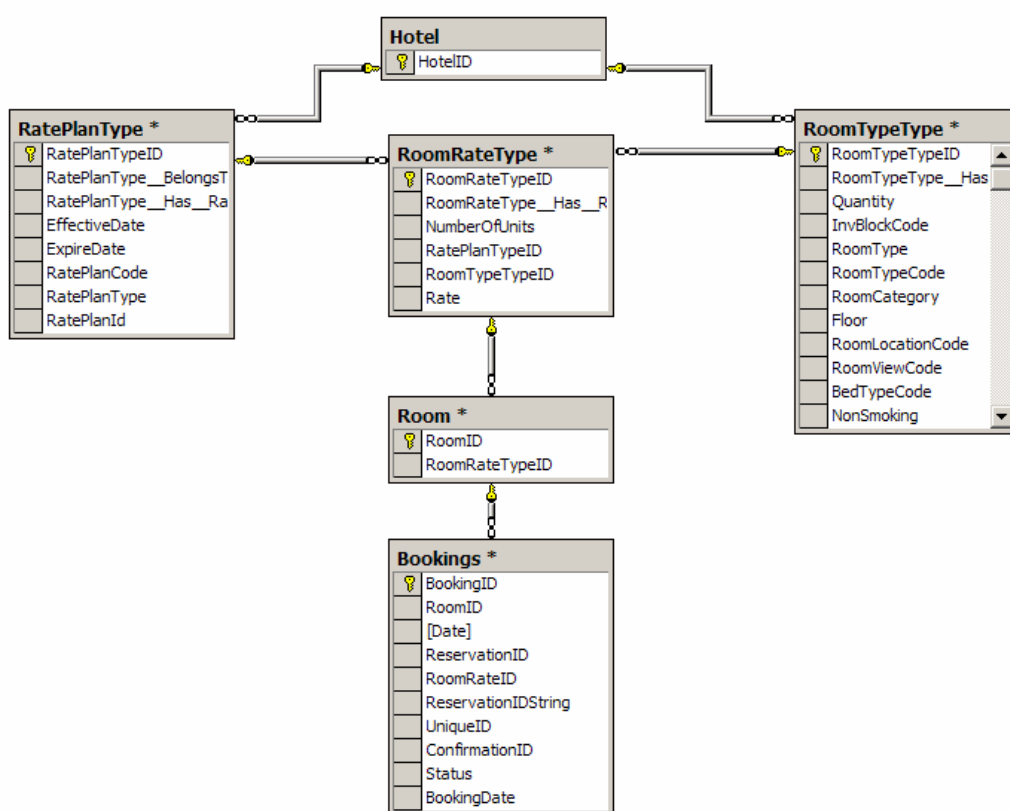
Εικόνα 13 Το XML Schema του OTA για το RatePlan

- **RoomRate**: αποτελεί τον συνδυασμό του RoomType και του RatePlan και χαρακτηρίζει την τιμή ενοικίασης ενός συγκεκριμένου RoomType για συγκεκριμένο RatePlan. Στο επόμενο σχήμα (Εικόνα 14) δίνεται ένα στιγμιότυπο από το XML Schema του RoomRate:



Εικόνα 14 Το XML Schema του OTA για το RoomRate

Με βάση τα παραπάνω κατασκευάστηκε η βάση δεδομένων, το σχήμα της οποίας φαίνεται στο παρακάτω σχήμα (Εικόνα 15). Σημειώνεται ότι στην εικόνα φαίνονται μόνο οι βασικοί πίνακες της βάσης δεδομένων, καθώς η γραφική αναπαράσταση ολόκληρης της βάσης δεν είναι εφικτή λόγω του μεγάλου αριθμού πινάκων¹⁴.



Εικόνα 15 Οι βασικοί πίνακες της βάσης δεδομένων για τις κρατήσεις ξενοδοχείων

¹⁴ Στο παράρτημα Β δίνεται το σύνολο των πινάκων της σχεσιακής βάσης δεδομένων σε μορφή πίνακα, όπου αναγράφονται βασικές πληροφορίες για κάθε πίνακα.

Ο πίνακας «Hotel» κρατά στο πεδίο «HotelID» τα IDs όλων των ξενοδοχείων που βρίσκονται στην βάση δεδομένων του πληροφοριακού συστήματος για τις τουριστικές επιχειρήσεις της Κρήτης. Η ύπαρξη αυτού του πίνακα είναι απαραίτητη γιατί ο αντίστοιχος πίνακας της βάσης δεδομένων υπάρχοντος πληροφοριακού συστήματος, ο οποίος ονομάζεται «SMEs» περιλαμβάνει όλες τις τουριστικές επιχειρήσεις της Κρήτης και όχι μόνο τα ξενοδοχεία.

Ο πίνακας «RatePlanType» περιέχει τα RatePlans που (μπορεί να) παρέχει κάθε ξενοδοχείο. Στον παρακάτω πίνακα περιγράφονται αναλυτικά τα πεδία του πίνακα «RatePlanType»:

Πίνακας RatePlanType		
Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
RatePlanTypeID	int	Το πρωτεύον κλειδί του πίνακα
RatePlanTypeBelongsToHotel	int	Κλειδί που συνδέει τον πίνακα «RatePlanType» με τον πίνακα «Hotel»
RatePlanTypeHasRateDescription	int	Κλειδί που δείχνει στον πίνακα λεκτικής περιγραφής του RatePlan
EffectiveDate	datetime	Ημερομηνία έναρξης του RatePlan
ExpireDate	datetime	Ημερομηνία λήξης του RatePlan
RatePlanCode	varchar	Χρησιμοποιείται για την αναζήτηση συγκεκριμένου κωδικού ή αναγνωριστικού (ID) κάποιου RatePlan, το οποίο είναι σε θέση να ζητήσει ο χρήστης.
RatePlanType	varchar	Μπορεί να πάρει συγκεκριμένες τιμές που χρησιμοποιούνται στην αναζήτηση καθορίζοντας την κατηγορία του RatePlan (π.χ. Corporate, Family, Rack Rate κτλ.).
RatePlanID	varchar	Καθορίζει ένα συγκεκριμένο αναγνωριστικό (ID) που σχετίζεται με το rate, και απαιτείται στην αίτηση για κράτηση για την απόκτηση του rate (π.χ. Corporate ID).

Ο πίνακας RoomTypeType περιέχει πληροφορίες σχετικά με τον τύπο των δωματίων. Στον παρακάτω πίνακα περιγράφονται αναλυτικά τα πεδία του πίνακα αυτού.

Πίνακας RoomTypeType		
Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
RoomTypeTypeID	int	Το πρωτεύον κλειδί του πίνακα.
HotelID	int	Κλειδί που δείχνει στο πίνακα των ξενοδοχείων.
RoomTypeType__Has__RoomDescription	int	Κλειδί που δείχνει στον πίνακα λεκτικής περιγραφής του RoomType.
Quantity	bigint	Το πλήθος των δωματίων αυτού του τύπου που παρέχει το ξενοδοχείο.
RoomType	varchar	Κωδικός που δείχνει τον τύπο του δωματίου (π.χ. double, king, κτλ.).
RoomTypeCode	varchar	Κωδικός δωματίου του συγκεκριμένου συστήματος που χρησιμοποιείται (π.χ. A1K, A1Q).
RoomCategory	varchar	Δείχνει την κατηγορία του δωματίου (π.χ. Moderate, Standard, Superior ή Deluxe).
Floor	Bigint	Δείχνει τον όροφο στον οποίο βρίσκεται το δωμάτιο.
RoomLocationCode	varchar	Δείχνει την τοποθέτηση του δωματίου σε σχέση με το ξενοδοχείο (π.χ. «κοντά στην έξοδο», «κοντά σε ανεγκυστήρα», κτλ.).
RoomViewCode	varchar	Δείχνει την θέα που έχει το δωμάτιο (π.χ. «θέα στον ωκεανό», «θέα στην πισίνα», «θέα στον κήπο» κτλ.).
BedTypeCode	varchar	Δείχνει τον τύπο των κρεβατιών που βρίσκονται στο δωμάτιο (π.χ. Double, Twin, Queen, ή King).
NonSmoking	bit	Δηλώνει την απαγόρευση ή μη του καπνίσματος.

Ο πίνακας RoomRateType περιέχει πληροφορίες σχετικά με τα RoomRates που παρέχει κάθε ξενοδοχείο.

Πίνακας RoomRateType		
Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
RoomRateTypeID	int	Το πρωτεύον κλειδί του πίνακα.
RoomRateType__Has__RoomRateDescription	int	Κλειδί που δείχνει στον πίνακα λεκτικής περιγραφής του RoomRate.
NumberOfUnits	bigint	Το πλήθος των δωματίων που δίνονται στο συγκεκριμένο RoomRate
RatePlanTypeID	int	Κλειδί που δείχνει στον πίνακα RatePlan
RoomTypeTypeID	int	Κλειδί που δείχνει στον πίνακα

		RoomTypeType
Rate	float	Η τιμή του συγκεκριμένου RoomRate

Ο πίνακας Room περιέχει όλα τα δωμάτια που υπάρχουν, καθώς και το RoomRate με το οποίο σχετίζονται.

Πίνακας Room		
Ο πίνακας αυτός περιέχει όλα τα δωμάτια και το RoomRate στο οποίο παρέχονται		
Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
RoomID	int	Το πρωτεύον κλειδί του πίνακα.
RoomRateID	int	Κλειδί που δείχνει στον πίνακα RoomRate

Ο πίνακας Bookings κρατά όλες τις κρατήσεις που γίνονται στο σύστημα.

Πίνακας Bookings		
Ο πίνακας αυτός αποθηκεύει όλες τις κρατήσεις που γίνονται		
Όνομα πεδίου	Τύπος δεδομένων	Περιγραφή
BookingID	int	Το πρωτεύον κλειδί του πίνακα.
RoomID	int	Κλειδί που δείχνει στον πίνακα Room. Δηλώνει ποιο ακριβώς δωμάτιο κρατήθηκε.
Date	datetime	Η ημερομηνία για την οποία γίνεται η κράτηση.
ReservationID	int	Ακέραιος αριθμός που δίνεται σε κάθε κράτηση. Χρησιμοποιείται εσωτερικά από το σύστημα για την ομαδοποίηση των κρατήσεων.
ReservationIDString	varchar	Αναγνωριστικό String που κατασκευάζεται μετά την επικύρωση της κράτησης, και χρησιμοποιείται από τον χρήστη για την αναγνώριση της κράτησης.
UniqueID	varchar	Αναγνωριστικό String που κατασκευάζεται κατά τη εκκίνηση της συνδιαλλαγής κράτησης και χρησιμοποιείται κατά τη διάρκεια της συνδιαλλαγής για την αναγνώριση της συγκεκριμένης κράτησης.
Status	varchar	Δείχνει την τρέχουσα κατάσταση της συνδιαλλαγής (π.χ. «σε εκκρεμότητα» και «επικυρώθηκε»)
BookingDate	datetime	Η ημερομηνία κατά την οποία εκτελέστηκε η κράτηση.

ProfileUniqueID	varchar	Αναγνωριστικό String που δηλώνει το UniqueID του προφίλ που συνδέεται με την κράτηση.
-----------------	---------	---

Οι παραπάνω πίνακες είναι οι βασικοί πίνακες της βάσης δεδομένων για τις κρατήσεις ξενοδοχείων. Το σύνολο των πινάκων της βάσης δεδομένων παρατίθεται στο Παράρτημα Γ της εργασίας.

4.3.2 Η Βάση Δεδομένων για το Προφίλ Χρήστη

Η Βάση Δεδομένων για το Προφίλ Χρήστη κατασκευάστηκε σύμφωνα με το αντίστοιχο XML Schema του OTA («OTA_Profile.xsd»). Για την κατασκευή της χρησιμοποιήθηκε η εφαρμογή XML→DB+Middleware. Αποτελείται από μεγάλο αριθμό πινάκων¹⁵ και τα αντίστοιχα stored procedures για την εκτέλεση των λειτουργιών της εισαγωγής, ανάκτησης και διαγραφής ενός προφίλ από τη βάση δεδομένων. Δεν θεωρούμε σκόπιμη την εξαντλητική περιγραφή του σχήματος της βάσης δεδομένων για το προφίλ του χρήστη σε αυτό το σημείο του κειμένου καθώς το σχήμα της βάσης αντιστοιχεί πλήρως στο XML Schema του OTA («OTA_Profile.xsd») και ο ενδιαφερόμενος αναγνώστης μπορεί να το βρει στο Παράρτημα Β.

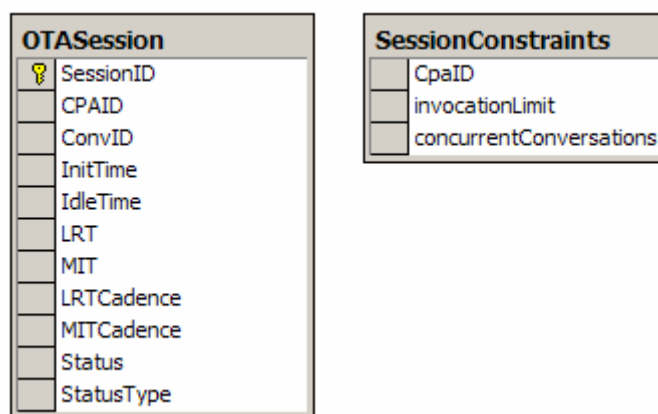
4.3.3 Η Βάση Δεδομένων για τα OTA Sessions

Οι πληροφορίες που σχετίζονται με τα OTA Sessions και με τα CPA που χρησιμοποιούνται ανά πάσα στιγμή από τον ebXML OTA Server αποθηκεύονται σε μια βάση δεδομένων. Στην βάση αυτή περιέχονται επίσης πληροφορίες όπως:

- Λειτουργικά χαρακτηριστικά του εκάστοτε Session που προκύπτουν από πληροφορίες που αναγράφονται στο CPA
- Τα IDs όλων των μηνυμάτων που καταφτάνουν στο σύστημα

Οι πίνακες της βάσης δεδομένων για τα OTA Sessions δίνονται στο παρακάτω σχήμα (Εικόνα 16):

¹⁵ Στο παράρτημα Β δίνεται το σύνολο των πινάκων της σχεσιακής βάσης δεδομένων του προφίλ σε μορφή πίνακα, όπου αναγράφονται βασικές πληροφορίες για κάθε πίνακα.



Εικόνα 16 Οι πίνακες της βάσης δεδομένων για τα OTA Sessions

Τα πεδία των παραπάνω πινάκων περιγράφονται στην επόμενη ενότητα όπου γίνεται η περιγραφή της υλοποίησης της υπηρεσίας διαχείρισης των OTA Sessions.

Στη συνέχεια του κεφαλαίου θα περιγραφούν αναλυτικά οι υπηρεσίες¹⁶ που υλοποιήθηκαν στα πλαίσια της εργασίας. Θα περιγραφούν τα XML Schemata τα οποία καθορίζουν την δομή των μηνυμάτων που ανταλλάσσονται σε κάθε περίπτωση, τα βήματα που ακολουθούνται στην υλοποίηση της κάθε υπηρεσίας και θα δοθούν παραδείγματα XML εγγράφων που περιέχουν τα μηνύματα αυτά.

Όπως αναφέρθηκε και στο κεφάλαιο 3, τα οι προδιαγραφές του OTA δεν απαιτούν την χρήση όλων των στοιχείων που περιγράφονται στα XML Schemata. Ο ρόλος των προδιαγραφών είναι να δώσουν μια βασική υποδομή (τα XML Schemata) πάνω στα οποία θα στηρίζονται τα μηνύματα που θα ανταλλάσσονται μεταξύ των τουριστικών επιχειρήσεων. Οι επιχειρήσεις αποφασίζουν ποια τμήματα των προδιαγραφών τους είναι απαραίτητα για να επικοινωνήσουν τις πληροφορίες που χρειάζονται. Επίσης τα XML Schemata που περιγράφουν τα μηνύματα περιλαμβάνουν το στοιχείο TPA_Extensions το οποίο μπορεί να χρησιμοποιηθεί για την ενσωμάτωση πληροφοριών που δεν μπορούν να αντιστοιχηθούν στα υπάρχοντα στοιχεία των XML σχημάτων. Με βάση αυτή την παρατήρηση, οι περιγραφές των XML σχημάτων που θα ακολουθήσουν θα επικεντρώνονται στα στοιχεία εκείνα που χρησιμοποιήθηκαν στην υλοποίηση της παρούσας εργασίας.

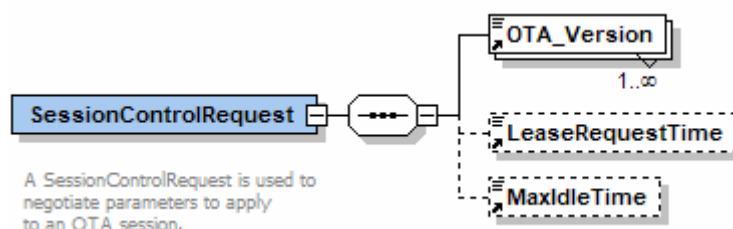
¹⁶ Για την υλοποίηση των υπηρεσιών χρησιμοποιήθηκε η έκδοση 2002B των προδιαγραφών του OTA.

4.4 Υλοποίηση της Υπηρεσίας διαχείρισης των OTA Sessions

Το πρώτο βήμα για την χρησιμοποίηση των υπηρεσιών του ebXML OTA Server είναι η δημιουργία ενός OTA Session. Το OTA Session χρησιμοποιείται στη συνέχεια για την εκτέλεση όλων των υπόλοιπων υπηρεσιών του συστήματος. Ο χρόνος λειτουργίας του OTA Session καθορίζεται από τον χρήστη, και κατά τη διάρκεια του μπορούν να εκτελεστούν οι υπόλοιπες OTA Web Services.

4.4.1 Περιγραφή της διαδικασίας δημιουργίας OTA Session

Το XML έγγραφο που αποστέλλεται για την δημιουργία ενός OTA Session υπακούει στο XML Schema που δίνεται στο επόμενο σχήμα (Εικόνα 17).



Εικόνα 17 Το XML Schema του OTA που παρέχει πληροφορίες για την δημιουργία ενός OTA Session

Το στοιχείο SessionControlRequest που είναι το αρχικό στοιχείο του XML Schema περιέχει τα παρακάτω στοιχεία:

- Το στοιχείο LeaseRequestTime καθορίζει τον μέγιστο χρόνο λειτουργίας του OTA Session.
- Το στοιχείο MaxIdleTime καθορίζει τον χρόνο που μπορεί να παραμείνει αχρησιμοποίητο το OTA Session.

Και τα δύο παραπάνω στοιχεία περιλαμβάνουν την ιδιότητα cadence που δηλώνει τον τύπο της χρονικής μονάδας, δηλαδή εάν πρόκειται για δευτερόλεπτα, λεπτά, ώρες ή μέρες.

Η κατασκευή ενός OTA Session περιλαμβάνει την εισαγωγή ορισμένων πληροφοριών στην αντίστοιχη βάση δεδομένων. Οι πληροφορίες αυτές είναι οι παρακάτω:

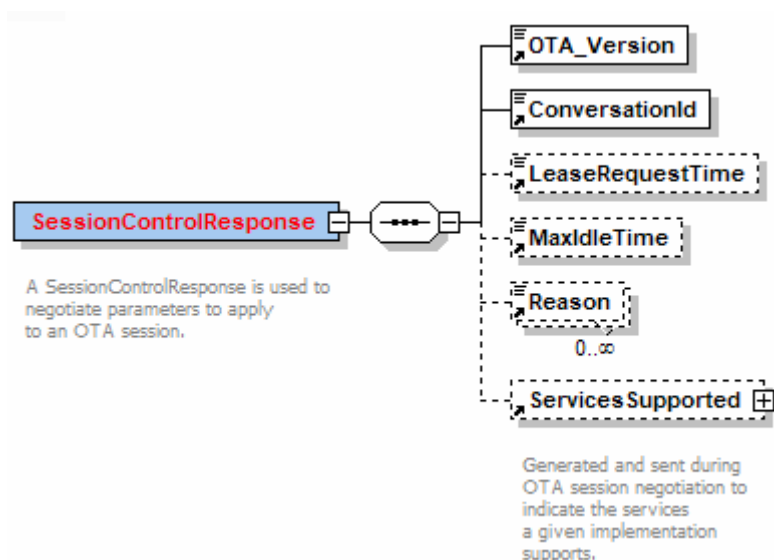
- CPAID – το ID του CPA που χρησιμοποιείται
- ConvID – το ConversationID που δημιουργείται για να διακρίνεται η συγκεκριμένη «συνομιλία».
- InitTime – ο χρόνος έναρξης του OTA Session
- IdleTime – το χρονικό διάστημα για το οποίο παραμένει αδρανές το OTA Session
- LRT – η τιμή του στοιχείου LeaseRequestTime που καθορίζει τον μέγιστο χρόνο λειτουργίας του OTA Session.
- MIT – η τιμή του στοιχείου MaxIdleTime που καθορίζει τον χρόνο που μπορεί να παραμείνει αχρησιμοποίητο το OTA Session.
- LRTCadence και MITCadence – η τιμές της ιδιότητας cadence των στοιχείων LeaseRequestTime και MaxIdleTime αντίστοιχα

Εισάγονται επίσης πληροφορίες σχετικά με τους περιορισμούς του CPA:

- invocationLimit – το πλήθος των κλήσεων στο CPA. Πρακτικά είναι το πλήθος των OTA Sessions που μπορούν να δημιουργηθούν με το CPA.
- concurrentConversations – το πλήθος των «συνομιλιών» που επιτρέπεται να «τρέχουν» ταυτόχρονα σε ένα OTA Session.

Οι πληροφορίες που αναφέρθηκαν παραπάνω αναπαρίστανται γραφικά στο σχήμα της βάσης δεδομένων που δίνεται στην Εικόνα 16.

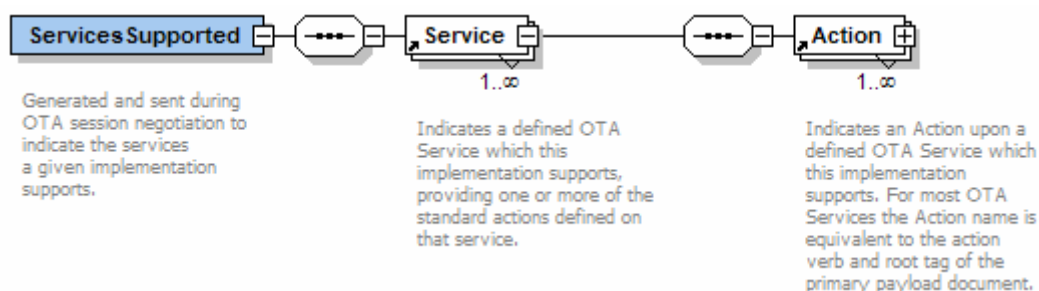
Η απόκριση στην αίτηση δημιουργίας του νέου OTA Session, υπακούει στο επόμενο σχήμα (Εικόνα 18).



Εικόνα 18 Το XML Schema που περιγράφει την απόκριση στην αίτηση για δημιουργία ενός OTA Session

Το μήνυμα απόκρισης περιέχει τα παρακάτω στοιχεία:

- **ConversationId** – το αναγνωριστικό της συγκεκριμένης συνομιλίας, δηλαδή του συγκεκριμένου OTA Session. Η χρήση της τιμής αυτού του στοιχείου και του CPAID απαιτείται στη συνέχεια από όλα τα μηνύματα των υπόλοιπων υπηρεσιών.
- **ServicesSupported** – περιγράφει τις υπηρεσίες (Services) που υποστηρίζει το σύστημα, καθώς και τις ενέργειες (Actions) που μπορούν να οριστούν σε αυτές (δηλαδή τις υπηρεσίες «HotelBooking» και «Profile» και τα αντίστοιχα Actions που έχουν υλοποιηθεί). Το XML Schema αυτού του πεδίου δίνεται στο επόμενο σχήμα (Εικόνα 19).



Εικόνα 19 Το XML Schema του στοιχείου ServicesSupported

- **LeaseRequestTime** και **MaxIdleTime** – επιστρέφονται οι παράμετροι που τέθηκαν κατά την αίτηση δημιουργίας νέου OTA Session, για την επαλήθευση της καταχώρισης των σωστών ρυθμίσεων.

4.4.2 Περιγραφή της διαδικασίας τερματισμού ενός OTA Session

Η αίτηση για τερματισμό ενός OTA Session αποτελείται από ένα ebXML μήνυμα το οποίο δεν περιλαμβάνει κάποιο επισυναπτόμενο OTA XML έγγραφο. Το μήνυμα αυτό περιλαμβάνει στην επικεφαλίδα του τα πεδία «Service» και «Action» καθώς και το πεδίο «ConversationID». Τα δύο πρώτα πεδία παίρνουν τις τιμές «Session» και «CloseRQ», ενώ το πεδίο «ConversationID» παίρνει την τιμή του ConversationID που χρησιμοποιείται στην συγκεκριμένη «συνομιλία». Η τιμή του πεδίου «Service» είναι αυτή που καθοδηγεί τον ebXML OTA Server να καλέσει τον SessionManager (τον διαχειριστή της υπηρεσίας Session), ο οποίος από την τιμή του πεδίου «Action» αναγνωρίζει ότι ζητείται τερματισμός του OTA Session. Η ταυτοποίηση του Session που θα τερματιστεί επιτυγχάνεται μέσω του συνδυασμού των πεδίων «CPAID» και «ConversationID». Η διαδικασία τερματισμού ενός Session αποτελεί στην ουσία την διαγραφή από την βάση δεδομένων των OTA Sessions της κατάλληλης εγγραφής.

Η απόκριση στην αίτηση τερματισμού ενός OTA Session είναι ένα ebXML μήνυμα, στο οποίο τα πεδία «Service» και «Action» έχουν τις τιμές «Session» και «CloseRS» αντίστοιχα.

4.5 Υλοποίηση της Υπηρεσίας Αναζήτησης Ξενοδοχείων

Στην ενότητα αυτή περιγράφεται η υλοποίηση της υπηρεσίας αναζήτησης ξενοδοχείων. Περιγράφονται τα κριτήρια της αναζήτησης, τα στάδια εκτέλεσης της διαδικασίας και οι παράμετροι που καθορίζουν τον τρόπο χρήσης των κριτηρίων αυτών – πλήρης ή μερική ικανοποίηση των κριτηρίων. Τέλος δίνονται παραδείγματα XML εγγράφων για αναζήτηση ξενοδοχείων και επιστροφή αποτελεσμάτων.

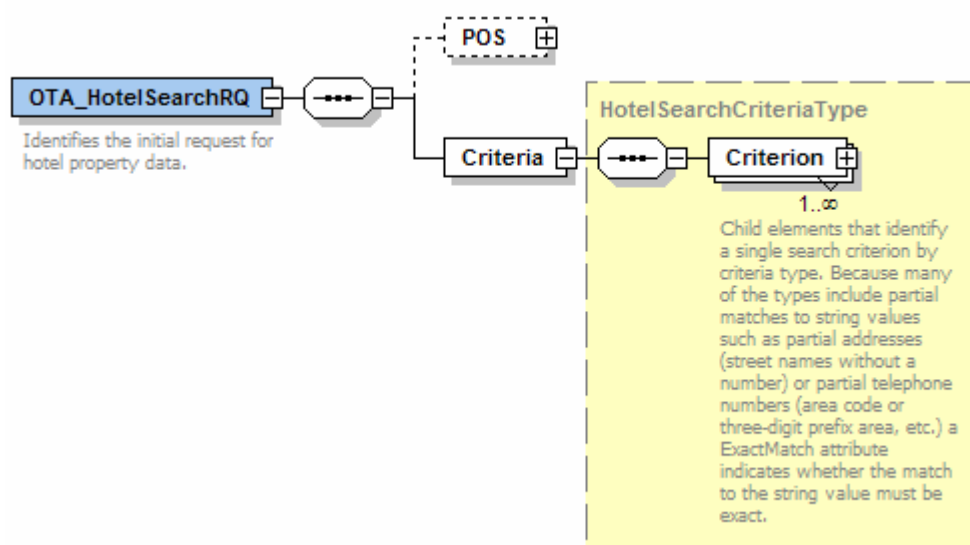
Για την υλοποίηση της υπηρεσίας αναζήτησης ξενοδοχείων, στο πληροφοριακό σύστημα που χρησιμοποιήθηκε, μέσω των μηνυμάτων που παρέχει το OTA είναι απαραίτητο να προηγηθεί μια αναλυτική περιγραφή των δύο μοντέλων στο επίπεδο των κριτηρίων και δυνατοτήτων αναζήτησης. Μπορούμε έτσι να εξετάσουμε πώς

μπορεί να γίνει η αντιστοίχιση των επιμέρους κριτηρίων του συστήματος που χρησιμοποιήθηκε στα αντίστοιχα πεδία που παρέχει το XML Schema του OTA.

Στην επόμενη ενότητα (4.5.1) περιγράφεται το XML Schema του OTA για τα μηνύματα της αναζήτησης ξενοδοχείων.

4.5.1 Περιγραφή του XML Schema του OTA για την αναζήτηση ξενοδοχείων

Το XML Schema του OTA που περιγράφει τα μηνύματα της αναζήτησης ξενοδοχείων δίνεται στην Εικόνα 20.



Εικόνα 20 Το XML Schema του OTA για την αναζήτηση ξενοδοχείων

Το αρχικό στοιχείο **OTA_HotelSearchRQ** περιέχει το στοιχείο **Criteria**, το οποίο μπορεί να έχει απεριόριστο πλήθος από στοιχεία **Criterion**. Το στοιχείο **Criterion** περιέχει τα κριτήρια της αναζήτησης ξενοδοχείων. Έτσι μπορούμε να εκτελέσουμε αναζήτηση ξενοδοχείων με απεριόριστο αριθμό κριτηρίων.

Τα κριτήρια που περιέχονται στο στοιχείο Criterion θα περιγραφούν στην ενότητα 4.5.3.

4.5.2 Τα κριτήρια αναζήτησης ξενοδοχείων που παρέχει το υπάρχον σύστημα

Το σύστημα που χρησιμοποιήθηκε παρέχει τη δυνατότητα αναζήτησης των μικρομεσαίων επιχειρήσεων τουρισμού της Κρήτης. Τα αποτελέσματα της αναζήτησης περιέχουν εκτενείς πληροφορίες για τις επιχειρήσεις, μεταξύ των οποίων είναι αριθμοί τηλεφώνων, fax, email για την επικοινωνία του χρήστη με κάποιον υπεύθυνο της επιχείρησης, καθώς και url για το site της επιχείρησης (όταν είναι διαθέσιμο) όπου ο χρήστης μπορεί να πάρει περισσότερες πληροφορίες για την επιχείρηση καθώς και να προχωρήσει σε μια ηλεκτρονική συνδιαλλαγή (αν η επιχείρηση υποστηρίζει), όπως είναι π.χ. η κράτηση δωματίου / -ων σε κάποιο ξενοδοχείο.

Αναλυτικότερα, το σύστημα παρέχει τα παρακάτω κριτήρια αναζήτησης:

Γενικά Χαρακτηριστικά Επιχείρησης <ul style="list-style-type: none"> • Κατηγορία (lux, A, B, Γ, Δ, Ε • Έτος κατασκευής (από, έως) • Κοντινότερη πόλη / χωριό • Απόσταση από την πόλη / χωριό (m) • κτλ. 	Περιγραφή Παραλίας <ul style="list-style-type: none"> • Απόσταση από την παραλία (m) • Λοιπά στοιχεία περιγραφής της παραλίας
Γενικές Διευκολύνσεις <ul style="list-style-type: none"> • Τηλεόραση • Θέρμανση • κτλ. 	Διευκολύνσεις Δωματίων <ul style="list-style-type: none"> • Κλιματισμός • Τηλέφωνο • κτλ.
Υπηρεσίες Προσφοράς Φαγητού <ul style="list-style-type: none"> • Αριθμός εστιατορίων • Υπηρεσία πρωινού • κτλ. 	Κουζίνα <ul style="list-style-type: none"> • Παραδοσιακή κρητική κουζίνα • Κουζίνα διαίτης • κτλ.
Συνεδριακές Διευκολύνσεις <ul style="list-style-type: none"> • Αριθμός αιθουσών • Overhead projector • κτλ. 	Αθλητικές Διευκολύνσεις <ul style="list-style-type: none"> • Mini Golf • Γήπεδο Tennis • κτλ.
Διευκολύνσεις Παραλίας <ul style="list-style-type: none"> • Μπαρ παραλίας • Καρέκλες παραλίας • Ναυαγосώστης • κτλ. 	Πισίνες <ul style="list-style-type: none"> • Αριθμός από πισίνες • Θερμαινόμενη πισίνα • κτλ.
Διευκολύνσεις για ΑΜΕΑ <ul style="list-style-type: none"> • Ειδικές ράμπες για τροχήλατα • Ειδικά διαμορφωμένα δωμάτια • κτλ. 	Ομιλούμενες Γλώσσες <ul style="list-style-type: none"> • Ελληνικά • Αγγλικά • κτλ.

Τρόποι Πληρωμής <ul style="list-style-type: none"> • Diners • Visa • Κτλ 	Γεωγραφική περιοχή <ul style="list-style-type: none"> • Όλη την Κρήτη • Σε έναν Δήμο ενός Νομού
--	--

Πίνακας 2: Κριτήρια αναζήτησης ξενοδοχείων στο υπάρχον σύστημα πληροφοριακό σύστημα τουριστικών επιχειρήσεων της Κρήτης

Η αναζήτηση παρουσιάζει τα εξής χαρακτηριστικά (τα οποία αφορούν στη λειτουργικότητα όπως αυτή προσφέρεται μέσα από την ιστοσελίδα του συστήματος και όχι στις δυνατότητες του μοντέλου)

- Τα αποτελέσματα της αναζήτησης πρέπει να ικανοποιούν είτε όλα τα κριτήρια (λογικό AND) ή τουλάχιστον ένα από τα κριτήρια (λογικό OR).
- Δεν παρέχεται η δυνατότητα πιο σύνθετων ερωτημάτων.
- Δεν παρέχεται η δυνατότητα για ταξινόμηση των αποτελεσμάτων ως προς τον βαθμό στον οποίο πληρούν τα κριτήρια αναζήτησης (rating).

4.5.3 Τα κριτήρια αναζήτησης ξενοδοχείων που παρέχουν οι προδιαγραφές του OTA (OTA_HotelSearchRQ/RS)

Η αναζήτηση ξενοδοχείων, σύμφωνα με τις προδιαγραφές του OTA, επιτυγχάνεται με το ζεύγος μηνυμάτων OTA_HotelSearchRQ / OTA_HotelSearchRS. Το μήνυμα OTA_HotelSearchRQ περιλαμβάνει τα κριτήρια με τα οποία θα γίνει η αναζήτηση, ενώ το μήνυμα OTA_HotelSearchRS επιστρέφει τα αποτελέσματα της αναζήτησης.

Αναλυτικότερα, το μήνυμα HotelSearchRQ περιλαμβάνει το element “Criteria” το οποίο μπορεί να περιέχει ένα ή περισσότερα elements “Criterion”. Το “Criterion” περιέχει τα κριτήρια με τα οποία γίνεται η αναζήτηση και είναι τα εξής:

Γεωγραφική θέση (Position) <ul style="list-style-type: none"> • Πλάτος (Latitude) • Μήκος (Longitude) • Ύψος (Altitude) 	Διεύθυνση (Address) <ul style="list-style-type: none"> • Οδός (AddressLine) • Πόλη (CityName) • Ταχυδρομικός Κώδικας (PostalCode) • Νομός (StateProv) • Χώρα (CountryName)
Τηλέφωνο (Telephone) <ul style="list-style-type: none"> • Κωδικός χώρας (CountryAccessCode) • Κωδικός περιοχής (AreaCityCode) • Αριθμός τηλεφώνου (PhoneNumber) 	Σημείο αναφοράς (RefPoint) Το στοιχείο Reference Point επιτρέπει την αναζήτηση ως προς την απόσταση από κάποιο σημείο αναφοράς.

Κωδικός αναφοράς (CodeRef) <ul style="list-style-type: none"> Κωδικός περιοχής (LocationCode) 	Κωδικός ξενοδοχείου (HotelRef) <ul style="list-style-type: none"> Κωδικός αλυσίδας (ChainCode) Όνομα ξενοδοχείου (HotelName)
Ακτίνα από το σημείο αναφοράς (Radius) <ul style="list-style-type: none"> Απόσταση (Distance) Μονάδα μέτρησης (DistanceMeasure, π.χ. χλμ) Κατεύθυνση (Direction) 	Διευκολύνσεις ξενοδοχείου (HotelAmenity) Διευκολύνσεις που παρέχει το ξενοδοχείο π.χ. πισίνα

Πίνακας 3: Κριτήρια αναζήτησης ξενοδοχείων στο μοντέλο του OTA

Το element “Criterion” έχει δύο optional attributes με τα οποία καθορίζονται ειδικά χαρακτηριστικά της αναζήτησης.

- ExactMatch (Boolean) με το οποίο καθορίζουμε αν θέλουμε να επιστραφούν τα ξενοδοχεία με πλήρη ή μερική ταύτιση του κάθε κριτηρίου που θέσαμε, π.χ. όνομα οδού χωρίς τον αριθμό.
- ImportanceType (αποδεκτές τιμές είναι: “Mandatory”, “High”, “Medium”, “Low”) με το οποίο καθορίζουμε πόσο σημαντικό είναι το κάθε κριτήριο για την αναζήτηση, αποδίδοντας έτσι «βάρος» στο κριτήριο.

Με βάση τα δύο παραπάνω χαρακτηριστικά μπορούμε να ρυθμίσουμε τις λογικές συνθήκες που θα καθορίσουν την αναζήτηση.

4.5.4 Σύγκριση των δύο μοντέλων

Το υπάρχον σύστημα είναι ένα Destination Management System για την Κρήτη. Από την άλλη πλευρά, οι προδιαγραφές που προτείνει το OTA είναι γενικές και έχουν στόχο να καλύψουν τις ανάγκες των συστημάτων τουρισμού για τα μηνύματα που θα πρέπει να ανταλλάσσουν μεταξύ τους, καλύπτοντας τα semantics του τομέα (αλλά και λεπτομέρειες), ώστε να μπορέσουν να αναπτυχθούν web services για τον τουριστικό τομέα. Μέσα σ’ αυτό το πλαίσιο το OTA καλύπτει πλήρως το υπάρχον σύστημα όσον αφορά την αντιστοίχιση συγκεκριμένων χαρακτηριστικών για τη δημιουργία κατάλληλων μηνυμάτων αναζήτησης ξενοδοχείων.

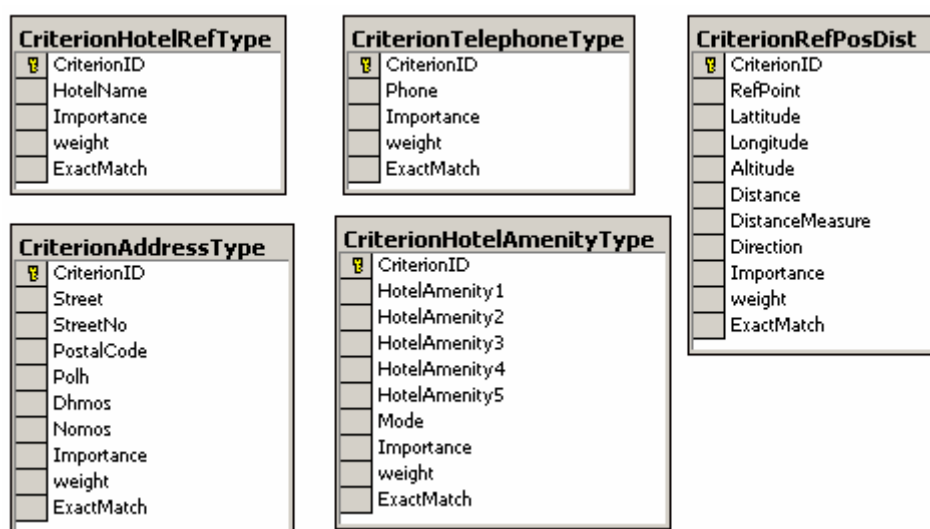
Με βάση το παραπάνω σκεπτικό μπορούμε να κάνουμε την αντιστοίχιση των κριτηρίων αναζήτησης ξενοδοχείων των δύο μοντέλων. Η αντιστοίχιση έχει ως σημείο αναφοράς τα κριτήρια του OTA και δίνεται στον επόμενο πίνακα.

Κριτήριο	OTA	Υπάρχον σύστημα
Position	✓	✓
Address	✓	✓
Telephone	✓	✓
RefPoint	✓	✓
CodeRef	✓	- *
HotelRef	✓	✓
Radius	✓	✓
HotelAmenity	✓	✓

*το CodeRef περιγράφει μια τοποθεσία πλησίον της οποίας ζητείται η εύρεση ξενοδοχείων, για παράδειγμα ένα αεροδρόμιο. Η τοποθεσία περιγράφεται μέσω συγκεκριμένου κωδικού. Το υπάρχον σύστημα δεν παρέχει τέτοιες πληροφορίες. Ωστόσο υπάρχει η δυνατότητα αναζήτησης ξενοδοχείων πλησίον κάποιου σημείου αναφοράς μέσω του πεδίου RefPoint, με το οποίο μπορεί να οριστεί μια τοποθεσία με το όνομα της.

4.5.5 Διαδικασία Αναζήτησης ξενοδοχείων

Η διαδικασία αναζήτησης ξενοδοχείων ενεργοποιείται με τη λήψη ενός έγκυρου OTA_HotelSearchRQ XML εγγράφου. Αρχικά ανακτάται η λίστα με τα κριτήρια της αναζήτησης (το στοιχείο Criterion) από το XML έγγραφο. Για κάθε αντικείμενο της λίστας ανακτούμε τα επιμέρους τα επιμέρους κριτήρια που το απαρτίζουν και τα αποθηκεύουμε στους αντίστοιχους πίνακες της βάσης δεδομένων. Οι πίνακες αυτοί που φιλοξενούν τις τιμές των κριτηρίων δίνονται σχηματικά στο παρακάτω σχήμα (Εικόνα 21)



Εικόνα 21 Οι Πίνακες Κριτηρίων Αναζήτησης ξενοδοχείων

Οι πίνακες αυτοί αποτελούνται από:

- Το πεδίο “*CriterionID*” που δείχνει την θέση του αντικείμενου στη λίστα των κριτηρίων και είναι το πρωτεύον κλειδί του πίνακα. Αυτό το πεδίο χρησιμεύει στην εύρεση των κριτηρίων που ανήκουν στο ίδιο αντικείμενο της λίστας και χρησιμοποιείται στην ταξινόμηση των αποτελεσμάτων ως προς το πλήθος των κριτηρίων που ικανοποιούνται.
- Τα πεδία του κάθε στοιχείου του κριτηρίου, όπως αυτά περιγράφονται στο αντίστοιχο XML Schema
- Τα πεδία *Importance* και *ExactMatch* που αντιστοιχούν στα αντίστοιχα πεδία *ImportanceType* και *ExactMatch* του στοιχείου *Criterion*.
- Το πεδίο *weight* που δηλώνει το βάρος το οποίο θα αποδοθεί στα αποτελέσματα που ικανοποιούν το συγκεκριμένο κριτήριο.

Στη συνέχεια εκτελείται το κύριο μέρος της αναζήτησης το οποίο αποτελείται από τα παρακάτω βήματα

1. Εάν το πλήθος των υποχρεωτικών κριτηρίων (*ImportanceType* = *Mandatory*) είναι θετικό, κατασκευάζεται δυναμικά ένα ερώτημα (SQL query) που περιέχει τα «Mandatory» κριτήρια. Το query εκτελείται για το σύνολο των ξενοδοχείων που υπάρχουν στη βάση δεδομένων και τα αποτελέσματα (εάν υπάρχουν) αποθηκεύονται σε έναν προσωρινό πίνακα.
2. Στη συνέχεια, εάν το πλήθος των μη υποχρεωτικών κριτηρίων (*ImportanceType* = *High*, *Medium* ή *Low*) είναι θετικό, τα αποτελέσματα που προέκυψαν στο βήμα 1 ελέγχονται ως προς το βαθμό στον οποίο ικανοποιούν τα μη υποχρεωτικά κριτήρια και βαθμολογούνται κατάλληλα σύμφωνα με συγκεκριμένα βάρη που θα περιγραφούν παρακάτω. Εάν στο βήμα 1 δεν προέκυψαν ξενοδοχεία που να ικανοποιούν τα υποχρεωτικά κριτήρια τότε όλα τα ξενοδοχεία φιλτράρονται ως προς τα μη υποχρεωτικά κριτήρια.

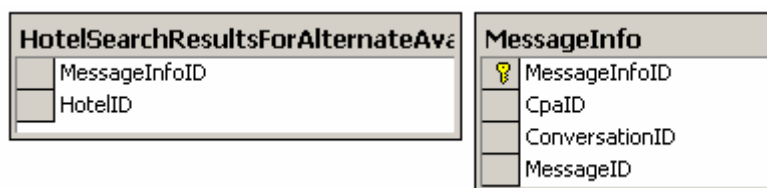
Αναλυτικότερα:

- Κατασκευάζεται ένας προσωρινός πίνακας temp1 όπου κρατούνται τα ξενοδοχεία που πρόκειται να φιλτραριστούν.
- Κατασκευάζεται ένας προσωρινός πίνακας temp2 στον οποίο αποθηκεύονται το ID του ξενοδοχείου που εξετάζεται, το ID του κριτηρίου με το γίνεται το φιλτράρισμα και το βάρος που εφαρμόζεται που

εξαρτάται από το κριτήριο (temp [HotelID int, CriterionID int, Weight float]).

- Εισάγονται στον πίνακα temp2 τα IDs των ξενοδοχείων με βάρος 1 (εάν πρόκειται για τα ξενοδοχεία που ικανοποιούν τα υποχρεωτικά κριτήρια) ή 0 (εάν πρόκειται για το σύνολο των ξενοδοχείων).
- Εισάγεται στον πίνακα temp2 τα (HotelID, CriterionID, Weight) που προκύπτουν από τη συνένωση του πίνακα των ξενοδοχείων με τον πίνακα του κάθε κριτηρίου.
- Τέλος τα δεδομένα του πίνακα temp2 ομαδοποιούνται με βάση το HotelID, υπολογίζονται το σύνολο των κριτηρίων που ικανοποιούνται από κάθε ξενοδοχείο και το συνολικό βάρος που αποδίδεται σε κάθε ξενοδοχείο και αποθηκεύονται σε έναν τελικό προσωρινό πίνακα «final» πριν καταλήξουν στον χρήστη.

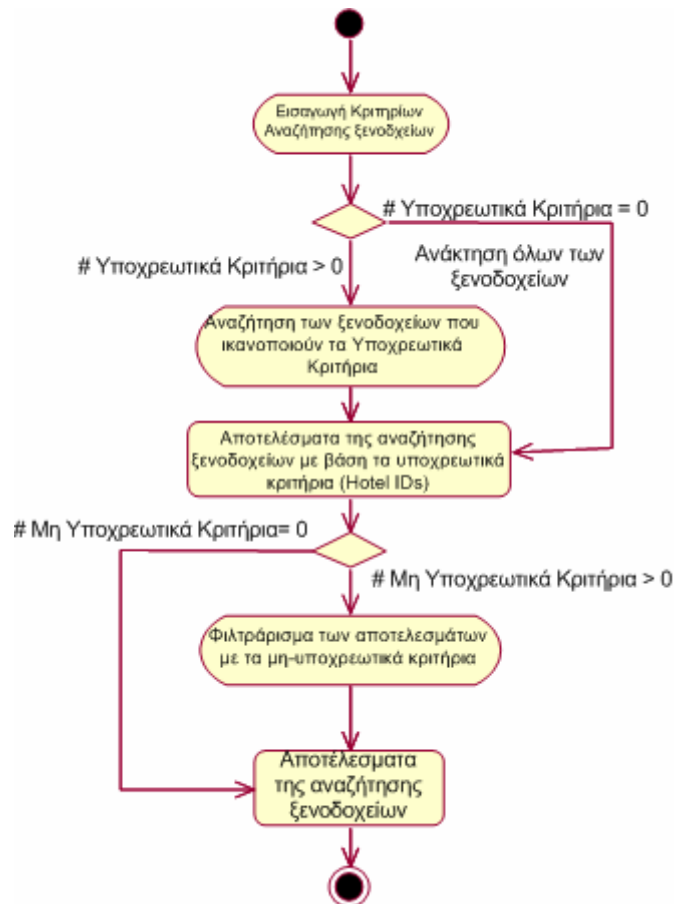
Τα αποτελέσματα της αναζήτησης αποθηκεύονται στον πίνακα “HotelSearchResultsForAlternateAvailability” για τυχόν μελλοντική χρήση τους στη διαδικασία αναζήτησης εναλλακτικής διαθεσιμότητας. Ο πίνακας αυτός περιέχει τα HotelIDs καθώς και το πεδίο MessageInfoID που αναφέρεται στον πίνακα “MessageInfo” ο οποίος περιέχει πληροφορίες για κάθε μήνυμα που έρχεται στο σύστημα, όπως είναι το CPAID, το ConversationID και το MessageID. Οι πίνακες αυτοί απεικονίζονται στο παρακάτω σχήμα (Εικόνα 22).



Εικόνα 22 Οι πίνακες «HotelSearchResultsForAlternateAvailability» και «MessageInfo»

3. Τέλος, τα αποτελέσματα ταξινομούνται ως προς το βαθμό στον οποίο ικανοποιούν τα κριτήρια και επιστρέφονται στο χρήστη.

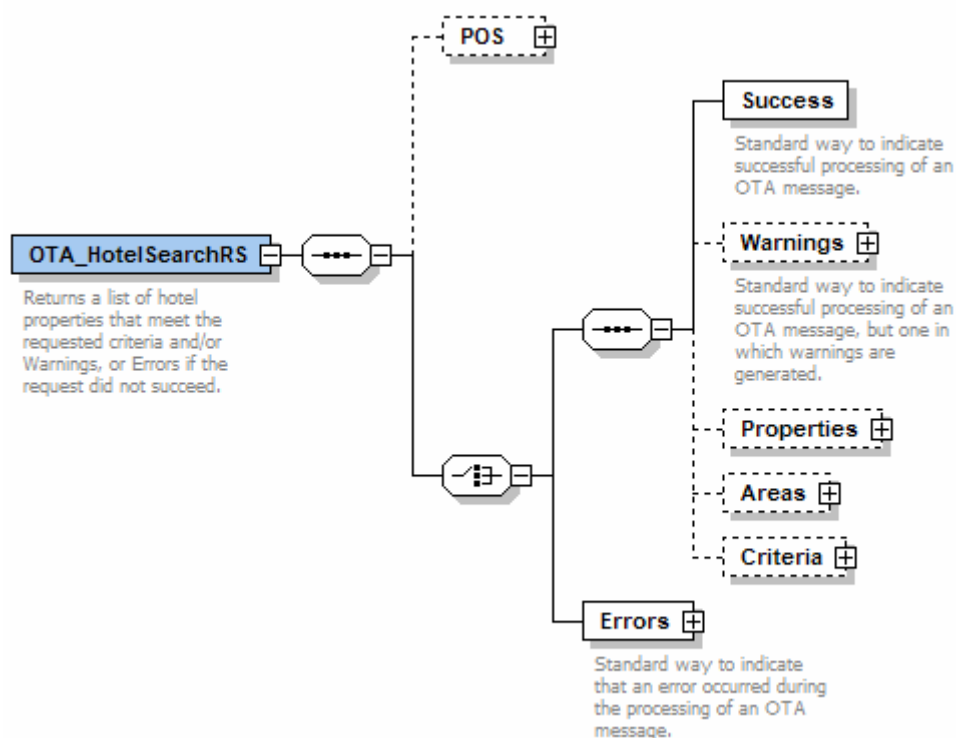
Το παρακάτω διάγραμμα δραστηριοτήτων (Εικόνα 23) καταδεικνύει τη ροή της διαδικασίας αναζήτησης ξενοδοχείων



Εικόνα 23 Το διάγραμμα δραστηριοτήτων της διαδικασίας αναζήτησης ξενοδοχείων

4.5.6 Περιγραφή του XML Schema του OTA για τα αποτελέσματα της αναζήτησης ξενοδοχείων

Το XML Schema του OTA που περιγράφει τα μηνύματα της επιστροφής των αποτελεσμάτων της αναζήτησης ξενοδοχείων δίνεται στο παρακάτω σχήμα (Εικόνα 24).



Εικόνα 24 Το XML Schema του OTA για την απόκριση στην αίτηση αναζήτησης ξενοδοχείων

Σύμφωνα με το παραπάνω σχήμα, τα XML έγγραφα που επιστρέφονται σαν αποτέλεσμα της αναζήτησης ξενοδοχείων περιλαμβάνουν τα παρακάτω στοιχεία:

- Το στοιχείο **Errors** το οποίο περιγράφει τα σφάλματα που προέκυψαν κατά τη διαδικασία αναζήτησης ξενοδοχείων στην περίπτωση ανεπιτυχώς εκτέλεσης της διαδικασίας, ή
- Στην περίπτωση επιτυχούς εκτέλεσης της διαδικασίας, ένα σύνολο από στοιχεία τα οποία είναι τα εξής:
 - **Success**: υποχρεωτικό στοιχείο με το οποίο καταδεικνύεται η επιτυχής έκβαση της διαδικασίας αναζήτησης.
 - **Warnings**: προαιρετικό στοιχείο που περιλαμβάνει ένα σύνολο από προειδοποιητικά μηνύματα σχετικά με την αναζήτηση. Για παράδειγμα, στην περίπτωση που το πλήθος των αποτελεσμάτων είναι πολύ μεγάλο μπορεί να εμφανιστεί ένα μήνυμα που να προτρέπει τον χρήστη να θέσει περισσότερα κριτήρια, ώστε να περιοριστεί το πλήθος των αποτελεσμάτων.
 - **Properties**: προαιρετικό στοιχείο που περιέχει πληροφορίες για τα ξενοδοχεία που βρέθηκαν. Περιλαμβάνει ένα στοιχείο **Property** για

κάθε ξενοδοχείο που βρέθηκε. Το στοιχείο Property περιλαμβάνει τα παρακάτω υποστοιχεία:

- VendorMessages: περιέχει τυχόν σχόλια που ορίζει ο ιδιοκτήτης του ξενοδοχείου, όπως για παράδειγμα μια γενική περιγραφή του ξενοδοχείου.
- Position: περιέχει τη γεωγραφική θέση του ξενοδοχείου (γεωγραφικό μήκος, πλάτος).
- Address: αναφέρει διεύθυνση του ξενοδοχείου.
- ContactNumbers: αναφέρει τα τηλέφωνα επικοινωνίας με το ξενοδοχείο.
- SearchValueMatch: περιέχει τον βαθμό στον οποίο το ξενοδοχείο ικανοποιεί τα κριτήρια αναζήτησης.

4.5.7 Παραδείγματα XML εγγράφων

Παρακάτω δίνεται ένα παράδειγμα XML εγγράφου για αναζήτηση ξενοδοχείων και το αντίστοιχο XML έγγραφο που περιέχει τα αποτελέσματα της αναζήτησης.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA/2002/11" MaxResponses="5">
  <Criteria>
    <Criterion ExactMatch="false" ImportanceType="Mandatory">
      <Address>
        <StateProv>Chania</StateProv>
      </Address>
      <RefPoint>Agia Anna Church - Kournas - Georgioupoli - Chania</RefPoint>
      <Radius Distance="5" />
      <HotelAmenity Code="Air conditioning" />
    </Criterion>
  </Criteria>
</OTA_HotelSearchRQ>
```

Όπως φαίνεται στο παραπάνω XML έγγραφο, ζητούνται τα πρώτα 5 ξενοδοχεία που βρίσκονται σε απόσταση 5 χιλιομέτρων από την εκκλησία της Αγίας Άννας στο χωριό Κουρνάς της Γεωργιούπολης του νομού Χανίων. Τα ξενοδοχεία πρέπει να παρέχουν κλιματισμό.

Το παρακάτω XML έγγραφο περιέχει τα αποτελέσματα της αναζήτησης.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRS xmlns="http://www.opentravel.org/OTA/2002/11" Target="Test">
  <Success />
  <Properties>
    <Property HotelCode="1232" HotelName="Kournas Village Hotel">
      <Address>
        <CityName>Kournas</CityName>
        <PostalCode>73007</PostalCode>
        <County>Georgioupoli</County>
        <StateProv>Chania</StateProv>
      </Address>
      <ContactNumbers>
        <ContactNumber PhoneNumber="08250 61416, 08250 61417" />
      </ContactNumbers>
      <SearchValueMatch Match="true">1.0</SearchValueMatch>
    </Property>
    <Property HotelCode="3279" HotelName="Kavros Beach">
      <Address>
        <CityName>Kournas</CityName>
        <PostalCode>73007</PostalCode>
        <County>Georgioupoli</County>
        <StateProv>Chania</StateProv>
      </Address>
      <ContactNumbers>
        <ContactNumber PhoneNumber="08250 61322" />
      </ContactNumbers>
      <SearchValueMatch Match="true">1.0</SearchValueMatch>
    </Property>
    <Property HotelCode="3281" HotelName="Mina">
      <Address>
        <CityName>Kournas</CityName>
        <PostalCode>73007</PostalCode>
        <County>Georgioupoli</County>
        <StateProv>Chania</StateProv>
      </Address>
      <ContactNumbers>
        <ContactNumber PhoneNumber="08250 61257" />
      </ContactNumbers>
      <SearchValueMatch Match="true">1.0</SearchValueMatch>
    </Property>
    <Property HotelCode="5590" HotelName="Manos Beach">
      <Address>
        <CityName>Kournas</CityName>
        <County>Georgioupoli</County>
        <StateProv>Chania</StateProv>
      </Address>
      <ContactNumbers>
        <ContactNumber PhoneNumber="08210 61221" />
      </ContactNumbers>
      <SearchValueMatch Match="true">1.0</SearchValueMatch>
    </Property>
    <Property HotelCode="5591" HotelName="Asimenia Ammoydia">
      <Address>
        <CityName>Kournas</CityName>
        <County>Georgioupoli</County>
        <StateProv>Chania</StateProv>
      </Address>
      <ContactNumbers>
        <ContactNumber PhoneNumber="08250 61223" />
      </ContactNumbers>
    </Property>
  </Properties>
</OTA_HotelSearchRS>
```

```
</ContactNumbers>  
<SearchValueMatch Match="true">1.0</SearchValueMatch>  
</Property>  
</Properties>  
</OTA_HotelSearchRS>
```

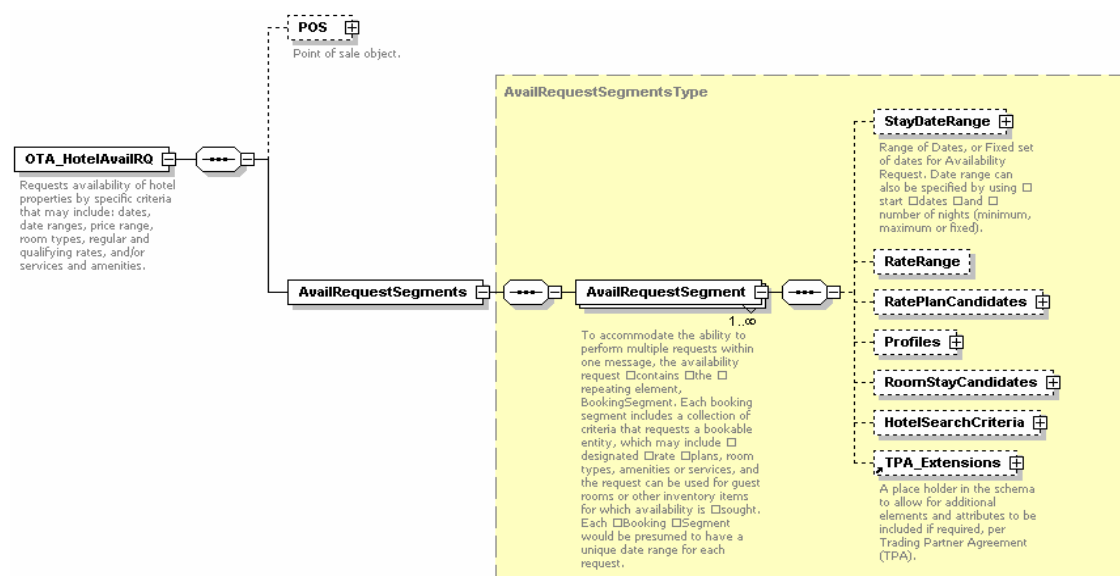
4.6 Υλοποίηση της Υπηρεσίας Αναζήτησης Διαθεσιμότητας

Στην ενότητα αυτή γίνεται η περιγραφή του XML Schema του OTA για την αναζήτηση διαθεσιμότητας σε ξενοδοχεία, καθώς και η περιγραφή της υλοποίησης της αντίστοιχης υπηρεσίας. Η αναζήτηση εναλλακτικής διαθεσιμότητας, στην περίπτωση που η αρχική διαδικασία δεν επιστρέφει αποτελέσματα, είναι μία ακόμη δυνατότητα της υπηρεσίας αυτής και μελετάται στην ενότητα αυτή. Μελετάται επίσης η χρήση του προφίλ χρήστη στο φιλτράρισμα των αποτελεσμάτων της αναζήτησης διαθεσιμότητας (και της εναλλακτικής διαθεσιμότητας), και προτείνεται μια επέκταση του αντίστοιχου XML Schema του OTA, ώστε να μπορούν να ενσωματωθούν παράμετροι χρήσης του προφίλ στο XML έγγραφο της αίτησης για αναζήτηση διαθεσιμότητας.

4.6.1 Περιγραφή των XML Σχημάτων του OTA για την αναζήτηση διαθεσιμότητας (OTA_HotelAvailRQ / OTA_HotelAvailRS)

Ο OTA παρέχει το κατάλληλο ζεύγος μηνυμάτων αίτησης/απόκρισης για την αναζήτηση διαθέσιμων δωματίων ή υπηρεσιών σε ξενοδοχεία. Το μήνυμα της αίτησης διαθεσιμότητας (OTA_HotelAvailRQ) περιέχει τα κριτήρια που χρειάζονται για την αναζήτηση, όπως το χρονικό διάστημα το οποίο μας ενδιαφέρει, το εύρος τιμών και ο τύπος των δωματίων, ενώ το μήνυμα που επιστρέφει τα αποτελέσματα (OTA_HotelAvailRS) περιέχει πληροφορίες όπως τα δωμάτια που βρέθηκαν, το κόστος τους κτλ.

Το XML Schema του OTA για την αναζήτηση διαθεσιμότητας σε ξενοδοχεία δίνεται στο παρακάτω σχήμα (Εικόνα 25).



Εικόνα 25 Το XML Schema του OTA για την αναζήτηση διαθεσιμότητας

Το αρχικό στοιχείο του XML Schema είναι το OTA_HotelAvailRQ και περιλαμβάνει το στοιχείο AvailRequestSegments το οποίο περιλαμβάνει το πολλαπλό στοιχείο AvailRequestSegment στο οποίο περιέχονται τα κριτήρια αναζήτησης διαθεσιμότητας.

Οι περιπτώσεις χρήσης που υποστηρίζονται από το XML Schema και υλοποιήθηκαν στην παρούσα εργασία είναι οι παρακάτω:

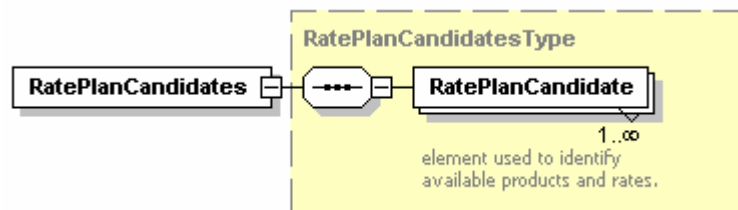
- Διαθεσιμότητα σε ένα ξενοδοχείο.
- Διαθεσιμότητα σε πολλά ξενοδοχεία.
- Διαθεσιμότητα με χρήση του προφίλ χρήστη.
- Εναλλακτική διαθεσιμότητα.
- Εναλλακτική διαθεσιμότητα με χρήση του προφίλ χρήστη.

4.6.2 Περιγραφή του στοιχείου AvailRequestSegment

Το στοιχείο AvailRequestSegment περιέχει τα κριτήρια αναζήτησης διαθεσιμότητας τα οποία είναι:

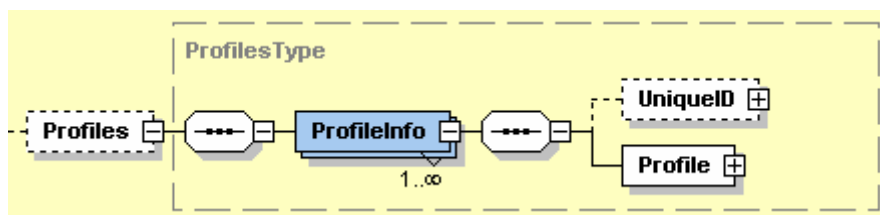
- StayDateRange - ορίζει την χρονική περίοδο στην οποία αναζητούμε διαθέσιμα δωμάτια. Περιέχει τις ιδιότητες:
 - Start – η ημερομηνία έναρξης για την οποία ενδιαφερόμαστε
 - Duration – η διάρκεια

- End – η τελική ημερομηνία που μας ενδιαφέρει
- RateRange – ορίζει το εύρος τιμών που επιθυμούμε. Περιέχει τις ιδιότητες:
 - CurrencyCode – το νόμισμα
 - MinRate – η ελάχιστη τιμή
 - MaxRate – η μέγιστη τιμή
- RatePlanCandidates – περιέχει το στοιχείο RatePlan, στο οποίο θέτουμε το Rate Plan που επιθυμούμε, μέσω της ιδιότητας RatePlanType.



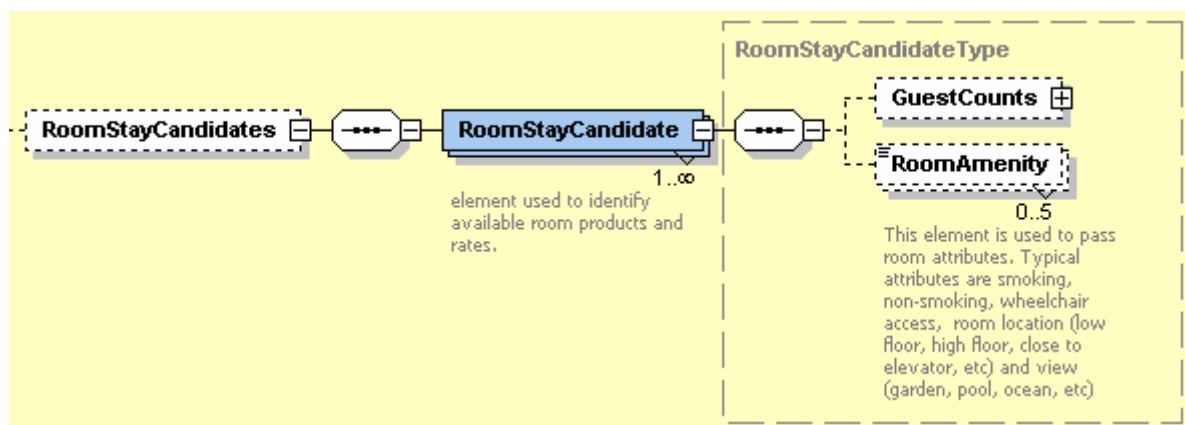
Εικόνα 26 Το XML Schema του στοιχείου RatePlanCandidates

- Profiles – τυχόν προφίλ που θέλουμε να χρησιμοποιήσουμε. Η χρήση του προφίλ γίνεται μέσω του στοιχείου UniqueID του στοιχείου ProfileInfo. Στο στοιχείο UniqueID δίνουμε την τιμή του UniqueID του προφίλ που μας ενδιαφέρει.



Εικόνα 27 Το XML Schema του στοιχείου Profiles

- RoomStayCandidates – πληροφορίες για τα δωμάτια που ψάχνουμε. Περιλαμβάνει τις ιδιότητες των υποψήφιων δωματίων, όπως για παράδειγμα τον τύπο του κρεβατιού, και ευκολίες που θέλουμε να παρέχουν τα δωμάτια (RoomAmenity).



Εικόνα 28 Το XML Schema του στοιχείου RoomStayCandidates.

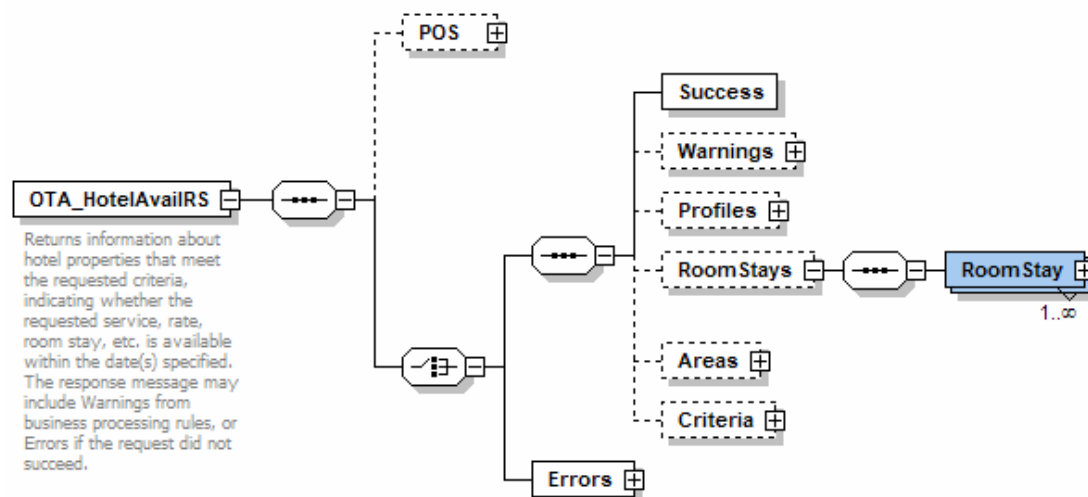
Οι ιδιότητες των υποψήφιων δωματίων συνοψίζονται στον παρακάτω πίνακα

Name	Type	Use	Annotation
RoomType	StringLength16	προαιρετικό	Πρόκειται για έναν κωδικό, η τιμή του οποίου δείχνει τον τύπο του δωματίου που ζητήθηκε, π.χ. Double, king, κτλ.
RoomLocationCode	OTA_CodeType	προαιρετικό	Δηλώνει την τοποθέτηση του δωματίου ως προς τον χώρο του ξενοδοχείου, π.χ. "Κοντά στην έξοδο", "κοντά στον ανελκυστήρα", "σε χαμηλό όροφο" ή "σε υψηλό όροφο".
RoomViewCode	OTA_CodeType	προαιρετικό	Δηλώνει την θέα που έχει το δωμάτιο, π.χ. "θέα στον ωκεανό", "θέα στην πισίνα" ή "θέα στον κήπο".
BedTypeCode	OTA_CodeType	προαιρετικό	Δηλώνει τον τύπο του κρεβατιού (ή κρεβατιών) που περιλαμβάνει το δωμάτιο, π.χ. Double, Twin, Queen, ή King.
NonSmoking	xs:boolean	προαιρετικό	Δηλώνει την απαγόρευση ή μη του καπνίσματος.
Quantity	Numeric1to999	προαιρετικό	Το πλήθος των δωματίων που ζητείται.

- TPA_Extensions – παρέχει τη δυνατότητα να γίνουν επεκτάσεις στο XML Schema ανάλογα με τις ανάγκες των συνεργατών. Αυτή η δυνατότητα αξιοποιήθηκε στην εργασία για να προσδώσουμε σημασιολογία για τον τρόπο χρήσης του προφίλ και της χρήσης εναλλακτικής διαθεσιμότητας.

Η διαδικασία αναζήτησης διαθεσιμότητας μπορεί να πετύχει, επιστρέφοντας τα αποτελέσματα, ή να αποτύχει, εάν δεν βρεθούν αποτελέσματα ή συμβεί κάποιο σφάλμα.

Το XML Schema του OTA για τα μηνύματα που επιστρέφουν τα αποτελέσματα της αναζήτησης διαθεσιμότητας δίνεται στο παρακάτω σχήμα (Εικόνα 29).



Εικόνα 29 Το XML Schema του OTA για τα μηνύματα που επιστρέφουν τα αποτελέσματα της αναζήτησης διαθεσιμότητας

Στην περίπτωση που δεν παραχθούν αποτελέσματα ή συμβεί κάποιο σφάλμα τότε επιστρέφονται τα κατάλληλα μηνύματα σφάλματος στο στοιχείο Errors. Στην περίπτωση που η διαδικασία αναζήτησης διαθεσιμότητας πετύχει και παραχθούν αποτελέσματα, τότε αυτά επιστρέφονται στο στοιχείο RoomStays. Το στοιχείο RoomStays αποτελείται από ένα ή περισσότερα στοιχεία RoomStay τα οποία περιγράφουν τα διαθέσιμα «προϊόντα» για μια συγκεκριμένη χρονική περίοδο σε ένα συγκεκριμένο ξενοδοχείο. Ο όρος προϊόντα αφορά στα RoomRates, δηλαδή στους διαθέσιμους τύπους δωματίων (RoomTypes) που προσφέρονται σε συγκεκριμένα διαθέσιμα RatePlans. Έτσι το στοιχείο RoomStay αποτελείται από τα παρακάτω στοιχεία:

- RatePlans – περιέχει τα διαθέσιμα RatePlans.
- RoomTypes – περιέχει τα διαθέσιμα RoomTypes.
- RoomRates – περιέχει τους διαθέσιμους συνδυασμούς RatePlan και RoomType, καθώς και την τιμή στην οποία παρέχονται.

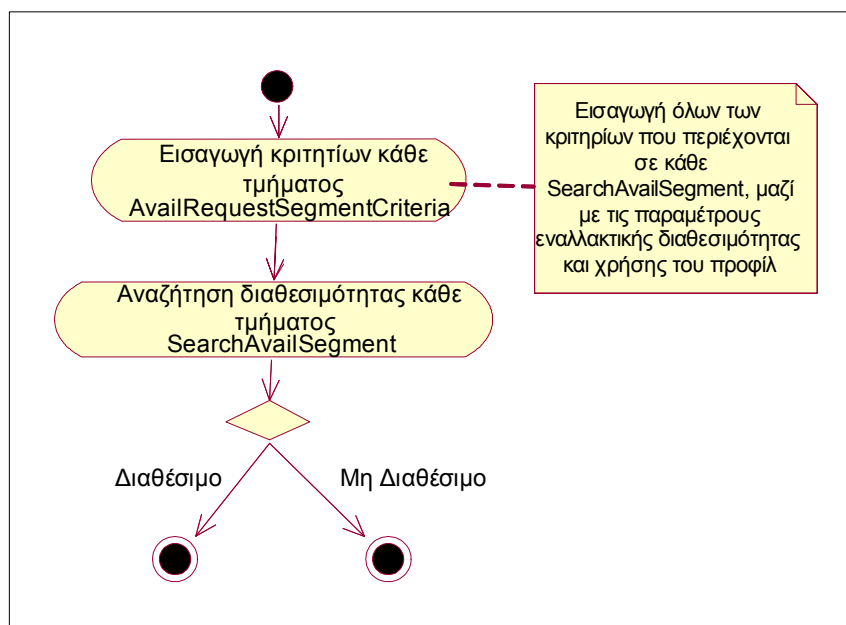
Στη επόμενη ενότητα γίνεται αναλυτική περιγραφή της διαδικασίας αναζήτησης διαθεσιμότητας.

4.6.3 Η διαδικασία της αναζήτησης διαθεσιμότητας

Η διαδικασία της αναζήτησης όπως αυτή υλοποιήθηκε στην παρούσα εργασία ακολουθεί τα παρακάτω βήματα:

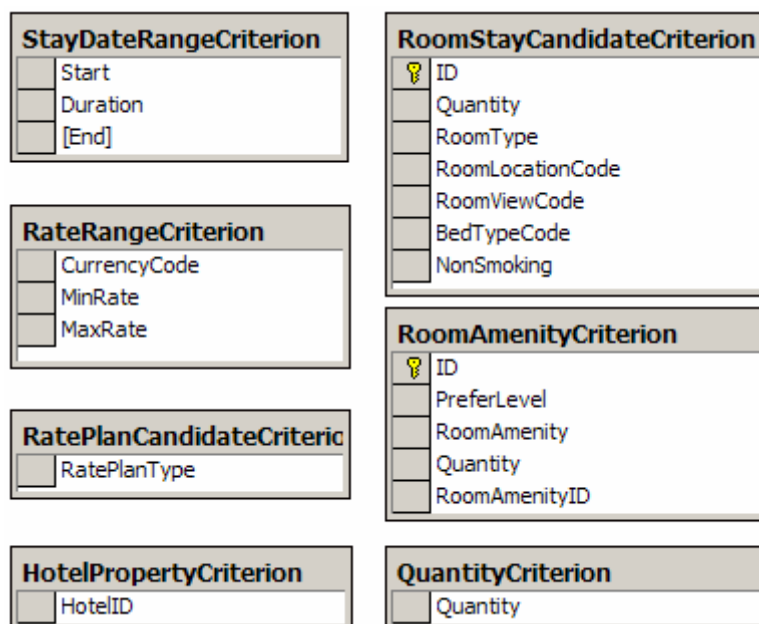
- Για κάθε στιγμιότυπο του στοιχείου AvailRequestSegment εισάγουμε τα κριτήρια που περιέχει στη βάση δεδομένων και ψάχνουμε για διαθεσιμότητα με βάση αυτά τα κριτήρια.
- Εάν το βήμα πετύχει συλλέγουμε τα αποτελέσματα και συνεχίζουμε με το επόμενο στιγμιότυπο του AvailRequestSegment.
- Εάν το βήμα αποτύχει, απλά συνεχίζουμε με το επόμενο στιγμιότυπο.

Η διαδικασία αυτή δίνεται διαγραμματικά στο παρακάτω σχήμα (Εικόνα 30).



Εικόνα 30 Το γενικό διάγραμμα δραστηριοτήτων της διαδικασίας αναζήτησης διαθεσιμότητας

Στο επόμενο σχήμα (Εικόνα 31) δίνονται οι πίνακες της βάσης δεδομένων στους οποίους αποθηκεύονται τα κριτήρια της αναζήτησης διαθεσιμότητας.

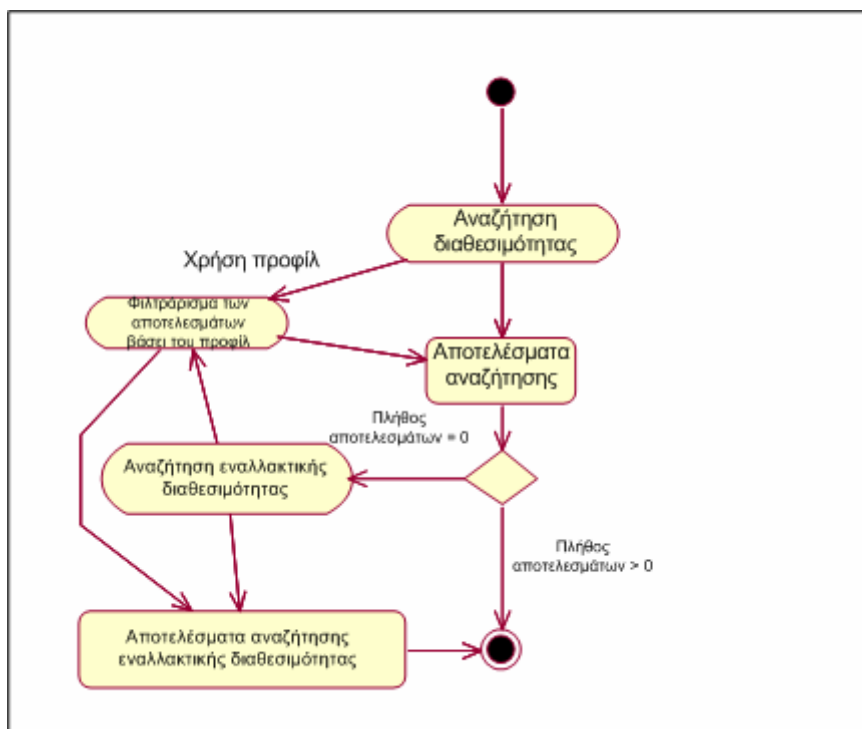


Εικόνα 31 Οι πίνακες της βάσης δεδομένων όπου κρατούνται τα κριτήρια της αναζήτησης διαθεσιμότητας

- Ο πίνακας StayDateRangeCriterion αποθηκεύει το κριτήριο που αφορά στην χρονική περίοδο κατά την οποία ζητείται η διαθεσιμότητα. Τα πεδία του πίνακα αυτού είναι τα εξής:
 - Start – η ημερομηνία έναρξης της χρονικής περιόδου
 - End – η ημερομηνία λήξης της χρονικής περιόδου
 - Duration – το χρονικό διάστημα
- Ο πίνακας RateRangeCriterion αποθηκεύει το κριτήριο του εύρους τιμών που πρέπει να ικανοποιούν τα διαθέσιμα δωμάτια. Τα πεδία του πίνακα αυτού είναι τα εξής:
 - MinRate – η ελάχιστη τιμή
 - MaxRate – η μέγιστη τιμή
 - CurrencyCode – το νόμισμα
- Ο πίνακας RatePlanCandidateCriterion αποθηκεύει το RatePlan που ζητείται.
- Ο πίνακας HotelPropertyCriterion αποθηκεύει τα ξενοδοχεία (HotelIDs) στα οποία αναζητούνται διαθέσιμα δωμάτια.
- Ο πίνακας RoomStayCandidateCriterion αποθηκεύει πληροφορίες σχετικά με τον τύπο των δωματίων που ζητείται. Τα πεδία του πίνακα αυτού είναι τα εξής:
 - RoomType – ο τύπος του δωματίου.

- RoomLocationCode – η τοποθεσία του δωματίου ως προς το ξενοδοχείο.
 - RoomViewCode – η θέα που έχει το δωμάτιο.
 - BedTypeCode – ο τύπος των κρεβατιών του δωματίου.
 - NonSmoking – αν είναι καπνιστών ή μη καπνιστών
- Ο πίνακας RoomAmenityCriterion αποθηκεύει πληροφορίες σχετικά με τις διευκολύνσεις που παρέχει το δωμάτιο. Τα πεδία του πίνακα αυτού είναι τα εξής:
 - RoomAmenity – η συγκεκριμένη διευκόλυνση που ζητείται.
 - PreferLevel – ο βαθμός προτίμησης της διευκόλυνσης, ο οποίος μπορεί να πάρει τις παρακάτω τιμές:
 - Only – η συγκεκριμένη διευκόλυνση θα πρέπει να παρέχεται οπωσδήποτε
 - Unacceptable - η συγκεκριμένη διευκόλυνση θα πρέπει να οπωσδήποτε να μην παρέχεται.
 - Preferred – προτιμάται να παρέχεται η συγκεκριμένη διευκόλυνση
- Ο πίνακας QuantityCriterion αποθηκεύει το πλήθος των δωματίων που ζητούνται στο πεδίο Quantity

Στο επόμενο σχήμα (Εικόνα 32) δίνεται το διάγραμμα δραστηριοτήτων της αναζήτησης διαθεσιμότητας για κάθε AvailRequestSegment



Εικόνα 32 Το Διάγραμμα δραστηριοτήτων της διαδικασίας αναζήτησης διαθεσιμότητας

Η διαδικασία της αναζήτησης ακολουθεί τα παρακάτω βήματα:

- Συλλέγονται όλα τα RoomTypes που αντιστοιχούν στα ξενοδοχεία που επιλέχθηκαν
- Στη συνέχεια από τα παραπάνω RoomTypes επιλέγονται εκείνα που ικανοποιούν τα κριτήρια που έχουν δοθεί μέσω των ιδιοτήτων του στοιχείου RoomStayCandidate
- Φιλτράρονται τα παραπάνω RoomTypes με τις διευκολύνσεις δωματίων που ορίστηκαν, ως εξής:
 - ο Διαγράφονται τα RoomTypes που έχουν διευκολύνσεις με *PreferLevel* = “Unacceptable”
 - ο Διαγράφονται τα RoomTypes που δεν έχουν διευκολύνσεις με *PreferLevel* = “Only”
 - ο Υπολογίζεται ο αριθμός των διευκολύνσεων με *PreferLevel* = “Preferred” που έχει το κάθε RoomType και του αποδίδεται το κατάλληλο βάρος.
 - ο Κατασκευάζεται ένας προσωρινός πίνακας «roomrate» με τα παρακάτω πεδία: (hotelid, rrid, rpid, rtid, rate, isAlternate, weight) όπου:

- hotelid – το id του ξενοδοχείου
 - rrid – το id του roomrate
 - rpid – το id του rateplan
 - rtid – το id του roomtype
 - isAlternate – παίρνει τιμή null, χρησιμοποιείται αργότερα στην εναλλακτική διαθεσιμότητα
 - weight – το βάρος το οποίο αποδίδεται στο συγκεκριμένο roomrate
- Εισάγονται στον πίνακα roomrate τα RoomTypes που βρέθηκαν παραπάνω και τα hotelid, rrid, rpid που συνδέονται με αυτά τα RoomTypes, καθώς και το πλήθος των “Preferred” διευκολύνσεων δωματίων που ικανοποιούν σαν το βάρος του κάθε roomrate.
 - Διαγράφονται από τον πίνακα «roomrate» εκείνα τα roomrates που δεν ικανοποιούν το κριτήριο RatePlanCandidate και το εύρος τιμών που δόθηκε
 - Ελέγχεται ο πίνακας των κρατήσεων ([Bookings]) και των δωματίων ([Room]) και υπολογίζεται ποια roomrates και σε τι ποσότητα είναι διαθέσιμα.

Εάν η διαδικασία αναζήτησης διαθεσιμότητας δεν επιστρέψει αποτελέσματα, τότε εκτελείται – εάν το έχει επιλέξει ο χρήστης – η διαδικασία αναζήτησης εναλλακτικής διαθεσιμότητας.

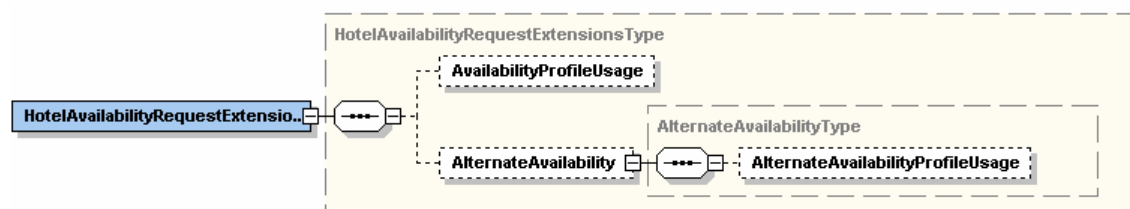
4.6.4 Η διαδικασία της αναζήτησης εναλλακτικής διαθεσιμότητας

Η διαδικασία αναζήτησης εναλλακτικής διαθεσιμότητας είναι όμοια με τη διαδικασία αναζήτησης διαθεσιμότητας. Η διαφορά τους έγκειται στο ότι η αναζήτηση εναλλακτικής διαθεσιμότητας εφαρμόζεται στο σύνολο των ξενοδοχείων που επιστράφηκαν από την αμέσως προηγούμενη αίτηση αναζήτησης ξενοδοχείων αφαιρώντας εκείνα τα ξενοδοχεία στα οποία εκτελέστηκε η κανονική αναζήτηση διαθεσιμότητας.

Για την εκτέλεση λοιπόν της διαδικασίας αναζήτησης εναλλακτικής διαθεσιμότητας απαιτείται να έχει προηγηθεί αίτηση αναζήτησης ξενοδοχείων, ώστε να υπάρχει διαθέσιμο ένα σύνολο ξενοδοχείων.

Για την υλοποίηση της διαδικασίας αναζήτησης εναλλακτικής διαθεσιμότητας καθώς και της χρήσης των προτιμήσεων του προφίλ πελάτη – που περιγράφεται παρακάτω – στην αναζήτηση διαθεσιμότητας και εναλλακτικής διαθεσιμότητας, απαιτήθηκε η επέκταση του XML Schema του OTA για την αναζήτηση διαθεσιμότητας, ώστε να δίνεται η δυνατότητα προσθήκης της κατάλληλης σημασιολογίας. Έτσι σχεδιάστηκε ένα XML Schema που περιγράφει τα δεδομένα που μπορούν να προστεθούν σε ένα μήνυμα αναζήτησης διαθεσιμότητας για να δώσει τις κατάλληλες πληροφορίες στο σύστημα όσον αφορά την εκτέλεση της εναλλακτικής διαθεσιμότητας και της χρήσης του προφίλ στην αναζήτηση διαθεσιμότητας και εναλλακτικής διαθεσιμότητας.

Το XML Schema που σχεδιάστηκε δίνεται στο παρακάτω σχήμα (Εικόνα 33).



Εικόνα 33 Το XML Schema που σχεδιάστηκε για την υποστήριξη της χρήσης του προφίλ χρήστη και προστέθηκε στο βασικό XML Schema του OTA

Το στοιχείο AvailabilityProfileUsage δηλώνει την επιθυμία του χρήστη να φιλτραριστούν τα αποτελέσματα της αναζήτησης με βάση τις προτιμήσεις που βρίσκονται στο προφίλ πελάτη. Οι τιμές που μπορεί να πάρει το στοιχείο αυτό είναι:

- ExactMatch – τα αποτελέσματα πρέπει να ικανοποιούν πλήρως τις προτιμήσεις του πελάτη όπως αυτές δηλώνονται στο προφίλ του
- WeightsOnly - τα αποτελέσματα φιλτράρονται ως προς τις προτιμήσεις του πελάτη και αποκτούν την ανάλογη βαρύτητα

Το στοιχείο AlternateAvailability δηλώνει τη χρήση της διαδικασίας εναλλακτικής διαθεσιμότητας στην περίπτωση που η κανονική αναζήτηση διαθεσιμότητας δεν επιστρέψει αποτελέσματα. Το στοιχείο AlternateAvailabilityProfileUsage δηλώνει τη χρήση του προφίλ στα αποτελέσματα της εναλλακτικής διαθεσιμότητας. Οι τιμές που παίρνει το στοιχείο είναι ίδιες με τις τιμές του στοιχείου AvailabilityProfileUsage.

Οι προδιαγραφές του επιτρέπουν επεκτάσεις στα XML Schemata, όπως αυτή που προαναφέρθηκε, μέσω του στοιχείου TPA_Extensions.

4.6.5 Η χρήση των προτιμήσεων του προφίλ πελάτη στην αναζήτηση διαθεσιμότητας και εναλλακτικής διαθεσιμότητας

Τα αποτελέσματα της αναζήτησης διαθεσιμότητας ή εναλλακτικής διαθεσιμότητας μπορούν να φιλτραριστούν με βάση τις προτιμήσεις ξενοδοχείων που τυχόν περιέχονται στο προφίλ πελάτη. Οι πληροφορίες του προφίλ χρήστη που χρησιμοποιούνται στην διαδικασία φιλτραρίσματος δίνονται στους παρακάτω πίνακες. Ο πρώτος πίνακας περιέχει τις προτιμήσεις του χρήστη που σχετίζονται με τα ξενοδοχεία, ενώ ο δεύτερος πίνακας περιέχει τις προτιμήσεις του χρήστη που σχετίζονται με τα δωμάτια.

Προτιμήσεις ξενοδοχείων			
όνομα	τύπος	ιδιότητες	περογραφή
PropertyNamePref	Element	PreferLevel	Όνομα ξενοδοχείου
PropertyAmenityPref	Element	PreferLevel, PropertyAmenityType	Διευκολύνσεις ξενοδοχείου

Προτιμήσεις δωματίων			
όνομα	τύπος	ιδιότητες	περογραφή
RoomAmenityPref	Element	PreferLevel, RoomAmenityType	Διευκολύνσεις δωματίων
RoomLocationPref	Element	PreferLevel, RoomLocationType	Τοποθεσία δωματίου
BedTypePref	Element	PreferLevel, BedType	Τύπος κρεβατιού

Υλοποιήθηκαν δύο διαφορετικές μέθοδοι χρήσης του προφίλ οι οποίες, όπως προαναφέρθηκε είναι:

- ExactMatch – τα αποτελέσματα πρέπει να ικανοποιούν πλήρως τις προτιμήσεις του πελάτη όπως αυτές δηλώνονται στο προφίλ του
- WeightsOnly - τα αποτελέσματα φιλτράρονται ως προς τις προτιμήσεις του πελάτη και αποκτούν την ανάλογη βαρύτητα

4.6.5.1 Η μέθοδος ExactMatch

Στη μέθοδο ExactMatch εκτελούνται τα παρακάτω βήματα

- Διαγράφονται από τη λίστα των αποτελεσμάτων όλα τα ξενοδοχεία και τα δωμάτια που δεν ικανοποιούν τις προτιμήσεις με *PreferLevel* = “Only”
- Διαγράφονται από τη λίστα των αποτελεσμάτων όλα τα ξενοδοχεία και τα δωμάτια που ικανοποιούν τις προτιμήσεις με *PreferLevel* = “Unacceptable”
- Εφαρμόζονται τα κατάλληλα βάρη στα ξενοδοχεία και τα δωμάτια της λίστας των αποτελεσμάτων που ικανοποιούν τις προτιμήσεις με *PreferLevel* = “Preferred”

4.6.5.2 Η μέθοδος **WeightsOnly**

Στη μέθοδο **WeightsOnly** δεν διαγράφονται ξενοδοχεία ή δωμάτια από τη λίστα των αποτελεσμάτων, αλλά αποδίδονται σε αυτά τα κατάλληλα βάρη ανάλογα με το *PreferLevel*, που ορίζονται παραμετρικά ως εξής:

- “Only”: a
- “Unacceptable”: b
- “Preferred”: c

4.6.6 Παραδείγματα XML εγγράφων

Παρακάτω δίνονται δύο XML έγγραφα σαν παράδειγμα χρήσης της μεθόδου αναζήτησης διαθεσιμότητας. Στο πρώτο XML έγγραφο που αποτελεί την αίτηση για αναζήτηση διαθεσιμότητας, περιγράφεται ότι αναζητούνται διαθέσιμα δωμάτια στα ξενοδοχεία με κωδικό «1232» και «5591» για την περίοδο 4-5 Μαΐου 2005. Επίσης πρέπει η τιμή του κάθε δωματίου πρέπει να κυμαίνεται από 10 έως 100 ευρώ και πρέπει να παρέχουν κλιματισμό (κωδικός 2).

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA/2002/11">
  <AvailRequestSegments>
    <AvailRequestSegment>
      <StayDateRange Start="2005-5-4" Duration="1" End="2005-5-5" />
      <RateRange CurrencyCode="eu" MinRate="10" MaxRate="100"/>
      <RatePlanCandidates>
        <RatePlanCandidate RatePlanType="3" />
      </RatePlanCandidates>
      <RoomStayCandidates>
        <RoomStayCandidate>
```

```

        <RoomAmenity PreferLevel="Only" RoomAmenity="2"
Quantity="1">2</RoomAmenity>
    </RoomStayCandidate>
</RoomStayCandidates>
<HotelSearchCriteria>
    <Criterion>
        <HotelRef HotelCode="1232" />
    </Criterion>
    <Criterion>
        <HotelRef HotelCode="5591" />
    </Criterion>
</HotelSearchCriteria>
</AvailRequestSegment>
</AvailRequestSegments>
</OTA_HotelAvailRQ>

```

Τα διαθέσιμα δωμάτια που πληρούν τα παραπάνω κριτήρια δίνονται στο παρακάτω XML έγγραφο.

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2002/11">
    <Success />
    <RoomStays MoreIndicator="no">
        <RoomStay PromotionCode="prom" IsAlternate="false">
            <RoomTypes>
                <RoomType RoomTypeCode="3919" NumberOfUnits="5">
                    <Amenity>Air conditioning</Amenity>
                    <Amenity>Cable television</Amenity>
                    <Amenity>Stove</Amenity>
                    <Amenity>Shower only</Amenity>
                    <Amenity>Loft</Amenity>
                </RoomType>
            </RoomTypes>
            <RatePlans>
                <RatePlan RatePlanCode="H1232RP3" EffectiveDate="2003-08-24"
ExpireDate="2003-10-24" RatePlanType="Corporate" RatePlanID="2846"/>
            </RatePlans>
            <RoomRates>
                <RoomRate RoomTypeCode="3919" NumberOfUnits="1" RatePlanCode="2846">
                    <Rates>
                        <Rate AgeQualifyingCode="1">
                            <Base AmountAfterTax="50.0" />
                            <Total AmountAfterTax="50.0" />
                        </Rate>
                    </Rates>
                </RoomRate>
            </RoomRates>
            <TimeSpan Start="2005-5-4" Duration="1" End="2005-5-5">
                <DateWindowRange></DateWindowRange>
            </TimeSpan>
            <BasicPropertyInfo HotelCode="1232" HotelName="Kournas Village Hotel">
                <Address>
                    <CityName>Kournas</CityName>
                    <PostalCode>73007</PostalCode>
                    <County>Georgiupoli</County>
                    <StateProv>Chania</StateProv>
                </Address>
            </BasicPropertyInfo>
        </RoomStay>
    </RoomStays>
</OTA_HotelAvailRS>

```

```

        <ContactNumbers>
            <ContactNumber PhoneNumber="08250 61416, 08250 61417" />
        </ContactNumbers>
    </BasicPropertyInfo>
    <AlternateInfo>0.0</AlternateInfo>
</RoomStay>
<RoomStay PromotionCode="prom" IsAlternate="false">
    <RoomTypes>
        <RoomType RoomTypeCode="20644" NumberOfUnits="13">
            <Amenity>Recliner</Amenity>
            <Amenity>Air conditioning individually controlled in room</Amenity>
            <Amenity>Washer/dryer</Amenity>
            <Amenity>Air conditioning</Amenity>
            <Amenity>CD player</Amenity>
        </RoomType>
    </RoomTypes>
    <RatePlans>
        <RatePlan RatePlanCode="H5591RP3" EffectiveDate="2003-08-24"
        ExpireDate="2003-10-22" RatePlanType="Corporate" RatePlanID="14877" />
    </RatePlans>
    <RoomRates>
        <RoomRate RoomTypeCode="20644" NumberOfUnits="1"
        RatePlanCode="14877">
            <Rates>
                <Rate AgeQualifyingCode="1">
                    <Base AmountAfterTax="80.0" />
                    <Total AmountAfterTax="80.0" />
                </Rate>
            </Rates>
        </RoomRate>
    </RoomRates>
    <TimeSpan Start="2005-5-4" Duration="1" End="2005-5-5">
        <DateWindowRange></DateWindowRange>
    </TimeSpan>
    <BasicPropertyInfo HotelCode="5591" HotelName="Asimenia Ammoydia">
        <Address>
            <CityName>Kournas</CityName>
            <County>Georgioupoli</County>
            <StateProv>Chania</StateProv>
        </Address>
        <ContactNumbers>
            <ContactNumber PhoneNumber="08250 61223" />
        </ContactNumbers>
    </BasicPropertyInfo>
    <AlternateInfo>0.0</AlternateInfo>
</RoomStay>
</RoomStays>
</OTA_HotelAvailRS>

```

4.7 Υλοποίηση της Διαδικασίας Κράτησης Ξενοδοχείων

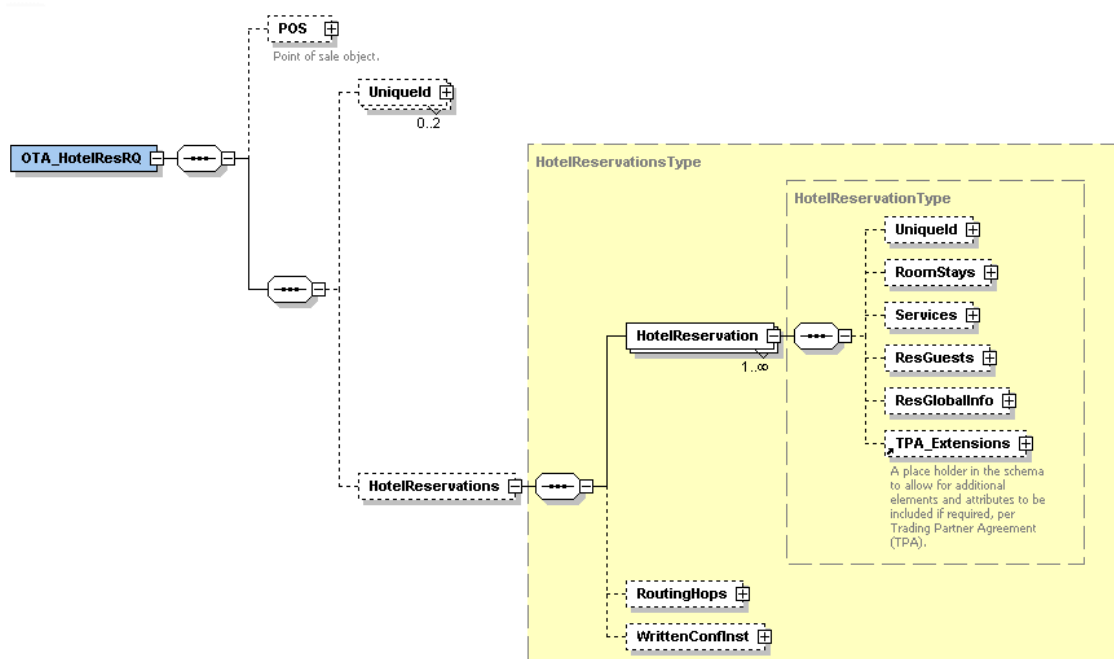
Στην ενότητα αυτή περιγράφεται η υλοποίηση των διαδικασιών διαχείρισης κρατήσεων δωματίων ξενοδοχείων και παρατίθενται τα αντίστοιχα XML Schemata του OTA. Περιγράφονται δηλαδή η δημιουργία μιας καινούργιας κράτησης, η

τροποποίηση και η αγνόηση μιας κράτησης που εκκρεμεί, η τροποποίηση μιας υπάρχουσας επικυρωμένης κράτησης, καθώς και η ακύρωση και η ανάκτηση μιας επικυρωμένης κράτησης.

4.7.1 Περιγραφή των XML σχημάτων του OTA για την διαχείριση κρατήσεων ξενοδοχείων (OTA_HotelResRQ/OTA_HotelResRS)

Το ζεύγος μηνυμάτων με το οποίο εκτελούνται οι συνδιαλλαγές για την πραγματοποίηση κρατήσεων περιγράφεται από το OTA_HotelResRQ¹⁷ και το OTA_HotelResRS XML Schema του OTA. Το OTA_HotelResRQ XML Schema περιγράφει τη δομή των μηνυμάτων που περιέχουν την αίτηση για κράτηση δωματίων σε ξενοδοχεία. Το OTA_HotelResRS XML Schema περιγράφει τη δομή των μηνυμάτων που περιέχουν την απόκριση στην αίτηση για κράτηση.

Το XML Schema του OTA για την αίτηση κράτησης δωματίων σε ξενοδοχεία δίνεται στο παρακάτω σχήμα (Εικόνα 34).



Εικόνα 34 Το XML Schema του OTA για την αίτηση κράτησης δωματίων σε ξενοδοχεία

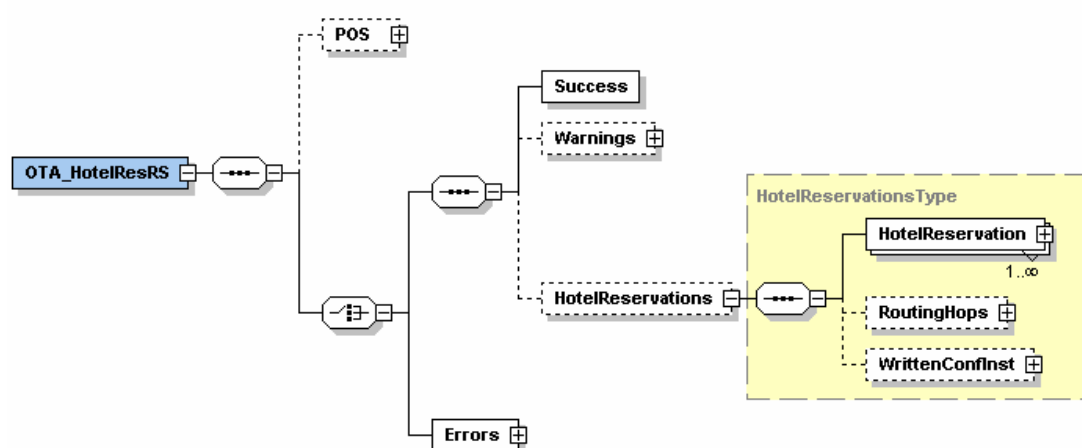
¹⁷ Το XML Schema OTA_HotelResRQ εκτός από την κράτηση δωματίων, μπορεί να χρησιμοποιηθεί και για την κράτηση διαφόρων υπηρεσιών που μπορεί να διαθέτει το ξενοδοχείο, μέσω του στοιχείου Services. Στη συγκεκριμένη όμως εργασία υλοποιήθηκε μόνο η κράτηση δωματίων, και οποιαδήποτε αναφορά που θα γίνει στη συνέχεια του κεφαλαίου στο XML Schema αυτό θα αναφέρεται στην κράτηση δωματίων.

Η συνδιαλλαγή μιας κράτησης μπορεί να αποτελείται από πολλά στάδια στα οποία ανταλλάσσονται μηνύματα του συγκεκριμένου ζεύγους (OTA_HotelResRQ/OTA_HotelResRS). Η κατάσταση στην οποία βρίσκεται κάθε στιγμή η κράτηση περιγράφεται από την τιμή της ιδιότητας ResResponseType του στοιχείου OTA_HotelResRS, ενώ η κατάσταση στην οποία θέλουμε να μεταβεί η κράτηση, δηλαδή η ενέργεια που θέλουμε να εκτελέσουμε (π.χ. αρχικοποίηση, τροποποίηση, αγνόηση ή επικύρωση), περιγράφεται από την τιμή της ιδιότητας ResStatus του στοιχείου OTA_HotelResRQ. Αυτές οι ιδιότητες και οι τιμές που παίρνουν περιγράφονται στη συνέχεια.

Το στοιχείο OTA_HotelResRQ, το οποίο είναι και το αρχικό στοιχείο του μηνύματος, περιέχει την ιδιότητα ResStatus που καταδεικνύει την κατάσταση της κράτησης. Οι τιμές που μπορεί να πάρει είναι:

- Initiate – για τη δημιουργία καινούργιας κράτησης.
- Modify – για τροποποίηση υπάρχουσας ή εκκρεμούς κράτησης.
- Ignore – για την αγνόηση εκκρεμούς κράτησης.
- Commit – για την επικύρωση μιας εκκρεμούς κράτησης.

Στο παρακάτω σχήμα (Εικόνα 35) δίνεται το XML Schema του OTA που περιγράφει την απόκριση στα μηνύματα τύπου OTA_HotelResRQ για δημιουργία, τροποποίηση, αγνόηση ή επικύρωση μιας κράτησης.



Εικόνα 35 Το XML Schema του OTA για την απόκριση στην αίτηση κράτησης ξενοδοχείων

Το αρχικό στοιχείο OTA_HotelResRS περιέχει την ιδιότητα ResResponseType η οποία μπορεί να πάρει μία από τις παρακάτω τιμές που καταδεικνύουν την κατάσταση της συνδιαλλαγής:

- Pending – δηλώνει ότι η κράτηση είναι σε εκκρεμότητα
- Modified - δηλώνει ότι η κράτηση τροποποιήθηκε
- Ignored - δηλώνει ότι η κράτηση αγνοήθηκε
- Committed - δηλώνει ότι η κράτηση επικυρώθηκε
- Cancelled - δηλώνει ότι η κράτηση ακυρώθηκε

Εάν η συνδιαλλαγή κράτησης επιτύχει τότε το XML έγγραφο θα περιέχει τα παρακάτω στοιχεία:

- Success (υποχρεωτικά) – καταδεικνύει ότι η συνδιαλλαγή πέτυχε
- HotelReservations (προαιρετικά) – περιέχει το στοιχείο HotelReservation το οποίο περιέχει όλες τις πληροφορίες σχετικά με τις κρατήσεις που έγιναν. Επίσης το στοιχείο HotelReservation περιέχει το στοιχείο ResGlobalInfo στο οποίο περιέχεται το αναγνωριστικό της κράτησης που έγινε (HotelReservation ID), ώστε να μπορεί στη συνέχεια ο χρήστης να ανακτήσει την κράτηση από την βάση δεδομένων του ebXML OTA Server και να την τροποποιήσει ή να την ακυρώσει.

Εάν η συνδιαλλαγή κράτησης αποτύχει τότε το XML έγγραφο θα περιέχει μόνο το στοιχείο Errors στο οποίο καταγράφονται τα σφάλματα που προέκυψαν κατά την συνδιαλλαγή.

4.7.2 Επεξεργασία κρατήσεων

Στη συνέχεια περιγράφονται η διαδικασία δημιουργίας καινούργιας κράτησης, η οποία μπορεί να περιλαμβάνει τροποποίηση ή και αγνόηση της κράτησης που βρίσκεται σε εξέλιξη, η διαδικασία τροποποίησης υπάρχουσας κράτησης, η διαδικασία ανάκτησης μιας υπάρχουσας κράτησης από την βάση δεδομένων του ebXML OTA Server, και η διαδικασία ακύρωσης μιας υπάρχουσας κράτησης.

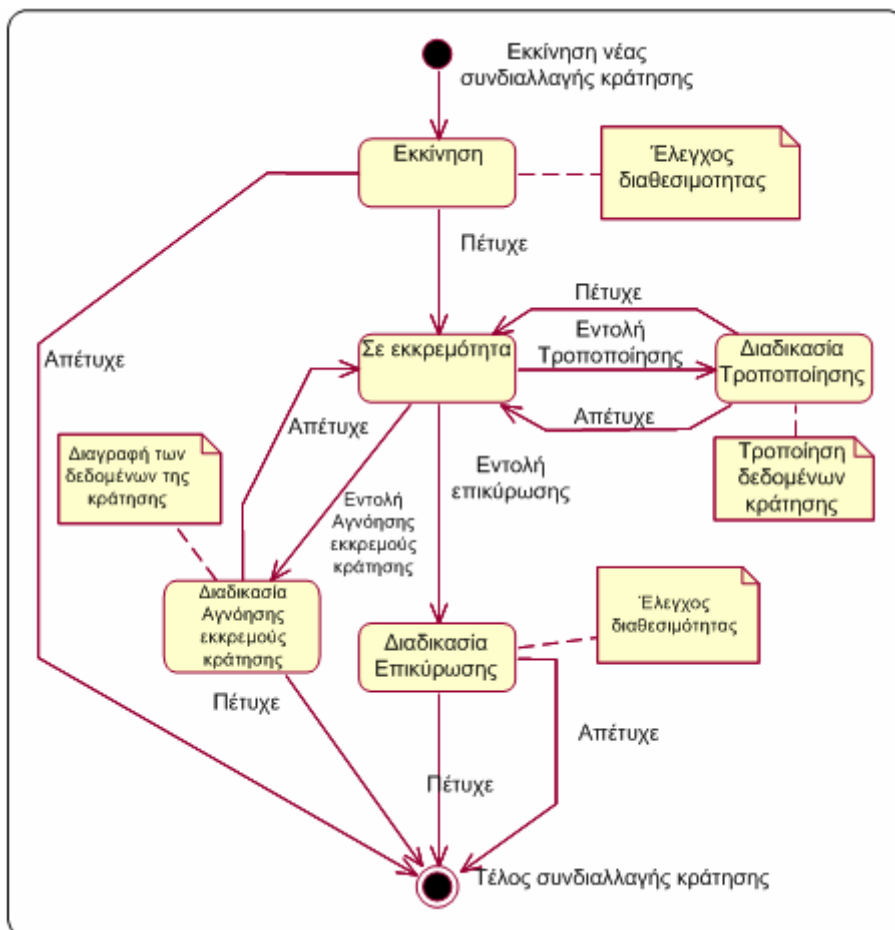
4.7.2.1 Περιγραφή της διαδικασίας δημιουργίας καινούργιας κράτησης

Η διαδικασία για την κράτηση δωματίων σε ξενοδοχεία μπορεί να περιλαμβάνει πολλά βήματα.

- Αρχικά ο χρήστης στέλνει ένα OTA_HotelResRQ XML μήνυμα με τα ξενοδοχεία και τα δωμάτια που θέλει να κρατήσει, θέτοντας στην ιδιότητα ResStatus την τιμή Initiate.
- Το σύστημα ελέγχει την διαθεσιμότητα των δωματίων που ζητούνται και εάν είναι διαθέσιμα τότε εκτελείται η κράτηση εισάγοντας στον πίνακα *Bookings* τα δωμάτια που κρατήθηκαν, θέτοντας όμως την τιμή *Pending* στο πεδίο *Status*. Στη συνέχεια επιστρέφεται στο χρήστη ένα OTA_HotelResRS μήνυμα με την ιδιότητα ResResponseType να έχει την τιμή Pending. Εάν τα δωμάτια που ζητήθηκαν δεν είναι διαθέσιμα τότε το σύστημα επιστρέφει σφάλμα μέσω του στοιχείου Errors.
- Στη συνέχεια ο χρήστης μπορεί να τροποποιήσει τα δεδομένα της κράτησης στέλνοντας ένα OTA_HotelResRQ XML μήνυμα με ResStatus = Modify, ή με ResStatus = Ignore εάν θέλει να ακυρώσει τη διαδικασία της κράτησης. Στην περίπτωση της τροποποίησης το σύστημα εξετάζει τη διαθεσιμότητα των νέων δεδομένων της κράτησης. Εάν είναι διαθέσιμα ενημερώνει τον πίνακα *Bookings* με τα καινούργια δωμάτια, θέτοντας και πάλι την τιμή *Pending* στο πεδίο *Status*. Στην περίπτωση που δεν είναι διαθέσιμα τα καινούργια δωμάτια επιστέφει σφάλμα μέσω του στοιχείου Errors.
- Εάν ο χρήστης αποφασίσει ότι είναι σίγουρος για την κράτηση πρέπει να στείλει ένα OTA_HotelResRQ XML μήνυμα με ResStatus = Commit για να επικυρώσει την κράτηση και να ολοκληρωθεί η διαδικασία. Γίνεται πάλι έλεγχος της διαθεσιμότητας των δωματίων που ζητούνται. Στην περίπτωση που είναι διαθέσιμα τότε:
 - Δημιουργείται το αναγνωριστικό HotelReservationID και αποθηκεύεται στον πίνακα *Bookings* της βάσης δεδομένων.
 - Επιστρέφεται ένα OTA_HotelResRS μήνυμα με ResResponseType = Committed, προσθέτοντας το αναγνωριστικό HotelReservationID που δημιουργήθηκε. Το αναγνωριστικό αυτό απαιτείται για μετέπειτα λειτουργίες ανάγνωσης, τροποποίησης ή ακύρωσης της κράτησης.

Εάν τα δωμάτια που ζητούνται δεν είναι διαθέσιμα τότε επιστρέφεται ένα OTA_HotelResRS μήνυμα με ResResponseType = Pending και τα κατάλληλα μηνύματα σφάλματος στο στοιχείο Errors.

Η διαδικασία αυτή δίνεται διαγραμματικά στο παρακάτω σχήμα (Εικόνα 36).



Εικόνα 36 Το διάγραμμα δραστηριοτήτων για την διαδικασία κράτησης δωματίων

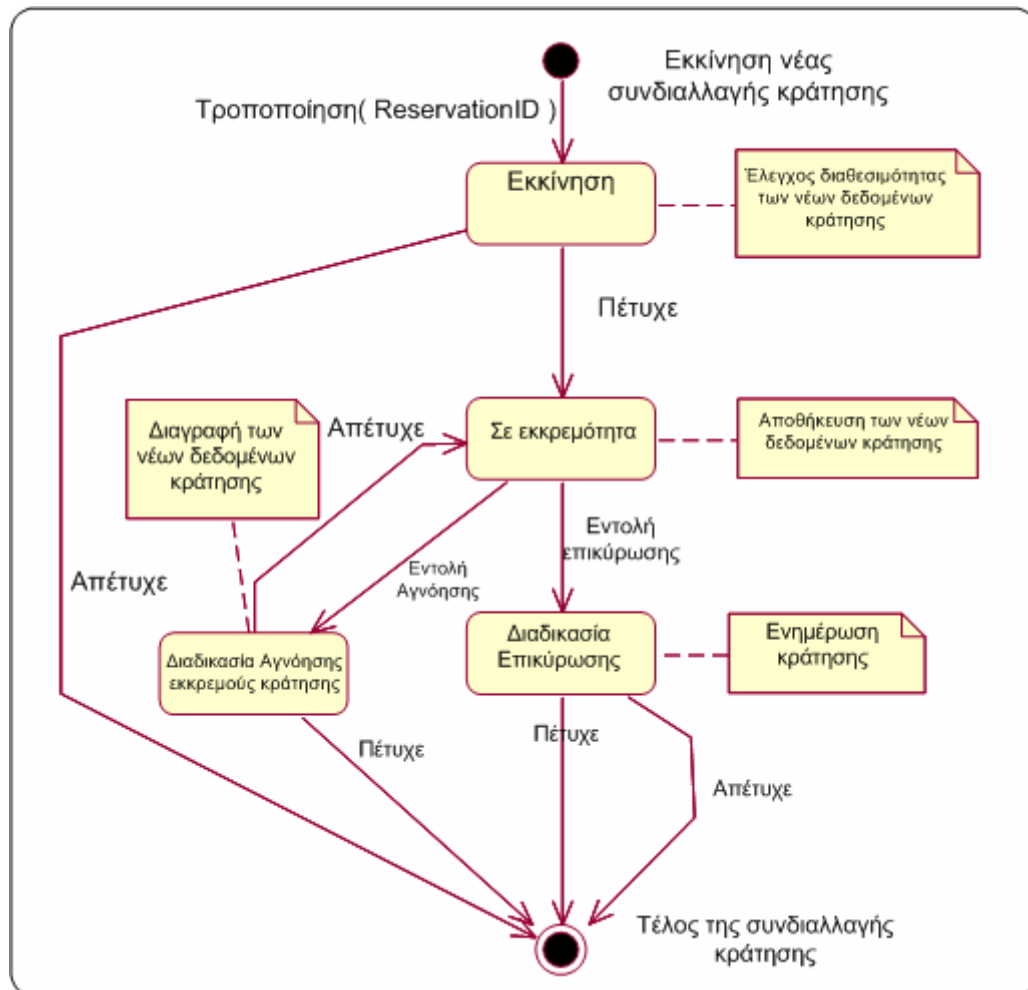
4.7.2.2 Περιγραφή της διαδικασίας τροποποίησης υπάρχουσας κράτησης

Στην περίπτωση που ο χρήστης επιθυμεί να τροποποιήσει μια υπάρχουσα κράτηση απαιτείται μια νέα σειρά ανταλλαγής μηνυμάτων. Τα μηνύματα πρέπει να συνοδεύονται με το HotelReservationID της υπάρχουσας κράτησης για να είναι σε θέση το σύστημα να αναγνωρίσει την εγγραφή για την οποία πρόκειται και να προχωρήσει στη διαδικασία της συνδιαλλαγής.

Τα βήματα που ακολουθούνται είναι τα εξής:

- Αρχικά ο χρήστης στέλνει ένα OTA_HotelResRQ XML μήνυμα με τα δεδομένα για την τροποποίηση της υπάρχουσας κράτησης, ορίζοντας ResStatus = Modify, και δίνοντας το HotelReservationID της υπάρχουσας κράτησης στο στοιχείο ResGlobalInfo.
- Κατόπιν το σύστημα ελέγχει εάν υπάρχει κράτηση με το συγκεκριμένο HotelReservationID. Εάν δεν υπάρχει κράτηση επιστρέφει ένα OTA_HotelResRS μήνυμα με το κατάλληλο σφάλμα στο στοιχείο Errors.
- Στη συνέχεια, εφόσον υπάρχει η κράτηση, το σύστημα ελέγχει τη διαθεσιμότητα του τροποποιημένου συνόλου δωματίων, κάνοντας χρήση του HotelReservationID ώστε να μην υπολογιστούν τα δωμάτια της υπάρχουσας κράτησης ως κλεισμένα. Εάν δεν είναι διαθέσιμα τα καινούργια δωμάτια επιστρέφει ένα OTA_HotelResRS μήνυμα με το κατάλληλο σφάλμα στο στοιχείο Errors. Εάν είναι διαθέσιμα τότε αποθηκεύει στη βάση δεδομένων τα καινούργια δωμάτια, θέτοντας όμως την τιμή *Pending* στο πεδίο *Status* και επιστρέφει στον χρήστη ένα OTA_HotelResRS μήνυμα με ResResponseType = Pending.
- Τώρα ο χρήστης είναι σε θέση να αγνοήσει την τροποποίηση, στέλνοντας ένα OTA_HotelResRQ XML μήνυμα με ResStatus = Ignore, να τροποποιήσει εκ νέου την κράτηση στέλνοντας ένα OTA_HotelResRQ XML μήνυμα με ResStatus = Modify, ή τέλος να επικυρώσει την τροποποιημένη κράτηση στέλνοντας ένα OTA_HotelResRQ XML μήνυμα με ResStatus = Commit.

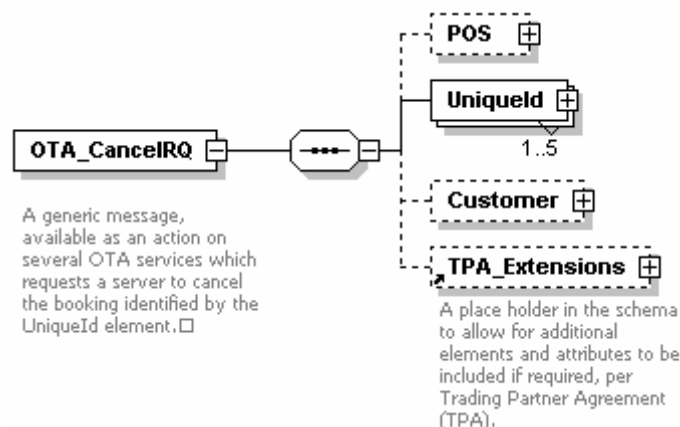
Στο παρακάτω σχήμα (Εικόνα 37) δίνεται το διάγραμμα δραστηριοτήτων της διαδικασίας τροποποίησης υπάρχουσας κράτησης.



Εικόνα 37 Το διάγραμμα δραστηριοτήτων της διαδικασίας τροποποίησης υπάρχουσας κράτησης

4.7.2.3 Περιγραφή της διαδικασίας ακύρωσης υπάρχουσας κράτησης

Το XML Schema του OTA που χρησιμοποιείται για την αίτηση ακύρωσης μιας υπάρχουσας κράτησης είναι κοινό σε όλους τους τομείς της τουριστικής βιομηχανίας που καλύπτει το OTA και είναι το OTA_CancelRQ το οποίο φαίνεται στο παρακάτω σχήμα (Εικόνα 38).



Εικόνα 38 Το XML Schema του OTA για την αίτηση ακύρωσης κράτησης

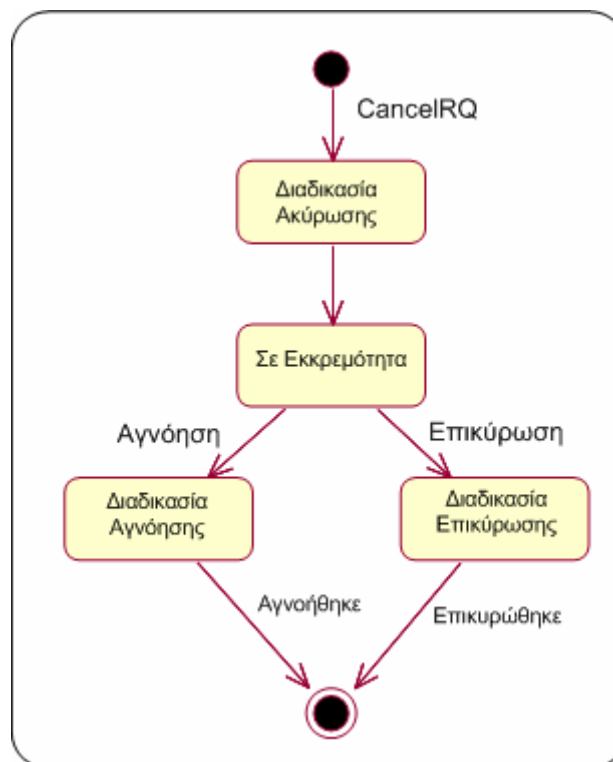
Το στοιχείο OTA_CancelRQ περιλαμβάνει την ιδιότητα CancelType η οποία δείχνει την ενέργεια που πρόκειται να εκτελεστεί στο συγκεκριμένο βήμα της συνδιαλλαγής.

Τυπικές τιμές που μπορεί να πάρει είναι οι Initiate, Ignore, Commit

Η διαδικασία που ακολουθείται για την ακύρωση μιας κράτησης είναι η εξής:

- Αρχικά ο χρήστης στέλνει ένα OTA_CancelRQ XML μήνυμα με CancelType = Initiate, UniqueId = HotelReservationID της κράτησης και πιθανόν το όνομα του χρήστη.
- Το σύστημα ελέγχει αρχικά την ύπαρξη της κράτησης στη βάση δεδομένων
- Επιστρέφεται στο χρήστη ένα OTA_CancelRS XML μήνυμα με CancelType = Pending, ενημερώνοντας τον για τυχόν πολιτικές ακύρωσης
- Στη συνέχεια ο χρήστης μπορεί είτε να επικυρώσει την ακύρωση στέλνοντας ένα OTA_CancelRQ XML μήνυμα με CancelType = Commit ή να αγνοήσει την ακύρωση στέλνοντας ένα OTA_CancelRQ XML μήνυμα με CancelType = Ignore

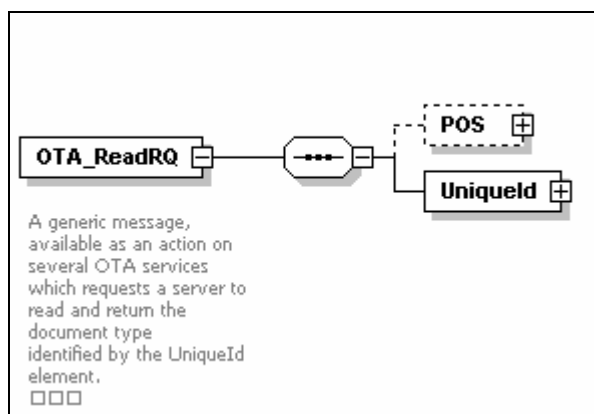
Στο παρακάτω σχήμα (Εικόνα 39) δίνεται το διάγραμμα δραστηριοτήτων της διαδικασίας ακύρωσης μιας υπάρχουσας κράτησης



Εικόνα 39 Το διάγραμμα δραστηριοτήτων της διαδικασίας ακύρωσης μιας υπάρχουσας κράτησης

4.7.2.4 Περιγραφή της διαδικασίας ανάκτησης υπάρχουσας κράτησης

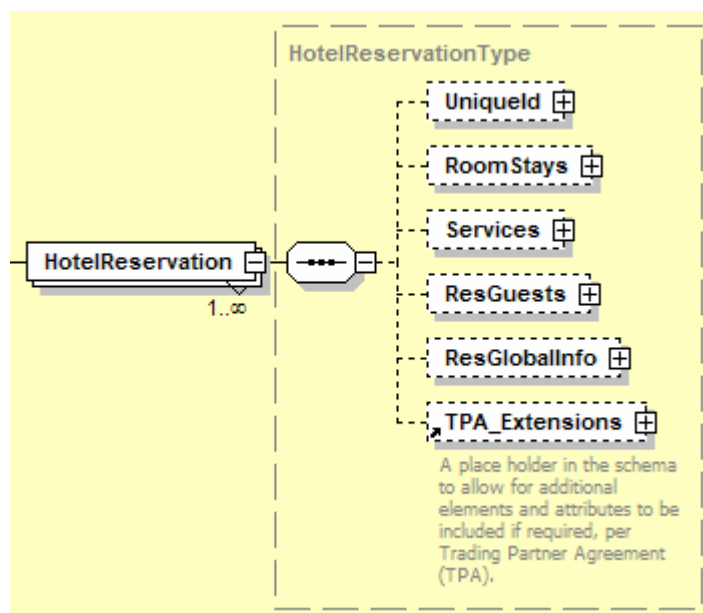
Πριν ο χρήστης προβεί στην τροποποίηση ή την ακύρωση μιας υπάρχουσας κράτησης είναι σκόπιμο να προηγηθεί μια ανάκτηση των δεδομένων της κράτησης, ώστε να έχει μια επισκόπηση των δεδομένων της υπάρχουσας κράτησης. Το XML Schema του OTA που χρησιμοποιείται για την αίτηση ανάκτησης μιας υπάρχουσας κράτησης είναι κοινό σε όλους τους τομείς της τουριστικής βιομηχανίας που καλύπτει το OTA όπου απαιτείται η ανάγνωση μιας εγγραφής, και είναι το **OTA_ReadRQ** το οποίο φαίνεται στο παρακάτω σχήμα (Εικόνα 40).



Εικόνα 40 Το XML Schema του OTA για το γενικό μήνυμα ανάγνωσης

Το στοιχείο `OTA_ReadRQ` που είναι και το αρχικό στοιχείο του μηνύματος περιλαμβάνει το στοιχείο `UniqueId` με το οποίο δίνεται το μοναδικό αναγνωριστικό της κράτησης (`HotelReservationID`) που πρέπει να ανακτηθεί από τη βάση δεδομένων.

Το σύστημα (ebXML OTA Server) ανακτά από τη βάση δεδομένων των κρατήσεων όλες τις πληροφορίες που αντιστοιχούν στην κράτηση που ζητείται, και τα επιστρέφει σε ένα XML έγγραφο που υπακούει στο XML Schema `OTA_HotelReservation`, η δομή του οποίου δίνεται στο παρακάτω σχήμα (Εικόνα 41).



Εικόνα 41 Το XML Schema του OTA που περιγράφει τις κρατήσεις ξενοδοχείων

Τα στοιχεία που υπάρχουν στο XML έγγραφο είναι τα παρακάτω:

- UniqueId – περιέχει το μοναδικό αναγνωριστικό της κράτησης
- RoomStays – περιέχει πληροφορίες για τα δωμάτια που έχουν κρατηθεί (RoomType), τα RatePlans, τις τιμές (RoomRates), την διάρκεια της κράτησης (TimeSpan), καθώς και το συνολικό κόστος της κράτησης (Total)
- ResGlobalInfo – περιέχει το αναγνωριστικό της κράτησης (HotelReservationID). Το αναγνωριστικό αυτό είναι διαφορετικό από το UniqueId. Το UniqueId χρησιμοποιείται κατά τη διάρκεια της κράτησης για να αναγνωρίζεται η κράτηση στο πίνακα bookings, ενώ το HotelReservationID δημιουργείται κατά την επικύρωση της κράτησης και χρησιμοποιείται στη συνέχεια από τον χρήστη για να μπορέσει να διαχειριστεί την κράτηση του. Δηλαδή το HotelReservationID χρησιμοποιείται σαν UniqueId στην αίτηση για ανάκτηση της υπάρχουσας κράτησης καθώς και στις υπόλοιπες λειτουργίες διαχείρισης της κράτησης (τροποποίηση και ακύρωση).

4.8 Υλοποίηση της υπηρεσίας διαχείρισης του προφίλ

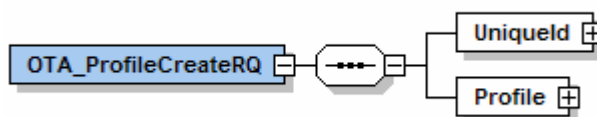
Στην ενότητα αυτή περιγράφονται τα μηνύματα που ανταλλάσσονται για την διαχείριση του προφίλ χρήστη, καθώς και τα κυριότερα τμήματα των XML σχημάτων που αφορούν στο προφίλ. Η διαχείριση του προφίλ περιλαμβάνει τέσσερις διαφορετικές λειτουργίες:

- Δημιουργία
- Ανάκτηση
- Διαγραφή και
- Τροποποίηση

Στη συνέχεια θα γίνει περιγραφή των παραπάνω λειτουργιών.

4.8.1 Δημιουργία νέου προφίλ

Η δημιουργία ενός προφίλ εκτελείται με την αποστολή στον ebXML OTA Server ενός μηνύματος OTA_ProfileCreateRQ. Το XML Schema του μηνύματος αυτού δίνεται στο παρακάτω σχήμα (Εικόνα 42).



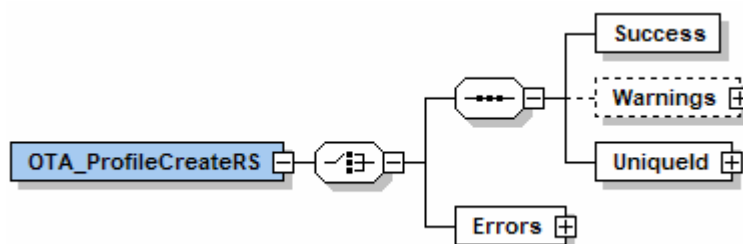
Εικόνα 42 Το XML Schema του OTA για τα μηνύματα δημιουργίας νέου προφίλ

Το αρχικό στοιχείο του μηνύματος, το OTA_ProfileCreateRQ έχει δύο υποστοιχεία:

- Το UniqueId – επιτρέπει να οριστεί ένα UniqueId που θα συσχετιστεί με το προφίλ και θα αποτελεί το μοναδικό χαρακτηριστικό του. Όλες οι μετέπειτα λειτουργίες στο υπάρχον προφίλ θα χρησιμοποιούν το UniqueId για να το αναγνωρίσουν στην βάση δεδομένων. Εάν δεν οριστεί από τον χρήστη το UniqueId τότε θα το δημιουργήσει το σύστημα.
- Το Profile – αυτό το στοιχείο περιέχει όλη την πληροφορία σχετικά με το προφίλ χρήστη, όπως προσωπικά στοιχεία, προτιμήσεις κτλ.

Τα δύο στοιχεία που προαναφέρθηκαν (UniqueId και Profile) αποθηκεύονται στην σχεσιακή βάση δεδομένων του ebXML OTA Server χρησιμοποιώντας την Java βιβλιοθήκη (Java API) που κατασκευάστηκε αυτόματα (όπως και η σχεσιακή βάση δεδομένων) με την χρήση της εφαρμογής XML → DB+Middleware. Όπως αναφέρθηκε και στην ενότητα 4.5.2, η σχεσιακή βάση δεδομένων καλύπτει ολόκληρο το XML Schema του προφίλ που περιγράφεται στις προδιαγραφές του OTA.

Στη συνέχεια επιστρέφεται το μήνυμα απόκρισης στην αίτηση, το οποίο υπακούει στο XML Schema OTA_ProfileCreateRS. Το XML Schema αυτό δίνεται στο παρακάτω σχήμα (Εικόνα 43).

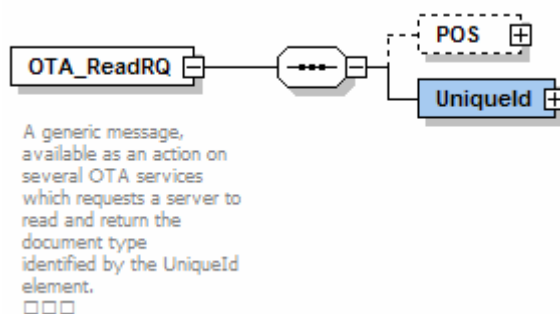


Εικόνα 43 Το XML Schema του OTA για την απόκριση στην αίτηση για δημιουργία νέου προφίλ

Σύμφωνα με το παραπάνω XML Schema, μετά από μια επιτυχημένη δημιουργία νέου προφίλ, το μήνυμα απόκρισης περιέχει το στοιχείο Success που δηλώνει την επιτυχή έκβαση της διαδικασίας, και το στοιχείο UniqueId το οποίο επιστρέφει το μοναδικό αναγνωριστικό που συσχετίστηκε με το προφίλ.

4.8.2 Ανάκτηση προφίλ από την βάση δεδομένων του ebXML OTA Server

Η αίτηση για ανάκτηση ενός προφίλ από την βάση δεδομένων χρησιμοποιεί το γενικό μήνυμα ανάγνωσης OTA_ReadRQ που περιγράφει ο OTA. Το XML Schema του μηνύματος δίνεται στο παρακάτω σχήμα (Εικόνα 44).

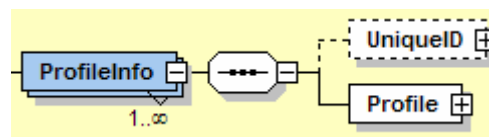


Εικόνα 44 Το XML Schema του OTA για την αίτηση ανάγνωσης

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, το μήνυμα αυτό χρησιμοποιείται και για την ανάγνωση μιας κράτησης από την αντίστοιχη βάση δεδομένων. Το σύστημα αναγνωρίζει ότι πρόκειται για ανάγνωση προφίλ (στην συγκεκριμένη περίπτωση του προφίλ, και για κράτηση στην περίπτωση που γίνεται αίτηση για ανάκτηση μιας κράτησης) τόσο από τη τιμή του πεδίου Service που περιέχεται στο ebXML μήνυμα (το οποίο στην περίπτωση του προφίλ έχει την τιμή «Profile» ενώ στην περίπτωση

της κράτησης έχει την τιμή «HotelBooking»), όσο και από την ιδιότητα «Type» του στοιχείου UniqueId. Η ιδιότητα αυτή στην περίπτωση του προφίλ παίρνει την τιμή «Profile», ενώ στην περίπτωση της κράτησης παίρνει την τιμή «Reservation».

Εάν το προφίλ που έχει το ζητούμενο UniqueId υπάρχει στην βάση δεδομένων, τότε αυτό ανακτάται από την βάση δεδομένων χρησιμοποιώντας την Java βιβλιοθήκη (Java API) και επιστρέφεται στον χρήστη με ένα XML έγγραφο που υπακούει στο XML Schema «ProfileInfo» το οποίο δίνεται στο παρακάτω σχήμα (Εικόνα 45).



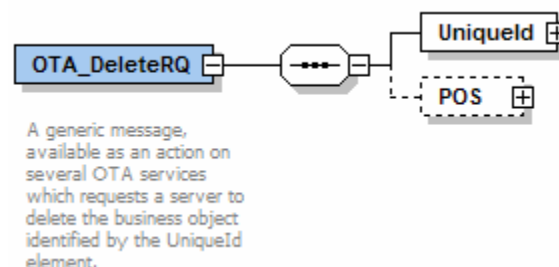
Εικόνα 45 Το XML Schema του OTA για το ProfileInfo

Όπως φαίνεται και στην εικόνα, το στοιχείο ProfileInfo περιέχει τα στοιχεία:

- UniqueId – το οποίο περιέχει το μοναδικό αναγνωριστικό του προφίλ και
- Profile - το οποίο περιέχει όλες τις πληροφορίες που περιέχονται στο προφίλ.

4.8.3 Διαγραφή προφίλ από την βάση δεδομένων του ebXML OTA Server

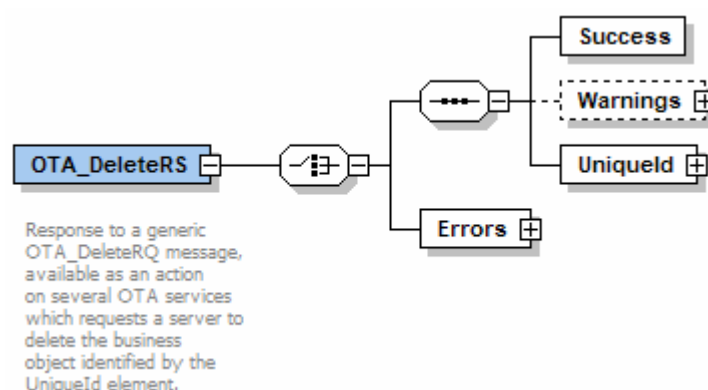
Για την διαγραφή ενός υπάρχοντος προφίλ χρησιμοποιείται το γενικό μήνυμα OTA_DeleteRQ, το XML Schema του οποίου δίνεται στο παρακάτω σχήμα (Εικόνα 46).



Εικόνα 46 Το XML Schema του OTA για την αίτηση διαγραφής

Το στοιχείο OTA_DeleteRQ περιέχει το στοιχείο UniqueId με το οποίο ορίζουμε ποιο προφίλ να διαγραφεί. Εάν υπάρχει στη βάση δεδομένων προφίλ με UniqueId ίσο με αυτό που δόθηκε σαν είσοδος, τότε το συγκεκριμένο προφίλ διαγράφεται από την βάση δεδομένων.

Η απόκριση στη αίτηση για διαγραφή γίνεται μέσω του μηνύματος OTA_DeleteRS, το XML Schema του οποίου δίνεται στο επόμενο σχήμα (Εικόνα 47).



Εικόνα 47 Το XML Schema του OTA για την απόκριση στην αίτηση διαγραφής

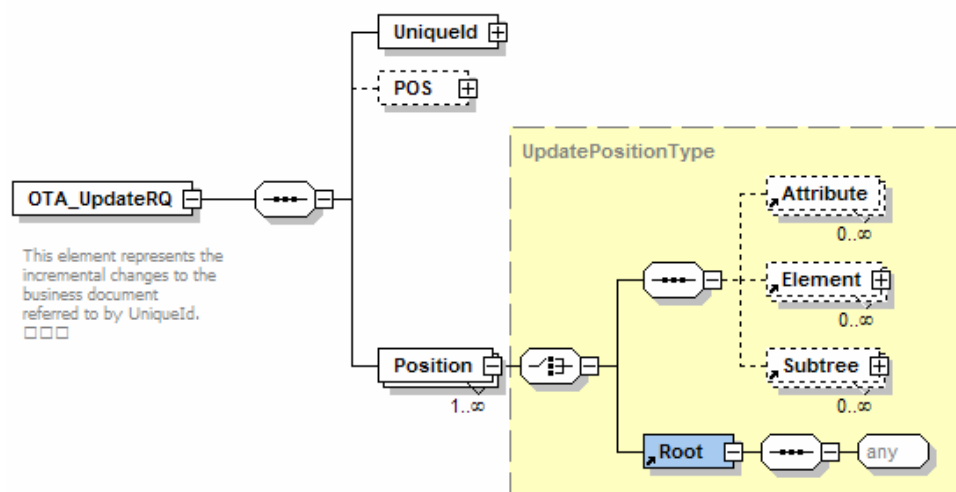
Εάν το συγκεκριμένο προφίλ δεν υπάρχει στην βάση δεδομένων τότε το μήνυμα περιέχει το στοιχείο Errors το οποίο καταγράφει το σφάλμα που προέκυψε. Στην αντίθετη περίπτωση το μήνυμα περιέχει το στοιχείο Success το οποίο υποδηλώνει την επιτυχή έκβαση της διαγραφής και το στοιχείο UniqueId το οποίο αναφέρεται στο UniqueId του προφίλ που διαγράφηκε.

4.8.4 Τροποποίηση προφίλ

Η τροποποίηση ενός υπάρχοντος προφίλ πραγματοποιείται σε δύο στάδια:

- Διαγραφή του υπάρχοντος προφίλ
- Εγγραφή του νέου προφίλ

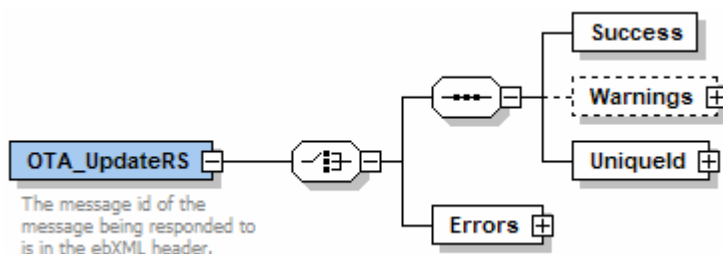
Το XML Schema που περιγράφει το μήνυμα για τροποποίηση ενός υπάρχοντος προφίλ δίνεται στο ακόλουθο σχήμα (Εικόνα 48).



Εικόνα 48 Το XML Schema του OTA για την αίτηση τροποποίησης

Το στοιχείο UniqueId προσδιορίζει το προφίλ που θέλουμε να τροποποιήσουμε, ενώ το καινούργιο προφίλ μπαίνει στο XML έγγραφο σαν παιδί του στοιχείο Root. Το καινούργιο προφίλ θα έχει το ίδιο UniqueId με το προηγούμενο προφίλ.

Η απόκριση στην αίτηση για τροποποίηση ενός προφίλ περιγράφεται με το XML Schema OTA_UpdateRS, το οποίο δίνεται στο ακόλουθο σχήμα (Εικόνα 49).



Εικόνα 49 Το XML Schema του OTA για την απόκριση στην αίτηση τροποποίησης

Μετά από μια επιτυχή τροποποίηση το μήνυμα θα περιλαμβάνει το στοιχείο Success το οποίο υποδηλώνει την επιτυχή έκβαση της διαδικασίας, και το στοιχείο UniqueId το οποίο θα έχει την ίδια τιμή με εκείνη που δόθηκε στο μήνυμα της αίτησης για τροποποίηση του προφίλ.

Στην περίπτωση που η διαδικασία αποτύχει, τότε το σφάλμα που προέκυψε καταγράφεται στο στοιχείο Errors.

4.9 Ανακεφαλαίωση

Στο κεφάλαιο αυτό έγινε αναλυτική περιγραφή της αρχιτεκτονικής του συστήματος ebXML OTA Server και των επιμέρους τμημάτων από τα οποία αποτελείται. Περιγράφηκε η χρήση της ebMS για την αποστολή και λήψη ebXML μηνυμάτων, οι διαχειριστές υπηρεσιών που αναλαμβάνουν την λειτουργία των OTA Web Services, έγινε αναλυτική περιγραφή των βάσεων δεδομένων που χρησιμοποιούνται για την διαχείριση των κρατήσεων, των προφίλ και των OTA Sessions. Στη συνέχεια έγινε αναλυτική περιγραφή της υλοποίησης των Web Services, παρουσιάστηκαν τα σχετικά XML Schemata, καθώς και στιγμιότυπα από πραγματικά XML έγγραφα. Συνοπτικά οι Web Services που υλοποιήθηκαν είναι οι παρακάτω:

- Session – υπηρεσία διαχείρισης των OTA Sessions. Περιλαμβάνει την δημιουργία ενός καινούργιου Session και το κλείσιμο ενός υπάρχοντος Session.
- HotelBooking – περιλαμβάνει τις υπηρεσίες που σχετίζονται με τα ξενοδοχεία όπως:
 - Αναζήτηση ξενοδοχείων
 - Αναζήτηση διαθεσιμότητας και εναλλακτικής διαθεσιμότητας (με χρήση του προφίλ χρήστη ή όχι)
 - Διαχείριση κρατήσεων ξενοδοχείων
 - Δημιουργία καινούργιας κράτησης
 - Ανάκτηση υπάρχουσας κράτησης
 - Τροποποίηση υπάρχουσας κράτησης
 - Ακύρωση υπάρχουσας κράτησης
- Διαχείριση των προφίλ χρηστών
 - Δημιουργία
 - Ανάκτηση
 - Διαγραφή
 - Τροποποίηση (ενημέρωση)

Όπως έχει ήδη αναφερθεί, για να μπορέσει ένας υποψήφιος πελάτης να χρησιμοποιήσει τις OTA Web Services που παρέχονται από τον ebXML OTA Server, πρέπει να χρησιμοποιήσει ένα CPA. Για την κατασκευή του CPA μπορεί να χρησιμοποιηθεί η εφαρμογή CPA Editor που περιγράφεται στο επόμενο κεφάλαιο.

Κεφάλαιο 5

Σχεδιασμός και Υλοποίηση του CPA Editor

5.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφονται οι προδιαγραφές της ebXML για την συμφωνία του πρωτοκόλλου συνεργασίας (Collaboration Protocol Agreement, CPA), καθώς και η εφαρμογή που σχεδιάστηκε και αναπτύχθηκε στα πλαίσια της παρούσας εργασίας για την κατασκευή CPA.

Όπως αναφέρεται στις προδιαγραφές της ebXML και του OTA, για να εκτελεστεί μια επιχειρηματική συνεργασία μεταξύ διαφορετικών συστημάτων απαιτείται η ύπαρξη ενός CPA, που να καθορίζει τους κανόνες της μεταξύ τους συνεργασίας. Όπως αναφέρθηκε στο κεφάλαιο 3, το CPA είναι ένα XML έγγραφο το οποίο χρησιμοποιείται από κοινού από τα συνεργαζόμενα συστήματα, και περιγράφει την συμφωνία του πρωτοκόλλου συνεργασίας. Καθορίζει δηλαδή τους κανόνες που διέπουν την συνεργασία, τα τεχνικά χαρακτηριστικά της επικοινωνίας μεταξύ των διαφορετικών συστημάτων, καθώς και τους διακριτούς ρόλους που έχει ο κάθε συνεργαζόμενος στα πλαίσια της συγκεκριμένης συνεργασίας.

Όπως επισημαίνεται στις προδιαγραφές της ebXML, το CPA μπορεί να κατασκευαστεί από την ένωση των προφίλ πρωτοκόλλου συνεργασίας (CPP) των συνεργατών, ή απλούστερα χρησιμοποιώντας έτοιμα πρότυπα, στα οποία οι συνεργάτες συμπληρώνουν τα δικά τους στοιχεία.

Η πρώτη μέθοδος αποτελεί πιο σύνθετη και κομψή προσέγγιση στην κατασκευή του CPA (σε σχέση με την δεύτερη μέθοδο), δίνοντας τη δυνατότητα αυτοματοποίησης της διαδικασίας κατασκευής CPA. Μέχρι σήμερα δεν υπάρχει μια καθιερωμένη διαδικασία για την υλοποίηση αυτής της μεθόδου. Εντούτοις γίνονται ερευνητικές προσπάθειες προς την κατεύθυνση αυτή.

Η δεύτερη μέθοδος είναι η πιο διαδεδομένη μέχρι σήμερα, καθώς επιτρέπει την άμεση κατασκευή του CPA, παρέχοντας στους συνεργάτες την δυνατότητα εποπτείας της διαδικασίας κατασκευής του.

Στην παρούσα εργασία χρησιμοποιήθηκε η δεύτερη μέθοδος. Σχεδιάστηκε και αναπτύχθηκε μια πρότυπη Web εφαρμογή που αναλαμβάνει την κατασκευή CPA και ονομάζεται CPA Editor. Η συγκεκριμένη εφαρμογή χρησιμοποιεί μια φόρμα δεδομένων την οποία ο πελάτης συμπληρώνει δίνοντας πληροφορίες που αφορούν το δικό του ebMS/OTA σύστημα, όπως είναι η διεύθυνση της εφαρμογής του στο διαδίκτυο (URI), το όνομα που επιθυμεί να αποδώσει στο CPA και τις υπηρεσίες που επιθυμεί να χρησιμοποιήσει, επιλέγοντας από αυτές που παρέχει ο ebXML OTA Server, δηλαδή τις υπηρεσίες HotelBooking και Profile, καθώς και ποιες ενέργειες (Actions) επιθυμεί να χρησιμοποιεί στα πλαίσια των παραπάνω υπηρεσιών (π.χ. OTA_ProfileCreateRQ για τη δημιουργία προφίλ). Η υπηρεσία Session δεν δίνεται σαν επιλογή καθώς η ύπαρξή της απαιτείται για την λειτουργία του συστήματος και την εκτέλεση των OTA Web Services. Τα υπόλοιπα δεδομένα που πρέπει (ή μπορεί) να περιέχει το CPA έχουν προεπιλεγμένες τιμές και δεν απεικονίζονται στη φόρμα δεδομένων. Στο παράρτημα Α δίνεται ένα παράδειγμα CPA που κατασκευάστηκε με την εφαρμογή CPA Editor.

Στη συνέχεια δίνεται λεπτομερέστερη περιγραφή των προδιαγραφών της ebXML για το CPA.

5.2 Περιγραφή των προδιαγραφών¹⁸ της ebXML για το CPA

Η συμφωνία του πρωτοκόλλου συνεργασίας (Collaboration Protocol Agreement, CPA), καθορίζει τους όρους στους οποίους οι συνεργάτες πρέπει να συμφωνήσουν

¹⁸ Η περιγραφή που γίνεται στο κεφάλαιο αυτό παρέχει γενικές πληροφορίες για το CPA, περιγράφοντας κυρίως στοιχεία και χαρακτηριστικά που βρίσκονται στα υψηλότερα επίπεδα της ιεραρχίας του XML σχήματος. Για πιο λεπτομερή περιγραφή των προδιαγραφών του CPA ο αναγνώστης καλείται να ανατρέξει στις επίσημες προδιαγραφές της ebXML. Η έκδοση των προδιαγραφών που χρησιμοποιήθηκε είναι η draft-cpp-cpa-015. Η συγκεκριμένη έκδοση επιλέχθηκε επειδή είναι αυτή που υποστηρίζεται από την υλοποίηση ebMS που χρησιμοποιήθηκε στην παρούσα εργασία.

προκειμένου να είναι σε θέση να συνεργαστούν ηλεκτρονικά στα πλαίσια της συγκεκριμένης συμφωνίας. Παρακάτω δίνεται η γενική δομή ενός CPA:

```
<?xml version="1.0" encoding="UTF-8"?>
<CollaborationProtocolAgreement xmlns="http://www.ebxml.org/namespaces/tradePartner"
xmlns:bpm="http://www.ebxml.org/namespaces/businessProcess" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xlink="http://www.w3.org/1999/xlink"
cpaid="YoursAndMyCPA"
version="1.2">
  <Status value="proposed"/>
  <Start>1988-04-07T18:39:09</Start>
  <End>1990-04-07T18:40:00</End>
  <!--ConversationConstraints MAY appear 0 or 1 times-->
  <ConversationConstraints invocationLimit="100" concurrentConversations="4"/>
  <PartyInfo>
    ...
  </PartyInfo>
  <PartyInfo>
    ...
  </PartyInfo>
  <!--ds:signature MAY appear 0 or more times-->
</CollaborationProtocolAgreement>
```

Το στοιχείο «CollaborationProtocolAgreement» είναι το αρχικό στοιχείο-ρίζα ενός CPA. Το στοιχείο αυτό περιλαμβάνει τα παρακάτω χαρακτηριστικά και υποστοιχεία:

- **cpaid:** μοναδικό αναγνωριστικό του CPA εγγράφου. Η τιμή που παίρνει πρέπει να χρησιμοποιείται και από τους δύο συνεργάτες, και μπορεί να χρησιμοποιηθεί σαν η τιμή του αντίστοιχου στοιχείου CPAId που βρίσκεται στην επικεφαλίδα του ebXML μηνύματος.
- **Version:** δηλώνει την έκδοση του CPA καθώς αυτό τροποποιείται κατά τη διάρκεια της διαπραγμάτευσής του.
- **Status:** δηλώνει την κατάσταση στην οποία βρίσκεται η κατασκευή/διαπραγμάτευση του CPA. Οι τιμές που μπορεί να πάρει είναι:
 - **Proposed:** δηλώνει ότι το CPA είναι υπό διαπραγμάτευση.
 - **Agreed:** δηλώνει ότι το CPA έγινε δεκτό και από τους δύο συνεργάτες.
 - **Signed:** δηλώνει ότι υπογράφηκε από τους συνεργάτες (π.χ. με ψηφιακή υπογραφή).
- **Start:** καταγράφει την ημερομηνία έναρξης της ισχύος του CPA.
- **End:** καταγράφει την ημερομηνία λήξης της ισχύος του CPA.
- **ConversationConstraints:** θέτει περιορισμούς στον αριθμό των συνομιλιών που εκτελούνται υπό το συγκεκριμένο CPA. Έχει τα παρακάτω χαρακτηριστικά:

- invocationLimit: καθορίζει τον μέγιστο αριθμό συνομιλιών που επιτρέπεται να εκτελεστούν υπό το συγκεκριμένο CPA.
- concurrentConversations: καθορίζει τον μέγιστο αριθμό συνομιλιών που επιτρέπεται να «τρέχουν» ταυτόχρονα υπό το συγκεκριμένο CPA.
- PartyInfo: το στοιχείο αυτό περιγράφει τα χαρακτηριστικά κάθε συνεργάτη στα πλαίσια της συγκεκριμένης συνεργασίας. Το CPA πρέπει να περιέχει ένα στοιχείο PartyInfo για κάθε συνεργάτη. Περισσότερες λεπτομέρειες για το στοιχείο PartyInfo δίνονται παρακάτω.

5.2.1 Περιγραφή του στοιχείου PartyInfo

Το στοιχείο PartyInfo περιγράφει του όρους υπό τους οποίους κάθε συνεργάτης εμπλέκεται στις επιχειρησιακές συνεργασίες που περιγράφονται στα έγγραφα περιγραφής επιχειρησιακών διαδικασιών (ebBPSS). Η γενική δομή του στοιχείου PartyInfo δίνεται παρακάτω:

```
<?xml version="1.0" encoding="UTF-8"?>
<PartyInfo>
  <PartyId type="..."> <!--one or more-->
  ...
  </PartyId>
  <PartyRef xlink:type="...", xlink:href="...">
  <CollaborationRole> <!--one or more-->
  ...
  </CollaborationRole>
  <Certificate> <!--one or more-->
  ...
  </Certificate>
  <DeliveryChannel> <!--one or more-->
  ...
  </DeliveryChannel>
  <Transport> <!--one or more-->
  ...
  </Transport>
  <DocExchange> <!--one or more-->
  ...
  </DocExchange>
</PartyInfo>
```

Το στοιχείο PartyInfo αποτελείται από τα παρακάτω υποστοιχεία:

- PartyId: παρέχει ένα αναγνωριστικό για τον συνεργάτη. Το στοιχείο παίρνει τιμές τύπου string, ενώ περιλαμβάνει και την ιδιότητα «type» η οποία

καθορίζει την εμβέλεια (scope) ή το πεδίο τιμών (namespace) της τιμής του στοιχείου. Το παρακάτω παράδειγμα δείχνει δύο περιπτώσεις χρήσης του στοιχείου:

```
<PartyId type="uriReference">urn:duns:123456789</PartyId>
<PartyId type="uriReference">urn:www.example.com</PartyId>
```

- PartyRef: παρέχει σύνδεση, σε μορφή URI, σε επιπρόσθετες πληροφορίες σχετικά με τον συνεργάτη, για παράδειγμα το URL στο οποίο βρίσκονται αυτές οι πληροφορίες. Για παράδειγμα:

```
<PartyRef
  xlink:type="simple":href=http://example2.com/ourInfo.xml
  type="uri-reference"/>
```

- CollaborationRole: συσχετίζει τον συνεργάτη με έναν συγκεκριμένο ρόλο στην επιχειρησιακή συνεργασία που περιγράφεται στο έγγραφο περιγραφής επιχειρησιακών διαδικασιών (ebBPSS¹⁹). Γενικά, μια επιχειρησιακή διαδικασία περιγράφεται με ρόλους όπως «αγοραστής» και «πωλητής» ή «πελάτης» και «εξυπηρετητής». Η συσχέτιση μεταξύ του συνεργάτη και του ρόλου που καλείται να «παίξει» στα πλαίσια μιας επιχειρησιακής διαδικασίας καθορίζεται τόσο στο CPP όσο και στο CPA. Στη συνέχεια θα δοθεί πιο λεπτομερής περιγραφή του στοιχείου CollaborationRole.
- Certificate: περιγράφει πληροφορίες πιστοποίησης. Ένα ή περισσότερα στοιχεία Certificate μπορούν να χρησιμοποιηθούν για διάφορες λειτουργίες ασφάλειας. Ένα παράδειγμα του στοιχείου Certificate είναι το παρακάτω:

```
<tp:Certificate tp:certId="CompanyA_SigningCert">
  <ds:KeyInfo>
    <ds:KeyName>
      pkcs11://tokenName/tomcat
    </ds:KeyName>
  </ds:KeyInfo>
</tp:Certificate>
```

- DeliveryChannel: αποτελεί έναν συνδυασμό των στοιχείων Transport και DocExchange (που περιγράφονται παρακάτω) και περιγράφει τα

¹⁹ Η αναλυτική περιγραφή των προδιαγραφών της ebXML που σχετίζονται με το BPSS ξεφεύγει από τα πλαίσια της παρούσας εργασίας. Για λεπτομερή περιγραφή των προδιαγραφών του BPSS ο αναγνώστης καλείται να ανατρέξει στις επίσημες προδιαγραφές της ebXML. Εντούτοις, στο σχετικό παράρτημα δίνεται το XML έγγραφο της πρότυπης επιχειρησιακής διαδικασίας (OTASimpleTransaction) που σχεδιάστηκε για τις ανάγκες της εργασίας.

χαρακτηριστικά της επικοινωνίας μηνυμάτων του συνεργάτη. Περιλαμβάνει επίσης παραμέτρους σχετικά με τις επιχειρησιακές συνδιαλλαγές και την επικοινωνία των μηνυμάτων. Ένα παράδειγμα του στοιχείου DeliveryChannel είναι το παρακάτω:

```
<tp:DeliveryChannel tp:channelId="TYPEA" tp:transportId="HTTPS01"
tp:docExchangeId="TYPEA">
  <tp:BusinessTransactionCharacteristics
    tp:isNonRepudiationRequired="false"
    tp:isNonRepudiationReceiptRequired="false"
    tp:isSecureTransportRequired="true" tp:isConfidential="none"
    tp:isAuthenticated="none" tp:isAuthorizationRequired="false"
    tp:timeToPerform="P1D"/>
  <tp:MessagingCharacteristics tp:syncReplyMode="responseOnly"
    tp:ackRequested="never" tp:ackSignatureRequested="never"
    tp:duplicateElimination="never"/>
</tp:DeliveryChannel>
```

- Transport: καθορίζει τις δυνατότητες δικτυακής επικοινωνίας του συνεργάτη, όπως είναι το πρωτόκολλο μεταφοράς, π.χ. HTTP, και το σημείο στο οποίο δέχεται μηνύματα (endpoint), π.χ. https://localhost:8443/ebXML/msh. Ένα παράδειγμα του στοιχείου Transport είναι το παρακάτω:

```
<tp:Transport tp:transportId="HTTPS01">
  <tp:TransportSender>
    <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
    <tp:TransportClientSecurity>
      ...
    </tp:TransportClientSecurity>
  </tp:TransportSender>
  <tp:TransportReceiver>
    <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
    <tp:Endpoint tp:uri="https://localhost:8443/ebXML/msh" tp:type="allPurpose"/>
    <tp:TransportServerSecurity>
      ...
    </tp:TransportServerSecurity>
  </tp:TransportReceiver>
</tp:Transport>
```

- DocExchange: παρέχει πληροφορίες σχετικά με την ανταλλαγή εγγράφων (μηνυμάτων) του συνεργάτη. Ένα παράδειγμα του στοιχείου Transport είναι το παρακάτω:

```

<tp:DocExchange tp:docExchangeId="TYPEA">
  <tp:ebXMLSenderBinding tp:version="2.0">
    <tp:PersistDuration>P1D</tp:PersistDuration>
  </tp:ebXMLSenderBinding>
  <tp:ebXMLReceiverBinding tp:version="2.0">
    <tp:PersistDuration>P1D</tp:PersistDuration>
  </tp:ebXMLReceiverBinding>
</tp:DocExchange>

```

5.2.2 Περιγραφή του στοιχείου CollaborationRole

Όπως αναφέρθηκε, το στοιχείο CollaborationRole συσχετίζει τον συνεργάτη με έναν συγκεκριμένο ρόλο στην επιχειρησιακή συνεργασία που περιέχεται στο έγγραφο περιγραφής επιχειρησιακών διαδικασιών (ebBPSS). Ένα παράδειγμα του στοιχείου CollaborationRole, το οποίο χρησιμοποιείται και στην κατασκευή CPA από τη εφαρμογή CPA Editor, δίνεται παρακάτω:

```

<tp:CollaborationRole id="SessionRequest" >
  <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction" xlink:type="simple"
  xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
  <tp:Role tp:name="OTA_Client" xlink:type="simple"
  xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Client"/>
  <tp:ApplicationCertificateRef tp:certId="CompanyA_AppCert"/>
  <tp:ServiceBinding>
    <tp:Service>Session</tp:Service>
    <tp:CanSend>
      <tp:ThisPartyActionBinding tp:id="OTAClient_ABID1" tp:action="CreateRQ"
      tp:packageId="SessionCreateRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
      </tp:ThisPartyActionBinding>
      <tp:OtherPartyActionBinding>OTAServer_ABID1
      </tp:OtherPartyActionBinding>
      <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAClient_ABID2"
        tp:action="CreateRS" tp:packageId="SessionCreateResponsePackage">
          <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
        <tp:OtherPartyActionBinding>OTAServer_ABID2
        </tp:OtherPartyActionBinding>
      </tp:CanReceive>
      <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAClient_ABID3"
        tp:action="ErrorRS" tp:packageId="ErrorResponsePackage">
          <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
        <tp:OtherPartyActionBinding>OTAServer_ABID3</tp:OtherPartyActionBinding>
      </tp:CanReceive>
    </tp:CanSend>
  </tp:ServiceBinding>

```

Το στοιχείο CollaborationRole περιλαμβάνει τα παρακάτω υποστοιχεία:

- ProcessSpecification: παρέχει σύνδεση με το έγγραφο περιγραφής επιχειρησιακών διαδικασιών (ebBPSS).
- Role: δηλώνει ποιόν ρόλο στην επιχειρησιακή διαδικασία είναι σε θέση να υποστηρίξει μέσω του στοιχείου ServiceBinding.
- ApplicationCertificateRef: καθορίζει ποιο πιστοποιητικό θα χρησιμοποιηθεί στην εφαρμογή για την ψηφιακή υπογραφή και την κρυπτογράφηση.
- ServiceBinding: περιγράφει όλα τα μηνύματα που αποστέλλονται ή λαμβάνονται από τον συνεργάτη μέσα στα πλαίσια της επιχειρησιακής διαδικασίας. Περιλαμβάνει τα παρακάτω υποστοιχεία:
 - Service: είναι τύπου string και χρησιμοποιείται κυρίως για την δρομολόγηση των μηνυμάτων όπως το αντίστοιχο στοιχείο στην επικεφαλίδα του ebXML μηνύματος.
 - CanSend , CanReceive: χρησιμοποιούνται μαζί με το στοιχείο Service κυρίως για την δρομολόγηση των μηνυμάτων στο κατάλληλο σημείο της εφαρμογής. Όπως φαίνεται και στο παραπάνω σχήμα, τα υποστοιχεία τους ThisPartyActionBinding και OtherPartyActionBinding περιέχουν το χαρακτηριστικό Action, το οποίο συμπεριλαμβάνεται μαζί με το στοιχείο Service στην επικεφαλίδα του ebXML μηνύματος. Το στοιχείο ThisPartyActionBinding συσχετίζει την ενέργεια (Action) που καλείται να εκτελέσει ο συγκεκριμένος συνεργάτης (στον οποίο αντιστοιχεί το στοιχείο PartyInfo στο οποίο περιέχεται το συγκεκριμένο στοιχείο ServiceBinding), ενώ το στοιχείο OtherPartyActionBinding συσχετίζει την ενέργεια (Action) που καλείται να εκτελέσει ο έτερος συνεργάτης (στον οποίο αντιστοιχεί το άλλο στοιχείο PartyInfo). Ο συνδυασμός Service και Action χρησιμοποιείται για τον καθορισμό του αντίστοιχου ζεύγους Υπηρεσία / Δράση που ορίζεται στις προδιαγραφές του OTA.

5.3 Περιγραφή της λειτουργίας και του γραφικού περιβάλλοντος του CPA Editor

Όπως προαναφέρθηκε, η εφαρμογή χρησιμοποιεί μια φόρμα δεδομένων την οποία ο πελάτης συμπληρώνει δίνοντας πληροφορίες που αφορούν το δικό του ebMS/OTA σύστημα. Οι πληροφορίες περιλαμβάνουν:

- Το όνομα του CPA, η τιμή του οποίου αποδίδεται στην τιμή του χαρακτηριστικού cpaId.
- Την ηλεκτρονική διεύθυνση του OTA Server και του υποψήφιου πελάτη. Τα πεδία αυτά έχουν προεπιλεγμένη τιμή και είναι απενεργοποιημένα γιατί στην παρούσα υλοποίηση και ο OTA Server και ο Demo Client που χρησιμοποιείται για την χρήση των Web Services λαμβάνουν μηνύματα από την ίδια ηλεκτρονική διεύθυνση. Εμφανίζονται στην φόρμα μόνο για την καλλίτερη κατανόηση του CPA.
- Τις υπηρεσίες που επιθυμεί να χρησιμοποιήσει, επιλέγοντας από αυτές που παρέχει ο ebXML OTA Server, δηλαδή τις υπηρεσίες HotelBooking και Profile, καθώς και ποιες ενέργειες (Actions) επιθυμεί να χρησιμοποιεί στα πλαίσια των παραπάνω υπηρεσιών (π.χ. OTA_ProfileCreateRQ για τη δημιουργία προφίλ). Η υπηρεσία Session είναι προεπιλεγμένη και δεν δίνεται σαν επιλογή καθώς η ύπαρξη της απαιτείται για την λειτουργία του συστήματος και την εκτέλεση των Web Services.

Το CPA που κατασκευάζεται αποτελείται από δύο στοιχεία PartyInfo, ένα για τον πελάτη των υπηρεσιών του OTA (οπότε και η ιδιότητα PartyName του στοιχείου PartyInfo παίρνει την τιμή «OTA_Client») και ένα για τον πάροχο των Web Services, που στην συγκεκριμένη περίπτωση είναι το σύστημα ebXML OTA Server (οπότε και η ιδιότητα PartyName του αντίστοιχου στοιχείου PartyInfo παίρνει την τιμή «OTA_Server»). Για κάθε υπηρεσία (Service) που επιλέγει ο χρήστης κατασκευάζεται ένα στοιχείο CollaborationRole, δίνοντας στην ιδιότητα id μια τιμή που σχετίζεται με την υπηρεσία που επιλέχθηκε. Για παράδειγμα, για την υπηρεσία Session, το id του στοιχείου CollaborationRole του στοιχείου PartyInfo που αναφέρεται στον πελάτη παίρνει την τιμή «SessionRequest», επειδή ο πελάτης είναι αυτός που καλεί την υπηρεσία, ενώ το id του στοιχείου CollaborationRole του στοιχείου PartyInfo που αναφέρεται στον ebXML OTA Server παίρνει την τιμή

«SessionResponse», μιας και ο ebXML OTA Server είναι αυτός που αποκρίνεται στην κλήση της υπηρεσίας).

Για κάθε ενέργεια (Action) σε μία υπηρεσία (Service) κατασκευάζεται ένα στοιχείο CanSend, στο οποίο ορίζεται τι αποστέλλεται και περιλαμβάνει και την ιδιότητα «action», η οποία παίρνει την τιμή της ενέργειας που επιλέχθηκε (π.χ. OTA_HotelSearchRQ). Κατασκευάζεται επίσης και ένα στοιχείο CanReceive, στο οποίο ορίζεται τι λαμβάνεται σαν αποτέλεσμα της ενέργειας που ορίζεται στο στοιχείο CanSend, στο οποίο επίσης περιλαμβάνεται η ιδιότητα «action» η οποία παίρνει την τιμή της ενέργειας απόκρισης του συνεργάτη στην συγκεκριμένη υπηρεσία (π.χ. OTA_HotelSearchRS).

Τα στοιχεία Transport, DocExchange και DeliveryChannel παίρνουν τιμές που ορίζει ο OTA στις προδιαγραφές του. Στο παράρτημα Α δίνεται το πλήρες XML έγγραφο ενός CPA που κατασκευάστηκε με την εφαρμογή CPA Editor, όπου εμφανίζονται και τα παραπάνω στοιχεία.

Στο παρακάτω σχήμα (Εικόνα 50) δίνεται το γραφικό περιβάλλον της εφαρμογής OTA CPA Editor.

OTA CPA Editor	
CPA Name	OTAClient_OTAServer_Demo_CPA
OTA Server URI	localhost:8080/ebXML/msh
OTA Client URI	localhost:8080/ebXML/msh
Select Service / Action	
Profile	Hotel Booking
Profile Create Request	Hotel Search Request
Profile Read Request	Hotel Availability Request
Profile Update Request	Hotel Reservation Request
Profile Delete Request	Read Hotel Reservation
	Cancel Hotel Reservation

Εικόνα 50 Το γραφικό περιβάλλον της εφαρμογής OTA CPA Editor

Η λειτουργία της εφαρμογής στηρίζεται στην μέθοδο `DataBinding`. Δηλαδή οι πληροφορίες που συλλέγονται από την φόρμα δεδομένων αντιστοιχίζονται στα κατάλληλα Java αντικείμενα. Τα αντικείμενα αυτά αποτελούν το στιγμιότυπο της ιεραρχίας κλάσεων που κατασκευάζεται από το XML Schema του CPA μέσω της διαδικασίας `DataBinding`.

Όταν ο χρήστης συμπληρώσει τα στοιχεία της φόρμας δεδομένων της εφαρμογής, πρέπει να πατήσει το πλήκτρο `Submit`, οπότε, τα στοιχεία συλλέγονται και οι τιμές τους ανατίθενται στα κατάλληλα Java αντικείμενα. Πατώντας το πλήκτρο `Reset` διαγράφονται οι τιμές που εισάχθηκαν στα Java αντικείμενα, και ο χρήστης μπορεί να τροποποιήσει τις επιλογές του. Πατώντας το πλήκτρο `Save` τα Java αντικείμενα μετατρέπονται σε ένα `string` που περιέχει το XML έγγραφο του CPA και αποθηκεύεται σαν αρχείο με το όνομα `CPA Name + ".xml"`. Πατώντας το πλήκτρο `View XML` εμφανίζεται το XML έγγραφο που κατασκευάστηκε.

5.4 Ανακεφαλαίωση

Στο κεφάλαιο αυτό έγινε μια περιγραφή των προδιαγραφών της ebXML για την συμφωνία του πρωτοκόλλου συνεργασίας (`Collaboration Protocol Agreement`, CPA), και της εφαρμογής CPA Editor. Παρουσιάστηκαν τα στοιχεία που αποτελούν ένα CPA δίνοντας παραδείγματα, και εξηγήθηκε ο ρόλος τους στην γενικότερη δομή του CPA. Έγινε επίσης περιγραφή της λειτουργίας και του γραφικού περιβάλλοντος της εφαρμογής CPA Editor.

Στο επόμενο κεφάλαιο περιγράφεται η εφαρμογή που αναπτύχθηκε (`Demo Client`) και χρησιμοποιεί όλες τις υπηρεσίες που υλοποιήθηκαν.

Κεφάλαιο 6

Περιγραφή του Demo Client

6.1 Εισαγωγή

Προκειμένου να ελεγχθεί η ορθή λειτουργία του συστήματος ebXML OTA Server που υλοποιήθηκε, καθώς και για να γίνει δοκιμαστική χρήσης των υπηρεσιών που υλοποιήθηκαν, κατασκευάστηκε μια δικτυακή εφαρμογή πελάτης (Demo Client) η οποία δίνει τη δυνατότητα χρησιμοποίησης όλης της λειτουργικότητας του ebXML OTA Server.

Συγκεκριμένα η εφαρμογή παρέχει τη δυνατότητα για:

- Αναζήτηση ξενοδοχείων με βάση τα κριτήρια που θέτει ο χρήστης, και εμφάνιση των αποτελεσμάτων της αναζήτησης.
- Αναζήτηση διαθέσιμων δωματίων ξενοδοχείων που πληρούν τα κριτήρια που θέτει ο χρήστης, και εμφάνιση των αποτελεσμάτων. Η αναζήτηση διαθεσιμότητας εκτελείται είτε στο σύνολο ή σε επιλεγμένο υποσύνολο των ξενοδοχείων που επέστρεψε η αναζήτηση ξενοδοχείων.
- Δημιουργία νέας κράτησης, τροποποίηση ή αργόνηση της. Ο χρήστης επιλέγει από τα αποτελέσματα της αναζήτησης διαθεσιμότητας ποια δωμάτια επιθυμεί να κρατήσει.
- Διαχείριση υπάρχουσας (αποθηκευμένης) κράτησης:
 - Ανάκτηση,
 - Τροποποίηση ή
 - Ακύρωση
- Δημιουργία νέου προφίλ χρήστη.
- Διαχείριση των προφίλ χρήστη:
 - Ανάκτηση,
 - Τροποποίηση ή
 - Διαγραφή

6.2 Επισκόπηση της εφαρμογής

Η εφαρμογή αποτελείται από τρία μέρη:

- Το γραφικό περιβάλλον που αποτελείται από ένα σύνολο από JSP σελίδες για την εισαγωγή δεδομένων από τον χρήστη και την εμφάνιση των αποτελεσμάτων.
- Την διασύνδεση με τον OTA Server μέσω ebMS, χρησιμοποιώντας ένα CPA που έχει κατασκευαστεί με τον CPA Editor.
- Μια βάση δεδομένων που αποθηκεύει κάποιες βασικές πληροφορίες σχετικά με την εφαρμογή, όπως:
 - Τα προφίλ που έχουν δημιουργηθεί.
 - Και τις κρατήσεις που έχουν γίνει.

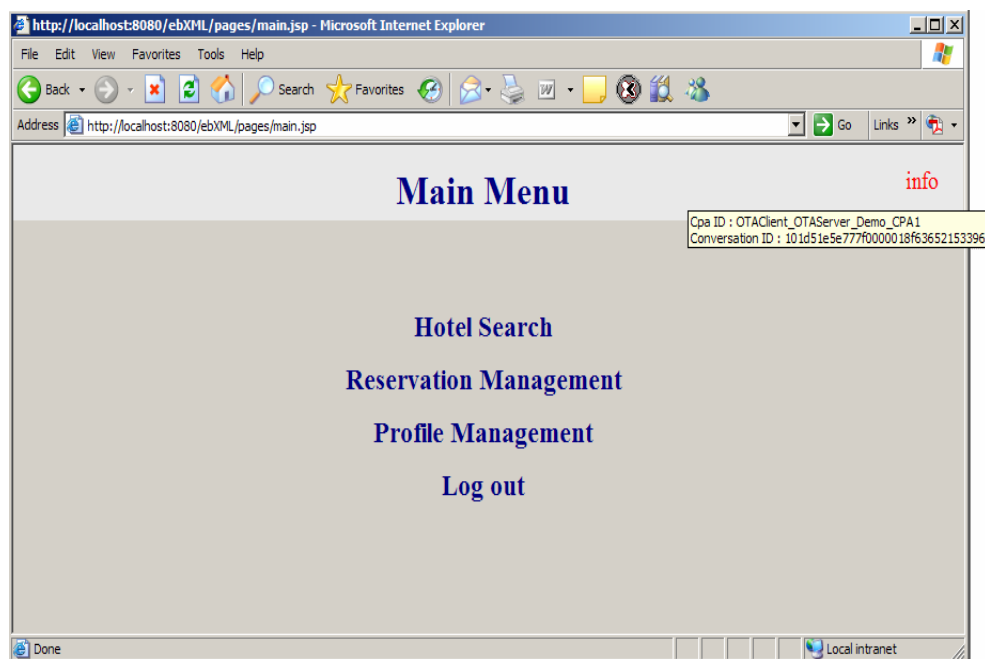
Αρχικά ο χρήστης πρέπει να επιλέξει ένα CPA (Εικόνα 51) το οποίο θα χρησιμοποιηθεί για την εκτέλεση των υπηρεσιών και να πατήσει το πλήκτρο «Select CPA». Στη συνέχεια, αποστέλλεται μία αίτηση για δημιουργία ενός νέου OTA Session με το CPA που επιλέχθηκε. Εάν η διαδικασία δημιουργίας νέου OTA Session επιτύχει, τότε ο χρήστης κατευθύνεται στο κύριο μενού της εφαρμογής (Εικόνα 52). Εκεί του δίνονται οι παρακάτω επιλογές:

- Να προχωρήσει στην αναζήτηση ξενοδοχείων. Η αναζήτηση διαθεσιμότητας και η κράτηση ξενοδοχείων εκτελούνται σαν συνέχεια της αναζήτησης ξενοδοχείων.
- Να διαχειριστεί τις κρατήσεις του.
- Να δημιουργήσει ή να ενημερώσει το προφίλ του.
- Να αποσυνδεθεί από το σύστημα (log out). Η αποσύνδεση εκτελείται μέσω της αποστολής αίτησης για τερματισμό του ενεργού OTA Session.

Στην πάνω δεξιά γωνία της οθόνης της εφαρμογής βρίσκεται πάντα μια ετικέτα που ονομάζεται «info». Όταν ο χρήστης περάσει το ποντίκι πάνω από την ετικέτα αυτή εμφανίζονται το CPA ID και το Conversation ID της σύνδεσης.



Εικόνα 51 Η βασική σελίδα της εφαρμογής Demo Client όπου γίνεται η επιλογή του CPA



Εικόνα 52 Το βασικό μενού της εφαρμογής Demo Client όπου ο χρήστης επιλέγει τη λειτουργία που επιθυμεί να εκτελέσει

Η σελίδα για την αναζήτηση ξενοδοχείων (Εικόνα 53) περιέχει ένα σύνολο από κριτήρια αναζήτησης που μπορεί να ορίσει ο χρήστης. Το αποτέλεσμα της αλληλεπίδρασης του χρήστη με τη σελίδα αυτή είναι να κατασκευαστεί ένα XML έγγραφο το οποίο περιέχει τα κριτήρια αναζήτησης ξενοδοχείων προκειμένου να

αποστέλλει στον OTA Server. Τα δεδομένα της σελίδας αποτελούν ένα Criterion. Όπως αναφέρθηκε στην ενότητα 4.5.5 Διαδικασία Αναζήτησης ξενοδοχείων, όπου περιγράφηκε η διαδικασία αναζήτησης ξενοδοχείων, η αίτηση αναζήτησης μπορεί να αποτελείται από περισσότερα του ενός Criterion και βρίσκονται κάτω από το στοιχείο Criteria. Πατώντας το πλήκτρο «Add» προστίθεται ένα καινούργιο Criterion στο XML έγγραφο της αναζήτησης ξενοδοχείων. Πατώντας το πλήκτρο «Search», το XML έγγραφο αποστέλλεται στον ebXML OTA Server. Πατώντας το πλήκτρο «Reset» διαγράφουμε τα κριτήρια που έχουμε εισάγει για να δώσουμε καινούργια. Πατώντας το πλήκτρο «Return» επιστρέφουμε στο κύριο μενού της εφαρμογής και το έγγραφο που έχουμε δημιουργήσει χάνεται.

The screenshot shows a web-based form titled "Hotel Search Request". On the left, there are four buttons: "Add", "Search", "Reset", and "Return". The form contains several input fields and checkboxes. The "Address" section includes fields for CountryName (Greece), StateProv, County, CityName, PostalCode, StreetNumber, PO_Box, BldgRoom, FormattedInd (checked), and AddressLine. The "Reference Point" section has a checked "Reference Point" checkbox, a "Radius" checkbox, and fields for Distance, Distance Measure (km), and Direction. The "HotelRef" section includes a checked "HotelRef" checkbox and fields for HotelName, ChainCode, BrandCode, HotelCode, HotelCityCode, HotelCodeContext, and AreaId. The "Telephone" section has a checked "Telephone" checkbox, a "FormattedInd" checkbox, and fields for PhoneNumber, CountryAccessCode, AreaCityCode, PIN, Extension, ShareSyncInd, ShareMarketInd, PhoneTechType, and PhoneLocationType. The "CodeRef" section has a "CodeRef" checkbox and fields for CodeRef, LocationCode, and CodeContext. The "Position" section has a "Position" checkbox and fields for Latitude, Longitude, and Altitude. On the right, there is a "HotelAmenities" section with a checked "HotelAmenities" checkbox and a list of amenities: Air conditioning, Baby sitting, Lounges/bars, Playground, Elevators, Currency exchange, Hairdresser/barber, Heated pool, Laundry/Valet service, Massage services, Parking, Security, Poolside snack bar, Bell staff/porter, Room service, Safe deposit box, and Video games.

Εικόνα 53 Η σελίδα εισαγωγής των κριτηρίων αναζήτησης ξενοδοχείων

Τα πεδία της φόρμας τα οποία μπορεί να συμπληρώσει ο χρήστης είναι (όπως φαίνεται και στην εικόνα) τα εξής:

- MaxResponses – ορίζει τον μέγιστο πλήθος αποτελεσμάτων που θέλουμε να επιστραφεί
- ExactMatch – ορίζουμε αν θέλουμε το κριτήριο να ικανοποιείται πλήρως ή όχι
- ImportanceType – ορίζουμε την βαρύτητα που θέλουμε να αποδώσουμε στο κριτήριο
- StateProv – ο νομός της Κρήτης που μας ενδιαφέρει
- County – ο δήμος που μας ενδιαφέρει

- CityName – το όνομα της πόλης που μας ενδιαφέρει
- PostalCode – ο ταχυδρομικός κώδικας της περιοχής που μας ενδιαφέρει
- StreetNumber, PO_Box – η οδός και ο αριθμός που μας ενδιαφέρουν
- ReferencePoint – ορίζουμε ένα σημείο αναφοράς για την αναζήτηση προσδιορίζοντας μια συγκεκριμένη τοποθεσία ενδιαφέροντος.
- Distance – η απόσταση από το σημείο αναφοράς που προσδιορίζει την ακτίνας ενός ιδεατού κύκλους με κέντρο το σημείο αναφοράς εντός του οποίου αναζητούμε ξενοδοχεία
- HotelName – το όνομα του ξενοδοχείου (εάν το ξέρουμε) το οποίο μας ενδιαφέρει
- PhoneNumber – ο αριθμός τηλεφώνου του ξενοδοχείου που μας ενδιαφέρει

Τα αποτελέσματα της αναζήτησης ξενοδοχείων που επιστρέφει ο ebXML OTA Server εμφανίζονται στην αντίστοιχη σελίδα (Εικόνα 54). Εμφανίζονται βασικές πληροφορίες για κάθε ξενοδοχείο, όπως:

- Το όνομα του ξενοδοχείου
- Η διεύθυνση
- Το τηλέφωνο
- Μια σύντομη περιγραφή του
- Η βαθμολογία του σύμφωνα με τα κριτήρια αναζήτησης

Ο χρήστης μπορεί να επιλέξει ένα ή περισσότερα ξενοδοχεία, μέσω του πεδίου που βρίσκεται δεξιά από κάθε ξενοδοχείο. Τα ξενοδοχεία που επιλέγονται χρησιμοποιούνται στη συνέχεια για την αναζήτηση διαθεσιμότητας. Έχοντας επιλέξει τα ξενοδοχεία που μας ενδιαφέρουν μεταφερόμαστε στην σελίδα αναζήτησης διαθεσιμότητας πατώντας το πλήκτρο «Continue». Πατώντας το πλήκτρο «Return» επιστρέφουμε στην σελίδα αναζήτησης ξενοδοχείων.

Continue

Return

Hotel Search Response

Menia Apartmentsweight : 1.2☐

Platanias, Kydonias , Platanias , Chania , 73100
Telephone : 08210 60615
Description : The apartment house is located in Platanias 15 km southwest of Chania next to the sandy beach. Platanias is a very busy area with a lot of shops, restaurants, bars and discos. The studios/apartments are for 2-4 people. All of the rooms have kitchen, air condition and balcony with beautiful sea view. Bus stop and mini-markets are very close. Total capacity : 120 people.

Ritaweight : 1.2☐

Platanias, Kydonias , Platanias , Chania , 73100
Telephone : 08210 60280-2

Kalliaweight : 1.2☐

Platanias, Kydonias , Platanias , Chania , 73100
Telephone : 08210 60280-2

Eratoweight : 1.2☐

Platanias, Kydonias , Platanias , Chania , 73100
Telephone : 08210 68824, 08210 68802, 08210

Elisso Hotel - Villageweight : 1.1☐

Platanias, Kydonias , Platanias , Chania
Telephone : 08210 60930-2

Εικόνα 54 Η σελίδα που εμφανίζει τα αποτελέσματα της αναζήτησης ξενοδοχείων

Η σελίδα για την αναζήτηση διαθεσιμότητας περιέχει τα κριτήρια που μπορεί να θέσει ο χρήστης (Εικόνα 55). Ο χρήστης μπορεί να επιλέξει ένα ή περισσότερα ξενοδοχεία (περιορίζοντας περισσότερο την επιλογή που έκανε στα αποτελέσματα της αναζήτησης ξενοδοχείων), να ορίσει αν θέλει να χρησιμοποιηθεί το προφίλ του και πώς, να ορίσει την επιλογή εναλλακτικής διαθεσιμότητας καθώς και τη χρήση του προφίλ χρήστη στην εναλλακτική διαθεσιμότητα. Όπως και στην αναζήτηση ξενοδοχείων, μπορεί να προσθέσει πολλές συλλογές από διαφορετικά κριτήρια στο XML έγγραφο της αναζήτησης διαθεσιμότητας.

Hotel Availability Request

Add

Search

Reset

Return

Select hotel(s)

Menia Apartments

Rita

Kallia

Select all hotels ☐

Select Profile

Alexopoulos Dimitris

Availability Profile Usage

None

Alternate Availability ☐

Alternate Availability Profile Usage

None

Stay Date Range *

check in date

Day Month Year

Duration* 1 day(s)

Room Type

Room Location

Room View

Bed Type

Smoking ☐

of Rooms

Guests

Age category Age Guest count

Over 21

Add Remove

Age category | Age | Count

Rate Range

min max currency

euro

Rate Plan

Club/concierge

Convention

Corporate

Day rate

Distressed inventory

Family plan

Government

Military

Negotiated

Package

Promotional

Regular/rack

Senior citizen

Tour/wholesale

Room Amenities (0-5 selections)

Adjoining rooms

Air conditioning

Alarm clock

All news channel

AM/FM radio

Baby listening device

Balcony/Lanai/Terrace

Barbeque grills

Add Remove

Preferred ☒

Unacceptable ☐

Only ☐

Εικόνα 55 Η σελίδα εισαγωγής των κριτηρίων αναζήτησης διαθεσιμότητας

- Το πεδίο «Select hotel(s)» μας δίνει τη δυνατότητα να επιλέξουμε ένα ή περισσότερα από τα ξενοδοχεία που υπάρχουν στην λίστα. Αν θέλουμε να τα επιλέξουμε όλα μπορούμε να το κάνουμε είτε από την λίστα αυτή είτε επιλέγοντας το πεδίο που βρίσκεται από κάτω και ονομάζεται «Select all hotels».
- Στο πεδίο «Select Profile» μπορούμε να επιλέξουμε εάν θέλουμε να χρησιμοποιήσουμε το προφίλ μας στην αναζήτηση διαθεσιμότητας. Το πεδίο «Availability Profile Usage» μας επιτρέπει να δηλώσουμε τον τρόπο χρήσης του προφίλ, FullMatch εάν θέλουμε πλήρη ταύτιση με τις προτιμήσεις που ορίζονται στο προφίλ, ή WeightsOnly εάν θέλουμε το προφίλ να χρησιμοποιηθεί για να αποδώσει επιπλέον βαρύτητα στα αποτελέσματα.
- Το πεδίο «Alternate Availability» μας επιτρέπει να επιλέξουμε εάν θέλουμε να χρησιμοποιήσουμε την δυνατότητα εναλλακτικής διαθεσιμότητας, ενώ το πεδίο «Alternate Availability Profile Usage» μας επιτρέπει να ορίσουμε την χρήση του προφίλ στην εναλλακτική διαθεσιμότητα.

- Στα πεδία «check in date» και «Duration» ορίζουμε την ημερομηνία και το χρονικό διάστημα για το οποίο αναζητούμε διαθεσιμότητα. Η χρήση των δύο αυτών πεδίων είναι υποχρεωτική για την λειτουργία της υπηρεσίας αναζήτησης διαθεσιμότητας.
- Τα πεδία που βρίσκονται στο πλαίσιο «Rate Range» δηλαδή το «min» και το «max» μας επιτρέπουν να ορίσουμε το ελάχιστο και το μέγιστο ποσό που θέλουμε να κοστίζουν τα δωμάτια.
- Στο πλαίσιο «RatePlan» μπορούμε να ορίσουμε το RatePlan που επιθυμούμε.
- Τα πεδία «Room Type», «Room Location», «Room View», «Bed Type» και «Smoking» μας επιτρέπουν να ορίσουμε τον τύπο του δωματίου, την τοποθεσία του δωματίου, την θέα που επιθυμούμε να έχει, τον τύπο του κρεβατιού και το εάν θέλουμε να είναι καπνιστών ή όχι.
- Το πεδίο «# of Rooms» μας επιτρέπει να ορίσουμε το πλήθος των δωματίων που επιθυμούμε.
- Το πεδίο «Room Amenities» μας επιτρέπει να επιλέξουμε μέχρι 5 διευκολύνσεις (τόσες επιτρέπονται από τις προδιαγραφές του OTA) που επιθυμούμε να διαθέτει ή να μην διαθέτει το δωμάτιο. Κάθε μια διευκόλυνση που θέλουμε να διαλέξουμε, την επιλέγουμε με το ποντίκι και πατάμε το πλήκτρο «Add», οπότε και προστίθεται στην λίστα η οποία εμφανίζεται κάτω από το πλήκτρο «Add» (όταν δεν έχουμε κάνει καμία επιλογή η λίστα δεν εμφανίζεται, εμφανίζεται όταν περιέχει τουλάχιστον ένα στοιχείο). Επίσης, πριν πατήσουμε το πλήκτρο «Add» μπορούμε να διαλέξουμε το είδος της προτίμησης που έχουμε στην συγκεκριμένη διευκόλυνση, επιλέγοντας ένα από τα πεδία «Preferred», «Unacceptable» και «Only».

Τα αποτελέσματα της αναζήτησης διαθεσιμότητας εμφανίζονται στην αντίστοιχη σελίδα (Εικόνα 56). Η σελίδα απεικονίζει τα διαθέσιμα RatePlans, RoomTypes και RoomRates κάθε ξενοδοχείου που ικανοποιούν τα κριτήρια που δόθηκαν. Δεξιά από κάθε αποτέλεσμα υπάρχει ένα πεδίο στο οποίο ο χρήστης δίνει τον αριθμό των

δωματίων που επιθυμεί να κρατήσει. Πατώντας το πλήκτρο «Continue» ενεργοποιείται η σελίδα δημιουργίας κράτησης.

Hotel Availability Response

Continue
Return

Elisso Hot

Rate Plan	Room Type	Room Rate	Quantity
Travel industry			
	4793	60.0	<input type="text"/>
Tour/wholesale			
	4790	80.0	<input type="text"/>
	4792	60.0	<input type="text"/>
	4793	60.0	<input type="text"/>
Corporate			
	4792	70.0	<input type="text"/>
Regular/rack			
	4790	100.0	<input type="text"/>
	4793	100.0	<input type="text"/>

This room type offers the following room amenities :
Pens
Telephone for hearing
Oven
Smoking
Computer

Kallia

Rate Plan	Room Type	Room Rate	Quantity
Tour/wholesale			
	4851	100.0	<input type="text"/>
	4854	50.0	<input type="text"/>
Distressed inventory			
	4851	80.0	<input type="text"/>
	4853	60.0	<input type="text"/>
	4854	60.0	<input type="text"/>
	4855	90.0	<input type="text"/>
	4856	70.0	<input type="text"/>

Εικόνα 56 Η σελίδα που εμφανίζει τα αποτελέσματα της αναζήτησης διαθεσιμότητας

Στη σελίδα δημιουργίας κράτησης (Εικόνα 57) δίνεται μια γενική επισκόπηση των επιλογών του χρήστη. Εμφανίζονται όλα τα ξενοδοχεία που επιθυμεί να κλείσει, τα συγκεκριμένα RatePlans, RoomTypes και RoomRates, καθώς και το πλήθος των δωματίων κάθε επιλογής και το συνολικό κόστος των κρατήσεων. Έτσι ο χρήστης έχει τη δυνατότητα να επανεξετάσει τις επιλογές του, να αφαιρέσει συγκεκριμένες επιλογές ή να αυξήσει τον αριθμό δωματίων κάθε επιλογής, βλέποντας ταυτόχρονα το συνολικό κόστος των επιλογών του.

Πατώντας το πλήκτρο «Set» κατασκευάζεται το XML έγγραφο για την δημιουργία νέας κράτησης, που περιέχει τις επιλογές του χρήστη. Στη συνέχεια, πατώντας το πλήκτρο «Initiate» αποστέλλεται στον ebXML OTA Server το XML έγγραφο για να

ξεκινήσει η συνδιαλλαγή της κράτησης. Ο ebXML OTA Server ελέγχει τη διαθεσιμότητα των ζητούμενων δωματίων και επιστρέφει στον χρήστη το αποτέλεσμα της διαδικασίας. Στη συνέχεια ο χρήστης μπορεί να επικυρώσει την κράτηση πατώντας το πλήκτρο «Commit», να τροποποιήσει την κράτηση πατώντας το πλήκτρο «Modify» ή να σταματήσει τη συνδιαλλαγή κράτησης πατώντας το πλήκτρο «Ignore».

hotel	Rate Plan	RoomType	Rate	# of rooms	
Erato	Promotional	4941	80.0	1	<input checked="" type="checkbox"/>
Kallia	Government	4856	90.0	1	<input checked="" type="checkbox"/>
		4853	100.0	1	<input checked="" type="checkbox"/>
	Regular/rack	4856	90.0	1	<input checked="" type="checkbox"/>
	Tour/wholesale	4851	100.0	1	<input checked="" type="checkbox"/>
Rita	Negotiated	4839	100.0	1	<input checked="" type="checkbox"/>
				560	€

Εικόνα 57 Η σελίδα για την προετοιμασία και εκτέλεση κράτησης ξενοδοχείων

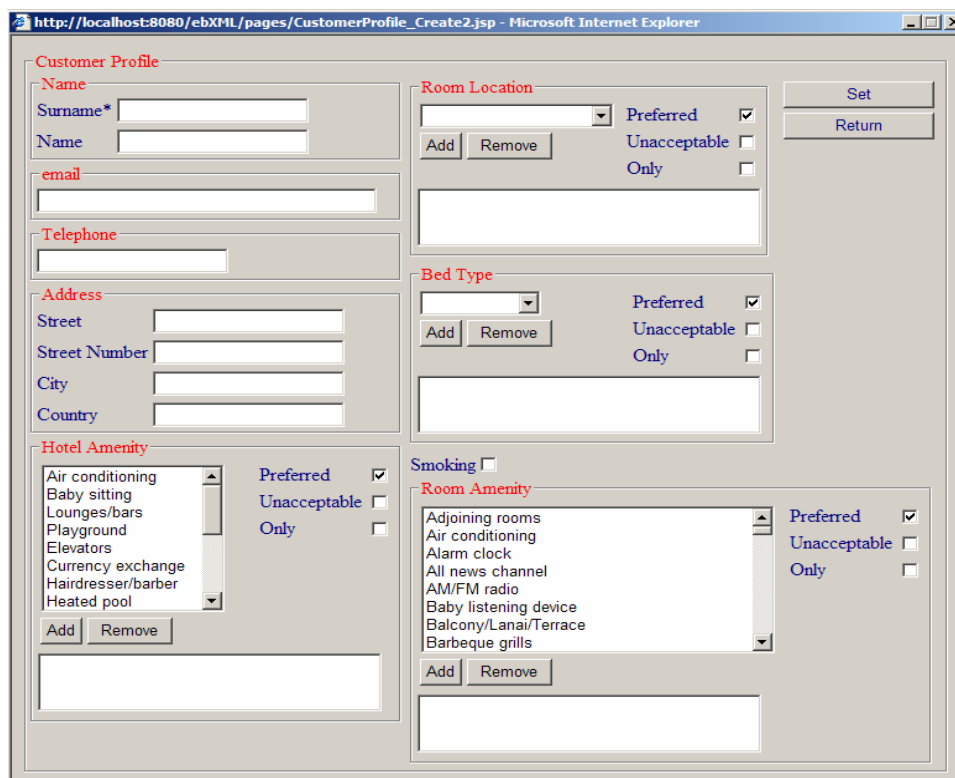
Πατώντας το πλήκτρο «Load Profile ID» εμφανίζεται η παρακάτω σελίδα (Εικόνα 58) όπου μπορούμε να επιλέξουμε ένα υπάρχον προφίλ για να το συσχετίσουμε με την κράτηση.

Εικόνα 58 Η σελίδα όπου γίνεται η επιλογή ενός υπάρχοντος προφίλ

Στην σελίδα αυτή υπάρχει ένα πλαίσιο όπου εμφανίζονται τα πεδία SurName και GivenName των προφίλ που υπάρχουν αποθηκευμένα στην βάση δεδομένων.

Μπορούμε να διαλέξουμε ένα από αυτά επιλέγοντάς το με το ποντίκι και πατώντας το πλήκτρο «Select». Στη συνέχεια πρέπει να πατήσουμε τα πλήκτρα «Return» για να επιστρέψουμε στην σελίδα δημιουργίας κράτησης (Εικόνα 57). Εάν πατήσουμε το πλήκτρο «Return» χωρίς να επιλέξουμε κάποιο προφίλ επιστρέφουμε απλά στην σελίδα δημιουργίας κράτησης.

Πατώντας το πλήκτρο «New Profile» εμφανίζεται μια σελίδα κατασκευής νέου προφίλ, όμοια με την κανονική σελίδα κατασκευής προφίλ που περιγράφεται παρακάτω. Η σελίδα που εμφανίζεται πατώντας το πλήκτρο «New Profile» στην σελίδα δημιουργίας νέας κράτησης δίνεται παρακάτω (Εικόνα 59). Πατώντας το πλήκτρο «Set» που βρίσκεται στο πάνω δεξιό τμήμα της φόρμας εισάγονται τα δεδομένα της φόρμας της σελίδας στα αντίστοιχα Java αντικείμενα του προφίλ. Στη συνέχεια πατώντας το πλήκτρο «Return» επιστρέφουμε στην σελίδα δημιουργίας νέας κράτησης.



Εικόνα 59 Η σελίδα όπου δημιουργείται ένα νέο προφίλ που θα συσχετιστεί με την νέα κράτηση

Μετά το τέλος της συνδιαλλαγής νέας κράτησης και την επιστροφή στην κεντρική σελίδα της εφαρμογής, ο χρήστης έχει τη δυνατότητα να ανακτήσει τις κρατήσεις του από την βάση δεδομένων του ebXML OTA Server, στέλνοντας το κατάλληλο XML έγγραφο ανάγνωσης (OTA_ReadRQ). Οι κρατήσεις του εμφανίζονται στην

αντίστοιχη σελίδα (Εικόνα 60), όπου ο χρήστης έχει τη δυνατότητα να κάνει αλλαγές στις κρατήσεις του, να επικυρώσει (εάν υπάρχει η κατάλληλη διαθεσιμότητα) αυτές ή να τις αγνοήσει, και τέλος να ακυρώσει τις κρατήσεις του.

hotel	Rate Plan	RoomType	Rate	# of rooms	
Titika	Promotional	951	60.0	1	<input checked="" type="checkbox"/>
	Military	951	100.0	1	<input checked="" type="checkbox"/>
Eligas	Convention	956	80.0	1	<input checked="" type="checkbox"/>
Iviskos	Day rate	1091	90.0	1	<input checked="" type="checkbox"/>
Rent Rooms	Family plan	1063	80.0	1	<input checked="" type="checkbox"/>
				410	€

Load Profile ID New Profile

Εικόνα 60 Η σελίδα όπου γίνεται η διαχείριση μίας κράτησης ξενοδοχείων

Εκτός από τις κρατήσεις, ο χρήστης έχει τη δυνατότητα να δημιουργήσει ή/και να επεξεργαστεί το προφίλ του. Στην αντίστοιχη σελίδα (Εικόνα 61) δίνεται ένα σύνολο πεδίων που μπορεί να συμπληρώσει ο χρήστης. Ο χρήστης μπορεί επίσης να ενημερώσει το προφίλ του ή/και να το διαγράψει. Η σελίδα που αφορά το προφίλ περιλαμβάνει ένα υποσύνολο των πεδίων που υποστηρίζονται από τις προδιαγραφές του OTA για το προφίλ, τα οποία θα αναφερθούν παρακάτω. Είναι σημαντικό όμως να επισημανθεί ότι ο ebXML OTA Server και η βάση δεδομένων του προφίλ υποστηρίζουν το σύνολο των πληροφοριών που περιγράφονται στις προδιαγραφές του OTA για το προφίλ.

Create Profile

Save

View XML

Return

Customer Profile

Name

Surname* Alexopoulos

Name Dimitris

email alexop@ced.tuc.gr

Telephone 6972098358

Address

Street El. Venizelou

Street Number

City Chania

Country Greece

Room Location

Away from the elevator Preferred ☒

Add Remove Unacceptable ☐

Only ☐

Away from the elevator Preferred

Bed Type

Double Preferred ☒

Add Remove Unacceptable ☐

Only ☐

Double Preferred

Hotel Amenity

Air conditioning Preferred ☒

Baby sitting Unacceptable ☐

Lounges/bars Only ☐

Playground

Elevators

Currency exchange

Hairdresser/barber

Heated pool

Add Remove

Air conditioning Preferred

Smoking ☐

Room Amenity

Adjoining rooms Preferred ☒

Air conditioning Unacceptable ☐

Alarm clock Only ☐

All news channel

AM/FM radio

Baby listening device

Balcony/Lanai/Terrace

Barbeque grills

Add Remove

Adjoining rooms Preferred

Εικόνα 61 Η σελίδα για την δημιουργία του προφίλ χρήστη

Τα πεδία που περιλαμβάνει η σελίδα για το προφίλ είναι τα εξής:

- Surname – το επίθετο του χρήστη
- Name – το μικρό όνομα του χρήστη
- Email – το email του
- Telephone – το τηλέφωνο του
- Address – η διεύθυνση του, η οποία αποτελείται από τα παρακάτω πεδία
- Street – η οδός
- Street Number – ο αριθμός της οδού
- City – το όνομα της πόλης
- Country – το όνομα της χώρας
- Hotel Amenity* – διευκολύνσεις ξενοδοχείου
- Room Location* – τοποθεσία δωματίου
- Bed Type* – τύπος κρεβατιού
- Room Amenity* – διευκολύνσεις δωματίου

* Για την επιλογή τιμών από αυτά τα πεδία εργαζόμαστε ως εξής: επιλέγουμε με το ποντίκι και πατάμε το πλήκτρο «Add», οπότε και προστίθεται στην αντίστοιχη λίστα

η οποία εμφανίζεται κάτω από το πλήκτρο «Add» (όταν δεν έχουμε κάνει καμία επιλογή η λίστα δεν εμφανίζεται, εμφανίζεται όταν περιέχει τουλάχιστον ένα στοιχείο). Επίσης, πριν πατήσουμε το πλήκτρο «Add» μπορούμε να διαλέξουμε το είδος της προτίμησης που έχουμε στην συγκεκριμένη διευκόλυνση, επιλέγοντας ένα από τα πεδία «Preferred», «Unacceptable» και «Only».

Η λειτουργία των πλήκτρων της σελίδας είναι η εξής:

- Πατώντας το πλήκτρο «save» η εφαρμογή στέλνει μια αίτηση για κατασκευή νέου προφίλ στον ebXML OTA Server.
- Πατώντας το πλήκτρο «View XML» εμφανίζεται το XML έγγραφο του προφίλ που φτιάξαμε.
- Πατώντας το πλήκτρο «Return» επιστρέφουμε στην προηγούμενη σελίδα.

Η σελίδα για την ενημέρωση του προφίλ είναι όμοια με την σελίδα κατασκευής προφίλ όσον αφορά τα πεδία εισαγωγής δεδομένων. Η διαφορά της με την σελίδα κατασκευής προφίλ είναι τα πλήκτρα ελέγχου που βρίσκονται στο αριστερό τμήμα της σελίδας. Η σελίδα αυτή δίνεται παρακάτω (Εικόνα 62).

Profile Update

Customer Profile

Name

Surname* Alexopoulos

Name Dimitris

email

Telephone 69925

Address

Street

Street Number

City

Country

Room Location

Preferred ☒

Unacceptable ☐

Only ☐

Bed Type

Preferred ☒

Unacceptable ☐

Only ☐

Hotel Amenity

Air conditioning Preferred ☒

Baby sitting Unacceptable ☐

Lounges/bars Only ☐

Playground

Elevators

Currency exchange

Hairdresser/barber

Heated pool

Room Amenity

Adjoining rooms Preferred ☒

Air conditioning Unacceptable ☐

Alarm clock Only ☐

All news channel

AM/FM radio

Baby listening device

Balcony/Lanai/Terrace

Barbeque grills

Εικόνα 62 Η σελίδα για την ενημέρωση του προφίλ χρήστη

Η λειτουργία των πλήκτρων της σελίδας αυτής είναι η εξής:

- Πατώντας το πλήκτρο «Update» η εφαρμογή στέλνει μια αίτηση για τροποποίησης του προφίλ στον ebXML OTA Server.
- Πατώντας το πλήκτρο «Delete» η εφαρμογή στέλνει μια αίτηση για διαγραφή του προφίλ στον ebXML OTA Server.
- Τα πλήκτρα «View XML» και «Return» χρησιμοποιούνται όπως περιγράφηκε στην σελίδα κατασκευής προφίλ. Δηλαδή το πρώτο εμφανίζει το XML έγγραφο που κατασκευάστηκε με τα καινούργια δεδομένα του προφίλ, και το δεύτερο επιστρέφει τον έλεγχο της εφαρμογής στην προηγούμενη σελίδα.

6.3 Ο Τρόπος λειτουργίας της εφαρμογής και η χρήση της ebMS

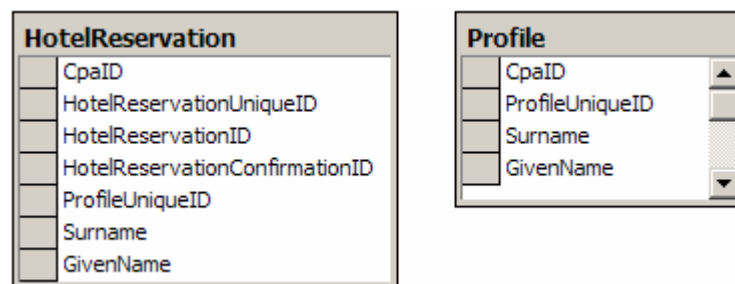
Η εφαρμογή στηρίζεται στη χρήση των αντικειμένων που κατασκευάζονται χρησιμοποιώντας την τεχνική Data Binding. Δηλαδή η εφαρμογή κατασκευάζει αντικείμενα για τις Java κλάσεις που παράγονται από το Breeze XML Binder για το εκάστοτε XML Schema. Στη συνέχεια οι τιμές που εισάγει ο χρήστης στα διάφορα πεδία του interface ανατίθενται στα αντίστοιχα αντικείμενα. Κατόπιν εκτελείται η διαδικασία Marshall όπου όλη πληροφορία που υπάρχει στα αντικείμενα αποτυπώνεται σε ένα String. Το String αυτό αποτελεί το XML έγγραφο που θα αποσταλεί στον ebXML OTA Server.

Το επόμενο βήμα είναι η κατασκευή του ebXML μηνύματος που θα περιέχει το XML έγγραφο και θα αποσταλεί στον ebXML OTA Server. Το μήνυμα κατασκευάζεται παίρνοντας πληροφορίες από το CPA που χρησιμοποιείται στην εκάστοτε σύνδεση, όπως είναι το URI στο οποίο θα αποσταλεί, χαρακτηριστικά του CPA για την αναγνώρισή του από τον ebXML OTA Server όπως το CPAID, το ConversationID που δείχνει την συγκεκριμένη συνδιαλλαγή και άλλα. Τα πεδία Service και Action παίρνουν τιμές που αντιστοιχούν στην υπηρεσία που καλείται. Για παράδειγμα όταν πρόκειται για αίτηση αναζήτησης ξενοδοχείων, οι τιμές που παίρνουν είναι HotelBooking και OTA_HotelSearchRQ αντίστοιχα. Το String που περιέχει το XML έγγραφο εισάγεται σαν attachment στο ebXML μήνυμα.

Το ebXML μήνυμα αποστέλλεται στον ebXML OTA Server και εν συνεχεία λαμβάνεται η απόκριση του ebXML OTA Server με το αντίστοιχο ebXML μήνυμα. Η εφαρμογή ανασύρει το XML έγγραφο που βρίσκεται σε μορφή String συνημμένο στο ebXML μήνυμα. Το String αυτό χρησιμοποιείται στη συνέχεια για την αντιστοίχιση του XML εγγράφου σε ιεραρχία από αντικείμενα Java μέσω της τεχνικής Data Binding (μέθοδος Unmarshall). Τέλος η εφαρμογή χρησιμοποιεί τα Java αντικείμενα για την παρουσίαση των αποτελεσμάτων στην οθόνη.

6.4 Η Βάση Δεδομένων του Demo Client

Για την άμεση και εύκολη πρόσβαση σε πληροφορίες σχετικά με τις κρατήσεις που έχουν γίνει και τα προφίλ που έχουν δημιουργηθεί, υπάρχει μια μικρή βάση δεδομένων που αποτελείται από δύο πίνακες και αποθηκεύει πληροφορίες που βοηθούν στην χρήση της εφαρμογής (Εικόνα 63).



HotelReservation	
<input type="checkbox"/>	CpaID
<input type="checkbox"/>	HotelReservationUniqueID
<input type="checkbox"/>	HotelReservationID
<input type="checkbox"/>	HotelReservationConfirmationID
<input type="checkbox"/>	ProfileUniqueID
<input type="checkbox"/>	Surname
<input type="checkbox"/>	GivenName

Profile	
<input type="checkbox"/>	CpaID
<input type="checkbox"/>	ProfileUniqueID
<input type="checkbox"/>	Surname
<input type="checkbox"/>	GivenName

Εικόνα 63 Οι πίνακες της βάσης δεδομένων του Demo Client

Στον πίνακα HotelReservation αποθηκεύονται πληροφορίες σχετικά με τις κρατήσεις που έχουν γίνει. Συγκεκριμένα, το πεδίο CpaID κρατά το ID του CPA που χρησιμοποιήθηκε για την σύνδεση με τον ebXML OTA Server, το HotelReservationUniqueID κρατά την τιμή του πεδίου UniqueID του πίνακα Bookings της βάσης δεδομένων όπου αποθηκεύονται οι κρατήσεις στον ebXML OTA Server, και το ProfileUniqueID αναφέρεται στο προφίλ που συνδέθηκε με την κράτηση.

Στον πίνακα Profile αποθηκεύονται πληροφορίες σχετικά με τα προφίλ που έχουν κατασκευαστεί. Συγκεκριμένα, το πεδίο CpaID, όπως προαναφέρθηκε, κρατά το ID του CPA που χρησιμοποιήθηκε για την σύνδεση με τον ebXML OTA Server, το

ProfileUniqueID κρατά την τιμή του πεδίουID του πίνακα UniqueIDType της βάσης δεδομένων του προφίλ.

6.5 Ανακεφαλαίωση

Στο κεφάλαιο αυτό έγινε μια επισκόπηση της εφαρμογής που κατασκευάστηκε για να δοκιμαστεί το σύστημα του ebXML OTA Server. Αναφέρθηκε η βασική λειτουργικότητα της εφαρμογής και παρουσιάστηκαν στιγμιότυπα από το γραφικό περιβάλλον, καθώς και οι πίνακες της βάσης δεδομένων που χρησιμοποιεί.

Στο επόμενο κεφάλαιο περιγράφεται η εφαρμογή που αυτοματοποιεί τη διαδικασία κατασκευής σχεσιακής βάσης δεδομένων από ένα XML Schema και των κατάλληλων προγραμματιστικών βιβλιοθηκών για τη διαχείριση της βάσης.

Κεφάλαιο 7

Περιγραφή της εφαρμογής XML → DB+Middleware

7.1 Εισαγωγή

Η μεγάλη πολυπλοκότητα που χαρακτηρίζει τα XML Schemata του OTA, καθιστά εξαιρετικά χρονοβόρα και επιρρεπή σε λάθη τη διαδικασία της μετατροπής ενός XML Schema σε σχεσιακό σχήμα. Για να δοθεί μια λύση στο πρόβλημα αυτό σχεδιάστηκε και υλοποιήθηκε μια εφαρμογή που αυτοματοποιεί την παραπάνω διαδικασία. Επιπλέον, η εφαρμογή αυτή αυτοματοποιεί την κατασκευή των κατάλληλων stored procedures για την εισαγωγή, ανάκτηση και διαγραφή XML εγγράφων από τη σχεσιακή βάση δεδομένων που κατασκευάστηκε, και μιας βιβλιοθήκης σε Java που χρησιμοποιεί τα stored procedures. Έτσι δίνεται η δυνατότητα να εκτελούνται οι λειτουργίες διαχείρισης της αποθήκευσης των XML εγγράφων μέσω απλών εντολών της Java.

Η εφαρμογή XML → DB+Middleware χρησίμευσε ιδιαίτερα σε πολλά στάδια της υλοποίησης της παρούσας διπλωματικής εργασίας. Εκτός από τις βάσεις δεδομένων όπου η συνεισφορά της εφαρμογής ήταν καθοριστική, επιτάχυνε σημαντικά και την κατασκευή ορισμένων τμημάτων του γραφικού περιβάλλοντος της εφαρμογής Demo Client, χρησιμοποιώντας τμήματα του κώδικα για Web Graphical User Interface που κατασκεύασε. Η μεθοδολογία που ακολουθείται στην εφαρμογή (το μοντέλο δεδομένων που σχεδιάστηκε) χρησιμοποιήθηκε επίσης για την κατασκευή αναφοράς (report) για τους πίνακες των βάσεων δεδομένων, στην οποία βασίστηκε η κατασκευή του παραρτήματος όπου περιγράφονται αναλυτικά οι πίνακες των σχεσιακών βάσεων δεδομένων του συστήματος.

Η εφαρμογή στηρίζεται στην ανάλυση της δομής της ιεραρχίας Java κλάσεων που παράγεται με την μέθοδο DataBinding από ένα XML Schema. Η προσέγγιση αυτή καθιστά ευκολότερη την διαδικασία της μετατροπής του XML Schema σε σχεσιακό

σχήμα, καθώς έχει ήδη κατασκευαστεί ένα μοντέλο δεδομένων που αναπαριστά το XML Schema (η ιεραρχία Java κλάσεων). Έτσι αποφεύγεται το στάδιο της ανάλυσης του αρχείου κειμένου που περιέχει το XML Schema. Επιπλέον παρέχεται η δυνατότητα άμεσης σύνδεσης των XML εγγράφων και της σχεσιακής βάσης δεδομένων με τον Java κώδικα που θα κατασκευαστεί για τις λειτουργίες διαχείρισης της βάσης.

7.2 Επισκόπηση της διαδικασίας που απαιτείται πριν τη χρήση της εφαρμογής

Χρησιμοποιώντας ένα XML Data Binding tool (Breeze) κατασκευάζουμε μια ιεραρχία από Java κλάσεις για το XML Schema. Αυτές οι Java κλάσεις παρέχουν μεθόδους για Marshal / UnMarshal (Java → XML / XML → Java) καθώς και για validation ενός XML document (επαλήθευση της συμβατότητας του ως προς το αντίστοιχο XML Schema). Το Breeze κατασκευάζει Java κλάσεις για όλα τα στοιχεία (Elements) και σύνθετους τύπους (Complex Types) που περιέχονται σε ένα αρχείο περιγραφής ενός XML σχήματος είτε άμεσα είτε έμμεσα μέσω άλλων αρχείων που συμπεριλαμβάνονται σε αυτό.

Στη συνέχεια μεταγλωττίζουμε τα Java αρχεία που δημιουργήθηκαν και παίρνουμε ένα σύνολο από “.class” αρχεία. Αυτά τα “.class” αρχεία χρησιμοποιούνται σαν είσοδος στην εφαρμογή (είτε στην αρχική μορφή – δηλαδή φάκελος με αρχεία- ή συμπιεσμένα σε ένα “.jar” αρχείο).

7.3 Επισκόπηση της λειτουργίας της εφαρμογής

Συνοπτικά η εφαρμογή ακολουθεί τα παρακάτω βήματα:

1. Αξιοποιεί την πληροφορία που παρέχεται από το Reflection API [18] της Java για να κατασκευάσει ένα δέντρο και στη συνέχεια έναν γράφο για το XML Schema από τις Java κλάσεις που δημιούργησε το Breeze. Το Reflection API χρησιμοποιείται για την δυναμική επισκόπηση των χαρακτηριστικών αυτών των Java κλάσεων (το όνομα της κλάσης, η κλάση την οποία επεκτείνει, τα ονόματα των μεθόδων της κλάσης, οι οποίες αντιστοιχούν στα υποστοιχεία (SubElements) ή στα χαρακτηριστικά (Attributes) του αντίστοιχου στοιχείου (Element), τα

ονόματα και ο τύπος δεδομένων των παραμέτρων των μεθόδων αυτών, καθώς και ο τύπος δεδομένων που επιστρέφουν αυτές οι μέθοδοι).

2. Συμπληρώνει τις πληροφορίες που χρειάζεται μέσω ενός XML Schema Parser της Oracle. Οι πληροφορίες αυτές αφορούν τα σχόλια που περιέχονται στο XML Schema (Annotation), τον ακριβή τύπο κάθε ιδιότητας (XSDType) και το σύνολο τιμών που τυχόν μπορεί να πάρει μια ιδιότητα (Enumeration Values). Οι πληροφορίες που συλλέγονται αντιστοιχίζονται σε πεδία αντικειμένων που είναι στιγμιότυπα των παρακάτω κλάσεων:

- a. DBTable – η βασική κλάση που περιγράφει έναν πίνακα στην σχεσιακή βάση δεδομένων, και αντιστοιχεί σε ένα στοιχείο (Element) ή σύνθετο τύπο (Complex Type) του XML Schema. Τα πεδία της κλάσης είναι τα εξής:
 - i. Name – το όνομα του στοιχείου ή του σύνθετου τύπου.
 - ii. Extends – ο σύνθετος τύπος τον οποίο επεκτείνει.
 - iii. BelongingTables – τα στοιχεία στα οποία ανήκει (εάν το στοιχείο αυτό είναι παιδί ενός ή περισσότερων στοιχείων).
 - iv. Has – τα στοιχεία που είναι παιδιά αυτού του στοιχείου (μόνο εκείνα που είναι σύνθετου τύπου).
 - v. Col – οι ιδιότητες (attributes) καθώς και τα στοιχεία παιδιά του που είναι απλού τύπου.
 - vi. SelectionTables – οι πίνακες επιλογής (εάν το στοιχείο περιέχει δομή Choice).
 - vii. FinalTable – δείχνει εάν αυτό το στοιχείο πρέπει να αντιστοιχηθεί αυτούσιο σε πίνακα της βάσης δεδομένων (μπορεί να μην αντιστοιχηθεί το ίδιο αλλά ο σύνθετος τύπος τον οποίο επεκτείνει).
 - viii. Annotation – το σχόλιο που υπάρχει στο XML Schema για το στοιχείο αυτό.
- b. DBCol – η κλάση που περιγράφει τις ιδιότητες (attributes) καθώς και τα στοιχεία παιδιά ενός στοιχείου τα οποία είναι απλού τύπου. Τα πεδία της κλάσης είναι τα εξής:
 - i. Name – το όνομα της ιδιότητας ή του στοιχείου.
 - ii. Type – ο τύπος του (String, integer κτλ.).

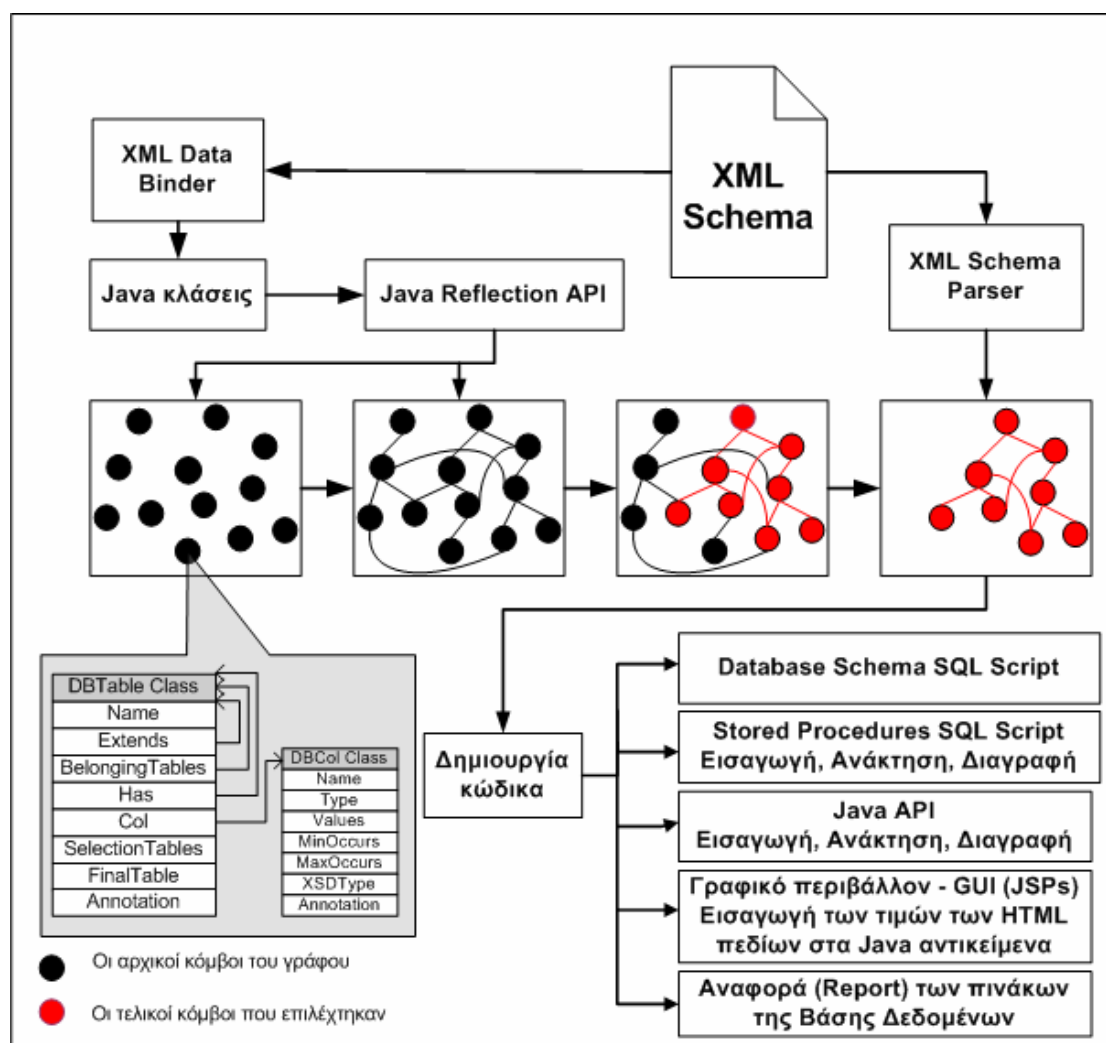
- iii. Values – το σύνολο των τιμών που τυχόν μπορεί να πάρει (εάν έχει EnumeratedValues).
- iv. MinOccurs – ο ελάχιστος αριθμός εμφανίσεων του στο XML έγγραφο.
- v. MaxOccurs – ο μέγιστος αριθμός εμφανίσεων του στο XML έγγραφο.
- vi. XSDType – ο τύπος του όπως ορίζεται στο XML Schema.
- vii. Annotation – το σχόλιο που υπάρχει στο XML Schema για την ιδιότητα ή το στοιχείο αυτό.

Το σύνολο των αντικειμένων των παραπάνω κλάσεων αναπαριστούν το XML Schema. Η αναπαράσταση αυτή έχει τη μορφή γράφου και όχι δέντρου, όπως είναι η μορφή ενός XML εγγράφου, επειδή τα στοιχεία που είναι του ίδιου τύπου αντιστοιχίζονται σε έναν κοινό κόμβο και δεν επαναλαμβάνονται όπως στο XML έγγραφο.

3. Στη συνέχεια ο γράφος αυτός χρησιμοποιείται για την κατασκευή του απαραίτητου κώδικα (Code Generation):
 - a. DB schema: ένα αρχείο με το SQL script για την κατασκευή των πινάκων της βάσης δεδομένων.
 - b. Stored Procedures: ένα αρχείο με το SQL script για την κατασκευή των απαραίτητων διαδικασιών (stored procedures) για την εισαγωγή, ανάκτηση και διαγραφή δεδομένων από τους πίνακες της βάσης δεδομένων.
 - c. Java κλάσεις: ένα σύνολο από Java κλάσεις που χρησιμοποιούν τις παραπάνω διαδικασίες (stored procedures), παρέχοντας έτσι μία προγραμματιστική βιβλιοθήκη (Java API) για τις λειτουργίες διαχείρισης της βάσης (εισαγωγή, ανάκτηση και διαγραφή).
 - d. Γραφικό περιβάλλον: παράγει ένα σύνολο από Java Server Pages αρχεία για την εισαγωγή δεδομένων στα αντικείμενα του Breeze.

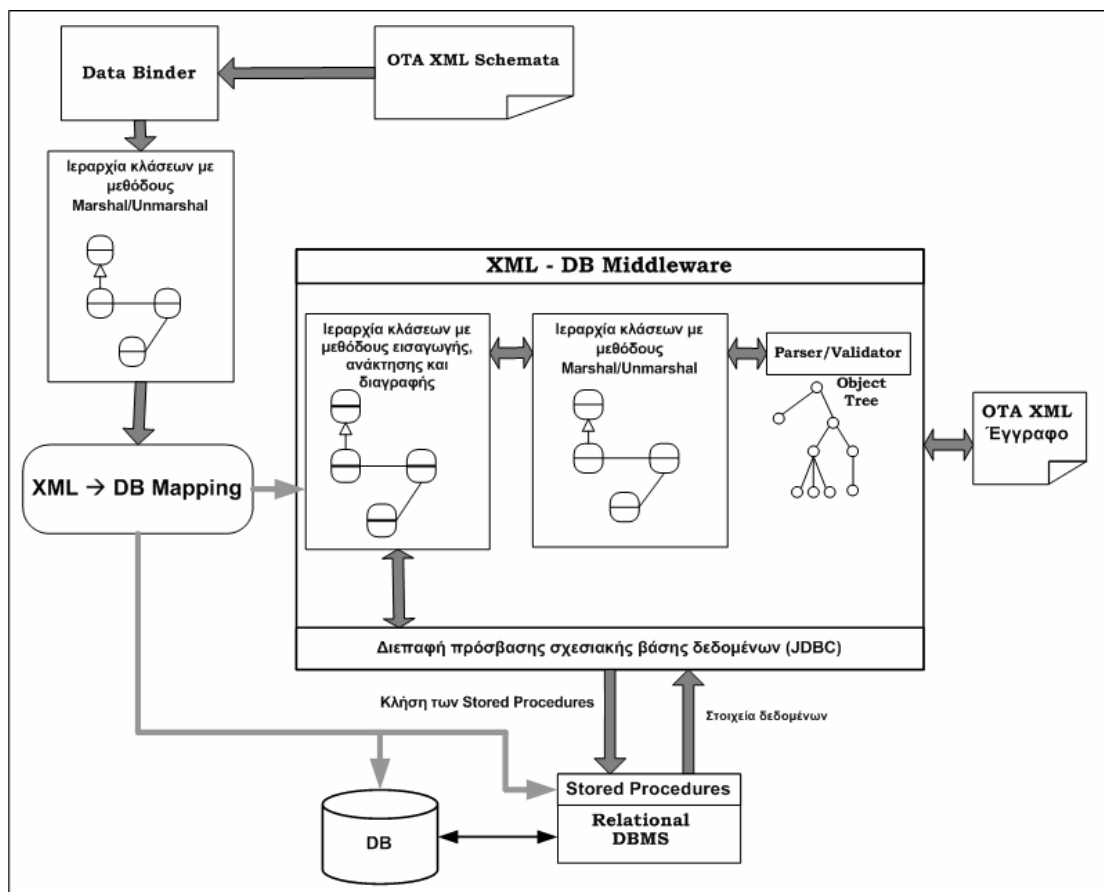
Η συνολική λειτουργία της εφαρμογής δίνεται στην Εικόνα 64. Το XML Schema μετατρέπεται σε ένα σύνολο από Java κλάσεις με την μέθοδο Data Binding. Τα χαρακτηριστικά των κλάσεων αυτών ανακτώνται μέσω του Reflection API και χρησιμοποιούνται για την κατασκευή αντικειμένων (objects) των κλάσεων DBTable και DBCol. Τα αντικείμενα αυτά εμφανίζονται στα τέσσερα πλαίσια στην παραπάνω

εικόνα σαν μαύροι και κόκκινοι κύκλοι. Στο πρώτο πλαίσιο εμφανίζεται ένα σύνολο από αντικείμενα που αντιστοιχούν στις Java κλάσεις που κατασκεύασε η μέθοδος Data Binding. Στο δεύτερο πλαίσιο εμφανίζονται οι σχέσεις που χτίζονται μεταξύ των αντικειμένων, σχηματίζοντας έτσι έναν γράφο με κόμβους τα παραπάνω αντικείμενα. Στο τρίτο πλαίσιο εμφανίζεται η δυνατότητα διαγραφής ενός υποσυνόλου του γράφου με ταυτόχρονη αναδιάταξη των σχέσεων μεταξύ των κόμβων. Στο τέταρτο πλαίσιο δίνεται ο τελικός γράφος που προκύπτει και θα χρησιμοποιηθεί στη συνέχεια για την κατασκευή του κώδικα για την βάση δεδομένων.



Εικόνα 64 Η αρχιτεκτονική της εφαρμογής XML → DB+Middleware

Παρακάτω (Εικόνα 65) δίνεται μια επισκόπηση της χρήσης της εφαρμογής για την κατασκευή των επιμέρους τμημάτων, και πώς αυτά συνεργάζονται μεταξύ τους για την διαχείριση XML εγγράφων στην βάση δεδομένων.



Εικόνα 65 Επισκόπηση της χρήσης της εφαρμογής για την διαχείριση XML εγγράφων στην βάση δεδομένων.

Το Reflection API είναι μια βιβλιοθήκη της Java που παρέχει πληροφορίες σε πραγματικό χρόνο για ένα αντικείμενο ή / και την κλάση του αντικειμένου. Μπορούμε έτσι να πάρουμε πληροφορίες για τις μεθόδους (όνομα, τύπος δεδομένων που επιστρέφεται, όνομα και τύπος των παραμέτρων), τα πεδία (όνομα και τύπος), τους constructors / destructors, ποια κλάση κληρονομεί κ.α. Μπορούμε επίσης να καλέσουμε τις δημόσιες (public) μεθόδους (όταν μελετάμε αντικείμενο) ή να δημιουργήσουμε ένα καινούργιο αντικείμενο (όταν μελετάμε κλάση).

Μελετώντας λοιπόν μια κλάση (που αντιστοιχεί ουσιαστικά σε ένα στοιχείο ή σύνθετο τύπο του XML Schema) αποκτούμε αρκετή από την πληροφορία που χρειαζόμαστε για το στοιχείο (ή σύνθετο τύπο) καθώς και τα στοιχεία παιδιά του. Η διαδικασία, που τελικά μας δίνει το δέντρο του XML Schema, αποτελείται από τα παρακάτω βήματα:

- Αρχικά για κάθε “.class” αρχείο ανακτούμε το όνομα της κλάσης (μέσω της μεθόδου `getName()` που παρέχει το Reflection API) και την κλάση (ή κλάσεις) που κληρονομεί και τα προσθέτουμε σε μια λίστα (η λίστα αποτελείται από αντικείμενα `DBTable` που θα εξηγηθούν παρακάτω).
- Στη συνέχεια για κάθε αρχείο ανακτούμε τις μεθόδους που παρέχει. Από αυτές τις μεθόδους αφαιρούμε αυτές που το όνομα τους αρχίζει από “get” και δεν ανήκουν σε εκείνες που χρησιμοποιούνται συγκεκριμένα από το Breeze (θα αναφερθούν παρακάτω) καθώς και αυτές που είναι κοινές σε όλες γενικά τις java κλάσεις. Από τις υπόλοιπες μεθόδους ανακτούμε το όνομά τους και τον τύπο που επιστρέφουν. Από τον τύπο που επιστρέφουν βλέπουμε εάν πρόκειται για ιδιότητα (Attribute) ή υποστοιχείο απλού τύπου (SimpleType Sub Element), εάν επιστρέφουν απλό τύπο δεδομένων (ακέραιο, αλφαριθμητικό κτλ.) ή υποστοιχείο σύνθετου τύπου (Complex Type Sub Element), εάν επιστρέφουν κάποια κλάση. Εάν μια μέθοδος επιστρέφει τύπο δεδομένων `java.util.Vector` καταλαβαίνουμε ότι πρόκειται για πολλαπλό στοιχείο το οποίο μπορούμε να το δούμε εξετάζοντας τον τύπο που επιστρέφει η μέθοδος `get<όνομα μεθόδου>At`. Εάν στο σύνολο των αρχικών μεθόδων υπάρχουν μέθοδοι που το όνομα τους αρχίζει από “is” καταλαβαίνουμε ότι σε αυτό το στοιχείο υπάρχει δομή επιλογής (Choice). Τα ονόματα των στοιχείων της δομής επιλογής τα παίρνουμε από τα ονόματα των μεθόδων τα οποία αρχίζουν από “is”.
- Κατόπιν διαμορφώνουμε τις σχέσεις μεταξύ των στοιχείων. Για κάθε στοιχείο της λίστας βλέπουμε ποια στοιχεία παιδιά έχει, ποιον σύνθετο τύπο επεκτείνει και προσθέτουμε αυτές τις πληροφορίες στα αντίστοιχα στοιχεία της λίστας. Έτσι, στο τέλος αυτής της διαδικασίας μπορούμε να ξέρουμε για κάθε στοιχείο ποιον σύνθετο τύπο επεκτείνει, ποιοι σύνθετοι τύποι επεκτείνονται από αυτόν, εάν αναφέρεται από άλλα στοιχεία σαν υποστοιχεία ποια είναι αυτά και αν μόνο το ίδιο αναφέρεται σε αυτά τα υποστοιχεία ή υπάρχουν και άλλα στοιχεία που αναφέρονται στα ίδια υποστοιχεία.
- Έπειτα, ξεκινώντας από ένα αρχικό στοιχείο ή σύνθετο τύπο, διατρέχουμε όλα τα υποστοιχεία και κατασκευάζουμε ένα δέντρο. Αφαιρούμε έτσι όλη την υπόλοιπη

πληροφορία που πήραμε από τα αρχεία της Java και η οποία είναι περιττή. Ακολουθεί μια αναδιαμόρφωση των σχέσεων για να αφαιρεθούν αναφορές σε στοιχεία που δεν ανήκουν στην καινούργια λίστα.

Στη συνέχεια αναλύεται (parse) το XML Schema με ένα πρόγραμμα ανάλυσης XML Schema της εταιρίας Oracle. Η ανάλυση αυτή αποσκοπεί στην ανάκτηση πρόσθετων πληροφοριών όπως είναι το μέγεθος της πολλαπλότητας ενός στοιχείου (MinOccurs, MaxOccurs) και ο ακριβής τύπος (XSDType) μιας ιδιότητας (Attribute) ενός στοιχείου.

Το τελικό βήμα είναι η δημιουργία του κώδικα που κατασκευάζει την βάση δεδομένων, τις ενσωματωμένες διαδικασίες (stored procedures), την προγραμματιστική βιβλιοθήκη (Java API) και το γραφικό περιβάλλον (Web GUI)

- a. Βάση Δεδομένων: Χρησιμοποιώντας απλούς κανόνες μετατροπής XML σχήματος σε σχεσιακό σχήμα, όπως αυτούς που προτείνει ο Bourett [3], δημιουργούνται οι εντολές για την κατασκευή των πινάκων. Οι κανόνες που χρησιμοποιήθηκαν είναι οι παρακάτω:
 - i. Όταν ένα στοιχείο έχει στοιχεία παιδιά που είναι απλού τύπου τότε αυτά τα υποστοιχεία μαζί με τις ιδιότητες (Attributes) του στοιχείου γίνονται πεδία του πίνακα. Τα υποστοιχεία που είναι σύνθετου τύπου (Complex Type) γίνονται ξεχωριστοί πίνακες.
 - ii. Εάν τα υποστοιχεία έχουν πολλαπλότητα 0 ή 1 τότε στον πίνακα του στοιχείου πατέρα (Parent Element) μπαίνει ένα πεδίο σαν foreign key στους πίνακες αυτών των υποστοιχείων που το όνομα του έχει την μορφή “ParentElement__Has__SubElement”, ενώ εάν έχουν πολλαπλότητα μεγαλύτερη του 1 τότε:
 - iii. Εάν ένα υποστοιχείο έχει μόνο ένα στοιχείο πατέρα (Parent Element) τότε στον πίνακα του υποστοιχείου μπαίνει ένα πεδίο σαν foreign key στο στοιχείο πατέρα, που το όνομα του έχει τη μορφή “SubElement__BelongsTo__ParentElement”. Διαφορετικά για κάθε στοιχείο πατέρα του υποστοιχείου δημιουργείται μια καινούργια σχέση (πίνακας) με το όνομα “ParentElement__Has__SubElement”.

- iv. Όταν ένα στοιχείο επεκτείνει κάποιον σύνθετο τύπο, τότε στον πίνακα αυτού του στοιχείου μπαίνει ένα πεδίο σαν foreign key στον πίνακα του σύνθετου τύπου, που το όνομα του έχει τη μορφή “Element__Extends__ComplexType”.
- v. Όταν ένα στοιχείο περιέχει μια δομή απλής επιλογής (Choice) τότε στον πίνακα του στοιχείου προσθέτονται δύο επιπλέον πεδία για κάθε στοιχείο επιλογής (Choice Element), ένα για το όνομα του κάθε στοιχείου επιλογής και ένα που χρησιμοποιείται για να δείχνει στον πίνακα του κάθε στοιχείου επιλογής (ID).
- vi. Εάν η δομή επιλογής δεν είναι απλή αλλά είναι επαναλαμβανόμενη (Repeating), τότε κατασκευάζεται ένας καινούργιος πίνακας που έχει τρία πεδία: ένα για το όνομα του στοιχείου επιλογής, ένα για να δείχνει στον πίνακα του στοιχείου επιλογής (ID) και ένα foreign key στο στοιχείο με όνομα της μορφής “RepeatingCoice__BelongsTo__Element”.
- vii. Όταν μια ιδιότητα (Attribute) παίρνει συγκεκριμένες τιμές (Enumerated), τότε μετά τη δήλωση του πεδίου στον πίνακα προστίθεται ένας περιορισμός (constraint) στο πεδίο του πίνακα ώστε να δέχεται μόνο κάποια από τις τιμές αυτές.

Παρακάτω (Εικόνα 66) δίνεται παράδειγμα του κώδικα SQL που παράγεται για την κατασκευή των πινάκων της βάσης δεδομένων

```

Create Table ElementType
(
  ElementTypeID int primary key,
  ElementType__Extends__ElementTypeSuperType int foreign key references...,
  ElementType__Belongs__ElementTypeParent int foreign key references ...,
  Attribute1 Arribute1Type,
  SimpleTypeElement1 SimpleTypeElement1Type,
  ElementType__Has__ComplexElement1 int foreign key references ...,
)

Create Table ElementType__Has__ElementType2
(
  ElementTypeID int not null foreign key references [ElementType]([ElementTypeID]),
  ElementType2ID int not null foreign key references [ElementType2]([ElementType2ID])
)

```

Εικόνα 66 Παραδείγματα του κώδικα SQL για την κατασκευή των πινάκων της βάσης δεδομένων.

b.Stored Procedures: Δημιουργούνται οι ρουτίνες για εισαγωγή, ανάκτηση και διαγραφή στη βάση για κάθε στοιχείο και σύνθετο τύπο που έχει γίνει

αντιστοιχηθεί. Παραδείγματα κώδικα των ρουτινών αυτών δίνονται παρακάτω (Εικόνα 67)

```
create proc InsertElementType
@ElementType__Belongs__ElementTypeParent int,
@Attribute1 Attribute1Type,
@SimpleTypeElement1 SimpleTypeElement1Type

as
insert into [ElementType] values (
  @ElementType__Belongs__ElementTypeParent,
  @Attribute1 ,
  @SimpleTypeElement1
)
```

```
create proc ReadElementType
@ElementTypeID int,

as
select * from [ElementType] where [ElementTypeID] = @ElementTypeID
```

```
create proc DeleteElementType
@ElementTypeID int,

as
delete from [ElementType] where [ElementTypeID] = @ElementTypeID
```

Εικόνα 67 Παραδείγματα του κώδικα SQL για την κατασκευή των διαδικασιών εισαγωγής, ανάκτησης και διαγραφής από τη βάση δεδομένων (stored procedures).

c. Java κλάσεις: Δημιουργείται ένα αρχείο Java για κάθε στοιχείο ή σύνθετο τύπο.

Αυτό το αρχείο περιλαμβάνει:

- i. Constructor που παίρνει σαν παράμετρο ένα αντικείμενο σύνδεσης με βάση δεδομένων (JDBC Connection object) για να γίνεται η σύνδεση με την βάση δεδομένων
- ii. Μία μέθοδο εισαγωγής (Write) που παίρνει σαν παράμετρο ένα αντικείμενο Breeze του ίδιου στοιχείου (ή σύνθετου τύπου) και επιστρέφει έναν ακέραιο αριθμό που είναι το πρωτεύον κλειδί (primary key) της εγγραφής που έγινε στον πίνακα.
- iii. Μία μέθοδο εισαγωγής (Write) που παίρνει σαν παράμετρο ένα σύνολο (Vector) από αντικείμενα Breeze του ίδιου στοιχείου (ή σύνθετου τύπου) και επιστρέφει ένα σύνολο (Vector) από ακέραιους αριθμούς που είναι τα πρωτεύοντα κλειδιά των εγγραφών που έγιναν στον πίνακα.
- iv. Μια μέθοδο ανάκτησης (Read) που παίρνει σαν παράμετρο έναν ακέραιο αριθμό που δείχνει στο πρωτεύον κλειδί της εγγραφής του πίνακα που

θέλουμε να διαβάσουμε, και επιστρέφει ένα αντικείμενο Breeze του ίδιου στοιχείου (ή σύνθετου τύπου).

- v. Μια μέθοδο ανάκτησης (Read) που παίρνει σαν παράμετρο ένα σύνολο από ακέραιους αριθμούς που δείχνουν στα πρωτεύοντα κλειδιά των εγγραφών του πίνακα που θέλουμε να διαβάσουμε, και επιστρέφει ένα σύνολο από αντικείμενα Breeze του ίδιου στοιχείου (ή σύνθετου τύπου).
- vi. Μια μέθοδο διαγραφής (Delete) που παίρνει σαν παράμετρο έναν ακέραιο αριθμό που δείχνει στο πρωτεύον κλειδί της εγγραφής του πίνακα που θέλουμε να διαγράψουμε
- vii. Μια μέθοδο διαγραφής (Delete) που παίρνει σαν παράμετρο ένα σύνολο από ακέραιους αριθμούς που δείχνουν στα πρωτεύοντα κλειδιά των εγγραφών του πίνακα που θέλουμε να διαγράψουμε. Παρακάτω δίνεται ένα παράδειγμα του κώδικα Java που παράγεται (Εικόνα 68)

```
package package_name;
import java.sql.*;
import java.util.*;
public class ElementType extends java.lang.Object {
    private Connection conn;
    private ResultSet rs;

    public ElementType (Connection Conn) {
        conn = Conn;
    }

    public int WriteElementType(element_package.ElementType elementtype) {
    }
    public Vector WriteElementType(Vector ElementTypeVector) {
    }
    public element_package.ElementType ReadElementType(int ElementTypeID) {
    }
    public Vector ReadElementType(Vector ElementTypeIDs) {
    }
    public void DeleteElementType(int ElementTypeID) {
    }
    public void DeleteElementType(Vector ElementTypeIDs) {
    }
}
```

Εικόνα 68 Παράδειγμα του κώδικα Java για την κατασκευή των διαδικασιών εισαγωγής, ανάκτησης και διαγραφής από τη βάση δεδομένων (Java API).

- d. Γραφικό περιβάλλον (Web GUI): Κατασκευάζεται ένα σύνολο από JSP αρχεία που περιέχουν τον απαραίτητο κώδικα και τα πεδία εισόδου

δεδομένων, ώστε ένας χρήστης να μπορεί να εισαγάγει δεδομένα. Τα δεδομένα εισόδου ελέγχονται μέσω javascript για την ορθότητά τους σύμφωνα με το XML Schema (δεν υπάρχει ακόμα πλήρης έλεγχος, κυρίως ελέγχεται το MaxOccurs και το μήκος του string όπου χρειάζεται). Στη συνέχεια φορτώνονται επάνω στα κατάλληλα αντικείμενα Breeze. Στην Εικόνα 69 δίνεται ένα στιγμιότυπο από το γραφικό περιβάλλον που κατασκεύασε η εφαρμογή για το προφίλ χρήστη.

The screenshot shows a web application interface for creating a customer profile. On the left, a tree view lists the data structure under 'ProfileType', including 'Accesses', 'Customer', 'Address', 'URL', 'Telephone', 'Email', 'CitizenCountryName', 'Document', 'CustLoyalty', 'PaymentForm', 'ContactPerson', 'EmployeeInfo', 'RelatedTraveler', 'PrefCollections', 'CompanyInfo', 'Affiliations', and 'Agreements'. The main area is titled 'Customer' and contains the following fields:

- BirthDate: Three dropdown menus for day, month, and year.
- CurrencyCode: A text input field.
- DecimalPlaces: A text input field.
- Gender: A dropdown menu.
- Deceased: A checkbox.
- LockoutType: A text input field.
- PhysChallName: A text input field.
- PetInfo: A text input field.
- PersonName: A sub-form containing:
 - SurnamePrefix: A text input field.
 - Surname*: A text input field.
 - ShareSynchInd: A dropdown menu.
 - ShareMarketInd: A dropdown menu.
 - NameType (OTA Codes): A dropdown menu.
 - NamePrefix: A text input field.
 - GivenName: A text input field.
 - MiddleName: A text input field.
 - NameSuffix: A text input field.
 - NameTitle: A text input field.

On the right side of the form, there are four buttons: 'Submit', 'Close', 'View XML', and 'Save to DB'.

Εικόνα 69 Στιγμιότυπο του γραφικού περιβάλλοντος για το προφίλ χρήστη που κατασκευάστηκε αυτόματα από την εφαρμογή XML → DB+Middleware

7.4 Περιγραφή της κλάσης ObjectCopier

Η μεθοδολογία διαχείρισης των XML εγγράφων που χρησιμοποιήθηκε στην υλοποίηση της εργασίας, δηλαδή η αναπαράσταση των XML εγγράφων σε ιεραρχία αντικειμένων, παρουσίαζε μια εγγενή δυσκολία που έπρεπε να αντιμετωπιστεί: την μεταφορά αυτούσιων τμημάτων XML εγγράφων από ένα XML έγγραφο σε ένα άλλο. Για παράδειγμα οι πληροφορίες που περιέχονταν στο XML έγγραφο της αίτησης για έναρξη της συνδιαλλαγής μιας νέας κράτησης, μεταφέρονταν αυτούσιες στο XML

έγγραφο της απόκρισης στην αίτηση. Τα δύο όμως αυτά XML έγγραφα αντιστοιχίζονται σε διαφορετικές ιεραρχίες αντικειμένων, με αποτέλεσμα να μην μπορεί να γίνει απλά τροποποίηση τους (cast).

Η λύση στο παραπάνω πρόβλημα, που ενέκυπτε σε πολλά σημεία της υλοποίησης, δόθηκε με την ανάπτυξη της κλάσης ObjectCopier. Η κλάση αυτή αναλαμβάνει να αντιγράψει τα δεδομένα που περιέχονται στα αντικείμενα μιας ιεραρχίας κλάσεων στα αντικείμενα μιας άλλης ιεραρχίας κλάσεων. Η κλάση χρησιμοποιεί το Reflection API για τον εντοπισμό των κλάσεων και των μεθόδων τους, και την κατασκευή αντικειμένων που είναι στιγμιότυπα των κλάσεων αυτών. Εντοπίζει δηλαδή τις κλάσεις που είναι όμοιες με τις αρχικές κλάσεις (όσον αφορά τον όνομα και τον τύπο των πεδίων τους), κατασκευάζει αντικείμενα των κατάλληλων κλάσεων, και αντιγράφει τα δεδομένα από την μία ιεραρχία αντικειμένων στην άλλη.

7.5 Ανακεφαλαίωση

Στο κεφάλαιο αυτό έγινε αναλυτική παρουσίαση της εφαρμογής XML – DB+Middleware. Περιγράφηκαν όλα τα στάδια της εφαρμογής καθώς και οι κανόνες στους οποίους στηρίζεται η λειτουργία της. Αναφέρθηκε επίσης η αναγκαιότητα της εφαρμογής καθώς και η συνεισφορά της στα διάφορα στάδια υλοποίησης της διπλωματικής εργασίας. Τέλος περιγράφηκε η κλαση ObjectCopier για την εύκολη αντιγραφή αντικειμένων μιας ιεραρχίας κλάσεων σε αντικείμενα μιας άλλης ιεραρχίας κλάσεων.

Στο επόμενο και τελευταίο κεφάλαιο γίνεται μια συνοπτική ανακεφαλαίωση της εργασίας, καθώς και πιθανές μελλοντικές επεκτάσεις.

Κεφάλαιο 8

Ανακεφαλαίωση και Μελλοντικές Επεκτάσεις

8.1 Ανακεφαλαίωση

Στην παρούσα εργασία μελετήθηκαν τα πρότυπα της ebXML και του OTA και η εφαρμογή τους στην ανάπτυξη ενός συστήματος παροχής υπηρεσιών διαδικτύου για τον τουρισμό.

Η ebXML είναι ένα πρότυπο για το ηλεκτρονικό εμπόριο στο διαδίκτυο. Χρησιμοποιείται για την επίτευξη διεπιχειρησιακών συνεργασιών (B2B) και την παροχή σύνθετων υπηρεσιών (B2B και B2C) στο διαδίκτυο. Ανήκει στην κατηγορία των πρωτοκόλλων συγχρονισμού (coordination protocol) και είναι κάθετο πρωτόκολλο καθώς σχεδιάστηκε και αναπτύχθηκε για το ηλεκτρονικό επιχειρείν (e-business). Τα διεθνή πρότυπα παροχής υπηρεσιών όπως η ebXML θέτουν τις βάσεις για την δημιουργία ενός διαλειτουργικού περιβάλλοντος διεπιχειρησιακών συνεργασιών.

Για να είναι πιο αποτελεσματική η διαλειτουργικότητα και η συνεργασία μεταξύ των επιχειρήσεων, οι ανταλλαγές μηνυμάτων πρέπει να στηρίζονται σε οντολογίες του πεδίου γνώσης (domain ontologies), που δίνουν ακριβή σημασιολογία στα μηνύματα που ανταλλάσσονται. Για τον τουριστικό τομέα τέτοια πηγή γνώσης και σημασιολογικής πληροφορίας συναντάμε στις προδιαγραφές του Open Travel Alliance (OTA). Ο OTA είναι ένας οργανισμός που αναπτύσσει προδιαγραφές διαλειτουργικότητας για την τουριστική βιομηχανία. Αποτελούμενος από τις μεγαλύτερες επιχειρήσεις και οργανισμούς στο χώρο του τουρισμού, συγκεντρώνει σημαντικό όγκο σημασιολογικής πληροφορίας και γνώσης που απορρέουν από την πολύχρονη εμπειρία των μελών που τον απαρτίζουν. Η γνώση αυτή αντανακλάται σε ένα σύνολο προδιαγραφών που περιγράφουν τα μηνύματα που ανταλλάσσονται μεταξύ των συνεργατών για την εκτέλεση των τουριστικών υπηρεσιών. Η περιγραφή

των μηνυμάτων γίνεται με την XML Schema, που είναι γλώσσα περιγραφής XML εγγράφων.

Συνδυάζοντας τις προδιαγραφές της ebXML για την εκτέλεση ηλεκτρονικών συνεργασιών με τις προδιαγραφές του OTA για την περιγραφή των ανταλλασσόμενων μηνυμάτων, επιτυγχάνεται η δημιουργία διαλειτουργικών διαδικτυακών υπηρεσιών για τον τουρισμό. Οι υπηρεσίες αυτές εμπεριέχουν τόσο τον συντακτικό καθορισμό των μηνυμάτων που ανταλλάσσονται, όσο και την σημασιολογική πληροφορία για την αμοιβαία κατανόηση της σημαντικής των υπηρεσιών μεταξύ των συνεργατών.

Όσον αφορά την υλοποίηση που έγινε στα πλαίσια της εργασίας, χρησιμοποιήθηκαν οι προδιαγραφές διαλειτουργικότητας της ebXML και του OTA για την υλοποίηση υπηρεσιών αναζήτησης ξενοδοχείων, αναζήτησης διαθεσιμότητας, διαχείρισης κρατήσεων ξενοδοχείων και διαχείρισης των προφίλ πελατών. Συγκεκριμένα χρησιμοποιήθηκε ένα πληροφοριακό σύστημα για τις τουριστικές επιχειρήσεις της Κρήτης, πάνω στο οποίο εφαρμόστηκαν οι προδιαγραφές της ebXML και του OTA για την κατασκευή Web Services.

Τα OTA Web Services που υλοποιήθηκαν υποστηρίζουν τις παρακάτω λειτουργίες:

- Αναζήτηση ξενοδοχείων (με εφαρμογή στην περιοχή της Κρήτης) χρησιμοποιώντας τα κριτήρια αναζήτησης που περιέχονται στις προδιαγραφές του OTA, και παρουσίαση των αποτελεσμάτων στον χρήστη.
- Αναζήτηση διαθέσιμων δωματίων σε ένα ή περισσότερα ξενοδοχεία, σύμφωνα με τα κριτήρια που ορίζει ο OTA. Η αναζήτηση διαθεσιμότητας μπορεί να συνδυαστεί με τις προτιμήσεις που ορίζονται στο προφίλ του πελάτη, περιορίζοντας έτσι το πλήθος των αποτελεσμάτων.
- Κράτηση δωματίων σε ένα ή περισσότερα ξενοδοχεία.
- Διαχείριση υπάρχουσας κράτησης από τον πελάτη:
 - Ανάκτηση των δεδομένων της κράτησης.
 - Τροποποίηση ή
 - Ακύρωση.
- Διαχείριση του προφίλ πελάτη:

- ο Δημιουργία
- ο Τροποποίηση
- ο Διαγραφή

Χρησιμοποιήθηκε η βάση δεδομένων ενός υπάρχοντος πληροφοριακού συστήματος για τις τουριστικές επιχειρήσεις της Κρήτης για την αναζήτηση ξενοδοχείων, δίνοντας έτσι τη δυνατότητα να ελεγχθεί η λειτουργικότητα του συστήματος σε πραγματικές συνθήκες.

Σχεδιάστηκε και αναπτύχθηκε σχεσιακή βάση δεδομένων για την διαχείριση των κρατήσεων δωματίων και την αναζήτηση διαθεσιμότητας. Η βάση δεδομένων σχεδιάστηκε χρησιμοποιώντας τις έννοιες και την δομή των XML σχημάτων του OTA, και κυρίως του XML Schema που περιγράφει την κράτηση ξενοδοχείων (OTA_HotelReservation).

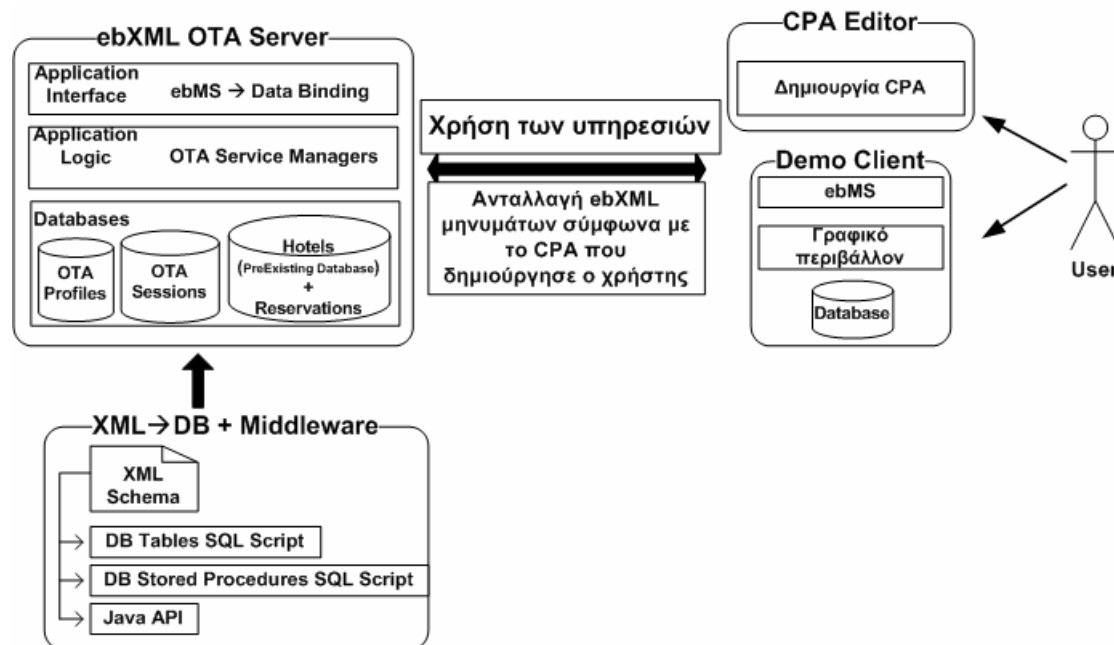
Για το προφίλ πελάτη κατασκευάστηκε μια βάση δεδομένων που καλύπτει το αντίστοιχο XML Schema του OTA σε όλη του την έκταση. Έγινε δηλαδή πλήρης αντιστοίχιση του XML Schema σε σχεσιακό σχήμα. Η ύπαρξη του προφίλ είναι ιδιαίτερα χρήσιμη στην ανάπτυξη νέων τουριστικών υπηρεσιών υψηλού επιπέδου, όπως η εξατομίκευση των αποτελεσμάτων αναζήτησης διαθεσιμότητας. Επιτυγχάνεται έτσι καλύτερη εξυπηρέτηση του πελάτη, αυξάνοντας τη εμπιστοσύνη του προς τις τουριστικές επιχειρήσεις.

Για την διευκόλυνση στην κατασκευή των παραπάνω σχεσιακών βάσεων δεδομένων , υλοποιήθηκε μια εφαρμογή που αυτοματοποιεί τη διαδικασία της μετατροπής ενός XML Schema – ή τμήματος αυτού – σε σχεσιακό σχήμα, καθώς την κατασκευή των απαραίτητων stored procedures και java κλάσεων για την διαχείριση των δεδομένων στη σχεσιακή βάση (εισαγωγή , ανάκτηση , διαγραφή).

Κατασκευάστηκε επίσης μια Web εφαρμογή για την επίδειξη της λειτουργικότητας του συστήματος, αποτελούμενη από ένα σύνολο JSP σελίδων για την εισαγωγή των δεδομένων και την παρουσίαση των αποτελεσμάτων. Η εφαρμογή επικοινωνεί με το σύστημα μέσω της υπηρεσίας μηνυμάτων της ebXML, χρησιμοποιώντας ένα CPA το

οποίο κατασκευάζει ο χρήστης με μια άλλη Web εφαρμογή που κατασκευάστηκε και ονομάζεται CPA Editor.

Τα τμήματα που υλοποιήθηκαν παρουσιάζονται συνοπτικά στην παρακάτω εικόνα:



8.2 Συμπεράσματα

Μέσα από την διαδικασία υλοποίησης της εργασίας, προέκυψαν ορισμένα συμπεράσματα τόσο όσον αφορά τις μεθόδους υλοποίησης όσο και τις ίδιες τις προδιαγραφές που χρησιμοποιήθηκαν.

Η διαχείριση των XML εγγράφων μέσω της μεθόδου Data Binding μπορούμε να πούμε ότι αποτελεί μονόδρομο σε υλοποιήσεις συστημάτων μεγάλης κλίμακας. Η αναπαράσταση των XML εγγράφων με ιεραρχίες αντικειμένων, ακολουθώντας την διαισθητική πρόσεγγιση του μοντέλου δεδομένων που περιγράφει το XML Schema και όχι την αντιμετώπιση του σαν μια απλή ιεραρχία από στοιχεία (Elements) και ιδιότητες (Attributes), προσφέρει πολλά πλεονεκτήματα: ευκολότερη και ταχύτερη υλοποίηση, συντήρηση και ενημέρωση του κώδικα, πιο ξεκάθαρο μοντέλο ως τμήμα ενός μεγαλύτερου συστήματος.

Η εφαρμογή **Breeze XML Data Binder** που χρησιμοποιήθηκε στην εργασία, ανταποκρίθηκε στις απαιτήσεις των περισσότερων τμημάτων της υλοποίησης, αλλά δεν κατάφερε να καλύψει το σύνολο των απαιτήσεων. Σε ορισμένα XML Schemata ο Java κώδικας που παρήγαγε η εφαρμογή δεν λειτούργησε με τον αναμενόμενο τρόπο, με αποτέλεσμα να χρειαστεί η συγγραφή επιπλέον κώδικα που να καλύπτει τα κενά λειτουργικότητας που απαιτούνταν. Το γεγονός αυτό αναιρεί τα πλεονεκτήματα της ίδιας της μεθόδου Data Binding. Το συμπέρασμα των παραπάνω είναι ότι η συγκεκριμένη εφαρμογή μπορεί να χρησιμοποιηθεί για σχετικά απλά XML Schemata, όμως δεν μπορεί να ανταποκριθεί αξιόπιστα στις απαιτήσεις μεγαλύτερων και πιο σύνθετων σχημάτων.

Ένα άλλο σημαντικό θέμα είναι η δυνατότητα σύνθετης διαχείρισης της XML και XML Schema, όπως είναι η αυτόματη κατασκευή βάσεων δεδομένων και κώδικα διαχείρισης των βάσεων αυτών και η κατασκευή γραφικού περιβάλλοντος για την σύνθεση XML εγγράφων που υπακούουν σε κάποιο XML Schema. Η εφαρμογή XML→DB+Middleware που αναπτύχθηκε στην εργασία, αποτελεί παράδειγμα τέτοιας λειτουργικότητας. Η χρήση τέτοιου είδους εφαρμογών μπορεί να βοηθήσει και να επιταχύνει σημαντικά την διαδικασία της υλοποίησης και της αναβάθμισης παρόμοιων συστημάτων.

Όσον αφορά στις προδιαγραφές του OTA πρέπει να σημειωθεί ότι υπήρξε πρόβλημα με την μη ικανοποιητική τεκμηρίωσή τους. Απαιτήθηκε ενδελεχής μελέτη των συνοδευτικών κειμένων όλων των εκδόσεων προκειμένου να διευκρινιστούν ορισμένες ασάφειες. Το πρόβλημα αυτό επιλύεται σταδιακά με την συνεχή έκδοση νέων προδιαγραφών και την κάλυψη πολλών κενών. Πέρα όμως από τα όποια προβλήματα υπάρχουν, η ύπαρξη ανοικτών προδιαγραφών διαλειτουργικότητας όπως του OTA συμβάλλει σημαντικά στην εξέλιξη των τουριστικών πληροφοριακών συστημάτων, τόσο μέσω της διαλειτουργικότητας μεταξύ τους, όσο και μέσω του εμπλουτισμού των υπάρχοντων συστημάτων με νέες δυνατότητες και πλούσιο περιεχόμενο.

Συνοψίζοντας τα παραπάνω συμπεράσματα που προέκυψαν από την διαδικασία της υλοποίησης, διαπιστώνουμε ότι η διαδικασία παροχής διαλειτουργικότητας

χρησιμοποιώντας τα πρότυπα της ebXML και του OTA, ακόμα και για υπάρχοντα τουριστικά συστήματα, είναι μεγάλη, πολύ χρονοβόρα γιατί πρέπει να αφομοιωθούν πολλές καινούργιες τεχνολογίες, και απαιτεί την ανάπτυξη μεγάλης ποσότητας λογισμικού.

Η υποστήριξη πολύπλοκων προτύπων όπως η ebXML και το OTA δημιουργεί πολλά προβλήματα στη φάση του σχεδιασμού και της υλοποίησης τα οποία απαιτούν προσεκτική μελέτη για την αποτελεσματική τους αντιμετώπιση. Στην παρούσα εργασία, τα προβλήματα αυτά αντιμετωπίστηκαν με τρόπο γενικό και δόθηκε ιδιαίτερη έμφαση στο να προκύψει μια γενική μεθοδολογία που να μπορεί να εφαρμοστεί με επιτυχία σε ανάλογα εγχειρήματα. Ως αποτέλεσμα της γενικής αυτής προσέγγισης προέκυψαν εργαλεία, τμήματα λογισμικού και οδηγίες που μπορούν να χρησιμεύσουν σε άλλες υλοποιήσεις και να μειώσουν σημαντικά το χρόνο υλοποίησης δίνοντας αποτελεσματικές λύσεις σε κρίσιμα τεχνικά προβλήματα. Έτσι, αν και στην παρούσα εργασία υλοποιήθηκε ένα μόνο τμήμα των προδιαγραφών του OTA, τέθηκαν οι βάσεις και αντιμετωπίστηκαν τα προβλήματα που μπορεί να προκύψουν σε μια μελλοντική πλήρη υλοποίηση τους σε ένα ολοκληρωμένο σύστημα.

Με βάση τα παραπάνω, ως στοιχεία της συνεισφοράς της παρούσας εργασίας αναφέρουμε:

- Την ενδελεχή μελέτη των προδιαγραφών της ebXML και του OTA, καταγράφοντας τους τρόπους που συνδυάζονται για την παροχή τουριστικών υπηρεσιών, καθώς και τα πλεονεκτήματα και τα μειονεκτήματα τους κατά την υλοποίηση τους.
- Την ανάπτυξη καινοτόμων υπηρεσιών όπως η εξατομίκευση των αποτελεσμάτων αναζήτησης διαθεσιμότητας μέσω των προτιμήσεων που περιέχονται στο προφίλ πελάτη, καθώς και η παροχή εναλλακτικών αποτελεσμάτων στην περίπτωση που η αρχική αναζήτηση δεν επιστρέψει αποτελέσματα.
- Την μεθοδολογία για την αυτόματη κατασκευή σχεσιακών βάσεων δεδομένων από XML schemata και την ανάπτυξη λογισμικού για τη διαχείριση των βάσεων δεδομένων του συστήματος.

- Τη γνώση και εμπειρία που αποκτήθηκε και η αξιολόγηση τόσο των προδιαγραφών όσο και των μεθόδων και των εργαλείων που χρησιμοποιήθηκαν.
- Την εφαρμοσιμότητα των τεχνικών που χρησιμοποιήθηκαν και των μεθόδων που αναπτύχθηκαν σε παρόμοια συστήματα μεγαλύτερης κλίμακας.
- Την αξιολόγηση της μεγάλης προσπάθειας που απαιτείται για την παροχή υπαρχόντων υπηρεσιών τουρισμού με πρότυπα υπηρεσιών ebXML και OTA
- Την αξιολόγηση τεχνικών γνώσεων και τεχνολογιών που απαιτούνται

8.4 Μελλοντικές Επεκτάσεις

Οι μελλοντικές επεκτάσεις της παρούσας εργασίας ομαδοποιούνται σε τρεις κατηγορίες:

- Επεκτάσεις οι οποίες αναφέρονται σε βελτιώσεις των αλγορίθμων αναζήτησης καθώς και των αλγορίθμων εξατομίκευσης αποτελεσμάτων αναζήτησης διαθεσιμότητας σύμφωνα με τις προτιμήσεις του χρήστη που ορίζονται στο προφίλ του.
- Επεκτάσεις που αφορούν την πληρέστερη υλοποίηση των υπηρεσιών του προτύπου OTA. Στην παρούσα εργασία υλοποιήθηκαν οι υπηρεσίες αναζήτησης ξενοδοχείων, διαθεσιμότητας και διαχείρισης κρατήσεων δωματίων σε ξενοδοχεία, καθώς η διαχείριση του προφίλ πελάτη. Οι προδιαγραφές του OTA όμως περιγράφουν ένα μεγάλο σύνολο τουριστικών υπηρεσιών, τόσο στον τομέα των ξενοδοχείων, όσο και σε άλλους τομείς της τουριστικής βιομηχανίας²⁰.
- Επεκτάσεις που αφορούν τη διασύνδεση του μοντέλου OTA με άλλα συστήματα που υποστηρίζουν διαλειτουργικότητα μέσω σύγχρονων μοντέλων βασισμένων σε σημασιολογικές περιγραφές αντικειμένων.

²⁰ Στην ενότητα 1.2 του πρώτου κεφαλαίου αναφέρονται οι τομείς της τουριστικής βιομηχανίας για τους οποίους περιγράφονται σχετικές υπηρεσίες στις προδιαγραφές του OTA.

8.3.1 Επεκτάσεις των αλγορίθμων αναζήτησης και εξατομίκευσης

Στην υλοποίηση των υπηρεσιών αναζήτησης ξενοδοχείων και διαθεσιμότητας, έχει γίνει μια συγκεκριμένη θεώρηση της σημασιολογίας των κριτηρίων αναζήτησης ξενοδοχείων και διαθεσιμότητας που οδήγησε σε συγκεκριμένους αλγορίθμους αναζήτησης. Αντίστοιχα, με μια ορισμένη θεώρηση της σημασιολογίας του προφίλ των χρηστών, όσον αφορά στις προτιμήσεις αναζήτησης διαθεσιμότητας, σχεδιάστηκαν συγκεκριμένοι αλγόριθμοι εξατομίκευσης των αποτελεσμάτων αναζήτησης διαθεσιμότητας σύμφωνα με τις προτιμήσεις του χρήστη που ορίζονται στο προφίλ του.

Οι παραπάνω αλγόριθμοι μπορούν να αποτελέσουν αντικείμενο ξεχωριστής μελέτης ώστε να προκύψουν εναλλακτικοί τρόποι αναζήτησης και εξατομίκευσης οι οποίοι στη συνέχεια θα μελετηθούν μεταξύ τους ως προς διάφορα κριτήρια απόδοσης. Αυτό αποτελεί ένα από τους βασικούς άξονες μελλοντικής εργασίας.

8.3.2 Πληρέστερη υλοποίηση των ΟΤΑ υπηρεσιών

Όπως αναφέρθηκε, το σύστημα διαχείρισης των κρατήσεων δεν συμπεριλαμβάνει compensations. Δηλαδή δεν γίνεται έλεγχος για τυχόν ποινές ακύρωσης κρατήσεων, προκαταβολικές πληρωμές και εγγυήσεις, και έλεγχο πιστωτικών καρτών. Τέτοιου είδους συνδιαλλαγές αποτελούν απαραίτητα τμήματα ενός ολοκληρωμένου ηλεκτρονικού συστήματος κρατήσεων ξενοδοχείων. Ένας μελλοντικός στόχος λοιπόν είναι να ενσωματωθούν αυτές οι δυνατότητες στο σύστημα.

Επιπλέον στο μέλλον θα μπορούσαν να υλοποιηθούν Web Services και για άλλα τουριστικά προϊόντα όπως ενοικίαση οχημάτων, κράτηση αεροπορικών θέσεων και ακτοπλοϊκών εισιτηρίων, καθώς και ο συνδυασμός τους για την παροχή πακέτων διακοπών.

8.3.3 Υποστήριξη διαλειτουργικότητας μέσω οντολογιών

Η σύγχρονη τάση για την υποστήριξη διαλειτουργικότητας στα πιο σημαντικά πεδία εφαρμογής, περιλαμβανομένου και του τουριστικού τομέα, είναι η περιγραφή των

σημαντικών υπηρεσιών του πεδίου και των σχετικών πληροφοριών με τη χρήση οντολογιών σε κάποια πρότυπη γλώσσα και η χρήση αυτών των οντολογιών από τις εφαρμογές για τη μεταξύ τους επικοινωνία. Σε ένα τέτοιο τεχνολογικό περιβάλλον, πρότυπα όπως το OTA, μπορούν να χρησιμοποιηθούν ως βάση για τον ορισμό οντολογιών που θα βασίζονται σε αυτά τα πρότυπα και θα εκφράζονται σε κάποια πρότυπη γλώσσα.

Μια μελλοντική επέκταση που προκύπτει άμεσα με αυτή τη λογική είναι ο καθορισμός μιας τέτοιας οντολογίας και οι αναγκαίοι μετασχηματισμοί που θα επιτρέπουν την αυτόματη μετατροπή των XML εγγράφων από τα OTA schemata προς την οντολογία αυτή και αντίστροφα.

Βιβλιογραφία

- [1] Alonso, G., Casati, F., Kuno, H. και Machiraju, V. Web Services Concepts, Architectures and Applications, Springer-Verlag, 2004, pp.354.
- [2] Amadeus, <http://www.amadeus.com/index.jsp>
- [3] Bourret, R. (2001) *Mapping W3C Schemas to Object Schemas to Relational Schemas*, <http://www.rpbourret.com/xml/SchemaMap.htm>
- [4] Breeze XML Binder, <http://www.breezefactor.com>
- [5] Cardoso J. (2004). *Semantic Web Processes and Ontologies for the Travel Industry*, AIG SIG SEMIS Bulletin Vol. 1, no 3, pp. 25-28, http://www.sigsemis.org/newsletter/october2004/ais_sigsemis_bulletin_vol1_issue3_2004.pdf
- [6] Dogac, A., *SATINE Project: Exploiting Web Services in the Travel Industry*, <http://www.srdc.metu.edu.tr/webpage/projects/satine>
- [7] Electronic Data Interchange (EDI), <http://www.unece.org/trade/untdid/>
- [8] electronic business XML (ebXML), <http://www.ebxml.org>
- [9] ebXML Business Process Specification Schema (BPSS), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-bp
- [10] ebXML Collaboration Protocol Profile and Agreement (CPPA), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-cppa
- [11] ebXML Message Service (ebMS), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-msg
- [12] ebXML Registry / Repository (Reg/Rep), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep
- [13] Federated European Tourism Information Systems Integration (FETISH), <http://www.fetish.t-6.it/>
- [14] Galileo, <http://www.galileo.com>
- [15] Galileo Web Services, <http://xml.coverpages.org/GalileoGlobalWS.html>
- [16] Harmonize Project, IST-2000-29329, Tourism Harmonization Network, <http://www.harmonise.org/>
- [17] Jakarta Apache Tomcat, <http://jakarta.apache.org/tomcat/index.html>
- [18] Java Reflection API, <http://java.sun.com/docs/books/tutorial/reflect/>

- [19] Nicolai, A. *Interoperation and Standards in the Computing Continuum Linking Online Travel and Tourism Services*, http://europa.eu.int/comm/enterprise/services/tourism/tourism_ist_conference/conftour_andreanicolai.pdf
- [20] Nicolai, A. (2001). *European Initiatives F.E.T.I.S.H. A vision for services interoperability in the Tourism Domain*. Enter 2001, Montreal. http://www.fetish.t6.it/resources/file/Montreal_Fetish-Enter2001.pdf
- [21] OASIS, <http://www.oasis-open.org/home/index.php>
- [22] Open Travel Alliance (OTA), www.opentravel.org
- [23] Web Ontology Language (OWL), <http://www.w3c.org/TR/2003/PR-owl-features-20031215/>
- [24] OWL-QL project reference page, <http://ksl.stanford.edu/projects/owl-ql/>
- [25] Pegasus Solutions, <http://www.pegas.com/>
- [26] Peters, J., *Datalex: Web services in the Travel Industry*, [http://www.eye-fortravel.com/papers/Web%20Services%20in%20the%20Travel%20Industry%20\(ATL-M0001.BB7\).pdf](http://www.eye-fortravel.com/papers/Web%20Services%20in%20the%20Travel%20Industry%20(ATL-M0001.BB7).pdf)
- [27] Sabre, <http://www.sabre.com>
- [28] Sabre Web Services, http://www.sabretravelnetwork.com/products_and_services/travel_agencies/hardware_software/sp2_007345.htm
- [29] Standard Generalized Markup Language (SGML), <http://www.w3.org/MarkUp/SGML/>
- [30] Simple Object Access Protocol (SOAP), <http://www.w3.org/TR/soap/>
- [31] Sybase Open Source for ebXML Messaging, <http://www.sybase.com>
- [32] United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), <http://www.unece.org/cefact/>
- [33] Universal Description, Discovery and Integration (UDDI), <http://www.uddi.org/>
- [34] Utell, <http://www.utell.com/static/whatisutell/default.asp>
- [35] Web Services, <http://www.w3c.org/2002/ws/>
- [36] Web Services Description Language (WSDL), <http://www.w3.org/TR/wsdl>
- [37] Worldspan, <http://www.worldspan.com/home.asp?fPageID=2>
- [38] The World Wide Web Consortium, (W3C), www.w3c.org
- [39] eXtensible Markup Language (XML), <http://www.w3c.org/XML/>
- [40] XML Schema, <http://www.w3c.org/XML/Schema>

Παράρτημα Α

Παράδειγμα εγγράφου CPA

```
<?xml version="1.0" encoding="UTF-8"?>
<tp:CollaborationProtocolAgreement xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:tp="http://www.oasis-
open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" tp:cpaid="OTAClient_OTAServer_Demo_CPA1" tp:version="1.2"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-cppa/schema/cpp-cpa-2_0.xsd draft-cpp-cpa-015.xsd">
  <tp:Status tp:value="proposed"/>
  <tp:Start>2005-01-26T15:58:35</tp:Start>
  <tp:End>2006-01-26T15:58:35</tp:End>
  <tp:ConversationConstraints tp:invocationLimit="1000" tp:concurrentConversations="200"/>
  <tp:PartyInfo tp:partyName="OTA_Client" tp:defaultMshChannelId="TYPEA">
    <tp:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">OTA_Client</tp:PartyId>
    <tp:PartyRef xlink:href="localhost:8080/ebXML/msh"/>
    <tp:CollaborationRole id="SessionRequest">
      <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction"
xlink:type="simple" xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
      <tp:Role tp:name="OTA_Client" xlink:type="simple"
xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Client"/>
      <tp:ApplicationCertificateRef tp:certId="CompanyA_AppCert"/>
      <tp:ServiceBinding>
        <tp:Service>uri:Session</tp:Service>
        <tp:CanSend>
          <tp:ThisPartyActionBinding tp:id="OTAClient_ABID1"
tp:action="CreateRQ" tp:packageId="SessionCreateRequestPackage">
            <tp:ChannelId>TYPEA</tp:ChannelId>
          </tp:ThisPartyActionBinding>
          <tp:OtherPartyActionBinding>OTAServer_ABID1</tp:OtherPartyActionBinding>
          <tp:CanReceive>
            <tp:ThisPartyActionBinding tp:id="OTAClient_ABID2"
tp:action="CreateRS" tp:packageId="SessionCreateResponsePackage">
              <tp:ChannelId>TYPEA</tp:ChannelId>
            </tp:ThisPartyActionBinding>
            <tp:OtherPartyActionBinding>OTAServer_ABID2</tp:OtherPartyActionBinding>
            <tp:CanReceive>
              <tp:ThisPartyActionBinding tp:id="OTAClient_ABID3"
tp:action="ErrorRS" tp:packageId="ErrorResponsePackage">
                <tp:ChannelId>TYPEA</tp:ChannelId>
              </tp:ThisPartyActionBinding>
              <tp:OtherPartyActionBinding>OTAServer_ABID3</tp:OtherPartyActionBinding>
              <tp:CanReceive>
                <tp:CanSend>
                  <tp:CanSend>
                    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID21"
tp:action="CloseRQ" tp:packageId="SessionCloseRequestPackage">
                      <tp:ChannelId>TYPEA</tp:ChannelId>
                    </tp:ThisPartyActionBinding>
                    <tp:OtherPartyActionBinding>OTAServer_ABID21</tp:OtherPartyActionBinding>
                    <tp:CanReceive>
                      <tp:ThisPartyActionBinding tp:id="OTAClient_ABID22"
tp:action="CloseRS" tp:packageId="SessionCloseResponsePackage">
                        <tp:ChannelId>TYPEA</tp:ChannelId>
                      </tp:ThisPartyActionBinding>
                      <tp:OtherPartyActionBinding>OTAServer_ABID22</tp:OtherPartyActionBinding>
                      <tp:CanReceive>
                        <tp:CanReceive>
                          <tp:ThisPartyActionBinding tp:id="OTAClient_ABID23"
tp:action="ErrorRS" tp:packageId="ErrorResponsePackage">
                            <tp:ChannelId>TYPEA</tp:ChannelId>
                          </tp:ThisPartyActionBinding>
```

```

<tp:OtherPartyActionBinding>OTAServer_ABID23</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  </tp:CanSend>
  </tp:ServiceBinding>
  </tp:CollaborationRole>
  <tp:CollaborationRole id="HotelBookingRequest">
    <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction"
xlink:type="simple" xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
    <tp:Role tp:name="OTA_Client" xlink:type="simple"
xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Client"/>
    <tp:ApplicationCertificateRef tp:certId="CompanyA_AppCert"/>
    <tp:ServiceBinding>
      <tp:Service>uri:HotelBooking</tp:Service>
      <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAClient_ABID1"
tp:action="OTA_HotelSearchRQ" tp:packageId="OTA_HotelSearchRequestPackage">
          <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
      </tp:CanSend>
    </tp:ServiceBinding>
  </tp:CollaborationRole>
</tp:OtherPartyActionBinding>OTAServer_ABID1</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  <tp:ThisPartyActionBinding tp:id="OTAClient_ABID2"
tp:action="OTA_HotelSearchRS" tp:packageId="OTA_HotelSearchResponsePackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
  </tp:ThisPartyActionBinding>
</tp:OtherPartyActionBinding>OTAServer_ABID2</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  </tp:CanSend>
  <tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID3"
tp:action="OTA_HotelAvailRQ" tp:packageId="OTA_HotelAvailRequestPackage">
      <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>
  </tp:CanSend>
</tp:OtherPartyActionBinding>OTAServer_ABID3</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  <tp:ThisPartyActionBinding tp:id="OTAClient_ABID4"
tp:action="OTA_HotelAvailRS" tp:packageId="OTA_HotelAvailResponsePackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
  </tp:ThisPartyActionBinding>
</tp:OtherPartyActionBinding>OTAServer_ABID4</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  </tp:CanSend>
  <tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID5"
tp:action="OTA_HotelResRQ" tp:packageId="OTA_HotelResRequestPackage">
      <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>
  </tp:CanSend>
</tp:OtherPartyActionBinding>OTAServer_ABID5</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  <tp:ThisPartyActionBinding tp:id="OTAClient_ABID6"
tp:action="OTA_HotelResRS" tp:packageId="OTA_HotelResResponsePackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
  </tp:ThisPartyActionBinding>
</tp:OtherPartyActionBinding>OTAServer_ABID6</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  </tp:CanSend>
  <tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID7"
tp:action="OTA_ReadRQ" tp:packageId="OTA_ReadRequestPackage">
      <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>
  </tp:CanSend>
</tp:OtherPartyActionBinding>OTAServer_ABID7</tp:OtherPartyActionBinding>
  </tp:CanReceive>
  <tp:ThisPartyActionBinding tp:id="OTAClient_ABID8"
tp:action="OTA_HotelReservation" tp:packageId="OTA_ReadResponsePackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
  </tp:ThisPartyActionBinding>
</tp:OtherPartyActionBinding>

```

```

<tp:OtherPartyActionBinding>OTAServer_ABID8</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:CanSend>
    </tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID9"
tp:action="OTA_CancelRQ" tp:packageId="OTA_CancelRequestPackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID9</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID10"
tp:action="OTA_CancelRS" tp:packageId="OTA_CancelResponsePackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID10</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:CanSend>
    </tp:ServiceBinding>
    </tp:CollaborationRole>
    <tp:CollaborationRole id="ProfileRequest">
        <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction"
xlink:type="simple" xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
        <tp:Role tp:name="OTA_Client" xlink:type="simple"
xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Client"/>
        <tp:ApplicationCertificateRef tp:certId="CompanyA_AppCert"/>
        <tp:ServiceBinding>
            <tp:Service>uri:Profile</tp:Service>
            </tp:CanSend>
            <tp:ThisPartyActionBinding tp:id="OTAClient_ABID11"
tp:action="OTA_CreateProfileRQ" tp:packageId="OTA_CreateProfileRequestPackage">
                <tp:ChannelId>TYPEA</tp:ChannelId>
                </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID11</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID12"
tp:action="OTA_CreateProfileRS" tp:packageId="OTA_CreateProfileResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID12</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:CanSend>
    </tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID13"
tp:action="OTA_ReadRQ" tp:packageId="OTA_ReadRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID13</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID14"
tp:action="OTA_ReadRS" tp:packageId="OTA_ReadResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID14</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:CanSend>
    </tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID15"
tp:action="OTA_UpdateRQ" tp:packageId="OTA_UpdateRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID15</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID16"
tp:action="OTA_UpdateRS" tp:packageId="OTA_UpdateResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

```

```

<tp:OtherPartyActionBinding>OTAServer_ABID16</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:CanSend>
    </tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID17"
tp:action="OTA_DeleteRQ" tp:packageId="OTA_DeleteRequestPackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID17</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    <tp:ThisPartyActionBinding tp:id="OTAClient_ABID18"
tp:action="OTA_DeleteRS" tp:packageId="OTA_DeleteResponsePackage">
    <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAServer_ABID18</tp:OtherPartyActionBinding>
    </tp:CanReceive>
    </tp:CanSend>
    </tp:ServiceBinding>
    </tp:CollaborationRole>
    <tp:Certificate tp:certId="CompanyA_AppCert">
    <ds:KeyInfo>
        <ds:KeyName>CompanyA_AppCert_Key</ds:KeyName>
    </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="CompanyA_ClientCert">
    <ds:KeyInfo>
        <ds:MgmtData>(qop=sybpks_intl)</ds:MgmtData>
    </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="CompanyA_ServerCert">
    <ds:KeyInfo>
        <ds:MgmtData>(qop=sybpks_intl)</ds:MgmtData>
    </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="CompanyA_SigningCert">
    <ds:KeyInfo>
        <ds:KeyName>pkcs11://tokenName/tomcat</ds:KeyName>
    </ds:KeyInfo>
    </tp:Certificate>
    <tp:Certificate tp:certId="TrustedRootCertA1">
    <ds:KeyInfo>
        <ds:KeyName>pkcs11://tokenName/tomcat</ds:KeyName>
    </ds:KeyInfo>
    </tp:Certificate>
    <tp:SecurityDetails tp:securityId="CompanyA_MessageSecurity">
    <tp:TrustAnchors>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA1"/>
    </tp:TrustAnchors>
    </tp:SecurityDetails>
    <tp:SecurityDetails tp:securityId="CompanyA_TransportSecurity">
    <tp:TrustAnchors>
        <tp:AnchorCertificateRef tp:certId="TrustedRootCertA1"/>
    </tp:TrustAnchors>
    </tp:SecurityDetails>
    <tp:DeliveryChannel tp:channelId="TYPEA" tp:transportId="HTTPS01" tp:docExchangeId="TYPEA">
    <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isSecureTransportRequired="true" tp:isConfidential="none"
tp:isAuthenticated="none" tp:isAuthorizationRequired="false" tp:timeToPerform="PID"/>
    <tp:MessagingCharacteristics tp:syncReplyMode="responseOnly" tp:ackRequested="never"
tp:ackSignatureRequested="never" tp:duplicateElimination="never"/>
    </tp:DeliveryChannel>
    <tp:DeliveryChannel tp:channelId="TYPEB" tp:transportId="HTTPS01" tp:docExchangeId="TYPEB">
    <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isSecureTransportRequired="true" tp:isConfidential="none"
tp:isAuthenticated="none" tp:isAuthorizationRequired="false" tp:timeToPerform="PID"/>
    <tp:MessagingCharacteristics tp:syncReplyMode="responseOnly" tp:ackRequested="never"
tp:ackSignatureRequested="never" tp:duplicateElimination="never"/>
    </tp:DeliveryChannel>
    <tp:Transport tp:transportId="HTTPS01">
    <tp:TransportSender>
        <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
    </tp:TransportSender>
    </tp:Transport>

```

```

        <tp:TransportSecurityProtocol
tp:version="3">SSL</tp:TransportSecurityProtocol>
        <tp:ClientCertificateRef tp:certId="CompanyA_ClientCert"/>
        <tp:ServerSecurityDetailsRef
tp:securityId="CompanyA_TransportSecurity"/>
        <tp:TransportClientSecurity>
        </tp:TransportSender>
        <tp:TransportReceiver>
        <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
        <tp:Endpoint tp:uri="https://localhost:8443/ebXML/msh" tp:type="allPurpose"/>
        <tp:TransportServerSecurity>
        <tp:TransportSecurityProtocol
tp:version="3">SSL</tp:TransportSecurityProtocol>
        <tp:ServerCertificateRef tp:certId="CompanyA_ServerCert"/>
        <tp:ClientSecurityDetailsRef
tp:securityId="CompanyA_TransportSecurity"/>
        <tp:TransportServerSecurity>
        </tp:TransportReceiver>
        </tp:Transport>
        <tp:DocExchange tp:docExchangeId="TYPEA">
        <tp:ebXMLSenderBinding tp:version="2.0">
        <tp:PersistDuration>PID</tp:PersistDuration>
        </tp:ebXMLSenderBinding>
        <tp:ebXMLReceiverBinding tp:version="2.0">
        <tp:PersistDuration>PID</tp:PersistDuration>
        </tp:ebXMLReceiverBinding>
        </tp:DocExchange>
        <tp:DocExchange tp:docExchangeId="TYPEB">
        <tp:ebXMLReceiverBinding tp:version="2.0">
        <tp:ReliableMessaging>
        <tp:Retries>5</tp:Retries>
        <tp:RetryInterval>1800</tp:RetryInterval>
        </tp:ReliableMessaging>
        </tp:ebXMLReceiverBinding>
        </tp:DocExchange>
        </tp:PartyInfo>
        <tp:PartyInfo tp:partyName="OTA_Server" tp:defaultMshChannelId="TYPEA">
        <tp:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:duns">OTA_Server</tp:PartyId>
        <tp:PartyRef xlink:type="simple" xlink:href="localhost:8080/ebXML/msh"/>
        <tp:CollaborationRole id="SessionResponse">
        <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction"
xlink:type="simple" xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
        <tp:Role tp:name="OTA_Server" xlink:type="simple"
xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Server"/>
        <tp:ApplicationCertificateRef tp:certId="CompanyB_AppCert"/>
        <tp:ServiceBinding>
        <tp:Service>uri:Session</tp:Service>
        <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID1"
tp:action="CreateRQ" tp:packageId="SessionCreateRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
        <tp:OtherPartyActionBinding>OTAClient_ABID1</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID2"
tp:action="CreateRS" tp:packageId="SessionCreateResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
        <tp:OtherPartyActionBinding>OTAClient_ABID2</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID3"
tp:action="ErrorRS" tp:packageId="ErrorResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
        <tp:OtherPartyActionBinding>OTAClient_ABID3</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:CanReceive>
        <tp:CanReceive>

```

```

        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID21"
tp:action="CloseRQ" tp:packageId="SessionCloseRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID21</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID22"
tp:action="CloseRS" tp:packageId="SessionCloseResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID22</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID23"
tp:action="ErrorRS" tp:packageId="ErrorResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID23</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:CanReceive>
        </tp:ServiceBinding>
        </tp:CollaborationRole>
        <tp:CollaborationRole id="HotelBookingResponse">
        <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction"
xlink:type="simple" xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
        <tp:Role tp:name="OTA_Server" xlink:type="simple"
xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Server"/>
        <tp:ApplicationCertificateRef tp:certId="CompanyB_AppCert"/>
        <tp:ServiceBinding>
        <tp:Service>uri:HotelBooking</tp:Service>
        <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID1"
tp:action="OTA_HotelSearchRQ" tp:packageId="OTA_HotelSearchRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID1</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID2"
tp:action="OTA_HotelSearchRS" tp:packageId="OTA_HotelSearchResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID2</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:CanReceive>
        <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID3"
tp:action="OTA_HotelAvailRQ" tp:packageId="OTA_HotelAvailRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID3</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID4"
tp:action="OTA_HotelAvailRS" tp:packageId="OTA_HotelAvailResponsePackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID4</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:CanReceive>
        <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID5"
tp:action="OTA_HotelResRQ" tp:packageId="OTA_HotelResRequestPackage">
        <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID5</tp:OtherPartyActionBinding>
        <tp:CanSend>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID6"
tp:action="OTA_HotelResRS" tp:packageId="OTA_HotelResResponsePackage">

```

```

        <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>

    <tp:OtherPartyActionBinding>OTAClient_ABID6</tp:OtherPartyActionBinding>
        <tp:CanSend>
        </tp:CanReceive>
        <tp:CanReceive>
            <tp:ThisPartyActionBinding tp:id="OTAServer_ABID7"
tp:action="OTA_ReadRQ" tp:packageId="OTA_ReadRequestPackage">
                <tp:ChannelId>TYPEA</tp:ChannelId>
            </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID7</tp:OtherPartyActionBinding>
            <tp:CanSend>
            <tp:ThisPartyActionBinding tp:id="OTAServer_ABID8"
tp:action="OTA_HotelReservation" tp:packageId="OTA_ReadResponsePackage">
                <tp:ChannelId>TYPEA</tp:ChannelId>
            </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID8</tp:OtherPartyActionBinding>
            <tp:CanSend>
            </tp:CanReceive>
            <tp:CanReceive>
                <tp:ThisPartyActionBinding tp:id="OTAServer_ABID9"
tp:action="OTA_CancelRQ" tp:packageId="OTA_CancelRequestPackage">
                    <tp:ChannelId>TYPEA</tp:ChannelId>
                </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID9</tp:OtherPartyActionBinding>
            <tp:CanSend>
            <tp:ThisPartyActionBinding tp:id="OTAServer_ABID10"
tp:action="OTA_CancelRS" tp:packageId="OTA_CancelResponsePackage">
                <tp:ChannelId>TYPEA</tp:ChannelId>
            </tp:ThisPartyActionBinding>

        <tp:OtherPartyActionBinding>OTAClient_ABID10</tp:OtherPartyActionBinding>
            <tp:CanSend>
            </tp:CanReceive>
            <tp:ServiceBinding>
            <tp:CollaborationRole>
            <tp:CollaborationRole id="ProfileResponse">
                <tp:ProcessSpecification tp:version="2.0" tp:name="OTASimpleTransaction"
xlink:type="simple" xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml" />
                <tp:Role tp:name="OTA_Server" xlink:type="simple"
xlink:href="http://localhost:8080/ebXML/cpas/OTASimpleTransaction.xml#OTA_Server"/>
                <tp:ApplicationCertificateRef tp:certId="CompanyB_AppCert"/>
            <tp:ServiceBinding>
                <tp:Service>uri:Profile</tp:Service>
            <tp:CanReceive>
                <tp:ThisPartyActionBinding tp:id="OTAServer_ABID11"
tp:action="OTA_CreateProfileRQ" tp:packageId="OTA_CreateProfileRequestPackage">
                    <tp:ChannelId>TYPEA</tp:ChannelId>
                </tp:ThisPartyActionBinding>

            <tp:OtherPartyActionBinding>OTAClient_ABID11</tp:OtherPartyActionBinding>
                <tp:CanSend>
                <tp:ThisPartyActionBinding tp:id="OTAServer_ABID12"
tp:action="OTA_CreateProfileRS" tp:packageId="OTA_CreateProfileResponsePackage">
                    <tp:ChannelId>TYPEA</tp:ChannelId>
                </tp:ThisPartyActionBinding>

            <tp:OtherPartyActionBinding>OTAClient_ABID12</tp:OtherPartyActionBinding>
                <tp:CanSend>
                </tp:CanReceive>
                <tp:CanReceive>
                    <tp:ThisPartyActionBinding tp:id="OTAServer_ABID13"
tp:action="OTA_ReadRQ" tp:packageId="OTA_ReadRequestPackage">
                        <tp:ChannelId>TYPEA</tp:ChannelId>
                    </tp:ThisPartyActionBinding>

            <tp:OtherPartyActionBinding>OTAClient_ABID13</tp:OtherPartyActionBinding>
                <tp:CanSend>
                <tp:ThisPartyActionBinding tp:id="OTAServer_ABID14"
tp:action="OTA_ReadRS" tp:packageId="OTA_ReadResponsePackage">
                    <tp:ChannelId>TYPEA</tp:ChannelId>
                </tp:ThisPartyActionBinding>

```



```

</tp:ThisPartyActionBinding>

<tp:OtherPartyActionBinding>OTAClient_ABID14</tp:OtherPartyActionBinding>
  <tp:CanSend>
    <tp:CanReceive>
      <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID15"
tp:action="OTA_UpdateRQ" tp:packageId="OTA_UpdateRequestPackage">
          <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
      </tp:CanReceive>
    </tp:CanSend>
  </tp:OtherPartyActionBinding>OTAClient_ABID15</tp:OtherPartyActionBinding>
  <tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAServer_ABID16"
tp:action="OTA_UpdateRS" tp:packageId="OTA_UpdateResponsePackage">
      <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>
  </tp:CanSend>
</tp:OtherPartyActionBinding>OTAClient_ABID16</tp:OtherPartyActionBinding>
  <tp:CanSend>
    <tp:CanReceive>
      <tp:CanReceive>
        <tp:ThisPartyActionBinding tp:id="OTAServer_ABID17"
tp:action="OTA_DeleteRQ" tp:packageId="OTA_DeleteRequestPackage">
          <tp:ChannelId>TYPEA</tp:ChannelId>
        </tp:ThisPartyActionBinding>
      </tp:CanReceive>
    </tp:CanSend>
  </tp:OtherPartyActionBinding>OTAClient_ABID17</tp:OtherPartyActionBinding>
  <tp:CanSend>
    <tp:ThisPartyActionBinding tp:id="OTAServer_ABID18"
tp:action="OTA_DeleteRS" tp:packageId="OTA_DeleteResponsePackage">
      <tp:ChannelId>TYPEA</tp:ChannelId>
    </tp:ThisPartyActionBinding>
  </tp:CanSend>
</tp:OtherPartyActionBinding>OTAClient_ABID18</tp:OtherPartyActionBinding>
  <tp:CanSend>
    <tp:CanReceive>
      <tp:ServiceBinding>
        <tp:CollaborationRole>
          <tp:Certificate tp:certId="CompanyB_AppCert">
            <ds:KeyInfo>
              <ds:KeyName>CompanyB_AppCert_Key</ds:KeyName>
            </ds:KeyInfo>
          </tp:Certificate>
          <tp:Certificate tp:certId="CompanyB_ClientCert">
            <ds:KeyInfo>
              <ds:MgmtData>(qop=sybpks_intl)</ds:MgmtData>
            </ds:KeyInfo>
          </tp:Certificate>
          <tp:Certificate tp:certId="CompanyB_ServerCert">
            <ds:KeyInfo>
              <ds:MgmtData>(qop=sybpks_intl)</ds:MgmtData>
            </ds:KeyInfo>
          </tp:Certificate>
          <tp:Certificate tp:certId="CompanyB_SigningCert">
            <ds:KeyInfo>
              <ds:KeyName>pkcs11://tokenName/tomcat</ds:KeyName>
            </ds:KeyInfo>
          </tp:Certificate>
          <tp:Certificate tp:certId="TrustedRootCertB1">
            <ds:KeyInfo>
              <ds:KeyName>pkcs11://tokenName/tomcat</ds:KeyName>
            </ds:KeyInfo>
          </tp:Certificate>
          <tp:SecurityDetails tp:securityId="CompanyB_MessageSecurity">
            <tp:TrustAnchors>
              <tp:AnchorCertificateRef tp:certId="TrustedRootCertB1"/>
            </tp:TrustAnchors>
          </tp:SecurityDetails>
          <tp:SecurityDetails tp:securityId="CompanyB_TransportSecurity">
            <tp:TrustAnchors>
              <tp:AnchorCertificateRef tp:certId="TrustedRootCertB1"/>
            </tp:TrustAnchors>
          </tp:SecurityDetails>
          <tp:DeliveryChannel tp:channelId="TYPEA" tp:transportId="HTTPS01" tp:docExchangeId="TYPEA">

```



```

        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isSecureTransportRequired="true" tp:isConfidential="none"
tp:isAuthenticated="none" tp:isAuthorizationRequired="false" tp:timeToPerform="PID"/>
        <tp:MessagingCharacteristics tp:syncReplyMode="responseOnly" tp:ackRequested="never"
tp:ackSignatureRequested="never" tp:duplicateElimination="never"/>
    </tp:DeliveryChannel>
    <tp:DeliveryChannel tp:channelId="TYPEB" tp:transportId="HTTPS01" tp:docExchangeId="TYPEB">
        <tp:BusinessTransactionCharacteristics tp:isNonRepudiationRequired="false"
tp:isNonRepudiationReceiptRequired="false" tp:isSecureTransportRequired="true" tp:isConfidential="none"
tp:isAuthenticated="none" tp:isAuthorizationRequired="false" tp:timeToPerform="PID"/>
        <tp:MessagingCharacteristics tp:syncReplyMode="responseOnly" tp:ackRequested="never"
tp:ackSignatureRequested="never" tp:duplicateElimination="never"/>
    </tp:DeliveryChannel>
    <tp:Transport tp:transportId="HTTPS01">
        <tp:TransportSender>
            <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
            <tp:TransportClientSecurity>
                <tp:TransportSecurityProtocol
tp:version="3">SSL</tp:TransportSecurityProtocol>
                    <tp:ClientCertificateRef tp:certId="CompanyB_ClientCert"/>
                    <tp:ServerSecurityDetailsRef
tp:securityId="CompanyB_TransportSecurity"/>
                </tp:TransportClientSecurity>
            </tp:TransportSender>
            <tp:TransportReceiver>
                <tp:TransportProtocol tp:version="1.1">HTTP</tp:TransportProtocol>
                <tp:Endpoint tp:uri="https://localhost:8443/ebXML/msh" tp:type="allPurpose"/>
                <tp:TransportServerSecurity>
                    <tp:TransportSecurityProtocol
tp:version="3">SSL</tp:TransportSecurityProtocol>
                        <tp:ServerCertificateRef tp:certId="CompanyB_ServerCert"/>
                        <tp:ClientSecurityDetailsRef
tp:securityId="CompanyB_TransportSecurity"/>
                    </tp:TransportServerSecurity>
                </tp:TransportReceiver>
            </tp:Transport>
            <tp:DocExchange tp:docExchangeId="TYPEA">
                <tp:ebXMLSenderBinding tp:version="2.0">
                    <tp:PersistDuration>PID</tp:PersistDuration>
                </tp:ebXMLSenderBinding>
                <tp:ebXMLReceiverBinding tp:version="2.0">
                    <tp:PersistDuration>PID</tp:PersistDuration>
                </tp:ebXMLReceiverBinding>
            </tp:DocExchange>
            <tp:DocExchange tp:docExchangeId="TYPEB">
                <tp:ebXMLReceiverBinding tp:version="2.0">
                    <tp:ReliableMessaging>
                        <tp:Retries>5</tp:Retries>
                        <tp:RetryInterval>1800</tp:RetryInterval>
                    </tp:ReliableMessaging>
                </tp:ebXMLReceiverBinding>
            </tp:DocExchange>
        </tp:PartyInfo>
        <tp:SimplePart id="SessionCreate_Request" tp:mimetype="application/xml">
            <tp:NamespaceSupported tp:location="http://www.ebxml.org/processes"
tp:version="1.0">http://www.ebxml.org/processes</tp:NamespaceSupported>
        </tp:SimplePart>
        <tp:SimplePart id="SessionCreate_Response" tp:mimetype="application/xml">
            <tp:NamespaceSupported tp:location="http://www.ebxml.org/processes"
tp:version="1.0">http://www.ebxml.org/processes</tp:NamespaceSupported>
        </tp:SimplePart>
        <tp:SimplePart id="Error_Response" tp:mimetype="application/xml">
            <tp:NamespaceSupported tp:location="http://www.ebxml.org/processes"
tp:version="1.0">http://www.ebxml.org/processes</tp:NamespaceSupported>
        </tp:SimplePart>
        <tp:SimplePart id="OTAClient_MsgHdr" tp:mimetype="text/xml">
            <tp:NamespaceSupported tp:location="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd" tp:version="2.0">http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-
2_0.xsd</tp:NamespaceSupported>
        </tp:SimplePart>
        <tp:SimplePart id="OTAServer_MsgHdr" tp:mimetype="text/xml">

```



```

</tp:SimplePart>
<tp:Packaging id="SessionCreateRequestPackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="SessionCreate_RequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAClient_MsgHdr"/>
      <tp:Constituent idref="SessionCreate_Request"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="SessionCreateResponsePackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="SessionCreate_ResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAServer_MsgHdr"/>
      <tp:Constituent idref="SessionCreate_Response"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="SessionCloseRequestPackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="SessionClose_RequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAClient_MsgHdr"/>
      <tp:Constituent idref="SessionClose_Request"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="SessionCloseResponsePackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="SessionClose_ResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAServer_MsgHdr"/>
      <tp:Constituent idref="SessionClose_Response"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="ErrorResponsePackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="Error_ResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAServer_MsgHdr"/>
      <tp:Constituent idref="Error_Response"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_HotelSearchRequestPackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="OTA_HotelSearchRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAClient_MsgHdr"/>
      <tp:Constituent idref="OTA_HotelSearch_Request"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_HotelSearchResponsePackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>
    <tp:Composite id="OTA_HotelSearchResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
      <tp:Constituent idref="OTAServer_MsgHdr"/>
      <tp:Constituent idref="OTA_HotelSearch_Response"/>
    </tp:Composite>
  </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_HotelAvailRequestPackage">
  <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
  <tp:CompositeList>

```

```

        <tp:Composite id="OTA_HotelAvailRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAClient_MsgHdr"/>
            <tp:Constituent idref="OTA_HotelAvail_Request"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_HotelAvailResponsePackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_HotelAvailResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAServer_MsgHdr"/>
            <tp:Constituent idref="OTA_HotelAvail_Response"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_HotelResRequestPackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_HotelResRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAClient_MsgHdr"/>
            <tp:Constituent idref="OTA_HotelRes_Request"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_HotelResResponsePackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_HotelResResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAServer_MsgHdr"/>
            <tp:Constituent idref="OTA_HotelRes_Response"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_ReadRequestPackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_ReadRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAClient_MsgHdr"/>
            <tp:Constituent idref="OTA_Read_Request"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_ReadResponsePackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_ReadResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAServer_MsgHdr"/>
            <tp:Constituent idref="OTA_Read_Response"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_CancelRequestPackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_CancelRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAClient_MsgHdr"/>
            <tp:Constituent idref="OTA_Cancel_Request"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
<tp:Packaging id="OTA_CancelResponsePackage">
    <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
    <tp:CompositeList>
        <tp:Composite id="OTA_CancelResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAServer_MsgHdr"/>
            <tp:Constituent idref="OTA_Cancel_Response"/>
        </tp:Composite>
    </tp:CompositeList>

```

```

        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_CreateProfileRequestPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_CreateProfileRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAClient_MsgHdr"/>
                <tp:Constituent idref="OTA_CreateProfile_Request"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_CreateProfileResponsePackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_CreateProfileResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAServer_MsgHdr"/>
                <tp:Constituent idref="OTA_CreateProfile_Response"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_ReadRequestPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_ReadRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAClient_MsgHdr"/>
                <tp:Constituent idref="OTA_Read_Request"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_ReadResponsePackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_ReadResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAServer_MsgHdr"/>
                <tp:Constituent idref="OTA_Read_Response"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_UpdateRequestPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_UpdateRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAClient_MsgHdr"/>
                <tp:Constituent idref="OTA_Update_Request"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_UpdateResponsePackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_UpdateResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAServer_MsgHdr"/>
                <tp:Constituent idref="OTA_Update_Response"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_DeleteRequestPackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>
            <tp:Composite id="OTA_DeleteRequestMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
                <tp:Constituent idref="OTAClient_MsgHdr"/>
                <tp:Constituent idref="OTA_Delete_Request"/>
            </tp:Composite>
        </tp:CompositeList>
    </tp:Packaging>
    <tp:Packaging id="OTA_DeleteResponsePackage">
        <tp:ProcessingCapabilities tp:parse="true" tp:generate="true"/>
        <tp:CompositeList>

```

```

        <tp:Composite id="OTA_DeleteResponseMsg" tp:mimetype="multipart/related"
tp:mimeparameters="type=text/xml;">
            <tp:Constituent idref="OTAServer_MsgHdr"/>
            <tp:Constituent idref="OTA_Delete_Response"/>
        </tp:Composite>
    </tp:CompositeList>
</tp:Packaging>
</tp:CollaborationProtocolAgreement>

```

Παράδειγμα εγγράφου BPSS

```

<?xml version="1.0" encoding="UTF-8"?>
<ProcessSpecification name="OTASimpleTransaction" uuid="[1234-5678-901234]" version="1.0">
    <DocumentSpecification name="OTARRequest">
        <BusinessDocument name="OTARRequest"/>
    </DocumentSpecification>
    <DocumentSpecification name="OTARResponse">
        <BusinessDocument name="OTARResponse"/>
    </DocumentSpecification>
    <BusinessTransaction name="OTARRequestTrasaction">
        <RequestingBusinessActivity name="OTARRequestTrasactionRequest">
            <DocumentEnvelope businessDocument="OTARRequest"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="OTARRequestTrasactionResponse"/>
    </BusinessTransaction>
    <BusinessTransaction name="OTARResponseTrasaction">
        <RequestingBusinessActivity name="OTARResponseTrasactionRequest">
            <DocumentEnvelope businessDocument="OTARResponse"/>
        </RequestingBusinessActivity>
        <RespondingBusinessActivity name="OTARResponseTrasactionResponse"/>
    </BusinessTransaction>
    <BinaryCollaboration name="OTATrasaction">
        <InitiatingRole name="OTA_Client"/>
        <RespondingRole name="OTA_Server"/>
        <Start toBusinessState="OTARQBTA"/>
        <Success fromBusinessState="OTARSBTA"/>
        <Failure fromBusinessState="OTARSBTA"/>
        <BusinessTransactionActivity
            businessTransaction="OTARRequestTrasaction"
            fromAuthorizedRole="OTA_Client" name="OTARQBTA" toAuthorizedRole="OTA_Server"/>
        <BusinessTransactionActivity
            businessTransaction="OTARResponse"
            fromAuthorizedRole="OTA_Server" name="OTARSBTA" toAuthorizedRole="OTA_Client"/>
        <Transition fromBusinessState="OTARQBTA" toBusinessState="OTARSBTA"/>
    </BinaryCollaboration>
</ProcessSpecification>

```

Παράρτημα Β

Οι πίνακες της Βάσης Δεδομένων του Προφίλ Πελάτη

1: ProfileInfo			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) ProfileInfo.			
ProfileInfoID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
ProfileInfo__Has__UniqueId	int		Foreign key στον πίνακα UniqueIdType που είναι ο τύπος του στοιχείου UniqueID, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) ProfileInfo ο οποίος αποθηκεύεται στον παρόντα πίνακα.

2: UniqueIdType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) UniqueIdType. Μοναδικό αναγνωριστικό που δίνεται σε κάθε προφίλ.			
UniqueIdTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
UniqueIdType__Has__CompanyName	int		Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου CompanyName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) UniqueIdType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
URL	varchar(200)		Η ηλεκτρονική διεύθυνση.
Instance	varchar(32)		Η αναγνώριση μιας εγγραφής όπως υπάρχει σε μια συγκεκριμένη χρονική στιγμή. Χρησιμοποιείται στα μηνύματα ενημέρωσης, όπου ο αποστολέας πρέπει να διαβεβαιώσει τον server ότι η αίτηση για ενημέρωση αναφέρεται στην πιο πρόσφατη έκδοση του αντικειμένου που θα ενημερωθεί.
Type	varchar(200)		Αναφέρεται στον τύπο του αντικειμένου το οποίο καθορίζεται από το στοιχείο UniqueId. Παίρνει τιμές από τον πίνακα κωδικών του OTA.
Id	varchar(32)		Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.

3: CompanyNameType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CompanyNameType. Προσδιορίζει μια επιχείρηση με βάση το όνομα της.			
CompanyNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου CompanyNameType.
CodeContext	varchar(32)		Καθορίζει τον τύπο του αναγνωριστικού αριθμού, για παράδειγμα DUNS, IATA ή κάποιος εσωτερικός κωδικός κτλ.
CompanyShortName	varchar(32)		Used for Character Strings, length 1 to 32
TravelSector	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

Code	varchar(8)	Αναγνωριστικός κωδικός της επιχείρησης.
------	------------	---

4: ProfileType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) ProfileType. Το αρχικό στοιχείο του προφίλ.

ProfileTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
ProfileType__BelongsTo__ProfileInfo	int		Foreign key στον πίνακα ProfileInfo ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) ProfileInfo στο οποίο ανήκει το στοιχείο ProfileType.
CreateDateTime	date		
CreatorID	varchar(32)		Used for Character Strings, length 1 to 32. ID of creator. The creator could be a software system identifier or an identifier of an employee responsible for the creation.
LastModifyDateTime	date		Η ημερομηνία της τελευταίας τροποποίησης.
LastModifierID	varchar(32)		Ο αναγνωριστικός κωδικός αυτού που έκανε την τροποποίηση. Μπορεί να είναι ο αναγνωριστικός κωδικός ενός συστήματος λογισμικού, ή ο αναγνωριστικός κωδικός ενός εργαζόμενου που είναι υπεύθυνος για την τροποποίηση.
ShareAllSynchInd	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No'
ShareAllMarketInd	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No'
ProfileType	varchar(200)		Κωδικός που καθορίζει τον τύπο του προφίλ, για παράδειγμα Πελάτης, Tour Operator, Επιχείρηση κτλ. Παίρνει τιμές από τον πίνακα κωδικών του OTA.

5: AccessesType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AccessesType. Περιέχει πληροφορίες σχετικά με την δημιουργία και τις αλλαγές που γίνονται στο προφίλ.

AccessesTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AccessesType__BelongsTo__ProfileType	int		Foreign key στον πίνακα ProfileType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) ProfileType στο οποίο ανήκει το στοιχείο AccessesType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
CreateDateTime	date		Η ημερομηνία κατασκευής.

6: Access

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) Access.

AccessID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Access__BelongsTo__AccessesType	int		Foreign key στον πίνακα AccessesType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου)

		AccessesType στο οποίο ανήκει το στοιχείο Access.
Access__Has__AccessPerson	int	Foreign key στον πίνακα PersonNameType που είναι ο τύπος του στοιχείου AccessPerson, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) Access ο οποίος αποθηκεύεται στον παρόντα πίνακα.
Access__Has__AccessComment	int	Foreign key στον πίνακα FreeTextType που είναι ο τύπος του στοιχείου AccessComment, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) Access ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ActionType	varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Create', 'Read', 'Update', 'Delete'
ActionDateTime	date	Η ημερομηνία κατά την οποία εκτελέστηκε η συγκεκριμένη ενέργεια.
Id	varchar(32)	Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.

7: PersonNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PersonNameType. Περιέχει το πλήρες όνομα του ατόμου.

PersonNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SurnamePrefix	varchar(16)		e.g "van der", "von", "de" etc.
Surname	varchar(64)		Το επίθετο του πελάτη. Απαιτείται για την ταυτοποίηση του ατόμου στο προφίλ
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
NameType	varchar(200)		Ο τύπος του ονόματος του ατόμου, π.χ. επίσημο, χαϊδευτικό, εναλλακτικό κτλ.

8: PersonNameType__Has__NamePrefix

Ενδιάμεσος πίνακας που συνδέει τον πίνακα PersonNameType με τον πίνακα SimpleText16 που αποθηκεύει τον τύπο του στοιχείου NamePrefix.

PersonNameTypeID	int	foreign key	Foreign key στον πίνακα PersonNameType.
NamePrefixID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText16 που αποθηκεύει τον τύπο του στοιχείου NamePrefix. Salutation of honorific. (e.g., Mr. Mrs., Ms., Miss, Dr., etc.)

9: PersonNameType__Has__GivenName

Ενδιάμεσος πίνακας που συνδέει τον πίνακα PersonNameType με τον πίνακα SimpleText64 που αποθηκεύει τον τύπο του στοιχείου GivenName.

PersonNameTypeID	int	foreign key	Foreign key στον πίνακα PersonNameType.
GivenNameID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText64 που αποθηκεύει τον τύπο του στοιχείου GivenName. Given name, first name or names

10: PersonNameType__Has__MiddleName

Ενδιάμεσος πίνακας που συνδέει τον πίνακα PersonNameType με τον πίνακα SimpleText64 που αποθηκεύει τον

τύπο του στοιχείου MiddleName.			
PersonNameTypeID	int	foreign key	Foreign key στον πίνακα PersonNameType.
MiddleNameID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText64 που αποθηκεύει τον τύπο του στοιχείου MiddleName. The middle name of the person name

11: PersonNameType__Has__NameSuffix			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα PersonNameType με τον πίνακα SimpleText16 που αποθηκεύει τον τύπο του στοιχείου NameSuffix.			
PersonNameTypeID	int	foreign key	Foreign key στον πίνακα PersonNameType.
NameSuffixID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText16 που αποθηκεύει τον τύπο του στοιχείου NameSuffix. Hold various name suffixes and letters e.g., Jr., Sr., III, Ret., Esq., etc.)

12: PersonNameType__Has__NameTitle			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα PersonNameType με τον πίνακα SimpleText16 που αποθηκεύει τον τύπο του στοιχείου NameTitle.			
PersonNameTypeID	int	foreign key	Foreign key στον πίνακα PersonNameType.
NameTitleID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText16 που αποθηκεύει τον τύπο του στοιχείου NameTitle. Degree or honors (e.g., Ph.D., M.D., etc.)

13: FreeTextType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) FreeTextType. Περιέχει πληροφορίες σε μορφή ελεύθερου κειμένου. Μπορεί να χρησιμοποιηθούν για να εκφράσουν ειδικές ανάγκες ή απαιτήσεις.			
FreeTextTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Language	varchar(200)		Η γλώσσα γραφής του κειμένου.
PCDATA	varchar(200)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου FreeTextType.

14: CustomerType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CustomerType. Περιέχει βασικές πληροφορίες σχετικά με την ταυτότητα του πελάτη, την τοποθεσία που βρίσκεται, τα οικονομικά του κτλ.			
CustomerTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CustomerType__BelongsTo__ProfileType	int		Foreign key στον πίνακα ProfileType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) ProfileType στο οποίο ανήκει το στοιχείο CustomerType.
CustomerType__Has__PersonName	int		Foreign key στον πίνακα PersonNameType που είναι ο τυπος του στοιχείου PersonName, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) CustomerType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
CurrencyCode	varchar(200)		Used for Alphabetic Strings, length exactly 3
BirthDate	date		Η ημερομηνία γέννησης του πελάτη.
DecimalPlaces	BigInt		
Gender	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Male', 'Female', 'Unknown'
Deceased	Boolean		Δείχνει εάν έχει πεθάνει.

LockoutType	varchar(16)	
-------------	-------------	--

15: CustomerType__Has__PhysChallName			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα SimpleText128 που αποθηκεύει τον τύπο του στοιχείου PhysChallName.			
CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
PhysChallNameID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText128 που αποθηκεύει τον τύπο του στοιχείου PhysChallName.

16: CustomerType__Has__PetInfo			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα SimpleText64 που αποθηκεύει τον τύπο του στοιχείου PetInfo.			
CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
PetInfoID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText64 που αποθηκεύει τον τύπο του στοιχείου PetInfo.

17: CustomerType__Has__Address			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα Address_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου Address_1 ο οποίος είναι ο τυπος του στοιχείου Address.			
CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
AddressID	String	foreign key	Foreign key στον πίνακα πίνακα Address_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου Address_1 ο οποίος είναι ο τυπος του στοιχείου Address.

18: CustomerType__Has__URL			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα URL_Type που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου URL_Type ο οποίος είναι ο τυπος του στοιχείου URL.			
CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
URLID	String	foreign key	Foreign key στον πίνακα πίνακα URL_Type που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου URL_Type ο οποίος είναι ο τυπος του στοιχείου URL.

19: CustomerType__Has__Telephone			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα TelephoneInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TelephoneInfoType ο οποίος είναι ο τυπος του στοιχείου Telephone.			
CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
TelephoneID	String	foreign key	Foreign key στον πίνακα πίνακα TelephoneInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TelephoneInfoType ο οποίος είναι ο τυπος του στοιχείου Telephone.

20: CustomerType__Has__Email			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα EmailType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmailType ο οποίος είναι ο τυπος του στοιχείου Email.			
CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
EmailID	String	foreign key	Foreign key στον πίνακα πίνακα EmailType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmailType ο οποίος είναι ο τυπος του στοιχείου Email.

21: CustomerType__Has__PaymentForm

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα PaymentFormType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormType ο οποίος είναι ο τυπος του στοιχείου PaymentForm.

CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
PaymentFormID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormType ο οποίος είναι ο τυπος του στοιχείου PaymentForm.

22: CustomerType__Has__ContactPerson

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα ContactPersonType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ContactPersonType ο οποίος είναι ο τυπος του στοιχείου ContactPerson.

CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
ContactPersonID	String	foreign key	Foreign key στον πίνακα πίνακα ContactPersonType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ContactPersonType ο οποίος είναι ο τυπος του στοιχείου ContactPerson.

23: CustomerType__Has__EmployeeInfo

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CustomerType με τον πίνακα EmployeeInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmployeeInfoType ο οποίος είναι ο τυπος του στοιχείου EmployeeInfo.

CustomerTypeID	int	foreign key	Foreign key στον πίνακα CustomerType.
EmployeeInfoID	String	foreign key	Foreign key στον πίνακα πίνακα EmployeeInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmployeeInfoType ο οποίος είναι ο τυπος του στοιχείου EmployeeInfo.

24: AddressInfoType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AddressInfoType. Πληροφορίες σχετικά με μια διεύθυνση προσδιορίζοντας μια τοποθεσία για συγκεκριμένους σκοπούς.

AddressInfoTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AddressInfoType__Extends__AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει τον σύνθετο τυπο AddressType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος AddressInfoType).
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
UseType	varchar(200)		Περιγράφει την χρήση της διεύθυνσης.

25: AddressType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AddressType.

AddressTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Type	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του ΟΤΑ.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.

BldgRoom	varchar(64)	Το όνομα του κτιρίου, το δωμάτιο, το διαμέρισμα ή ο αριθμός της σουίτας.
CityName	varchar(64)	Το όνομα της πόλης, π.χ. Χανιά.
PostalCode	varchar(16)	Ταχυδρομικός κώδικας.
County	varchar(32)	Το όνομα του Δήμου, π.χ. Δήμος Ακρωτηρίου.
FormattedInd	Boolean	Ορίζει την δόμηση της πληροφορίας σε ένα σύνολο πεδίων ή την απλή περιγραφή της σε ένα string.

26: AddressType __Has__ AddressLine

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AddressType με τον πίνακα SimpleText64 που αποθηκεύει τον τύπο του στοιχείου AddressLine.

AddressTypeID	int	foreign key	Foreign key στον πίνακα AddressType.
AddressLineID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText64 που αποθηκεύει τον τύπο του στοιχείου AddressLine.

27: StreetNmbrType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) StreetNmbrType. Περιέχει την οδό και τον αριθμό επί της οδού.

StreetNmbrTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
StreetNmbrType __BelongsTo__ AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AddressType στο οποίο ανήκει το στοιχείο StreetNmbrType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου StreetNmbrType.
PO_Box	varchar(16)		Ο αριθμός του γραμματοκιβωτίου.

28: StateProvType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) StateProvType. Το όνομα του νομού ή της περιοχής ή κάποιος κωδικός για τον προσδιορισμό της τοποθεσίας.

StateProvTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
StateProvType __BelongsTo__ AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AddressType στο οποίο ανήκει το στοιχείο StateProvType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου StateProvType.
StateCode	varchar(8)		Ο κωδικός του νομού.

29: CountryNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CountryNameType. Το όνομα της χώρας που περιέχεται στην διεύθυνση.

CountryNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CountryNameType __BelongsTo__ AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AddressType στο οποίο ανήκει το στοιχείο CountryNameType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου CountryNameType.

Code	varchar(200)	Τυποποιημένος κωδικός χώρας (ISO 3166).
------	--------------	---

30: URL_Type

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) URL_Type. Η διεύθυνση της ιστοσελίδας.

URL_TypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Type	varchar(16)		Used for Character Strings, length 1 to 16. Defines the purpose of the URL address, such as personal, business, public, etc.
PCDATA	varchar(200)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου URL_Type.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.

31: TelephoneType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) TelephoneType. Περιέχει μια δομή για την αποθήκευση του αριθμού τηλεφώνου.

TelephoneTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Extension	varchar(200)		Used for Numeric Strings, maximum length 5. Extension to reach a specific party at the phone number.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
FormattedInd	Boolean		Ορίζει την δόμηση της πληροφορίας σε ένα σύνολο πεδίων ή την απλή περιγραφή της σε ένα string.
PhoneLocationType	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.
PhoneTechType	varchar(200)		Δείχνει τον τύπο του τηλεφώνου, π.χ. φάξ, κινητό, κτλ.
CountryAccessCode	varchar(200)		Ο διεθνής τηλεφωνικός κωδικός της χώρας.
AreaCityCode	varchar(200)		Ο τηλεφωνικός κωδικός της περιοχής, π.χ. 28210.
PhoneNumber	varchar(32)		Ο αριθμός τηλεφώνου.
PIN	varchar(8)		Ο προσωπικός αναγνωριστικός κωδικός του τηλεφώνου (PIN).

32: TelephoneInfoType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) TelephoneInfoType. Πληροφορίες σχετικά με τον αριθμό τηλεφώνου, συμπεριλαμβάνοντας τον αριθμό και την χρήση του.

TelephoneInfoTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
TelephoneInfoType__Extends__TelephoneType	int		Foreign key στον πίνακα TelephoneType ο οποίος αποθηκεύει τον σύνθετο τύπο TelephoneType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος TelephoneInfoType).
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.

PhoneUseType	varchar(200)	Περιγράφει τον τύπο τηλεφώνου στα πλαίσια της γενικής του χρήσης, όπως Σπίτι, Εργασία, Επείγουσα Επικοινωνία, Ταξιδιωτικός Πράκτορας, κτλ.
--------------	--------------	--

33: EmailType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) EmailType. Περιέχει την ηλεκτρονική διεύθυνση (email).

EmailTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου EmailType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
EmailType	varchar(200)		Καθορίζει τον τύπο της ηλεκτρονικής διεύθυνσης, όπως προσωπική επαγγελματική κτλ.

34: PaymentFormType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PaymentFormType. Περιγράφει τους τρόπους πληρωμής που διαθέτει το άτομο.

PaymentFormTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SelectedTable	varchar(200)		Ο πίνακας που επιλέγεται. Το όνομα του πίνακα που θα επιλεγεί πρέπει να είναι ένα από τα: PaymentCard, BankAcct, DirectBill, Voucher
SelectedTableID	int		Το primary Key του SelectedTable.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.
CostCenterId	varchar(32)		Κωδικός για την αναγνώριση της διεύθυνσης έκδοσης λογαριασμών.

35: PaymentCardType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PaymentCardType. Πληροφορίες για μια συγκεκριμένη πιστωτική κάρτα.

PaymentCardTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PaymentCardType__Has__Address	int		Foreign key στον πίνακα AddressType που είναι ο τύπος του στοιχείου Address, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) PaymentCardType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.

ShareMarketInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
EffectiveDate	varchar(200)	Ημερομηνία έναρξης ισχύος.
ExpireDate	varchar(200)	Ημερομηνία λήξης ισχύος.
CardHolderName	varchar(64)	Το όνομα του ιδιοκτήτη της κάρτας.
CardType	varchar(200)	Δείχνει τον τύπο της μαγνητικής κάρτας.
CardCode	varchar(200)	
CardNumber	varchar(200)	Ο αριθμός της πιστωτικής κάρτας που αναγράφεται ανάγλυφα πάνω στην κάρτα.
SeriesCode	varchar(200)	Τα ψηφία επαλήθευσης που τυπώνονται στην κάρτα μαζί με τον ανάγλυφο αριθμό.

36: IssuerNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) IssuerNameType. Το όνομα της τράπεζας ή του οργανισμού που χορηγεί την κάρτα (π.χ. τράπεζα).

IssuerNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
IssuerNameType__BelongsTo__PaymentCardType	int		Foreign key στον πίνακα PaymentCardType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) PaymentCardType στο οποίο ανήκει το στοιχείο IssuerNameType.
BankId	varchar(64)		Ο κωδικός της τράπεζας που εκδίδει την κάρτα.

37: BankAcctType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) BankAcctType. Οι τραπεζικοί λογαριασμοί του πελάτη, είτε για επιταγές, είτε για ηλεκτρονική μεταφορά χρημάτων.

BankAcctTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
BankAcctName	varchar(64)		Το όνομα του τραπεζικού λογαριασμού.
BankId	varchar(64)		Ο αναγνωριστικός κωδικός της τράπεζας.
AcctType	varchar(32)		Περιγράφει τον τραπεζικό λογαριασμό που χρησιμοποιείται για την χρηματοδότηση του ταξιδιού, όπως: τρεχούμενος, ταμειευτήριο, κτλ.
BankAcctNumber	varchar(32)		Ο κωδικός του λογαριασμού που δίνεται από την τράπεζα.

38: DirectBillType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) DirectBillType. Το όνομα και η τοποθεσία της επιχείρησης για την αποστολή τιμολογίου για τα ποσά που καταβλήθηκαν για τις τουριστικές υπηρεσίες.

DirectBillTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
DirectBillType__Has__Address	int		Foreign key στον πίνακα Address_1 που είναι ο τυπος του στοιχείου Address, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) DirectBillType ο οποίος

		αποθηκεύεται στον παρόντα πίνακα.
DirectBillType__Has__CompanyName	int	Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου CompanyName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) DirectBillType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DirectBill_Id	varchar(32)	Ο κωδικός του οργανισμού που θα χρεωθεί απευθείας για τις τουριστικές υπηρεσίες.

39: VoucherType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VoucherType. Ένα τρόπος πληρωμής με voucher ή coupon.

VoucherTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
SeriesCode	varchar(32)		Αναγνώριση σειράς κουπονιών ή αποδείξεων που αναγνωρίζονται μέσω σειριακών αριθμών.

40: ContactPersonType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) ContactPersonType. Τα ονόματα των ατόμων με τα οποία έχει επαφές ο πελάτης.

ContactPersonTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
ContactPersonType__Has__PersonName	int		Foreign key στον πίνακα PersonNameType που είναι ο τύπος του στοιχείου PersonName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) ContactPersonType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
ContactType	varchar(32)		Ο τύπος της επαφής στα πλαίσια της τουριστικής χρήσης, όπως: μόνιμη, προσωρινή, ταξιδιωτικός πράκτορας κτλ.
Relation	varchar(32)		Δείχνει τον τύπο της σχέσης με το άτομο ή την επιχείρηση στο προφίλ, όπως σύζυγος, τέκνο, οικογένεια, επιχειρηματικός συνεργάτης κτλ.
EmergencyFlag	Boolean		
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικείμενου σε μια συλλογή από παρόμοια αντικείμενα.

41: ContactPersonType__Has__Address			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα ContactPersonType με τον πίνακα Address_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου Address_1 ο οποίος είναι ο τυπος του στοιχειου Address.			
ContactPersonTypeID	int	foreign key	Foreign key στον πίνακα ContactPersonType.
AddressID	String	foreign key	Foreign key στον πίνακα πίνακα Address_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου Address_1 ο οποίος είναι ο τυπος του στοιχειου Address.

42: ContactPersonType__Has__URL			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα ContactPersonType με τον πίνακα URL_Type που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου URL_Type ο οποίος είναι ο τυπος του στοιχειου URL.			
ContactPersonTypeID	int	foreign key	Foreign key στον πίνακα ContactPersonType.
URLID	String	foreign key	Foreign key στον πίνακα πίνακα URL_Type που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου URL_Type ο οποίος είναι ο τυπος του στοιχειου URL.

43: ContactPersonType__Has__Telephone			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα ContactPersonType με τον πίνακα TelephoneInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TelephoneInfoType ο οποίος είναι ο τυπος του στοιχειου Telephone.			
ContactPersonTypeID	int	foreign key	Foreign key στον πίνακα ContactPersonType.
TelephoneID	String	foreign key	Foreign key στον πίνακα πίνακα TelephoneInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TelephoneInfoType ο οποίος είναι ο τυπος του στοιχειου Telephone.

44: ContactPersonType__Has__Email			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα ContactPersonType με τον πίνακα EmailType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmailType ο οποίος είναι ο τυπος του στοιχειου Email.			
ContactPersonTypeID	int	foreign key	Foreign key στον πίνακα ContactPersonType.
EmailID	String	foreign key	Foreign key στον πίνακα πίνακα EmailType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmailType ο οποίος είναι ο τυπος του στοιχειου Email.

45: ContactPersonType__Has__CompanyName			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα ContactPersonType με τον πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τυπος του στοιχειου CompanyName.			
ContactPersonTypeID	int	foreign key	Foreign key στον πίνακα ContactPersonType.
CompanyNameID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τυπος του στοιχειου CompanyName.

46: ContactPersonType__Has__EmployeeInfo			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα ContactPersonType με τον πίνακα EmployeeInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmployeeInfoType ο οποίος είναι ο τυπος του στοιχειου EmployeeInfo.			
ContactPersonTypeID	int	foreign key	Foreign key στον πίνακα ContactPersonType.

EmployeeInfoID	String	foreign key	Foreign key στον πίνακα EmployeeInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmployeeInfoType ο οποίος είναι ο τυπος του στοιχείου EmployeeInfo.
----------------	--------	-------------	---

47: EmployeeInfoType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) EmployeeInfoType. Employment identification; using an employee Id number, title, level within the company, and an indication of their status, i.e.: active, retired, on leave, or terminated from employment. Additional information about an employee can be entered into the element, including the name of the employee, if it differs from the person name identified as a customer or contact.

EmployeeInfoTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου EmployeeInfoType.
EmployeeId	varchar(16)		Ο κωδικός του εργαζόμενου.
EmployeeLevel	varchar(16)		Το επίπεδο του εργαζόμενου στην επιχείρηση (π.χ. η θέση του στην ιεραρχία) που του παρέχει προνόμια.
EmployeeTitle	varchar(16)		Ο τίτλος του εργαζόμενου στην επιχείρηση.
EmployeeStatus	varchar(200)		Η κατάσταση του εργαζόμενου.

48: DocumentType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) DocumentType. Περιέχει πληροφορίες για συγκεκριμένα έγγραφα.

DocumentTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
DocumentType__BelongsTo__CustomerType	int		Foreign key στον πίνακα CustomerType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τυπου) CustomerType στο οποίο ανήκει το στοιχείο DocumentType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
BirthDate	date		Η ημερομηνία δημιουργίας του εγγράφου,
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
Gender	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Male', 'Female', 'Unknown'
DocHolderName	varchar(64)		Το όνομα του ιδιοκτήτη του εγγράφου.
DocIssueAuthority	varchar(64)		Η ομάδα ή σχέση που χορήγησε το έγγραφο
DocIssueLocation	varchar(64)		Η τοποθεσία έκδοσης του εγγράφου
DocId	varchar(32)		Μοναδικός αναγνωριστικός αριθμός του εγγράφου.
DocType	varchar(200)		Καθορίζει τον τύπο του εγγράφου, π.χ. διαβατήριο, ταυτότητα, δίπλωμα οδήγησης κτλ.

49: DocumentType__Has__DocLimitations

Ενδιάμεσος πίνακας που συνδέει τον πίνακα DocumentType με τον πίνακα SimpleText64 που αποθηκεύει τον τύπο του στοιχείου DocLimitations.

DocumentTypeID	int	foreign	Foreign key στον πίνακα DocumentType.
----------------	-----	---------	---------------------------------------

		key	
DocLimitationsID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText64 που αποθηκεύει τον τύπο του στοιχείου DocLimitations.

50: CitizenCountryNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CitizenCountryNameType. Το όνομα της χώρας της οποίας ισχυρίζεται ότι είναι υπήκοος.

CitizenCountryNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CitizenCountryNameType__BelongsTo__CustomerType	int		Foreign key στον πίνακα CustomerType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) CustomerType στο οποίο ανήκει το στοιχείο CitizenCountryNameType.
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
Code	varchar(200)		Ο κωδικός της χώρας.

51: CustLoyaltyType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CustLoyaltyType. Περιέχει πληροφορίες για το πρόγραμμα ανταμοιβής συχνής χρήσης προσθέτοντας πόντους, για παρεχόμενες υπηρεσίες.

CustLoyaltyTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CustLoyaltyType__BelongsTo__CustomerType	int		Foreign key στον πίνακα CustomerType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) CustomerType στο οποίο ανήκει το στοιχείο CustLoyaltyType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
TravelSector	varchar(200)		Δηλώνει τον τουριστικό τομέα.
SingleVendorInd	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'SingleVndr', 'Alliance'
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.
ProgramID	varchar(16)		Δηλώνει την επιχείρηση στην οποία ανήκει το πρόγραμμα αξιοπιστίας.
MembershipId	varchar(32)		Μοναδικό αναγνωριστικό του μέλους στο πρόγραμμα, π.χ. αριθμός μέλους, αριθμός λογαριασμού κτλ.
LoyalLevel	varchar(16)		Δηλώνει ειδικά προνόμια του προγράμματος που δόθηκαν στο συγκεκριμένο άτομο.
SignupDate	date		

52: RelatedTravelerType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) RelatedTravelerType. Περιέχει πληροφορίες για τα προφίλ άλλων ταξιδιωτών που σχετίζονται με το συγκεκριμένο άτομο.

RelatedTravelerTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
RelatedTravelerType__BelongsTo__CustomerType	int		Foreign key στον πίνακα CustomerType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) CustomerType στο οποίο ανήκει το στοιχείο RelatedTravelerType.
RelatedTravelerType__Has__PersonName	int		Foreign key στον πίνακα PersonNameType που είναι ο τύπος του στοιχείου PersonName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) RelatedTravelerType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
RelatedTravelerType__Has__UniqueId	int		Foreign key στον πίνακα UniqueIdType που είναι ο τύπος του στοιχείου UniqueId, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) RelatedTravelerType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
Relation	varchar(32)		Δείχνει τον τύπο της σχέσης με το άτομο στο προφίλ, όπως σύζυγος, τέκνο, οικογένεια, επιχειρηματικός συνεργάτης κτλ.

53: PreferencesType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PreferencesType. Περιέχει συλλογές από τουριστικές προτιμήσεις ή ανάγκες του πελάτη.

PreferencesTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PreferencesType__BelongsTo__ProfileType	int		Foreign key στον πίνακα ProfileType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) ProfileType στο οποίο ανήκει το στοιχείο PreferencesType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.

54: PrefCollection

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PrefCollection.

PrefCollectionID	int	primary	Το πρωτεύον κλειδί του πίνακα.
------------------	-----	---------	--------------------------------

	key	
PrefCollection__BelongsTo__PreferencesType	int	Foreign key στον πίνακα PreferencesType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) PreferencesType στο οποίο ανήκει το στοιχείο PrefCollection.
ShareSynchInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
TravelPurpose	varchar(200)	Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

55: CommonPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετού τύπο (ComplexType) CommonPrefType. Προτιμήσεις ή ανάγκες που εμπεριέχονται σε μίς συλλογή αλλά είναι ανεξάρτητες από συγκεκριμένες τουριστικές υπηρεσίες.

CommonPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CommonPrefType__BelongsTo__PrefCollection	int		Foreign key στον πίνακα PrefCollection ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) PrefCollection στο οποίο ανήκει το στοιχείο CommonPrefType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
SmokingAllowed	Boolean		Δηλώνει εάν επιτρέπεται το κάπνισμα.
PrimaryLangID	varchar(200)		Δηλώνει την κύρια γλώσσα ομιλίας.
AltLangID	varchar(200)		Δηλώνει εναλλακτική γλώσσα ομιλίας.

56: CommonPrefType__Has__TicketDistribPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα TicketDistribPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TicketDistribPrefType ο οποίος είναι ο τυπος του στοιχείου TicketDistribPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
TicketDistribPrefID	String	foreign key	Foreign key στον πίνακα πίνακα TicketDistribPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TicketDistribPrefType ο οποίος είναι ο τυπος του στοιχείου TicketDistribPref.

57: CommonPrefType__Has__LoyaltyPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τυπος του στοιχείου LoyaltyPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
------------------	-----	-------------	---

LoyaltyPrefID	String	foreign key	Foreign key στον πίνακα πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τυπος του στοιχειου LoyaltyPref.
---------------	--------	-------------	---

58: CommonPrefType_Has_PaymentFormPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τυπος του στοιχειου PaymentFormPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
PaymentFormPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τυπος του στοιχειου PaymentFormPref.

59: CommonPrefType_Has_MealPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα MealPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MealPrefType ο οποίος είναι ο τυπος του στοιχειου MealPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
MealPrefID	String	foreign key	Foreign key στον πίνακα πίνακα MealPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MealPrefType ο οποίος είναι ο τυπος του στοιχειου MealPref.

60: CommonPrefType_Has_SpecRequestPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τυπος του στοιχειου SpecRequestPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
SpecRequestPrefID	String	foreign key	Foreign key στον πίνακα πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τυπος του στοιχειου SpecRequestPref.

61: CommonPrefType_Has_MediaEntertainPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα MediaEntertainPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MediaEntertainPrefType ο οποίος είναι ο τυπος του στοιχειου MediaEntertainPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
MediaEntertainPrefID	String	foreign key	Foreign key στον πίνακα πίνακα MediaEntertainPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MediaEntertainPrefType ο οποίος είναι ο τυπος του στοιχειου MediaEntertainPref.

62: CommonPrefType_Has_PetInfoPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CommonPrefType με τον πίνακα PetInfoPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PetInfoPrefType ο οποίος είναι ο τυπος του στοιχειου PetInfoPref.

CommonPrefTypeID	int	foreign key	Foreign key στον πίνακα CommonPrefType.
PetInfoPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PetInfoPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PetInfoPrefType ο οποίος είναι ο τυπος του στοιχειου PetInfoPref.

63: TicketDistribPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) TicketDistribPrefType. Περιέχει τον προτιμώμενο τρόπο (για την συγκεκριμένη συλλογή) διανομής εισιτηρίων			
TicketDistribPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου TicketDistribPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
DistribType	varchar(200)		Δηλώνει την μέθοδο διανομής εισιτηρίων, όπως φαξ, ηλεκτρονικό ταχυδρομείο (email), κτλ.
TicketTime	varchar(200)		

64: LoyaltyPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) LoyaltyPrefType. Περιέχει τα προγράμματα αξιοπιστίας που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.			
LoyaltyPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου LoyaltyPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.

65: PaymentFormPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PaymentFormPrefType. Τρόποι πληρωμής που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.			
PaymentFormPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PaymentFormPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.

66: MealPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) MealPrefType. Προτιμήσεις φαγητών και ποτών που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.			
MealPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου MealPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
MealType	varchar(200)		Ο τύπος του φαγητού.
FavoriteFood	varchar(64)		Προτιμώμενο φαγητό.
Beverage	varchar(64)		Προτιμώμενος τύπος ποτού.

67: SpecRequestPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) SpecRequestPrefType. Ειδικές απαιτήσεις που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.

SpecRequestPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου SpecRequestPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

68: MediaEntertainPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) MediaEntertainPrefType. Προτιμήσεις διασκέδασης και ψυχαγωγίας που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.

MediaEntertainPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου MediaEntertainPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

69: PetInfoPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PetInfoPrefType. Καθορίζει προτιμήσεις σχετικά με τα κατοικίδια που συνοδεύουν τον πελάτη.

PetInfoPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PetInfoPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

70: NamePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) NamePrefType. Το όνομα που θα χρησιμοποιηθεί στην συγκεκριμένη συλλογή προτιμήσεων.

NamePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
NamePrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) CommonPrefType στο οποίο ανήκει το στοιχείο NamePrefType.
NamePrefType__Has__PersonName	int		Foreign key στον πίνακα PersonNameType που είναι ο τύπος του στοιχείου PersonName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) NamePrefType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
NamePrefType__Has__UniqueId	int		Foreign key στον πίνακα UniqueIdType που είναι ο τύπος του στοιχείου UniqueId, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) NamePrefType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

71: PhonePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) PhonePrefType. Οι τηλεφωνικοί αριθμοί που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή

προτιμήσεων.			
PhonePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PhonePrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) CommonPrefType στο οποίο ανήκει το στοιχείο PhonePrefType.
PhonePrefType__Has__Telephone	int		Foreign key στον πίνακα TelephoneInfoType που είναι ο τύπος του στοιχείου Telephone, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) PhonePrefType ο οποίος αποθηκεύεται στον παρόντα πίνακα.

72: AddressPrefType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AddressPrefType. Οι διευθύνσεις που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.			
AddressPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AddressPrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) CommonPrefType στο οποίο ανήκει το στοιχείο AddressPrefType.
AddressPrefType__Has__Address	int		Foreign key στον πίνακα Address_1 που είναι ο τύπος του στοιχείου Address, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) AddressPrefType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.

73: InterestPrefType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) InterestPrefType. Τα ενδιαφέροντα του πελάτη που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.			
InterestPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
InterestPrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) CommonPrefType στο οποίο ανήκει το στοιχείο InterestPrefType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου InterestPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

74: InsurancePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) InsurancePrefType. Οι πολιτικές ταξιδιωτικής ασφάλειας που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.

InsurancePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
InsurancePrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) CommonPrefType στο οποίο ανήκει το στοιχείο InsurancePrefType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου InsurancePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.

75: SeatingPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) SeatingPrefType. Οι προτιμήσεις καθίσματος που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων

SeatingPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SeatingPrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) CommonPrefType στο οποίο ανήκει το στοιχείο SeatingPrefType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου SeatingPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
SeatDirection	varchar(16)		Η κατεύθυνση προς την οποία βρίσκεται το κάθισμα.
SeatLocation	varchar(200)		Η τοποθεσία του καθίσματος.
SeatPosition	varchar(200)		Η θέση του καθίσματος.
SeatRow	varchar(200)		Η γραμμή στην οποία βρίσκεται το κάθισμα

76: RelatedTravelerPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) RelatedTravelerPrefType. Τα ονόματα των σχετιζόμενων ταξιδιωτών που θα χρησιμοποιηθούν στην συγκεκριμένη συλλογή προτιμήσεων.

RelatedTravelerPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
RelatedTravelerPrefType__BelongsTo__CommonPrefType	int		Foreign key στον πίνακα CommonPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) CommonPrefType στο οποίο ανήκει το στοιχείο RelatedTravelerPrefType.
RelatedTravelerPrefType__Has__UniqueId	int		Foreign key στον πίνακα UniqueIdType που είναι ο τυπος του

		στοιχείου UniqueId, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) RelatedTravelerPrefType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
PreferLevel	varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

77: VehicleProfileRentalPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleProfileRentalPrefType. Προτιμήσεις για την ενοικίαση οχημάτων.

VehicleProfileRentalPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
VehicleProfileRentalPrefType__BelongsTo__PrefCollection	int		Foreign key στον πίνακα PrefCollection ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τυπου) PrefCollection στο οποίο ανήκει το στοιχείο VehicleProfileRentalPrefType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
SmokingAllowed	Boolean		Δηλώνει εάν επιτρέπεται το κάπνισμα.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
GasPrePay	Boolean		

78: VehicleProfileRentalPrefType__Has__VendorPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα VehicleProfileRentalPrefType με τον πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τυπος του στοιχείου VendorPref.

VehicleProfileRentalPrefTypeID	int	foreign key	Foreign key στον πίνακα VehicleProfileRentalPrefType.
VendorPrefID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τυπος του στοιχείου VendorPref.

79: VehicleProfileRentalPrefType__Has__LoyaltyPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα VehicleProfileRentalPrefType με τον πίνακα LoyaltyPref_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPref_1 ο οποίος είναι ο τυπος του στοιχείου LoyaltyPref.

VehicleProfileRentalPrefTypeID	int	foreign key	Foreign key στον πίνακα VehicleProfileRentalPrefType.
LoyaltyPrefID	String	foreign	Foreign key στον πίνακα πίνακα LoyaltyPref_1 που

		key	αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPref_1 ο οποίος είναι ο τυπος του στοιχειου LoyaltyPref.
--	--	-----	--

80: VehicleProfileRentalPrefType__Has__PaymentFormPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα VehicleProfileRentalPrefType με τον πίνακα PaymentFormPref_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPref_1 ο οποίος είναι ο τυπος του στοιχειου PaymentFormPref.

VehicleProfileRentalPrefTypeID	int	foreign key	Foreign key στον πίνακα VehicleProfileRentalPrefType.
PaymentFormPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormPref_1 που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPref_1 ο οποίος είναι ο τυπος του στοιχειου PaymentFormPref.

81: CompanyNamePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CompanyNamePrefType. Προσδιορίζει την προτιμώμενη επιχείρηση με βάση το όνομα της.

CompanyNamePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CompanyNamePrefType__Extends__CompanyNameType	int		Foreign key στον πίνακα CompanyNameType ο οποίος αποθηκεύει τον σύνθετο τυπο CompanyNameType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος CompanyNamePrefType).
PreferLevel		varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

82: RHPHPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) RHPHPrefType. Ορίζει την προτίμηση για κάποιο στοιχείο στο οποίο γίνεται αναφορά μέσω του Reference Place Holder (RPH). Συχνά ένα αντικείμενο αποθηκεύεται σε μια συλλογή από όμοια αντικείμενα, και χρειάζεται να δηλωθεί η προτίμηση χρήσης ενός από τα αντικείμενα αυτά. Αυτός ο σύνθετος τύπος μπορεί να χρησιμοποιηθεί για τον καθορισμό του επιπέδου προτίμησης και να παρέχει ένα δείκτη στο αντικείμενο μέσα στη συλλογή.

RHPHPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.

83: CoverageCoreType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CoverageCoreType. Ο σύνθετος τύπος CoverageCoreType ορίζει τα απαραίτητα και πιο συχνά χρησιμοποιούμενα τμήματα τα οποία σε συνδυασμό ορίζουν ένα συγκεκριμένο τύπο κάλυψης οχήματος.

CoverageCoreTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Code	varchar(8)		Επιτρέπει την χρήση συγκεκριμένου κωδικού που περιγράφει την κάλυψη που καθορίζεται.
CoverageType	varchar(200)		Καθορίζει τον τύπο της κάλυψης, π.χ. collision damage waiver

84: VehicleCoveragePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleCoveragePrefType. Ο σύνθετος τύπος VehicleCoveragePrefType ορίζει την προτίμηση

για έναν συγκεκριμένο τύπο ασφαλιστικής κάλυψης για την ενοικίαση οχήματος.			
VehicleCoveragePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
VehicleCoveragePrefType__BelongsTo__VehicleProfileRentalPrefType	int		Foreign key στον πίνακα VehicleProfileRentalPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) VehicleProfileRentalPrefType στο οποίο ανήκει το στοιχείο VehicleCoveragePrefType.
VehicleCoveragePrefType__Extends__CoverageCoreType	int		Foreign key στον πίνακα CoverageCoreType ο οποίος αποθηκεύει τον σύνθετο τυπο CoverageCoreType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος VehicleCoveragePrefType).
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

85: VehicleSpecialReqPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleSpecialReqPrefType. Ο σύνθετος τύπος VehicleSpecialReqPrefType ορίζει την προτίμηση για μια συγκεκριμένη ειδική αίτηση.

VehicleSpecialReqPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
VehicleSpecialReqPrefType__BelongsTo__VehicleProfileRentalPrefType	int		Foreign key στον πίνακα VehicleProfileRentalPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τυπου) VehicleProfileRentalPrefType στο οποίο ανήκει το στοιχείο VehicleSpecialReqPrefType.
PCDATA	varchar(255)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου VehicleSpecialReqPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

86: VehicleCoreType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleCoreType. The VehicleCoreType complex type identifies the core data that is used to describe a vehicle.

VehicleCoreTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AirConditionInd	Boolean		Δηλώνει την ύπαρξη κλιματισμού.
TransmissionType	varchar(200)		Καθορίζει ένα σύνολο έγκυρων τιμών για τον τύπο μετάδοσης κίνησης ενός οχήματος. Οι έγκυρες τιμές είναι «αυτόματο» και «μη-αυτόματο».

87: VehicleTypeType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleTypeType. Ο σύνθετος τύπος VehicleTypeType ορίζει το τι αναφέρεται ως τύπος ενός οχήματος. Η γκάμα των οχημάτων χωρίζεται σε τύπους για την υποβοήθηση της αποσαφήνισης της αίτησης και της περιγραφής ενός συγκεκριμένου οχήματος. Ο τύπος του οχήματος ορίζεται ως ο συνδυασμός της κατηγορίας οχήματος (αυτοκίνητο, φορτηγό κτλ.) και ο αριθμός των θυρών που υπάρχουν στο όχημα.

VehicleTypeTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
-------------------	-----	-------------	--------------------------------

VehicleTypeType__BelongsTo__VehicleCoreType	int	Foreign key στον πίνακα VehicleCoreType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) VehicleCoreType στο οποίο ανήκει το στοιχείο VehicleTypeType.
VehicleCategory	varchar(200)	Περιγράφει την κατηγορία του οχήματος.
DoorCount	BigInt	Ο αριθμός των θυρών.

88: VehicleClassType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) VehicleClassType. The VehicleClassType complex type defines what is referred to as the class of vehicle. Within a type of vehicle (for example, within the type "Car") there is still a wide range of different vehicles. A class of vehicle is used to define that specific vehicle within the broader range of the vehicle in that same vehicle type. Use of vehicle class in conjunction with a vehicle type aids in clarifying a request and aids in describing a specific vehicle. A vehicle class is currently defined as just the size of the vehicle.

VehicleClassTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
VehicleClassType__BelongsTo__VehicleCoreType	int		Foreign key στον πίνακα VehicleCoreType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) VehicleCoreType στο οποίο ανήκει το στοιχείο VehicleClassType.
Size	varchar(200)		Περιγράφει το μέγεθος του οχήματος.

89: VehiclePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) VehiclePrefType. Ο σύνθετος τύπος VehiclePrefType καθορίζει τις πληροφορίες που θα χρησιμοποιηθούν για τον καθορισμό της προτίμησης ενός οχήματος.

VehiclePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
VehiclePrefType__BelongsTo__VehicleProfileRentalPrefType	int		Foreign key στον πίνακα VehicleProfileRentalPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) VehicleProfileRentalPrefType στο οποίο ανήκει το στοιχείο VehiclePrefType.
VehiclePrefType__Extends__VehicleCoreType	int		Foreign key στον πίνακα VehicleCoreType ο οποίος αποθηκεύει τον σύνθετο τύπο VehicleCoreType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος VehiclePrefType).
TypePref	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
ClassPref	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
AirConditionPref	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
TransmissionPref	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

90: VehicleEquipmentCoreType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleEquipmentCoreType. Ο σύνθετος τύπος VehicleEquipmentCoreType περιέχει τα κύρια δεδομένα που περιγράφουν κάποιον ειδικό εξοπλισμό που σχετίζεται με το όχημα που θα ενοικιαστεί.

VehicleEquipmentCoreTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Quantity	BigInt		Η ποσότητα του ζητούμενου εξοπλισμού.
EquipType	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του ΟΤΑ.

91: VehicleEquipmentPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) VehicleEquipmentPrefType. Ο σύνθετος τύπος VehicleEquipmentPrefType ορίζει τα δεδομένα που θα χρησιμοποιηθούν για την παροχή πληροφοριών για τον προτιμώμενο ειδικό εξοπλισμό κατά την ενοικίαση ενός οχήματος.

VehicleEquipmentPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
VehicleEquipmentPrefType__BelongsTo__VehicleProfileRentalPrefType	int		Foreign key στον πίνακα VehicleProfileRentalPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) VehicleProfileRentalPrefType στο οποίο ανήκει το στοιχείο VehicleEquipmentPrefType.
VehicleEquipmentPrefType__Extends__VehicleEquipmentCoreType	int		Foreign key στον πίνακα VehicleEquipmentCoreType ο οποίος αποθηκεύει τον σύνθετο τύπο VehicleEquipmentCoreType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος VehicleEquipmentPrefType).
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

92: AirlinePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AirlinePrefType. Ορίζει μια συλλογή από προτιμήσεις που σχετίζονται με τα αεροπορικά ταξίδια.

AirlinePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AirlinePrefType__BelongsTo__PrefCollection	int		Foreign key στον πίνακα PrefCollection ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) PrefCollection στο οποίο ανήκει το στοιχείο AirlinePrefType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
SmokingAllowed	Boolean		Δηλώνει εάν επιτρέπεται το κάπνισμα.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PassengerTypeCode	varchar(200)		Κατηγορία επιβάτη χρησιμοποιώντας τυποποιημένους κωδικούς.
AirTicketType	varchar(200)		Τύπος αεροπορικού εισιτηρίου που προτιμάται. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω

		τιμές: 'eTicket', 'Paper'.
--	--	----------------------------

93: AirlinePrefType__Has__VendorPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τυπος του στοιχειου VendorPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
VendorPrefID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τυπος του στοιχειου VendorPref.

94: AirlinePrefType__Has__TicketDistribPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα TicketDistribPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TicketDistribPrefType ο οποίος είναι ο τυπος του στοιχειου TicketDistribPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
TicketDistribPrefID	String	foreign key	Foreign key στον πίνακα πίνακα TicketDistribPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TicketDistribPrefType ο οποίος είναι ο τυπος του στοιχειου TicketDistribPref.

95: AirlinePrefType__Has__LoyaltyPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τυπος του στοιχειου LoyaltyPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
LoyaltyPrefID	String	foreign key	Foreign key στον πίνακα πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τυπος του στοιχειου LoyaltyPref.

96: AirlinePrefType__Has__PaymentFormPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τυπος του στοιχειου PaymentFormPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
PaymentFormPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τυπος του στοιχειου PaymentFormPref.

97: AirlinePrefType__Has__AirportOriginPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα AirportPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AirportPrefType ο οποίος είναι ο τυπος του στοιχειου AirportOriginPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
AirportOriginPrefID	String	foreign key	Foreign key στον πίνακα πίνακα AirportPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AirportPrefType ο οποίος είναι ο τυπος του στοιχειου AirportOriginPref.

98: AirlinePrefType__Has__AirportRoutePref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα AirportPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AirportPrefType ο οποίος είναι ο τυπος του στοιχειου AirportRoutePref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
AirportRoutePrefID	String	foreign key	Foreign key στον πίνακα πίνακα AirportPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AirportPrefType ο οποίος είναι ο τυπος του στοιχειου AirportRoutePref.

99: AirlinePrefType _Has_ MealPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα MealPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MealPrefType ο οποίος είναι ο τυπος του στοιχειου MealPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
MealPrefID	String	foreign key	Foreign key στον πίνακα πίνακα MealPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MealPrefType ο οποίος είναι ο τυπος του στοιχειου MealPref.

100: AirlinePrefType _Has_ SpecRequestPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τυπος του στοιχειου SpecRequestPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
SpecRequestPrefID	String	foreign key	Foreign key στον πίνακα πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τυπος του στοιχειου SpecRequestPref.

101: AirlinePrefType _Has_ MediaEntertainPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα MediaEntertainPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MediaEntertainPrefType ο οποίος είναι ο τυπος του στοιχειου MediaEntertainPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
MediaEntertainPrefID	String	foreign key	Foreign key στον πίνακα πίνακα MediaEntertainPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MediaEntertainPrefType ο οποίος είναι ο τυπος του στοιχειου MediaEntertainPref.

102: AirlinePrefType _Has_ PetInfoPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AirlinePrefType με τον πίνακα PetInfoPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PetInfoPrefType ο οποίος είναι ο τυπος του στοιχειου PetInfoPref.

AirlinePrefTypeID	int	foreign key	Foreign key στον πίνακα AirlinePrefType.
PetInfoPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PetInfoPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PetInfoPrefType ο οποίος είναι ο τυπος του στοιχειου PetInfoPref.

103: LocationType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) LocationType. Code and optional string to describe a location point.

LocationTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(200)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου LocationType.
LocationCode	varchar(8)		Ο κωδικός της περιοχής.
CodeContext	varchar(32)		Ορίζει τον τύπο του κωδικού όπως IATA, ARC, ή κάποιος εσωτερικός κωδικός κτλ.

104: AirportPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AirportPrefType. Προτιμήσεις αεροδρομίου αναχώρησης, χρησιμοποιώντας τυποποιημένους κωδικούς αεροδρομίων (IATA).

AirportPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AirportPrefType__Extends__LocationType	int		Foreign key στον πίνακα LocationType ο οποίος αποθηκεύει τον σύνθετο τύπο LocationType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος AirportPrefType).
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

105: FlightTypePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) FlightTypePrefType. Δηλώνει προτιμήσεις για συγκεκριμένους τύπους πτήσεων, όπως απευθείας, ή με στάσεις.

FlightTypePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
FlightTypePrefType__BelongsTo__AirlinePrefType	int		Foreign key στον πίνακα AirlinePrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) AirlinePrefType στο οποίο ανήκει το στοιχείο FlightTypePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
FlightType	varchar(200)		Καθορίζει έναν συγκεκριμένο τύπο πτήσης. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Nonstop', 'Direct', 'Connection'.
MaxConnections	BigInt		Μέγιστος αριθμός στάσεων.

106: FareRestrictPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) FareRestrictPrefType. Δηλώνει προτιμήσεις για περιορισμούς αποδεκτών ή μη αποδεκτών περιορισμών στα αεροπορικά εισιτήρια για μια συγκεκριμένη ταξιδιωτική περίπτωση.

FareRestrictPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
FareRestrictPrefType__BelongsTo__AirlinePrefType	int		Foreign key στον πίνακα AirlinePrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) AirlinePrefType στο οποίο ανήκει το στοιχείο FareRestrictPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
FareRestriction	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

107: EquipmentType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) EquipmentType. Τριψήφιος κωδικός και περιγραφή ενός εξοπλισμού.

EquipmentTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
-----------------	-----	-------------	--------------------------------

PCDATA	varchar(64)	Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου EquipmentType.
AirEquipType	varchar(3)	Ο τύπος του εξοπλισμού.
ChangeofGauge	Boolean	

108: EquipmentTypePref

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) EquipmentTypePref.

EquipmentTypePrefID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
EquipmentTypePref__BelongsTo__AirlinePrefType	int		Foreign key στον πίνακα AirlinePrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AirlinePrefType στο οποίο ανήκει το στοιχείο EquipmentTypePref.
EquipmentTypePref__Extends__EquipmentType	int		Foreign key στον πίνακα EquipmentType ο οποίος αποθηκεύει τον σύνθετο τυπο EquipmentType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος EquipmentTypePref).
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

109: CabinPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CabinPrefType. Δηλώνει προτιμήσεις για την επιλογή αεροπορικής καμπίνας για μια συγκεκριμένη ταξιδιωτική περίπτωση.

CabinPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CabinPrefType__BelongsTo__AirlinePrefType	int		Foreign key στον πίνακα AirlinePrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AirlinePrefType στο οποίο ανήκει το στοιχείο CabinPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
Cabin	varchar(200)		Ο τύπος της καμπίνας. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'First', 'Business', 'Economy'

110: SeatPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) SeatPrefType. Ορίζει προτιμήσεις για καθίσματα και τύπους καθισμάτων.

SeatPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SeatPrefType__BelongsTo__AirlinePrefType	int		Foreign key στον πίνακα AirlinePrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AirlinePrefType στο οποίο ανήκει το στοιχείο SeatPrefType.
SmokingAllowed	Boolean		Δηλώνει εάν επιτρέπεται το κάπνισμα.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
SeatNumber	varchar(200)		Αριθμός καθίσματος.
SeatPreference	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του ΟΤΑ.

111: SSR_PrefType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) SSR_PrefType. Ορίζει προτιμήσεις για ειδικές υπηρεσίες που απαιτούνται για αεροπορικά ταξίδια, χρησιμοποιώντας τυποποιημένους κωδικούς της βιομηχανίας (SSR-OSI).			
SSR_PrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SSR_PrefType__BelongsTo__AirlinePrefType	int		Foreign key στον πίνακα AirlinePrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) AirlinePrefType στο οποίο ανήκει το στοιχείο SSR_PrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
SSR_Code	varchar(200)		Κωδικός της ειδικής υπηρεσίας που ζητείται. Αναφέρεται σε λίστα τυποποιημένων κωδικών.

112: HotelPrefType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) HotelPrefType. Περιέχει τις προτιμήσεις ξενοδοχείων.			
HotelPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
HotelPrefType__BelongsTo__PrefCollection	int		Foreign key στον πίνακα PrefCollection ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) PrefCollection στο οποίο ανήκει το στοιχείο HotelPrefType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
SmokingAllowed	Boolean		Δηλώνει εάν επιτρέπεται το κάπνισμα.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RatePlanCode	varchar(128)		Ο κωδικός του προτιμώμενου rate plan.
HotelGuestType	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

113: HotelPrefType__Has__LoyaltyPref			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τύπος του στοιχείου LoyaltyPref.			
HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
LoyaltyPrefID	String	foreign key	Foreign key στον πίνακα πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τύπος του στοιχείου LoyaltyPref.

114: HotelPrefType__Has__PaymentFormPref			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τύπος του στοιχείου PaymentFormPref.			

HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
PaymentFormPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τύπος του στοιχείου PaymentFormPref.

115: HotelPrefType _Has_ MealPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα MealPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MealPrefType ο οποίος είναι ο τύπος του στοιχείου MealPref.

HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
MealPrefID	String	foreign key	Foreign key στον πίνακα πίνακα MealPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MealPrefType ο οποίος είναι ο τύπος του στοιχείου MealPref.

116: HotelPrefType _Has_ SpecRequestPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τύπος του στοιχείου SpecRequestPref.

HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
SpecRequestPrefID	String	foreign key	Foreign key στον πίνακα πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τύπος του στοιχείου SpecRequestPref.

117: HotelPrefType _Has_ MediaEntertainPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα MediaEntertainPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MediaEntertainPrefType ο οποίος είναι ο τύπος του στοιχείου MediaEntertainPref.

HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
MediaEntertainPrefID	String	foreign key	Foreign key στον πίνακα πίνακα MediaEntertainPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου MediaEntertainPrefType ο οποίος είναι ο τύπος του στοιχείου MediaEntertainPref.

118: HotelPrefType _Has_ PetInfoPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα PetInfoPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PetInfoPrefType ο οποίος είναι ο τύπος του στοιχείου PetInfoPref.

HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
PetInfoPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PetInfoPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PetInfoPrefType ο οποίος είναι ο τύπος του στοιχείου PetInfoPref.

119: HotelPrefType _Has_ HotelChainPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα HotelPrefType με τον πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τύπος του στοιχείου HotelChainPref.

HotelPrefTypeID	int	foreign key	Foreign key στον πίνακα HotelPrefType.
HotelChainPrefID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τύπος του στοιχείου HotelChainPref.

120: PropertyNamePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PropertyNamePrefType. Το όνομα του προτιμώμενου ξενοδοχείου.

PropertyNamePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PropertyNamePrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο PropertyNamePrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PropertyNamePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

121: PropertyLocationPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PropertyLocationPrefType. Ορίζει προτιμήσεις για την τοποθεσία των ξενοδοχείων.

PropertyLocationPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PropertyLocationPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο PropertyLocationPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PropertyLocationPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PropertyLocationType	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

122: PropertyTypePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PropertyTypePrefType. Ορίζει προτιμήσεις για τους τύπους των ξενοδοχείων.

PropertyTypePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PropertyTypePrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο PropertyTypePrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PropertyTypePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PropertyType	varchar(128)		Ο τύπος του ξενοδοχείου.

123: PropertyClassPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PropertyClassPrefType. Ορίζει προτιμήσεις για την κατηγορία των ξενοδοχείων.			
PropertyClassPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PropertyClassPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο PropertyClassPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PropertyClassPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PropertyClassType	varchar(200)		Δηλώνει την κατάταξη του ξενοδοχείου όπως ορίζεται στον πίνακα HotelCategoryCode.

124: PropertyAmenityPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PropertyAmenityPrefType. Ορίζει προτιμήσεις για τις διευκολύνσεις των ξενοδοχείων.			
PropertyAmenityPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PropertyAmenityPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο PropertyAmenityPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PropertyAmenityPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PropertyAmenityType	varchar(200)		Ορίζει τις διευκολύνσεις που προσφέρονται από το ξενοδοχείο όπως περιέχονται στον πίνακα HoteAmenityCode.

125: RoomAmenityPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) RoomAmenityPrefType. Ορίζει προτιμήσεις για τις διευκολύνσεις δωματίων των ξενοδοχείων			
RoomAmenityPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
RoomAmenityPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο RoomAmenityPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου RoomAmenityPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
Quantity	BigInt		Used for Numeric values, from 1 to 999

		inclusive
RoomAmenity	varchar(200)	Δηλώνει τον τύπο των διευκολύνσεων δωματίου που προσφέρονται από το ξενοδοχείο όπως ορίζονται στον πίνακα RoomAmenityCode.

126: RoomLocationPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) RoomLocationPrefType. Ορίζει προτιμήσεις για την τοποθεσία των δωματίων των ξενοδοχείων.

RoomLocationPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
RoomLocationPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο RoomLocationPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου RoomLocationPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RoomLocationType	varchar(200)		Ορίζει την τοποθεσία του δωματίου στο ξενοδοχείο όπως ορίζεται στον πίνακα RoomLocationCode.

127: BedTypePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) BedTypePrefType. Ορίζει προτιμήσεις για το μέγεθος και τα χαρακτηριστικά των κρεβατιών των δωματίων των ξενοδοχείων.

BedTypePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
BedTypePrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο BedTypePrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου BedTypePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
BedType	varchar(200)		Δηλώνει τον τύπο κρεβατιού που ζητείται όπως ορίζεται στον πίνακα BedTypeCode.

128: FoodSrvcPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) FoodSrvcPrefType. Ορίζει προτιμήσεις για τον τύπο των υπηρεσιών σίτισης που παρέχει το ξενοδοχείο.

FoodSrvcPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
FoodSrvcPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο FoodSrvcPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου FoodSrvcPrefType.

PreferLevel	varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
FoodSrcvType	varchar(200)	Δηλώνει τους τύπους υπηρεσιών σίτισης που προσφέρονται από το ξενοδοχείο όπως ορίζονται στον πίνακα MainCuisineCode.

129: RecreationSrcvPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) RecreationSrcvPrefType. Ορίζει προτιμήσεις για τον τύπο των υπηρεσιών αναψυχής του ξενοδοχείου.

RecreationSrcvPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
RecreationSrcvPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο RecreationSrcvPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου RecreationSrcvPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
RecreationSrcvType	varchar(200)		Δηλώνει τις υπηρεσίες αναψυχής που παρέχονται από το ξενοδοχείο όπως ορίζονται στον πίνακα RecreationalSrcvCode.

130: BusinessSrcvPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) BusinessSrcvPrefType. Ορίζει προτιμήσεις για τον τύπο των επιχειρησιακών υπηρεσιών που παρέχει τον ξενοδοχείο.

BusinessSrcvPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
BusinessSrcvPrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) HotelPrefType στο οποίο ανήκει το στοιχείο BusinessSrcvPrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου BusinessSrcvPrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
BusinessSrcvType	varchar(200)		Δηλώνει τους τύπους επιχειρησιακών υπηρεσιών που παρέχονται από το ξενοδοχείο όπως ορίζονται στον πίνακα BusinessSrcvs.

131: PersonalSrcvPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PersonalSrcvPrefType. Ορίζει προτιμήσεις για τον τύπο των προσωπικών υπηρεσιών που παρέχει το ξενοδοχείο.

PersonalSrcvPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
------------------------	-----	-------------	--------------------------------

PersonalSrvcPrefType__BelongsTo__HotelPrefType	int	Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) HotelPrefType στο οποίο ανήκει το στοιχείο PersonalSrvcPrefType.
PCDATA	varchar(128)	Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PersonalSrvcPrefType.
PreferLevel	varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PersonalSrvcType	varchar(200)	Δηλώνει τον τύπο προσωπικών υπηρεσιών που παρέχονται από το ξενοδοχείο όπως ορίζονται στον πίνακα PersonalSrvcCode.

132: SecurityFeaturePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) SecurityFeaturePrefType. Ορίζει προτιμήσεις για τον τύπο των χαρακτηριστικών ασφάλειας του ξενοδοχείου.

SecurityFeaturePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SecurityFeaturePrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) HotelPrefType στο οποίο ανήκει το στοιχείο SecurityFeaturePrefType.
PCDATA	varchar(200)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου SecurityFeaturePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'

133: PhysChallFeaturePrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PhysChallFeaturePrefType. Ορίζει προτιμήσεις για τον τύπο των χαρακτηριστικών που απαιτούνται για την ικανοποίηση των αναγκών των ατόμων με ειδικές ανάγκες.

PhysChallFeaturePrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PhysChallFeaturePrefType__BelongsTo__HotelPrefType	int		Foreign key στον πίνακα HotelPrefType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) HotelPrefType στο οποίο ανήκει το στοιχείο PhysChallFeaturePrefType.
PCDATA	varchar(128)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου PhysChallFeaturePrefType.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
PhysChallFeature	varchar(200)		Δηλώνει τον τύπο των διευκολύνσεων που παρέχονται από το ξενοδοχείο για την εξυπηρέτηση των ατόμων με ειδικές ανάγκες, όπως ορίζονται στον πίνακα PhyslyChallengedFeatureCode.

134: OtherSrvcPrefType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο

(ComplexType) OtherSrvPrefType. Ορίζει άλλες τουριστικές προτιμήσεις που δεν εμπίπτουν στις προηγούμενες κατηγορίες.			
OtherSrvPrefTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
OtherSrvPrefType__BelongsTo__PrefCollection	int		Foreign key στον πίνακα PrefCollection ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) PrefCollection στο οποίο ανήκει το στοιχείο OtherSrvPrefType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
PreferLevel	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Only', 'Unacceptable', 'Preferred'
OtherSrvName	varchar(32)		Το όνομα των άλλων τουριστικών υπηρεσιών που ορίζονται στην συγκεκριμένη συλλογή προτιμήσεων.
TravelPurpose	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

135: OtherSrvPrefType__Has__VendorPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα OtherSrvPrefType με τον πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τυπος του στοιχείου VendorPref.

OtherSrvPrefTypeID	int	foreign key	Foreign key στον πίνακα OtherSrvPrefType.
VendorPrefID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNamePrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNamePrefType ο οποίος είναι ο τυπος του στοιχείου VendorPref.

136: OtherSrvPrefType__Has__LoyaltyPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα OtherSrvPrefType με τον πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τυπος του στοιχείου LoyaltyPref.

OtherSrvPrefTypeID	int	foreign key	Foreign key στον πίνακα OtherSrvPrefType.
LoyaltyPrefID	String	foreign key	Foreign key στον πίνακα πίνακα LoyaltyPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyPrefType ο οποίος είναι ο τυπος του στοιχείου LoyaltyPref.

137: OtherSrvPrefType__Has__PaymentFormPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα OtherSrvPrefType με τον πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τυπος του στοιχείου PaymentFormPref.

OtherSrvPrefTypeID	int	foreign key	Foreign key στον πίνακα OtherSrvPrefType.
PaymentFormPrefID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormPrefType ο οποίος είναι ο τυπος του στοιχείου PaymentFormPref.

138: OtherSrcvPrefType__Has__SpecRequestPref

Ενδιάμεσος πίνακας που συνδέει τον πίνακα OtherSrcvPrefType με τον πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τυπος του στοιχειου SpecRequestPref.

OtherSrcvPrefTypeID	int	foreign key	Foreign key στον πίνακα OtherSrcvPrefType.
SpecRequestPrefID	String	foreign key	Foreign key στον πίνακα πίνακα SpecRequestPrefType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου SpecRequestPrefType ο οποίος είναι ο τυπος του στοιχειου SpecRequestPref.

139: CompanyInfoType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CompanyInfoType.

CompanyInfoTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CompanyInfoType__BelongsTo__ProfileType	int		Foreign key στον πίνακα ProfileType ο οποίος αποθηκεύει το περιεχόμενο του στοιχειου (ή σύνθετου τυπου) ProfileType στο οποίο ανήκει το στοιχείο CompanyInfoType.

140: CompanyInfoType__Has__URL

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα URL_Type που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου URL_Type ο οποίος είναι ο τυπος του στοιχειου URL.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
URLID	String	foreign key	Foreign key στον πίνακα πίνακα URL_Type που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου URL_Type ο οποίος είναι ο τυπος του στοιχειου URL.

141: CompanyInfoType__Has__Email

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα EmailType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmailType ο οποίος είναι ο τυπος του στοιχειου Email.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
EmailID	String	foreign key	Foreign key στον πίνακα πίνακα EmailType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmailType ο οποίος είναι ο τυπος του στοιχειου Email.

142: CompanyInfoType__Has__TravelArranger

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τυπος του στοιχειου TravelArranger.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
TravelArrangerID	String	foreign key	Foreign key στον πίνακα πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τυπος του στοιχειου TravelArranger.

143: CompanyInfoType__Has__CompanyName

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τυπος του στοιχειου CompanyName.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
-------------------	-----	-------------	--

CompanyNameID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τυπος του στοιχειου CompanyName.
---------------	--------	-------------	---

144: CompanyInfoType__Has__AddressInfo

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα AddressInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AddressInfoType ο οποίος είναι ο τυπος του στοιχειου AddressInfo.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
AddressInfoID	String	foreign key	Foreign key στον πίνακα πίνακα AddressInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AddressInfoType ο οποίος είναι ο τυπος του στοιχειου AddressInfo.

145: CompanyInfoType__Has__TelephoneInfo

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα TelephoneInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TelephoneInfoType ο οποίος είναι ο τυπος του στοιχειου TelephoneInfo.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
TelephoneInfoID	String	foreign key	Foreign key στον πίνακα πίνακα TelephoneInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TelephoneInfoType ο οποίος είναι ο τυπος του στοιχειου TelephoneInfo.

146: CompanyInfoType__Has__BusinessLocale

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα AddressType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AddressType ο οποίος είναι ο τυπος του στοιχειου BusinessLocale.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
BusinessLocaleID	String	foreign key	Foreign key στον πίνακα πίνακα AddressType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου AddressType ο οποίος είναι ο τυπος του στοιχειου BusinessLocale.

147: CompanyInfoType__Has__PaymentForm

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα PaymentFormType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormType ο οποίος είναι ο τυπος του στοιχειου PaymentForm.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
PaymentFormID	String	foreign key	Foreign key στον πίνακα πίνακα PaymentFormType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου PaymentFormType ο οποίος είναι ο τυπος του στοιχειου PaymentForm.

148: CompanyInfoType__Has__ContactPerson

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα ContactPersonType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ContactPersonType ο οποίος είναι ο τυπος του στοιχειου ContactPerson.

CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
ContactPersonID	String	foreign key	Foreign key στον πίνακα πίνακα ContactPersonType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ContactPersonType ο οποίος είναι ο τυπος του στοιχειου ContactPerson.

149: CompanyInfoType__Has__LoyaltyProgram

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CompanyInfoType με τον πίνακα LoyaltyProgramType που

αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyProgramType ο οποίος είναι ο τύπος του στοιχείου LoyaltyProgram.			
CompanyInfoTypeID	int	foreign key	Foreign key στον πίνακα CompanyInfoType.
LoyaltyProgramID	String	foreign key	Foreign key στον πίνακα πίνακα LoyaltyProgramType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyProgramType ο οποίος είναι ο τύπος του στοιχείου LoyaltyProgram.

150: TravelArrangerType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) TravelArrangerType. Επιχειρήσεις ή άτομα που είναι υπεύθυνα για την κατάρτιση ταξιδιωτικών πλάνων ή συνδιαλλαγών είτε για τον πελάτη είτε για την επιχείρηση.

TravelArrangerTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
TravelArrangerType__Extends__CompanyNameType	int		Foreign key στον πίνακα CompanyNameType ο οποίος αποθηκεύει τον σύνθετο τύπο CompanyNameType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος TravelArrangerType).
ShareSynchInd		varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd		varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DefaultInd		Boolean	Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
RPH		varchar(200)	(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.
TravelArrangerType		varchar(32)	Ο τύπος του φορέα που καταρτίζει τα ταξιδιωτικά σχέδια ή τις συναλλαγές (π.χ. ταξιδιωτικό πρακτορείο).

151: LoyaltyProgramType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) LoyaltyProgramType. Προσδιορίζει την συνδρομή ή το πρόγραμμα αξιοπιστίας που προσφέρεται από την επιχείρηση, από το όνομα του προγράμματος.

LoyaltyProgramTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA		varchar(32)	Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου LoyaltyProgramType.
SingleVendorInd		varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'SingleVndr', 'Alliance'
RPH		varchar(200)	(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.
ProgramCode		varchar(16)	Ο κωδικός του προγράμματος.
LoyaltyLevel		varchar(16)	Δηλώνει ειδικά προνόμια του προγράμματος που δόθηκαν στο συγκεκριμένο άτομο.

152: AffiliationsType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AffiliationsType. Ορίζει μια οντότητα που παρέχει τουριστικές υπηρεσίες ή προνόμια σε έναν πελάτη ή μια επιχείρηση.

AffiliationsTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AffiliationsType__BelongsTo__ProfileType	int		Foreign key στον πίνακα ProfileType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) ProfileType στο οποίο ανήκει το στοιχείο AffiliationsType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.

153: AffiliationsType__Has__TravelArranger

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AffiliationsType με τον πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τύπος του στοιχείου TravelArranger.

AffiliationsTypeID	int	foreign key	Foreign key στον πίνακα AffiliationsType.
TravelArrangerID	String	foreign key	Foreign key στον πίνακα πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τύπος του στοιχείου TravelArranger.

154: OrganizationType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) OrganizationType. Συνδρομή σε οργανισμό που έχει τουριστικά προνόμια, προγράμματα ή εκπτώσεις.

OrganizationTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
OrganizationType__BelongsTo__AffiliationsType	int		Foreign key στον πίνακα AffiliationsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή σύνθετου τύπου) AffiliationsType στο οποίο ανήκει το στοιχείο OrganizationType.
OrganizationType__Has__OrgName	int		Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου OrgName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) OrganizationType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
OfficeType	varchar(200)		Δείχνει εάν πρόκειται για το κύριο γραφείο, το

		τοπικό γραφείο ή το παράρτημα ενός οργανισμού. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Main', 'Field', 'Division', 'Regional', 'Remote'.
--	--	---

155: OrganizationType__Has__TravelArranger

Ενδιάμεσος πίνακας που συνδέει τον πίνακα OrganizationType με τον πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τυπος του στοιχειου TravelArranger.

OrganizationTypeID	int	foreign key	Foreign key στον πίνακα OrganizationType.
TravelArrangerID	String	foreign key	Foreign key στον πίνακα πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τυπος του στοιχειου TravelArranger.

156: OrganizationType__Has__RelatedOrgName

Ενδιάμεσος πίνακας που συνδέει τον πίνακα OrganizationType με τον πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τυπος του στοιχειου RelatedOrgName.

OrganizationTypeID	int	foreign key	Foreign key στον πίνακα OrganizationType.
RelatedOrgNameID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τυπος του στοιχειου RelatedOrgName.

157: OrgMemberName

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) OrgMemberName.

OrgMemberNameID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
OrgMemberName__BelongsTo__OrganizationType	int		Foreign key στον πίνακα OrganizationType ο οποίος αποθηκεύει το περιεχόμενο του στοιχειου (ή σύνθετου τυπου) OrganizationType στο οποίο ανήκει το στοιχείο OrgMemberName.
OrgMemberName__Extends__PersonNameType	int		Foreign key στον πίνακα PersonNameType ο οποίος αποθηκεύει τον σύνθετο τυπο PersonNameType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος OrgMemberName).
Level	varchar(32)		Used for Character Strings, length 1 to 32. Level in employer organization (e.g. seniority) that coveys privileges.
Title	varchar(32)		Used for Character Strings, length 1 to 32. Title of employee in the employer company that conveys rank or privileges.
Id	varchar(32)		Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.

158: EmployerType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) EmployerType. Επιχείρηση ή οργανισμός στον οποίο εργάζεται ο πελάτης.

EmployerTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
----------------	-----	-------------	--------------------------------

EmployerType__BelongsTo__AffiliationsType	int	Foreign key στον πίνακα AffiliationsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) AffiliationsType στο οποίο ανήκει το στοιχείο EmployerType.
EmployerType__Has__CompanyName	int	Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου CompanyName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) EmployerType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
DefaultInd	Boolean	Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
EffectiveDate	date	Ημερομηνία έναρξης ισχύος.
ExpireDate	date	Ημερομηνία λήξης ισχύος.
OfficeType	varchar(200)	Δείχνει εάν πρόκειται για το κύριο γραφείο, το τοπικό γραφείο ή το παράρτημα ενός οργανισμού. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Main', 'Field', 'Division', 'Regional', 'Remote'.

159: EmployerType__Has__TravelArranger

Ενδιάμεσος πίνακας που συνδέει τον πίνακα EmployerType με τον πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τύπος του στοιχείου TravelArranger.

EmployerTypeID	int	foreign key	Foreign key στον πίνακα EmployerType.
TravelArrangerID	String	foreign key	Foreign key στον πίνακα πίνακα TravelArrangerType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου TravelArrangerType ο οποίος είναι ο τύπος του στοιχείου TravelArranger.

160: EmployerType__Has__LoyaltyProgram

Ενδιάμεσος πίνακας που συνδέει τον πίνακα EmployerType με τον πίνακα LoyaltyProgramType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyProgramType ο οποίος είναι ο τύπος του στοιχείου LoyaltyProgram.

EmployerTypeID	int	foreign key	Foreign key στον πίνακα EmployerType.
LoyaltyProgramID	String	foreign key	Foreign key στον πίνακα πίνακα LoyaltyProgramType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου LoyaltyProgramType ο οποίος είναι ο τύπος του στοιχείου LoyaltyProgram.

161: EmployerType__Has__EmployeeInfo

Ενδιάμεσος πίνακας που συνδέει τον πίνακα EmployerType με τον πίνακα EmployeeInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmployeeInfoType ο οποίος είναι ο τύπος του στοιχείου EmployeeInfo.

EmployerTypeID	int	foreign key	Foreign key στον πίνακα EmployerType.
EmployeeInfoID	String	foreign key	Foreign key στον πίνακα πίνακα EmployeeInfoType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου EmployeeInfoType ο οποίος είναι ο τύπος του στοιχείου EmployeeInfo.

162: EmployerType__Has__RelatedEmployer

Ενδιάμεσος πίνακας που συνδέει τον πίνακα EmployerType με τον πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τύπος του στοιχείου RelatedEmployer.

EmployerTypeID	int	foreign key	Foreign key στον πίνακα EmployerType.
RelatedEmployerID	String	foreign key	Foreign key στον πίνακα πίνακα CompanyNameType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου CompanyNameType ο οποίος είναι ο τύπος του στοιχείου RelatedEmployer.

163: EmployerType__Has__InternalRefNmbr			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα EmployerType με τον πίνακα FreeTextType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου FreeTextType ο οποίος είναι ο τυπος του στοιχειου InternalRefNmbr.			
EmployerTypeID	int	foreign key	Foreign key στον πίνακα EmployerType.
InternalRefNmbrID	String	foreign key	Foreign key στον πίνακα πίνακα FreeTextType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου FreeTextType ο οποίος είναι ο τυπος του στοιχειου InternalRefNmbr.

164: TravelClubType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) TravelClubType. Προσδιορίζει ένα τουριστικό όμιλο που προσφέρει ειδικά προνόμια και σχετικές υπηρεσίες.			
TravelClubTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
TravelClubType__BelongsTo__AffiliationsType	int		Foreign key στον πίνακα AffiliationsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχειου (ή σύνθετου τυπου) AffiliationsType στο οποίο ανήκει το στοιχείο TravelClubType.
TravelClubType__Has__TravelClubName	int		Foreign key στον πίνακα CompanyNameType που είναι ο τυπος του στοιχειου TravelClubName, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) TravelClubType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.

165: ClubMemberName			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) ClubMemberName.			
ClubMemberNameID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
ClubMemberName__BelongsTo__TravelClubType	int		Foreign key στον πίνακα TravelClubType ο οποίος αποθηκεύει το περιεχόμενο του στοιχειου (ή σύνθετου τυπου) TravelClubType στο οποίο ανήκει το στοιχείο ClubMemberName.
ClubMemberName__Extends__PersonNameType	int		Foreign key στον πίνακα PersonNameType ο οποίος αποθηκεύει τον σύνθετο τυπο PersonNameType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος ClubMemberName).
Id	varchar(32)		Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.

166: InsuranceType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) InsuranceType. Η τουριστική ασφάλεια του πελάτη ή της επιχείρησης.			
InsuranceTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
InsuranceType__BelongsTo__AffiliationsType	int		Foreign key στον πίνακα AffiliationsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) AffiliationsType στο οποίο ανήκει το στοιχείο InsuranceType.
InsuranceType__Has__InsuredName	int		Foreign key στον πίνακα PersonNameType που είναι ο τύπος του στοιχείου InsuredName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) InsuranceType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
InsuranceType__Has__InsuranceCompany	int		Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου InsuranceCompany, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) InsuranceType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
InsuranceType__Has__Underwriter	int		Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου Underwriter, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) InsuranceType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικειμένου σε μια συλλογή από παρόμοια αντικείμενα.
InsuranceType	varchar(32)		Ο τύπος της ασφαλιστικής πολιτικής.
PolicyNumber	varchar(32)		Ο αναγνωριστικός αριθμός της ασφαλιστικής πολιτικής που δίνεται από την ασφαλιστική εταιρεία.

167: AgreementsType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AgreementsType. Τμήμα του επιχειρηματικού προφίλ που περιέχει πληροφορίες σχετικά με τις επιχειρηματικές συμφωνίες.			
AgreementsTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AgreementsType__BelongsTo__ProfileType	int		Foreign key στον πίνακα ProfileType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) ProfileType στο οποίο ανήκει το στοιχείο AgreementsType.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από

		τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.

168: CertificationType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CertificationType. Πληροφορίες σχετικά με τις πιστοποιήσεις ή τις διαπιστεύσεις που διαθέτει η επιχείρηση.

CertificationTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CertificationType__BelongsTo__AgreementsType	int		Foreign key στον πίνακα AgreementsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AgreementsType στο οποίο ανήκει το στοιχείο CertificationType.
CertificationType__Extends__FreeTextType	int		Foreign key στον πίνακα FreeTextType ο οποίος αποθηκεύει τον σύνθετο τυπο FreeTextType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος CertificationType).
Id	varchar(32)		Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
SingleVendorInd	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'SingleVndr', 'Alliance'

169: AllianceConsortiumType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AllianceConsortiumType. Παρέχει πληροφορίες σχετικά με συμμαχίες και συνεταιρισμούς μελών που ομαδοποιούνται για την απόκτηση επιχειρηματικών συμφωνιών για τουριστικές υπηρεσίες και προνόμια.

AllianceConsortiumTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AllianceConsortiumType__BelongsTo__AgreementsType	int		Foreign key στον πίνακα AgreementsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AgreementsType στο οποίο ανήκει το στοιχείο AllianceConsortiumType.
Id	varchar(32)		Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.

170: AllianceMember

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AllianceMember.

AllianceMemberID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AllianceMember__BelongsTo__AllianceConsortiumType	int		Foreign key στον πίνακα AllianceConsortiumType ο οποίος

		αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) AllianceConsortiumType στο οποίο ανήκει το στοιχείο AllianceMember.
AllianceMember__Extends__CompanyNameType	int	Foreign key στον πίνακα CompanyNameType ο οποίος αποθηκεύει τον σύνθετο τύπο CompanyNameType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος AllianceMember).
MemberCode	varchar(8)	Δηλώνει τον κωδικό μέλους.

171: CommissionInfoType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CommissionInfoType. Περιέχει πληροφορίες για συμφωνίες προμήθειας με την επιχείρηση.

CommissionInfoTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CommissionInfoType__BelongsTo__AgreementsType	int		Foreign key στον πίνακα AgreementsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) AgreementsType στο οποίο ανήκει το στοιχείο CommissionInfoType.
CommissionInfoType__Extends__FreeTextType	int		Foreign key στον πίνακα FreeTextType ο οποίος αποθηκεύει τον σύνθετο τύπο FreeTextType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος CommissionInfoType).
Amount	String		Το ποσό/
CurrencyCode	varchar(200)		Ο κωδικός του νομίσματος.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DecimalPlaces	BigInt		
CommissionPlanCode	varchar(32)		Δηλώνει την οικονομική συμφωνία μεταξύ των συνεργατών.

SimpleText16

Ο πίνακας αυτός αποθηκεύει την τιμή των στοιχείων που είναι τύπου String μήκους 16 χαρακτήρων.

SimpleText16ID	int	Primary Key	Το πρωτεύον κλειδί του πίνακα.
Text16	Varchar(16)		Το πεδίο αυτό κρατά την λεκτική τιμή String των στοιχείων.

SimpleText64

Ο πίνακας αυτός αποθηκεύει την τιμή των στοιχείων που είναι τύπου String μήκους 64 χαρακτήρων.

SimpleText64ID	int	Primary Key	Το πρωτεύον κλειδί του πίνακα.
Text64	Varchar(64)		Το πεδίο αυτό κρατά την λεκτική τιμή String των στοιχείων.

SimpleText128

Ο πίνακας αυτός αποθηκεύει την τιμή των στοιχείων που είναι τύπου String μήκους 128 χαρακτήρων.			
SimpleText128ID	int	Primary Key	Το πρωτεύον κλειδί του πίνακα.
Text128	Varchar(128)		Το πεδίο αυτό κρατά την λεκτική τιμή String των στοιχείων.

Παράρτημα Γ

Οι πίνακες τα Βάσης Δεδομένων για τις κρατήσεις ξενοδοχείων που αντιστοιχούν στο RatePlan

1: RatePlanType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) RatePlanType. Περιέχει λεπτομέρειες σχετικά με το rate plan.			
RatePlanTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
RatePlanType__Has__RatePlanDescription	int		Foreign key στον πίνακα ParagraphType που είναι ο τύπος του στοιχείου RatePlanDescription, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) RatePlanType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
RatePlanCode	varchar(16)		Ο κωδικός του συγκεκριμένου rate plan που ορίζεται από το σύστημα.
RateIndicator	varchar(200)		Καθορίζει ειδικές συνθήκες που σχετίζονται με το κόστος. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'ChangeDuringStay', 'MultipleNights', 'Exclusive', 'OnRequest', 'LimitedAvailability', 'AvailableForSale', 'ClosedOut', 'OtherAvailable'.
RatePlanType	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.
RatePlanID	varchar(64)		Καθορίζει έναν ειδικό κωδικό ID που σχετίζεται με την τιμή και απαιτείται από την αίτηση κράτησης για την απόκτηση του συγκεκριμένου προϊόντος.

2: RatePlanType__Has__Guarantee			
Ενδιάμεσος πίνακας που συνδέει τον πίνακα RatePlanType με τον πίνακα GuaranteeType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου GuaranteeType ο οποίος είναι ο τύπος του στοιχείου Guarantee.			
RatePlanTypeID	int	foreign key	Foreign key στον πίνακα RatePlanType.
GuaranteeID	String	foreign key	Foreign key στον πίνακα πίνακα GuaranteeType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου GuaranteeType ο οποίος είναι ο τύπος του στοιχείου Guarantee.

3: GuaranteeType			
Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) GuaranteeType. Περιέχει πληροφορίες σχετικά με την εγγύηση που απαιτείται για να γίνει μια κράτηση.			
GuaranteeTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
GuaranteeType__Has__Deadline	int		Foreign key στον πίνακα DeadlineType που είναι ο τύπος του στοιχείου Deadline, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) GuaranteeType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
RetributionType	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές:

		'ResAutoCancelled', 'ResNotGuaranteed'
GuaranteeCode	varchar(32)	Ο κωδικός της εγγύησης.
GuaranteeType	varchar(200)	Ο τύπος της εγγύησης.
HoldTime	time	

4: GuaranteeType__Has__GuaranteeDescription

Ενδιάμεσος πίνακας που συνδέει τον πίνακα GuaranteeType με τον πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ParagraphType ο οποίος είναι ο τυπος του στοιχείου GuaranteeDescription.

GuaranteeTypeID	int	foreign key	Foreign key στον πίνακα GuaranteeType.
GuaranteeDescriptionID	String	foreign key	Foreign key στον πίνακα πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ParagraphType ο οποίος είναι ο τυπος του στοιχείου GuaranteeDescription.

5: DeadlineType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) DeadlineType. Το χρονικό περιθώριο πριν την έναρξη της ισχύος μιας ποινής ακύρωσης.

DeadlineTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AbsoluteDeadline	varchar(200)		Ορίζει την απόλυτη προθεσμία λήξης. Μπορεί να χρησιμοποιηθεί είτε αυτό το πεδίο είτε τα πεδία offset.
OffsetTimeUnit	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Year', 'Month', 'Week', 'Day', 'Hour', 'Second', 'FullDuration'
OffsetUnitMultiplier	BigInt		Used for Numeric values, from 0 to 999 inclusive. The number of units of DeadlineTimeUnit.
OffsetDropTime	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'BeforeArrival', 'AfterBooking'

6: ParagraphType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) ParagraphType. Νέα παράγραφος μιας υπό ενότητας ενός μηνύματος δομημένου κειμένου.

ParagraphTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Language	varchar(200)		Η γλώσσα στην οποία είναι γραμμένο το κείμενο.
CreateDateTime	date		Η ημερομηνία κατασκευής.
ParagraphNumber	BigInt		Ο αριθμός της παραγράφου
CreatorID	varchar(32)		Used for Character Strings, length 1 to 32. ID of creator. The creator could be a software system identifier or an identifier of an employee responsible for the creation.
LastModifyDateTime	date		Η ημερομηνία της τελευταίας τροποποίησης.
LastModifierID	varchar(32)		Ο αναγνωριστικός κωδικός αυτού που έκανε την τροποποίηση. Μπορεί να είναι ο αναγνωριστικός κωδικός ενός συστήματος λογισμικού, ή ο αναγνωριστικός κωδικός ενός εργαζόμενου που είναι υπεύθυνος για την τροποποίηση.

7: ParagraphTypeRepeatingChoice

Αυτός ο πίνακας χρησιμοποιείται για την υποστήριξη της δομής πολλαπλής επιλογής (RepeatingChoice).

RepeatingChoice__BelongsTo__ParagraphTypeID	int	foreign key	Foreign key στον πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του στοιχείου ParagraphType στο οποίο ανήκει αυτή η δομή RepeatingChoice.
SelectedTable	String	Foreign key	περιέχει το όνομα του πίνακα που επιλέγεται. Το όνομα του πίνακα που θα

			επιλεχθεί πρέπει να είναι ένα από τα: Text, SimpleText, SimpleText
SelectedTableID	int	Foreign key	foreign key στον πίνακα που επιλέγεται.

8: FormattedTextType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) FormattedTextType.

FormattedTextTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Language	varchar(200)		Η γλώσσα γραφής του κειμένου.
PCDATA	varchar(200)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου FormattedTextType.
Formatted	Boolean		Δηλώνει εάν το κείμενο είναι δομημένο ή όχι.

9: GuaranteesAccepted

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) GuaranteesAccepted.

GuaranteesAcceptedID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
GuaranteesAccepted__BelongsTo__GuaranteeType	int		Foreign key στον πίνακα GuaranteeType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) GuaranteeType στο οποίο ανήκει το στοιχείο GuaranteesAccepted.

10: PaymentFormType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PaymentFormType. Περιγράφει τους τρόπους πληρωμής που διαθέτει το άτομο.

PaymentFormTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
SelectedTable	varchar(200)		Ο πίνακας που επιλέγεται. Το όνομα του πίνακα που θα επιλεχθεί πρέπει να είναι ένα από τα: PaymentCard, BankAcct, DirectBill, Voucher
SelectedTableID	int		Το primary Key του SelectedTable.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
CostCenterId	varchar(32)		Κωδικός για την αναγνώριση της διεύθυνσης έκδοσης λογαριασμών.
RPH	varchar(200)		(Reference Place Holder) – ένας δείκτης για την αναγνώριση ενός αντικείμενου σε μια συλλογή από παρόμοια αντικείμενα.

11: PaymentCardType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) PaymentCardType. Πληροφορίες για μια συγκεκριμένη πιστωτική κάρτα.

PaymentCardTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PaymentCardType__Has__Address	int		Foreign key στον πίνακα AddressType που είναι ο τυπος του στοιχείου Address, το οποίο περιλαμβάνεται στον σύνθετο τυπο (Complex Type) PaymentCardType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του

		προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)	Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
EffectiveDate	varchar(200)	Ημερομηνία έναρξης ισχύος.
ExpireDate	varchar(200)	Ημερομηνία λήξης ισχύος.
CardHolderName	varchar(64)	Το όνομα του ιδιοκτήτη της κάρτας.
CardType	varchar(200)	Δείχνει τον τύπο της μαγνητικής κάρτας.
CardCode	varchar(200)	Ο κωδικός της κάρτας.
CardNumber	varchar(200)	Ο αριθμός της πιστωτικής κάρτας που αναγράφεται ανάγλυφα πάνω στην κάρτα.
SeriesCode	varchar(200)	Τα ψηφία επαλήθευσης που τυπώνονται στην κάρτα μαζί με τον ανάγλυφο αριθμό.

12: AddressType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AddressType.

AddressTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Type	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
BldgRoom	varchar(64)		Το όνομα του κτιρίου, το δωμάτιο, το διαμέρισμα ή ο αριθμός της σουίτας.
CityName	varchar(64)		Το όνομα της πόλης, π.χ. Χανιά.
PostalCode	varchar(16)		Ταχυδρομικός κώδικας.
County	varchar(32)		Το όνομα του Δήμου, π.χ. Δήμος Ακρωτηρίου.
FormattedInd	Boolean		Ορίζει την δόμηση της πληροφορίας σε ένα σύνολο πεδίων ή την απλή περιγραφή της σε ένα string.

13: AddressType__Has__AddressLine

Ενδιάμεσος πίνακας που συνδέει τον πίνακα AddressType με τον πίνακα SimpleText64 που αποθηκεύει τον τύπο του στοιχείου AddressLine.

AddressTypeID	int	foreign key	Foreign key στον πίνακα AddressType.
AddressLineID	int	foreign key	Foreign key στον πίνακα απλού κειμένου SimpleText64 που αποθηκεύει τον τύπο του στοιχείου AddressLine.

14: StreetNmbrType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) StreetNmbrType. Περιέχει την οδό και τον αριθμό επί της οδού.

StreetNmbrTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
StreetNmbrType__BelongsTo__AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AddressType στο οποίο ανήκει το στοιχείο StreetNmbrType.

PCDATA	varchar(64)	Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου StreetNmbrType.
PO_Box	varchar(16)	Ο αριθμός του γραμματοκιβωτίου.

15: StateProvType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) StateProvType. Το όνομα του νομού ή της περιοχής ή κάποιος κωδικός για τον προσδιορισμό της τοποθεσίας.

StateProvTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
StateProvType__BelongsTo__AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AddressType στο οποίο ανήκει το στοιχείο StateProvType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου StateProvType.
StateCode	varchar(8)		Ο κωδικός του νομού.

16: CountryNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CountryNameType. Το όνομα της χώρας που περιέχεται στην διεύθυνση.

CountryNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CountryNameType__BelongsTo__AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AddressType στο οποίο ανήκει το στοιχείο CountryNameType.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου CountryNameType.
Code	varchar(200)		Τυποποιημένος κωδικός χώρας (ISO 3166).

17: IssuerNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) IssuerNameType. Το όνομα της τράπεζας ή του οργανισμού που χορηγεί την κάρτα (π.χ. τράπεζα).

IssuerNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
IssuerNameType__BelongsTo__PaymentCardType	int		Foreign key στον πίνακα PaymentCardType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) PaymentCardType στο οποίο ανήκει το στοιχείο IssuerNameType.
BankId	varchar(64)		Ο κωδικός της τράπεζας που εκδίδει την κάρτα.

18: BankAcctType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) BankAcctType. Οι τραπεζικοί λογαριασμοί του πελάτη, είτε για επιταγές, είτε για ηλεκτρονική μεταφορά χρημάτων.

BankAcctTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες

		marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
BankAcctName	varchar(64)	Το όνομα του τραπεζικού λογαριασμού.
BankId	varchar(64)	Ο αναγνωριστικός κωδικός της τράπεζας.
AcctType	varchar(32)	Περιγράφει τον τραπεζικό λογαριασμό που χρησιμοποιείται για την χρηματοδότηση του ταξιδιού, όπως: τρεχούμενος, ταμειευτηρίου, κτλ.
BankAcctNumber	varchar(32)	Ο κωδικός του λογαριασμού που δίνεται από την τράπεζα.

19: DirectBillType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) DirectBillType. Το όνομα και η τοποθεσία της επιχείρησης για την αποστολή τιμολογίου για τα ποσά που καταβλήθηκαν για τις τουριστικές υπηρεσίες.

DirectBillTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
DirectBillType__Has__Address	int		Foreign key στον πίνακα Address_1 που είναι ο τύπος του στοιχείου Address, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) DirectBillType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
DirectBillType__Has__CompanyName	int		Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου CompanyName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) DirectBillType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ShareSynchInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για συγχρονισμό των πληροφοριών που διατηρούνται από άλλους παροχείς τουριστικών υπηρεσιών. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
ShareMarketInd	varchar(200)		Καθορίζει την άδεια να διαμοιραστούν τα δεδομένα του προφίλ για ανάγκες marketing. Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Yes', 'No', 'Inherit'.
DirectBill_Id	varchar(32)		Ο κωδικός του οργανισμού που θα χρεωθεί απευθείας για τις τουριστικές υπηρεσίες.

20: AddressInfoType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AddressInfoType. Πληροφορίες σχετικά με μια διεύθυνση προσδιορίζοντας μια τοποθεσία για συγκεκριμένους σκοπούς.

AddressInfoTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AddressInfoType__Extends__AddressType	int		Foreign key στον πίνακα AddressType ο οποίος αποθηκεύει τον σύνθετο τύπο AddressType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος AddressInfoType).
DefaultInd	Boolean		Ορίζει την ύπαρξη προεπιλεγμένης τιμής.
UseType	varchar(200)		Περιγράφει την χρήση της διεύθυνσης.

21: CompanyNameType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CompanyNameType. Προσδιορίζει μια επιχείρηση με βάση το όνομα της.

CompanyNameTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
PCDATA	varchar(64)		Το πεδίο αυτό περιέχει την λεκτική τιμή (String) του στοιχείου CompanyNameType.
Code	varchar(8)		Αναγνωριστικός κωδικός της επιχείρησης.

CodeContext	varchar(32)	Καθορίζει τον τύπο του αναγνωριστικού αριθμού, για παράδειγμα DUNS, IATA ή κάποιος εσωτερικός κωδικός κτλ.
CompanyShortName	varchar(32)	Ένα σύντομο όνομα της επιχείρησης.
TravelSector	varchar(200)	Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του ΟΤΑ.

22: VoucherType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) VoucherType. Ένα τρόπος πληρωμής με voucher ή coupon.

VoucherTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
EffectiveDate	date		Ημερομηνία έναρξης ισχύος.
ExpireDate	date		Ημερομηνία λήξης ισχύος.
SeriesCode	varchar(32)		Αναγνώριση σειράς κουπονιών ή αποδείξεων που αναγνωρίζονται μέσω σειριακών αριθμών.

23: GuaranteeAccepted

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) GuaranteeAccepted.

GuaranteeAcceptedID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
GuaranteeAccepted__BelongsTo__GuaranteesAccepted	int		Foreign key στον πίνακα GuaranteesAccepted ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) GuaranteesAccepted στο οποίο ανήκει το στοιχείο GuaranteeAccepted.
GuaranteeAccepted__Extends__PaymentFormType	int		Foreign key στον πίνακα PaymentFormType ο οποίος αποθηκεύει τον σύνθετο τυπο PaymentFormType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος GuaranteeAccepted).
Default	Boolean		

24: CommentType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) CommentType. Περιέχει μια συλλογή από σχόλια. Τα σχόλια αφορούν σε οποιοδήποτε τμήμα της κράτησης (σε όλη την κράτηση ή σε τμήμα αυτής). Τα σχόλια που αφορούν σε κάθε τμήμα καθορίζονται από τα ResCommentsRPHs.

CommentTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CommentType__BelongsTo__GuaranteeType	int		Foreign key στον πίνακα GuaranteeType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) GuaranteeType στο οποίο ανήκει το στοιχείο CommentType.

25: Comment

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) Comment.

CommentID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Comment__BelongsTo__CommentType	int		Foreign key στον πίνακα CommentType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) CommentType στο οποίο ανήκει το στοιχείο Comment.
Comment__Extends__ParagraphType	int		Foreign key στον πίνακα ParagraphType ο οποίος αποθηκεύει τον σύνθετο τυπο ParagraphType τον οποίο επεκτείνει το στοιχείο (ή σύνθετος τύπος Comment).

CommentOriginatorCode	varchar(16)	Used for Character Strings, length 1 to 16. CommentOriginatorCode : String Unique identifier for the system which created the comment.
GuestViewable	Boolean	

26: CancelPenaltiesType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CancelPenaltiesType.

CancelPenaltiesTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CancelPenaltiesType__BelongsTo__RatePlanType	int		Foreign key στον πίνακα RatePlanType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) RatePlanType στο οποίο ανήκει το στοιχείο CancelPenaltiesType.

27: CancelPenaltyType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CancelPenaltyType. Ορίζει την συγκεκριμένη πολιτική ακύρωσης που εφαρμόζει το ξενοδοχείο.

CancelPenaltyTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CancelPenaltyType__BelongsTo__CancelPenaltiesType	int		Foreign key στον πίνακα CancelPenaltiesType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) CancelPenaltiesType στο οποίο ανήκει το στοιχείο CancelPenaltyType.
CancelPenaltyType__Has__Deadline	int		Foreign key στον πίνακα DeadlineType που είναι ο τύπος του στοιχείου Deadline, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CancelPenaltyType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
ConfirmClassCode	varchar(16)		Κωδικός επιβεβαίωσης.
PolicyCode	varchar(16)		Κωδικός της πολιτικής.
NonRefundable	Boolean		Δηλώνει εάν το ποσό της ποινής μπορεί να επιστραφεί ή όχι.

28: CancelPenaltyType__Has__PenaltyDescription

Ενδιάμεσος πίνακας που συνδέει τον πίνακα CancelPenaltyType με τον πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ParagraphType ο οποίος είναι ο τύπος του στοιχείου PenaltyDescription.

CancelPenaltyTypeID	int	foreign key	Foreign key στον πίνακα CancelPenaltyType.
PenaltyDescriptionID	String	foreign key	Foreign key στον πίνακα πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ParagraphType ο οποίος είναι ο τύπος του στοιχείου PenaltyDescription.

29: AmountPercentType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) AmountPercentType. Defines the percentage basis for calculating the fee amount or the amount .

AmountPercentTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AmountPercentType__BelongsTo__CancelPenaltyType	int		Foreign key στον πίνακα CancelPenaltyType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) CancelPenaltyType στο οποίο ανήκει το στοιχείο AmountPercentType.
Amount	String		Το ποσό.

Percent	String	Το ποσοστό.
CurrencyCode	varchar(200)	Το νόμισμα.
DecimalPlaces	BigInt	Τα δεκαδικά ψηφία.
TaxInclusive	Boolean	
FeesInclusive	Boolean	
NmbrOfNights	BigInt	Ο αριθμός των διανυκτερεύσεων.
BasisType	varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'FullStay', 'Nights', 'FirstLast'

30: TaxesType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) TaxesType. Ορίζει μια συλλογή από φόρους.

TaxesTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
TaxesType__BelongsTo__AmountPercentType	int		Foreign key στον πίνακα AmountPercentType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) AmountPercentType στο οποίο ανήκει το στοιχείο TaxesType.
Amount	String		Το ποσό.
CurrencyCode	varchar(200)		Το νόμισμα.
DecimalPlaces	BigInt		Τα δεκαδικά ψηφία.

31: Tax

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) Tax.

TaxID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Tax__BelongsTo__TaxesType	int		Foreign key στον πίνακα TaxesType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τυπου) TaxesType στο οποίο ανήκει το στοιχείο Tax.
Type	varchar(200)		Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Inclusive', 'Exclusive', 'Cumulative'
Amount	String		Το ποσό.
Percent	String		Το ποσοστό.
CurrencyCode	varchar(200)		Το νόμισμα.
Code	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του OTA.

32: Tax _Has_ TaxDescription

Ενδιάμεσος πίνακας που συνδέει τον πίνακα Tax με τον πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ParagraphType ο οποίος είναι ο τυπος του στοιχείου TaxDescription.

TaxID	int	foreign key	Foreign key στον πίνακα Tax.
TaxDescriptionID	String	foreign key	Foreign key στον πίνακα πίνακα ParagraphType που αποθηκεύει τα περιεχόμενα του σύνθετου τύπου ParagraphType ο οποίος είναι ο τυπος του στοιχείου TaxDescription.

33: RatePlanInclusions

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) RatePlanInclusions.

RatePlanInclusionsID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
----------------------	-----	-------------	--------------------------------

RatePlanInclusions__BelongsTo__RatePlanType	int	Foreign key στον πίνακα RatePlanType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) RatePlanType στο οποίο ανήκει το στοιχείο RatePlanInclusions.
RatePlanInclusions__Has__RatePlanInclusionDescription	int	Foreign key στον πίνακα ParagraphType που είναι ο τύπος του στοιχείου RatePlanInclusionDescription, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) RatePlanInclusions ο οποίος αποθηκεύεται στον παρόντα πίνακα.
TaxInclusive	Boolean	
ServiceFeeInclusive	Boolean	

34: CommissionType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CommissionType. Περιέχει τις πληροφορίες που απαιτούνται για τον καθορισμό του ποσού που θα πληρωθεί στον ταξιδιωτικό πράκτορα. Το ποσό υπολογίζεται εφαρμόζοντας το ποσοστό στο ποσό πληρωμής και προσθέτοντας οποιοδήποτε άλλο ποσό απαιτείται.

CommissionTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
CommissionType__BelongsTo__RatePlanType	int		Foreign key στον πίνακα RatePlanType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετου τύπου) RatePlanType στο οποίο ανήκει το στοιχείο CommissionType.
CommissionType__Has__Comment	int		Foreign key στον πίνακα Comment_1 που είναι ο τύπος του στοιχείου Comment, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CommissionType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
CommissionType__Has__Uniqueld	int		Foreign key στον πίνακα UniqueldType που είναι ο τύπος του στοιχείου Uniqueld, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CommissionType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
CommissionType__Has__CommissionableAmount	int		Foreign key στον πίνακα CurrencyAmountType που είναι ο τύπος του στοιχείου CommissionableAmount, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CommissionType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
CommissionType__Has__PrepaidAmount	int		Foreign key στον πίνακα CurrencyAmountType που είναι ο τύπος του στοιχείου PrepaidAmount, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CommissionType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
CommissionType__Has__FlatCommission	int		Foreign key στον πίνακα CurrencyAmountType που είναι ο τύπος του στοιχείου FlatCommission, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CommissionType ο οποίος αποθηκεύεται στον παρόντα πίνακα.

CommissionType__Has__CommissionPayableAmount	int	Foreign key στον πίνακα CurrencyAmountType που είναι ο τύπος του στοιχείου CommissionPayableAmount, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) CommissionType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
Percent	String	Το ποσοστό
CurrencyCode	varchar(200)	Το νόμισμα.
DecimalPlaces	BigInt	Τα δεκαδικά ψηφία.
StatusType	varchar(200)	Το πεδίο αυτό μπορεί να πάρει μία από τις παρακάτω τιμές: 'Full', 'Partial', 'Non-paying', 'No-show', 'Adjustment.'
ReasonCode	varchar(8)	Ο λόγος.
BillToID	varchar(8)	

35: UniqueIdType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) UniqueIdType. Μοναδικό αναγνωριστικό που δίνεται σε κάθε προφίλ.

UniqueIdTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
UniqueIdType__Has__CompanyName	int		Foreign key στον πίνακα CompanyNameType που είναι ο τύπος του στοιχείου CompanyName, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) UniqueIdType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
URL	varchar(200)		Η ηλεκτρονική διεύθυνση.
Instance	varchar(32)		Η αναγνώριση μιας εγγραφής όπως υπάρχει σε μια συγκεκριμένη χρονική στιγμή. Χρησιμοποιείται στα μηνύματα ενημέρωσης, όπου ο αποστολέας πρέπει να διαβεβαιώσει τον server ότι η αίτηση για ενημέρωση αναφέρεται στην πιο πρόσφατη έκδοση του αντικειμένου που θα ενημερωθεί.
Type	varchar(200)		Αναφέρεται στον τύπο του αντικειμένου το οποίο καθορίζεται από το στοιχείο UniqueId. Παίρνει τιμές από τον πίνακα κωδικών του OTA.
Id	varchar(32)		Μοναδικό αναγνωριστικό που δίνεται από το σύστημα. Μπορεί να χρησιμοποιηθεί για να δείχνει σε ένα primary-key σε μια βάση δεδομένων.

36: CurrencyAmountType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) CurrencyAmountType.

CurrencyAmountTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
Amount	String		Το ποσό.
CurrencyCode	varchar(200)		Το νόμισμα.
DecimalPlaces	BigInt		Τα δεκαδικά ψηφία.

37: MealsIncluded

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή σύνθετο τύπο (ComplexType) MealsIncluded.

MealsIncludedID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
MealsIncluded__BelongsTo__RatePlanType	int		Foreign key στον πίνακα RatePlanType ο οποίος

		αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) RatePlanType στο οποίο ανήκει το στοιχείο MealsIncluded.
Breakfast	Boolean	Πρωινό
Lunch	Boolean	Μεσημεριανό
Dinner	Boolean	Βραδυνό.

38: AdditionalDetailsType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AdditionalDetailsType.

AdditionalDetailsTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AdditionalDetailsType__BelongsTo__RatePlanType	int		Foreign key στον πίνακα RatePlanType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) RatePlanType στο οποίο ανήκει το στοιχείο AdditionalDetailsType.

39: AdditionalDetailType

Αυτός ο πίνακας αποθηκεύει τις πληροφορίες που περιέχονται στο στοιχείο (Element) ή συνθετο τύπο (ComplexType) AdditionalDetailType.

AdditionalDetailTypeID	int	primary key	Το πρωτεύον κλειδί του πίνακα.
AdditionalDetailType__BelongsTo__AdditionalDetailsType	int		Foreign key στον πίνακα AdditionalDetailsType ο οποίος αποθηκεύει το περιεχόμενο του στοιχείου (ή συνθετού τύπου) AdditionalDetailsType στο οποίο ανήκει το στοιχείο AdditionalDetailType.
AdditionalDetailType__Has__DetailDescription	int		Foreign key στον πίνακα ParagraphType που είναι ο τύπος του στοιχείου DetailDescription, το οποίο περιλαμβάνεται στον σύνθετο τύπο (Complex Type) AdditionalDetailType ο οποίος αποθηκεύεται στον παρόντα πίνακα.
Type	varchar(200)		Χρησιμοποιείται για τους κωδικούς στον πίνακα κωδικών του ΟΤΑ.
Amount	String		Το ποσό.
CurrencyCode	varchar(200)		Το νόμισμα.
Code	varchar(16)		

SimpleText

Ο πίνακας αυτός αποθηκεύει την τιμή των στοιχείων που είναι τύπου String χωρίς συγκεκριμένο μήκος χαρακτήρων.

SimpleText16ID	int	Primary Key	Το πρωτεύον κλειδί του πίνακα.
Text	varchar(200)		Το πεδίο αυτό κρατά την λεκτική τιμή String των στοιχείων.

SimpleText64

Ο πίνακας αυτός αποθηκεύει την τιμή των στοιχείων που είναι τύπου String μήκους 64 χαρακτήρων.

SimpleText64ID	int	Primary Key	Το πρωτεύον κλειδί του πίνακα.
Text64	Varchar(64)		Το πεδίο αυτό κρατά την λεκτική τιμή String των στοιχείων.