

# Ηλεκτρονικός φάκελος ασθενών με χρήση XML Web Services και BPEL

---

Δημήτρης – Ηλίας Γκανάτσιος

A.M. ME07085

Τμήμα Ψηφιακών Συστημάτων

Πανεπιστήμιο Πειραιώς

Ιούλιος 2011

# Περίληψη

---

Η παρούσα διπλωματική εργασία με τίτλο «Ηλεκτρονικός Φάκελος Ασθενών με χρήση XML Web Services και BPEL» πραγματοποιήθηκε στο πλαίσιο του μεταπτυχιακού προγράμματος σπουδών του Πανεπιστημίου Πειραιά και του τμήματος «Ψηφιακών Συστημάτων», στην κατεύθυνση «Δικτυοκεντρικά Συστήματα».

Η εργασία έχει σαν σκοπό να παρουσιάσει την ανάγκη ανάπτυξης ενός συστήματος ιατρικού φακέλου ασθενών, η πρόσβαση στον οποίο πραγματοποιείται μέσω υπηρεσιών διαδικτύου (Web Services). Παράλληλα, στα πλαίσια της εργασίας, αναπτύχθηκε ένα σύστημα ιατρικών διαδικασιών με χρήση ροών εργασίας (Workflows).

Στην αρχή, θα δώσουμε μια αναλυτική περιγραφή των υπηρεσιών δικτύου και των ροών εργασίας αλλά και των τεχνολογιών οι οποίες συνθέτουν αυτές τις υπηρεσίες. Οι τεχνολογίες αυτές αποτελούνται κυρίως από τη γλώσσα XML, το πρωτόκολλο SOAP, τη γλώσσα περιγραφής των υπηρεσιών WSDL και τη γλώσσα BPEL.

Εν συνεχεία, θα αναλύσουμε τον ηλεκτρονικό φάκελο ασθενών, θα καταγράψουμε τα πλεονεκτήματα και τα μειονεκτήματα καθώς και κάποια πρότυπα ηλεκτρονικού φακέλου ασθενών. Ακόμη, προτείνουμε ενέργειες που θα ήταν καλό να γίνουν από τους αρμόδιους φορείς υγειονομικής περίθαλψης ούτως ώστε να βοηθηθεί η μετάβαση από τον παραδοσιακό φάκελο ασθενούς στον ηλεκτρονικό φάκελο.

Το σύστημα που αναπτύξαμε στα πλαίσια της διπλωματικής εργασίας βασίζεται στις ήδη προαναφερθείσες τεχνολογίες (Web Services, SOAP, XML, BPEL) και αποτελεί μια συλλογή εφαρμογών σχετικών με τον ηλεκτρονικό φάκελο ασθενών.

## Περιεχόμενα

Κατάλογος σχημάτων .....	6
1. Εισαγωγή.....	8
1.1 Εισαγωγή.....	8
1.2 Σκοπός.....	9
1.3 Δομή.....	9
2. Υπηρεσίες Διαδικτύου (Web Services) .....	11
2.1 Γενικά.....	11
2.2 Σχεδιαστικές αρχές δικτυακών υπηρεσιών .....	11
2.3 Μοντέλο λειτουργίας και ρόλοι .....	12
2.3 Πλεονεκτήματα.....	15
2.4 Οι βασικές τεχνολογίες που συνθέτουν τις υπηρεσίες διαδικτύου.....	18
2.4.1 Επεκτάσιμη Γλώσσα Σήμανσης (eXtensible Markup Language, XML) .....	18
2.4.1.1 Ανάλυση της XML.....	19
2.4.1.2 XML σχήμα (schema) .....	19
2.4.2 Απλό πρωτόκολλο πρόσβασης αντικειμένου (Simple Object Access Protocol, SOAP).....	20
2.4.2.1 Γενικά.....	20
2.4.2.2 Φάκελος (Envelope).....	22
2.4.2.3 Κωδικοποίηση.....	23
2.4.2.4 Επίκληση (Invocation).....	23
2.4.3 SOAP-RPC ή SOAP ανταλλαγή μηνυμάτων (SOAP-messaging).....	26
2.4.4 Γλώσσα περιγραφής Υπηρεσιών διαδικτύου (Web Service Description Language, WSDL).....	26
2.5 Representational State Transfer (REST).....	30
2.5.1 Περιορισμοί .....	31
2.7 Σύγκριση SOAP με REST .....	32
2.6 JavaScript Object Notation (JSON).....	33
2.6.1 Ιστορικά .....	33
2.6.2 Τύποι δεδομένων και σύνταξη .....	33
2.8 Σύγκριση XML με JSON .....	35

2.9 BPEL (Business Process Execution Language) .....	36
2.9.1 Στόχοι της BPEL .....	38
2.9.2 Η γλώσσα BPEL .....	39
2.7.3 Σύνθεση υπηρεσιών.....	39
2.7.4 Η BPEL στη σύνθεση των υπηρεσιών .....	40
4.5 Δομή Διαδικασίας.....	42
4.6 Παράδειγμα χρήσης BPEL.....	44
3. Cloud Computing .....	47
3.1 Επισκόπηση.....	47
3.2 Windows Azure .....	50
3.3 Εκτέλεση εφαρμογών στο Windows Azure .....	52
4. Ηλεκτρονικός Φάκελος Ασθενών .....	57
4.1 Ορισμοί .....	57
4.2 Χαρακτηριστικά .....	60
4.2.1 Πλεονεκτήματα.....	62
4.2.2 Προβλήματα – μειονεκτήματα .....	64
4.2.3 Η δομή των δεδομένων στον ΗΦΑ σε σχέση με τον μη ηλεκτρονικό φάκελο.....	65
4.3 Λειτουργίες Διαχείρισης .....	67
- Λήψη δεδομένων.....	68
- Αποθήκευση.....	68
- Επεξεργασία.....	68
- Επικοινωνία πληροφορίας.....	68
- Παρουσίαση πληροφορίας.....	69
- Ασφάλεια .....	69
4.4 Πρότυπα Ηλεκτρονικού Ιατρικού Φακέλου .....	69
4.4.1 Επίπεδο Υγείας 7 (Health Level 7) .....	69
Υπόβαθρο και οργανωτική δομή.....	70
Τομείς ενδιαφέροντος .....	71
Συγκεκριμένοι στόχοι HL7.....	72

4.4.2 Συνοχή φακέλου φροντίδας (Continuity of Care Record, CCR) .....	74
Σκοπός.....	75
Ανάγκη .....	75
Βασικά στοιχεία .....	75
Διαθεσιμότητα.....	76
Υπολογισμός Χρησιμότητας.....	76
Συνέπειες .....	77
4.4.4 Το διεθνές σύστημα κατηγοριοποίησης ασθενειών (ICD) .....	78
4.4.5 EN13606.....	79
4.5 Μετάβαση από τον κλασικό στον ηλεκτρονικό ιατρικό φάκελο .....	80
4.5.1 Ενημέρωση.....	80
4.5.2 Εκπαίδευση χρηστών .....	81
4.5.3 Παρακίνηση χρηστών .....	81
4.5.4 Περισσότερη εσωτερική οργάνωση .....	81
4.5.5 Συνεχής μέριμνα .....	81
5. Μεθοδολογία έρευνας .....	83
Χαρακτηριστικά της μεθόδου.....	84
Η μεθοδολογία έρευνας στην ανάπτυξη ενός οργανισμού .....	85
6. Υλοποίηση.....	90
6.1 Περιγραφή συστήματος.....	90
6.2 Αρχιτεκτονική του συστήματος .....	91
6.2.1 Αρχιτεκτονική δικτύου.....	91
6.1.2 Αρχιτεκτονική εφαρμογής .....	93
Εφαρμογή του Web Client.....	93
Εφαρμογή του Application Server .....	93
Βάση δεδομένων του συστήματος.....	94
Ασφάλεια του συστήματος και έλεγχος πρόσβασης .....	95
Εγκατάσταση του συστήματος .....	100
Α. η εγκατάσταση του web site.....	100

B. η εγκατάσταση του application server .....	100
Γ. η εγκατάσταση του database server .....	100
Δ. η εγκατάσταση του database security server .....	101
6.2 Ανάλυση των επιμέρους λειτουργιών του συστήματος.....	101
Η εφαρμογή από την πλευρά του χρήστη της κεντρικής γραμματείας.....	101
Η εφαρμογή από την πλευρά του χρήστη της γραμματείας ραντεβού των ακτινολογικών κέντρων .....	104
Η εφαρμογή από την πλευρά του χρήστη ακτινολόγου ιατρού .....	106
Η εφαρμογή από την πλευρά του χρήστη λογιστή του νοσοκομείου .....	108
Η εφαρμογή από την πλευρά του θεράποντα ιατρού του νοσοκομείου .....	110
Η εφαρμογή από την πλευρά του διαχειριστή.....	110
Ελεγκτικό όργανο.....	112
Καταγραφή συμβάντων.....	113
Στατιστικά που είναι δυνατόν να εκδοθούν μέσω της εφαρμογής .....	114
Διαχείριση λογαριασμών χρηστών και αντικειμένων .....	114
Εισαγωγή νέων χρηστών.....	114
Ορισμός επιπέδου πρόσβασης στους χρήστες .....	115
Δημιουργία βάσης δεδομένων.....	116
use case διαγράμματα για την εφαρμογή.....	122
Χρήση BPEL από την εφαρμογή.....	127
Χρήση HL7 από την εφαρμογή .....	128
7. Συμπεράσματα – χρήση ηλεκτρονικού φάκελου ασθενούς στην Ελλάδα.....	130
8. Βιβλιογραφία.....	132
9.1 Νομικό πλαίσιο για απόρρητο και ασφάλεια της επεξεργασίας .....	134
9.2 Παραδοχές για την ανάπτυξη της εφαρμογής .....	135

## Κατάλογος σχημάτων

Σχήμα 1 Λειτουργίες και ρόλοι υπηρεσιών [3]	15
Σχήμα 2 XML Σχήμα	20
Σχήμα 3 SOAP Φάκελος	21
Σχήμα 4 SOAP XML Μήνυμα	22
Σχήμα 5 C# κώδικας για υπηρεσία διαδικτύου η οποία προσθετει δυο δεκαδικουσ αριθμους	26
Σχήμα 6 WSDL περιγραφή σε XML	30
Σχήμα 7 JSON κώδικας	34
Σχήμα 8 XML αναπαράσταση του JSON	35
Σχήμα 9 XML αναπαράσταση του JSON με attributes	35
Σχήμα 10 επιχειρησιακές διαδικασίες ταξιδιωτικού γραφείου [3]	40
Σχήμα 11 Παράδειγμα BPEL [2]	42
Σχήμα 12 Παράδειγμα BPEL [3]	45
Σχήμα 13 Η πλατφόρμα Windows Azure υποστηρίζει εφαρμογές, δεδομένα και υποδομή στο σύννεφο [4]	48
Σχήμα 14 Το Windows Azure παρέχει υπολογιστικές δυνατότητες και χώρο αποθήκευσης βασισμένο σε Windows για εφαρμογές σύννεφου [4]	51
Σχήμα 15 Οι εφαρμογές στο Windows Azure μπορούν να αποτελούνται από Web ρόλους και Worker ρόλους, και ο το κάθε αντικείμενο του ρόλου να τρέχει στη δικιά του εικονική μηχανή [4]	53
Σχήμα 16 Η επεξεργασία της πληροφορίας οδηγεί σε δεδομένα [12]	58
Σχήμα 17 Παράδειγμα εφαρμογής παρακολούθησης ασθενών [12]	61
Σχήμα 18 Παράδειγμα εφαρμογής παρακολούθησης ασθενών [12]	62
Σχήμα 19 Δομή ιατρικού φάκελου ασθενούς [12]	66
Σχήμα 20 Δομή του ICD9	79
Σχήμα 21 έρευνα δράσης [15]	85
Σχήμα 22 "reflective spiral" [15]	88
Σχήμα 23 Προοδευτική επίλυση προβλήματος μέσω της έρευνας δράσης [15]	89
Σχήμα 24 Αναπαράσταση συστήματος	92
Σχήμα 25 Περιμετρική ασφάλεια	95
Σχήμα 26 Εξωτερική ασφάλεια	96
Σχήμα 27 Προστασία εξυπηρετητών διαδικτύου	97
Σχήμα 28 IP Security	98
Σχήμα 29 Κατάλογος χρηστών και ασφάλεια	99
Σχήμα 30 Ασφάλεια σε επίπεδο σκληρού δίσκου	100
Σχήμα 31 αλληλεπίδραση γραμματείας με άλλους χρήστες	104
Σχήμα 32 αλληλεπίδραση της γραμματείας ακτινολόγου	106
Σχήμα 33 αλληλεπίδραση ακτινολόγου	108
Σχήμα 34 αλληλεπίδραση λογιστή	109
Σχήμα 35 αλληλεπίδραση διαχειριστή	112
Σχήμα 36 αλληλεπίδραση auditor	113
Σχήμα 37 δημιουργία χρηστών	115

Σχήμα 38 κατανομή ακτινολογικών συσκευών ανά υγειονομική μονάδα	116
Σχήμα 39 Κατανομή υπαλλήλων ανά τμήμα	116
Σχήμα 40 κατανομή ασθενών ανά τμήμα	117
Σχήμα 41 πίνακες για αποθήκευση ακτινολογικών εξετάσεων	118
Σχήμα 42 αντικείμενα για ρόλους χρηστών	119
Σχήμα 43 αντικείμενα που επιστρέφονται από τα web services	120
Σχήμα 44 application workflow	121
Σχήμα 45 δεξιότητες στη χρήση ηλεκτρονικού υπολογιστή (παρατηρητήριο για την κοινωνία της πληροφορίας)	131



# 1. Εισαγωγή

---

## 1.1 Εισαγωγή

Επιλέξαμε το χώρο της υγείας επειδή αποτελεί νευραλγικό τμήμα κάθε σύγχρονης κοινωνίας, αφού αποτελεί μέρος της Κοινωνικής Πρόνοιας κάθε κράτους και σημαντικό παράγοντα της ποιότητας ζωής των πολιτών. Στο χώρο της υγείας δραστηριοποιούνται κρατικοί και ιδιωτικοί φορείς που συνεργάζονται και αλληλοσυμπληρώνονται ως προς τις υπηρεσίες υγείας που προσφέρουν στο κοινωνικό σύνολο.

Παραδοσιακά, στο χώρο της υγείας, η ανταλλαγή δεδομένων και ο προγραμματισμός ιατρικών πράξεων γίνεται με χειρόγραφο τρόπο, με συγγραφή πληροφοριών πάνω σε χαρτί, χρήση κλασσικού ιατρικού φακέλου κ.α. Ο τρόπος αυτός δημιουργούσε και δημιουργεί πάρα πολλά προβλήματα, με το μεγαλύτερο να είναι η καθυστέρηση στην περάτωση των διαφόρων ιατρικών διαδικασιών.

Ο τομέας της υγείας αποτελείται από μικρές, αυτόνομες μονάδες στις οποίες, μέχρι πρότινος, η ανταλλαγή δεδομένων ήταν ελάχιστη, ενώ οι χρήση ΤΠΕ (Τεχνολογίες Πληροφορικής και Τηλεπικοινωνιών) περιοριζόταν στα απολύτως απαραίτητα. Παρόλα αυτά, πιέσεις από το κράτος, από πολίτες και διάφορους οργανισμούς έχει οδηγήσει στην ανάπτυξη ΤΠΕ και στον τομέα της υγείας.

Έτσι, τα τελευταία χρόνια όμως, η κατάσταση έχει αρχίσει να παίρνει διαφορετική τροπή. Αναπτύσσονται ιατρικές εφαρμογές οι οποίες επικοινωνούν μεταξύ τους με ανοιχτά πρωτόκολλα και πρότυπα, αποθηκεύοντας πληροφορίες για τους ασθενείς, βοηθώντας το ιατρικό προσωπικό στην εκπόνηση διάγνωσης και επιταχύνοντας τις ιατρικές διαδικασίες.

Κατά την εκπόνηση της διπλωματικής εργασίας μελετήθηκαν οι υπάρχουσες πρακτικές στο χώρο της Υγείας (και συγκεκριμένα για τον παραδοσιακό φάκελο ασθενών). Κατά τη δημιουργία του πληροφοριακού συστήματος το οποίο αναπτύχθηκε στα πλαίσια της εργασίας ακολουθήθηκαν οι πρακτικές που χρησιμοποιούνται σήμερα από τις μονάδες υγείας, έτσι ώστε η προσαρμογή του ιατρικού προσωπικού στο νέο σύστημα να είναι όσο το δυνατόν πιο ανώδυνη αλλά και να επιτευχθεί ο στόχος της εξοικονόμησης χρόνου.

Θα αναλύσουμε, επίσης, τους λόγους για τους οποίους χρησιμοποιήθηκαν υπηρεσίες διαδικτύου και ροές εργασίας καθώς και το σκοπό τον οποίο επιτυγχάνουν.

## 1.2 Σκοπός

Σκοπός της εργασίας είναι η υλοποίηση ενός πληροφοριακού συστήματος ηλεκτρονικού φακέλου ασθενών τον οποίο θα καθιστά προσβάσιμο μέσω υπηρεσιών διαδικτύου και μιας ιατρικής εφαρμογής που θα στηρίζει μια ιατρική διαδικασία μέσω ροών εργασίας (workflows). Μέρος της υποδομής του συστήματος στηρίζεται σε υπηρεσίες υπολογιστικού νέφους (cloud computing). Αντικειμενικοί στόχοι της εργασίας είναι:

- Η ανάλυση των τεχνολογιών που χρησιμοποιούνται για ανταλλαγή ηλεκτρονικών μηνυμάτων μεταξύ απομακρυσμένων συστημάτων
- Ορισμός και επεξήγηση των διαλειτουργικών δομών για αποθήκευση και ανταλλαγή ιατρικής πληροφορίας
- Παρουσίαση πλεονεκτημάτων και μειονεκτημάτων από τη χρήση ηλεκτρονικών συναλλαγών στο χώρο της υγείας
- Κατασκευή πληροφοριακού συστήματος ηλεκτρονικής ανταλλαγής ιατρικών μηνυμάτων και ροών εργασίας στο χώρο της υγείας με χρήση τεχνολογιών SOAP, XML, ροών εργασίας

## 1.3 Δομή

Η εργασία αποτελείται από 9 κεφάλαια.

- Στο πρώτο κεφάλαιο θα δούμε μια εισαγωγή των θεμάτων που θα ακολουθήσουν και μια εισαγωγή στο ερευνητικό πρόβλημα
- Στο δεύτερο κεφάλαιο υπάρχει περιγραφή των τεχνολογιών που αφορούν τις δικτυακές υπηρεσίες. Πιο συγκεκριμένα περιγράφουμε τις τεχνολογίες μετάδοσης δεδομένων για απομακρυσμένες συσκευές/υπολογιστές SOAP, WSDL, XML, REST, JSON κ.α.
- Στο τρίτο κεφάλαιο περιγράφεται η πλατφόρμα Windows Azure για υπολογιστικά νέφη, η οποία χρησιμοποιείται από την εφαρμογή που κατασκευάστηκε. Παράλληλα, αναλύονται τα οφέλη των τεχνολογιών υπολογιστικού νέφους γενικότερα

- Στο τέταρτο κεφάλαιο γίνεται περιγραφή του ηλεκτρονικού φάκελου ασθενούς και των συναφών τεχνολογιών πληροφορικής οι οποίες και χρησιμοποιούνται από την εφαρμογή που κατασκευάστηκε στα πλαίσια της εργασίας
- Στο πέμπτο κεφάλαιο περιγράφεται η μεθοδολογία που χρησιμοποιήθηκε κατά την έρευνα (συγκεκριμένα, η έρευνα δράσης) και ο τρόπος χρήσης της
- Στο έκτο κεφάλαιο αναλύεται η εφαρμογή που κατασκευάστηκε στα πλαίσια της διπλωματικής εργασίας. Θα δούμε μια ανάλυση του συστήματος, περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν καθώς και εικόνες (screenshots) από την εφαρμογή.
- Στο έβδομο κεφάλαιο παρουσιάζονται τα συμπεράσματα καθώς και μερικές αναφορές στην απορροφητικότητα των τεχνολογιών ηλεκτρονικού φάκελου ασθενούς στην Ελλάδα
- Στο όγδοο κεφάλαιο αναφέρεται η βιβλιογραφία
- Στο ένατο κεφάλαιο θα βρούμε το παράρτημα

## 2. Υπηρεσίες Διαδικτύου (Web Services)

---

### 2.1 Γενικά

Ξεκινάμε την περιγραφή των υπηρεσιών διαδικτύου δίνοντας έναν μικρό και περιεκτικό ορισμό από το W3C. Εν συνεχεία, θα αναλύσουμε περισσότερο τον όρο.

*«Το W3C προσδιορίζει μια υπηρεσία διαδικτύου (Web Service) σαν ένα σύστημα λογισμικού σχεδιασμένο να υποστηρίζει διαλειτουργική αλληλεπίδραση υπολογιστή με υπολογιστή πάνω από το Διαδίκτυο (Internet).» [22]*

Σύμφωνα με το W3C, μια υπηρεσία έχει συνήθως μια διεπαφή (σημείο στο οποίο είναι ορατή από το διαδίκτυο), προσφέρει πρόσβαση σε λειτουργίες (operations) μιας εφαρμογής λογισμικού (στη βιβλιογραφία αναφέρεται συχνά και ως application server) και η πρόσβαση σε αυτή επιτυγχάνεται μέσω ενός συγκεκριμένου μηχανισμού πρόσβασης. Η διεπαφή μιας υπηρεσίας διαδικτύου υλοποιείται με SOAP/XML ή JSON (τεχνολογίες μετάδοσης δεδομένων), μπορεί περιγράφεται με WSDL (γλώσσα περιγραφής υπηρεσιών διαδικτύου) και η υπηρεσία μπορεί να καταχωρείται και να ανακαλύπτεται με διάφορους τρόπους. Οι παραπάνω τεχνολογίες θα αναλυθούν στη συνέχεια.

Ένας εναλλακτικός ορισμός αναφέρει ότι μια υπηρεσία διαδικτύου είναι ένα κομμάτι επιχειρησιακής λογικής το οποίο βρίσκεται κάπου στο Διαδίκτυο και το οποίο είναι προσπελάσιμο μέσω πρωτοκόλλων διαδικτύου όπως το HTTP. Το W3C επίσης αναφέρει ότι μπορούν να εντοπιστούν δύο είδη υπηρεσιών διαδικτύου, αυτές που ακολουθούν το πρότυπο REST χρησιμοποιώντας το πρωτόκολλο HTTP και μη αποθηκεύοντας κατάσταση (stateless) και αυτές που χρησιμοποιούν το πρωτόκολλο SOAP και έχουν διάφορες λειτουργίες (operations).

### 2.2 Σχεδιαστικές αρχές δικτυακών υπηρεσιών

Μέσα στους στόχους των δικτυακών υπηρεσιών είναι να συνδυάσουν αρμονικά τεχνολογίες από διαφορετικούς κατασκευαστές, πάνω από το διαδίκτυο, με μια κοινή γλώσσα ανταλλαγής μηνυμάτων αλλά και εύκολη επαναχρησιμοποίηση. Έτσι, οι βασικές σχεδιαστικές αρχές των δικτυακών υπηρεσιών είναι:

- Χρήση πρωτοκόλλων που βασίζονται στο δίκτυο (web)
  - o Υπηρεσίες ιστού που στηρίζονται στο HTTP πρωτόκολλο
  - o Τα πρωτόκολλα μπορούν να διαβούν τα τείχη προστασίας (firewalls) και μπορούν να λειτουργούν σε ετερογενές περιβάλλον
- Διαλειτουργικότητα
  - o Το SOAP (ή το JSON) καθορίζει ένα κοινό πρότυπο που επιτρέπει σε διαφορετικά συστήματα να επικοινωνήσουν
- Σχήμα XML (XML Schema) ή JSON
  - o Έγγραφα αναγνωρίσιμα και από άνθρωπο αλλά και από μηχανή
- Συναρμολογησιμότητα
  - o Τα συστατικά υπηρεσιών (βιβλιοθήκες με τις οποίες δημιουργήθηκαν) είναι επαναχρησιμοποιήσιμα
- Διαθεσιμότητα
  - o Οι υπηρεσίες είναι διαθέσιμες στα συστήματα που επιθυμούν να τις χρησιμοποιήσουν
  - o Οι υπηρεσίες πρέπει να εκτεθούν έξω από το ιδιαίτερο σύστημα στο οποίο είναι διαθέσιμες
- Περιγραφή αναγνώσιμη από μηχανή και άνθρωπο
  - o Χρησιμοποιούμενη για να προσδιορίσει τη διεπαφή, τη θέση και τις πληροφορίες πρόσβασης
- Εφαρμογή-ανεξαρτησία
  - o Διαθεσιμότητα, ανεξάρτητη των διεπαφών των διάφορων υπηρεσιών
- Δημοσιοποίηση
  - o Εξερευνησιμες (από άνθρωπο ή μηχανή) αποθήκες περιγραφών υπηρεσιών [2-4]

### 2.3 Μοντέλο λειτουργίας και ρόλοι

Οι υπηρεσίες Παγκόσμιου Ιστού στηρίζονται στο υπηρεσιοστρεφές μοντέλο. Το μοντέλο αυτό υποστηρίζει την παράλληλη συνεργασία και ύπαρξη των παρακάτω

- **Πάροχος υπηρεσίας:** Η υπηρεσία που καλείται να υποστηρίξει τη λειτουργία άλλων υπηρεσιών ή εφαρμογών, εκτελώντας ένα σύνολο από ενέργειες. Με αυτό τον τρόπο, ο πάροχος υπηρεσίας

λαμβάνει αιτήματα προς εξυπηρέτηση από άλλες υπηρεσίες ή εφαρμογές τα οποία και εκτελεί μέσω της λειτουργικότητάς της και εν συνεχεία επιστρέφει το αποτέλεσμα.

- **Πελάτης υπηρεσίας:** Η υπηρεσία που καλεί μια άλλη υπηρεσία προκειμένου να της θέσει κάποιο αίτημα προς εξυπηρέτηση.

- **Μητρώο υπηρεσιών:** Υπάρχουν δύο βασικά ερωτήματα που πρέπει να απαντηθούν σχετικά με τον τρόπο λειτουργίας των δικτυακών υπηρεσιών. Τα ερωτήματα αυτά είναι:

A) πώς ένας πελάτης γνωρίζει την ύπαρξη του παρόχου υπηρεσίας;

B) με ποιον τρόπο τον καλεί ώστε να του αναθέσει μια εργασία;

Η απάντηση σε αυτά τα ερωτήματα δίνεται μέσα από τη λειτουργία του μητρώου υπηρεσιών. Το μητρώο υπηρεσιών είναι ένα ευρετήριο καταχώρησης περιγραφικών υπηρεσιών που αποσκοπεί στη διευκόλυνση του εντοπισμού και στην ενεργοποίηση-κλήση υπηρεσιών. Οι πάροχοι ενημερώνουν το μητρώο υπηρεσιών για την ύπαρξή τους και ζητούν να γίνει μία καταχώρηση στο ευρετήριο. Το μητρώο δημιουργεί μια νέα καταχώρηση με το όνομα και την περιγραφή της υπηρεσίας, η οποία μπορεί να αναζητηθεί από πελάτες.

Οι λειτουργίες που χρησιμοποιούνται ώστε να είναι εφικτή η υλοποίηση του υπηρεσιοστρεφούς μοντέλου είναι οι κάτωθι:

- **Λειτουργίες έκδοσης (publish):** Ο πάροχος υπηρεσίας πρέπει να ενημερώσει τις άλλες υπηρεσίες και εφαρμογές για την ύπαρξή του και τη λειτουργικότητα που παρέχει. Αυτό γίνεται κατά τη λειτουργία της έκδοσης κατά την οποία ο πάροχος περιγράφει και καταχωρεί την υπηρεσία στο μητρώο υπηρεσιών. Μητρώα υπηρεσιών υπάρχουν πολλά και η λογική της καταχώρησης μιας υπηρεσίας στο μητρώο δεν διαφέρει από την καταχώρηση ενός δικτυακού τόπου σε μια μηχανή αναζήτησης.

Η λειτουργία της έκδοσης αποτελείται από δύο στάδια

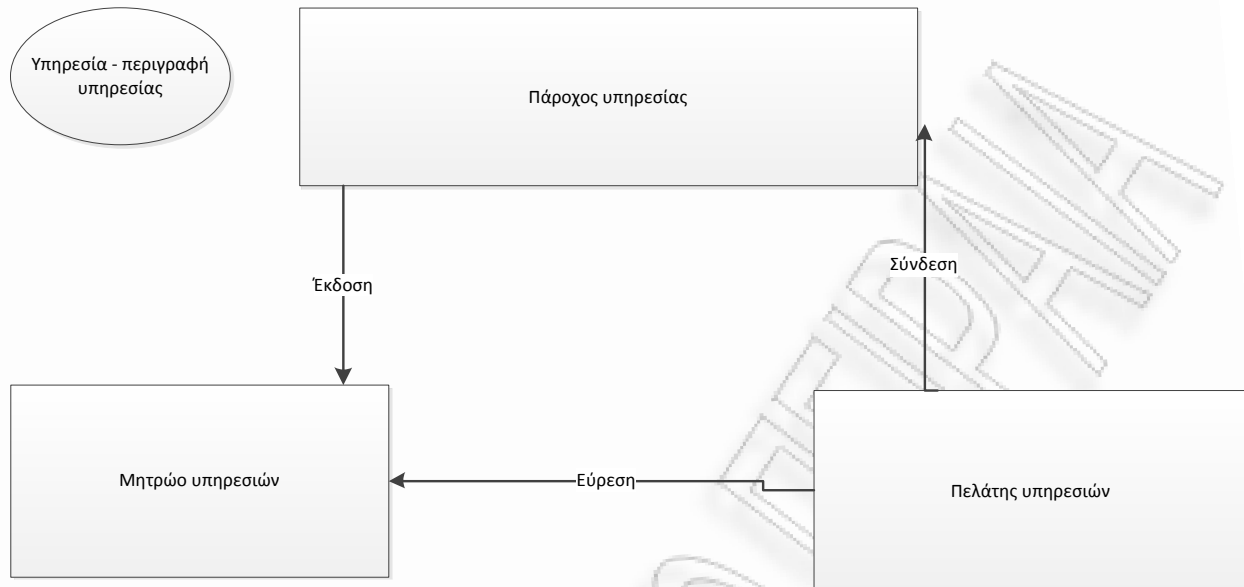
- Στο πρώτο στάδιο η υπηρεσία περιγράφεται χρησιμοποιώντας τη γλώσσα περιγραφής υπηρεσιών παγκοσμίου ιστού (WSDL). Η περιγραφή μιας υπηρεσίας εστιάζει σε τρεις κύριες κατηγορίες πληροφοριών όπως
  - **Πληροφορίες παρόχου:** Οι πληροφορίες που περιγράφουν τον προγραμματιστή της υπηρεσίας καθώς και το τμήμα ή τον οργανισμό που παρέχει αυτή την υπηρεσία.
  - **Πληροφορίες υπηρεσίας:** Πληροφορίες που σχετίζονται με τις βασικές λειτουργίες της υπηρεσίας.
  - **Τεχνικές πληροφορίες:** Αναφέρονται σε λεπτομέρειες υλοποίησης και στον μηχανισμό κλήσης αυτής της υπηρεσίας από μία άλλη
- Στο δεύτερο στάδιο πραγματοποιείται η καταχώρηση της υπηρεσίας στο μητρώο υπηρεσιών, όπου και αποθηκεύονται και τα τρία είδη που αναφέρθηκαν παραπάνω.

- **Λειτουργία αναζήτησης (find):** Η λειτουργία αναζήτησης σχετίζεται με τις ενέργειες που πρέπει να γίνουν από έναν πελάτη και το μητρώο υπηρεσίας έτσι ώστε να αναζητηθεί και να εντοπιστεί ένα ή περισσότερα web services. Αρχικά, ένας πελάτης στέλνει ένα αίτημα στο μητρώο υπηρεσιών για να ανακαλύψει μια υπηρεσία και ακολούθως επιλέγει αυτή που επιθυμεί, σύμφωνα με τον όρο αναζήτησης που έθεσε.

Η λειτουργία της αναζήτησης αποτελείται κι αυτή από δύο στάδια

- **Στάδιο ανακάλυψης:** Σε αυτό το στάδιο μια υπηρεσία θέτει ένα ερώτημα στο μητρώο υπηρεσιών προκειμένου να βρει μια υπηρεσία. Το ερώτημα αυτό περιλαμβάνει όρους αναζήτησης όπως τεχνικά χαρακτηριστικά, τύπο της υπηρεσίας κ.α. Το στάδιο αυτό μπορεί να είναι
  - ο **Στατικό:** σε αυτή την περίπτωση σκοπός είναι ο εντοπισμός της περιγραφής της διεπαφής μιας υπηρεσίας προκειμένου να χρησιμοποιηθεί.
  - ο **Δυναμικό:** αφορά την αναζήτηση που γίνεται κατά την εκτέλεση ενός προγράμματος προκειμένου να αναζητηθεί ο τρόπος κλήσεως μιας υπηρεσίας
- **Στάδιο επιλογής:** Σε αυτό το στάδιο επιλέγονται μία ή περισσότερες υπηρεσίες. Η επιλογή γίνεται είτε χειροκίνητα (επιθεώρηση της λίστας αποτελεσμάτων και επιλογή) είτε αυτόματα (χρήση της πιο συμφέρουσας υπηρεσίας). Οι όροι αναζήτησης μπορεί να περιλαμβάνουν διάφορα κριτήρια, όπως αν η υπηρεσία επιστρέφει σχετικά άμεσα, την ασφάλεια της υπηρεσίας κ.α.

-**Λειτουργία σύνδεσης και εκτέλεσης:** Η λειτουργία αυτή είναι η πιο σημαντική από όλες καθώς μέσω αυτής η υπηρεσία πελάτης συνδέεται με τον πάροχο υπηρεσίας και του αναθέτει μία εργασία προς εκτέλεση. Κατά τη λειτουργία αυτή, χρησιμοποιούνται οι τεχνικές πληροφορίες της υπηρεσίας του πάροχου όπως αυτές έχουν καταχωρηθεί στο μητρώο υπηρεσιών. Η έμμεση σύνδεση και εκτέλεση πραγματοποιείται μέσω ενός διαμεσολαβητή υπηρεσιών ο οποίος αναλαμβάνει να εξυπηρετήσει τον πελάτη. Έτσι, ο πελάτης επικοινωνεί και στέλνει ένα αίτημα στο διαμεσολαβητή. Εκείνος, έρχεται σε επαφή με το μητρώο υπηρεσιών ώστε να εντοπίσει έναν πάροχο, ο οποίος και καλείται προκειμένου να επιστρέψει το αποτέλεσμα στον πελάτη.



ΣΧΗΜΑ 1 ΛΕΙΤΟΥΡΓΙΕΣ ΚΑΙ ΡΟΛΟΙ ΥΠΗΡΕΣΙΩΝ [3]

## 2.3 Πλεονεκτήματα

Εξαιτίας των σχεδιαστικών αρχών με βάση τις οποίες δημιουργήθηκαν οι υπηρεσίες διαδικτύου, είναι προφανές πως έχουν αρκετά πλεονεκτήματα. Τα πλεονεκτήματα αυτά είναι κυρίως η διαλειτουργικότητα, η εύκολη δημοσίευση και η επεκτασιμότητα. Αναλυτικά:

- **Διαλειτουργικότητα:** Μια υπηρεσία διαδικτύου παρέχει ανεξαρτησία τόσο από λειτουργικό σύστημα όσο και από το υλικό (hardware). Οποιοδήποτε πρόγραμμα που συμβαδίζει με αυτή την τεχνολογία μπορεί πολύ εύκολα να προσπελάσει μια τέτοια υπηρεσία.
- **Ενσωμάτωση:** Σε ένα υπάρχον πληροφοριακό σύστημα που λειτουργεί μέσα στο Διαδίκτυο η δημιουργία μιας υπηρεσίας ιστού δεν απαιτεί μεγάλες αλλαγές στον μηχανισμό του συστήματος
- **Διαθεσιμότητα και δημοσίευση:** Οι πληροφορίες για τις υπηρεσίες ιστού δημοσιεύονται οπότε η εύρεση και η χρήση τους μπορεί να είναι ταχύτατες. Πρόσθετη βοήθεια για τη χρησιμοποίηση μιας υπηρεσίας ιστού παρέχεται και από τα εργαλεία ανάπτυξης λογισμικού (Integrated Development Environments)



- **Επεκτασιμότητα:** Μια έτοιμη υπηρεσία ιστού είναι δυνατό να ανανεωθεί με εύκολο τρόπο παρέχοντας έτσι επιπρόσθετες υπηρεσίες στους χρήστες του. Παράλληλα, μπορούν να δημοσιεύονται καινούριες εκδόσεις μιας υπηρεσίας χωρίς «σπάνε» (break) λογισμικά που έχουν πρόσβαση στις προηγούμενες εκδόσεις
- **Μικρό κόστος δημιουργίας και χρήσης:** Εφόσον σε ένα λογισμικό υπάρχει ήδη κάποια διαδικασία που χρειάζεται να επεκταθεί σε online υπηρεσία, η δημιουργία μιας υπηρεσίας ιστού γίνεται πολύ γρήγορα και εύκολα στις περισσότερες των περιπτώσεων. Επίσης, το κόστος ενσωμάτωσης μιας υπηρεσίας ιστού σε κάποια ιστοσελίδα ή σε δικτυακή εφαρμογή είναι πάρα πολύ μικρό και γίνεται πολύ εύκολα, με τη χρήση κατάλληλων προγραμματιστικών εργαλείων. Ακόμα και στις περιπτώσεις που η χρήση κάποιας υπηρεσίας ιστού γίνεται με ενοικίαση (subscription based model) το συνολικό κόστος της χρήσης είναι αρκετά πιο μικρό από το κόστος δημιουργίας της υπηρεσίας αυτής.
- **Χρήση λογισμικού:** Όλο το λογισμικό και ειδικότερα οι ιστοσελίδες που χρησιμοποιούν υπηρεσίες γίνονται πιο λειτουργικές και πιο φιλικές αφού παρέχουν περισσότερες υπηρεσίες στους χρήστες
- **XML/JSON:** Χρησιμοποιείται XML/JSON στο επίπεδο αναπαράστασης δεδομένων για όλα τα πρωτόκολλα υπηρεσιών ιστού και των τεχνολογιών που δημιουργούνται, έτσι οι τεχνολογίες μπορούν να διαλειτουργούν στο βασικό τους επίπεδο. Σαν επίπεδο μεταφοράς η XML ή η JSON αποβάλλει κάθε δικτύωση, λειτουργικό σύστημα ή δέσμευση πλατφόρμας που έχει πρωτόκολλο (το λεγόμενο vendor lock-in)
- **Loosely coupled:** Ένας καταναλωτής μιας υπηρεσίας ιστού δεν είναι δεσμευμένος με την συγκεκριμένη υπηρεσία ιστού άμεσα, η διεπαφή της υπηρεσίας ιστού μπορεί να αλλάξει με την πάροδο του χρόνου χωρίς να επηρεάσει την δυνατότητα του πελάτη να αλληλεπιδράσει με την υπηρεσία. Ένα “tightly-coupled” σύστημα υπονοεί ότι οι λογικές πελατών και κεντρικών υπολογιστών είναι στενά συνδεδεμένες μεταξύ τους, που σημαίνει ότι εάν μια διεπαφή αλλάξει, η άλλη θα πρέπει επίσης να ενημερώνεται. Η υιοθέτηση μιας “loosely coupled” αρχιτεκτονικής τείνει να κάνει τα συστήματα λογισμικού πιο εύχρηστα και επιτρέπει την απλούστερη ενοποίηση διαφορετικών συστημάτων.
- **Coarse-grained:** Οι αντικειμενοστρεφείς τεχνολογίες όπως η πλατφόρμα .NET εκθέτουν τις υπηρεσίες τους μέσω ξεχωριστών μεθόδων. Μια μεμονωμένη μέθοδος είναι μια πάρα πολύ

λεπτή λειτουργία που παρέχει κάθε χρήσιμη ικανότητα σε ένα συνεταιρικό επίπεδο. Η ανάπτυξη ενός προγράμματος .NET από την αρχή απαιτεί τη δημιουργία πολλών “fine-grained” μεθόδων οι οποίες εισχωρούν έπειτα σε μια “coarse-grained” υπηρεσία που καταναλώνεται είτε από έναν πελάτη είτε από μια άλλη υπηρεσία. Οι επιχειρήσεις και οι διεπαφές που εκθέτουν πρέπει να είναι “coarse-grained”. Η τεχνολογία υπηρεσιών ιστού παρέχει έναν φυσικό τρόπο ορισμού “coarse-grained” υπηρεσιών που έχουν πρόσβαση στο σωστό πόρο από όλη την επιχειρησιακή λογική.

- **Δυνατότητα να είναι σύγχρονες και ασύγχρονες:** Ο συγχρονισμός αναφέρεται στη σύνδεση του πελάτη με την εκτέλεση της υπηρεσίας. Στις σύγχρονες επικλήσεις, ο πελάτης εμποδίζει και περιμένει την υπηρεσία να ολοκληρώσει την λειτουργία της πριν συνεχίσει. Οι ασύγχρονες λειτουργίες επιτρέπει στον πελάτη να καλέσει μια υπηρεσία και έπειτα να εκτελέσει άλλες λειτουργίες (για παράδειγμα, μια κλήση υπηρεσίας διαδικτύου από κινητό πρέπει να είναι ασύγχρονη, καθώς όσο το κινητό περιμένει για δεδομένα πρέπει να μπορεί να ανταποκρίνεται στις εντολές του χρήστη). Οι ασύγχρονοι πελάτες ανακτούν το αποτέλεσμα τους αργότερα, ενώ οι σύγχρονοι πελάτες λαμβάνουν το αποτέλεσμα τους όταν η υπηρεσία έχει ολοκληρωθεί.
- **Υποστήριξη κλήσεων εξ'αποστάσεως διαδικασίας (Remote Procedure Calls, RPCs):** Οι υπηρεσίες ιστού επιτρέπουν στους πελάτες να καλέσουν τις διαδικασίες, τις λειτουργίες και τις μεθόδους σε απομακρυσμένα αντικείμενα χρησιμοποιώντας ένα πρωτόκολλο επικοινωνίας βασισμένο στη γλώσσα XML. Οι απομακρυσμένες διαδικασίες εκθέτουν τις παραμέτρους εισαγωγής και εξαγωγής που μια υπηρεσία ιστού πρέπει να υποστηρίξει. Μια υπηρεσία ιστού υποστηρίζει RPC με την παροχή υπηρεσιών της, ισοδύναμων με εκείνες ενός παραδοσιακού συστατικού, ή με τη μετάφραση των εισερχομένων κλήσεων σε μια κλήση ενός .NET συστατικού.
- **Υποστήριξη ανταλλαγής μηνυμάτων:** Ένα από τα βασικά πλεονεκτήματα της XML και της JSON είναι ο γενικός τρόπος που αναπαριστά όχι μόνο τα στοιχεία αλλά και σύνθετα έγγραφα. Αυτά τα έγγραφα μπορούν να είναι απλά, όπως όταν αντιπροσωπεύουν μια τρέχουσα διεύθυνση, ή μπορούν να είναι σύνθετα, αντιπροσωπεύοντας ένα ολόκληρο αντικείμενο (object). Οι υπηρεσίες ιστού υποστηρίζουν τη διαφανή ανταλλαγή των εγγράφων για να διευκολύνουν την επιχειρησιακή ενοποίηση. [3, 20-22]

## 2.4 Οι βασικές τεχνολογίες που συνθέτουν τις υπηρεσίες διαδικτύου

Αρκετές τεχνολογίες έχουν παρουσιαστεί μέσω του όρου «υπηρεσίες διαδικτύου» και πολύ περισσότερες θα παρουσιαστούν στα επόμενα έτη. Στην πραγματικότητα, το φαινόμενο υπηρεσιών διαδικτύου έχει αυξηθεί τόσο γρήγορα που διάφορες ανταγωνιστικές τεχνολογίες προσπαθούν να παρέχουν τις ίδιες δυνατότητες. Εντούτοις, το όραμα των υπηρεσιών διαδικτύου για την συνεχόμενη παγκόσμια επιχειρησιακή ενοποίηση δεν πρόκειται να είναι εφικτό εκτός αν οι βασικές τεχνολογίες υποστηρίζονται από κάθε σημαντική επιχείρηση λογισμικού στον κόσμο. Κατά τη διάρκεια των προηγούμενων ετών, τέσσερις βασικές τεχνολογίες έχουν προκύψει ως παγκόσμια πρότυπα που αποτελούν τον πυρήνα της τεχνολογίας υπηρεσιών δικτύου.

### 2.4.1 Επεκτάσιμη Γλώσσα Σήμανσης (eXtensible Markup Language, XML)

Όλα τα έγγραφα υπηρεσιών διαδικτύου είναι γραμμένα σε γλώσσα XML και το XML σχήμα χρησιμοποιείται για να καθορίσει τα στοιχεία που χρησιμοποιούνται στην επικοινωνία υπηρεσιών ιστού.

Η ονομασία XML προκύπτει από τα αρχικά των λέξεων eXtensible Markup Language. Είναι μια γλώσσα σήμανσης που ξεκίνησε να αναπτύσσεται από το W3C (World Wide Web Consortium) κυρίως για να ξεπεράσει τους περιορισμούς της HTML.

Η HTML είναι μια πάρα πολύ δημοφιλής γλώσσα σήμανσης. Σύμφωνα με κάποιες μελέτες υπάρχουν πάνω από 1 δισεκατομμύριο σελίδες στο δίκτυο και όλες βασίζονται στην HTML. Υποστηρίζεται από χιλιάδες εφαρμογές συμπεριλαμβανομένων των πλοηγών διαδικτύου, των επεξεργαστών κειμένου, του λογισμικού για ηλεκτρονικό ταχυδρομείο, των βάσεων δεδομένων και άλλων.

Αρχικά το διαδίκτυο ήταν η λύση για την δημοσίευση επιστημονικών κειμένων. Σήμερα έχει γίνει ένα μέσο που χρησιμοποιείται ως επί το πλείστον από τους χρήστες υπολογιστών, tablets αλλά και κινητών συσκευών. Σπουδαιότερα το διαδίκτυο είναι ένα αλληλεπιδρόμενο μέσο επειδή υποστηρίζει εφαρμογές όπως τα online καταστήματα, τις ηλεκτρονικές τραπεζικές συναλλαγές, το ηλεκτρονικό εμπόριο και την διακίνηση ιδεών. Για να εξυπηρετήσει και να ανταπεξέλθει σε αυτή την πρωτοφανή δημοσιότητα, η HTML επεκτάθηκε με τα χρόνια. Πολλά νέα HTML tags (ετικέτες, κώδικας που δηλώνει στον φυλλομετρητή (browser) την εμφάνιση κάποιου αντικειμένου) έχουν παρουσιαστεί.

Συμπερασματικά, αν και η HTML είναι μια δημοφιλής και επιτυχημένη γλώσσα σήμανσης, έχει ορισμένα βασικά μειονεκτήματα. Η XML αναπτύχθηκε για να αντιμετωπίσει αυτά τα μειονεκτήματα και όχι για νεωτερισμό. Η XML υπάρχει λόγω της επιτυχίας της HTML. Για αυτό η XML ενσωματώνει ορισμένα επιτυχημένα χαρακτηριστικά της HTML. [20-22]

#### **2.4.1.1 Ανάλυση της XML**

Αν συνοψίζαμε την XML σε μια πρόταση, θα λέγαμε ότι είναι «μία ομάδα προτύπων» για την ανταλλαγή και δημοσίευση πληροφοριών με ένα δομημένο τρόπο». Η έμφαση στη δομή δεν μπορεί να υποτιμηθεί.

Η XML είναι μια γλώσσα που χρησιμοποιείται στην περιγραφή και τον χειρισμό δομημένων εγγράφων. Τα XML έγγραφα δεν περιορίζονται σε βιβλία και άρθρα ή ακόμη και ιστοσελίδες και μπορεί να περιλαμβάνουν αντικείμενα σε μια εφαρμογή πελάτη/εξυπηρετητή.

Πάντως η XML προσφέρει την ίδια δενδροειδή δομή σε όλες τις παραπάνω εφαρμογές. Η XML δεν υποδεικνύει ή επιβάλλει τις λεπτομέρειες αυτής της δομής – ούτε καν υποδεικνύει το γέμισμα του δέντρου.

Η XML είναι ένας ευέλικτος μηχανισμός που παρέχει την δομή συγκεκριμένων εφαρμογών. Αυτή παρέχει ένα μηχανισμό κωδικοποίησης των χειριζόμενων από την εφαρμογή πληροφοριών αλλά και της υποκείμενης δομής.

Η XML προσφέρει επίσης αρκετούς μηχανισμούς για τον χειρισμό των πληροφοριών, όπως την εμφάνισή τους,, την ανάκτησή τους από την εφαρμογή και άλλους.

#### **2.4.1.2 XML σχήμα (schema)**

Ένα σχήμα είναι ο καθορισμός της έννοιας/μορφοποίησης/τύπου δεδομένων κάθε μιας από τις ετικέτες (tags) μέσα σε ένα έγγραφο XML. Για παράδειγμα, το παρακάτω σχήμα δηλώνει ότι το XML έγγραφο προς έγκριση (validation) θα πρέπει να έχει ένα στοιχείο ονόματι “note”, και μέσα σε αυτά μια αλληλουχία από άλλα στοιχεία ονόματι “to”, “from”, “heading”, “body” όλα τύπου string (συμβολοσειράς). Ένα XML έγγραφο μπορεί να περιέχει μια αναφορά σε ένα XML σχήμα, έτσι ώστε να μπορεί εύκολα να εγκριθεί (validate) από κάποιο λογισμικό XML validator.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

ΣΧΗΜΑ 2 XML ΣΧΗΜΑ

## 2.4.2 Απλό πρωτόκολλο πρόσβασης αντικειμένου (Simple Object Access Protocol, SOAP)

### 2.4.2.1 Γενικά

Ως SOAP ορίζουμε το απλό πρωτόκολλο πρόσβασης αντικειμένου (Simple Object Access Protocol). Είναι μία από τις πιο σύγχρονες τάσεις στον προγραμματισμό διαδικτυακών εφαρμογών και είναι αναπόσπαστο κομμάτι των υπηρεσιών διαδικτύου που έχουν καταλάβει την τελευταία γενιά του web programming.

Το SOAP είναι ένα πρωτόκολλο, απλό στη χρήση (αλλά όχι στην ανάπτυξη) το οποίο βασίζεται στην ιδέα του ότι σε μια κατανεμημένη αρχιτεκτονική είναι απαραίτητη η ανταλλαγή πληροφοριών. Επιπλέον, σε ένα σύστημα που πιθανώς είναι υπερφορτωμένο και αποτελείται από διαδικασίες το SOAP πρωτόκολλο είναι αρκετά ελαφρύ. Τέλος, επιτρέπει την ανταλλαγή μηνυμάτων μέσω του HTTP πρωτοκόλλου, επιτρέποντας κατά συνέπεια στον χρήστη να αποφεύγει όλα τα δύσκολα θέματα όπως τα τείχη προστασίας (firewalls).

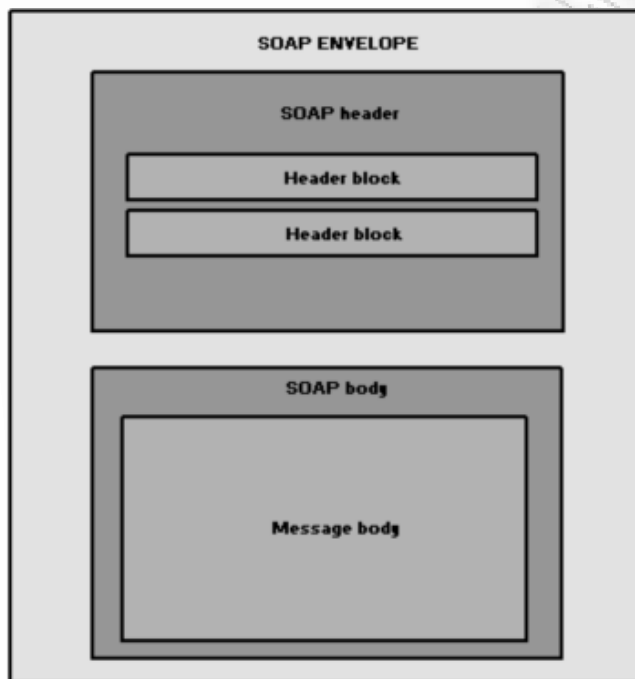
Τρία βασικά στοιχεία αποτελούν την προδιαγραφή του SOAP: ο φάκελος SOAP, ένα σύνολο κανόνων κωδικοποίησης και μια έννοια αλληλεπίδρασης μεταξύ αιτήρησης και απάντησης.

Στην πραγματικότητα το SOAP χρησιμοποιείται για να επικοινωνήσει ο πελάτης με την υπηρεσία ιστού. Τόσο το αίτημα όσο και η απάντηση είναι SOAP μηνύματα. Το σώμα του μηνύματος (του οποίου η γραμματική καθορίζεται από το WSDL) περιλαμβάνεται μέσα στον SOAP φάκελο “envelop”.

Το SOAP παρέχει έναν καθορισμένο φάκελο “envelope” μέσα στον οποίο ένα μήνυμα μπορεί να μεταφερθεί. Το SOAP είναι ο μηχανισμός για την μεταφορά μηνυμάτων μεταξύ απομακρυσμένων εφαρμογών και δεν δίνει καμία πληροφορία για το περιεχόμενο του φακέλου. Ο αποστολέα και ο παραλήπτης θα πρέπει να κατανοήσουν το μήνυμα.

Μπορούμε στο παρακάτω σχήμα να δούμε ότι ένα SOAP μήνυμα αποτελείται από

- Έναν φάκελο “envelop”
- Μια επικεφαλίδα “header”
- Το σώμα “body”
- Το σώμα μπορεί να περιέχει ένα “fault” στοιχείο



ΣΧΗΜΑ 3 SOAP ΦΑΚΕΛΟΣ

Ο φάκελος περιβάλλει ολόκληρο το SOAP μήνυμα, η επικεφαλίδα περιέχει επιπλέον πληροφορία που θα μεταφερθεί όπως το σώμα του εγγράφου, το σώμα περιέχει τον πυρήνα του SOAP εγγράφου είτε το ίδιο το XML μήνυμα. Τέλος, το Fault περιέχει πληροφορίες για σφάλματα-εξαιρέσεις (exceptions).

#### 2.4.2.2 Φάκελος (Envelope)

Ο SOAP φάκελος είναι ανάλογος με τον φάκελο ενός πραγματικού, παραδοσιακού γράμματος. Περιέχει πληροφορία για το μήνυμα το οποίο κωδικοποιείται σε ένα SOAP φορτίο συμπεριλαμβανομένων δεδομένων που αφορούν στον παραλήπτη και στον αποστολέα καθώς επίσης και λεπτομέρειες για το ίδιο το μήνυμα. Για παράδειγμα, η επικεφαλίδα του SOAP φακέλου μπορεί να καθορίσει τον τρόπο με τον οποίο το μήνυμα πρέπει να επεξεργαστεί. Πριν μια εφαρμογή προχωρήσει με την επεξεργασία ενός μηνύματος, μπορεί να προσδιορίσει πληροφορία για το ίδιο το μήνυμα, συμπεριλαμβανομένου ακόμα και του αν μπορεί να διαχειριστεί το μήνυμα.

Εξαιρουμένης της περίπτωσης του πρωτοκόλλου XML-RPC (Remote Procedure Call) κλήσεων, με το SOAP προκύπτει πραγματική ερμηνεία με σκοπό να προσδιοριστεί κάτι για το μήνυμα. Ένα τυπικό SOAP μήνυμα μπορεί επίσης να περιλαμβάνει τον τρόπο κωδικοποίησης.

Στο παρακάτω σχήμα φαίνεται ο SOAP φάκελος με συγκεκριμένη κωδικοποίηση. Στο συγκεκριμένο παράδειγμα, δηλώνεται η διαδικασία λήψης τιμής μιας μετοχής από μια υπηρεσία διαδικτύου

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
</soap:Header>
  <soap:Body>
    <m:GetStockPrice xmlns:m="http://www.example.org/stock">
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

ΣΧΗΜΑ 4 SOAP XML ΜΗΝΥΜΑ

Από το σχήμα φαίνεται πως η κωδικοποίηση καθορίζεται μέσα στον φάκελο επιτρέποντας στην εφαρμογή να καθορίσει χρησιμοποιώντας την τιμή του γνωρίσματος *encodingStyle*, το εάν μπορεί να διαβάσει το εισερχόμενο μήνυμα το οποίο βρίσκεται μέσα στο σώμα (body).

### 2.4.2.3 Κωδικοποίηση

Το δεύτερο βασικό στοιχείο που το SOAP φέρνει προς συζήτηση είναι μια απλή έννοια κωδικοποίησης τύπων δεδομένων που καθορίζονται από τον χρήστη. Με το SOAP τα XML σχήματα μπορούν να χρησιμοποιηθούν εύκολα για να καθορίσουν νέες δομές δεδομένων και αυτές οι δομές μπορούν εύκολα να αναπαρασταθούν με την XML σαν κομμάτι του SOAP. Εξαιτίας αυτής της ενοποίησης με το XML σχήμα, μπορεί να κωδικοποιηθεί κάθε τύπος δεδομένων σε ένα SOAP μήνυμα που περιγράφεται λογικά με την XML.

### 2.4.2.4 Επίκληση (Invocation)

Στο παρακάτω τμήμα κώδικα θα δούμε πως μπορούμε να καλέσουμε μια υπηρεσία με χρήση της τεχνολογίας Windows Communication Foundation της Microsoft και με την προγραμματιστική γλώσσα C#. Στο συγκεκριμένο παράδειγμα, υποθέτουμε ότι θέλουμε να καλέσουμε μια SOAP υπηρεσία η οποία επιτελεί μια απλή πρόσθεση, οπότε και δέχεται σαν παραμέτρους 2 δεκαδικούς και μας επιστρέφει έναν τρίτο, ο οποίος είναι και το άθροισμά τους.

Ας αναλύσουμε μερικά σημεία του κώδικα, ώστε να γίνει κατανοητός ο τρόπος της επίκλησης

- Το interface *IMath* και η κλάση *Math* περιγράφουν την υπηρεσία. Πιο συγκεκριμένα, το *IMath* είναι αυτό που φαίνεται (exposed) σαν υπηρεσία διαδικτύου. Το *IMath* περιέχει μία λειτουργία (operation) η οποία ονομάζεται *Add*. Αυτή η λειτουργία υλοποιείται από την κλάση *Math* (η οποία υλοποιεί – implements – το interface *IMath*) και δεν είναι κάτι άλλο από μία απλή συνάρτηση (function) της C#, η οποία δέχεται δύο δεκαδικούς και επιστρέφει το άθροισμά τους. Η κλάση *Math*
- Το interface *IMath* έχει δύο πολύ συγκεκριμένες ιδιότητες (attributes). Η ιδιότητα *ServiceContract* δηλώνει στη μηχανή εκτέλεσης (runtime) ότι περιγράφει μία υπηρεσία διαδικτύου και η ιδιότητα *OperationContract* δηλώνει ότι περιγράφει μία λειτουργία (operation) μιας υπηρεσίας διαδικτύου. Οπότε, κατά την εκτέλεση του προγράμματος, εκτίθεται (exposed) μια υπηρεσία με μία λειτουργία, μαζί με τις περιγραφές τους. Η



υλοποίηση τους δεν ενδιαφέρει (και δεν πρέπει να ενδιαφέρει) τον όποιον καλέσει τη συγκεκριμένη υπηρεσία. Παρόλα αυτά, εφόσον η κλάση Math υλοποιεί το interface IMath, τότε και η κλάση Math περιέχει την υλοποίηση της υπηρεσίας διαδικτύου.

- Η κλήση της υπηρεσίας αποτελείται από 3 στάδια
  - Στο πρώτο στάδιο, δηλώνουμε τον τρόπο που θα επικοινωνήσουμε με την υπηρεσία αλλά και τη διεύθυνση δικτύου στην οποία βρίσκεται. Επιλέγουμε τον τρόπο basicHttpBinding, ο οποίος είναι ο πιο απλός και διαλειτουργικός τρόπος επικοινωνίας μεταξύ υπηρεσιών δικτύου – συμβατός με το βασικό προφίλ του W3C (Basic Profile 1.1) - και εισάγουμε τη διεύθυνση της υπηρεσίας μέσω της κλάσης EndPointAddress.
  - Εν συνεχεία, στο δεύτερο στάδιο δημιουργούμε ένα κανάλι επικοινωνίας (μέσω της κλάσης ChannelFactory), όπου δίνουμε σαν παράμετρο και την περιγραφή της υπηρεσίας (μέσω του interface IMath) μαζί τον τρόπο επικοινωνίας και τη διεύθυνση της υπηρεσίας
  - Στο τρίτο στάδιο έχουμε δύο παραδείγματα κλήσης της υπηρεσίας
    - Ανοίγοντας το κανάλι επικοινωνίας
    - Αποστολή της αίτησης – αναμονή για τα αποτελέσματα (εκτός κι αν η κλήση γίνει σε διαφορετικό νήμα)
    - Επιστροφή της κλήσης – εμφάνιση των αποτελεσμάτων

```
using System;
using System.ServiceModel;

// This code generated by svcutil.exe.
[ServiceContract()]
interface IMath
{
    [OperationContract()]
    double Add(double A, double B);
}
```

```
public class Math : IMath
{
    public double Add(double A, double B)
    {
        return A + B;
    }
}

public sealed class Test
{
    static void Main()
    {
        // Code not shown.
    }

    public void Run()
    {
        // This code is written by an εφαρμογή developer.
        // Create a channel factory.
        BasicHttpBinding myBinding = new BasicHttpBinding();

        EndpointAddress myEndpoint = new EndpointAddress("http://localhost/MathService/Ep1");

        ChannelFactory<IMath> myChannelFactory = new ChannelFactory<IMath>(myBinding,
myEndpoint);

        // Create a channel.
        IMath wcfClient1 = myChannelFactory.CreateChannel();
        double s = wcfClient1.Add(3, 39);
        Console.WriteLine(s.ToString());
        ((IClientChannel)wcfClient1).Close();

        // Create another channel.
    }
}
```

```
IMath wcfClient2 = myChannelFactory.CreateChannel();  
s = wcfClient2.Add(15, 27);  
Console.WriteLine(s.ToString());  
((IClientChannel)wcfClient2).Close();  
myChannelFactory.Close();  
}  
}
```

ΣΧΗΜΑ 5 C# ΚΩΔΙΚΑΣ ΓΙΑ ΥΠΗΡΕΣΙΑ ΔΙΑΔΙΚΤΥΟΥ Η ΟΠΟΙΑ ΠΡΟΣΘΕΤΕΙ ΔΥΟ ΔΕΚΑΔΙΚΟΥΣ ΑΡΙΘΜΟΥΣ

### 2.4.3 SOAP-RPC ή SOAP ανταλλαγή μηνυμάτων (SOAP-messaging)

Πριν ξεκινήσουμε την ανάπτυξη ενός συστήματος που βασίζεται στο SOAP θα πρέπει να επιλέξουμε ανάμεσα στην μεθοδολογία SOAP-RPC και την SOAP-messaging.

Η πρώτη σκέψη πριν αποφασίσουμε ποια από τις δύο μεθοδολογίες θα χρησιμοποιήσουμε θα πρέπει να σχετίζεται με τον αρχιτεκτονικό σχεδιασμό και όχι με την υλοποίηση (coding). Θα πρέπει να προσδιορίσουμε εάν θέλουμε μία RPC ή μια SOAP messaging υπηρεσία. Η RPC μεθοδολογία είναι πολύ γνωστή, ένας πελάτης επικαλείται μια εξ'αποστάσεως διαδικασία που βρίσκεται σε κάποιον εξυπηρετητή και λαμβάνει ως απάντηση. Σε αυτό το σενάριο το SOAP απλώς ενεργεί σαν ένα πιο εξελιγμένο XML-RPC σύστημα που επιτρέπει καλύτερα χειρισμό λαθών και καλύτερη διαβίβαση πολύπλοκων τύπων (complex types) μέσα στο δίκτυο.

Θα πρέπει να κοιτάξουμε την εφαρμογή μας και θα πρέπει να προσδιορίσουμε τι ακριβώς ζητάμε από το SOAP. Εάν έχουμε έναν εξυπηρετητή και ένα σύνολο από πελάτες που χρειάζεται να διενεργούν άλλες κλήσεις από απόσταση τότε το RPC είναι η καλύτερη επιλογή. Εάν από την άλλη πρόκειται για μεγαλύτερα συστήματα τα οποία ανταλλάσσουν δεδομένα περισσότερο και λιγότερο διενεργούν συγκεκριμένες επιχειρησιακές λειτουργίες κατά απαίτηση, τότε το SOAP messaging είναι προτιμότερο. [21,22]

### 2.4.4 Γλώσσα περιγραφής Υπηρεσιών διαδικτύου (Web Service Description Language, WSDL)

Η WSDL (Web Service Description Language) είναι μια XML τεχνολογία που περιγράφει την διεπαφή μιας υπηρεσίας δικτύου με έναν τυποποιημένο τρόπο. Η WSDL συγκεκριμενοποιεί τον τρόπο με τον οποίο μια υπηρεσία διαδικτύου αναπαριστά τις παραμέτρους εισόδου και εξόδου μιας επίκλησης εξωτερικά, την δομή της λειτουργίας και την φύση της επίκλησης. Η WSDL επιτρέπει σε διάσπαρτους πελάτες να καταλάβουν αυτόματα τον τρόπο με τον οποίο θα αλληλεπιδράσουν με μια υπηρεσία διαδικτύου.

Το WSDL αρχείο θα περιγράφει πολλά κρίσιμα κομμάτια πληροφορίας που ο πελάτης μιας υπηρεσίας θα χρειαστεί:

- Το όνομα της υπηρεσίας, συμπεριλαμβάνοντας το URN της.
- Τη διεύθυνση στην οποία μπορούμε να βρούμε την υπηρεσίας (συνήθως πρόκειται για μια HTTP URL διεύθυνση)
- Οι διαθέσιμες μέθοδοι επίκλησης
- Οι παράμετροι εισόδου και εξόδου για κάθε μέθοδο

Κάθε ένα από τα κομμάτια δεδομένων είναι άχρηστο εάν είναι μόνο του, σε περίπτωση όμως που τα κομμάτια ενωθούν αντικατοπτρίζουν την ολοκληρωμένη εικόνα του πελάτη της υπηρεσίας. Επιπλέον, ένα WSDL κείμενο περικλείει στοιχεία του XML σχήματος, παραμέτρους του XML-RPC μοντέλου και λίγο από ό,τι άλλο έχουμε αναφέρει ως τώρα. [1]

Παρακάτω βλέπουμε ένα απόσπασμα από τον κώδικα WSDL (τρία έγγραφα, το κάθε ένα ξεκινάει από την επικεφαλίδα “<?xml”) που δημιουργείται για την υπηρεσία Math που περιγράψαμε παραπάνω. Αναλυτικά

- Στο στοιχείο definitions βλέπουμε τους χώρους ονομάτων (namespaces) που τυπικά προσδιορίζουν ένα WSDL έγγραφο
- Στο στοιχείο locations βλέπουμε μια αναφορά σε ένα ακόμα WSDL έγγραφο (<http://localhost:65182/MathService.svc?wsdl=wsdl0>) που περιγράφει περαιτέρω την υπηρεσία διαδικτύου
- Στο δεύτερο έγγραφο βλέπουμε ότι το όνομα της υπηρεσίας είναι MathService και το όνομα της λειτουργίας είναι Add

- Στο τρίτο έγγραφο βλέπουμε αναφορά και για τις δύο παραμέτρους της υπηρεσίας Add, την a και την b, που αντιστοιχούν στον τύπο δεδομένων double (δεκαδικός)

Σε όλα τα σύγχρονα περιβάλλοντα προγραμματισμού (Integrated Development Environment) υπάρχουν εργαλεία που διαβάζουν τον κώδικα WSDL και δημιουργούν τον απαραίτητο κώδικα σε υψηλού επιπέδου γλώσσες προγραμματισμού (C#, Java) για την κλήση μιας υπηρεσίας. Έτσι, ο προγραμματιστής λογισμικού δεν χρειάζεται να μπαίνει σε λεπτομέρειες όπως του πώς να μετατρέψει μία κλάση C# σε ένα XML SOAP έγγραφο προκειμένου να το στείλει στην υπηρεσία διαδικτύου.

```
<?xml version="1.0" encoding="utf-8" ?>
_ <wsdl:definitions name="MathService" targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://tempuri.org/"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
xmlns:wsa10="http://www.w3.org/2005/08/addressing"
xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata">
<wsdl:import namespace=""
location="http://localhost:65182/MathService.svc?wsdl=wsdl0" />
<wsdl:types />
_ <wsdl:binding name="BasicHttpBinding_MathService" type="MathService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
_ <wsdl:operation name="Add">
<soap:operation soapAction="urn:MathService/Add" style="document" />
_ <wsdl:input>
<soap:body use="literal" />
</wsdl:input>
_ <wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
_ <wsdl:service name="MathService">
_ <wsdl:port name="BasicHttpBinding_MathService"
```

```

binding="tns:BasicHttpBinding_MathService">
<soap:address location="http://localhost:65182/MathService.svc" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

<?xml version="1.0" encoding="utf-8" ?>
= <wsdl:definitions targetNamespace=""
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"
  xmlns:wsa10="http://www.w3.org/2005/08/addressing"
  xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex"
  xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata">
= <wsdl:types>
= <xsd:schema targetNamespace="/Imports">
<xsd:import schemaLocation="http://localhost:65182/MathService.svc?xsd=xsd0"
/>
<xsd:import schemaLocation="http://localhost:65182/MathService.svc?xsd=xsd1"
  namespace="http://schemas.microsoft.com/2003/10/Serialization/" />
  </xsd:schema>
</wsdl:types>
= <wsdl:message name="MathService_Add_InputMessage">
<wsdl:part name="parameters" element="Add" />
  </wsdl:message>
= <wsdl:message name="MathService_Add_OutputMessage">
<wsdl:part name="parameters" element="AddResponse" />
  </wsdl:message>
= <wsdl:portType name="MathService">
= <wsdl:operation name="Add">
<wsdl:input wsaw:Action="urn:MathService/Add"
  message="MathService_Add_InputMessage" />
<wsdl:output wsaw:Action="urn:MathService/AddResponse"
  message="MathService_Add_OutputMessage" />
  </wsdl:operation>
</wsdl:portType>
</wsdl:definitions>

```

```

<?xml version="1.0" encoding="utf-8" ?>
- <xs:schema elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
- <xs:element name="Add">
- <xs:complexType>
- <xs:sequence>
<xs:element minOccurs="0" name="a" type="xs:double" />
<xs:element minOccurs="0" name="b" type="xs:double" />
  </xs:sequence>
</xs:complexType>
</xs:element>
- <xs:element name="AddResponse">
- <xs:complexType>
- <xs:sequence>
<xs:element minOccurs="0" name="AddResult" type="xs:double" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

ΣΧΗΜΑ 6 WSDL ΠΕΡΙΓΡΑΦΗ ΣΕ XML

## 2.5 Representational State Transfer (REST)

Το Representational State Transfer (REST) είναι ένα στυλ αρχιτεκτονικής για κατακευματισμένα συστήματα το οποίο χρησιμοποιεί ίδιες τεχνικές με το πρωτόκολλο του διαδικτύου (World Wide Web). Ο όρος εισήχθη από τον Roy Fielding στην διδακτορική του διατριβή το 2000. Ο Fielding είναι ένας από τους συγγραφείς του HTTP πρωτοκόλλου, για τις εκδόσεις 1.0 και 1.1. [27,28]

Το REST αναπτύχθηκε παράλληλα με το HTTP/1.1, βασισμένο στο σχεδιασμό του HTTP/1.0. Η μεγαλύτερη υλοποίηση του REST είναι το δίκτυο παγκοσμίου ιστού. Το REST μας δείχνει πως η αρχιτεκτονική του Internet αναπτύχθηκε με τον χαρακτηρισμό και την κατηγοριοποίηση των 4 τμημάτων του Internet, ήτοι των origin servers, gateways, proxies και clients.

Η αρχιτεκτονική του REST αποτελείται από πελάτες και εξυπηρετητές (clients and servers). Οι πελάτες δημιουργούν αιτήματα προς τους εξυπηρετητές και, εκείνοι τα επεξεργάζονται και επιστρέφουν

κατάλληλες απαντήσεις. Τα αιτήματα και οι απαντήσεις είναι κατασκευασμένα με βάση τη μεταφορά πόρων. Ένας πόρος μπορεί να είναι κάτι στο δίκτυο στο οποίο μπορεί να του απευθυνθεί αίτημα.

Ο πελάτης μπορεί να αρχίσει να στέλνει αιτήματα, και να περιμένει για απάντηση (εκτός και αν το αίτημα είναι τύπου fire and forget). Τα μηνύματα που μεταφέρονται μέσω REST μπορεί να είναι οποιασδήποτε μορφής, όπως για παράδειγμα δυαδικά (binary), POX (Plain Old XML – κλασσικά XML έγγραφα) ή JSON (JavaScript Object Notation – μια ειδική γλώσσα σήμανσης, η οποία περιγράφεται παρακάτω).

### 2.5.1 Περιορισμοί

Η αρχιτεκτονική του REST περιγράφει τους 6 παρακάτω περιορισμούς, ενώ αφήνει τις λεπτομέρειες της υλοποίησης ανοιχτές στον προγραμματιστή

- **Διαχωρισμός client και server.** Οι πελάτες είναι διαχωρισμένοι από τους εξυπηρετητές με μια συγκεκριμένη διεπαφή. Για παράδειγμα, οι πελάτες δεν ανησυχούν για την αποθήκευση των δεδομένων, αλλά αφήνεται στον εξυπηρετητή. Έτσι, υπάρχει φορητότητα στον κώδικα του πελάτη. Επίσης, οι εξυπηρετητές δεν ασχολούνται με την διεπαφή του χρήστη, αρά είναι απλούστεροι και επεκτάσιμοι (scalable).
- **Stateless.** Η επικοινωνία πελάτη και εξυπηρετητή επιβάλλει να μην αποθηκεύεται πληροφορία (client context) στον εξυπηρετητή κατά τη διάρκεια αιτήματος/απάντησης. Κάθε αίτημα από τον πελάτη πρέπει να ενσωματώνει χαρακτηριστικά για την αναγνώριση του αιτήματος (όπως για παράδειγμα πληροφορίες εισόδου – login information).
- **Cacheable.** Το HTTP πρωτόκολλο, από τη φύση του, επιτρέπει στους πελάτες να αποθηκεύουν (cache) πληροφορίες τις οποίες κατεβάζουν από κάποιον εξυπηρετητή. Έτσι, οι REST απαντήσεις πρέπει να προσδιορίζουν αν είναι εφικτό να αποθηκευτούν στον πελάτη, με άμεση συνέπεια να μικραίνουν το φόρτο του εξυπηρετητή σε ενδεχόμενο νέο αίτημα
- **Layered System.** Ένας πελάτης δεν μπορεί να προσδιορίσει αν είναι κατευθείαν συνδεδεμένος στον εξυπηρετητή ή σε κάποιον ενδιάμεσο. Οι ενδιάμεσοι εξυπηρετητές μπορούν να βελτιώσουν την απόδοση και την ασφάλεια του συστήματος
- **Code on demand.** Οι εξυπηρετητές μπορούν να μεταφέρουν/επεκτείνουν τμήμα της λογικής στον πελάτη (π.χ. με χρήση JavaScript, Silverlight, Flash)



- **Uniform Interface.** Οι REST υπηρεσίες θα πρέπει να είναι loosely coupled, ώστε να μπορούν να αναπτυχθούν παράλληλα σε πελάτη και εξυπηρετητή.

## 2.7 Σύγκριση SOAP με REST

Πολλές συζητήσεις γίνονται τελευταία για τη σύγκριση του SOAP με το REST. Η αλήθεια είναι ότι δεν είναι απόλυτα συγκρίσιμα, μιας και το SOAP είναι πρωτόκολλο, ενώ το REST είναι αρχιτεκτονική. Εντούτοις, θα αποπειραθούμε να συγκρίνουμε το SOAP με την πιο συχνή χρήση του REST η οποία είναι η υλοποίηση web services που απαντούν/δέχονται αιτήσεις με HTTP ρήματα (verbs) και τα μηνύματα είναι σε JSON.

- Χρήση εργαλείων. Στο SOAP είναι πολύ πιο εύκολη η παραγωγή των προγραμματιστικών κλάσεων για επικοινωνία με το web service, καθώς με τα σύγχρονα εργαλεία προγραμματιστών (Integrated Development Environments – IDEs) είναι πολύ εύκολη η παραγωγή των λεγόμενων “proxy” κλάσεων, λόγω της ύπαρξης του WSDL. Το εργαλείο αναγνωρίζει το WSDL και παράγει τον απαιτούμενο κώδικα δημιουργίας των XML μηνυμάτων στη γλώσσα που έχει επιλέξει ο προγραμματιστής (π.χ. .NET, Java). Στην περίπτωση του REST δεν υπάρχει κάτι αντίστοιχο.
- Μορφή μηνυμάτων. Μπορεί το XML να θεωρείται ότι είναι καλύτερα αναγνώσιμο από άνθρωπο (human readable) αλλά το REST (με JSON) είναι πιο ελαφρύ σαν μορφοποίηση, καθώς δεν περιέχει ετικέτες (tags) για κάθε στοιχείο. Αυτό έχει ιδιαίτερη σημασία, καθώς είναι πιο γρήγορο στην ανάγνωση από μηχανή σε σχέση με το XML αλλά και μεταφέρει λιγότερη πληροφορία, το οποίο είναι ιδιαίτερα σημαντικό για χρήση από κινητές συσκευές, τόσο από άποψη απόδοσης αλλά και κόστους χρήσης δικτύων κινητής τηλεφωνίας.
- Προσωρινή αποθήκευση δεδομένων (caching). Στην περίπτωση της προσωρινής αποθήκευσης, το REST είναι πιο βολικό, καθώς χρησιμοποιεί το caching του HTTP πρωτοκόλλου και του εξυπηρετητή διαδικτύου, οπότε προγραμματίζεται πολύ εύκολα. Στην περίπτωση του XML, απαιτεί ειδικές ρυθμίσεις στον κώδικα της εφαρμογής.
- Κατανομή φόρτου (load balancing). Πάλι, λόγω του HTTP πρωτοκόλλου, γίνεται μια ρύθμιση στον web server και μπορεί πολύ εύκολα να υπάρχει μια φάρμα εξυπηρετητών (server farm) η οποία και θα αποστέλλει/δέχεται REST (με JSON) μηνύματα.
- Επιχειρησιακές διαδικασίες (enterprise processes). Εδώ σαφώς υπερτερεί το SOAP, κυρίως λόγω της σουίτας πρωτοκόλλων που φέρνει (WS-Security, WS-Transactions και όλα τα υπόλοιπα WS-\* πρωτόκολλα). Στην περίπτωση του REST, εκτός από την προσθήκη SSL πιστοποιητικού στον εξυπηρετητή διαδικτύου, όλα τα υπόλοιπα πρέπει να γίνουμε σε επίπεδο εφαρμογής.

Καταλήγοντας, δεν μπορεί να ανακηρυχτεί νικητής μεταξύ των δύο τεχνολογιών, καθώς και οι δύο έχουν σαφή πλεονεκτήματα και μειονεκτήματα. Γενικά, είθισται να χρησιμοποιείται το SOAP σε περισσότερο «εταιρικά» περιβάλλοντα, ενώ η άνθηση του REST πηγάζει κυρίως από τη χρήση τους σε διάφορους δικτυακούς τόπους κοινωνικής δικτύωσης, όπως το Facebook και το Twitter.

## 2.6 JavaScript Object Notation (JSON)

Το JSON είναι ένα ανοιχτό πρότυπο σχεδιασμένο για δεδομένα αναγνωρίσιμα από άνθρωπο. Προέρχεται από τη γλώσσα JavaScript και πιο συγκεκριμένα από την αναπαράσταση απλών τύπων δεδομένων και συσχετισμένων (associative) πινάκων. Παρά τη σχέση της με την JavaScript, είναι ανεξάρτητη από την γλώσσα προγραμματισμού. Το JSON format ορίστηκε αρχικά από τον Douglas Crockford και περιγράφεται στο RFC 4627. Χρησιμοποιείται για αποθήκευση και μεταφορά δεδομένων, σαν υποκατάστατο της XML. [27-28]

### 2.6.1 Ιστορικά

Το JSON χρησιμοποιήθηκε από τη State Software, μία εταιρία που συνίδρυσε ο Crockford το 2001. Η σελίδα [www.json.org](http://www.json.org) ξεκίνησε να λειτουργεί το 2002. Το 2005, το Yahoo ξεκίνησε να προσφέρει μερικές από τις υπηρεσίες ιστού του σε JSON. Η Google ξεκίνησε να προσφέρει υπηρεσίες σε JSON το 2006.

### 2.6.2 Τύποι δεδομένων και σύνταξη

Οι βασικοί τύποι της JSON είναι

- Number
- String
- Boolean
- Array
- Object
- Null

Το παρακάτω παράδειγμα δείχνει αναπαράσταση ενός Person αντικειμένου σε JSON. Το αντικείμενο έχει πεδία strings για όνομα και επώνυμο, αριθμό για ηλικία, ένα αντικείμενο για τη διεύθυνση και μία λίστα (πίνακα) για αντικείμενα τηλεφωνικών αριθμών.

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "age": 25,
```

```

"address":
{
  "streetAddress": "21 2nd Street",
  "city": "New York",
  "state": "NY",
  "postalCode": "10021"
},
"phoneNumber":
[
  {
    "type": "home",
    "number": "212 555-1234"
  },
  {
    "type": "fax",
    "number": "646 555-4567"
  }
]
}

```

ΣΧΗΜΑ 7 JSON ΚΩΔΙΚΑΣ

Το αντίστοιχο σε XML θα μπορούσε να ήταν:

```

<Object>
  <Property><Key>firstName</Key>    <String>John</String></Property>
  <Property><Key>lastName</Key>     <String>Smith</String></Property>
  <Property><Key>age</Key>           <Number>25</Number></Property>
  <Property><Key>address</Key>
    <Object>
      <Property><Key>streetAddress</Key> <String>21          2nd
Street</String></Property>
      <Property><Key>city</Key>         <String>New York</String></Property>
      <Property><Key>state</Key>       <String>NY</String></Property>
      <Property><Key>postalCode</Key>  <String>10021</String></Property>
    </Object>
  </Property>
</Object>

```

```

    </Object>
  </Property>
  <Property><Key>phoneNumber</Key>
    <Array>
      <Object>
        <Property><Key>type</Key>          <String>home</String></Property>
        <Property><Key>number</Key>       <String>212  555-
1234</String></Property>
      </Object>
      <Object>
        <Property><Key>type</Key>          <String>fax</String></Property>
        <Property><Key>number</Key>       <String>646  555-
4567</String></Property>
      </Object>
    </Array>
  </Property>
</Object>

```

ΣΧΗΜΑ 8 XML ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ JSON

ή, με χρήση attributes

```

<Person firstName="John" lastName="Smith" age="25">
  <Address streetAddress="21 2nd Street" city="New York" state="NY"
postalCode="10021" />
  <PhoneNumbers>
    <PhoneNumber type="home" number="212 555-1234"/>
    <PhoneNumber type="fax" number="646 555-4567"/>
  </PhoneNumbers>
</Person>

```

ΣΧΗΜΑ 9 XML ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ JSON ΜΕ ATTRIBUTES

## 2.8 Σύγκριση XML με JSON

Τα τελευταία χρόνια έχει αρχίσει και χρησιμοποιείται πολύ περισσότερο το JSON. Αυτό γιατί, με την άνθηση της χρήσης των κινητών συσκευών, οι προγραμματιστές άρχισαν να το χρησιμοποιούν

περισσότερο, καθώς είναι πιο ελαφρύ οπότε και μεταφέρει λιγότερα δεδομένα μέσω των δικτύων κινητής τηλεφωνίας, όπως αναφέρθηκε παραπάνω. Παρόλα αυτά, η XML είναι πιο εύκολα αναγνώσιμη από άνθρωπο, ενώ, λόγω της παλαιότητάς της, υπάρχουν πολλά δείγματα κώδικα για τις περισσότερες γλώσσες προγραμματισμού για την ανάκτησή και δημιουργία μηνυμάτων XML. Επιπροσθέτως, έχουν δημιουργηθεί πολλά frameworks για τους ίδιους σκοπούς, όπως XPath, XQuery, XML DOM (και οι τρεις είναι τεχνολογίες για επεξεργασία εγγράφων XML), LINQ to XML (τεχνολογία της Microsoft για επεξεργασία εγγράφων XML με μία γλώσσα παρόμοια της SQL) κ.α.

Η χρήση ή της μίας ή της άλλης γλώσσας εξαρτάται καταρχάς από την αρχιτεκτονική/πρωτόκολλο (SOAP ή REST), από τις γνώσεις/εμπειρία των προγραμματιστών και από την τεχνολογία η οποία θα χρησιμοποιηθεί για την υλοποίηση των δικτυακών υπηρεσιών.

## 2.9 BPEL (Business Process Execution Language)

Η γλώσσα εκτέλεσης επιχειρησιακών διαδικασιών (BPEL) είναι ένα πρότυπο που δημιουργήθηκε από τον διεθνή οργανισμό προτυποποίησης OASIS για να προσδιορίσει δράσεις μεταξύ επιχειρησιακών διαδικασιών με υπηρεσίες διαδικτύου. Οι διεργασίες στην BPEL μπορούν να εξάγουν και να εισάγουν πληροφορία αποκλειστικά με τη χρήση υπηρεσιών διαδικτύου. [19]

Η IBM και η Microsoft είχαν δημιουργήσει τις δικές τους, (αρκετά παρόμοιες) γλώσσες προγραμματισμού υψηλού επιπέδου, τις WSFL και XLANG αντίστοιχα. Με την εξάπλωση της BPML, την επιτυχία του BPMI.org και το ανοιχτό BPMS κίνημα των JBoss και Intanio Inc., η IBM και η Microsoft αποφάσισαν να συνδυάσουν αυτές τις γλώσσες σε μια καινούρια, την BPEL4WS (σημαίνει BPEL for Web Services και είναι μία μορφή της γλώσσας BPEL για παρουσίαση – expose – επιχειρησιακών διαδικασιών με χρήση υπηρεσιών διαδικτύου). Τον Απρίλιο του 2003, η BEA, η IBM, η Microsoft, η SAP και η Siebel προτείνανε το BPEL4WS 1.1 στον OASIS για προτυποποίηση μέσω της Web Services BPEL Technical Committee. Ενώ το BPEL4WS εμφανίστηκε ως έκδοση 1.0 και 1.1, η επιτροπή ψήφισε να ονομάσει την προδιαγραφή ως WS-BPEL 2.0.

Οι συναλλαγές με υπηρεσίες διαδικτύου είναι εφικτό να περιγραφούν με δύο τρόπους: εκτελέσιμες επιχειρησιακές διαδικασίες και αφηρημένες επιχειρησιακές διαδικασίες. Οι εκτελέσιμες επιχειρησιακές διαδικασίες μοντελοποιούν την πραγματική συμπεριφορά ενός συμμετέχοντα σε μία επιχειρησιακή συναλλαγή περιλαμβάνοντας, για παράδειγμα, όλες τις αλληλεπιδράσεις του με τις δικτυακές υπηρεσίες που μπορεί να λαμβάνουν μέρος στη διαδικασία, όλους τους ελέγχους ασφαλείας κ.α.. Οι αφηρημένες επιχειρησιακές διαδικασίες είναι μερικώς προσδιορισμένες διαδικασίες που δεν

έχουν ως αυτοσκοπό να εκτελεστούν αλλά να παρουσιάσουν τη διαδικασία σε κάποιον ενδιαφερόμενο. Μία αφηρημένη διαδικασία σχεδόν πάντα κρύβει κάποια/ες από τις λεπτομέρειες της επιχειρησιακής διαδικασίας. Οι αφηρημένες διαδικασίες εξυπηρετούν έναν περιγραφικό ρόλο, με περισσότερες από μία περιπτώσεις χρήσης (use case) και περιλαμβάνουν παρατηρήσιμη συμπεριφορά και/η πρότυπο διαδικασίας. Η WS-BPEL έχει σαν δουλειά να μοντελοποιήσει τη συμπεριφορά και των επιχειρησιακών και των αφηρημένων διαδικασιών.

Η WS-BPEL παρέχει μία γλώσσα για τον προσδιορισμό των εκτελέσιμων και αφηρημένων επιχειρησιακών διαδικασιών. Με αυτό τον τρόπο, επεκτείνει το μοντέλο αλληλεπίδρασης των Υπηρεσιών Διαδικτύου και του επιτρέπει να υποστηρίξει επιχειρησιακές συναλλαγές. Η WS-BPEL προσδιορίζει έναν διαλειτουργικό μοντέλο που θα διευκολύνει την επέκταση της αυτόματης ολοκλήρωσης διαδικασιών και εσωτερικά αλλά και μεταξύ των διαδικασιών.

Η BPEL στηρίζεται στην XML και έχει σαν στόχο τον «προγραμματισμό υψηλού επιπέδου» (programming in the large). Οι ιδέες του προγραμματισμού υψηλού επιπέδου και του προγραμματισμού χαμηλού επιπέδου (programming in the small) διαχωρίζουν δύο απόψεις συγγραφής των εκτελούμενων για μεγάλο χρονικό διάστημα διεργασιών, τις οποίες που βρίσκει, τυπικά, κάποιος σε επιχειρησιακές διαδικασίες. Για παράδειγμα, μία διαδικασία ανάληψης χρημάτων από ένα αυτόματο μηχάνημα τραπεζής (ATM) στον «προγραμματισμό υψηλού επιπέδου» θα περιλάμβανε (μεταξύ άλλων) πρόσβαση σε υπηρεσία του τμήματος λογαριασμών για έλεγχο υπολοίπου, πρόσβαση στην υπηρεσία καρτών για έλεγχο της κάρτας του πελάτη κ.α. Αντιθέτως, η ίδια διαδικασία στον «προγραμματισμό χαμηλού επιπέδου» θα έπρεπε να περιγραφεί πολύ πιο αναλυτικά, με αναφορές στις προσβάσεις στο σύστημα αρχείων, σε βάσεις δεδομένων, ελέγχους ασφαλείας κ.λ.π.

Ο προγραμματισμός υψηλού επιπέδου (όταν μιλάμε για συγγραφή επιχειρησιακών διαδικασιών) γενικά αναφέρεται στην υψηλού επιπέδου μετάβαση καταστάσεων – η BPEL αναφέρεται σε αυτό ως μια αφηρημένη διεργασία (abstract process). Μια αφηρημένη διεργασία στη BPEL αντιπροσωπεύει πληροφορία όπως το πότε να περιμένει για μηνύματα, πότε να αποστείλει, πότε να λάβει δράση για αποτυχημένες συναλλαγές κ.α. Ο προγραμματισμός χαμηλού επιπέδου, αντιθέτως, αντιμετωπίζει με μικρής διάρκειας προγραμματιστική συμπεριφορά, που εκτελείται συχνά ως μία συναλλαγή και περιλαμβάνει πρόσβαση στην τοπική λογική και σε πόρους όπως αρχεία, βάσεις δεδομένων κ.α. Η

ανάπτυξη της BPEL ξεκίνησε από την ιδέα ότι ο προγραμματισμός υψηλού επιπέδου χρειάζεται διαφορετικές γλώσσες σε σχέση με τον προγραμματισμό χαμηλού επιπέδου.

### 2.9.1 Στόχοι της BPEL

Αρχικά, υπήρχαν οι κάτωθι στόχοι για την BPEL

- Να προσδιορίσει τις επιχειρησιακές διεργασίες που αλληλεπιδρούν με εξωτερικές οντότητες μέσω λειτουργιών υπηρεσιών διαδικτύου που προσδιορίζονται μέσω WSDL 1.1, και «υπογράφονται» ως υπηρεσίες διαδικτύου με πρότυπο WSDL 1.1. Οι αλληλεπιδράσεις είναι «αφηρημένες» με την έννοια ότι η εξάρτηση είναι στον τύπο της θύρας επικοινωνίας, όχι στην ίδια τη θύρα.
- Να προσδιορίσει τις επιχειρησιακές διαδικασίες με μια γλώσσα βασισμένη στην XML. Δεν προσδιορίζει γραφικά τις διαδικασίες ούτε δίνει συγκεκριμένες μεθοδολογίες σχεδιασμού για αυτές
- Να προσδιορίσει ένα σύνολο από ιδέες σχετικές με ολοκλήρωση Δικτυακών Υπηρεσιών που πρόκειται να χρησιμοποιηθούν τόσο από τους εξωτερικές και εσωτερικές όψεις μιας επιχειρησιακής διαδικασίας. Μια τέτοια διαδικασία προσδιορίζει την συμπεριφορά μιας αυτόνομης οντότητας, τυπικά λειτουργώντας σε συσχέτισμό με άλλες οντότητες. Αναγνωρίζεται ότι κάθε τέτοιο σενάριο χρήσης θα απαιτεί συγκεκριμένες επεκτάσεις – ο αριθμός των οποίων θα πρέπει να είναι ο μικρότερος δυνατός – οι οποίες θα δοκιμαστούν σε απαιτήσεις όπως εισαγωγή/εξαγωγή και έλεγχο συμμόρφωσης.
- Να παρέχει και ιεραρχικό και γραφικό τρόπο έλεγχου συστήματος, και να επιτρέπει η αφομοίωσή τους να είναι όσο πιο απλή γίνεται.
- Να παρέχει μηχανισμούς επεξεργασίας δεδομένων για να προσδιορίσουν επιχειρησιακά δεδομένα και έλεγχο ροής.
- Να υποστηρίξει έναν μηχανισμό ταυτοποίησης για διεργασίες που επιτρέπει τον προσδιορισμό αναγνωριστικών στο επίπεδο εφαρμογής.
- Να υποστηρίξει την δημιουργία και τον τερματισμό διεργασιών
- Να προσδιορίσει ένα μακράς διάρκειας μοντέλο συναλλαγών βασισμένο σε αποδεδειγμένες τεχνικές όπως δράσης σε αποτυχία

- Να χρησιμοποιήσει υπηρεσίες διαδικτύου σαν το μοντέλο αποσύνθεσης και επανασύνθεσης επιχειρησιακών διαδικασιών
- Να βασιστεί σε πρότυπα υπηρεσιών διαδικτύου όσο περισσότερο γίνεται, με έναν δομημένο τρόπο [3,30]

### 2.9.2 Η γλώσσα BPEL

Η BPEL είναι μια γλώσσα ολοκλήρωσης, όχι μια γλώσσα χωρογραφίας. Η κύρια διαφορά μεταξύ ολοκλήρωσης και χωρογραφίας είναι η εκτελεσιμότητα και ο έλεγχος. Η ολοκλήρωση προσδιορίζει μία εκτελέσιμη διαδικασία που περιλαμβάνει ανταλλαγή μηνυμάτων με άλλα συστήματα, έτσι ώστε οι ροές να ελέγχονται από τον σχεδιαστή της ολοκλήρωσης. Η χωρογραφία περιγράφει ένα πρωτόκολλο για P2P αλληλεπιδράσεις, προσδιορίζοντας διάφορα πράγματα όπως τις συμφωνίες για διαλειτουργικότητα. Ένα τέτοιο πρωτόκολλο δεν είναι κατευθείαν εκτελέσιμο, καθώς επιτρέπει πολλές διαφορετικές αντιπροσωπεύσεις.

Η επικέντρωση της BPEL σε σύγχρονες επιχειρησιακές διαδικασίες, καθώς και το παρελθόν WSFL και της XLANG, οδήγησαν την BPEL να προσδιορίσει τις δικτυακές υπηρεσίες σαν τον εξωτερικό επικοινωνιακό μηχανισμό. Έτσι, οι υποδομές ανταλλαγής μηνυμάτων της BPEL εξαρτώνται από τη χρήση WSDL για να περιγράψουν εξερχόμενα και εισερχόμενα μηνύματα.

Επιπροσθέτως, η γλώσσα BPEL υποστηρίζει

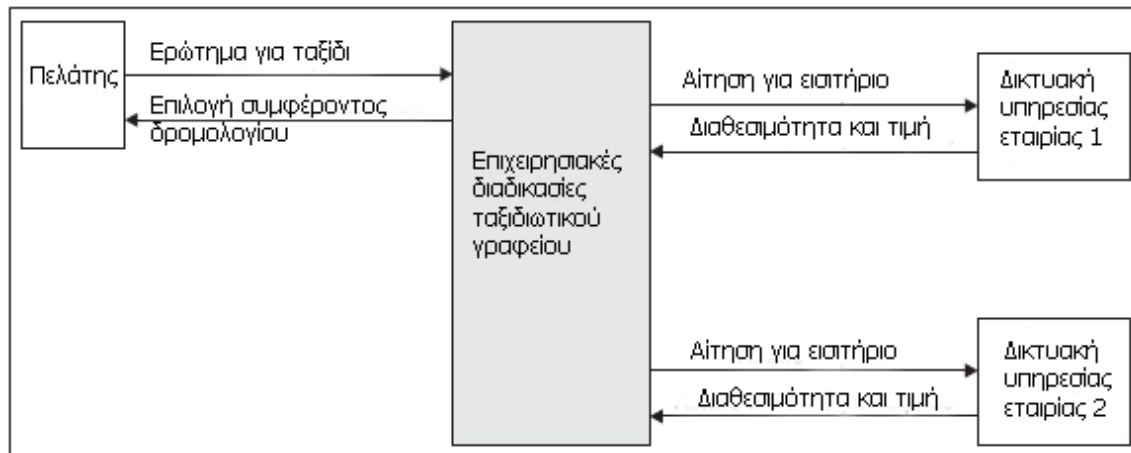
- Ένα μηχανισμό αλληλοσυσχέτισης μηνυμάτων μέσω ιδιοτήτων
- Μεταβλητές σε XML και WSDL
- Ένα μοντέλο προσθέτων (plug-ins) για να επιτραπεί η συγγραφή εκφράσεων και ερωτημάτων σε πολλαπλές γλώσσες (π.χ. η BPEL υποστηρίζει XPath 1.0)
- Δομές δομημένου προγραμματισμού (if-then-else, while)
- Ένα σύστημα ορατότητας μεταβλητών (scope), που επιτρέπει την ενθυλάκωση λογικής με τοπικές μεταβλητές, τμήματα κώδικα που χειρίζονται λάθη και χειριστές συμβάντων (event – handlers)

### 2.7.3 Σύνθεση υπηρεσιών

Οι υπηρεσίες παρέχουν ευθυγραμμισμένες επιχειρησιακές διαδικασίες. Για να δημιουργήσουμε μια SOA αρχιτεκτονική, θα πρέπει να συνθέσουμε τις υπηρεσίες έτσι ώστε το συνονθύλευμα των



υπηρεσιών να παρέχει υποστήριξη για όλες τις επιχειρησιακές διαδικασίες (μια συλλογή δραστηριοτήτων μέσω των οποίων καλούνται οι υπηρεσίες). Σε ένα πραγματικό σενάριο συνήθως δημιουργούμε δύο είδη επιχειρησιακών διαδικασιών, εκείνες που παρέχουν ενδοεπιχειρησιακές υπηρεσίες και εκείνες που θα καταναλώσουν υπηρεσίες που παρέχονται από εξωτερικές πηγές. Σαν παράδειγμα μπορούμε να μελετήσουμε την περίπτωση ενός ταξιδιωτικού γραφείου. Σε μια απλή περίπτωση, η επιχειρησιακή διαδικασία θα ζητήσει το όνομα του υπαλλήλου που πραγματοποιεί μια κράτηση, τον προορισμό, τις ημερομηνίες κ.α. Εν συνεχεία, η διαδικασία θα πραγματοποιήσει μια κλήση σε διάφορες υπηρεσίες ιστού διαφόρων αεροπορικών εταιριών για να δει αν υπάρχει διαθεσιμότητα και να επιστρέψει τις τιμές. Μπορούμε να δούμε μια δομή των υπηρεσιών που συνθέτονται στο παρακάτω σχήμα (σχήμα 10).



ΣΧΗΜΑ 10 ΕΠΙΧΕΙΡΗΣΙΑΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ ΤΑΞΙΔΙΩΤΙΚΟΥ ΓΡΑΦΕΙΟΥ [3]

Σαν συνολική εικόνα της επιχειρησιακής διαδικασίας, δεν μας ενδιαφέρει από πού η υπηρεσία ιστού ανακτά τις απαραίτητες πληροφορίες για τις τιμές των εισιτηρίων. Επιπροσθέτως, δεν μας απασχολεί αν οι υπηρεσίες ιστού αποτελούνται/συντίθενται από άλλες υπηρεσίες. Ο πελάτης, επίσης, βλέπει τη διαδικασία όπως οποιαδήποτε άλλη υπηρεσία, χωρίς να τον ενδιαφέρει η κατασκευή της διαδικασίας.

Η σύνθεση των υπηρεσιών σε επιχειρησιακές διαδικασίες μπορεί να πραγματοποιηθεί μέσω οποιουδήποτε γνωστού framework κατασκευής επιχειρησιακών εφαρμογών (.NET, J2EE κ.α.). Παρόλα αυτά, υπάρχει μεγάλος κίνδυνος για τη δημιουργία μιας δυσκολοσυντήρητης εφαρμογής, καθώς η ροή εργασίας και η επιχειρησιακή λογική δεν πρέπει να συνδέονται. Για αυτό το λόγο, έχουν αναπτυχθεί αρκετά πακέτα λογισμικού ονόματι BPM (Business Process Management). [3]

#### 2.7.4 Η BPEL στη σύνθεση των υπηρεσιών

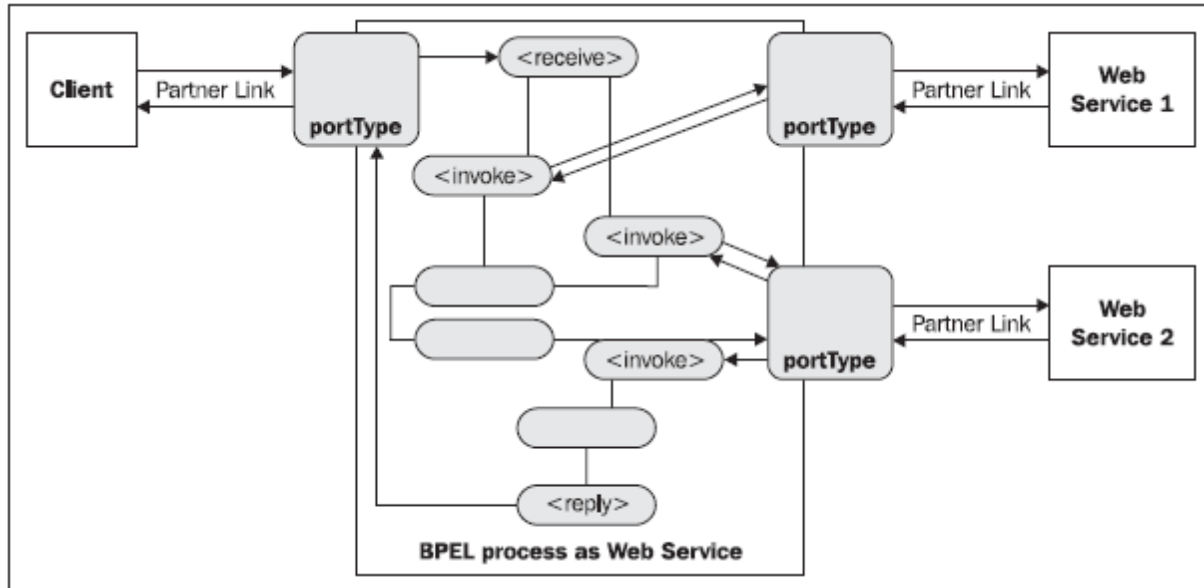
Η BPEL χρησιμοποιείται από τις επιχειρήσεις για να τυποποιήσει την ενσωμάτωση των επιχειρηματικών εφαρμογών και να επεκτείνει την ενσωμάτωση τους σε απομονωμένα συστήματα. Μεταξύ των επιχειρήσεων, η BPEL επιτρέπει την ευκολότερη και αποτελεσματικότερη ενσωμάτωση με τις

υπηρεσίες που παρέχονται από συνεργάτες εταιρίες. Η BPEL είναι μια από τις βασικές τεχνολογία σε περιβάλλοντα όπου οι λειτουργίες είναι ήδη, ή πρόκειται να εκτεθούν μέσω των υπηρεσιών Ιστού.

Μέχρις ενός σημείου, η BPEL μοιάζει με τις παραδοσιακές γλώσσες προγραμματισμού. Προσφέρει δομές, όπως οι βρόχοι, διακλαδώσεις (branches), μεταβλητές, αναθέσεις, κ.α. που μας επιτρέπουν να καθορίσουμε τις επιχειρησιακές διαδικασίες με έναν αλγοριθμικό τρόπο. Η WS-BPEL (υπερσύνολο της BPEL για έκθεση των επιχειρησιακών διαδικασιών μέσω υπηρεσιών διαδικτύου) υποστηρίζει επίσης τις δομημένες δραστηριότητες για την κατασκευή της επιχειρησιακής λογικής για μια διαδικασία. Το χαρακτηριστικό σενάριο είναι ότι παραλαμβάνεται ένα μήνυμα σε μία εκτελέσιμη διαδικασία WS-BPEL. Η διαδικασία μπορεί έπειτα να καλέσει μια σειρά εξωτερικών υπηρεσιών για να συγκεντρώσει πρόσθετα δεδομένα, και να αποκριθεί στον πελάτη που έκανε την αίτηση. Η BPEL είναι μια εξειδικευμένη γλώσσα που εστιάζεται στον καθορισμό των επιχειρησιακών διαδικασιών. Καθιστά εύκολη την πρόσβαση στις διαδικασίες των υπηρεσιών Ιστού είτε σύγχρονα είτε ασύγχρονα, σειριακά ή παράλληλα. Παρέχει επίσης ένα πλούσιο λεξιλόγιο για το χειρισμό σφαλμάτων, γεγονός πολύ σημαντικό δεδομένου ότι οι επιχειρησιακές διαδικασίες πρέπει να αντιδρούν στις αποτυχίες με έναν έξυπνο τρόπο.

Σε ένα χαρακτηριστικό σενάριο, η επιχειρησιακή διαδικασία BPEL λαμβάνει ένα αίτημα. Για να το εκπληρώσει, η διαδικασία καλεί τις εμπλεκόμενες υπηρεσίες Ιστού και αποκρίνεται στον πελάτη. Επειδή η διαδικασία BPEL επικοινωνεί με άλλες υπηρεσίες Ιστού, στηρίζεται σε μεγάλο ποσοστό στην περιγραφή WSDL των υπηρεσιών Ιστού που επικαλούνται από τη σύνθετη υπηρεσία Ιστού.

Η παρακάτω εικόνα παρουσιάζει μία διαδικασία BPEL και τη σχέση της με τις υπηρεσίες Ιστού μέσω των συνδέσμων συνεργατών (partner links):[3]



ΣΧΗΜΑ 11 ΠΑΡΑΔΕΙΓΜΑ BPEL [2]

#### 4.5 Δομή Διαδικασίας

Μια επιχειρησιακή διαδικασία είναι μια συλλογή κλήσεων υπηρεσιών και σχετικών δραστηριοτήτων που παράγουν ένα επιχειρησιακό αποτέλεσμα ή καλούν συγκεκριμένες διαδικασίες του οργανισμού, είτε μέσα σε έναν οργανισμό είτε στα τμήματά του (αν είναι μεγάλος). Κάθε διαδικασία BPEL αποτελείται από συγκεκριμένα βήματα, τα οποία αποκαλούνται δραστηριότητες. Κάθε δραστηριότητα αντιπροσωπεύει ένα μικρό «βήμα» μιας επιχειρησιακής διαδικασίας.

Επί παραδείγματι, η ανάληψη ενός ποσού από μία τράπεζα μέσω ενός μηχανήματος αυτόματης εξυπηρέτησης (ATM) θα μπορούσε να περιλαμβάνει τις ακόλουθες δραστηριότητες

##### Επιχειρησιακή διαδικασία – Ανάληψη

- Έλεγχος της κάρτας τραπέζης του χρήστη (μέσω αποστολής των στοιχείων στο τμήμα καρτών)
- Επιλογή των διαθέσιμων λογαριασμών του χρήστη (μέσω αποστολής των στοιχείων στο τμήμα λογαριασμών)
- Εισαγωγή του ποσού
- Έλεγχος για πιστωτικό υπόλοιπο (μέσω αποστολής των στοιχείων στο τμήμα λογαριασμών)
- Εξαγωγή των χαρτονομισμάτων από το μηχάνημα

Το στοιχείο ρίζα (root element) οποιασδήποτε διαδικασίας BPEL είναι ο ορισμός <process>. Κάθε διαδικασία έχει μια ιδιότητα ονόματος και ένα χώρο ονόματος (namespace), όπως φαίνεται στο ακόλουθο παράδειγμα.

```
<process name="demoBpel"
targetNamespace="http://www.testnamespace.com/test/demoBpel"
xmlns:tns="http://www.testnamespace.com/test/demoBpel"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/business-process/"
xmlns="http://schemas.xmlsoap.org/ws/2004/03/business-process/"
xmlns:ns2="http://www.w3.org/2001/XMLSchema">
<import ...>
</import>
<partnerLinks>
...
</partnerLinks>
<variables>
...
</variables>
<sequence>
...
</sequence>
...
</process>
```

Κάποιες από τις βασικές δραστηριότητες της BPEL που χρησιμοποιούνται, είναι οι εξής:

- <invoke> Επιτρέπει στη διαδικασία να επικαλεσθεί μια one-way ή μία request-response λειτουργία σε ένα port type (ή διεπαφή) που προσφέρεται από μία διαδικασία συνεργάτη.

```
<invoke partnerLink="insuranceA"
portType="ins:ComputeInsurancePremiumPT"
operation="ComputeInsurancePremium"
inputVariable="InsuranceRequest"
outputVariable="InsuranceAResponse" >
</invoke>
```

- <reply> Η BPEL καθορίζει τις βασικές δραστηριότητες που απαιτούνται για να δημιουργήσουν τις συνθέσεις WebService. Όταν δημιουργείται μια σύγχρονη ανταλλαγή μηνυμάτων, κάθε στοιχείο <receive> έχει το αντίστοιχο <reply> στοιχείο. Το στοιχείο <reply> επιτρέπει στη διαδικασία υπηρεσιών να στείλει ένα μήνυμα προς απάντηση σε ένα λαμβανόμενο μήνυμα από μια υπηρεσία συνεργάτη επειδή το στοιχείο συνδέεται με το ίδιο στοιχείο partnerLink όπως το αντίστοιχο <receive> στοιχείο του.

```
<reply partnerLink="client"
portType="com:InsuranceSelectionPT"
operation="SelectInsurance"
variable="InsuranceSelectionResponse" >
</reply>
```

- <receive> Οι δραστηριότητες υπηρεσιών Ιστού καθορίζονται από τη BPEL για να δημιουργήσουν WebService συνθέσεις. Το στοιχείο <receive> καθορίζει τις πληροφορίες που αναμένονται από μια διαδικασία υπηρεσιών ενεργώντας ως φορέας παροχής υπηρεσιών που αναμένει να λάβει ένα αίτημα από μια εξωτερική διαδικασία ενεργώντας ως καταναλωτής υπηρεσιών.

```
<receive partnerLink="client" portType="com:InsuranceSelectionPT"
operation="SelectInsurance" variable="InsuranceRequest" createInstance="yes">
</receive>
```

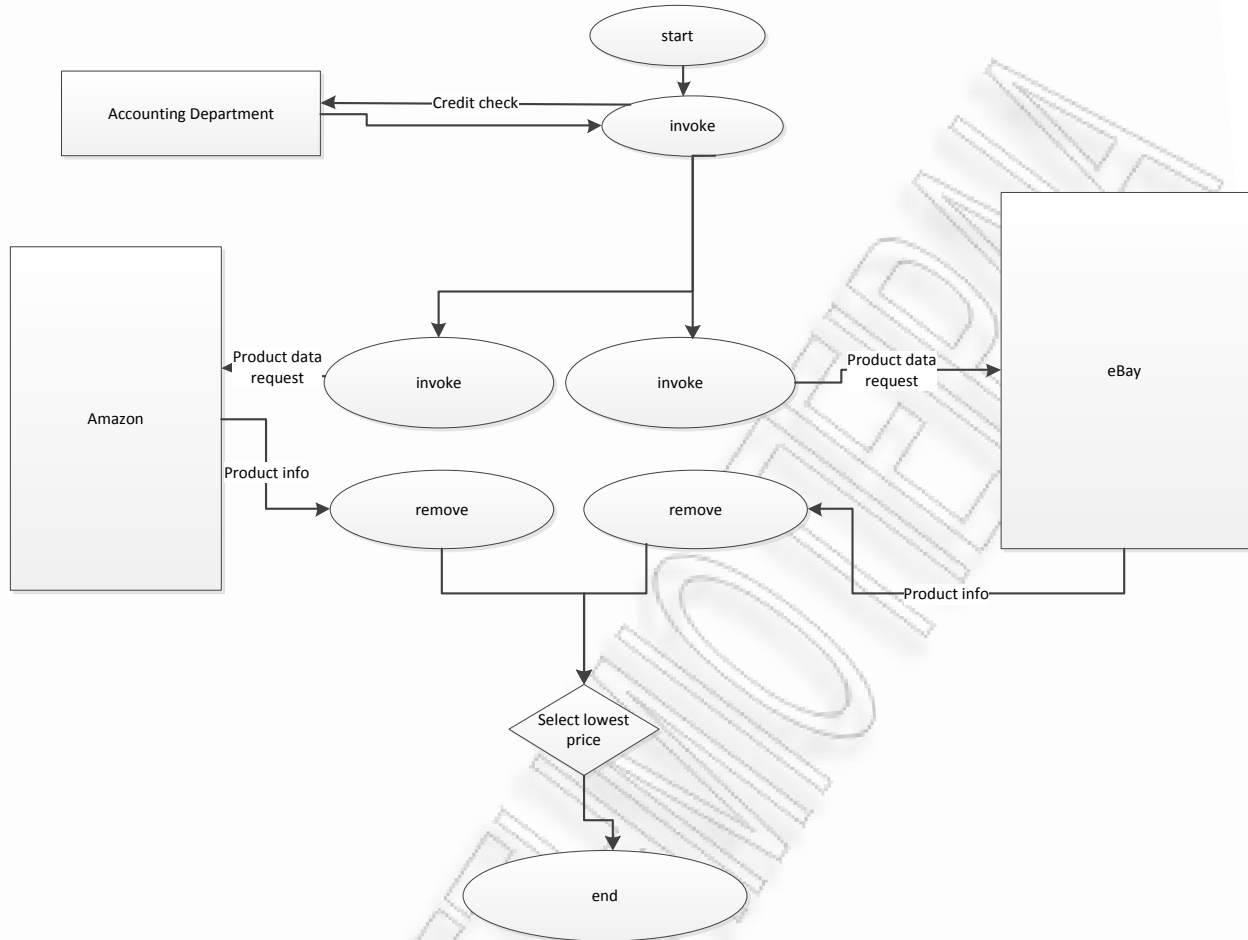
- <assign>, <copy>, <from> και <to> Τα στοιχεία αυτά, μας επιτρέπουν να αντιγράψουμε δεδομένα από μία μεταβλητή σε μία άλλη, καθ' όλη τη διαδικασία.
- Η ένδειξη των σφαλμάτων με τη <throw>. Το στοιχείο αυτό μας επιτρέπει να δημιουργήσουμε ρητά ένα σφάλμα, σε απάντηση σε έναν συγκεκριμένο όρο για να καταδείξουμε ότι έχει εμφανιστεί ένα λάθος κατά την εκτέλεση της διαδικασίας.
- <exit> Το στοιχείο αυτό τερματίζει όλη την διαδικασία ενορχήστρωσης υπηρεσιών με την καταστροφή της.
- <scope> Το στοιχείο αυτό μας δίνει τη δυνατότητα να υποδιαιρέσουμε περιοχές της λογικής μέσα στον καθορισμό διαδικασίας σε πεδία. Μπορούμε έπειτα να καθορίσουμε μεταβλητές, faultHandlers, correlationSets, compensationHandlers και στοιχεία eventHandlers που είναι τοπικές στο πεδίο.

Παρακάτω μπορούμε να δούμε ορισμένα παραδείγματα δομημένων δραστηριοτήτων

- <sequence>. Διαδοχική επεξεργασία των δραστηριοτήτων που βρίσκονται μέσα σε αυτήν
- <switch>. Επιλογή ενός κλάδου (branch) που θα εκτελεστεί
- <flow>. Παράλληλη εκτέλεση μιας σειράς από διεργασίες.
- <while>. Χρησιμοποιείται για τον καθορισμό βρόχων.
- <if>, <elseif>, <else>. Επιλογή ενός κλάδου εκτέλεσης βασισμένου στη λογική.

#### 4.6 Παράδειγμα χρήσης BPEL

Θα δώσουμε ένα παράδειγμα χρήσης BPEL από έναν online μεταπωλητή αντικειμένων. Όταν ληφθεί μια παραγγελία από έναν πελάτη μέσω του δικτυακού τόπου, υπάρχει μία ροή εργασίας για τον καθορισμό της αξίας πίστωσης του πελάτη και την αυτόματη εύρεση του οικονομικότερου αντικειμένου. Η ροή ελέγχει το πιστωτικό υπόλοιπο του πελάτη στο σύστημα πιστωτικού καταλόγου και στέλνει αιτήσεις τιμών για συγκεκριμένα προϊόντα στο Amazon και στο eBay (ηλεκτρονικά καταστήματα, με δημόσιες διεπαφές για υπηρεσίες διαδικτύου). Η απάντηση περιλαμβάνει τρέχουσες τιμές και από τα δύο καταστήματα, οπότε η ροή εργασίας επιλέγει τη χαμηλότερη.



ΣΧΗΜΑ 12 ΠΑΡΑΔΕΙΓΜΑ BPEL [3]

Οι υπηρεσίες παγκόσμιου ιστού εμφανίζονται ως μπλοκ, περιγραφές δεδομένων υψηλού επιπέδου υπάρχουν στα βελάκια και εντολές υποδεικνύουν τις συγκεκριμένες ενέργειες. Πιο συγκεκριμένα, δεν αναφέρεται ο τρόπος με τον οποίο επεξεργάζονται οι υπηρεσίες παγκόσμιου ιστού στους αντίστοιχους εξυπηρετητές. Μόνο η ακολουθία (sequence) των υπηρεσιών (services) και των δεδομένων που εισέρχονται και εξέρχονται στις υπηρεσίες είναι σημαντικά στην BPEL.

Ο κώδικας BPEL για τον έλεγχο πιστωτικού υπολοίπου θα μπορούσε να μοιάζει με τον ακόλουθο

```
<!--sample of generated BPEL code -->
<invoke name="myInvoke" partnerLink="AccountingDept"
portType="services:CreditRatingService" operation="process"
inputVariable="crIn" outputVariable="crOut">
</invoke>
```

Ένα αρχείο BPEL για αυτή τη διαδικασία θα πρέπει να περιέχει

- Ένα μέρος στο οποίο θα αναγνωρίζονταν οι διάφοροι partners των υπηρεσιών παγκόσμιου ιστού που εμπλέκονται στη διαδικασία
- Τις μεταβλητές δεδομένων που ανταλλάσσονται
- Τη ροή εργασίας
- Τους απαραίτητους χειριστές λαθών
- Τις απαραίτητες αναιρέσεις

Ο κώδικας BPEL, όταν συνδυάζεται με WSDL, δημιουργεί όλη την πληροφορία έτσι ώστε να μπορούν να γίνουν σωστά orchestrate οι δικτυακές υπηρεσίες. Ένα αρχείο BPEL μπορεί να εκτελεστεί σε έναν εξυπηρετητή. Μια μηχανή BPEL μπορεί να τρέξει όταν εκτελεστεί μια επιχειρησιακή διαδικασία και να συντονίσει τις διάφορες δραστηριότητες σύμφωνα με τις εντολές που περιέχονται στο BPEL αρχείο. (3)

## 3. Cloud Computing

---

### 3.1 Επισκόπηση

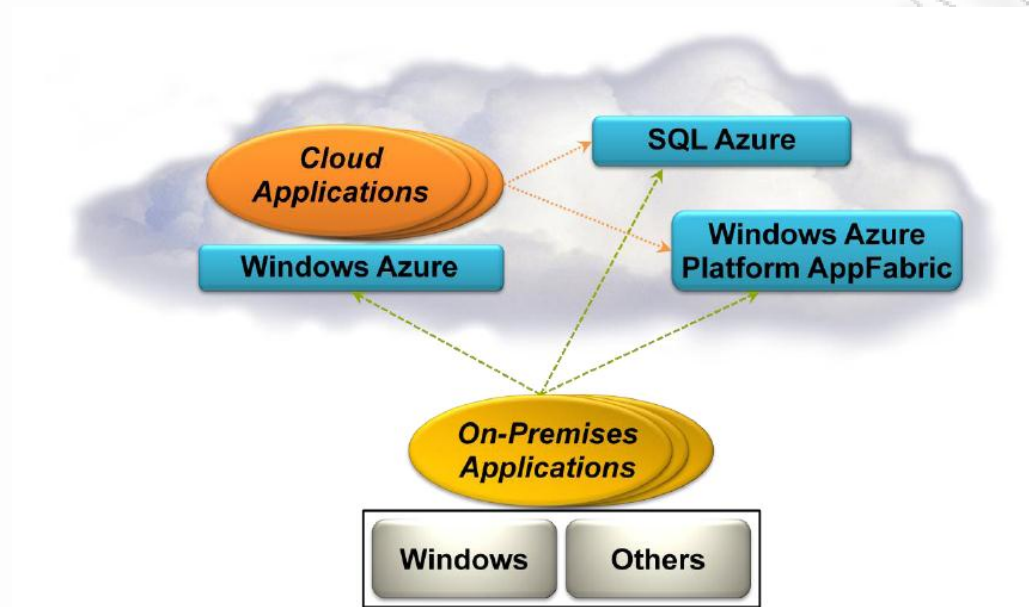
Ένας απλός ορισμός του cloud computing είναι η χρήση από κάποιον οργανισμό του πλήθους των εξυπηρετητών (θεωρητικά απεριόριστο) που βρίσκονται σε κέντρα δεδομένων (data centers) μεγάλων οργανισμών και είναι διαθέσιμοι για ενοικίαση με χρονοχρέωση. Ο οργανισμός θα μπορούσε να προβεί στην αγορά και συντήρηση ιδιόκτητων μηχανών αλλά θα έπρεπε να επωμιστεί το κόστος που αυτό συνεπάγεται (αγορά υλικού, αγορά/εγκατάσταση/συντήρηση λογισμικού, αγορά ιδιόκτητης γραμμής για πρόσβαση στο διαδίκτυο, ψύξη για τα μηχανήματα κ.α.). Αυτό το μοντέλο λέγεται “on premise”. Αντιθέτως, στην περίπτωση χρήσης υπολογιστικού νέφους, τόσο ο κώδικας όσο και τα δεδομένα βρίσκονται σε εξυπηρετητές στο υπολογιστικό νέφος, όπου κάποιος άλλος (ο προμηθευτής που παρέχει την υποδομή υπολογιστικού νέφους (cloud infrastructure)) διαχειρίζεται και συντηρεί το σύστημα που χρησιμοποιούν, οπότε και ο οργανισμός γλιτώνει τα κόστη που αναφέρθησαν προηγουμένως. Επιπροσθέτως, εφαρμογές που τρέχουν μέσα σε έναν οργανισμό μπορούν να αποθηκεύουν δεδομένα στο υπολογιστικό νέφος ή να εξαρτώνται από άλλες δικτυακές υπηρεσίες που μπορεί να τρέχουν στο υπολογιστικό νέφος. Έτσι, η χρήση cloud computing για τις υπηρεσίες ενός οργανισμού δεν είναι αποκλειστική. Ο οργανισμός μπορεί να χρησιμοποιεί τόσο cloud computing όσο και on premise υποδομή.

Όμως είτε μια εφαρμογή τρέχει στο υπολογιστικό νέφος, είτε χρησιμοποιεί υπηρεσίες που παρέχονται από το υπολογιστικό νέφος, είτε και τα δύο, απαιτεί κάποιου είδους πλατφόρμα προκειμένου να εκτελεστεί. Γενικά, μία εφαρμογή πλατφόρμας (ή ένα λειτουργικό σύστημα υπολογιστικού νέφους) μπορεί να οριστεί ως οτιδήποτε το οποίο παρέχει προσβάσιμες υπηρεσίες σε προγραμματιστές για τη δημιουργία και τη φιλοξενία εφαρμογών.

Η πλατφόρμα Windows Azure της Microsoft είναι ένα σύνολο τεχνολογιών υπολογιστικού νέφους (Compute υπηρεσία – διαχειρίζεται τις εφαρμογές του χρήστη, Storage υπηρεσία – αποθηκεύει τα δεδομένα του χρήστη, SQL υπηρεσία – φιλοξενεί τις βάσεις δεδομένων του χρήστη κ.α.), κάθε μια εκ των οποίων παρέχει ένα συγκεκριμένο σύνολο από υπηρεσίες στους προγραμματιστές εφαρμογών.



Όπως φαίνεται στο σχήμα 13, η πλατφόρμα Windows Azure μπορεί να χρησιμοποιηθεί και από εφαρμογές που τρέχουν στο υπολογιστικό νέφος, αλλά και από εφαρμογές που τρέχουν σε εταιρικά κέντρα εξυπηρετητών (on-premises datacenters). [4]



ΣΧΗΜΑ 13 Η ΠΛΑΤΦΟΡΜΑ WINDOWS AZURE ΥΠΟΣΤΗΡΙΖΕΙ ΕΦΑΡΜΟΓΕΣ, ΔΕΔΟΜΕΝΑ ΚΑΙ ΥΠΟΔΟΜΗ ΣΤΟ ΣΥΝΝΕΦΟ [4]

Η πλατφόρμα Azure Services είναι μία πλατφόρμα που τρέχει στο σύννεφο, η οποία επιτρέπει σε εφαρμογές να φιλοξενοούνται και να τρέχουν σε Microsoft κέντρα εξυπηρετητών. Παρέχει ένα λειτουργικό σύστημα σύννεφου, που ονομάζεται Windows Azure και το οποίο χρησιμεύει ως περιβάλλον (runtime) στο οποίο εκτελούνται οι εφαρμογές και παρέχει ένα σύνολο από υπηρεσίες που επιτρέπουν την ανάπτυξη, την διαχείριση και φιλοξενία εφαρμογών εκτός των εταιρικών κέντρων εξυπηρετητών. Όλες οι υπηρεσίες που προσφέρει το Azure αλλά και οι εφαρμογές που κατασκευάζονται και τις χρησιμοποιούν εκτελούνται πάνω στο Windows Azure.

Τα βασικά συστατικά μέρη της πλατφόρμας Windows Azure είναι :

- Το Windows Azure έχει τρία κύρια συστατικά μέρη: Compute, Storage και Fabric, τα οποία θα παρουσιαστούν παρακάτω. Όπως φαίνεται και από τις ονομασίες τους, το Compute παρέχει υπολογιστικό περιβάλλον με Web Role και Worker Role, ενώ το Storage εστιάζει στην παροχή

κλιμακούμενου χώρου αποθήκευσης (Blobs, Tables, Queue, Drives) για ανάγκες μεγάλης κλιμάκωσης (scaling).

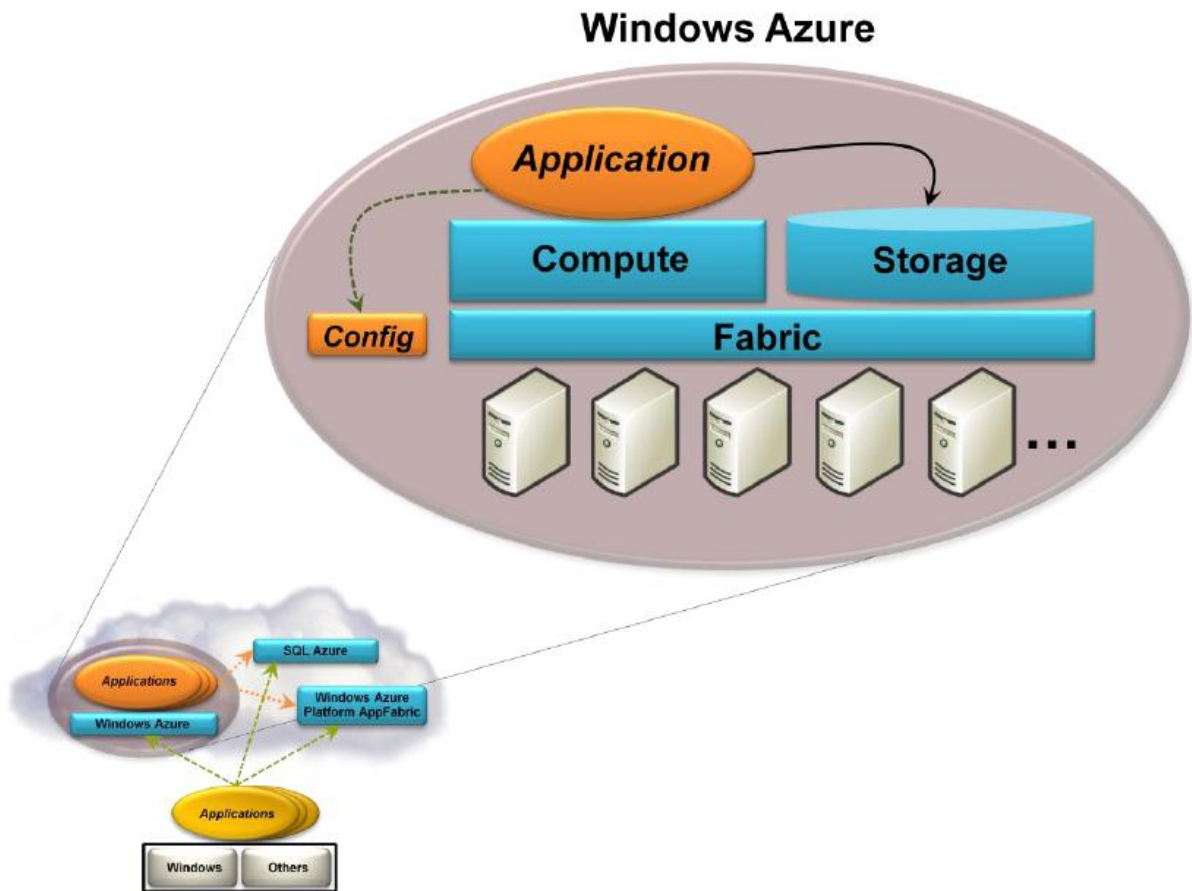
- Το περιβάλλον φιλοξενίας του Windows Azure ονομάζεται Fabric Controller – το οποίο δημιουργεί (provision) τους απαραίτητους πόρους και τις εικονικές μηχανές (virtual machines) σε ένα δίκτυο, αυτόματα διαχειρίζεται πόρους, διαχειρίζεται τον διαμοιρασμό φόρτου εργασίας (load balancing), αντιγράφει εικονικές μηχανές σε άλλους γεωγραφικούς τόπους (geo-replication) και τη διάρκεια ζωής μιας εφαρμογής (application lifecycle) χωρίς να απαιτείται από τις φιλοξενούμενες εφαρμογές να ασχολούνται με αυτές τις απαιτήσεις. Επιπροσθέτως, παρέχει και άλλες υπηρεσίες που απαιτούν οι περισσότερες εφαρμογές — όπως το Windows Azure Storage Service που δίνει στις εφαρμογές την ικανότητα να αποθηκεύσουν μη δομημένα δεδομένα όπως binary large objects, queues, drives και μη σχεσιακούς πίνακες. Οι εφαρμογές επίσης μπορούν να χρησιμοποιούν άλλες υπηρεσίες που είναι τμήμα του Azure Services Platform.[4]
- Το Azure επίσης προσφέρει Content Delivery Network (CDN) υπηρεσίες ως επιλογή. Το Azure CDN επιτρέπει παγκόσμια low-latency μεταφορά στατικού περιεχομένου από το Azure Storage στους τελικούς χρήστες από πολλά κέντρα εξυπηρητητών σε όλο τον κόσμο.
- SQL Azure: Η Microsoft SQL Azure Database είναι μια υπηρεσία σχεσιακής βάσης βασισμένη στο υπολογιστικό νέφος, βασισμένο σε Microsoft SQL Server τεχνολογίες. Παρέχει ένα υψηλής διαθεσιμότητας, επεκτάσιμο, σε πολλούς εξυπηρητητές (multi-tenant) υπηρεσία βάσης δεδομένων που φιλοξενείται από τη Microsoft στο σύννεφο. Η SQL Azure Database διευκολύνει τη δημιουργία (provisioning) και την διανομή (deployment) πολλών βάσεων δεδομένων. Οι προγραμματιστές δεν χρειάζεται να εγκαταστήσουν, να παραμετροποιήσουν και να ενημερώνουν κάποιο λογισμικό. Η υψηλή διαθεσιμότητα και η ανεκτικότητα σε λάθη είναι ενσωματωμένες, και δεν υπάρχει ανάγκη για φυσική διαχείριση. Η SQL Azure Database υποστηρίζει Transact-SQL (T-SQL), την ίδια γλώσσα δομημένων ερωτημάτων που υποστηρίζει το Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων Microsoft SQL Server. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν υπάρχουσα γνώση σε T-SQL ανάπτυξη και ένα γνώσιμο σχεσιακό μοντέλο δεδομένων, σε άμεση συσχέτιση με τον τρόπο που δουλεύουν σε ήδη υπάρχουσες βάσεις δεδομένων εγκατεστημένες σε κέντρα εξυπηρητητών. Επιπροσθέτως, οι προγραμματιστές μπορούν να χρησιμοποιήσουν O/R Mappers τους οποίους ήδη χρησιμοποιούν, για αν υποβάλλουν ερωτήματα στην SQL Azure Database (π.χ. Entity Framework,

nHibernate, Linq To SQL). Οι προγραμματιστές και οι διαχειριστές μπορούν να συγχρονίσουν δεδομένα από έναν SQL Server εγκατεστημένο σε τοπικό κέντρο εξυπηρητών με έναν SQL Azure (και αντιστρόφως) μέσω του SQL Azure Data Sync. [5]

- SQL Azure Data Sync. Κάνοντας χρήση του Microsoft Sync Framework 2.0 μαζί με το Microsoft Sync Framework Power Pack for SQL Azure November CTP οι προγραμματιστές μπορούν να:
  - Συνδέσουν υπάρχοντες SQL Servers στο SQL Azure
  - Δημιουργήσουν νέες εφαρμογές στο Windows Azure χωρίς να σταματήσουν να χρησιμοποιούν τις υπάρχουσες εφαρμογές εγκατεστημένες τοπικά
  - Μετατρέψουν τις εφαρμογές σε Windows Azure και σε SQL Azure ώστε να εκτελούνται και χωρίς σύνδεση στο δίκτυο. Έτσι, θα προσδώσουν ένα cached μοντέλο τύπου Microsoft Outlook [5]
- Windows Azure platform AppFabric: Το AppFabric παρέχει υπηρεσίες σύννεφου για τη σύνδεση εφαρμογών που τρέχουν στο σύννεφο ή τοπικά. Το Azure AppFabric αποτελείται από δύο τμήματα, το Access Control και το Service Bus.
  - Το Access Control δίνει τη δυνατότητα για τη δημιουργία κανόνων για αυθεντικοποίηση και εξουσιοδότηση.
  - Το Service Bus παρέχει πολλές λειτουργίες μηνυμάτων (messaging patterns) για REST, SOAP, και Windows Communication Foundation TCP πρωτόκολλο. [6]
- Microsoft DataMarket: Το Microsoft DataMarket είναι μία υπηρεσία που επιτρέπει σε προγραμματιστές και υπαλλήλους γραφείου να ανακαλύψουν, να προμηθευτούν και να χρησιμοποιήσουν υπάρχοντα σύνολα δεδομένων (datasets) μέσω της πλατφόρμας Windows Azure. [7]

### 3.2 Windows Azure

Στο πιο υψηλό επίπεδο, το Windows Azure είναι εύκολα κατανοητό: Είναι μια πλατφόρμα για την εκτέλεση Windows εφαρμογές και την αποθήκευση των δεδομένων στο υπολογιστικό νέφος. Το σχήμα 14 απεικονίζει τα βασικά συστατικά του.



ΣΧΗΜΑ 14 ΤΟ WINDOWS AZURE ΠΑΡΕΧΕΙ ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΔΥΝΑΤΟΤΗΤΕΣ ΚΑΙ ΧΩΡΟ ΑΠΟΘΗΚΕΥΣΗΣ ΒΑΣΙΣΜΕΝΟ ΣΕ WINDOWS ΓΙΑ ΕΦΑΡΜΟΓΕΣ ΣΥΝΝΕΦΟΥ [4]

Όπως προκύπτει και από την εικόνα, το Windows Azure τρέχει σε έναν πολύ μεγάλο αριθμό μηχανών, οι οποίες βρίσκονται σε Microsoft κέντρα εξυπηρητητών (σε όλο τον κόσμο) και είναι προσβάσιμες μέσω διαδικτύου.

Η Windows Azure υπολογιστική υπηρεσία (compute service) είναι βασισμένη στα Windows. Οι προγραμματιστές μπορούν να κατασκευάσουν εφαρμογές χρησιμοποιώντας το .NET Framework, μη διαχειρίσιμο (unmanaged) κώδικα, ή άλλες προσεγγίσεις. Αυτές οι εφαρμογές είναι γραμμένες σε συνηθισμένες γλώσσες για Windows, χρησιμοποιώντας διάφορα Integrated Development Environments (εργαλεία για προγραμματιστές).

Οι εφαρμογές Windows Azure και οι συνηθισμένες (desktop, web, mobile, etc.) εφαρμογές έχουν πρόσβαση στην Windows Azure υπηρεσία αποθήκευσης (storage service), κάνοντας χρήση μιας RESTful προσέγγισης. Αυτή η υπηρεσία επιτρέπει την αποθήκευση μεγάλων αρχείων (blobs), παρέχει ουρές μηνυμάτων για την επικοινωνία μεταξύ συστατικών των Windows Azure εφαρμογών, και παρέχει επίσης μια μορφή πινάκων με μία απλή γλώσσα ερωτημάτων (που μοιάζει με τις βάσεις NoSQL). Για εφαρμογές που χρειάζονται το παραδοσιακό σχεσιακό μοντέλο αποθήκευσης, η Windows Azure πλατφόρμα παρέχει SQL Azure βάση δεδομένων. Κάθε εφαρμογή που χρησιμοποιεί τη Windows Azure πλατφόρμα μπορεί να χρησιμοποιεί οποιονδήποτε συνδυασμό των παραπάνω.

Για την επίτευξη αυτών των στόχων απαιτείται αποτελεσματική διαχείριση. Στο Windows Azure, κάθε εφαρμογή έχει ένα αρχείο ρυθμίσεων (configuration file), γραμμένο σε απλή XML. Με την τροποποίηση αυτού του αρχείου χειροκίνητα ή αυτόματα (π.χ. από κάποια άλλη υπηρεσία), ο διαχειριστής μιας εφαρμογής μπορεί να ελέγξει διάφορες πλευρές της συμπεριφοράς της, όπως τον καθορισμό του αριθμού των instances της εφαρμογής (αριθμός εικονικών μηχανών) που θα πρέπει να τρέξει στο Windows Azure.

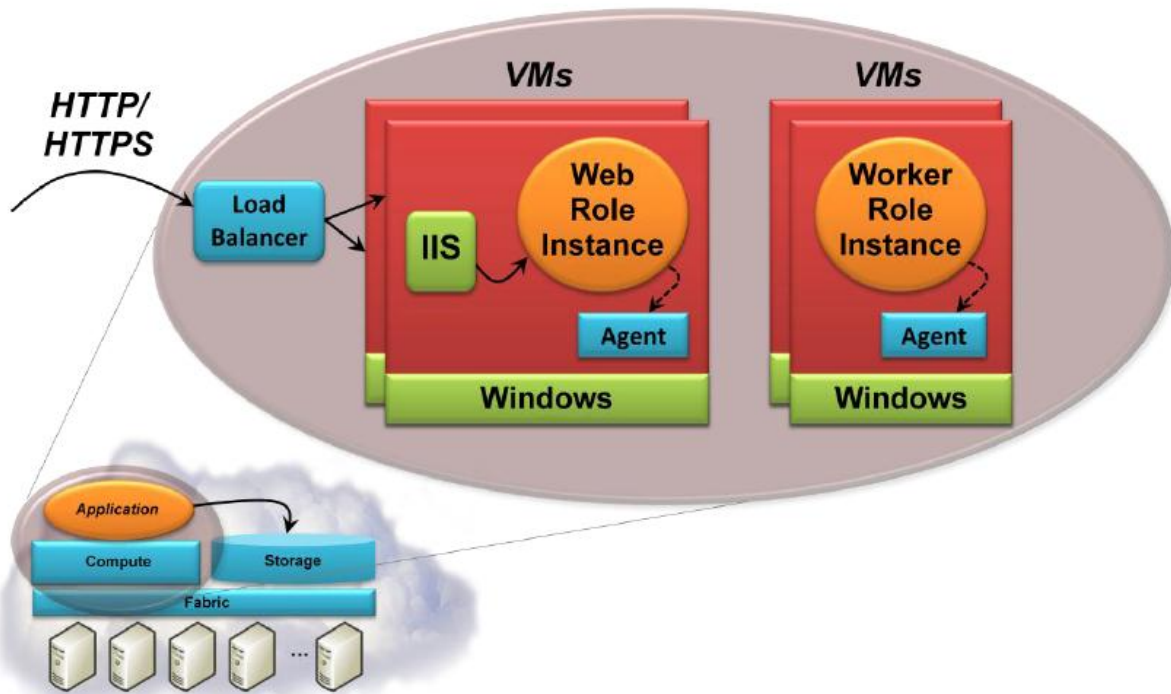
Για να επιτρέψει στους πελάτες του να δημιουργούν, να διαχειρίζονται και να καταγράφουν πληροφοριακά μηνύματα ή σφάλματα στις εφαρμογές τους, το Windows Azure παρέχει μία ιστοσελίδα προσβάσιμη από οποιονδήποτε φυλλομετρητή διαδικτύου (browser). Ο πελάτης δίνει ένα Windows Live ID, μετά επιλέγει εάν θέλει να δημιουργήσει ένα λογαριασμό φιλοξενίας για την εκτέλεση των εφαρμογών, ένα λογαριασμό αποθήκευσης για αποθήκευση δεδομένων, ή και τα δύο. [4,8]

### **3.3 Εκτέλεση εφαρμογών στο Windows Azure**

Στο Windows Azure, μία εφαρμογή τυπικά εκτελείται σε πολλές εικονικές μηχανές. Κάθε μία από αυτές τις μηχανές τρέχει στο δικό της λειτουργικό Windows Server 2008 R2. Αυτές οι εικονικές μηχανές (VMs) είναι ειδικά σχεδιασμένες για cloud εφαρμογές.

Ο προγραμματιστής δεν δημιουργεί αυτά τα VMs. Δεν ανεβάζει επίσης στο Azure μία εικόνα (image) ενός λειτουργικού συστήματος, οπότε δεν χρειάζεται να ανησυχεί για την συντήρησή του. Αντί αυτού, ο προγραμματιστής δημιουργεί εφαρμογές χρησιμοποιώντας Web roles και/ή Worker roles, και μετά δίνει στο Windows Azure την πληροφορία για το πόσα αντικείμενα (instances) του κάθε ρόλου θα

τρέξει. Το Windows Azure δημιουργεί ένα VM για κάθε αντικείμενο, και μετά τρέχει την εφαρμογή σε αυτά τα VMs. Το σχήμα 15 δείχνει αυτή τη διαδικασία.



ΣΧΗΜΑ 15 ΟΙ ΕΦΑΡΜΟΓΕΣ ΣΤΟ WINDOWS AZURE ΜΠΟΡΟΥΝ ΝΑ ΑΠΟΤΕΛΟΥΝΤΑΙ ΑΠΟ WEB ΡΟΛΟΥΣ ΚΑΙ WORKER ΡΟΛΟΥΣ, ΚΑΙ Ο ΤΟ ΚΑΘΕ ΑΝΤΙΚΕΙΜΕΝΟ ΤΟΥ ΡΟΛΟΥ ΝΑ ΤΡΕΧΕΙ ΣΤΗ ΔΙΚΙΑ ΤΟΥ ΕΙΚΟΝΙΚΗ ΜΗΧΑΝΗ [4]

Κάθε αντικείμενο Δικτυακού Ρόλου (Web Role) δέχεται εισερχόμενες HTTP (ή HTTPS, αν υπάρχει εγκατεστημένο πιστοποιητικό) κλήσεις. Για εξυπηρετητής διαδικτύου (web server) χρησιμοποιείται ο Internet Information Services (IIS) 7. Ένας Web role μπορεί να δημιουργηθεί χρησιμοποιώντας ASP.NET WebForms, ASP.NET MVC, WCF, απλή HTML, ή άλλες τεχνολογίες που λειτουργούν με τον εξυπηρετητή διαδικτύου της Microsoft, Internet Information Services (IIS). Το Windows Azure παρέχει εγκατεστημένο διαμοιρασμό φόρτου (built-in load balancing) ώστε να διαμοιράσει κλήσεις ανάμεσα σε αντικείμενα Web ρόλων που ανήκουν στην ίδια εφαρμογή (εφόσον ο developer έχει ενεργοποιήσει περισσότερα από ένα αντικείμενα).

Τα Web roles έχουν ως στόχο να δέχονται και να επεξεργάζονται αιτήσεις HTTP χρησιμοποιώντας τον IIS. Παρ' όλα αυτά, δεν είναι όλα Web εφαρμογές, και γι αυτό το Windows Azure παρέχει επίσης Worker roles. Ένα Worker role αντικείμενο είναι παρόμοιο με ένα Web role αντικείμενο, με τη βασική διαφορά ότι ένα Worker role δεν έχει τον IIS προεγκατεστημένο να τρέχει σε κάθε αντικείμενο, και έτσι, αντίθετα με τους Web roles, τα Worker role αντικείμενα δεν φιλοξενούνται στον IIS. Ένας Worker role όμως μπορεί να δέχεται HTTP requests από τον «έξω κόσμο», και οι προγραμματιστές μπορούν να τρέξουν ακόμα και έναν άλλο Web server, όπως ο Apache, σε ένα Worker role αντικείμενο. Επίσης, οι worker ρόλοι μπορούν να τρέξουν εκτελέσιμα αρχεία (executable files, .exe).

Τα Worker role αντικείμενα μπορούν να επικοινωνούν με Web role αντικείμενα με πολλούς τρόπους. Μία επιλογή είναι αν χρησιμοποιηθούν Windows Azure ουρές μηνυμάτων (π.χ. ένα Web role αντικείμενο μπορεί να εισάγει ένα αντικείμενο εργασίας (work item) σε μία ουρά μηνυμάτων, και ένα Worker role αντικείμενο μπορεί να το εξάγει και να το επεξεργαστεί). Μία εναλλακτική είναι οι Worker roles και οι Web roles να στήσουν απευθείας επικοινωνία μέσω της τεχνολογίας Windows Communication Foundation (TCP, HTTP κ.α.) ή μέσω μιας άλλης τεχνολογίας απομακρυσμένης σύνδεσης.

Είτε τρέχει ένα Web role αντικείμενο είτε ένα Worker role αντικείμενο, κάθε VM περιέχει επίσης έναν Windows Azure πράκτορα (agent) που επιτρέπει στη εφαρμογή να αλληλεπιδρά με το Windows Azure fabric, όπως φαίνεται στην Εικόνα 3. Αυτός ο πράκτορας έχει ένα καλώς ορισμένο API που επιτρέπει στο αντικείμενο να κάνει διάφορες λειτουργίες, όπως το να βρει τον αρχικό φάκελο για τοπική αποθήκευση (local storage) στο VM στο οποίο εκτελείται.

Τόσο για τους Web roles όσο και για τους Worker roles, το Windows Azure επιτρέπει στους προγραμματιστές να επιλέγουν ανάμεσα σε τέσσερις CPU επιλογές: 1 πυρήνα, 2 πυρήνες, 4 πυρήνες, και 8 πυρήνες. Εφόσον σε κάθε VM ανατίθενται ένας ή περισσότεροι πυρήνες, οι εφαρμογές έχουν προβλέψιμη απόδοση. Για την αύξηση της απόδοσης, ο ιδιοκτήτης μιας εφαρμογής μπορεί να αυξήσει τον αριθμό των αντικειμένων στα οποία τρέχει (αυτό ορίζεται στο αρχείο ρυθμίσεων της εφαρμογής). Ο προγραμματιστής μπορεί επίσης να προγραμματίσει την εφαρμογή να κλιμακώνεται αυτόματα. Σημαντική λεπτομέρεια είναι ότι το fabric αντιλαμβάνεται όταν ένα Web ή Worker role instance έχει αποτύχει, για να ξεκινήσει αυτόματα ένα νέο.

Κατά συνέπεια, για να είναι επεκτάσιμη, τα Windows Azure Web role αντικείμενα δεν πρέπει να αποθηκεύουν κατάσταση (stateless). Κάθε κατάσταση σχετική με τον πελάτη (client-specific state) θα πρέπει να είναι καταγεγραμμένη σε κάποιο χώρο μόνιμης αποθήκευσης (persistent) (πιθανές επιλογές περιλαμβάνουν Windows Azure storage ή βάση δεδομένων SQL Azure ή απλά να αποθηκεύονται σε ένα cookie στον φυλλομετρητή του χρήστη). Επιπροσθέτως, το ενσωματωμένο σύστημα διαμοιρασμού φόρτου του Windows Azure υποχρεώνει τις εφαρμογές να μην αποθηκεύουν κατάσταση.

Οι Web roles και οι Worker roles έχουν υλοποιηθεί με την χρήση Windows τεχνολογιών, καθώς η υποδομή του Azure βασίζεται στο λειτουργικό σύστημα Windows. Παρ όλα αυτά, η μετακίνηση ήδη υπάρχοντων εφαρμογών στο Windows Azure πιθανότατα θα απαιτήσει κάποιες μικρές αλλαγές. Για παράδειγμα, οι Windows Azure εφαρμογές σήμερα μπορούν να τρέξουν μόνο σε λειτουργία χρήστη—σε λειτουργία διαχειριστή δεν επιτρέπεται, κυρίως για λόγους σταθερότητας. Σε μεγάλο βαθμό, όμως, η εκτέλεση εφαρμογών στο Windows Azure και τοπικά (on Windows Server 2008) δεν παρουσιάζει μεγάλες διαφορές.

Για τους προγραμματιστές, η κατασκευή μιας Windows Azure εφαρμογής μοιάζει επίσης πολύ με την κατασκευή μιας παραδοσιακής ASP.NET και/ή Windows εφαρμογής. Το Visual Studio παρέχει έτοιμο κώδικα (project templates) για την κατασκευή Windows Azure Web roles, Worker roles, ή συνδυασμό των δύο παραπάνω, και οι προγραμματιστές μπορούν να χρησιμοποιήσουν οποιαδήποτε γλώσσα προγραμματισμού για τα Windows. Επιπροσθέτως, το Windows Azure software development kit περιέχει μια έκδοση του Windows Azure περιβάλλοντος που εκτελείται στο μηχάνημα του προγραμματιστή. Γνωστό και ως Windows Azure Development Fabric, περιλαμβάνει ένα Windows Azure storage εξομοιωτή, ένα Windows Azure πράκτορα, και οτιδήποτε άλλο «βλέπει» μια εφαρμογή που τρέχει στο σύννεφο. Το development fabric είναι στην πραγματικότητα μία (πολύ μικρή) έκδοση του Azure, η οποία τρέχει στον υπολογιστή του προγραμματιστή. Κάθε προγραμματιστής μπορεί να δημιουργήσει και να αποσφαλματώσει μια εφαρμογή χρησιμοποιώντας το Visual Studio, μετά να ανεβάσει την εφαρμογή στους δοκιμαστικούς (staging) και παραγωγικούς (production) εξυπηρετητές του Windows Azure όταν είναι έτοιμη. Βέβαια, κάποια πράγματα είναι αρκετά διαφορετικά στο σύννεφο, σε σχέση με την εκτέλεση εφαρμογών στον υπολογιστή του προγραμματιστή. Προφανώς, δεν είναι δυνατόν να γίνει αποσφαλμάτωση με αποσφαλματωτή (debugger) στο λειτουργικό σύστημα του σύννεφου, οπότε η αποσφαλμάτωση βασίζεται σε καταγραφή συμβάντων. Το Windows Azure παρέχει



επίσης και άλλες υπηρεσίες για προγραμματιστές, όπως πληροφορίες για την χρήση του επεξεργαστή μιας εφαρμογής που εκτελείται, εισερχόμενο και εξερχόμενο μέγεθος πληροφορίας (bandwidth), και αποθήκευση. Οι προγραμματιστές μπορούν να υποβάλλουν ερωτήματα για αυτές τις πληροφορίες και να προσαρμόζουν ανάλογα διάφορες πτυχές της εφαρμογής. [8]

## 4. Ηλεκτρονικός Φάκελος Ασθενών

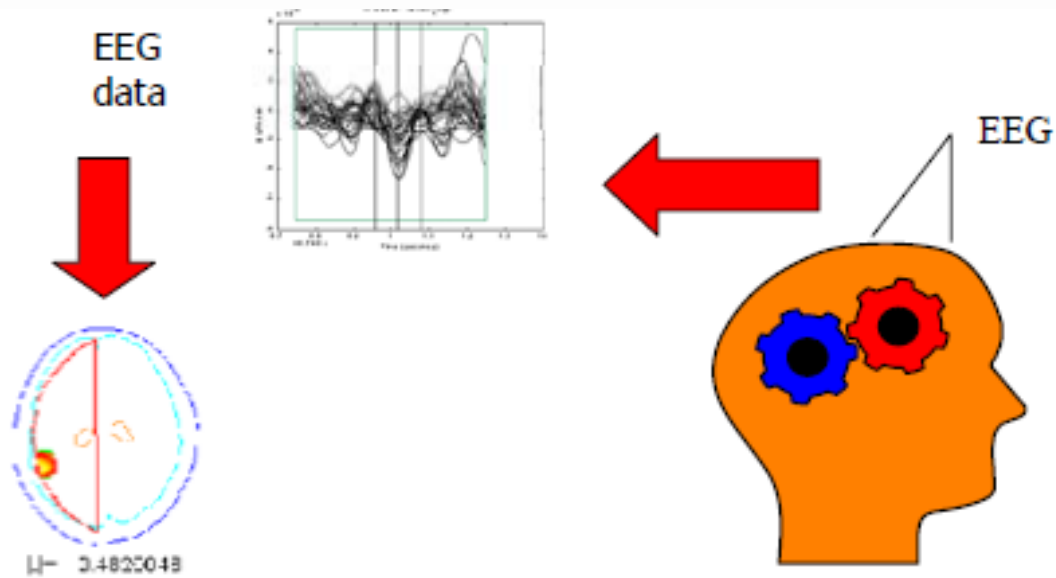
---

Στο κεφάλαιο αυτό γίνεται η εισαγωγή στην έννοια, τη δομή και τη σημασία του Ηλεκτρονικού Φακέλου Ασθενών (ΗΦΑ). Θα αναλύσουμε τις βασικές έννοιες της δικτύωσης του ΗΦΑ και της διαχείρισης της πληροφορίας του, τις βασικότερες λειτουργίες επί του ΗΦΑ, και των πληροφοριακών συστημάτων ενός νοσοκομείου. Εν συνεχεία, θα μιλήσουμε για την έννοια της ολοκλήρωσης, τα ολοκληρωμένα πληροφοριακά συστήματα υγείας (ΟΠΣΥ) και το ποιοι παράγοντες οδηγούν στην ανάγκη για αυτά.

### 4.1 Ορισμοί

Ξεκινώντας την ανάλυση του όρου «Ηλεκτρονικός Φάκελος Ασθενούς» θα περιγράψουμε τον όρο «Ιατρική Πληροφορία» και «Ιατρικό δεδομένο». Ως Ιατρικό Δεδομένο ορίζουμε οποιαδήποτε απλή παρατήρηση σε έναν ασθενή (π.χ. μέτρηση της πίεσης, της θερμοκρασίας κ.α.). Ιατρική Πληροφορία είναι η οποιαδήποτε γνώση η οποία έχει αποκτηθεί από γεγονότα και δεδομένα. Η αξία της πληροφορίας καθορίζεται από το άτομο ή τον οργανισμό ο οποίος την κατέχει, αλλά και με τον τρόπο με τον οποίο αποκτήθηκε. Συνεπώς, μπορεί να είναι ανεκτίμητη ή και χωρίς καμία αξία. Για παράδειγμα, μια εξέταση αίματος από την μία είναι ζωτικής σημασίας για έναν ιατρό και από την άλλη για το προσωπικό της αποθήκης αυτό σημαίνει μόνο κατανάλωση κάποιων υλικών. Η πληροφορία θεωρείται περιουσία και πάγιο, καθώς αναλώνεται πολύτιμος χρόνος, κόπος και (συνεπώς) χρήμα για τη απόκτησή της. Οπότε, ως Ιατρική Πληροφορία ορίζουμε την επεξεργασία ιατρικών δεδομένων. [9]

Τώρα μπορούμε να δούμε τη σχέση μεταξύ των δεδομένων, της γνώσης και της πληροφορίας.



ΣΧΗΜΑ 16 Η ΕΠΕΞΕΡΓΑΣΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΟΔΗΓΕΙ ΣΕ ΔΕΔΟΜΕΝΑ [12]

Είναι γνωστό ότι ελλιπείς πληροφορίες οδηγούν σε λάθος αποφάσεις, στην ιατρική όμως μια λάθος απόφαση σημαίνει επέμβαση στη ζωή κάποιου ασθενούς. Είναι ευνόητο, λοιπόν, ότι η Ιατρική Πληροφορία και η ορθότητά της είναι μείζονος σημασίας.

Στην Υγεία η πληρότητα και η διαθεσιμότητα των δεδομένων και της πληροφορίας είναι ζωτικής σημασίας. Ημιτελής πληροφορία μπορεί να οδηγήσει σε κακή διάγνωση και θεραπεία, σε σπατάλη χρημάτων και πόρων και σε καταστάσεις οι οποίες απειλούν τη ζωή. Καταλαβαίνουμε λοιπόν πόση αξία έχει η Ιατρική Πληροφορία ως προς την εγκυρότητα της, την ακεραιότητά της και την διαφύλαξή της.

Είναι κοινή διαπίστωση ότι ο όγκος των πληροφοριών που σχετίζονται με την φροντίδα του ασθενούς έχει αυξηθεί κατά πολύ τα τελευταία χρόνια, πράγμα που σε μεγάλο βαθμό οφείλεται στην ενσωμάτωση αυξημένου αριθμού εργαστηριακών και παρακλινικών εξετάσεων στους φακέλους των ασθενών. Ακόμη, τα διαχειριστικά καθήκοντα των ιατρών γίνονται διαρκώς περισσότερα, καθώς η πολυπλοκότητα των ιδρυμάτων παροχής υπηρεσιών υγείας αυξάνει. [10]

Οι φάκελοι ασθενών που είναι βασισμένοι σε χαρτί είναι ακόμα και τώρα (2011) η πιο συνηθισμένη μέθοδος για καταγραφή ιατρικής πληροφορίας. Η πλειοψηφία των γιατρών βρίσκει ακόμα ότι η

καταγραφή πληροφορίας στο χαρτί είναι η πιο εύκολη μέθοδος. Παρόλα αυτά, όσο εύκολη και να είναι η συμπλήρωση για το γιατρό, αυτοί οι φάκελοι απαιτούν μεγάλο φυσικό όγκο για την αποθήκευσή τους. Επίσης, το κόστος της αποθήκευσης αλλά και του φιλμ (ακτινογραφίες) διαφέρει δραματικά από το κόστος της ηλεκτρονικής αποθήκευσης. Ακόμη, όταν χάρτινοι φάκελοι ασθενούς αποθηκεύονται σε διαφορετικούς χώρους, η ένωσή τους σε μία κοινή τοποθεσία είναι εξαιρετικά χρονοβόρα. Εξαιτίας των παραπάνω, πολλά ασφαλιστικά ταμεία, κυβερνήσεις, νοσοκομειακοί οργανισμοί ανά τον κόσμο προωθούν τη χρήση των ηλεκτρονικών φακέλων.

Πριν δώσουμε τον ορισμό του «Ηλεκτρονικού Φακέλου Ασθενών», θα αναφέρουμε τι εννοούμε όταν μιλάμε για βάσεις δεδομένων στην Ιατρική. Οι ιατρικές βάσεις δεδομένων

- Παράγουν πληροφορίες με βάση πρότυπα και αλγόριθμους
- Συλλέγουν πληροφορίες
- Ταξινομούν και οργανώνουν πληροφορίες
- Αποθηκεύουν πληροφορίες
- Επεξεργάζονται πληροφορίες
- Δεσμεύουν πληροφορίες (όσον αφορά την προσπέλαση)
- Διασφαλίζουν πληροφορίες
- Διαχειρίζονται και αποδίδουν πληροφορίες προς χρήση

Παρακάτω δίνουμε έναν ορισμό του Ηλεκτρονικού Φακέλου

«είναι ένα σύστημα σχεδιασμένο έτσι ώστε να υποστηρίζει την απόλυτη διαθεσιμότητα και την ακρίβεια ιατρικών ή άλλων πληροφοριών με σκοπό την παροχή ιατρικής περίθαλψης». [11]

Γενικά, ο ηλεκτρονικός φάκελος ασθενούς είναι μία έννοια που διαρκώς εξελίσσεται και προσδιορίζεται ως μια συλλογή από ιατρικές πληροφορίες ιδιωτών ή πληθυσμών. Αποθηκεύεται σε ψηφιακή μορφή, μπορεί να διαμοιράζεται μεταξύ διαφορετικών ιατρικών λογισμικών και είναι δυνατό να μεταφέρεται μέσω του δικτύου. Ένας ηλεκτρονικός φάκελος μπορεί να περιλαμβάνει μεγάλο πλήθος δεδομένων σε πλήρη ή περιληπτική μορφή, όπως δημογραφικά στοιχεία, ιατρικό ιστορικό,

φαρμακευτική αγωγή και αλλεργίες, προσωπικά στοιχεία όπως βάρος και ύψος αλλά και λογιστικές πληροφορίες.

## 4.2 Χαρακτηριστικά

Τα λογισμικά Ηλεκτρονικού Φάκελου Υγείας αποτελούν συστήματα διαχείρισης ιατρικών φακέλων που βασίζονται σε ηλεκτρονικούς υπολογιστές. Ως εκ τούτου, η αποθήκευση και ανάκληση των δεδομένων γίνεται γρήγορα και με ασφάλεια. Επιπλέον, καθίσταται δυνατή η επεξεργασία των δεδομένων και η άμεση μεταφορά τους με ηλεκτρονικά μέσα, σε οποιαδήποτε απόσταση. Το σύστημα καταγραφής των δεδομένων που σχετίζεται με τους ασθενείς γίνεται τώρα πιο αποτελεσματικό, χάρη στη χρήση νέων τεχνολογιών, αλλά και εμπλουτίζεται εκμεταλλευόμενο τις δυνατότητες της νέας τεχνολογίας.

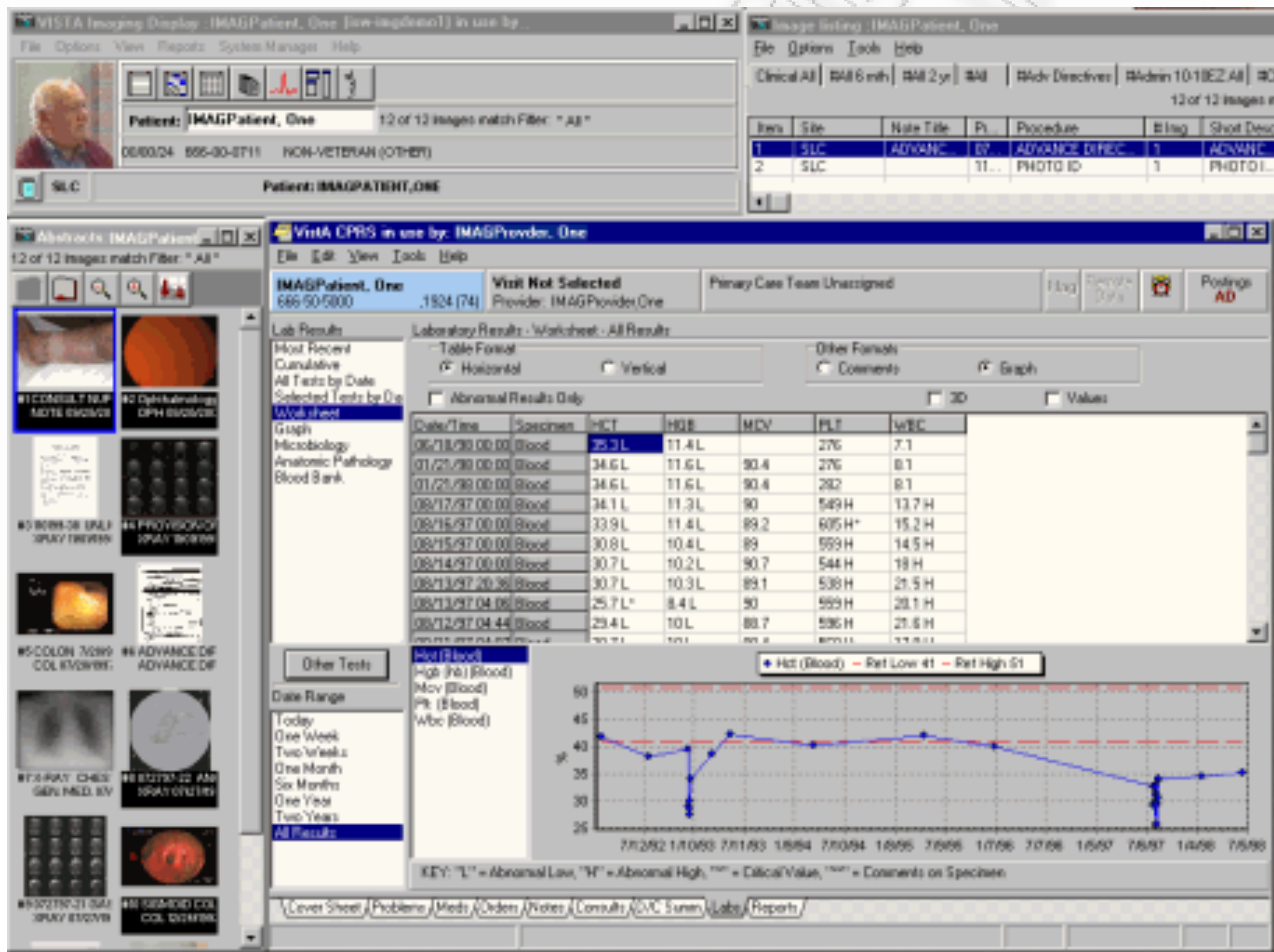
Ο ΗΦΑ πρέπει να περιέχει όλα τα δεδομένα που σχετίζονται με αυτόν, άσχετα με την μορφή στην οποία βρίσκονται.

- Το ιστορικό, η κλινική εξέταση και τα αποτελέσματα εργαστηριακών εξετάσεων, βρίσκονται σε μορφή κειμένου
- Οι διάφορες εξετάσεις (ακτινογραφίες, τομογραφίες – αξονικές, μαγνητικές, απλές – υπέρηχοι κ.α.) βρίσκονται σε μορφή στατικών εικόνων
- Τα ηλεκτροκαρδιογραφήματα βρίσκονται σε μορφή βιοσημάτων (bio-signals, ηλεκτρονικά κωδικοποιημένα έξοδα κάποιας καταγραφικής συσκευής)
- Τα αποτελέσματα των ενδοσκοπικών εξετάσεων (γαστροσκόπηση, κολonosκόπηση κ.α. βρίσκονται σε μορφή βίντεο)
- Το ηχοκαρδιογράφημα είναι σε μορφή ήχου

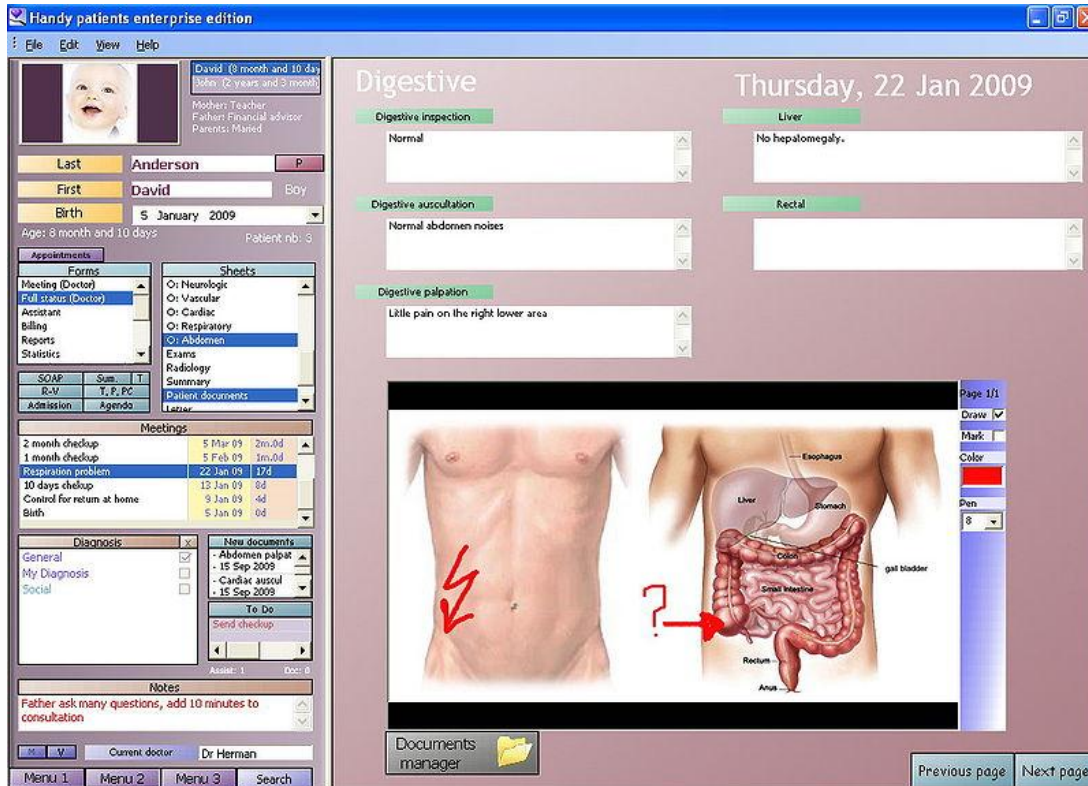
Η συνήθης τακτική είναι να συνοδεύουν τον φάκελο του ασθενούς και οι αντίστοιχες εξετάσεις, στην μορφή με την οποία παράγονται από το εργαστήριο (ακτινογραφικό φιλμ, έντυπα με αποτελέσματα βιοχημικών εξετάσεων, χαρτί ηλεκτροκαρδιογράφου κ.α.).

Έτσι, ο φάκελος καθίσταται ογκώδης, η πιθανότητα να χαθούν δεδομένα μεγαλύτερη, ενώ η χρονική συσχέτιση των διαφόρων εξετάσεων με το ιστορικό και την κλινική εξέταση δεν γίνεται άμεσα προφανής.

Ένα σύστημα ΗΦΑ είναι το πληροφοριακό σύστημα που διαχειρίζεται και δουλεύει σε ένα σύνολο από ΗΦΑ και βοηθά στην πρόσβαση και την κοινή χρησιμοποίηση της υπάρχουσας πληροφορίας σε διαπιστευμένο προσωπικό με φιλικό τρόπο. Για να μπορεί να χρησιμοποιηθεί σήμερα, θα πρέπει να είναι ασφαλές σύστημα και να εξασφαλίζει την «εμπιστευτική» υφή των δεδομένων. Σε ένα σύστημα ΗΦΑ, όλα τα δεδομένα ενσωματώνονται στον φάκελο του ασθενούς χωρίς να παίζει σημαντικό ρόλο η μορφή τους. Σε διάφορα σημεία του κειμένου του ιστορικού και της κλινικής εξετάσεως ενσωματώνονται ακτινολογικές ή βιοχημικές εξετάσεις, πράγμα που κάνει αμέσως εμφανή την συσχέτιση των εν λόγω εξετάσεων με την γενικότερη κατάσταση του ασθενούς. [10]



ΣΧΗΜΑ 17 ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΑΣΘΕΝΩΝ [12]



ΣΧΗΜΑ 18 ΠΑΡΑΔΕΙΓΜΑ ΕΦΑΡΜΟΓΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΑΣΘΕΝΩΝ [12]

#### 4.2.1 Πλεονεκτήματα

Μερικά από τα βασικά πλεονεκτήματα της χρησιμοποίησης ενός συστήματος ηλεκτρονικού ιατρικού φακέλου είναι:

- Μείωση κόστους λειτουργίας νοσηλευτικών ιδρυμάτων
- Μείωση χρήσης έντυπου ιατρικού φακέλου
- Έλεγχος της λανθασμένης φαρμακευτικής συνταγογράφησης
- Μείωση επανάληψης ιατρικών εξετάσεων ως και 30%
- Κέρδος χρόνου στην αναζήτηση και συμπλήρωση στοιχείων ιατρικού φακέλου
- Δυνατότητα πολλαπλής ανάγνωσης περιστατικού. Τηλεϊατρική
- Επιδημιολογικές μελέτες
- Βελτίωση ποιότητας ιατρικής περίθαλψης
- Εξ'αποστάσεως πρόσβαση σε στοιχεία ιατρικού φακέλου

- Ευκολία εκμάθησης των λογισμικών
- Φιλικότητα του γραφικού περιβάλλοντος
- Προώθηση της ιατρικής βασισμένης σε αποδείξεις (evidence-based medicine)

Είναι γεγονός πως ο ηλεκτρονικός ιατρικός φάκελος διευκολύνει την καθημερινή δουλειά των ιατρών και των νοσοκόμων αλλά και την καθημερινή ζωή των ασθενών. Ακολουθούν μερικά παραδείγματα χρήσεων του ΗΦΑ, τα οποία μπορούν να διευκολύνουν τον ιατρό στην καθημερινή του εργασία. Παράλληλα, αρκετά (αν όχι όλα) από τα παρακάτω θα πρέπει να υλοποιούνται σε ένα σύστημα ΗΦΑ, ώστε να είναι επιτυχημένο.

- **Εύκολη διαχείριση φακέλων ασθενών**
- **Ευχρηστία αποθήκευσης, ενημέρωσης και πλοήγησης στις ιατρικές πληροφορίες**
- **Παρουσίαση δεδομένων με απλό και κατανοητό τρόπο**
- **Εύκολη και εύχρηστη αναζήτηση**
- **Ανάλυση πληθυσμού (Population Analysis):** Μπορούμε να βρούμε όσους ασθενείς πληρούν ορισμένες προϋποθέσεις, π.χ. όσους πρέπει να κάνουν συγκεκριμένο εμβόλιο έτσι ώστε να ειδοποιηθούν να προσέλθουν με τα απαραίτητα παραστατικά.
- **Ειδικά Έγγραφα (Documents):** Ο φάκελος συνδέεται με έγγραφα που μπορούν να έχουν χρησιμότητα στην καθημερινή ζωή. Τα έγγραφα είναι γενικά πρότυπα (templates) και παίρνουν συγκεκριμένη μορφή ανάλογα με τον τρέχοντα κάθε φορά ασθενή. Παραδείγματα τέτοιων εγγράφων είναι τα ακόλουθα:
  - Παραπεμπτικό εξετάσεων
  - Διακομιστήριο
  - Πρόσκληση για διενέργεια εμβολιασμών
  - Πρόσκληση για διενέργεια μαστογραφίας
- **Βάση Φαρμάκων και κωδικοποίηση αυτών:** Κάθε φάρμακο μπορεί να καταχωρείται στη βάση φαρμάκων με συγκεκριμένη κωδικοποίηση, ώστε να είναι δυνατή η επεξεργασία στοιχείων που αφορούν τη χορήγησή τους
- **Καθημερινή παρακολούθηση της πορείας της υγείας του ασθενή αλλά και συστήματα υποβοήθησης διάγνωσης:** Πρόκειται για «έξυπνα» προγράμματα, τα οποία «διαβάζουν» και επεξεργάζονται τα δεδομένα ενός ή περισσότερων φακέλων, προκειμένου να



υποβοηθήσουν τον ιατρό στο έργο του. Η υποβοήθηση αυτή συνιστάται, συνήθως, στην παροχή οδηγιών: εφιστούν την προσοχή σε σημαντικές παραμέτρους του ασθενούς, εντοπίζουν αλληλεπιδράσεις φαρμάκων, συνιστούν την καλύτερη, σε κάθε περίπτωση, θεραπεία.

- **Τηλεϊατρική:** Ο ΗΦΑ ενός ασθενούς μπορεί να μεταφερθεί μέσω απλών τηλεφωνικών γραμμών, δορυφορικών ζεύξεων, γραμμών ISDN και ADSL ή οποιουδήποτε άλλου επικοινωνιακού μέσου, σε οποιαδήποτε απόσταση. Μπορεί, έτσι, ο ιατρός να ενημερώσει σχετικά με κάποιον ασθενή οποιονδήποτε συναδέλφο του διαθέτει συμβατό λογισμικό ΗΙΦ. Ως εκ τούτου, μπορεί να ζητήσει τη συμβουλή του συναδέλφου του όσον αφορά στο συγκεκριμένο περιστατικό. Προκειμένου να καταστεί δυνατό όλα τα λογισμικά ΗΙΦ να είναι συμβατά μεταξύ τους, έτσι ώστε όλοι οι ιατροί να μπορούν να ανταλλάξουν τους φακέλους των ασθενών τους, αναπτύσσονται πρότυπες αρχιτεκτονικές Ηλεκτρονικών Ιατρικών Φακέλων, σε επίπεδο Ευρωπαϊκό αλλά και παγκόσμιο.

Ένας τυπικός φάκελος υγείας θα πρέπει να περιλαμβάνει:

- Ταυτοποίηση των δεδομένων και ξεχωριστή ταξινόμηση των αρχείων
- Διαχωρισμό των εξετάσεων (επίσκεψη, περίθαλψη, χειρουργική επέμβαση, ακτινολογική κ.α.)
- Χρονολογική ταξινόμηση
- Γεγονότα
- Κλινικά δεδομένα
- Εργαστηριακά αποτελέσματα
- Διερευνήσεις
- Σήματα, εικόνες και video
- Αποφάσεις και διαγνώσεις
- Επεμβάσεις, θεραπείες και συνταγές
- Συσχετισμούς

#### 4.2.2 Προβλήματα – μειονεκτήματα

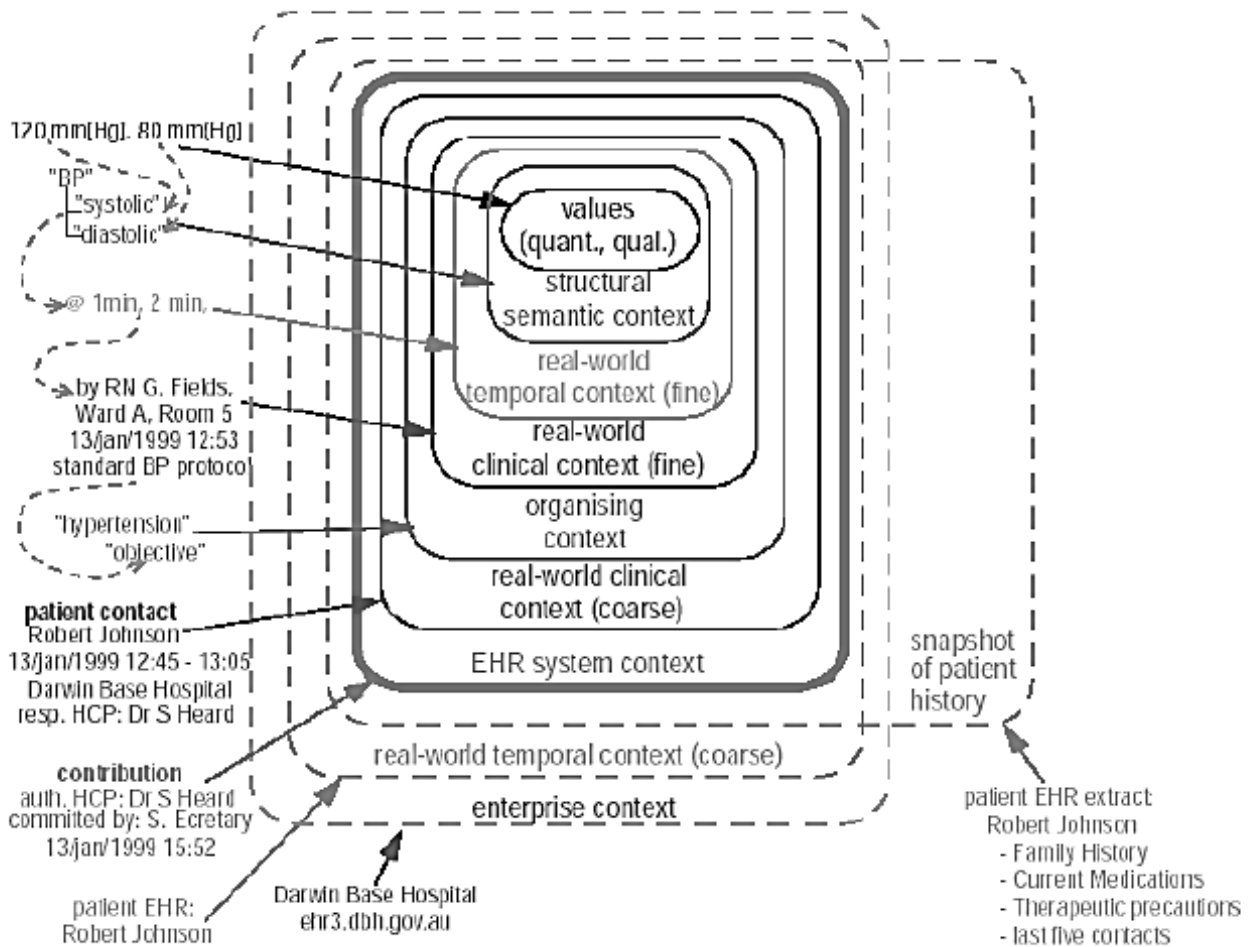
Τα προβλήματα του ηλεκτρονικού φακέλου ασθενούς προκύπτουν λόγω έλλειψης εκπαίδευσης, υψηλού κόστους και επιπλοκές στον τομέα της ασφάλειας των δεδομένων. Συνοπτικά:

- Δεν υπάρχουν διαθέσιμα ομοιογενή μητρώα ασθενών
- Χρήση πολύπλοκων interfaces
- Υπάρχουν θέματα προστασίας των δεδομένων
- Ανάγκη για στρατηγικό σχέδιο ασφάλειας
- Υπάρχουν ανοικτά ηθικά και νομικά ζητήματα
- Ανάγκη για συμφωνία των εμπλεκόμενων φορέων
- Θεσμικό Πλαίσιο
- Η μεγάλη ποικιλία πρωτοκόλλων βάσεων δεδομένων
- Η αντίσταση της πλειοψηφίας των γιατρών στις νέες τεχνολογίες
- Απαιτεί εκπαιδευμένο προσωπικό
- Υψηλότερο κόστος
- Έλλειψη παιδείας των εμπλεκόμενων φορέων σε θέματα πληροφορικής
- Ακριβές λύσεις και χρονοβόρες, όσον αφορά την υλοποίηση
- Ασυμβατότητα ιατρικής κουλτούρας σε διεθνές επίπεδο

Γενικά, κάποιοι κριτικοί αναφέρουν ότι εν τέλει, η χρήση ΗΦΑ μπορεί να μην είναι οικονομικά συμφέρουσα για έναν οργανισμό/κράτος. Επίσης, κάποιοι γιατροί πιστεύουν ότι βραχυπρόθεσμα δεν υπάρχουν τόσα πολλά οφέλη από τη χρήση ΗΦΑ καθώς απαιτείται εκμάθηση των λογισμικών και αλλαγή του καθημερινού τρόπου λειτουργίας της εργασίας τους.

#### **4.2.3 Η δομή των δεδομένων στον ΗΦΑ σε σχέση με τον μη ηλεκτρονικό φάκελο**

Ένας τύπος δεδομένων με μεγάλη σημασία στον ΗΦΑ είναι το «γενικό πλαίσιο/σύνολο περιστάσεων» στο οποίο αναφέρεται. Κάθε είδος γνώσης στον ΗΦΑ αποκτά έννοια μέσα από τα συμφραζόμενα του χρόνου, τοποθεσίας και των ιδιαιτεροτήτων των κλινικών και των ασθενών κατά τη διάρκεια της φροντίδας, καθώς δεν θα είχε νόημα. Δηλαδή, για κάθε δεδομένο θα πρέπει να υπάρχει η πληροφορία του ποιος/τι/πού/πότε/γιατί έκανε την αντίστοιχη καταχώρηση.



ΣΧΗΜΑ 19 ΔΟΜΗ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ ΑΣΘΕΝΟΥΣ [12]

Σύνθετα δεδομένα που μπορούν να υπάρξουν μέσα σε ένα ΗΦΑ είναι:

- Παραγγελίες εξετάσεων/παραπεμπτικά και παθολογία
- Ζωτικά σήματα
- Τομογραφίες, ηλεκτροφυσιολογικά και άλλα πολυμεσικά (multimedia) δεδομένα
- Δημογραφικά στοιχεία
- Οδηγίες/κλινικά πρωτόκολλα

Σε ένα σύστημα ΗΦΑ απαιτείται και κατάλληλη διαχείριση των εισόδων των πληροφοριών των ασθενών καθώς αυτά μπορεί να προέρχονται από αρκετές πηγές διαφορετικών γνωστικών περιοχών/αντικειμένων, όπως:

- Κλινικοί ιατροί
- Άλλοι επαγγελματίες υγείας (κοινωνικοί λειτουργοί, νοσηλευτές)
- Εισερχόμενα δεδομένα από εργαστηριακό εξοπλισμό
- Ασθενείς

Βέβαια, σε κατανεμημένα πληροφοριακά συστήματα η έννοια του ΗΦΑ αποκτά μια νέα διάσταση: δεν έχει νόημα του τι περιέχεται σε ένα φάκελο και τι όχι, αφού «τα πάντα υπάρχουν παντού» στο πληροφοριακό περιβάλλον του συστήματος υγείας. Συνεπώς, πολλοί ασθενείς έχουν πληροφορίες φακέλων σε πολλαπλά σημεία. Επίσης, τμήματα του ΗΦΑ μπορεί να χρειάζονται από διαφορετικούς χρήστες σε διαφορετικές στιγμές.

Μέχρι στιγμής, πάντως,

- Ο παραδοσιακός έντυπος φάκελος δεν έπαψε να υπάρχει
- Η εισαγωγή στοιχείων στο σύστημα γίνεται έπειτα από την συμπλήρωση της χάρτινης καρτέλας
- Η αναζήτηση της ιατρικής πληροφορίας σε ΗΦΑ επιτυγχάνεται στον ελάχιστο δυνατό χρόνο
- Ανάλυση στοιχείων και παρακολούθηση της κίνησης της κλινικής γίνεται πλέον σε ημερήσιο επίπεδο [12]

### 4.3 Λειτουργίες Διαχείρισης

Οι λειτουργίες διαχείρισης της ιατρικής πληροφορίας που σχετίζονται με ΗΦΑ αφορούν:

- Λήψη δεδομένων
  - o Η πληροφορία που λαμβάνεται από τον ασθενή
- Αποθήκευση
  - o Η αποθήκευση των δεδομένων που λαμβάνεται από τον ασθενή
- Επεξεργασίας πληροφορίας
  - o Η επεξεργασία των ήδη αποθηκευμένων δεδομένων από τον ιατρό του ασθενούς
- Επικοινωνία πληροφορίας

- Η μεταφορά της πληροφορίας είτε μέσα στον ίδιο ιατρικό οργανισμό (π.χ. νοσοκομείο) είτε μεταξύ διαφορετικών οργανισμών (π.χ. υγειονομικές περιφέρειες)
- Παρουσίαση πληροφορίας
  - Ο τρόπος με τον οποίο παρουσιάζεται η πληροφορία (Διεπαφή χρήστη – Graphical User Interface)
- Ασφάλεια
  - Τα ποικίλα στρώματα ασφαλείας που πρέπει να χρησιμοποιούνται για τη διασφάλιση των ευαίσθητων πληροφοριών του ασθενούς

Πιο αναλυτικά:

#### - **Λήψη δεδομένων**

Η πηγή των δεδομένων μπορεί να είναι:

- Κοντά ή μακριά από συσκευές παρακολούθησης
- Από εφαρμογές τηλεϊατρικής
- Κατευθείαν από τον ασθενή

Τα δεδομένα εισάγονται με πληκτρολόγιο, με κινητό τηλέφωνο, με επιτραπέζιο υπολογιστή (tablet), με αναγνώριση φωνής, γραφής, ή βιολογικών χαρακτηριστικών και μετάδοση από μια συσκευή.

#### - **Αποθήκευση**

Η αποθήκευση αναφέρεται στην φυσική τοποθεσία των δεδομένων. Στους ΗΦΑ τα δεδομένα μπορεί να είναι κατανεμημένα σε πολλαπλά συστήματα σε διαφορετικά μέρη. Εξαιτίας αυτού υπάρχει ανάγκη για κοινά πρωτόκολλα πρόσβασης, προγράμματα και ταυτοποίηση στοιχείων.

#### - **Επεξεργασία**

Η επεξεργασία αναφέρεται στην εφαρμογή συναρτήσεων για αποτελεσματική ανεύρεση και επεξεργασία δεδομένων ώστε να ληφθεί η πληροφορία από τα δεδομένα, ή να εξαχθεί γνώση. Περιέχονται εδώ εργαλεία στήριξης απόφασης, π.χ. “alerts” και “alarms”.

#### - **Επικοινωνία πληροφορίας**

Εδώ, αναφερόμαστε στη διαλειτουργικότητα των συστημάτων για ανταλλαγή δεδομένων ανάμεσα σε διαφορετικά εν γένει συστήματα και πλατφόρμες. Επίσης, η επικοινωνία σχετίζεται και με την Ηλεκτρονικός φάκελος ασθενών με χρήση XML Web Services και BPEL

μεταφορά των δεδομένων είτε μέσα σε μία υγειονομική μονάδα είτε μέσα σε έναν υγειονομικό οργανισμό (π.χ. το Εθνικό Σύστημα Υγείας στην Ελλάδα) καθώς και με τον τρόπο που αυτή μεταδίδεται (ασύρματα, ενσύρματα).

- **Παρουσίαση πληροφορίας**

Εξουσιοδοτημένοι πάροχοι φροντίδας και άλλοι με νόμιμες χρήσεις (π.χ. Διοικητικό Προσωπικό) έχουν την πληροφορία σε μορφή που τους εξυπηρετεί. Οι πάροχοι φροντίδας μπορεί να θέλουν να έχουν μια περιορισμένη άποψη δεδομένων με αναφορά στην πηγή των δεδομένων ή του προβλήματος, ή το χρόνο εμφάνισης κ.α. Ακόμη, εδώ μπορεί να ενταχθούν και οι «οθόνες» που παρουσιάζονται στον ιατρικό υπάλληλο (π.χ. ιατρό) μέσα από τον υπολογιστή ώστε να μπορεί να προβάλλει και να επεξεργάζεται την ιατρική πληροφορία.

- **Ασφάλεια**

Η ιατρική πληροφορία πρέπει να είναι (α) ακέραια και (β) εμπιστευτική. Τα συστήματα πρέπει να σχεδιάζονται ώστε να είναι σύμφωνα με τους νόμους, κανονισμούς και πρότυπα. Αυτό προϋποθέτει ποικίλα στρώματα ασφαλείας καθώς και μηχανισμούς ταυτοποίησης (authentication) και εξουσιοδότησης (authorization). (13) (10)

## **4.4 Πρότυπα Ηλεκτρονικού Ιατρικού Φακέλου**

### **4.4.1 Επίπεδο Υγείας 7 (Health Level 7)**

Το HL7 αναφέρεται στο επίπεδο υγείας 7 (Health Level 7), μια πλήρως εθελοντική, μη κερδοσκοπική οργάνωση υγείας που ενεπλάκη στην ανάπτυξη διεθνών προτύπων υγειονομικής περίθαλψης. Ο όρος HL7 αναφέρεται επίσης σε μερικά από τα συγκεκριμένα πρότυπα που δημιουργούνται από τον οργανισμό, όπως για παράδειγμα: HL7 v2.x, v3.0 κ.α.

Το HL7 και τα μέλη του αφιερώνονται στην παροχή ενός περιεκτικού πλαισίου (και των σχετικών προτύπων) για την ανταλλαγή, την ολοκλήρωση, τη διανομή και την ανάκτηση των ηλεκτρονικών πληροφοριών υγείας. Τα πρότυπα, που υποστηρίζουν την κλινική πρακτική και τη διαχείριση, παράδοση και αξιολόγηση των υγειονομικών υπηρεσιών, είναι τα περισσότερο χρησιμοποιούμενα παγκοσμίως.

Τα νοσοκομεία συνήθως χρησιμοποιούν πολλά διαφορετικά ηλεκτρονικά υπολογιστικά συστήματα για όλες τις λειτουργίες τους, από την τιμολόγηση των αρχείων μέχρι την παρακολούθηση ασθενούς. Όλα αυτά τα συστήματα πρέπει να επικοινωνήσουν το ένα με το άλλο (διεπαφή) όταν λαμβάνουν τις νέες πληροφορίες. Το HL7 είναι μια γλώσσα με την οποία τα διάφορα συστήματα υγειονομικής περίθαλψης μπορούν να το κάνουν αυτό. Υπάρχει μια μεγάλη ποσότητα πληροφοριών που τα συστήματα υγειονομικής περίθαλψης πρέπει να μεταβιβάσουν. Η σύνδεση των διάφορων συστημάτων υγειονομικής περίθαλψης είναι δύσκολη χωρίς μια κοινή γλώσσα (όπως το HL7).

Στόχος του HL7 είναι η κλινική διαλειτουργικότητα, δηλαδή «Να παρέχει πρότυπα στις ανταλλαγές, την διαχείριση και την ολοκλήρωση των δεδομένων που υποστηρίζουν την φροντίδα του ασθενούς, αλλά και την διαχείριση, διανομή και αποτίμηση σε ολόκληρο το περιβάλλον της φροντίδας υγείας». Η στρατηγική του HL7 είναι η συνεχής καινοτομία.

### **Υπόβαθρο και οργανωτική δομή**

Ο οργανισμός HL7 έχει αυξηθεί από 14 μέλη το 1987 σε πάνω από 2200 μέλη παγκοσμίως σήμερα, συμπεριλαμβανομένων 500 εταιρικών μελών και διεθνείς θυγατρικές σε 33 χώρες. Αυτά τα μέλη μοιράζονται μια δέσμευση, για την ανάπτυξη και την πρόοδο των κλινικών και διοικητικών προτύπων στην υγειονομική περίθαλψη. Χρησιμοποιώντας ένα καθορισμένο με σαφήνεια σύνολο λειτουργικών διαδικασιών, τα μέλη του HL7 – συμπεριλαμβανομένων των προμηθευτών, των συμβούλων και των πληρωτών – έχουν πείρα τεχνολογίας πληροφοριών σε όλα τα τμήματα της βιομηχανίας υγειονομικής περίθαλψης. Συλλογικά, αναπτύσσουν τα πρότυπα με σκοπό να αυξήσουν την αποτελεσματικότητα, την αποδοτικότητα και την ποιότητα παράδοσης υγειονομικής περίθαλψης.

Ο οργανισμός διευθύνεται από ένα διοικητικό συμβούλιο, το οποίο περιλαμβάνει οκτώ εκλεγμένες θέσεις και τρεις διορισμένες θέσεις. Τα μέλη του HL7 είναι συλλογικά γνωστά ως «ομάδα εργασίας». Η ομάδα εργασίας είναι αρμόδια για τον καθορισμό του HL7 τυποποιημένου πρωτοκόλλου και αποτελείται από τις μόνιμες διοικητικές επιτροπές, τις ομάδες ειδικού ενδιαφέροντος και τις τεχνικές επιτροπές. Οι μόνιμες διοικητικές επιτροπές εστιάζουν στις οργανωτικές ή προωθητικές δραστηριότητες, όπως η εκπαίδευση, η εφαρμογή, η διαφήμιση, η έκδοση και βελτίωση απόδοσης και η σχεδίαση. Οι ομάδες ειδικού ενδιαφέροντος χρησιμεύουν ως ένα πεδίο δοκιμής για να ερευνήσουν τις νέες περιοχές που μπορεί να χρειαστούν κάλυψη στα HL7 δημοσιοποιημένα πρότυπα. Οι Τεχνικές

Επιτροπές είναι άμεσα αρμόδιες για το περιεχόμενο των προτύπων, πλαισιώνοντας την πραγματική γλώσσα των προδιαγραφών.

Μια συχνή παρερμηνεία για το HL7 είναι ότι η ομάδα αναπτύσσει λογισμικό. Στην πραγματικότητα, η αρχική αποστολή του HL7 είναι να δημιουργηθούν ευέλικτα, χαμηλού κόστους πρότυπα, οδηγίες και μεθοδολογίες που θα επιτρέψουν την ανταλλαγή και τη διαλειτουργικότητα των ηλεκτρονικών χαρτών υγείας. Τέτοιες οδηγίες ή πρότυπα δεδομένων είναι ένα σύνολο κανόνων που επιτρέπουν στις πληροφορίες να μοιραστούν και να υποβληθούν σε επεξεργασία κατά τρόπο ομοιόμορφο και συνεπή. Χωρίς πρότυπα δεδομένων, οι οργανώσεις υγειονομικής περίθαλψης δεν θα μπορούσαν να μοιραστούν εύκολα κλινικές πληροφορίες. Θεωρητικά, αυτή η δυνατότητα να ανταλλαχθούν οι πληροφορίες πρέπει να βοηθήσει να ελαχιστοποιηθεί η τάση να είναι η ιατρική φροντίδα γεωγραφικά απομονωμένη και ιδιαίτερα μεταβλητή.

Οι περισσότεροι οργανισμοί ανάπτυξης προτύπων (Standards Developing Organizations), συμπεριλαμβανομένου του HL7, αναπτύσσουν πρότυπα για μια ιδιαίτερη περιοχή υγειονομικής περίθαλψης όπως το φαρμακείο, οι ιατρικές συσκευές, οι συναλλαγές απεικόνισης ή ασφάλειας. Το φάσμα του HL7 είναι κλινικά δεδομένα και δεδομένα διαχείρισης. Τα πρότυπα ανταλλαγής μηνυμάτων είναι ιδιαίτερα σημαντικά επειδή καθορίζουν πως οι πληροφορίες συσκευάζονται και διαβιβάζονται από το ένα συμβαλλόμενο μέρος σε άλλο. Τέτοια πρότυπα θέτουν τη γλώσσα, τύπους δομών και δεδομένων που απαιτούνται για τη συνεχή ολοκλήρωση από ένα σύστημα σε άλλο. Αυτή την περίοδο, τα πρότυπα ανταλλαγής μηνυμάτων HL7 υποστηρίζονται από κάθε σημαντικό προμηθευτή πληροφοριακών συστημάτων στις Ηνωμένες Πολιτείες.

### **Τομείς ενδιαφέροντος**

Το 1994, το HL7 αναγνωρίστηκε από το ANSI (διεθνή οργανισμό προτύπων). Στα έτη από την ίδρυσή του, το HL7 έχει επεκτείνει την επιρροή του πέρα από τα παραδοσιακά πρωτόκολλα ανταλλαγής μηνυμάτων. Σήμερα οι πρωτοβουλίες ανάπτυξης προτύπων HL7 περιλαμβάνουν τα εξής:

- Τυποποίηση της απεικόνισης γνώσης
- Προδιαγραφή των συστατικών για τη διαχείριση πλαισίου
- Υποστήριξη για την ανταλλαγή στοιχείων υγειονομικής περίθαλψης χρησιμοποιώντας μεσίτες αιτήματος αντικειμένου (object request broker)



- Τυποποίηση των δομών εγγράφων XML
- Λειτουργικές προδιαγραφές για ένα ηλεκτρονικό αρχείο υγείας
- Δουλειά στον τομέα της ασφάλειας, της μυστικότητας, της εμπιστευτικότητας και της υπευθυνότητας

Τέτοια καινοτομία έχει επιτρέψει τα πάντα, από τη διαθεσιμότητα ενός διαδικτυακού ιατρικού αρχείου ασθενούς στη συνταγή ενός φαρμακείου που ανταλλάσσεται με ένα HL7 έγγραφο XML. Στην πραγματικότητα, η HL7 αρχιτεκτονική φακέλου ασθενούς στην έκδοση 3.0 επιτρέπει ένα κοινό σχήμα για την ανταλλαγή του ιατρικού αρχείου ενός ασθενή μεταξύ των διαφορετικών συστημάτων νοσοκομείων ή ακόμα και των διαφορετικών νοσοκομείων. Αυτό το HL7 πρότυπο χρησιμεύει ως θεμέλιο για το ηλεκτρονικό ιατρικό αρχείο.

Ο σκοπός των δραστηριοτήτων του HL7 δεν περιορίζεται στο ηλεκτρονικό ιατρικό αρχείο, εντούτοις. Πράγματι, οι πρόσφατες δραστηριότητες και τα πρότυπα HL7 έχουν περιλάβει τη διαμόρφωση και τη μεθοδολογία, το λεξιλόγιο, την υποστήριξη κλινικής απόφασης, την οικονομική διαχείριση, τη διοίκηση, τη ρυθμισμένη κλινική έρευνα και τη διαχείριση πληροφοριών, το σχεδιασμό και τις διοικητικές μέριμνες, τις κλινικές οδηγίες, τα κυβερνητικά προγράμματα, το φάρμακο, την ασφάλεια και την υπευθυνότητα, τα πρότυπα XML, και την απάντηση δημόσιας υγείας και έκτακτης ανάγκης.

Το HL7 έχει επιτρέψει τη διαλειτουργικότητα μεταξύ των συστημάτων ηλεκτρονικής διοίκησης ασθενών (Patient Administration Systems), των συστημάτων ηλεκτρονικής πρακτικής διαχείρισης (Electronic Practice Management), των συστημάτων εργαστηριακών πληροφοριών (Laboratory Information Systems). Το HL7 καλύπτει τον πλήρη κύκλο ζωής μιας προδιαγραφής προτύπων συμπεριλαμβανομένης της ανάπτυξης, της υιοθέτησης, της αναγνώρισης αγοράς και της χρησιμοποίησης.

### **Συγκεκριμένοι στόχοι HL7**

Ο κύριος στόχος του HL7 είναι η ανάπτυξη ενός προτύπου το οποίο θα επιτρέπει τη μεταφορά και ανταλλαγή δεδομένων (ιατρικής πληροφορίας) μεταξύ εντελώς διαφορετικών οργανισμών και συστημάτων. Αναλυτικότερα:

- Ανάπτυξη κατανοητών, με δυνατότητα επέκτασης προτύπων που επιτρέπουν δομημένη, κωδικοποιημένη πληροφορία υγειονομικής περίθαλψης, που ανταλλάσσεται μεταξύ πληροφοριακών εφαρμογών διατηρώντας τη σημασία.
- Ανάπτυξη μιας επίσημης μεθοδολογίας για την υποστήριξη της δημιουργίας HL7 προτύπων.
- Εκπαίδευση της βιομηχανίας υγειονομικής περίθαλψης, τους φορείς χάραξης πολιτικής, και το ευρύ κοινό σχετικά με τα οφέλη της τυποποίησης πληροφοριών υγειονομικής περίθαλψης γενικά και για τα HL7 πρότυπα συγκεκριμένα.
- Προώθηση της χρήσης HL7 προτύπων παγκοσμίως μέσω της δημιουργίας διεθνών θυγατρικών οργανισμών HL7, οι οποίοι συμμετέχουν στην ανάπτυξη HL7 των προτύπων.
- Συνεργασία με άλλες οργανώσεις ανάπτυξης προτύπων και εθνικούς και διεθνείς οργανισμούς (π.χ. ANSI και ISO), στις περιοχές υποδομής υγειονομικής περίθαλψης και πληροφοριών για προώθηση της χρήσης των υποστηρικτικών και συμβατών προτύπων.
- Συνεργασία με τους χρήστες τεχνολογίας πληροφοριών υγειονομικής περίθαλψης για να εξασφαλιστεί πως τα HL7 πρότυπα καλύπτουν τις πραγματικές απαιτήσεις, και ότι οι προσπάθειες ανάπτυξης κατάλληλων προτύπων δημιουργούνται από τον οργανισμό HL7 για να καλύψουν τις προκύπτουσες απαιτήσεις.

### ***RIM – ISO/HL7 21731***

Το μοντέλο αναφοράς πληροφορίας (RIM) είναι ο ακρογωνιαίος λίθος της ανάπτυξης της έκδοσης 3 του HL7 και ένα αναγκαίο μέρος της μεθοδολογίας ανάπτυξης του HL7. Το RIM εκφράζει τα δεδομένα που απαιτούνται σε μια συγκεκριμένη έκφανση της κλινικής λειτουργίας και παρέχει μια συγκεκριμένη αναπαράσταση των μεταπληροφοριών (semantics) και λεξικών συνδέσεων που υπάρχουν μεταξύ των μηνυμάτων που μεταφέρονται στο επίπεδο του HL7.

### ***Επίπεδο ανάπτυξης (development framework) HL7 – ISO/HL7 27931***

Το επίπεδο ανάπτυξης (HDF) του HL7 προτύπου 3 είναι μια διαδικασία συνεχούς εξέλιξης που αποζητά να αναπτύξει πρότυπα και ορισμούς που αναπτύσσουν τη διαλειτουργικότητα μεταξύ πληροφοριακών συστημάτων υγείας.

Το HDF επίσης καταγράφει τις διαδικασίες, εργαλεία, κανόνες και οτιδήποτε σχετίζεται με τα πρότυπα ανάπτυξης του HL7. Κάποια στιγμή, το HL7 θα περιλαμβάνει όλες τις προδιαγραφές των προτύπων του

HL7, συμπεριλαμβανομένων των καινούριων προτύπων που θα προκύψουν από την ανάλυση πληροφοριακών συστημάτων υγείας.

### **Ανταλλαγή μηνυμάτων στο HL7 v3**

Στην έκδοση 3, το πρότυπο ανταλλαγής μηνυμάτων προσδιορίζει μια σειρά από ηλεκτρονικά μηνύματα για να υποστηριχθούν όλες οι ιατρικές ροές εργασίας. Τα μηνύματα αυτά βασίζονται στη σύνταξη της XML.

#### **4.4.2 Συνοχή φακέλου φροντίδας (Continuity of Care Record, CCR)**

Η συνοχή φακέλου φροντίδας (CCR) είναι ένα πρότυπο φακέλου υγείας που αναπτύσσεται από κοινού από την ASTM International, τον Ιατρικό Σύλλογο Μασαχουσέτης (Massachusetts Medical Society, MMS), το HIMS, την αμερικανική ακαδημία των οικογενειακών παθολόγων (American Academy of Family Physicians, AAFP), την αμερικανική ακαδημία παιδιατρικής (American Academy of Pediatrics, AAP), και από άλλους προμηθευτές πληροφορικής υγείας.

Το πρότυπο CCR είναι πρότυπο περίληψης υγείας ασθενούς. Είναι ένας τρόπος να δημιουργηθούν ευέλικτα έγγραφα που περιέχουν τις πιο σχετικές και έγκαιρες/έγκυρες πληροφορίες υγείας για έναν ασθενή, και να σταλούν αυτές ηλεκτρονικά από έναν γιατρό σε άλλον. Περιέχει διάφορα τμήματα, όπως οι ασφαλιστικές πληροφορίες του ασθενούς, η διάγνωση και ο κατάλογος προβλημάτων, τα φάρμακα, οι αλλεργίες και το σχέδιο περίθαλψης. Όλα αυτά αντιπροσωπεύουν ένα στιγμιότυπο “snapshot” των στοιχείων υγείας ενός ασθενούς που μπορεί να είναι χρήσιμα ή και ακόμη να σώσουν ζωές. Το ASTM CCR πρότυπο έχει σχεδιαστεί για να επιτρέψει την εύκολη χρήση από έναν παθολόγο χρησιμοποιώντας ένα ηλεκτρονικό σύστημα αρχείων υγείας.

Επειδή εκφράζεται με την τυποποιημένη γλώσσα ανταλλαγής δεδομένων XML, ένα CCR μπορεί να δημιουργηθεί, να διαβαστεί και να ερμηνευθεί από οποιοδήποτε εφαρμογές λογισμικού ηλεκτρονικού φακέλου ασθενούς. Ένα CCR μπορεί επίσης να εξαχθεί με άλλη μορφή, όπως PDF και Microsoft Word.

Το CCR γίνεται το επιλεγμένο πρότυπο για την ανταλλαγή πληροφοριών ασθενών και για την βιομηχανική προσπάθεια να βελτιωθεί η ποιότητα της δυνατότητας μεταφοράς πληροφοριών υγείας για την αντιμετώπιση λαθών. Το CCR φέρει στην επιφάνεια την δυνατότητα των ηλεκτρονικών ιατρικών

φακέλων (Electronic Medical Records, EMRs) να λειτουργήσουν με τα πρωτόκολλα επικοινωνίας, το υλικό, το λογισμικό, την εφαρμογή, και τα στρώματα συμβατότητας δεδομένων. Υποστηρίζεται ότι με καλύτερες πολιτικές καθοδήγησης και την τρέχουσα τεχνολογία, η συνοχή της περίθαλψης θα διευκολύνει μια ομαλή μετάβαση των πληροφοριών ασθενών μέσω πολλαπλών πρακτικών από διαφορετικά EMRs.

### **Σκοπός**

Ένας από τους σκοπούς των ηλεκτρονικών ιατρικών φακέλων (EMRs) είναι να αυξηθεί η δυνατότητα πρόσβασης και διανομής των φακέλων υγείας ανάμεσα σε εξουσιοδοτημένα ξεχωριστά μέλη. Το CCR παρέχει μια πρόσθετη χρησιμότητα στο EMR, το να χρησιμεύσει ως ένα μέσο με δυνατότητα μεταφοράς πληροφοριών. Στα οικονομικά η οριακή χρησιμότητα είναι η πρόσθετη χρησιμότητα (ικανοποίηση ή όφελος) που ένας καταναλωτής αντλεί από μια πρόσθετη μονάδα προϊόντων ή μιας υπηρεσίας. Σύμφωνα με τα χρηστικά, η κοινωνία πρέπει να στοχεύσει στο να μεγιστοποιήσει τη συνολική χρησιμότητα των ξεχωριστών μελών, που στοχεύουν στη «μέγιστη ευτυχία για το μέγιστο αριθμό». Κατά συνέπεια, το CCR χρησιμεύει ως μια πρόσθετη μονάδα υπηρεσίας για τα EMRs που στοχεύουν στη μεγιστοποίηση της συνολικής χρησιμότητας της βιομηχανίας υγειονομικής περίθαλψης.

### **Ανάγκη**

Τα τελευταία χρόνια η ανάγκη για τη δυνατότητα μεταφοράς των πληροφοριών ασθενών είναι θέμα συζήτησης σχεδόν σε κάθε τομέα της υγειονομικής περίθαλψης. Διάφοροι παράγοντες διαδραματίζουν έναν ρόλο στην απόφαση να εισαχθούν τα πρότυπα δυνατότητας μεταφοράς CCR. Κατ'αρχήν, υπάρχει ανάγκη για τη δυνατότητα μεταφοράς που θα επιτρέψει στον επόμενο προμηθευτή εύκολη πρόσβαση στα πιο πρόσφατα αρχεία ασθενών. Δεύτερον, υπάρχει μια ανάγκη για το προσωπικό αρχείο υγείας που περιέχει τις εισαγμένες πληροφορίες ασθενούς. Τρίτον, υπάρχει ανάγκη για τη συγκράτηση του κόστους με την ανάπτυξη μιας συστηματικότερης προσέγγισης στη δυνατότητα μεταφοράς πληροφοριών υγειονομικής περίθαλψης.

### **Βασικά στοιχεία**

Αυτή τη στιγμή το CCR αποτελείται από έξι υποχρεωτικά βασικά στοιχεία. Τα τμήματα είναι:

- Η επικεφαλίδα ή πληροφορία προσδιορισμού εγγράφου
- Πληροφορία προσδιορισμού ασθενούς

- Πληροφορία οικονομικών και ασφάλειας ασθενούς
- Κατάσταση υγείας του ασθενή
- Τεκμηρίωση περίθαλψης
- Σύσταση σχεδίου περίθαλψης

### **Διαθεσιμότητα**

Η κατάσταση διαθεσιμότητας της λειτουργίας CCR για τα συστήματα EMR μπορεί ευρέως να διαρεθεί σε τέσσερις κατηγορίες. Η πρώτη προσέγγιση είναι να ενσωματωθούν όλα τα σχετικά βασικά στοιχεία σε ένα εννοιολογικό πρότυπο. Η δεύτερη προσέγγιση επεκτείνεται των βασικών στοιχείων σε υποδιαιρέσεις για τη συντήρηση της ακεραιότητας των δεδομένων. Η τρίτη προσέγγιση βρίσκεται στην εξαγωγή και την αποθήκευση όλων των σχετικών ιατρικών δεδομένων σε μια συνδυασμένη συνοχή του αρχείου περίθαλψης (CCR). Η διαφορά μεταξύ ενός CCR και ενός EMR είναι πως το CCR δεν είναι μια απόλυτη έκδοση του EMR. Δεν απαιτεί τα γεγονότα να βρίσκονται σε χρονολογική σειρά όπως στο EMR. Το CCR περιλαμβάνει λιγότερα διηγήματα, ελεύθερα κείμενα και είναι πιο συνοπτικό. Η τέταρτη προσέγγιση CCR έχει ως σκοπό να υποστηριχθεί ενεργά η διαλειτουργικότητα και η λειτουργία εισαγωγής του CCR από άλλα EMRs.

### **Υπολογισμός Χρησιμότητας**

Η χρησιμότητα της εφαρμογής CCR μπορεί να υπολογιστεί λαμβάνοντας υπόψη τις παρακάτω συνιστώσες

- Ακριβείς πληροφορίες ασθενών για τον επόμενο παροχέα υγειονομικής περίθαλψης (γιατρούς)
- Ενημερωμένη διαθεσιμότητα των φακέλων αξιολόγησης και συμβουλής ασθενούς
- Βέλτιστη χρησιμοποίηση των τεχνολογικών πόρων
- Να καταστεί η διαδικασία δυνατότητας μεταφοράς πληροφοριών υγείας απαλλαγμένη από τα επαναλαμβανόμενα δεδομένα
- Εύκολη καθιέρωση της πληροφορίας και ασφάλισης των ασθενών
- Μείωση του κόστους και του χρόνου που συνδέεται με τις επαναλαμβανόμενες εξετάσεις, μέσω της δυνατότητας μεταφοράς
- Ελαχιστοποίηση της προσπάθειας για έγκαιρη ενημέρωση των φακέλων ασθενών

- Χρήση XML ως τεχνολογία για εύκολη δυνατότητα μεταφοράς κλινικής πληροφορίας

Η συνολική χρησιμότητα σημαίνει τη συνολική ικανοποίηση που προέρχεται από τον καταναλωτή από όλες τις μονάδες οικονομικών αγαθών που καταναλώνονται μαζί, δηλ. το σύνολο ποσού των χρησιμότητων που βιώνονται από τον καταναλωτή από όλες τις μονάδες οικονομικών αγαθών. Η συνολική χρησιμότητα της βιομηχανίας υγειονομικής περίθαλψης θα περιλάμβανε έτσι την ικανοποίηση που παράγεται από τη χρησιμότητα του CCR. Τα μέλη “actors” που εμπλέκονται στο στάδιο παραγωγής της χρησιμότητας στη βιομηχανία υγειονομικής περίθαλψης περιλαμβάνουν:

- Ασθενής
- Ρεσεψιονίστ
- Νοσοκόμα
- Παθολόγος
- Προσωπικό διαφήμισης
- Διευθυντής γραφείων
- Φαρμακείο
- Εργαστήριο
- Αναφορά του παθολόγου
- Ασφαλιστική εταιρία
- Νοσοκομεία

Κάθε μέλος έχει έναν μεμονωμένο ρόλο που συμβάλλει αποτελεσματικά προς το πλήρες σύστημα υγειονομικής περίθαλψης. Σύμφωνα με τον Jeannie L. Haggerty, “η συνοχή της περίθαλψης επιτυγχάνεται με την γεφύρωση των ιδιαίτερων στοιχείων στα διαφορετικά επεισόδια που δρόμου της περίθαλψης, της μεσολαβήσεις από διαφορετικούς προμηθευτές, ή τις αλλαγές στην κατάσταση της αρρώστιας, παράλληλα με την υποστήριξη απόψεων που διαρκούν με την πάροδο του χρόνου, όπως οι αξίες των ασθενών, τις συνεχείς σχέσεις και το σχέδιο περίθαλψης”.

### **Συνέπειες**

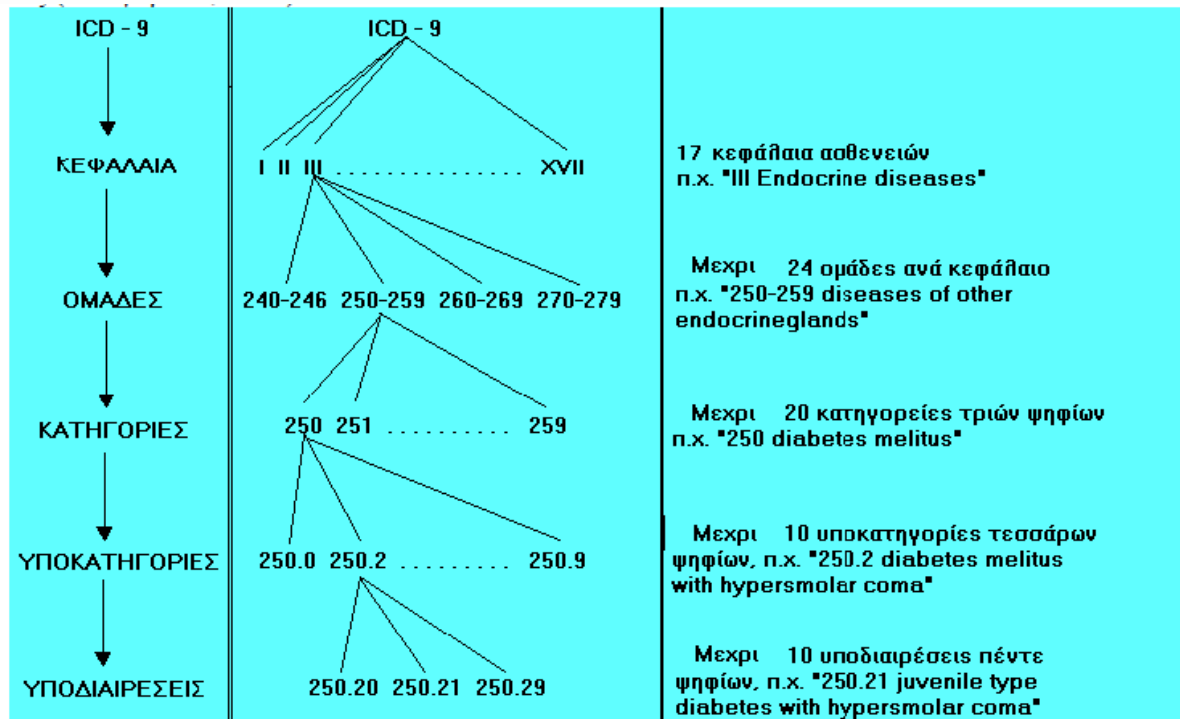
Οι επιλογές που γίνονται όσον αφορά τη διαδικασία εφαρμογής CCR είναι συχνά σύνθετες και απαιτούν την κατάλληλη κατανόηση των προτύπων και της δομής δεδομένων των βασικών στοιχείων. Η εφαρμογή πρέπει να κρατηθεί σύμφωνη με τις οργανωτικές εξελίξεις, ειδικά στις περιοχές όπου οι

διαδικασίες εργασίας πρέπει να αναδομηθούν ή να ξανασχεδιαστούν με την εισαγωγή του CCR. Είναι απαραίτητο να προσεγγιστεί το πρόγραμμα CCR ως μεγάλο σχέδιο από την έναρξή του. Ένας αποτελεσματικός συγχρονισμός του προγράμματος θα προσφέρει το ικανοποιητικό πεδίο για τις εξελίξεις, εξασφαλίζοντας ότι αυτές μπορούν να ενσωματωθούν επαρκώς στο σύστημα EMR.

Αν και το CCR είναι ένα πρότυπο ανάπτυξης, μια σημαντική πρόκληση θα ήταν αυτή της προσέλκυσης των προμηθευτών EMR για να ενσωματώσουν τα πρότυπα CCR. Τα ουσιαστικά πλεονεκτήματα της ηλεκτρονικής εφαρμογής ιατρικών φακέλων στα πρότυπα CCR υπάρχουν. Η σημασία τμιας αποτελεσματικής CCR είναι η καθολική αναγνώριση. Κατά συνέπεια, η εμπιστοσύνη στην αποτελεσματικότητα της βιομηχανίας υγειονομικής περίθαλψης μπορεί να αυξηθεί με την χρήση του βασισμένου στο XML πρότυπο CCR. Η θεωρία χρησιμότητας αποδεικνύει ότι όσο μεγαλύτερη η ανάγκη για τη δυνατότητα μεταφοράς πληροφοριών υγείας τόσο μεγαλύτερο θα ήταν το ποσοστό αποδοχής του CCR προτύπου.

#### **4.4.4 Το διεθνές σύστημα κατηγοριοποίησης ασθενειών (ICD)**

Μία από τις αιχμές του δόρατος για την απόλυτη εκμετάλλευση του ΗΦΑ αποτελεί ο συνδυασμός του με συστήματα κατηγοριοποίησης των ασθενειών (π.χ. International Classification of Diseases, 9<sup>th</sup> revision, Clinical Modification, ICD-9-CM, ή το πιο πρόσφατο ICD-10), που αναπτύχθηκε αρχικά με πρωτοβουλία της παγκόσμιας οργάνωσης υγείας (WHO) και του Εθνικού Κέντρου Στατιστικής της Υγείας των ΗΠΑ. Σκοπός του ήταν η αποθήκευση, ανάκτηση και επεξεργασία της κλινικής πληροφορίας με χρήση αλφαριθμητικών κωδικών. Η δομή του IDC-9-CM φαίνεται στην παρακάτω εικόνα.



ΣΧΗΜΑ 20 ΔΟΜΗ ΤΟΥ ICD9

#### 4.4.5 EN13606

Ο στόχος του προτύπου Health Informatics – Electronic Health Record Communication (EN 13606) είναι να παρέχει μια σταθερή αρχιτεκτονική πληροφορία για επικοινωνία τμήματος ή ολόκληρου του ΗΦΑ ενός ασθενούς. Αυτό γίνεται για να υποστηριχθεί η διαλειτουργικότητα των συστημάτων που χρειάζονται να επικοινωνήσουν δεδομένα του ΗΦΑ μέσω ηλεκτρονικών μηνυμάτων ή κατανεμημένων αντικειμένων.

Το όνομα αυτού του Ευρωπαϊκού προτύπου υποδηλώνει ότι δεν έχει σαν σκοπό να περιγράψει την εσωτερική αρχιτεκτονική ή σχεδίαση βάσης δεδομένων πληροφοριακών συστημάτων ΗΦΑ αλλά ουσιαστικά περιλαμβάνει τη χρήση ενός «αντικειμένου» (object). Αυτά τα αντικείμενα μπορεί να αποθηκεύονται, διαβάζονται και ανταλλάσσονται. Δεν περιγράφει σε καμία περίπτωση τα είδη των κλινικών λογισμικών που μπορεί να ζητάνε δεδομένα ΗΦΑ. Για αυτό το λόγο, το μοντέλο πληροφορίας που προτείνεται ονομάζεται τμήμα ΗΦΑ (HER Extract). Αυτό μπορεί να χρησιμοποιηθεί για να περιγράψει ένα μήνυμα, ένα XML έγγραφο ή σχήμα ή μια διεπαφή αντικειμένου.



## 4.5 Μετάβαση από τον κλασσικό στον ηλεκτρονικό ιατρικό φάκελο

### 4.5.1 Ενημέρωση

Η αποτελεσματική υλοποίηση δικτυακών υπηρεσιών στα νοσοκομεία προϋποθέτει την ενεργό συμμετοχή και εναρμόνιση όλων των εμπλεκομένων στις νέες διαδικασίες. Οι αρχικοί χρήστες των νέων διαδικασιών οι οποίες υποστηρίζονται από πληροφοριακά συστήματα υγείας θα δώσουν το πρώτο στίγμα επιτυχούς εφαρμογής των ηλεκτρονικών διαδικασιών στην καθημερινή λειτουργία του νοσοκομείου.

Για ένα επιτυχημένο ξεκίνημα, λοιπόν, θα πρέπει να εκπαιδευτούν ώστε να κατανοήσουν και να υιοθετήσουν τις νέες πρακτικές, αλλά κυρίως να παρακινηθούν κατάλληλα ώστε να συμμετέχουν ενεργά και εποικοδομητικά, συμβάλλοντας έτσι στην επίτευξη των συνολικών στόχων ικανοποίησης των τελικών αποδεκτών των υπηρεσιών (πολίτες/ασθενείς) και αποτελεσματικής λειτουργικότητας των επιμέρους μονάδων του νοσοκομείου.

Συνεπώς, θα πρέπει να προσδιοριστούν οι διάφορες ομάδες χρηστών του πληροφοριακού συστήματος υγείας, να μελετηθούν οι δυνατότητές τους και να δημιουργηθεί ένα πλάνο στον γενικότερο σχεδιασμό, για την παρακίνησή τους ώστε να συμμετάσχουν ενεργά και ουσιαστικά στην εφαρμογή των δικτυακών υπηρεσιών.

Η διοίκηση θα πρέπει πριν από την εισαγωγή νέων ηλεκτρονικών διαδικασιών να λάβει υπόψη της κυρίως τον ανθρώπινο παράγοντα, πέρα από την υλικοτεχνική υποδομή και την τεχνολογική λύση. Η διασφάλιση της ενεργούς συμμετοχής των εργαζομένων είναι αναγκαία συνθήκη για την επίτευξη πρωτογενών και δευτερογενών στόχων, που κυμαίνονται από την ικανοποίηση των ασθενών έως τη μείωση του λειτουργικού κόστους και την αύξηση της παραγωγικότητας.

Οι χρήστες θα πρέπει έγκαιρα να ενημερώνονται για τις επερχόμενες αλλαγές, ώστε να υπάρχει κατάλληλος χρόνος για επίλυση αποριών και εξοικείωση τους με την ιδέα της νέας πραγματικότητας, για να μη νιώθουν ότι απειλούνται.

Καλό θα ήταν επίσης, να επιλεγούν σε συνεργασία με τους ιθύνοντες του κάθε τμήματος άτομα με έφεση και ενδιαφέρον, τα οποία αφενός θα προσαρμοστούν ευκολότερα στο νέο περιβάλλον και αφετέρου θα λειτουργήσουν ως υποκινητές για τους συναδέλφους τους.

#### **4.5.2 Εκπαίδευση χρηστών**

Όλοι οι εργαζόμενοι – χρήστες των νέων υπηρεσιών (διοικητικό, νοσηλευτικό και επιστημονικό προσωπικό) θα πρέπει να λάβουν εκπαίδευση ώστε να κατανοήσουν τις νέες διαδικασίες. Στα πλαίσια της εκπαίδευσης-κατάρτισής τους οι δυνατότητες των νέων εργαλείων και η λειτουργικότητα των συστημάτων θα πρέπει να μεταδοθούν με απλά λόγια, ώστε να γίνουν κατανοητά από όλα τα επίπεδα χρηστών.

Η εκπαίδευση των χρηστών θα πρέπει να γίνεται ανά ομάδες, βάσει του βαθμού εξοικειώσής τους και των γνώσεών τους.

#### **4.5.3 Παρακίνηση χρηστών**

Είναι καλό να δοθούν στους εργαζόμενους διάφορα απλά κίνητρα, τα οποία θα βοηθήσουν περισσότερο στην συμμετοχή τους και την ενασχόληση τους με την μετατροπή των διαδικασιών του νοσοκομείου ώστε να επιτελούνται με ηλεκτρονικά μέσα. Σε αυτά περιέχεται και η διατήρηση υπαρχόντων εφαρμογών, ήδη ενσωματωμένων στις καθημερινές λειτουργίες της εκάστοτε μονάδας υγείας, με τις οποίες οι χρήστες είναι ήδη εξοικειωμένοι. Περαιτέρω παρακινητικά κίνητρα θα πρέπει επίσης να δοθούν στους εργαζόμενους και να περιλαμβάνουν ελάττωση αντικειμένου (φόρτου) εργασίας, βελτίωση συνθηκών εργασίας αλλά και, πιθανώς, κάποια οικονομικά οφέλη.

#### **4.5.4 Περισσότερη εσωτερική οργάνωση**

Ανάλογα με την εκάστοτε τεχνική λύση που θα υιοθετήσουμε, ενδέχεται να προκύψει η ανάγκη αναδιοργάνωσης για να υποστηριχτούν καλύτερα οι νέες διαδικασίες. Η συγκρότηση νέων τμημάτων είναι ένα βήμα για την ταχύτερη εκπλήρωση των στόχων ηλεκτρονικοποίησης των διαδικασιών μιας ιατρικής μονάδας.

Η εσωτερική αναδιοργάνωση και ο άρτιος υλικοτεχνικός εξοπλισμός εμπεριέχονται βέβαια στο στάδιο του σχεδιασμού. Θα πρέπει όμως να εξετάζεται η πλήρωση των αρχικών στόχων σε συνάρτηση με το βαθμό αξιοποίησης των διαθέσιμων πόρων.

#### **4.5.5 Συνεχής μέριμνα**

Είναι απαραίτητη η συνεχής παρακολούθηση κατά τα μεταβατικά στάδια υιοθέτησης νέων πρακτικών, καθώς και η άμεση επίλυση των προβλημάτων που ενδέχεται να προκύψουν. Θα πρέπει, λοιπόν, να προβλέψουμε την ύπαρξη ειδικής ομάδας (όπως π.χ. το τμήμα IT, αν υπάρχει) η οποία θα είναι

διαθέσιμη, ώστε να διευκολύνει και να υποστηρίζει (π.χ. με on-site εκπαίδευση) το προσωπικό συνεχώς, για ένα σεβαστό χρονικό διάστημα ούτως ώστε να μην υπάρχει πρόβλημα στην καθημερινή λειτουργία των διαφόρων τμημάτων, αλλά και να μην επιβαρύνεται ο εργασιακός φόρτος.

Οι χρήστες θα πρέπει να έχουν την υποστήριξη από τα ανώτερα κλιμάκια, ώστε να πληρούνται όλες οι αρχικές προϋποθέσεις που θα συμβάλλουν στην διευκόλυνση του προσωπικού κατά το μεταβατικό στάδιο ολοκλήρωσης των αλλαγών. Η υποστήριξη θα πρέπει να είναι συνεχής (όχι μόνο στην αρχή) και να εμπεριέχει τόσο παροχή βοήθειας για την επίλυση τεχνικών δυσκολιών όσο και ενθάρρυνση των χρηστών για να συμμετάσχουν ενεργά στις νέες διαδικασίες. Επίσης, θα πρέπει να λαμβάνεται υπόψη η διαφορετικότητα των χρηστών (λόγω μορφωτικού επιπέδου κυρίως αλλά και προδιάθεσης απέναντι στην επικείμενη αλλαγή).

Η εναρμόνιση των δικτυακών υπηρεσιών στην καθημερινή λειτουργία μιας μονάδας υγείας απαιτεί σχεδιασμό και αποφασιστικότητα από την πλευρά της διοίκησης σε όλα τα στάδια υλοποίησης.

## 5. Μεθοδολογία έρευνας

---

Η μεθοδολογία έρευνας που χρησιμοποιήθηκε για την παρούσα διπλωματική εργασία είναι η έρευνα δράσης (action research).

Η έρευνα δράσης είναι μιας μικρής έντασης παρέμβαση στη λειτουργία του πραγματικού κόσμου και μια άμεση εξέταση των αποτελεσμάτων της παρέμβασης αυτής. Κύριο χαρακτηριστικό της είναι ότι λειτουργεί κατά περίπτωση – ενδιαφέρεται για τη διάγνωση ενός προβλήματος σε συγκεκριμένο πλαίσιο και για την επίλυσή του στο πλαίσιο αυτό. Οι ερευνητές εργάζονται με άλλους για να προτείνουν μεθόδους και τρόπους δράσης έτσι ώστε να βοηθήσουν την κοινότητα να αναπτύξει και να βελτιώσει τον τρόπο που εργάζεται. Ο Kurt Lewin, καθηγητής τότε στο MIT, ήταν ο πρώτος που χρησιμοποίησε τον όρο “action research” το 1944. Στο paper που δημοσίευσε το 1946 με τίτλο “Action Research and Minority Problems” περιέγραψε τη μεθοδολογία έρευνας ως “μια συγκριτική έρευνα στις καταστάσεις και στα αποτελέσματα διαφόρων μορφών κοινωνικής δράσης και έρευνας που οδηγεί σε κοινωνική δράση”, η οποία χρησιμοποιεί “συγκεκριμένα βήματα, με κάθε ένα να αποτελείται από έναν κύκλο σχεδιασμού, δράσης και εύρεσης δεδομένων σχετικά με το αποτέλεσμα μιας δράσης”. [14,15]

Η έρευνα δράσης είναι μια αλληλεπιδραστική διεργασία η οποία προσπαθεί να ισορροπήσει δράσεις που έχουν υλοποιηθεί σε ένα συνεργατικό περιβάλλον με συνεργατική ανάλυση και έρευνα σε δεδομένα ούτως ώστε να κατανοηθούν οι αιτίες που προκαλούν μία προσωπική ή μια επαγγελματική αλλαγή (π.χ. σε έναν οργανισμό). Μετά από έξι δεκαετίες ανάπτυξης της έρευνας δράσης, πολλές μεθοδολογίες έχουν αναπτυχθεί που προσαρμόζουν την ισορροπία στην προσοχή είτε στις δράσεις που γίνανε είτε στην έρευνα που προκύπτει από την κατανόηση των δράσεων. Αυτή η διαφοροποίηση υπάρχει μεταξύ

- Αυτών που οδηγούνται από το πρόγραμμα δράσης του ερευνητή και αυτών που οδηγούνται από τους συμμετέχοντες
- Αυτών που έχουν κυρίως σαν κίνητρο την κατάκτηση κάποιων στόχων και αυτών που έχουν σαν κίνητρο το στόχο της προσωπικής, οργανωτικής ή κοινωνικής μεταμόρφωσης
- Έρευνα πρώτου, δεύτερου, τρίτου προσώπου (1<sup>st</sup>-, to 2<sup>nd</sup>-, to 3<sup>rd</sup>-person research)
  - «Η έρευνα μου είναι οι δράσεις μου, με στόχο την προσωπική μου αλλαγή»

- Η έρευνα μιας ομάδας χρηστών, με στόχο την βελτίωση της ομάδας
- Μια πιο επιστημονική (σε σχέση με τις παραπάνω) έρευνα με στόχο θεωρητική γνώση και/ή αλλαγές σε μεγάλη κλίματα

Επιπροσθέτως, η έρευνα δράσης απαιτεί συνήθως (όχι απαραίτητα) συνεργασία – ομάδες ερευνητών να δουλεύουν μαζί. Είναι, επίσης, συμμετοχική (τα ίδια τα μέλη των ομάδων λαμβάνουν μέρος άμεσα ή έμμεσα στην έρευνα) και συνεχώς επεκτάσιμη – γίνονται συνεχώς αλλαγές που αξιολογούνται με σκοπό την επίτευξη όσο το δυνατόν καλύτερου αποτελέσματος ή αλλαγή της εφαρμοζόμενης πρακτικής. Οι σκοποί της έρευνα δράσης κατατάσσονται σε πέντε κυρίως κατηγορίες

- Είναι ένα μέσο για την αντιμετώπιση προβλημάτων των οποίων η διάγνωση γίνεται σε συγκεκριμένες συνθήκες ή για τη βελτίωση δεδομένων συνθηκών.
- Είναι ένα μέσο άμεσης και πρακτικής εκπαίδευσης των διδασκόντων, τροφοδοτώντας τους με νέες μεθόδους και δίνοντάς τους τη δυνατότητα αξιολόγησης των μεθόδων που χρησιμοποιούν.
- Αποτελεί τρόπο για εισαγωγή επιπρόσθετων και καινοτόμων προσεγγίσεων στις μεθόδους διδασκαλίας.
- Βοηθά στην επικοινωνία του πρακτικού δασκάλου με τον ερευνητή.
- Παρότι στερείται της ισχύος μιας πλήρους επιστημονικής έρευνας, είναι ένας τρόπος ο οποίος παρέχει μια εναλλακτική προσέγγιση της επίλυσης των προβλημάτων στην τάξη.

### **Χαρακτηριστικά της μεθόδου**

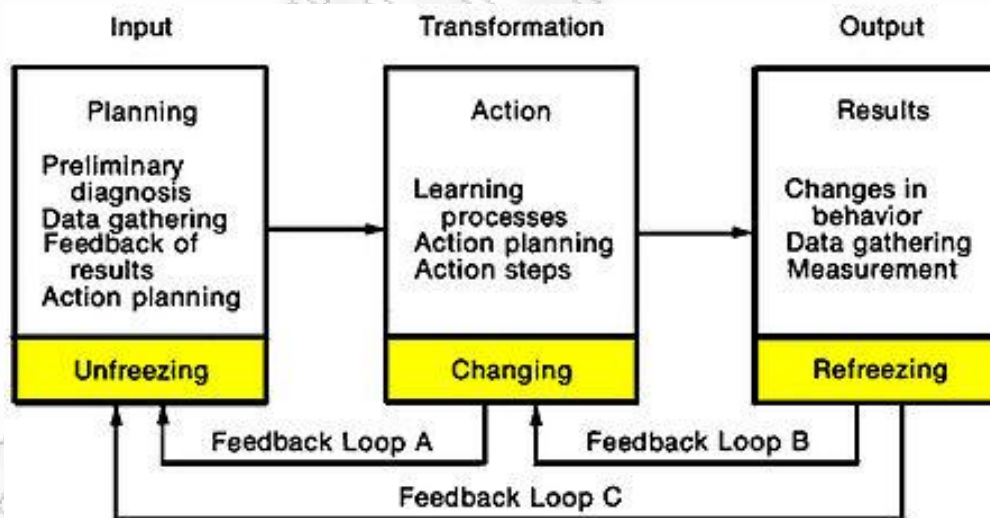
- Πρόκειται για μια διαδικασία προσανατολισμένη στο στόχο και σχεδιασμένη να αντιμετωπίσει ένα συνολικό πρόβλημα σε μια υπάρχουσα κατάσταση. Αυτό σημαίνει ότι έχουμε μια βήμα προς βήμα διαδικασία η οποία παρακολουθείται σε διάφορες χρονικές στιγμές και με μια ποικιλία μηχανισμών (πχ συνεντεύξεις, ερωτηματολόγια κλπ) και της οποίας η ανάδραση μπορεί να μεταφραστεί σε τροποποιήσεις, προσαρμογές, αλλαγές προσανατολισμού και επαναπροσδιορισμούς προς όφελος της τρέχουσας διεργασίας.
- Ευελιξία και η προσαρμοστικότητα. Οι αξίες αυτές αποκαλύπτονται στις αλλαγές που μπορεί να λάβουν χώρα κατά την εφαρμογή της και στη δυνατότητα άμεσου ελέγχου.
- Είναι σε μεγάλο βαθμό εμπειρική.

## Η μεθοδολογία έρευνας στην ανάπτυξη ενός οργανισμού

Οι Wendell L. French και Cecil Bell ορίζουν την ανάπτυξη ενός οργανισμού ως «βελτίωση στις διαδικασίες ενός οργανισμού» μέσω μεθοδολογίας έρευνας. Αν μπορούσαμε να πούμε μία φράση για την φιλοσοφία της ανάπτυξης οργανισμού, θα ήταν η «μεθοδολογία έρευνας» με τον τρόπο που την συνέλαβε ο Kurt Lewin και αργότερα εξελίχθηκε από άλλους επιστήμονες. Ο Lewin, που είχε σαν ερευνητικό ενδιαφέρον την κοινωνική αλλαγή (και πιο συγκεκριμένα, την μόνιμη κοινωνική αλλαγή) πίστευε ότι το κίνητρο για αλλαγή ήταν άμεσα συσχετισμένο με τη δράση. Αν οι άνθρωποι είναι ενεργοί σε αποφάσεις που τους επηρεάζουν, είναι πολύ πιο πιθανό να υιοθετήσουν νέους τρόπους και αλλαγές. [14]

Ο Lewin περιγράφει την διεργασία της αλλαγής με τρία βήματα

- Ξεπάγωμα (unfreezing): Αντιμέτωπος με ένα δίλημμα, ένα άτομο ή μία ομάδα αναγνωρίζει την ανάγκη για αλλαγή
- Αλλαγή (changing): Γίνεται διάγνωση της κατάστασης και νέα μοντέλα συμπεριφοράς αναπτύσσονται και ερευνώνται
- Επαναπάγωμα (refreezing): Η εφαρμογή της νέας συμπεριφοράς αξιολογείται, και αν εξελιχθεί, καταλήγει να υιοθετείται.



ΣΧΗΜΑ 21 ΕΡΕΥΝΑ ΔΡΑΣΗΣ [15].

Στην παραπάνω εικόνα μπορούμε να δούμε τα βήματα και τις διεργασίες που απαιτούνται σε μια σχεδιασμένη αλλαγή μέσω έρευνας δράσης. Η έρευνα δράσης αποτυπώνεται σαν μια κυκλική διεργασία αλλαγών. Ο κύκλος ξεκινάει με μια σειρά από δράσεις σχεδιασμού από τον «πελάτη» και τον «σύμβουλο αλλαγής», οι οποίοι δουλεύουν μαζί. Τα βασικά στοιχεία αυτού του επιπέδου περιλαμβάνουν

- αρχική διάγνωση
- συλλογή δεδομένων
- ανάλυση στα αποτελέσματα
- από κοινού σχεδιασμό

Στη γλώσσα της θεωρίας συστημάτων, η αρχική διάγνωση είναι η φάση της εισόδου δεδομένων, στην οποία ο πελάτης αντιλαμβάνεται τα προβλήματα (τα οποία δεν είχαν προσδιοριστεί μέχρι τώρα), καταλαβαίνει ότι μπορεί να χρειαστεί βοήθεια προκειμένου να εφαρμόσει αλλαγές και μοιράζεται με τον σύμβουλο την διαδικασία της διάγνωσης του προβλήματος.

Στο δεύτερο επίπεδο της έρευνας δράσης είναι η δράση της «μεταμόρφωσης». Σε αυτό το επίπεδο, υπάρχουν δράσεις σχετικές με εκμάθηση διαδικασιών (ίσως στη μορφή ανάλυσης των ρόλων στον οργανισμό) και τη σχεδίαση και εκτέλεση αλλαγών συμπεριφοράς στον οργανισμό. Όπως φαίνεται στην παραπάνω εικόνα, τα σχόλια σε αυτό το επίπεδο θα μετακινηθούν μέσω του “Feedback Loop A” και θα είχαν ως αποτέλεσμα αλλαγή προηγούμενου σχεδιασμού ώστε να φέρουν τις δραστηριότητες εκμάθησης του πελάτη σε μεγαλύτερη συσχέτιση με τους στόχους της αλλαγής. Σε αυτό το επίπεδο περιλαμβάνεται μια δραστηριότητα σχεδιασμού δράσεων που εκτελείται τόσο από τον σύμβουλο όσο και από τον πελάτη. Ύστερα από ένα εργαστήριο ή συνεδρίες εκμάθησης αυτά τα βήματα εκτελούνται κατά τη διάρκεια της εργασίας σαν μέρος του σταδίου μεταμόρφωσης.

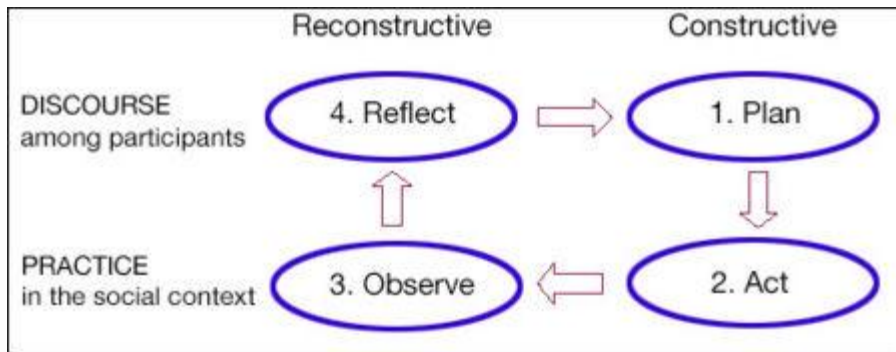
Στο τρίτο επίπεδο της έρευνας δράσης είναι τα αποτελέσματα. Σε αυτό το επίπεδο περιλαμβάνονται πραγματικές αλλαγές στη συμπεριφορά (αν κριθούν αναγκαίες) που είναι εκροή των διορθωτικών δράσεων που γίνανε στο δεύτερο επίπεδο. Τα δεδομένα συγκεντρώνονται πάλι από τον πελάτη έτσι ώστε να μπορεί να αποσαφηνιστεί αν υπήρξε πρόοδος και γίνονται απαραίτητες αλλαγές στις δραστηριότητες εκμάθησης. Μικρές διορθώσεις/τροποποιήσεις τέτοιας φύσης μπορεί να γίνουν στις δραστηριότητες εκμάθησης μέσω του “Feedback Loop B”. Μεγάλες διορθώσεις/τροποποιήσεις και

επαναξιολογήσεις θα επιστρέψουν το έργο στο αρχικό επίπεδο, για βασικές αλλαγές στο πρόγραμμα. Το μοντέλο μεθοδολογίας έρευνας που φαίνεται στην παραπάνω εικόνα ακολουθεί πιστά τον κύκλο του Lewin που περιλαμβάνει σχεδιασμό, δράση και καταμέτρηση αποτελεσμάτων. Επίσης, απεικονίζει άλλα τμήματα του γενικού μοντέλου αλλαγής του Lewin.

Το επίπεδο σχεδίασης είναι μια περίοδος κατανόησης του προβλήματος. Το επίπεδο δράσης είναι η περίοδος της αλλαγής, δηλαδή η δοκιμή νέων μορφών συμπεριφοράς και η προσπάθεια κατανόησης και αντιμετώπισης με τα προβλήματα του συστήματος. Φυσικά, τα όρια μεταξύ των δύο επιπέδων δεν είναι σαφώς καθορισμένα καθώς πρόκειται για ένα συνεχές έργο/διεργασία. Το επίπεδο των αποτελεσμάτων είναι η περίοδος στην οποία δοκιμάζονται νέες συμπεριφορές και τεχνικές στα πλαίσια της καθημερινής δουλειάς, και, αν είναι επιτυχημένες, γίνονται μέλη του συστήματος. Η μεθοδολογία έρευνας είναι επικεντρωμένη στο πρόβλημα, στον πελάτη, στις δράσεις. Υποχρεώνει τον πελάτη να εισέλθει σε μια διαδικασία διάγνωσης, εκμάθησης, εύρεσης προβλημάτων και επίλυσής τους. Τα δεδομένα δεν επιστρέφονται σε μία απλή έγγραφη αναφορά αλλά συζητούνται σε ανοιχτές συνεδρίες. Ο πελάτης και ο σύμβουλος συνεργάζονται στον προσδιορισμό και στην κατηγοριοποίηση συγκεκριμένων προβλημάτων, στην εύρεση μεθόδων για τον προσδιορισμό των αιτιών τους και στην ανάπτυξη σχεδίων για τη ρεαλιστική και πρακτική αντιμετώπισή τους. Η επιστημονική μέθοδος με τη μορφή της συλλογής δεδομένων, σχηματισμού υποθέσεων, δοκιμών υποθέσεων και καταμέτρησης αποτελεσμάτων είναι μέρος της διαδικασίας, όχι όμως σαν πρωτεύον μέσο δράσης. Η μεθοδολογία έρευνας, επίσης, ζωντανεύει έναν μακροπρόθεσμο, κυκλικό, με αυτόματη διόρθωση μηχανισμό για τη διατήρηση και την βελτίωση της αποτελεσματικότητας του συστήματος/οργανισμού του πελάτη με το να αφήνει σε αυτόν εργαλεία και μεθοδολογίες για αυτοανάλυση και αυτοανανέωση.

Παρακάτω μπορούμε να δούμε ένα άλλο σχήμα για τα στάδια της έρευνας δράσης

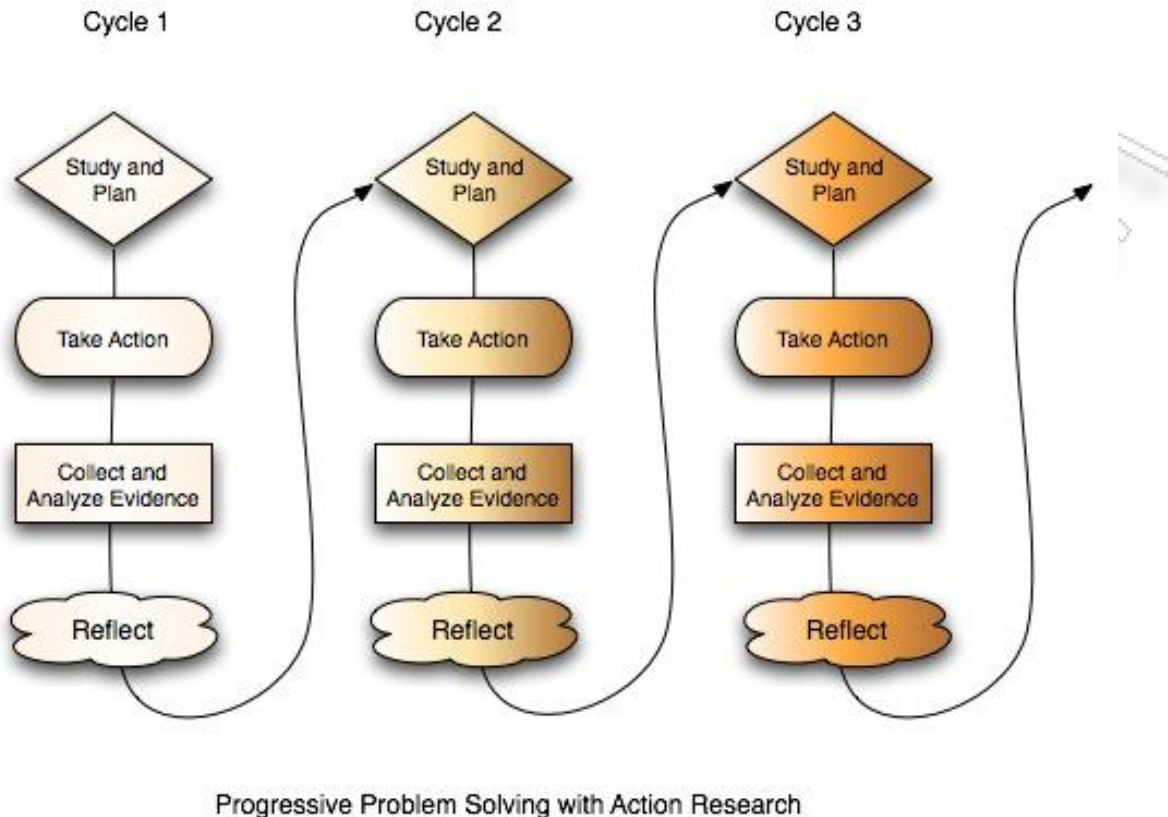




ΣΧΗΜΑ 22 "REFLECTIVE SPIRAL" [15]

Το παραπάνω σχήμα δείχνει έναν κύκλο δράσης και αντίδρασης, διαχωρισμένο στα στάδια του σχεδιασμού, δράσης, παρατήρησης και επιβολής. Κάθε μία από αυτές τις φάσεις εξαρτάται από την προηγούμενη και επηρεάζει άμεσα την επόμενη (έτσι, για παράδειγμα, η φάση δράσης εξαρτάται από το σχεδιασμό και επηρεάζει την παρατήρηση). Ο κύκλος αυτός μπορεί να ξεκινήσει από οποιοδήποτε στάδιο και δεν τερματίζει μετά την συμπλήρωση ενός πλήρους κύκλου αλλά ξεκινάει έναν καινούριο (για αυτό και λέγεται σπιδράλ αντί για κύκλος).

Τέλος, στο παρακάτω σχήμα (από το κέντρο για τη συνεργατική έρευνα δράσης – Center for Collaborative Action Research) μπορούμε να δούμε την επαναληπτική φύση της μεθοδολογίας.



ΣΧΗΜΑ 23 ΠΡΟΟΔΕΥΤΙΚΗ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΜΕΣΩ ΤΗΣ ΕΡΕΥΝΑΣ ΔΡΑΣΗΣ [15]

Οι ερευνητές δρουν και αναζητούν τη μάθηση από τις πράξεις που έχουν ήδη κάνει. Το θέμα της έρευνας δράσης εστιάζεται στις δράσεις που έχουν γίνει, στην αλλαγή και στη θεωρία της αλλαγής που έχουν τα άτομα που την προωθούνε και την στηρίζουνε. Ενώ ο σχεδιασμός της έρευνας δράσης μπορεί να προέρχεται από έναν και μόνο άνθρωπο, οι κοινωνικές δράσεις που τρέχουν χωρίς την συνεργατική βοήθεια άλλων δεν είναι αποτελεσματικές. Για να είναι επιτυχείς, οι ερευνητές που χρησιμοποιούν έρευνα δράσης πρέπει να τη σχεδιάσουν με τέτοιο τρόπο έτσι ώστε να φέρουν ακόμα περισσότερους ανθρώπους (που επηρεάζονται από τα αποτελέσματα της έρευνας) μέσα σε αυτήν. Ο στόχος είναι, προφανώς, η όσο το δυνατόν καλύτερη κατανόηση της κατάστασής τους έτσι ώστε η αλλαγή που θα επέλθει να είναι όσο το δυνατόν θετικότερη. [15]

## 6. Υλοποίηση

---

Ο χώρος της υγείας αποτελεί νευραλγικό τμήμα κάθε σύγχρονης κοινωνίας, αφού αποτελεί μέρος της Κοινωνικής Πρόνοιας κάθε κράτους και σημαντικό παράγοντα της ποιότητας ζωής των πολιτών. Στο χώρο της υγείας δραστηριοποιούνται κρατικοί και ιδιωτικοί φορείς που συνεργάζονται και αλληλοσυμπληρώνονται ως προς τις υπηρεσίες υγείας που προσφέρουν στο κοινωνικό σύνολο.

Το έργο SharpHealth εντάσσεται στην κατηγορία των συστημάτων R.I.S. (Radiology Information System) που χρησιμοποιούνται από τα ακτινολογικά τμήματα των κατά τόπους υγειονομικών μονάδων για την αποθήκευση, διαχείριση και διάθεση ιατρικών - ακτινολογικών δεδομένων. Το σύστημα μας δίνει τη δυνατότητα για παρακολούθηση και προγραμματισμό εξετάσεων ασθενών, καταχώρηση της ιατρικής διάγνωσης και δυνατότητα αποθήκευσης και εμφάνισης ακτινολογικών εξετάσεων.

Το έργο βασίζεται σε τέσσερις πυλώνες για το σχεδιασμό και την υλοποίησή του. Τη διαθεσιμότητα των πληροφοριών, την ασφάλεια των ιατρικών δεδομένων, τη διαλειτουργικότητα μέσω της διεθνούς ιατρικής και τεχνικής πρακτικής και τέλος, τη φιλικότητα προς τον τελικό χρήστη.

### 6.1 Περιγραφή συστήματος

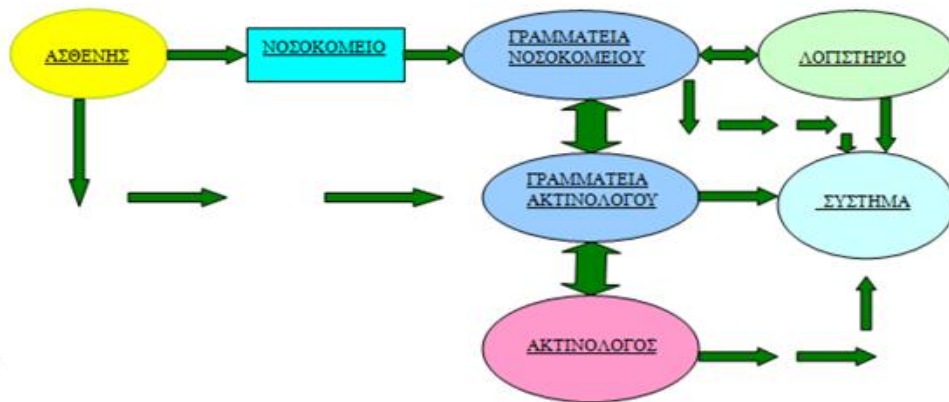
Το σύστημα SharpHealth αποτελεί ένα καταμετρημένο πληροφοριακό σύστημα που απευθύνεται σε μονάδες υγείας, υγειονομικής περιφέρειας. Έχει σαν σκοπό να αυτοματοποιήσει τις διαδικασίες που εμπλέκονται μεταξύ των ακτινολογικών κέντρων, των ιατρών των ασθενών που απαιτούν ακτινογραφίες και των γραμματέων που βρίσκονται σε μία υγειονομική μονάδα καθώς και την αλληλεπίδραση μεταξύ τους.

Οι ασθενείς που εισέρχονται σε μία υγειονομική μονάδα προσκομίζουν την εντολή ακτινογραφίας, μέσω της κεντρικής γραμματείας, αυτή προωθείται στη γραμματεία του ακτινολογικού, όπου ορίζεται ραντεβού για ακτινογραφία. Ο ακτινολόγος ιατρός εκτελεί την ακτινογραφία η οποία αποστέλλεται στο θεράποντα ιατρό του ασθενή για διάγνωση. Το σύστημα SharpHealth δίνει την δυνατότητα επικοινωνίας μεταξύ της κεντρικής γραμματείας, της γραμματείας των ραντεβού και του ακτινολογικού τμήματος της μονάδας, ενώ επίσης και την δυνατότητα επικοινωνίας και ανταλλαγής δεδομένων με άλλες μονάδες της ευρύτερης περιφέρειας.

Η εφαρμογή που αναπτύχθηκε σχετίζεται με τη διαδικασία που ακολουθείται προκειμένου ένας ασθενής, που απευθύνεται στο νοσοκομείο και χρήζει εξέτασης ακτινολόγου, να προωθείται στο συγκεκριμένο τμήμα και να πραγματοποιεί τις εκάστοτε ιατρικές εξετάσεις (ακτινογραφίες) από τον αρμόδιο γιατρό. Κρίνεται σκόπιμο να αναφέρουμε ότι το πληροφοριακό σύστημα που αναπτύσσεται στο νοσοκομείο δίνει τη δυνατότητα της συνεργασίας όλων των γιατρών της ίδιας ειδικότητας

(ακτινολόγων) που δρουν σε διαφορετικά νοσοκομεία - ιατρικά κέντρα τα οποία και συνεργάζονται μεταξύ τους.

Τη συγκεκριμένη εφαρμογή θα είναι σε θέση να χρησιμοποιούν οι γραμματείες νοσοκομείων, οι γραμματείες ακτινολογικού τμήματος κάθε νοσοκομείου - ιατρικού κέντρου, οι γιατροί ακτινολόγοι, ο διαχειριστής του συστήματος και τέλος ο υπεύθυνος διαχειριστή συστήματος. Προκειμένου να γίνει πιο κατανοητός ο τρόπος με τον οποίο αλληλεπιδρούν οι χρήστες μεταξύ τους αλλά και τρόπος με τον οποίο θα πρέπει να συνεργάζονται μεταξύ τους παρουσιάζεται το πιο κάτω σχετικό διάγραμμα (εικόνα 1).



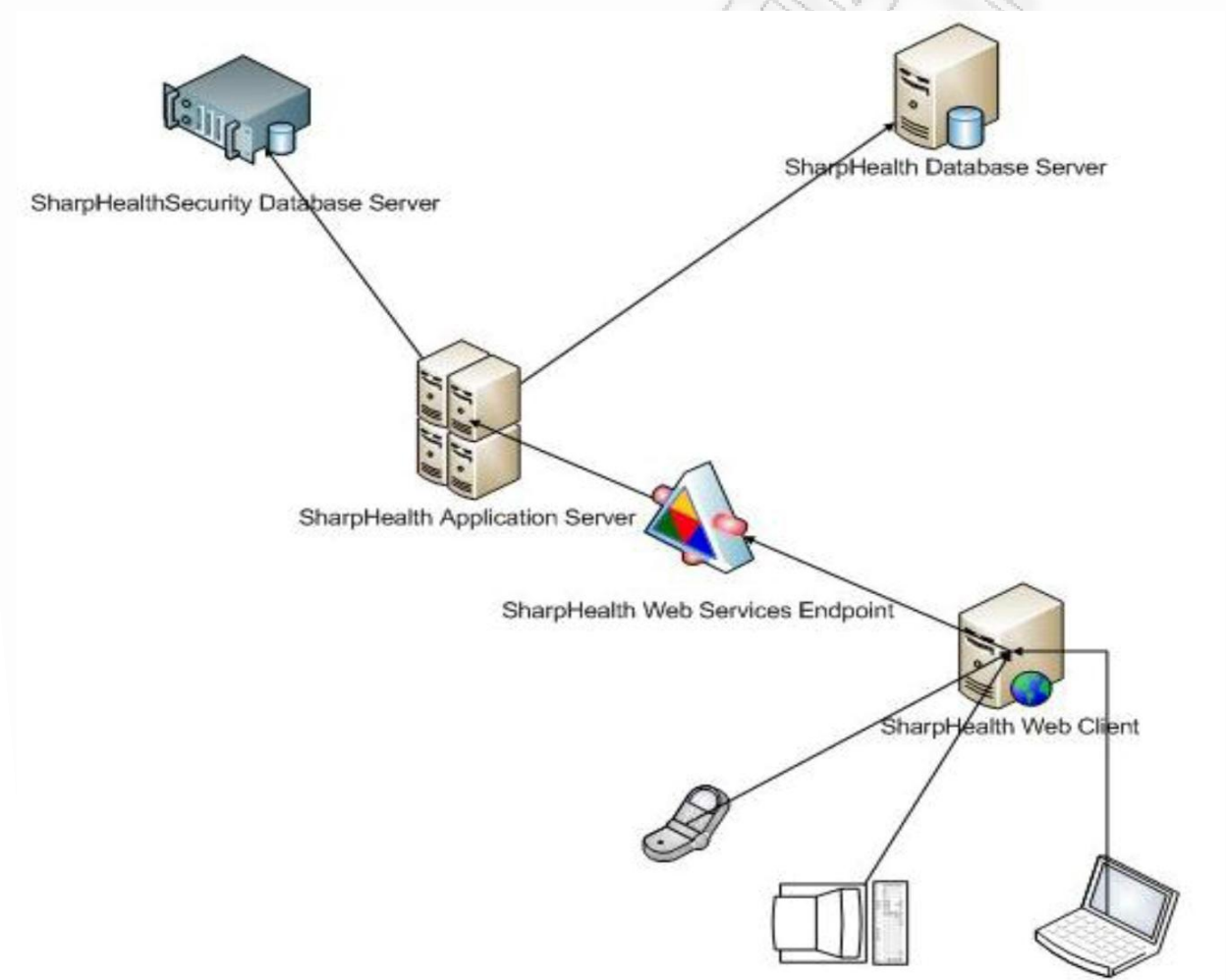
ΕΙΚΟΝΑ 1 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΧΡΗΣΤΩΝ

Όπως φαίνεται στην εικόνα 1 ο ασθενής που προσέρχεται στο νοσοκομείο, προκειμένου να πραγματοποιήσει την ακτινολογική εξέταση, απευθύνεται στη γραμματεία του νοσοκομείου. Με τη σειρά της η γραμματεία νοσοκομείου αφού πραγματοποιήσει τις προβλεπόμενες για αυτή διαδικασίες, που θα αναφερθούν με λεπτομέρεια στο επόμενο κεφάλαιο, θα ενημερώσει το λογιστήριο για τον τρόπο πληρωμής της εξέτασης και στη συνέχεια θα προωθήσει τον ασθενή στη γραμματεία ακτινολογικού. Η γραμματεία ακτινολογικού θα πρέπει με τη σειρά της να κλείσει ραντεβού στον ασθενή για το πότε θα μπορούσε να πραγματοποιήσει τη συγκεκριμένη ιατρική εξέταση βάσει της διαθεσιμότητας του γιατρού. Επιπλέον μέσω του διαγράμματος γίνεται φανερό ότι η γραμματεία νοσοκομείου, αλλά και ακτινολογικού τμήματος βρίσκονται σε άμεση επικοινωνία ανταλλαγής στοιχείων όπως και το ίδιο ισχύει μεταξύ των ακτινολόγου - γραμματείας ακτινολόγου και λογιστηρίου - γραμματείας νοσοκομείου.

## 6.2 Αρχιτεκτονική του συστήματος

### 6.2.1 Αρχιτεκτονική δικτύου

Το πληροφοριακό σύστημα SharpHealth ακολουθεί μία συγκεκριμένη αρχιτεκτονική, στην οποία βασίζεται η λειτουργικότητα του όλου συστήματος καθώς και η επικοινωνία μεταξύ των διαφόρων υποσυστημάτων αυτού. Η σωστή εγκατάσταση και επικοινωνία μεταξύ των υποσυστημάτων αυτών εξασφαλίζει σωστή και ασφαλή λειτουργία του συστήματος. Το σύστημα SharpHealth αποτελείται από τα εξής: ΤΟ Server της βάσης δεδομένων (Database Server), τον Server ασφαλείας (Security Server), Τον Server της εφαρμογής (Application Server) και τους πελάτες της εφαρμογής (Web Clients), οι οποίοι συνδέονται στον Application Server για να στέλνουν και να δέχονται πληροφορίες από τις βάσεις δεδομένων. Η αναπαράσταση του συστήματος φαίνεται στο παρακάτω σχήμα (Σχήμα 18).



ΣΧΗΜΑ 24 ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΥΣΤΗΜΑΤΟΣ

Τα πρωτόκολλα επικοινωνίας που θα χρησιμοποιήσουμε βασίζονται στο TCP/IP & Ethernet στο φυσικό στρώμα. Η αρχιτεκτονική του συστήματος συνίσταται από τρία επιμέρους τμήματα - αρχιτεκτονικές.

#### ***Intra-site connection***

Η υποδομή που επιτρέπει τους απομακρυσμένους Η/Υ να συνδεθούν φυσικά, και να επικοινωνήσουν με τους κεντρικούς υπολογιστές (servers).

#### ***Core Infrastructure***

Η υποδομή που επιτρέπει την επικοινωνία από το computer room με τους Η/Υ του ΠΓΝ και με τα απομακρυσμένα sites.

#### ***Site Infrastructure***

Η υποδομή που επιτρέπει την επικοινωνία των επιμέρους Η/Υ μέσα στο κάθε site.

#### ***Η δομή του δικτύου***

Η υλοποίηση του δικτύου είναι «αστέρας». Απαιτείται δομημένη καλωδίωση κατηγορίας Cat. 5e για Ethernet δίκτυα. Το δίκτυο είναι switched με distribution layer στο ΠΓΝ και access layer σε όλα τα sites. Για την επικοινωνία μεταξύ των sites απαιτείται σύνδεση με γραμμή ADSL (ελάχιστη απαίτηση 128 kbps Upstream) ή κατ' ελάχιστον ISDN 128 kbps.

### **6.1.2 Αρχιτεκτονική εφαρμογής**

Η εφαρμογή σχεδιάστηκε και υλοποιήθηκε σε Windows 2008 Server R2 με multi-tier και Web based αρχιτεκτονικές κυρίως για λόγους ασφαλείας και δευτερευόντως για λόγους ταχύτητας και ευκολίας πρόσβασης ανεξάρτητα από τον τελικό client.

#### **Εφαρμογή του Web Client**

Η εφαρμογή δοκιμάστηκε και λειτουργεί στους τρεις δημοφιλέστερους Web Browsers (IE 6 and higher, Mozilla Firefox 3.0 and higher, Chrome 10 and higher & Opera 6 and higher).

#### **Εφαρμογή του Application Server**

Το κυρίως επίπεδο είναι ΤΟ επίπεδο εφαρμογής στο οποίο εκτελείται η εφαρμογή. Το επίπεδο της εφαρμογής αποτελεί τον συνδετικό κρίκο του όλου συστήματος και παρέχει τις υπηρεσίες που χρησιμοποιούνται για την ανταλλαγή δεδομένων από τις βάσεις, ενώ καθιστά δυνατή την επικοινωνία της εφαρμογής στο διαδίκτυο με σκοπό την ανταλλαγή δεδομένων με άλλα νοσοκομεία που χρησιμοποιούν την εφαρμογή αυτή, και την εξυπηρέτηση των πελατών (Web Clients) που συνδέονται σε αυτήν.

Ως πλατφόρμα για την οποία έχει γραφεί η εφαρμογή μας χρησιμοποιείται το Microsoft .NET Framework 4 σε Microsoft Visual Studio 2010. Για την επικοινωνία με τους clients απαιτείται η χρήση Web Server ο οποίος υλοποιείται από τον Internet Information Server 8.0 που είναι ενσωματωμένος στο λειτουργικό των Windows 2008 Server R2.

Επιπλέον χρησιμοποιείται η υπηρεσία Windows Communication Foundation η οποία αποτελείται από ένα σύνολο από βιβλιοθήκες για Service-Oriented Αρχιτεκτονικές και μας επιτρέπει να κάνουμε "ορατές" διάφορες συναρτήσεις και λειτουργίες της εφαρμογής μας μέσω του διαδικτύου. Για την προσπέλαση αυτών των αντικειμένων χρησιμοποιούνται τα πρωτόκολλα επικοινωνίας WSDL (web services definition language) και η ανταλλαγή δεδομένων γίνεται μέσω της τεχνολογίας SOAP (simple object access protocol).

Μέσω του Workflow Foundation η ροή εργασιών και οι συναρτήσεις καλούνται με συγκεκριμένη αλληλουχία έτσι ώστε μια διεργασία για να εκτελεστεί έχει σαν προαπαιτούμενο να έχει εκτελεστεί κάποια άλλη πρώτα. Για να αποθηκεύουμε δεδομένα του Workflow χρησιμοποιούμε την βάση δεδομένων Persistence Service η οποία καταγράφει τις κινήσεις που γίνονται ακριβώς στην βάση Logging. Για την πρόσβαση στη βάση δεδομένων χρησιμοποιείται LINQ , μέσω του οποίου καλούνται αντικείμενα που βρίσκονται στην βάση.

Σε ξεχωριστό server της υλοποιείται Active Directory Directory Service για την αυθεντικοποίηση των υπολογιστών και των χρηστών αλλά και την απόδοση πολιτικών για ολόκληρο το σύστημα (System wide). Στον ίδιο server της υλοποιείται Domain Name Service (DNS) Server με Active Directory Integrated Zones και Active Directory Authenticated Response προκειμένου να μην απαντά σε Name Resolution ερωτήματα clients που δεν έχουν γίνει authenticated από το Active Directory Service του Domain. Στον ίδιο υπολογιστή εκτελείται και η υπηρεσία του Domain Host Configuration Protocol DHCP στην οποία για λόγους ασφαλείας κάθε Client ο οποίος επιτρέπεται να εξυπηρετηθεί από τον DHCP πρέπει να έχει υπάρχει εκ των προτέρων σύνδεση τις διεύθυνσης που θα του αποδοθεί με την Media Access Control Address του δικτυακού του interface. Τελικά σε αυτόν τον υπολογιστή υλοποιείται και το service του Certificate Server για εσωτερική χρήση όπως θα περιγραφεί στο κεφάλαιο ασφάλεια.

### **Βάση δεδομένων του συστήματος**

Η Βάση δεδομένων του συστήματος αποτελείται από το σύνολο δεδομένων του όλου συστήματος , τα οποία έχουν ενταχθεί σε μια δομή ώστε να μπορούν να χρησιμοποιηθούν από τους διάφορους χρήστες. Τα δεδομένα αυτά καλούνται μέσω της βάσης από την κεντρική εφαρμογή και στέλνονται στους χρήστες ,ενώ ταυτόχρονα η βάση αποθηκεύει όποια νέα δεδομένα σταλούν σε αυτήν.

Το επίπεδο του συστήματος διαχείρισης βάσεως δεδομένων υλοποιείται σε ξεχωριστό υπολογιστή από το επίπεδο εφαρμογής κυρίως για λόγους ασφαλείας εξαιτίας του ευαίσθητου των δεδομένων που χειρίζεται η εφαρμογή. Το λειτουργικό σύστημα που στο οποίο έχει εγκατασταθεί το σύστημα διαχείρισης βάσης δεδομένων είναι ο Windows 2008 Server R2 και το σύστημα διαχείρισης δεδομένων είναι ο Microsoft SQL Server 2008. Όλοι οι servers σε φυσικό επίπεδο έχουν NTFS file system

προκειμένου να μπορούμε να ελέγχουμε την πρόσβαση των χρηστών σε επίπεδο αρχείου και να μπορούμε να έχουμε το μπορούμε να καταγράψουμε την πρόσβαση σε αρχεία μέσω του συστήματος καταγραφής των Windows.

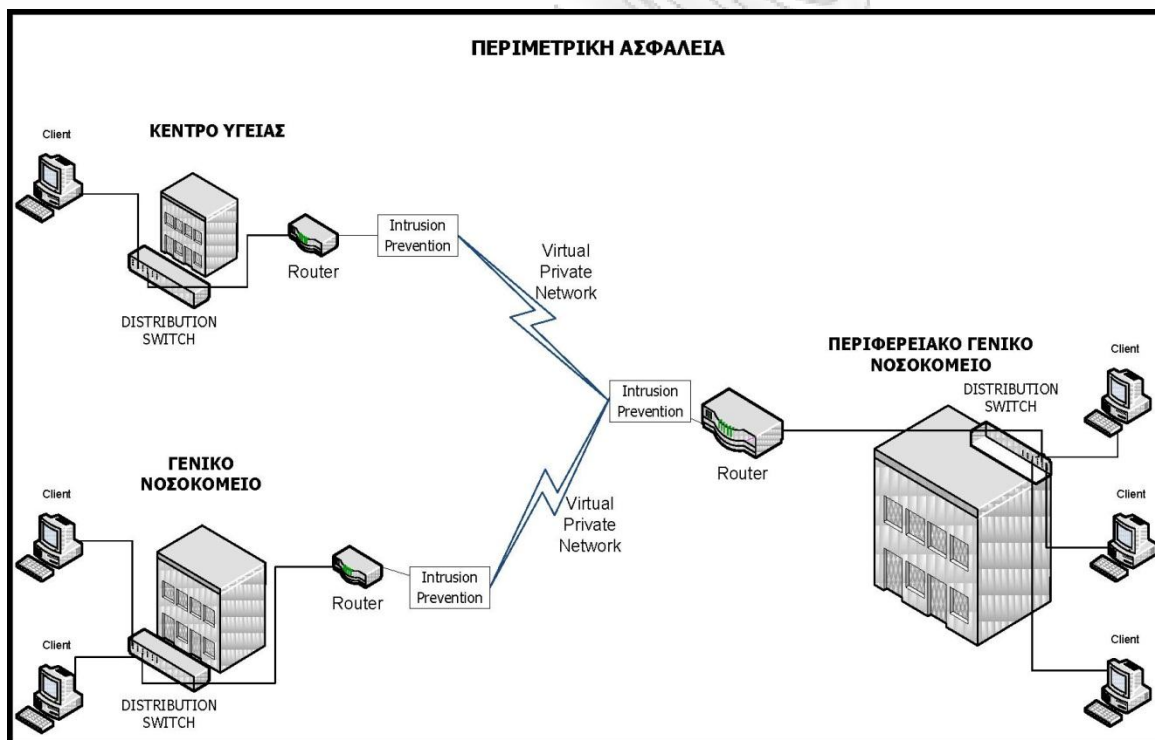
Επίσης για την ασφάλεια του συστήματος και την ασφαλή ανταλλαγή των δεδομένων μεταξύ της εφαρμογής και των χρηστών , όλες οι επικοινωνίες κρυπτογραφούνται με το SSL πιστοποιητικό ασφαλείας.

### Ασφάλεια του συστήματος και έλεγχος πρόσβασης

Η ασφάλεια του συστήματος υλοποιείται μέσω πολλαπλών επιπέδων ασφαλείας.

#### Περιμετρική ασφάλεια

Η περιμετρική ασφάλεια υλοποιείται με την δημιουργία Virtual Private Networks VPN με χρήση πιστοποιητικών κρυπτογράφησης και αυθεντικοποίησης σε επίπεδο συνδεσιμότητας μονάδας. Κάθε μονάδα Κέντρο Υγείας κτλ που θέλει να συνδεθεί για να χρησιμοποιήσει την εφαρμογή, συνδέεται μέσω VPN που δημιουργείται πάνω σε μια σύνδεση στο Internet. Για την φυσική σύνδεση προτείνεται ένα bandwidth όχι λιγότερο από 128Kbps. Πάνω στον router επικοινωνίας απαιτούνται intrusion detection δυνατότητες και δυνατότητα VPN.

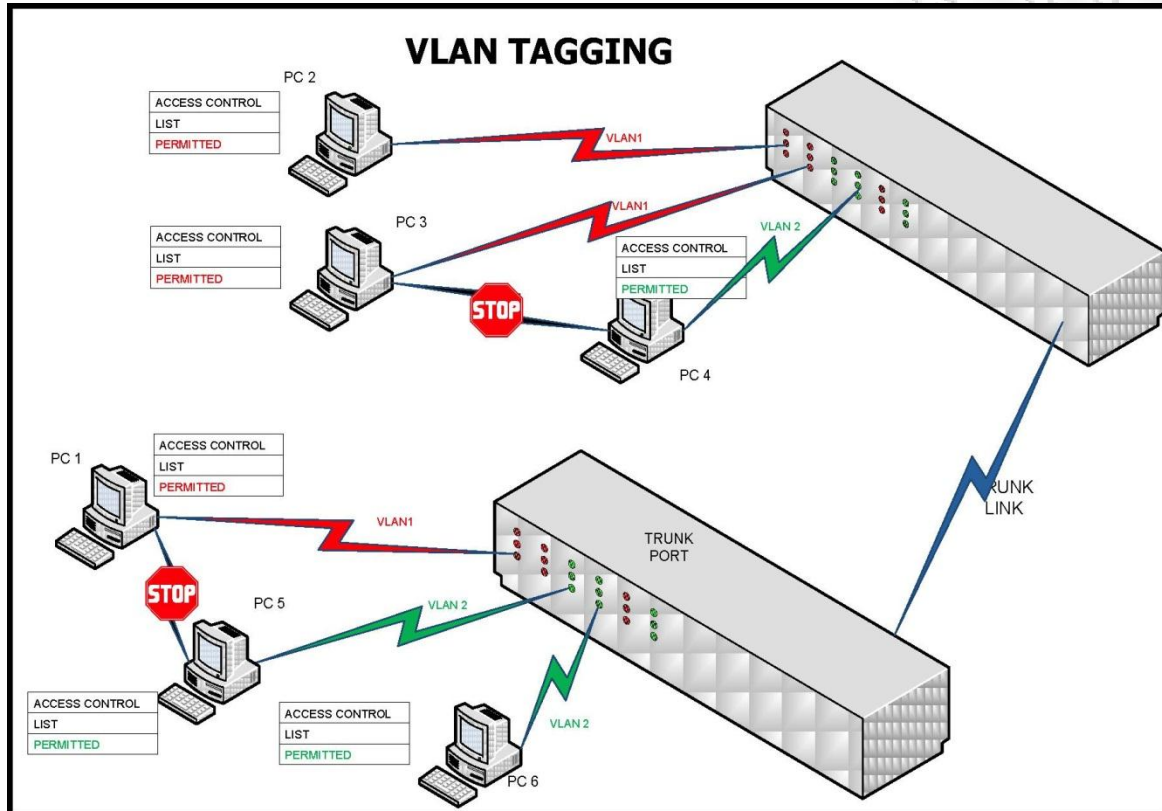


ΣΧΗΜΑ 25 ΠΕΡΙΜΕΤΡΙΚΗ ΑΣΦΑΛΕΙΑ

#### Εξωτερική ασφάλεια



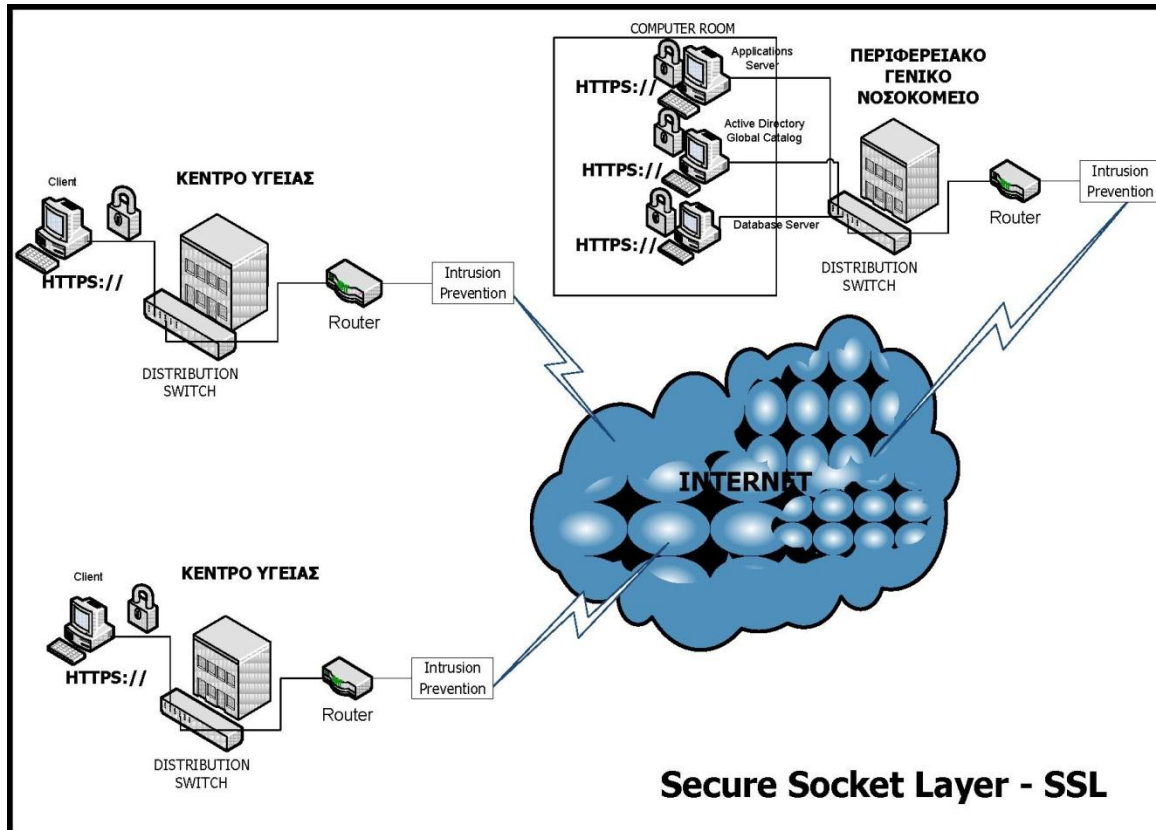
Προκειμένου να εξασφαλίσουμε ότι δεν μπορεί κάποιος να υποκλέψει δεδομένα εισάγοντας κάποιον καταγραφέα επικοινωνιών δικτύου σε κάποιο σημείο του δικτύου μας (Network Sniffer) προτείνεται η δημιουργία VLAN και access lists στα switches του δικτύου μας προκειμένου να επιτρέπουν μόνον την επικοινωνία μεταξύ των εγκεκριμένων υπολογιστών αλλά και πορτών των δικτύων μας.



ΣΧΗΜΑ 26 ΕΞΩΤΕΡΙΚΗ ΑΣΦΑΛΕΙΑ

### Ασφάλεια επικοινωνιών εφαρμογής

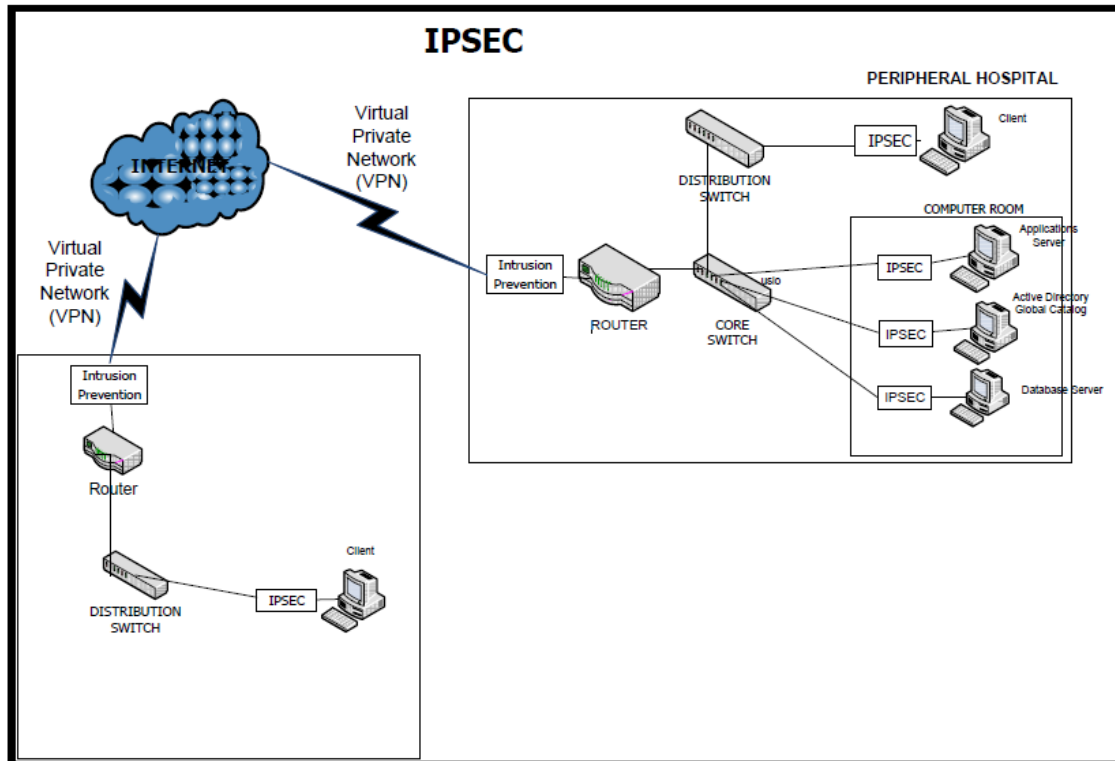
Για να εξασφαλίσουμε την ασφαλή μεταφορά των δεδομένων μέσα από δίκτυα όχι πάντα ασφαλή, όλες οι επικοινωνίες μεταξύ του application server και των clients γίνονται με ασφαλές HTTP πρωτόκολλο (HTTPS) με την χρήση του Transport Layer Protocol TLS - Secure Socket Layer Protocol SSL (V 3.0) χρήση πιστοποιητικών κρυπτογράφησης και αυθεντικοποίησης από τον Windows Certificate Server.



ΣΧΗΜΑ 27 ΠΡΟΣΤΑΣΙΑ ΕΞΥΠΗΡΕΤΗΤΩΝ ΔΙΑΔΙΚΤΥΟΥ

### Ασφάλεια μεταφοράς δεδομένων υπολογιστικών συστημάτων

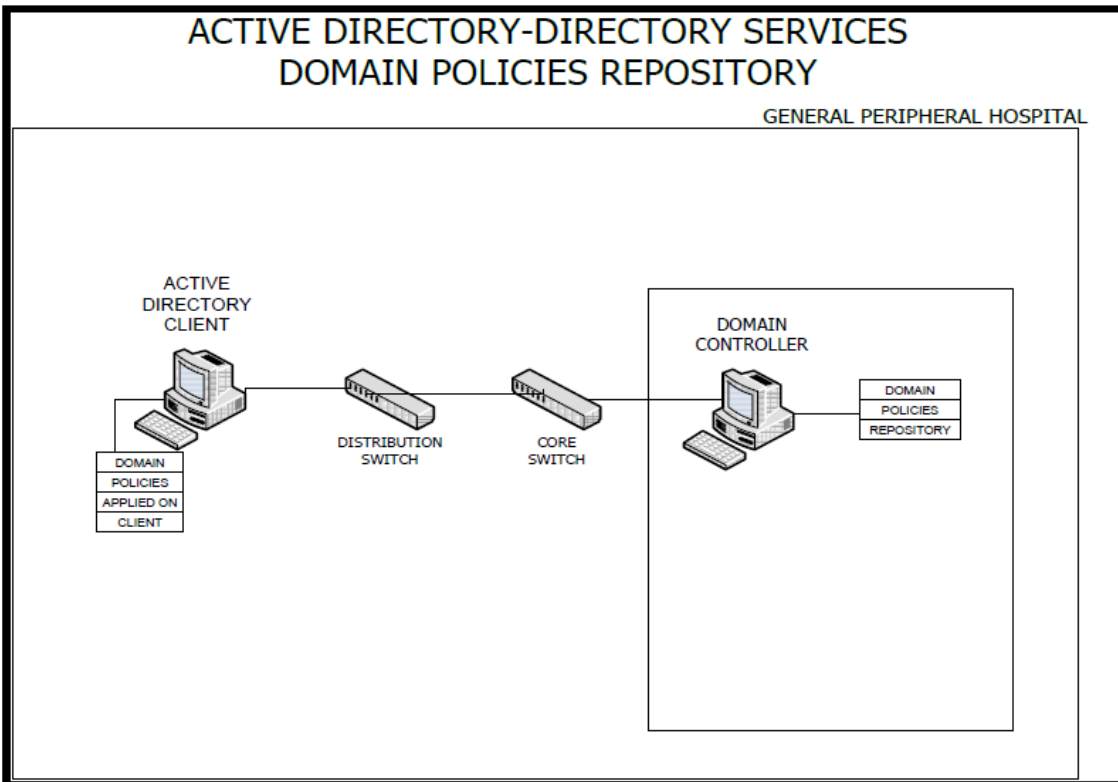
Όλοι οι υπολογιστές που συμμετέχουν στο Υπολογιστικό Σύστημα (clients και servers πλην από τον Server που έχει την υπηρεσία του DHCP ) θα έχουν εγκατεστημένο τον IPSec driver προκειμένου ολόκληρη η επικοινωνία να γίνεται κρυπτογραφημένη και ελεγχόμενη ως προς τον αποστολέα παραλήπτη. Ο IPSec driver εγκαθίσταται χαμηλά στο TCP/IP stack και εμποδίζει την επικοινωνία (ακόμα και σε επίπεδο discovery) του υπολογιστή με άλλους υπολογιστές οι οποίοι δεν έχουν τα απαιτούμενα credentials. Έτσι ακόμα και στην περίπτωση που κάποιος αποκτήσει πρόσβαση σε κάποιο υπολογιστή του συστήματος και εγκαταστήσει ένα Trojan Horse για παράδειγμα προσπαθώντας να υποκλέψει και να μεταφέρει πληροφορίες που διέρχονται από το συγκεκριμένο σύστημα το IPSec καθιστά την επικοινωνία μεταξύ συστημάτων που δεν έχουν αποδεκτά μεταξύ τους credentials αδύνατη.



ΣΧΗΜΑ 28 IP SECURITY

### Ασφάλεια πρόσβασης στην εφαρμογή

Τα Active Directory Domain Directory Services είναι ένα Security Boundary που μας επιτρέπει να ελέγχουμε κεντρικά την πρόσβαση στους πόρους του συστήματος. Αναλυτικά μας δύναται η δυνατότητα να ελέγξουμε ποιοι χρήστες μπορούν να χρησιμοποιήσουν ποιους τερματικούς σταθμούς, ο κάθε χρήστης ποιες εφαρμογές μπορεί να χρησιμοποιήσει και τελικά τι επίπεδο πρόσβασης έχει στους πόρους του συστήματος για παράδειγμα εάν μπορεί να διαβάσει ένα αρχείο, εάν μπορεί να κάνει αλλαγές κλπ. Επίσης εάν έχει την δυνατότητα να χρησιμοποιήσει τις USB ή σειριακές θύρες προκειμένου να εξαγάγει ένα αρχείο ή να στείλει σε ένα άλλο υπολογιστικό σύστημα κλπ. Οι διαχειριστές του συστήματος προσδιορίζουν για κάθε χρήση τα δικαιώματα που έχει για ολόκληρο το υπολογιστικό σύστημα. Επίσης μέσα από το Active Directory Directory Services προσδιορίζονται και οι τεχνικοί ρόλοι όπως ο Domain Administrator και οι Backup Operators οι οποίοι έχουν την δυνατότητα να πάρουν Backup την βάση δεδομένων αλλά δεν μπορούν να την διαβάσουν καθότι τα αρχεία είναι κρυπτογραφημένα (μόνο με την συνεργία του Domain Administrator).

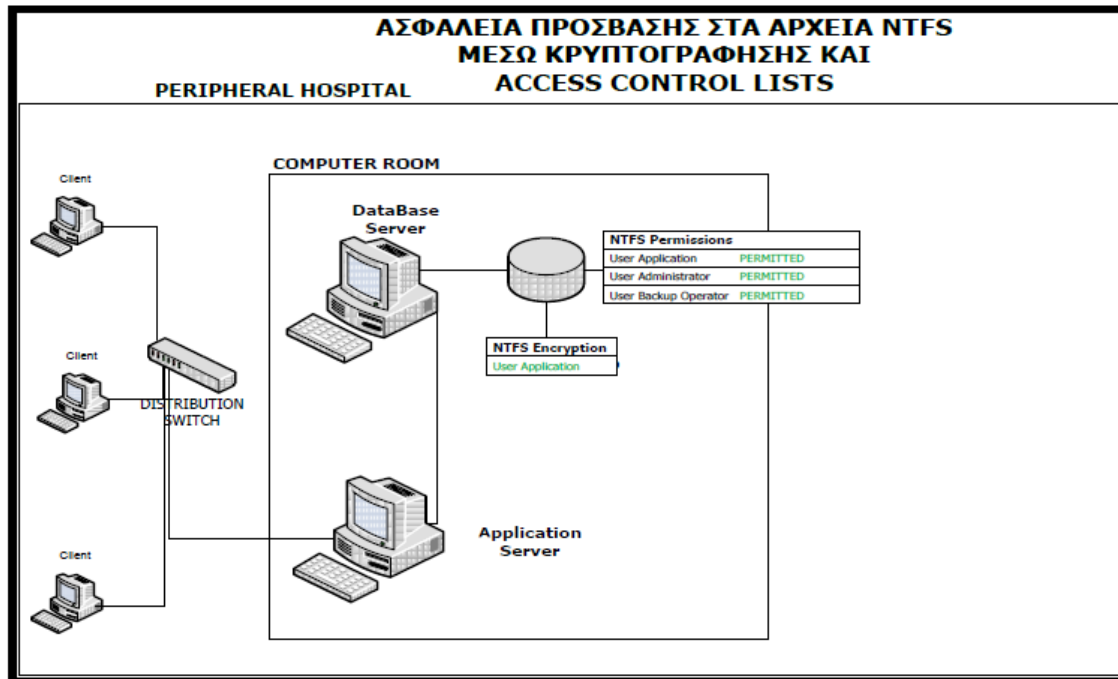


ΣΧΗΜΑ 29 ΚΑΤΑΛΟΓΟΣ ΧΡΗΣΤΩΝ ΚΑΙ ΑΣΦΑΛΕΙΑ

### **Ασφάλεια πρόσβασης στα αρχεία NTFS, Windows Encryption System – Διαχωρισμός ρόλων για την διαχείριση δεδομένων**

Το σύστημα διαχείρισης των αρχείων που θα χρησιμοποιηθεί είναι το New Technology File System (NTFS) που μας επιτρέπει την λεπτομερή διαχείριση των πόρων σε επίπεδο κάθε ενός πόρου ξεχωριστά αλλά μας επιτρέπει και ομαδική διαχείριση. Για ΤΟ NTFS οι πόροι του συστήματος είναι τα αρχεία, οι φάκελοι και οι εκτυπωτές. Ειδικά για τα αρχεία του συστήματος της βάσης δεδομένων οι μοναδικοί που έχουν πρόσβαση είναι η εφαρμογή, οι Backup Operators και οι Domain Administrators αποκλειστικά. Επίσης με Windows Encryption System όλα τα αρχεία της βάσης δεδομένων παραμένουν κρυπτογραφημένα και μόνον η εφαρμογή που είναι και ο owner των αρχείων μπορεί να τα αποκρυπτογραφήσει, κατόπιν ζήτησης κάποιου χρήστη της εφαρμογής ο οποίος έχει δικαίωμα να τα προσπελάσει προκειμένου να κάνει χρήση αυτών.

Τελικά με τον διαχωρισμό ρόλων επιτρέπουμε μόνον στον Administrator να έχει και πρόσβαση στα αρχεία αλλά και την δυνατότητα να πάρει το ownership προκειμένου να τα αποκρυπτογραφήσει. Οι Backup Operators έχουν την δυνατότητα να διαβάζουν και να γράφουν τα αρχεία της βάσης δεδομένων χωρίς όμως να έχουν την δυνατότητα να τα αποκρυπτογραφήσουν δηλαδή πρακτικά δεν μπορούν να τα διαβάσουν παρότι μπορούν να τα εξάγουν.



ΣΧΗΜΑ 30 ΑΣΦΑΛΕΙΑ ΣΕ ΕΠΙΠΕΔΟ ΣΚΛΗΡΟΥ ΔΙΣΚΟΥ

## Εγκατάσταση του συστήματος

### A. η εγκατάσταση του web site

Για την εγκατάσταση του Web Site χρειαζόμαστε να κάνουμε τα εξής: Καταρχήν χρειάζεται να κάνουμε εγκατάσταση του αρχείου Sharphealthwebsite.msi (Microsoft installer) και στην συνέχεια ρυθμίζουμε, αν χρειαστεί, το αρχείο Web.config έτσι ώστε να είναι ικανό να δει τα αντίστοιχα Web Services.

### B. η εγκατάσταση του application server

Για την εγκατάσταση του Application Server απαιτείται εγκατάσταση του αρχείου Sharphealthapplicationserver.msi και κατόπιν χρησιμοποιούμε το Certificate το οποίο και έχει γίνει Sign από έναν Trusted Certificate Authority. Στην συνέχεια το Certificate αυτό, το περνάμε στο Local Certificate Store του μηχανήματος με τη χρήση του εργαλείου Certmgr.exe.

Κατόπιν, χρησιμοποιούμε το εργαλείο Find private key προκειμένου να δώσουμε Read Permissions στον Web Server. Το Virtual Directory που έχει δημιουργηθεί το κάνουμε configure σαν application πραγματοποιώντας τους τις κατάλληλες ρυθμίσεις για να εκτελεστεί.

Τέλος ρυθμίζουμε το αρχείο Web.config έτσι ώστε να βλέπει το Certificate που περάσαμε.

### Γ. η εγκατάσταση του database server

Για την εγκατάσταση του Database Server απαιτείται πρώτα η εγκατάσταση του SQL Server 2008. Εν συνεχεία , τα αρχεία Sharphealth.mdf και SharphealthWF.mdf τα κάνουμε attach στο Database Server.

#### Δ. η εγκατάσταση του database security server

Για την εγκατάσταση του Database Server απαιτείται πρώτα η εγκατάσταση του SQL Server 2005. Εν συνεχεία , το αρχείο Sharphealthsecurity.mdf το κάνουμε attach στο Database Server.

## 6.2 Ανάλυση των επιμέρους λειτουργιών του συστήματος

Όταν κάποιος θελήσει να χρησιμοποιήσει την εφαρμογή του συστήματος SharpHealth , πρώτα εισέρχεται στην οθόνη της εισαγωγής ονόματος και κωδικού πρόσβασης (Εικόνα 2) , όπου ο χρήστης εισάγοντας τα στοιχεία του εισέρχεται στο σύστημα και ανάλογα με τον ρόλο του , έχει πρόσβαση στις αντίστοιχες υπηρεσίες και δεδομένα της βάσης δεδομένων.

**SharpHealth**

Παρακαλώ  
πληκτρολογήστε τα  
στοιχεία σας για να  
συνδεθείτε

Όνομα  
χρήστη

Κωδικός

**Καλώς ήρθατε!**

Το Sharphealth είναι ένα πληροφο υποβληθεί σε μία ακτινογραφία.

**Ακτινολογικές εξετάσεις**

- Ακτινογραφία θώρακος
- Μαγνητική
- Αξονική
- Υπέρηχοι
- Μαστογραφία
- γ-camera

ΕΙΚΟΝΑ 2 ΟΘΟΝΗ ΕΙΣΟΔΟΥ

#### Η εφαρμογή από την πλευρά του χρήστη της κεντρικής γραμματείας

Ο πρώτος τύπος χρήστη είναι η κεντρική γραμματεία του εκάστοτε νοσοκομείου. Ο υπάλληλος της γραμματείας, πρώτα συνδεθεί στο σύστημα πληκτρολογώντας τα στοιχεία του, κατευθύνεται στην κεντρική σελίδα από την οποία έχει τις εξής επιλογές: Μπορεί να επιλέξει να εισάγει έναν νέο ασθενή που έρχεται στο νοσοκομείο για πρώτη φορά , εισάγοντας τα στοιχεία του.

SharpHealth

Καλώς ήρθες, **Σεκρεταριάδου Κατερίνα**  
 Ειδικότητα: **Γραμματεία Ιατρού**  
 Τμήμα: **Ορθοπαιδικό**  
 Νοσοκομείο: **Ευαγγελισμός**

[Αποσύνδεση](#)

Εισαγωγή Ασθενή ▶  
 Αναζήτηση Ασθενή ▶  
 Αίτηση Ακτινογραφίας ▶  
 Αρχική Σελίδα

Εισαγωγή στοιχείων καρτέλας Ασθενή  
 Παρακαλώ εισάγεται όλα τα παρακάτω πεδία:

Όνομα:

Επίθετο:

Όνομα Μητέρας:

Όνομα Πατέρα:

Ημ.Γέννησης: 

< Ιούνιος 2011 >						
Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Διεύθυνση:

Εθνικότητα: Ευ:  Άλλη:

Φύλο: Άντρας:  Γυναίκα:

Ασφάλεια:

Αριθμός ασφάλισης:

Κινητό:

ΕΙΚΟΝΑ 3 ΟΘΟΝΗ ΓΡΑΜΜΑΤΕΙΑΣ

Μπορεί επίσης να κάνει αναζήτηση για έναν ασθενή που ήδη εγγεγραμμένος, δηλαδή έχει ξαναέρθει σε ένα από τα νοσοκομεία που έχουν την εφαρμογή SharpHealth. Έτσι θα μπορεί να δει ή και να επεξεργαστεί τα στοιχεία του. Με την αναζήτηση εμφανίζεται η λίστα των ασθενών που ικανοποιούν τα κριτήρια αυτά και αν επιλεγεί κάποιος από αυτούς, εμφανίζονται τα στοιχεία του.

Στην συνέχεια η κεντρική γραμματεία μπορεί να επιλέξει να κάνει μια αίτηση για ακτινογραφία δίνοντας τα στοιχεία του πελάτη που θέλει να κάνει την ακτινογραφία για αναζήτηση. Έπειτα επιλέγει

το νοσοκομείο στο οποίο θα γίνει η ακτινογραφία , το ακτινολογικό τμήμα στο οποίο θα εκτελεστεί καθώς και το αντίστοιχο μηχάνημα.

Όπως αναφέρθηκε και στο κεφάλαιο των παραδοχών θεωρούμε ότι κάθε νοσοκομείο - ιατρικό κέντρο παρουσιάζει μια κεντρική γραμματεία. Ο ρόλος της κεντρικής γραμματείας νοσοκομείου είναι να κάνει αποτύπωση του λόγου εισαγωγής του ασθενή στο (περιφερειακό) νοσοκομείο - ιατρικό κέντρο ανοίγοντας του καρτέλα στην οποία και θα αποτυπώνονται με κάθε λεπτομέρεια τα προσωπικά του στοιχεία .

Στην περίπτωση που ο ασθενής έχει ήδη καρτέλα τότε η γραμματεία νοσοκομείου οφείλει να ενημερώσει το ιστορικό του ασθενούς στο αντίστοιχο πεδίο της καρτέλας του για το είδος της επίσκεψής του (ακτινογραφία).

Η γραμματεία νοσοκομείου λοιπόν έχοντας πρόσβασή στις καρτέλες ασθενών θα απευθύνεται και θα ελέγχει αν ο ασθενής είναι νέος για το νοσοκομείο ή όχι και θα του ανοίγει καρτέλα ή απλά θα την ενημερώνει με περαιτέρω ιατρικές πληροφορίες αντίστοιχα. Στη συνέχεια η γραμματεία νοσοκομείου θα πρέπει να προωθεί τον ασθενή στο αντίστοιχο τμήμα (ακτινολογικό) στο οποίο και θα πραγματοποιήσει την ιατρική εξέταση (ακτινογραφία) που απαιτείται να κάνει.

Επιπροσθέτως, αρμοδιότητα της γραμματείας νοσοκομείου είναι να ελέγξει με ποιο ταμείο είναι συμβεβλημένος ο ασθενής και να ενημερώνει την καρτέλα του, αλλά και τον ίδιο για τον τρόπο πληρωμής της εξέτασης που θα πραγματοποιήσει. Στην περίπτωση που ο ασθενής προσέρχεται στο νοσοκομείο παρουσιάζοντας έκτακτη ανάγκη αντιμετώπισης το πρόβλημα υγείας που φέρει η γραμματεία θα απευθύνεται στο πεδίο της εφαρμογής όπου απλά θα σημειώνει την ώρα προσέλευσης του ασθενούς στο νοσοκομείο και την αιτία, εάν αυτή είναι γνωστή. Στη συνέχεια και έχοντας αντιμετωπιστεί το πρόβλημα του και έχοντας λάβει ενημέρωση από τον αρμόδιο θεράποντα ιατρό, θα προβεί στις προβλεπόμενες διαδικασίες ανοίγματος καρτέλας ασθενούς ή ενημέρωσης της αν ήδη υπάρχει.

Στο πιο κάτω διάγραμμα (σχήμα 25) απεικονίζεται η αλληλεπίδραση που έχει η γραμματεία νοσοκομείου με άλλους χρήστες των οποίων τα χαρακτηριστικά γνωρίσματα θα αναλυθούν πιο κάτω.





ΣΧΗΜΑ 31 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΓΡΑΜΜΑΤΕΙΑΣ ΜΕ ΑΛΛΟΥΣ ΧΡΗΣΤΕΣ

### Η εφαρμογή από την πλευρά του χρήστη της γραμματείας ραντεβού των ακτινολογικών κέντρων

Ο δεύτερος τύπος χρήστη είναι η γραμματεία των ακτινολογικών τμημάτων του εκάστοτε νοσοκομείου. Ο ρόλος της γραμματείας αυτής είναι να παίρνει τις αιτήσεις για ακτινογραφία που βρίσκονται σε αναμονή και να κλείνει ραντεβού για ακτινογραφία σε συγκεκριμένη ημερομηνία. Έτσι έχει τις εξής δυνατότητες : Μπορεί να δει τις αιτήσεις που βρίσκονται σε αναμονή , έτσι ώστε να κλείσει ραντεβού. Ενώ επίσης μπορεί να δει τα ήδη προγραμματισμένα ραντεβού μέσω του ημερολογίου. (Εικόνα 4)

**SharpHealth**

Καλώς ήρθες, Γραμματιάδου  
Στέλλα  
Ειδικότητα: Γραμματεία  
Ακτινολογικού  
Τμήμα: Ακτινολογικό  
Νοσοκομείο: Ευαγγελισμός

[Αποσύνδεση](#)

Προγραμματισμός Ακτινογραφίας ▶  
Αρχική Σελίδα

12 : 00 AM  15 ▼

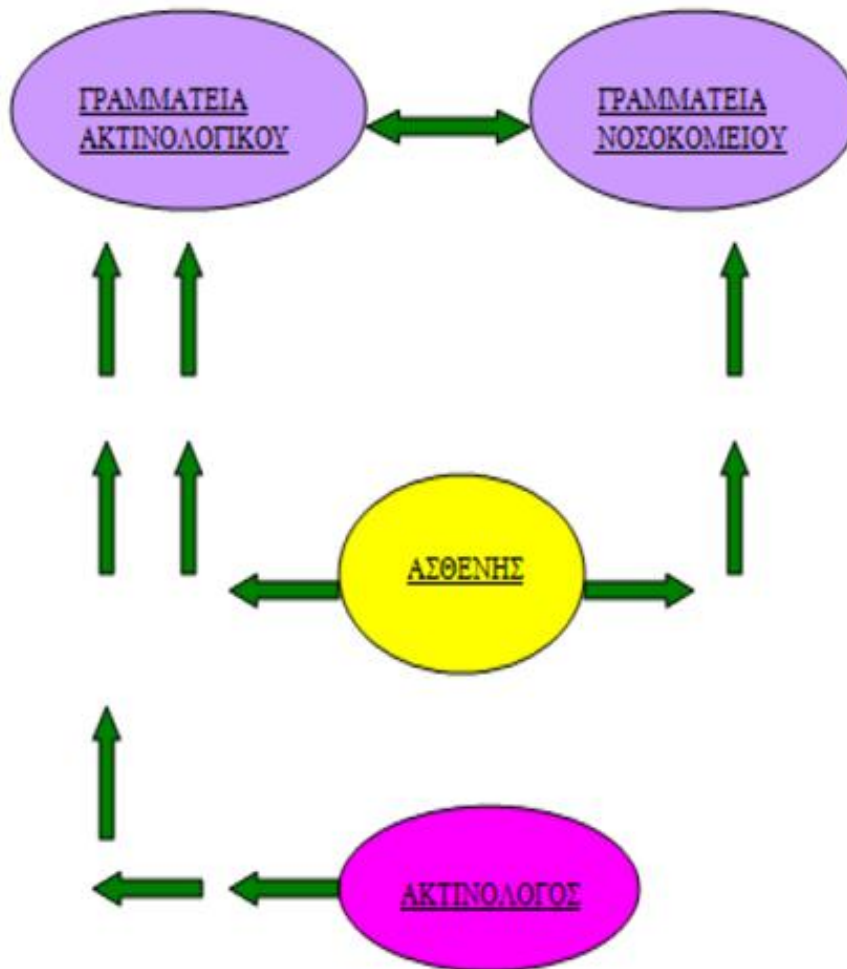
Ιούνιος 2011						
Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>
<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>
<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>
<u>27</u>	<u>28</u>	<b>29</b>	<u>30</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>

Μάιος	Ιούνιος 2011					Ιούλιος
Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>
<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>
<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>
<u>27</u>	<u>28</u>	<b>29</b>	<u>30</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>

ΕΙΚΟΝΑ 4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΡΑΝΤΕΒΟΥ

Σύμφωνα με τα όσα αναφέρθηκαν στο κεφάλαιο των παραδοχών λαμβάνουμε ως δεδομένο ότι κάθε ακτινολογικό τμήμα νοσοκομείου φέρει μια γραμματεία ακτινολογικού. Η γραμματεία λοιπόν ακτινολογικού ελέγχει στο σύστημα μέσω του αντίστοιχου πεδίου ότι όντως ο ασθενής διαθέτει καρτέλα ασθενούς στο νοσοκομείο και ότι χρήζει ακτινολογικής εξέτασης. Στη συνέχεια ελέγχοντας τη διαθεσιμότητα του ακτινολόγου, ορίζει ραντεβού ενημερώνοντας τον ασθενή για την ακριβή ημέρα και ώρα, αλλά και το σύστημα ανατρέχοντας στο αντίστοιχο πεδίο με σκοπό την έγκαιρη ενημέρωση και του γιατρού. Στην περίπτωση κατά την οποία ο ασθενής απευθύνεται απευθείας στην γραμματεία του ακτινολογικού, έχοντας ιατρική εντολή από εξωτερικό γιατρό για το είδος της εξέτασης που θα κάνει (ακτινογραφία), θα πρέπει η γραμματεία ακτινολογικού να αποστείλει τον ασθενή στη γραμματεία νοσοκομείου για την πραγματοποίηση της προβλεπόμενης διαδικασίας εισαγωγής ασθενούς ή αν

χρήζει άμεσης ανάγκης ο ασθενής να την πραγματοποιήσει η ίδια. Πιο κάτω παρατίθεται σχετικό διάγραμμα (σχήμα 26) αλληλεπίδρασης της γραμματείας ακτινολόγου με τους άλλους χρήστες.



ΣΧΗΜΑ 32 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΤΗΣ ΓΡΑΜΜΑΤΕΙΑΣ ΑΚΤΙΝΟΛΟΓΟΥ

### Η εφαρμογή από την πλευρά του χρήστη ακτινολόγου ιατρού

Ο δεύτερος τύπος χρήστη είναι ο κάθε ακτινολόγος ιατρός του εκάστοτε νοσοκομείου. Ο ακτινολόγος ιατρός αφού εισαχθεί στην εφαρμογή, βλέπει τα ραντεβού που έχουν κανονιστεί για ακτινογραφία και βρίσκονται σε αναμονή.

## SharpHealth

Καλώς ήρθες,  
Ακτινολογόπουλος  
ατρός  
Ειδικότητα:  
Ακτινολόγος  
Γμήμα: Ακτινολογικό  
Νοσοκομείο:  
Ευαγγελισμός

Επιλέξτε μέρα

Ιούνιος 2011									
Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ			
30	31	1	2	3	4	5			
6	7	8	9	10	11	12			
13	14	15	16	17	18	19			
20	21	22	23	24	25	26			
27	28	29	30	1	2	3			
4	5	6	7	8	9	10			

[Αποσύνδεση](#)

Αρχική Σελίδα

	Ασθενής	Σημειώσεις	Μηχάνημα	Πληροφορίες
<a href="#">Εκτέλεση</a>	Gkanatsios Dimitris-Ilias		Μηχάνημα ενδαγγειακής υπερηχοτομογραφίας	<a href="#">Πληροφορίες Ασθενή</a>

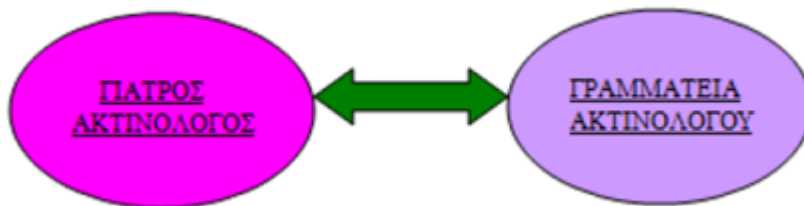
ΕΙΚΟΝΑ 5 ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ ΑΚΤΙΝΟΛΟΓΟΥ

Μπορεί να επιλέξει να δει πληροφορίες σχετικά με τον ασθενή και το ιστορικό του , καθώς και να εκτελέσει την ακτινογραφία του ασθενή ανεβάζοντας την εικόνα της ακτινογραφίας και γράφοντας κάποια σχόλια , το είδος της ασθένειας.

Μόλις ελεγχθεί η διαθεσιμότητα του ακτινολόγου γιατρού από τη γραμματεία ακτινολογικού, ορίζεται ραντεβού και συμπληρώνεται στο αντίστοιχο πεδίο για να είναι ενημερωμένος τόσο ο γιατρός όσο και το νοσοκομείο. Ο ακτινολόγος καθημερινά ενημερώνεται ανατρέχοντας στο αντίστοιχο πεδίο της εφαρμογής για τις εξετάσεις που έχει να πραγματοποιήσει και μετά το πέρας της καθημιάς συμπληρώνει με ιατρικές παρατηρήσεις τις καρτέλες των ασθενών απευθυνόμενος στο ιατρικό ιστορικό τους. Έχει σημειωθεί ως παραδοχή ότι όλοι οι ακτινολόγοι του συγκεκριμένου περιφερειακού νοσοκομείου, αλλά και ιατρικού κέντρου που συνεργάζεται με αυτό, έχουν τη δυνατότητα της πρόσβασης στην καρτέλα οποιουδήποτε ασθενούς ακτινολογικού που επισκέπτεται τα εν λόγω συνεργαζόμενα ιατρικά κέντρα. Με γνώμονα αυτό γίνεται φανερό ότι στην περίπτωση βάρδιας των γιατρών ακτινολόγων και έκτακτης επίσκεψης ασθενών οι γιατροί ακτινολόγοι είναι σε θέση να γνωρίζουν κάθε φορά το ιατρικό ιστορικό κάθε ασθενούς και τις προηγούμενες γνωματεύσεις συναδέλφων τους για αυτούς.

ΕΙΚΟΝΑ 6 ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΚΤΙΝΟΓΡΑΦΙΑΣ

Στο παρακάτω διάγραμμα (σχήμα 33) παρουσιάζεται η αλληλεπίδραση των γιατρών ακτινολόγων με τους υπόλοιπους χρήστες του συστήματος.



ΣΧΗΜΑ 33 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΑΚΤΙΝΟΛΟΓΟΥ

### Η εφαρμογή από την πλευρά του χρήστη λογιστή του νοσοκομείου

Ένας ακόμα τύπος χρήστη είναι ο λογιστής του εκάστοτε νοσοκομείου. Ο λογιστής είναι υπεύθυνος για την χρέωση των ασθενών για την ακτινογραφία ή την εξέταση στην οποία θα υποβληθούν. Με την

είσοδο του στην εφαρμογή βλέπει τις εξετάσεις για ακτινογραφία που βρίσκονται σε αναμονή και δεν έχουν πληρωθεί. Αφού οι ασθενείς πληρώσουν γίνεται επιβεβαίωση από τον λογιστή, στην εφαρμογή ότι έχουν πληρωθεί.

Στην παρακάτω εικόνα παρουσιάζεται η καρτέλα στην οποία απευθύνεται το λογιστήριο προκειμένου να ελέγξει το ταμείο με το οποίο είναι συμβεβλημένος ο ασθενής ή να σημειώσει το ταμείο του στην περίπτωση κατά την οποία ο ασθενής είναι καινούργιος.

**SharpHealth**

Καλώς ήρθες, Logistis Λογιστής  
 Ειδικότητα: Λογιστής  
 Τμήμα: Ακτινολογικό  
 Νοσοκομείο: Ευαγγελισμός

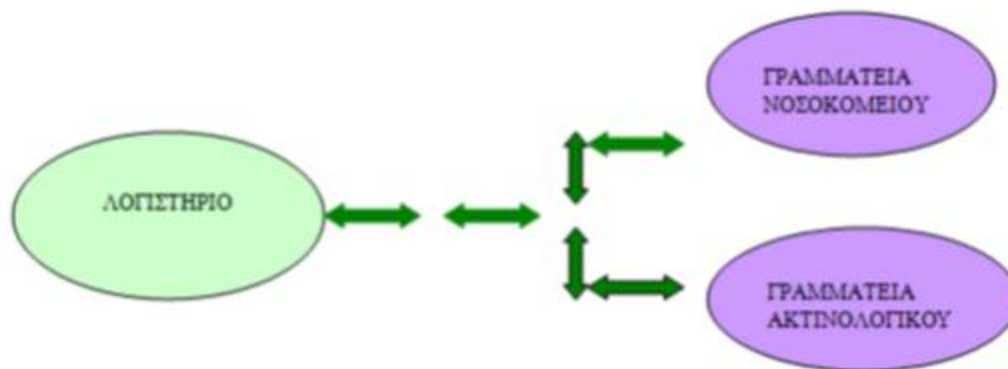
[Αποσύνδεση](#)

Accounting ▶

Ημέρα-Ωρα	Όνομα ασθενή	Μηχάνημα	
19/6/2011 12:00:00 πμ	Gkanatsios Dimitris-Ilias	Μαστογράφος	<a href="#">Χρέωση</a>
1/7/2011 12:00:00 πμ	Gkanatsios Dimitris-Ilias	Μηχάνημα ενδαγγειακής υπερηχοτομογραφίας	<a href="#">Χρέωση</a>

ΕΙΚΟΝΑ 7 ΟΘΟΝΗ ΛΟΓΙΣΤΗ

Ο ρόλος του λογιστηρίου είναι να ελέγχει τις καρτέλες των ασθενών που είναι προς εξέταση, προκειμένου να διαπιστώνουν με ποιο ταμείο είναι συμβεβλημένοι και να αποφαινόνται για τον τρόπο πληρωμής των εξετάσεων που πραγματοποιούν. Επιπλέον το λογιστήριο θα ενημερώνει την καρτέλα του ασθενούς ότι ο ασθενής πλήρωσε το αντίστοιχο ποσό και θα απευθύνεται στην γραμματεία νοσοκομείου ή γραμματεία ακτινολογικού για οποιαδήποτε διευκρινιστική επισήμανση περί ασθενούς. Το λογιστήριο θα βρίσκεται σε άμεση επικοινωνία με τις δυο γραμματείες κάτι το οποίο και αναδεικνύεται και σχηματικά πιο κάτω (σχήμα 34).



ΣΧΗΜΑ 34 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΛΟΓΙΣΤΗ

## Η εφαρμογή από την πλευρά του θεράποντα ιατρού του νοσοκομείου

Εν τέλει ο τελευταίος τύπος χρήστη είναι ο θεράπων ιατρός του εκάστοτε ασθενή που τον έχει αναλάβει, ο οποίος κάνει την διάγνωση από την ακτινογραφία που έχει ήδη κάνει ο ασθενής, αφού έχει ήδη πληρώσει για αυτήν. Ο θεράπων ιατρός αφού λάβει την ακτινογραφία που έβγαλε ο ασθενής την εξετάζει και ενημερώνει την καρτέλα του ασθενή για την ασθένεια του και σχόλια σχετικά με τον τρόπο αντιμετώπισης της.

### SharpHealth

Καλώς ήρθες, Ιατρόπουλος  
Ιατρός  
Ειδικότητα: Καρδιολόγος  
Τμήμα: Ορθοπαιδικό  
Νοσοκομείο: Ευαγγελισμός

Στοιχεία  
Ασθενής: Gkanatsios Dimitris-Ilias  
Σημειώσεις: test  
Ημέρα εξέτασης: 12:00:00 πμ  
Σημειώσεις ακτινολόγου

[Αποσύνδεση](#)

Αναζήτηση Ασθενή  
Αρχική Σελίδα

[Ακτινογραφία](#)

Διαγνωθείσες ασθένειες από ακτινολόγο

I. Infectious & Parasitic Diseases Venereal disease, other spec.

Γράψτε τα πρώτα γράμματα της ασθένειας με λατινικούς χαρακτήρες:  [Αναζήτηση](#)

Επιλέξτε ασθένεια και πατήστε στο κουμπί "Προσθήκη"

Mononucleosis |43 ▾ [Προσθήκη](#)

Σχόλια - Διάγνωση

serious infection|

[Καταχώρηση](#)

ΕΙΚΟΝΑ 8 ΟΘΟΝΗ ΘΕΡΑΠΟΝΤΑ ΙΑΤΡΟΥ

## Η εφαρμογή από την πλευρά του διαχειριστή

Εν τέλει ο τελευταίος τύπος χρήστη είναι ο διαχειριστής του συστήματος (Administrator) ο οποίος έχει και τα περισσότερα δικαιώματα πάνω στην εφαρμογή. Ο διαχειριστής μπορεί να εισάγει νέους χρήστες στο σύστημα ή και να διαγράψει κάποιους από αυτούς. Μπορεί επίσης να επεξεργαστεί στοιχεία που αφορούν τα νοσοκομεία και τα τμήματα των νοσοκομείων. Ενώ τέλος μπορεί να εισάγει και μηχανήματα του νοσοκομείου που συσχετίζονται με την εφαρμογή.

## SharpHealth

Διαχειριστής

Καλώς ήρθες EUROPElv-digkan

- Διαχείριση
  - ο Εισαγωγή Νοσοκομείου
  - ο Εισαγωγή Τμήματος
  - ο Εισαγωγή Μηχανημάτων
- Διαχείριση χρηστών
  - ο Χρήστες/Groups
- Λειτουργίες
  - ο Ενεργό/Μη ενεργό

FirstName	LastName	
Λογιστής	Logistis	<a href="#">Edit</a>
Manager	Manageridis	<a href="#">Edit</a>
Ιατρός	Ιατρόπουλος	<a href="#">Edit</a>
Admin	Admin	<a href="#">Edit</a>
Ιατρός	Ακτινολογόπουλος	<a href="#">Edit</a>
Κατερίνα	Σεκρεταριάδου	<a href="#">Edit</a>
Στέλλα	Γραμματιάδου	<a href="#">Edit</a>

[Δημιουργία](#)

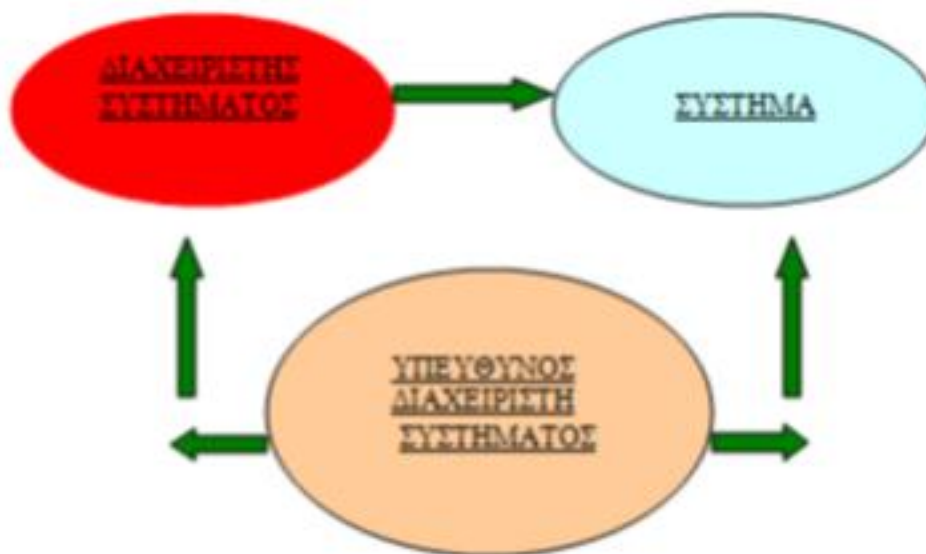
ΕΙΚΟΝΑ 9 ΟΘΟΝΗ ΔΙΑΧΕΙΡΙΣΤΗ

Ο διαχειριστής συστήματος έχει την ιδιότητα της πάσης φύσεως επέμβασης στο σύστημα με στόχο και σκοπό τη βελτιστοποίηση της λειτουργίας του, την διόρθωση πιθανών σφαλμάτων που μπορούν να προκύψουν και την εκπαίδευση του προσωπικού για τον χειρισμό των επιμέρους λειτουργιών του. Σε κάθε περίπτωση ο διαχειριστής του συστήματος (administrator) δεν έχει τη δυνατότητα της πρόσβασης σε καρτέλες ασθενών αλλά και σε οποιουδήποτε είδους στοιχεία ασθενών που αφορούν ευαίσθητα



προσωπικά δεδομένα τους. Το σύστημα είναι σε θέση να καταγράφει την ακριβή ημέρα και ώρα κατά την οποία ο διαχειριστής του συστήματος προέβη σε οποιαδήποτε ενέργεια, σε σχέση με αυτό, καθώς επίσης υποχρέωση του διαχειριστή συστήματος είναι να περιγράφει κάθε φορά το λόγο επέμβασης του στο σύστημα. Το πεδίο στο οποίο θα αναφέρονται οι κινήσεις του διαχειριστή συστήματος θα είναι προσβάσιμο μόνο από τις ελεγκτικές υπηρεσίες (auditors).

Στο σημείο αυτό θα πρέπει να αναφερθεί ότι οι αρμοδιότητες καθώς επίσης και ο τρόπος ελέγχου του διαχειριστή συστήματος στο σύστημα αναφέρονται και περιγράφονται λεπτομερώς στο ειδικό εγχειρίδιο διαχειριστή συστήματος που συνοδεύει την συγκεκριμένη εφαρμογή. Στο πιο κάτω διάγραμμα (σχήμα 35) παρουσιάζεται η αλληλεπίδραση του διαχειριστή συστήματος με τους άμεσα συνδεδεμένους με αυτόν χρήστες του συστήματος.



ΣΧΗΜΑ 35 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΔΙΑΧΕΙΡΙΣΤΗ

### Ελεγκτικό όργανο

Ο ελεγκτής (auditor) συστήματος έχει την δυνατότητα του ελέγχου οποιασδήποτε ενέργειας του διαχειριστή συστήματος και την προστασία απόρρητων πληροφοριών ασθενών από ενδεχόμενη διαρροή τους προς μη αρμοδίους. Ο λόγος ύπαρξης του υπεύθυνου διαχειριστή συστήματος, ο οποίος μπορεί και να είναι ο διευθυντής νοσοκομείου, έγκειται στην ανάγκη της καταγραφής προβλημάτων ή δυσλειτουργιών του συστήματος και ενημέρωση του διαχειριστή συστήματος για αυτό, αλλά και την πρόταση ενσωμάτωσης επιπλέον λειτουργιών στο σύστημα με σκοπό την ακόμα καλύτερη προσαρμογή του στις σύγχρονες ιατρικές ανάγκες. Επιπλέον ο υπεύθυνος διαχειριστής συστήματος έχει τη δυνατότητα της πρόσβασης σε όλα τα πεδία της εφαρμογής και βρίσκεται σε άμεση επικοινωνία με το διαχειριστή συστήματος. Πιο κάτω παρουσιάζεται το διάγραμμα (σχήμα 30)

αλληλεπίδρασης του υπεύθυνου διαχειριστή συστήματος με τους υπόλοιπους διαχειριστές του συστήματος.



ΣΧΗΜΑ 36 ΑΛΛΗΛΕΠΙΔΡΑΣΗ AUDITOR

## Καταγραφή συμβάντων

### *Event log των Windows*

Κατά την διάρκεια εκτέλεσης της εφαρμογής το Event Log των Windows ενημερώνεται άμεσα για τυχόν σφάλματα που μπορεί να προκύψουν είτε για άλλες πληροφορίες.

### *Logging των κλήσεων και των απαντήσεων των web services*

Επιπλέον γίνεται καταγραφή των SOAP μηνυμάτων που δέχονται αλλά και απαντάνε τα Web Services. Επίσης στην συγκεκριμένη περιοχή γίνεται και καταγραφή για διάφορα σφάλματα που μπορεί να προκύψουν από τα Web Services.

### *Custom logging στο data layer της εφαρμογής*

Επίσης στο Data Layer της εφαρμογής γίνεται καταγραφή της κάθε κλήσης που γίνεται προς τις βάσεις δεδομένων της εφαρμογής.

### *Logging των workflows με χρήση του SQL Workflow tracking service*

Μια ακόμα καταγραφή που γίνεται, είναι η καταγραφή της ροής εργασιών (Workflows) από την δημιουργία τους, μέχρι και την ολοκλήρωσή τους στην βάση δεδομένων.

### **Logging των κλήσεων της ιστοσελίδας**

Τέλος γίνεται χρήση της υπηρεσίας του IIS , για την καταγραφή των αιτημάτων που γίνονται για σελίδες του Web Site κατά την διάρκεια λειτουργίας της εφαρμογής.

### **Στατιστικά που είναι δυνατόν να εκδοθούν μέσω της εφαρμογής**

- Στατιστικά στοιχεία εκθέσεων πεπραγμένων
- Σύνολο εντολών
- Σύνολο εσωτερικών εντολών
- Σύνολο εντολών εξωτερικών ιατρείων
- Σύνολο εντολών εισαχθέντων ασθενών
- Σύνολο εξωτερικών εντολών (ασθενείς που διαβιβάστηκαν σε άλλο Νοσοκομείο ή Κέντρο Υγείας)
- Ηλικιακή και ανά Φύλο κατανομή ασθενών
- Ποσοστό ασφαλισμένων ανά Ταμείο
- Σύνολο ασθενών ανά πάθηση
- Σύνολο ασθενών ανά είδος εξέτασης
- Σύνολο ασθενών ανά πάθηση που διαβιβάστηκαν σε άλλο Νοσοκομείο ή Κέντρο Υγείας
- Σύνολο επειγόντων περιστατικών
- Σύνολο ασθενών ανά ιατρό
- Σύνολο ασθενών ανά νοσηλευτικό προσωπικό
- Εκτελεσθείσες εντολές (Σύνολα)
- Εκκρεμείς εντολές (Σύνολα)

### **Διαχείριση λογαριασμών χρηστών και αντικειμένων**

Για να μπορέσει να εισαχθεί ένας χρήστης στο σύστημα πρέπει να έχει έναν προσωπικό λογαριασμό με τον οποίο εισάγοντας το όνομα του και τον προσωπικό του κωδικό το και μόνο θα μπορεί να εισαχθεί στο σύστημα. Για την δημιουργία , τροποποίηση ή διαγραφή λογαριασμών για κάθε χρήστη, είναι υπεύθυνος ο διαχειριστής (Administrator) του συστήματος. Σε αυτόν πρέπει να απευθύνονται αυτοί που θέλουν να δημιουργήσουν νέο λογαριασμό ή να τροποποιήσουν στοιχεία ενός υπάρχοντος λογαριασμού.

### **Εισαγωγή νέων χρηστών**

Ο Administrator για να μπορέσει να εισάγει έναν καινούργιο χρήστη στο σύστημα και να δώσει ρόλους σε αυτόν χρειάζεται να δημιουργήσει έναν νέο λογαριασμό για αυτόν. Επιλέγει από το μενού αριστερά την ετικέτα "χρήστες/Groups" και αμέσως του εμφανίζεται μία λίστα με τους υπάρχοντες χρήστες της εφαρμογής. Εκεί έχει την δυνατότητα να επεξεργαστεί τον λογαριασμό ενός ήδη υπάρχον χρήστη του συστήματος ή να δημιουργήσει έναν καινούργιο.

### Ορισμός επιπέδου πρόσβασης στους χρήστες

Ο Administrator μπορεί να ορίσει το επίπεδο πρόσβασης σε έναν ήδη υπάρχον χρήστη του συστήματος ή σε έναν καινούργιο χρήστη. Αυτό μπορεί να το πραγματοποιήσει πατώντας edit δίπλα από το όνομα του υπάρχον χρήστη ή να επιλέξει δημιουργία. Στην συνέχεια του εμφανίζεται η σχετική καρτέλα μέσα από την οποία μπορεί να δώσει ή να επεξεργαστεί τα στοιχεία του και να καθορίσει τον ρόλο του .Ο ρόλος του κάθε χρήστη καθορίζει το επίπεδο πρόσβασης του.

**SharpHealth**

Διαχειριστής

Τμήμα  
Αλεξάνδρα - Ακτινολογικό

Καλώς ήρθες EUROPE\w-digkan Όνομα

Επώνυμο

Κατηγορία Doctor

Συγκεκριμένη κατηγορία

Ρόλος  
Administrator

Όνομα χρήστη

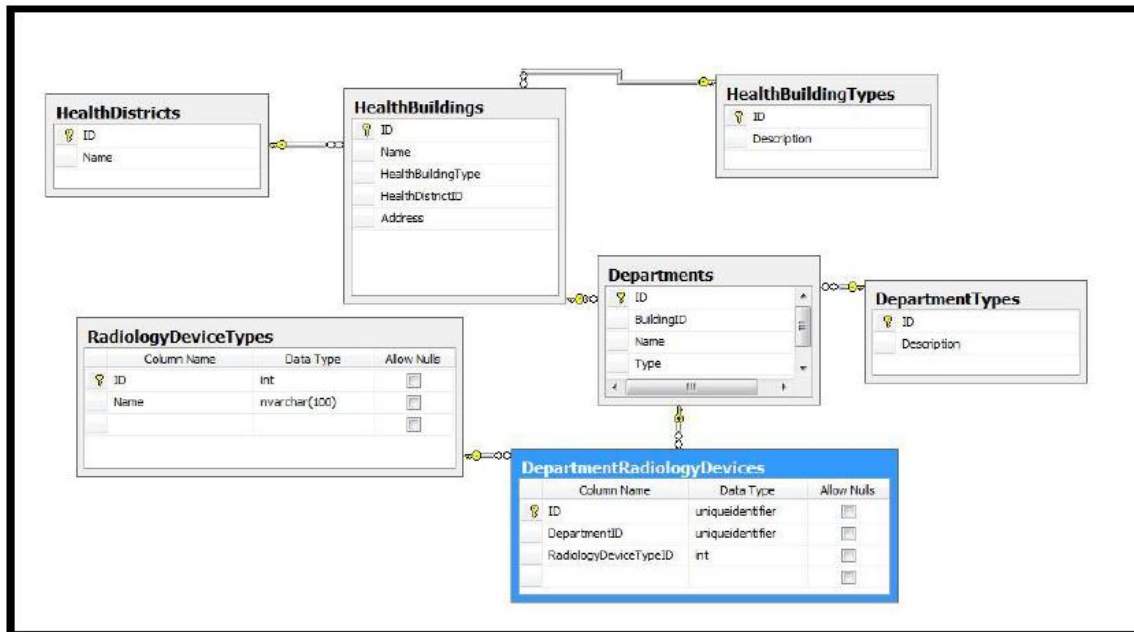
Κωδικός

Αποθήκευση

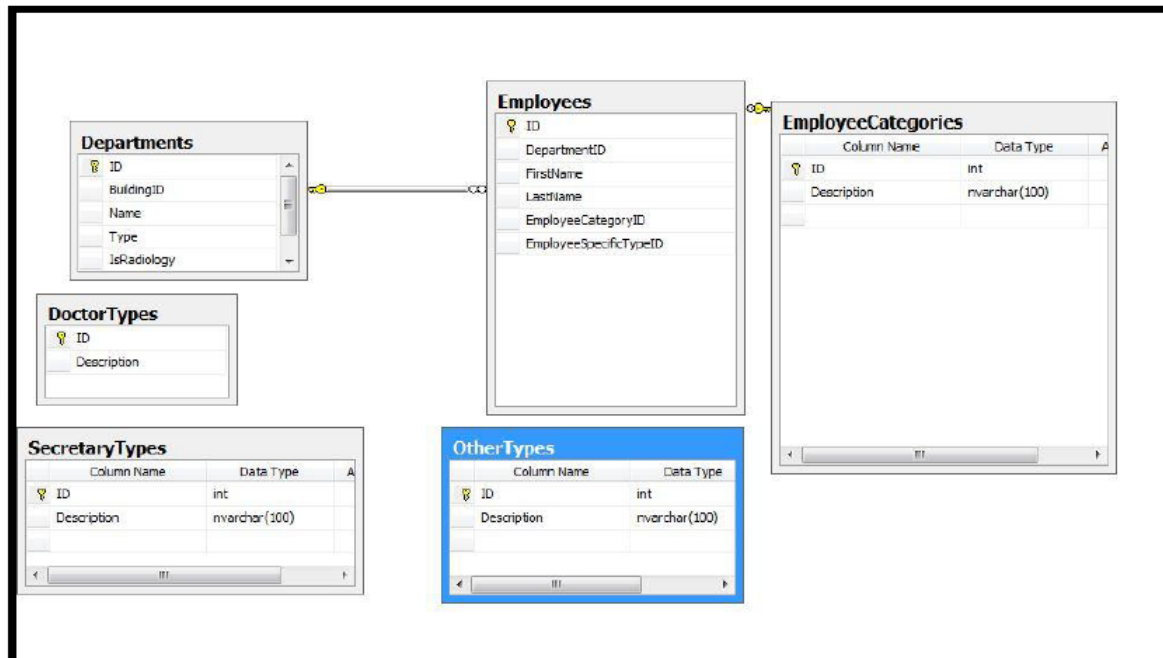
- Διαχείριση
  - Εισαγωγή Νοσοκομείου
  - Εισαγωγή Τμήματος
  - Εισαγωγή Μηχανημάτων
- Διαχείριση χρηστών
  - Χρήστες/Groups
- Λειτουργίες
  - Ενεργό/Μη ενεργό

ΣΧΗΜΑ 37 ΔΗΜΙΟΥΡΓΙΑ ΧΡΗΣΤΩΝ

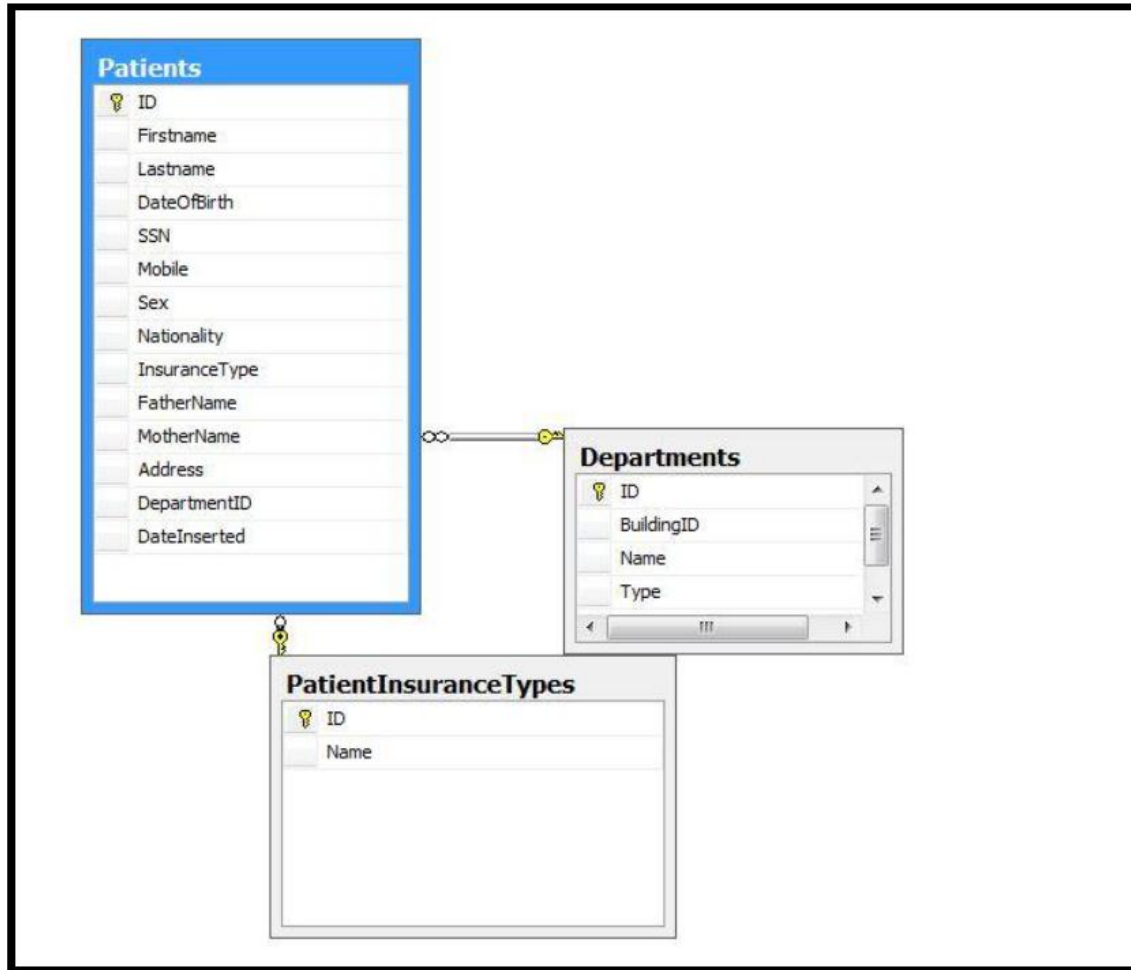
## Δημιουργία βάσης δεδομένων



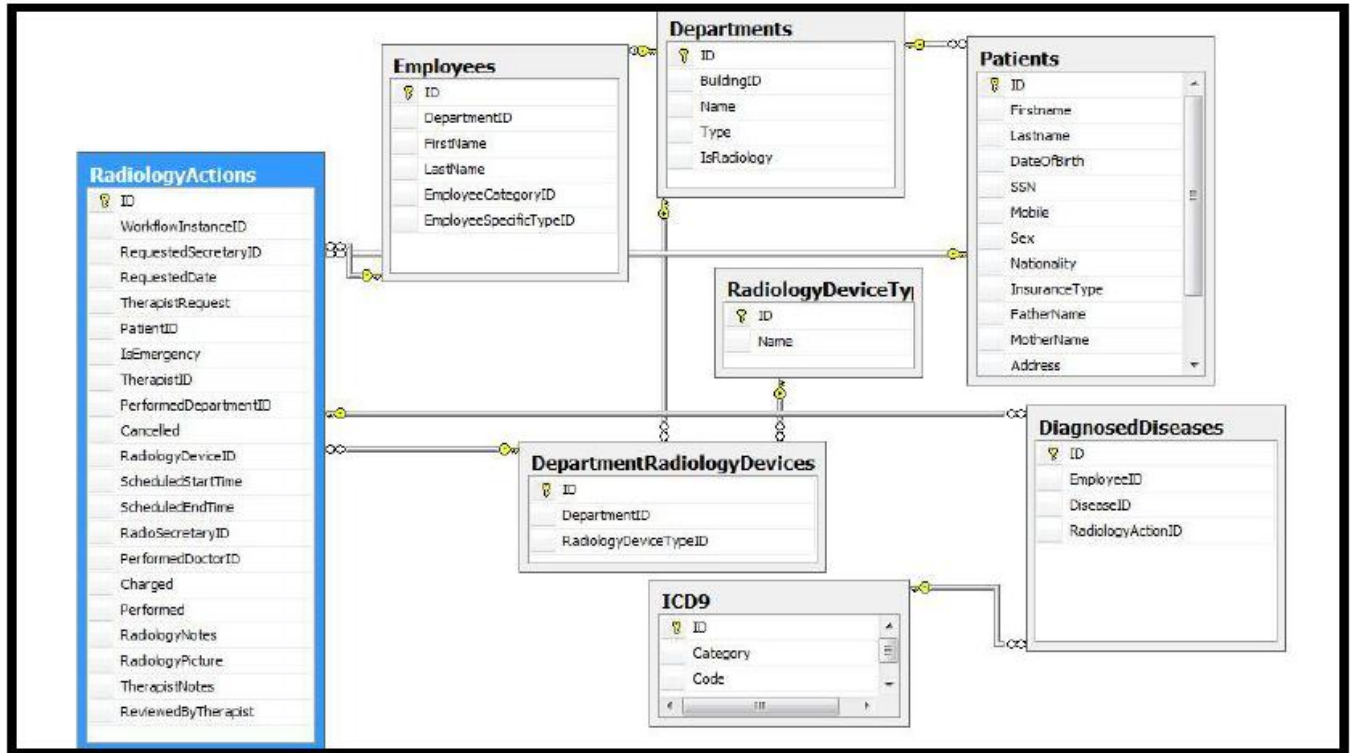
ΣΧΗΜΑ 38 ΚΑΤΑΝΟΜΗ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΣΥΣΚΕΥΩΝ ΑΝΑ ΥΓΕΙΟΝΟΜΙΚΗ ΜΟΝΑΔΑ



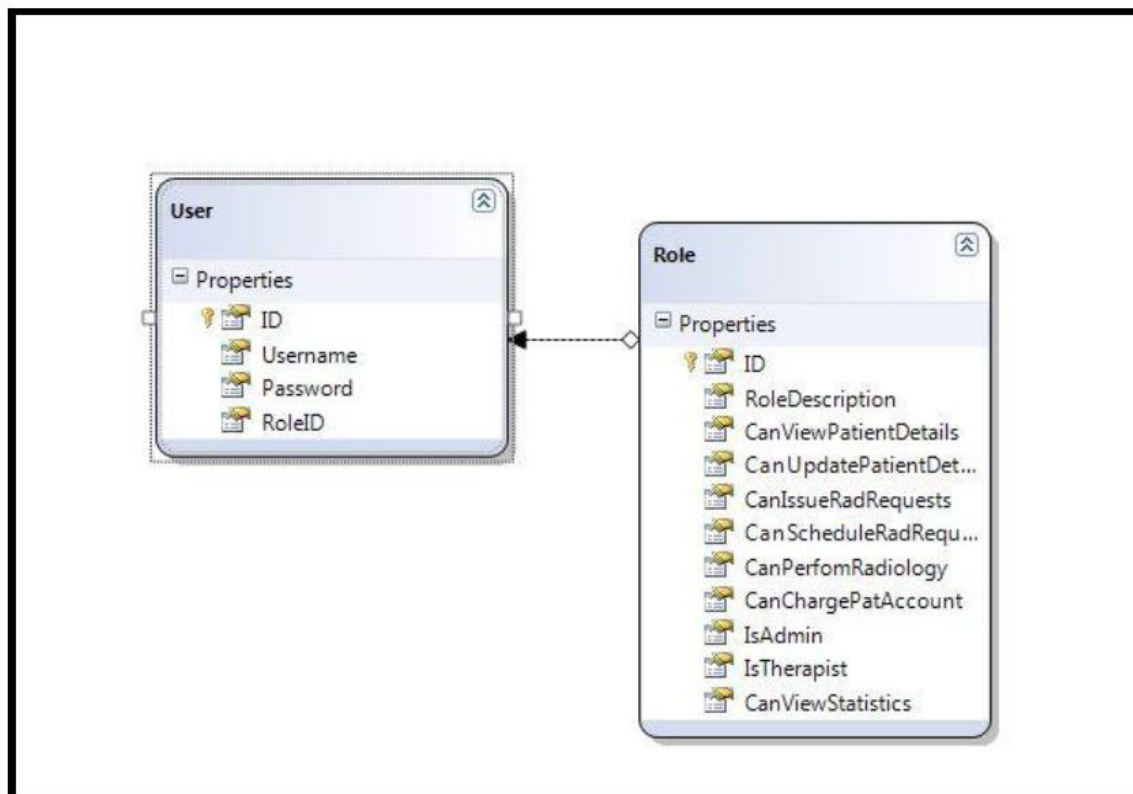
ΣΧΗΜΑ 39 ΚΑΤΑΝΟΜΗ ΥΠΑΛΛΗΛΩΝ ΑΝΑ ΤΜΗΜΑ



ΣΧΗΜΑ 40 ΚΑΤΑΝΟΜΗ ΑΣΘΕΝΩΝ ΑΝΑ ΤΜΗΜΑ

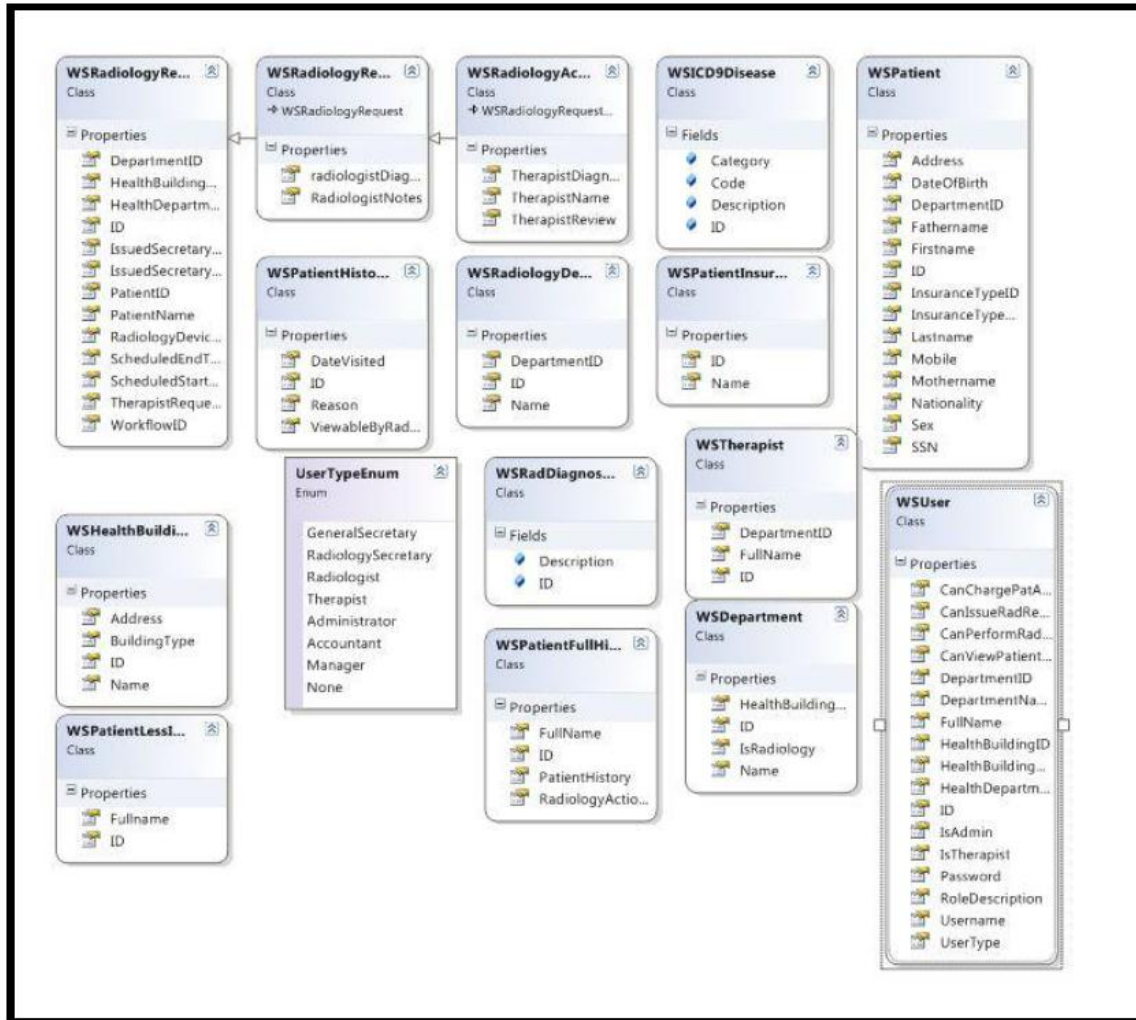


ΣΧΗΜΑ 41 ΠΙΝΑΚΕΣ ΓΙΑ ΑΠΟΘΗΚΕΥΣΗ ΑΚΤΙΝΟΛΟΓΙΚΩΝ ΕΞΕΤΑΣΕΩΝ

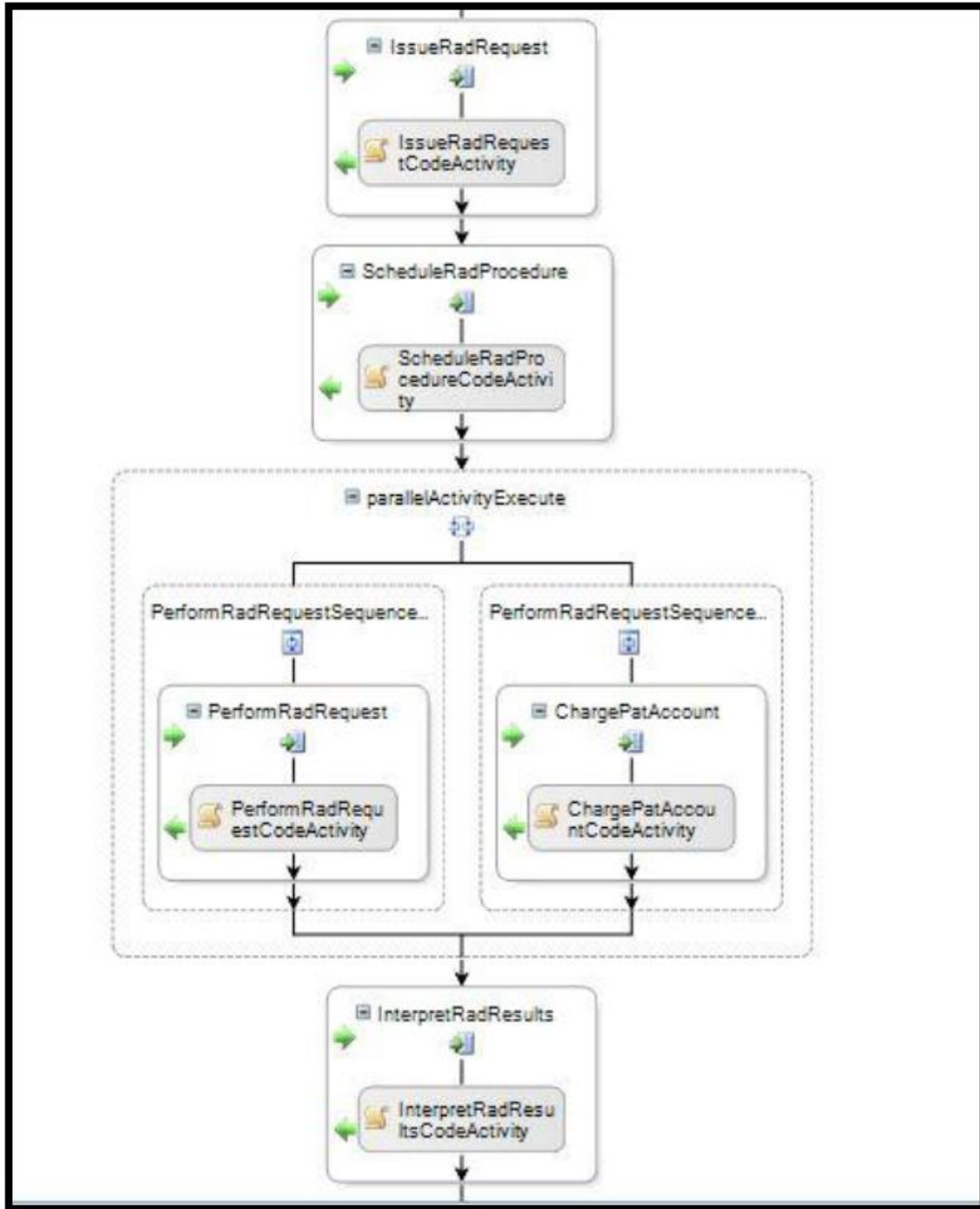


ΣΧΗΜΑ 42 ΑΝΤΙΚΕΙΜΕΝΑ ΓΙΑ ΡΟΛΟΥΣ ΧΡΗΣΤΩΝ



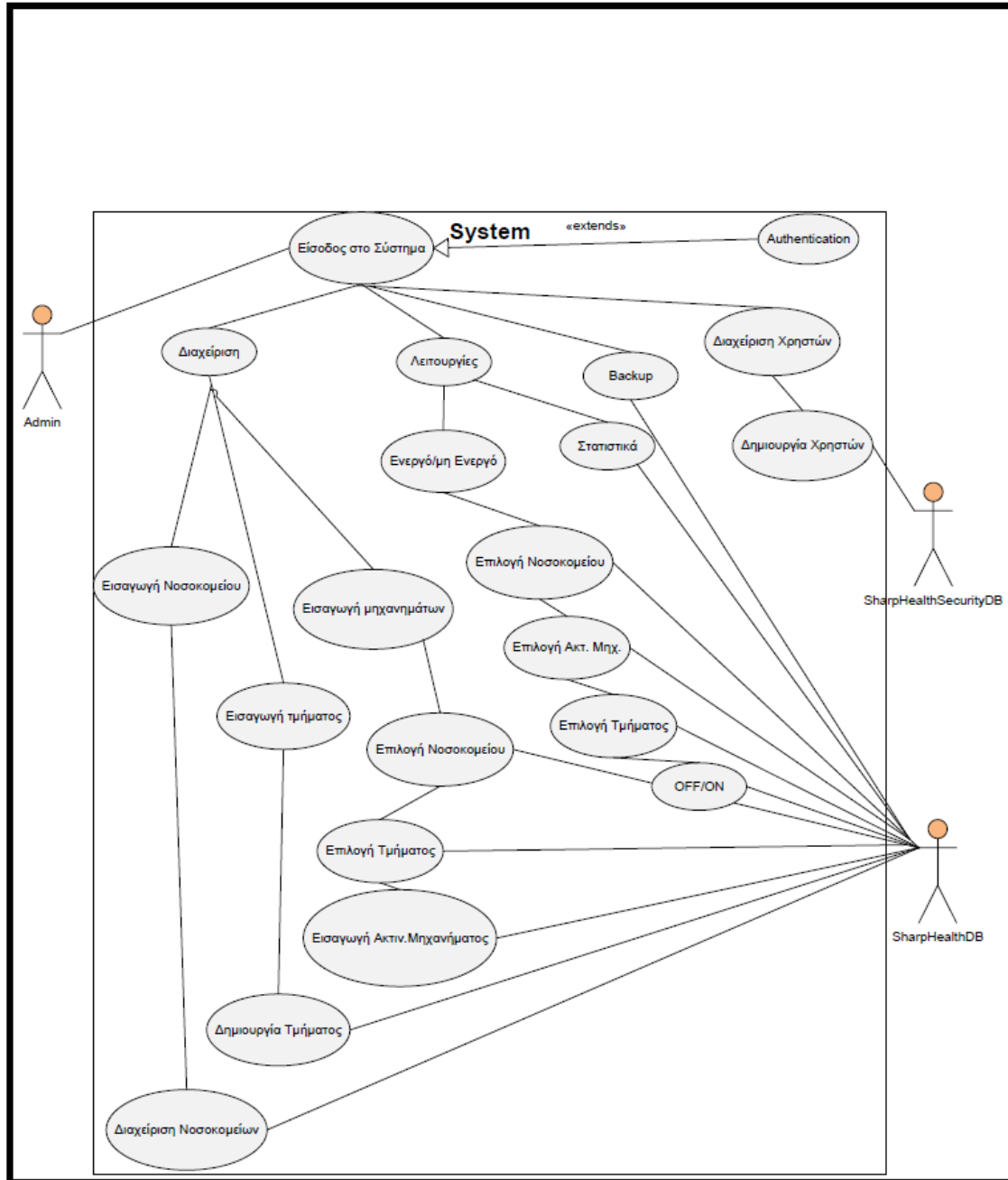


ΣΧΗΜΑ 43 ΑΝΤΙΚΕΙΜΕΝΑ ΠΟΥ ΕΠΙΣΤΡΕΦΟΝΤΑΙ ΑΠΟ ΤΑ WEB SERVICES

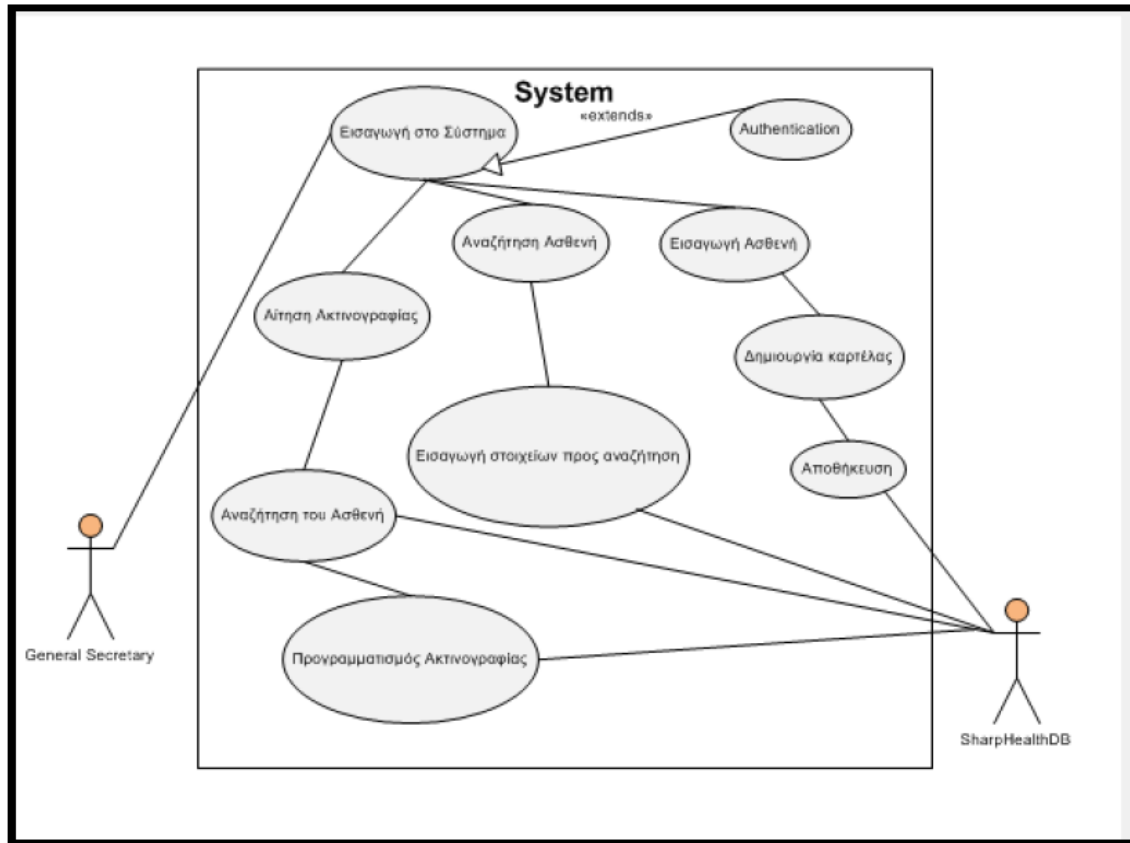


ΣΧΗΜΑ 44 APPLICATION WORKFLOW

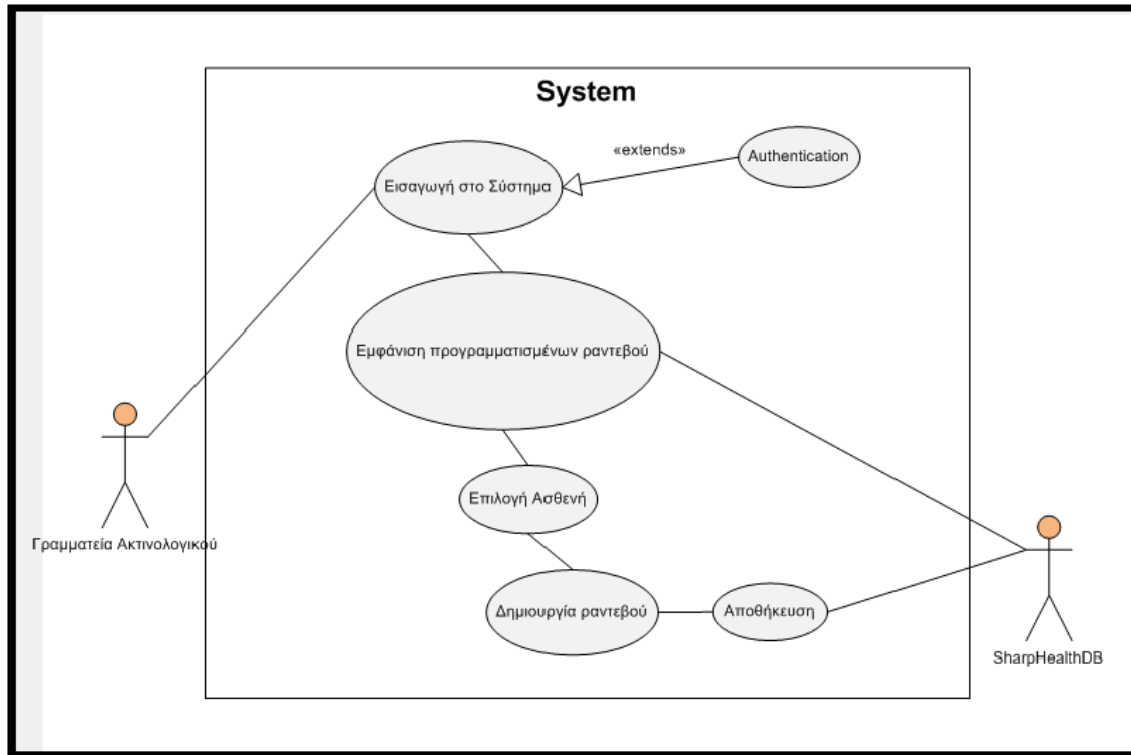
## use case διαγράμματα για την εφαρμογή



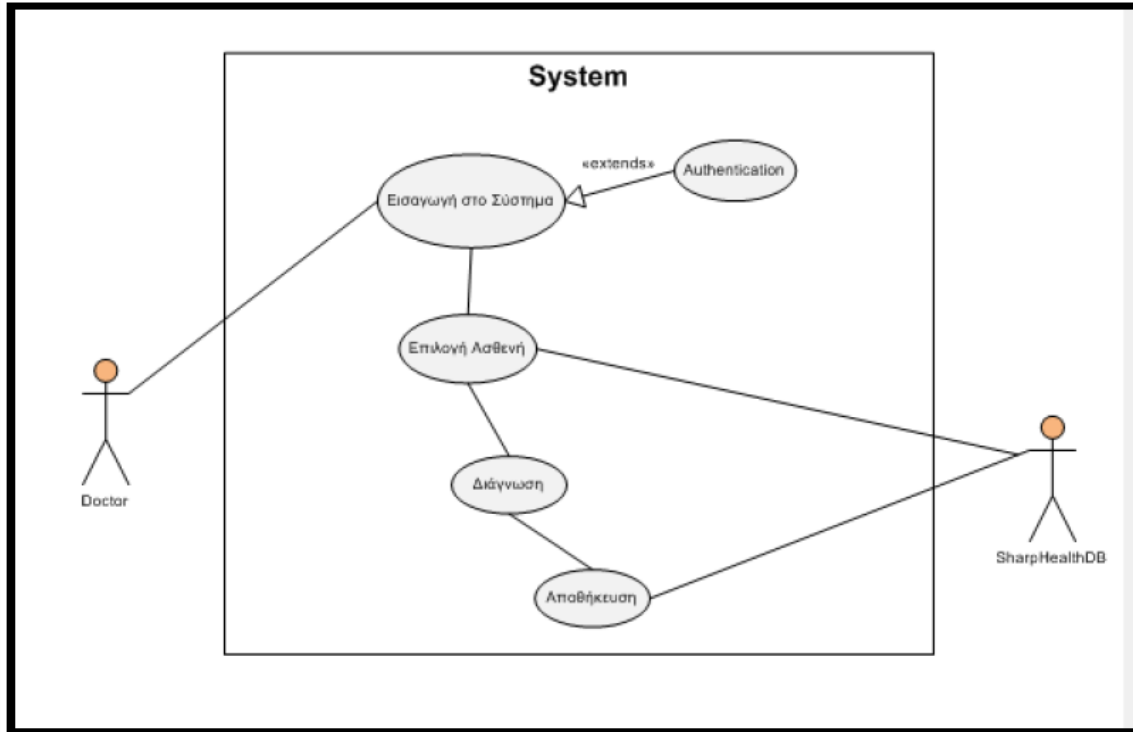
ΕΙΚΟΝΑ 10 ADMINISTRATOR USE CASE



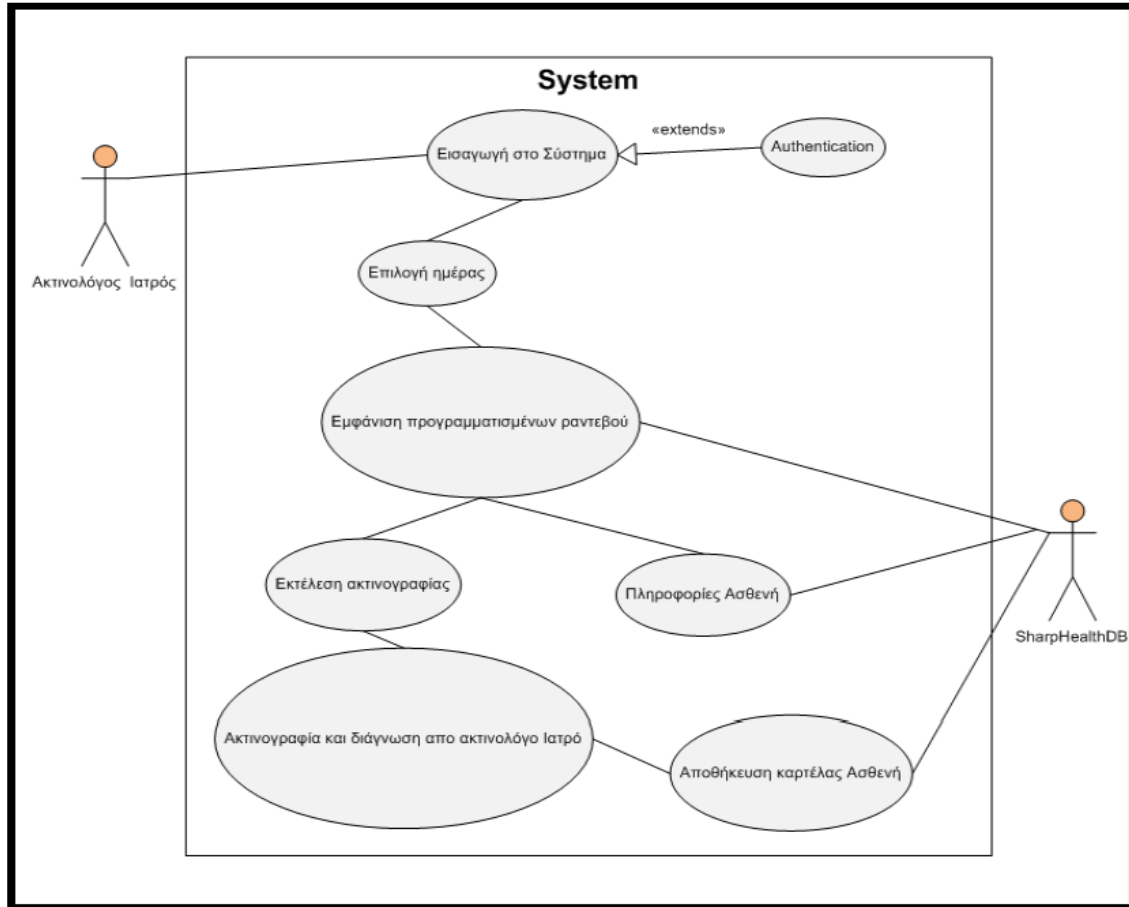
ΕΙΚΟΝΑ 11 USE CASE ΓΡΑΜΜΑΤΕΙΑΣ ΜΗ ΑΚΤΙΝΟΛΟΓΙΚΟΥ ΤΜΗΜΑΤΟΣ ΝΟΣΟΚΟΜΕΙΟΥ



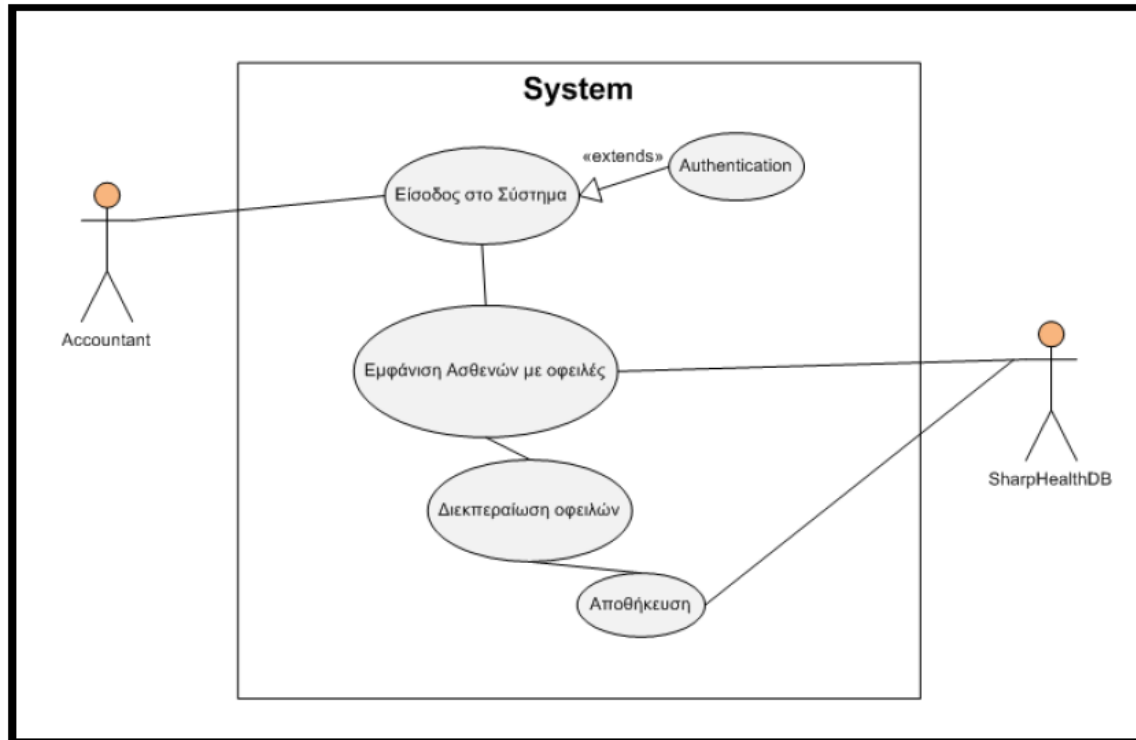
ΕΙΚΟΝΑ 12 USE CASE ΓΡΑΜΜΑΤΕΙΑΣ ΑΚΤΙΝΟΛΟΓΙΚΟΥ ΤΜΗΜΑΤΟΣ



ΕΙΚΟΝΑ 13 USE CASE ΔΙΑΓΡΑΜΜΑ ΘΕΡΑΠΟΝΤΟΣ ΙΑΤΡΟΥ



ΕΙΚΟΝΑ 14 USE CASE ΔΙΑΓΡΑΜΜΑ ΙΑΤΡΟΥ ΑΚΤΙΝΟΛΟΓΙΚΟΥ ΤΜΗΜΑΤΟΣ



ΕΙΚΟΝΑ 15 USE CASE ΔΙΑΓΡΑΜΜΑ ΛΟΓΙΣΤΗΡΙΟΥ

## Χρήση BPEL από την εφαρμογή

Το Workflow Foundation το οποίο και χρησιμοποιήσαμε για την κατασκευή της εφαρμογής έχει τη δυνατότητα να χρησιμοποιήσει εξωτερική βιβλιοθήκη (BPEL for Windows Workflow Foundation March CTP) (16) ώστε να μπορέσει να εξάγει τις ροές εργασίας που δημιουργούνται σε BPEL. Έτσι, με τη χρήση του εργαλείου «BPEL Export Tool» και με τη χρήση των βιβλιοθηκών δημιουργήθηκε το κάτωθι αρχείο BPEL που προσδιορίζει τη ροή εργασίας για την εφαρμογή που κατασκευάσαμε.

```

<process name="RadIssueWorkflow" targetNamespace="http://tempuri.org/"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:tns="http://tempuri.org/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/">
  <partnerLinks>
    <partnerLink name="file" partnerLinkType="tns:FileLinkType" myRole="application" />
    <partnerLink name="external" partnerLinkType="tns:RadIssueWorkflowLinkType"
myRole="application" />
  </partnerLinks>
  <variables>
    <variable name="RadIssueWorkflowMessage" messageType="tns:SendRadIssueWorkflowSoapIn" />
    <variable name="writeLineIn" messageType="tns:writeLineSoapIn" />
  </variables>
</process>
    
```



```

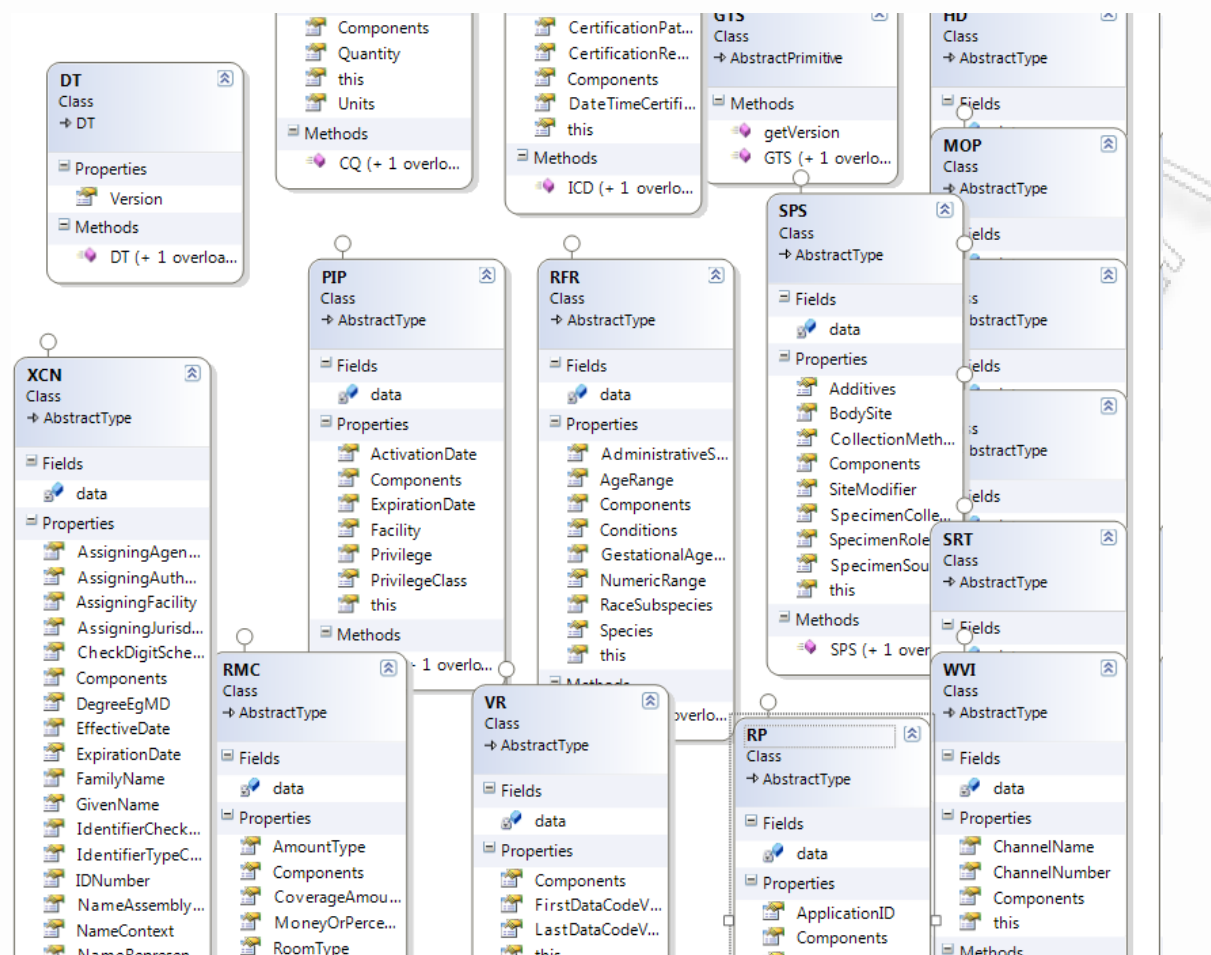
</variables>
<sequence name="RadIssueWorkflow">
  <receive name="IssueRadRequest" partnerLink="external"
portType="tns:IssueRadRequestWorkflowServiceSoap" operation="tns:IssueRadRequestWorkflow"
variable="RadIssueWorkflowMessage" createInstance="yes" />
  <receive name="ScheduleRadProcedure" partnerLink="external"
portType="tns:ScheduleRadProcedureWorkflowServiceSoap"
operation="tns:ScheduleRadProcedureIssueWorkflow" variable="RadIssueWorkflowMessage"
createInstance="yes" />
  <flow name="RadIssueBpelFlow">
    <sequence name="RadIssueBpelSquence">
      <receive name="PerformRadRequest" partnerLink="external"
portType="tns:PerformRadRequestWorkflowServiceSoap"
operation="tns:PerformRadRequestIssueWorkflow" variable="RadIssueWorkflowMessage"
createInstance="yes" />
    </sequence>
    <sequence name="RadIssueBpelSquence2">
      <receive name="ChargePatAccount" partnerLink="external"
portType="tns:ChargePatAccountWorkflowServiceSoap" operation="tns:ChargePatAccountWorkflow"
variable="ChargePatAccountWorkflowMessage" createInstance="yes" />
    </sequence>
  </flow>
  <receive name="InterpretRadResults" partnerLink="external"
portType="tns:InterpretRadResultsWorkflowServiceSoap"
operation="tns:InterpretRadResultsWorkflow" variable="InterpretRadResultsWorkflowMessage"
createInstance="yes" />
  <invoke name="WriteLine" partnerLink="file" portType="tns:FileServiceSoap"
operation="tns:writeline" inputVariable="writeLineIn" />
</sequence>
</process>

```

Με τη χρήση των συγκεκριμένων βιβλιοθηκών, το σύστημα μπορεί να εισάγει και να εξάγει BPEL ροές εργασίας προκειμένου να χρησιμοποιηθούν από άλλα συστήματα διαφορετικών κατασκευαστών.

### Χρήση HL7 από την εφαρμογή

Για την εξαγωγή του ιατρικού φακέλου σε HL7 μέσα από την εφαρμογή χρησιμοποιήθηκε η βιβλιοθήκη ανοικτού κώδικα NHapi (17) η οποία παρέχει μοντελοποιημένες όλες τις κλάσεις που απαιτούνται για τη μετατροπή της πληροφορίας του ιατρικού φακέλου ασθενών σε HL7 έκδοση 2.5. Παρακάτω μπορείτε να δείτε μία εικόνα από τις κλάσεις που περιέχει το NHapi.



ΕΙΚΟΝΑ 16 HL7 ΚΛΑΣΕΙΣ ΤΟΥ ΝΗΑΡΙ

Οι κλάσεις αυτές μπορούν να χρησιμοποιηθούν από κάποιο εξαγωγικό λογισμικό (exporter – migrator), το οποίο θα μπορείτε είτε να

- Λειτουργεί σαν διαμεσολαβητής (proxy) μεταξύ των υφιστάμενων υπηρεσιών και κάποιου που ενδιαφέρεται να παραλάβει την ιατρική πληροφορία σε HL7, με τον διαμεσολαβητή να αναλαμβάνει την μετατροπή ή
- Να συνδέεται κατευθείαν στη βάση δεδομένων του συστήματος και να εξάγει την πληροφορία σε HL7 στον οποιοδήποτε τη ζητήσει. Βεβαίως, σε αυτή την περίπτωση τίθενται θέματα ασφαλείας του συστήματος τα οποία η οποιαδήποτε προσέγγιση θα πρέπει να τα λάβει υπόψη της πολύ σοβαρά.

## 7. Συμπεράσματα – χρήση ηλεκτρονικού φάκελου ασθενούς στην Ελλάδα

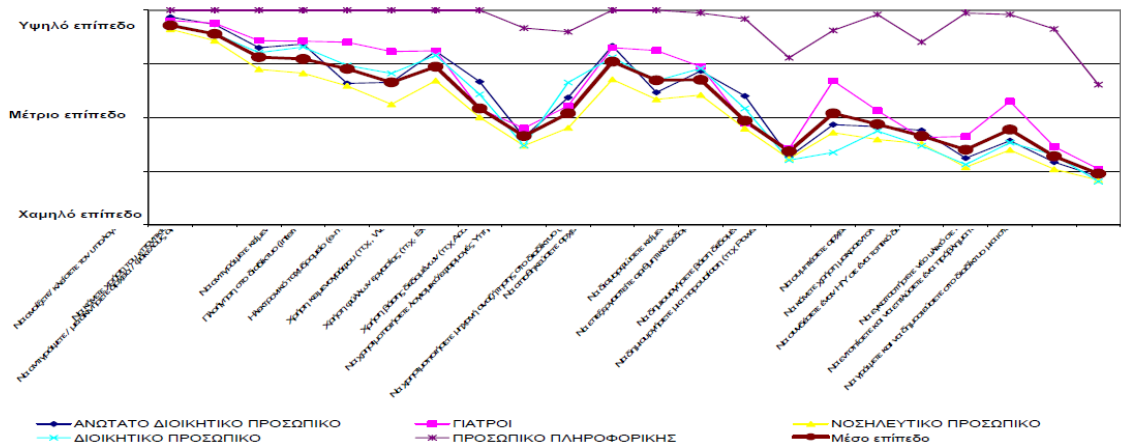
---

Με τη χρήση της υπηρεσιοστρεφούς αρχιτεκτονικής διαδικτύου αντιμετωπίζονται τα περισσότερα προβλήματα συμβατότητας και διαλειτουργικότητας μεταξύ προγραμμάτων λογισμικού που προέρχονται από διαφορετικούς κατασκευαστές και (μπορεί να) εκτελούνται σε τελείως ασύμβατες μεταξύ τους πλατφόρμες. Παρόλα αυτά, υπάρχουν ακόμα διάφορα ανοιχτά θέματα που αφορούν την απόδοση, την ασφάλεια, την ποιότητα επικοινωνίας, την αρχιτεκτονική κατασκευής και τον τρόπο ενσωμάτωσης υπηρεσιών σε ένα σύστημα. Επίσης, οι δύο κυρίαρχες τεχνολογίες για υπηρεσίες διαδικτύου (SOAP και REST) ακόμα «μάχονται» μεταξύ τους, με την δεύτερη να έχει κερδίσει σε εφαρμογές πιο κοντά στους απλούς χρήστες (facebook, twitter) ενώ η πρώτη είναι αναντικατάστατη στον χώρο των επιχειρήσεων.

Σχετικά με τον ηλεκτρονικό φάκελο ασθενών, τα οφέλη που μπορεί να αποκομίσει μια υγειονομική περιφέρεια ή ακόμα και ένα ολόκληρο κράτος με την υιοθέτησή σου είναι πάρα πολλά. Το σύστημα το οποίο αναπτύχθηκε βασίζεται σε πρότυπες τεχνολογίες και έχει εύκολη εγκατάσταση, συντήρηση και χρήση. Στη χώρα μας όμως είναι αρκετά δύσκολη η περαιτέρω εισχώρηση της τεχνολογίας στο χώρο της υγείας, ειδικά σε δημόσιους φορείς. Σε σχετική έρευνα που διεξήχθη υπό την αιγίδα του Παρατηρητηρίου για την Κοινωνία της Πληροφορίας, σε 1536 εργαζομένους σε φορείς υγείας

- Το 87% απάντησε πως έχει χρησιμοποιήσει ηλεκτρονικό υπολογιστή αλλά η γνώση αυτή προήλθε από προσωπική διάθεση για ενασχόληση
- Το 61% έχει πρόσβαση σε ηλεκτρονικό υπολογιστή στην υπηρεσία τους αλλά μόνο το 34% τον χρησιμοποιεί για την εργασία του (ελάχιστοι σε καθημερινή βάση)

- Το επίπεδο των δεξιοτήτων στη χρήση υπολογιστή είναι υψηλό για την πραγματοποίηση απλών λειτουργιών ενώ σχετικά χαμηλό σε πιο εξειδικευμένες ενέργειες



ΣΧΗΜΑ 45 ΔΕΞΙΟΤΗΤΕΣ ΣΤΗ ΧΡΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΥΠΟΛΟΓΙΣΤΗ (ΠΑΡΑΤΗΡΗΤΗΡΙΟ ΓΙΑ ΤΗΝ ΚΟΙΝΩΝΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ)

- Εξειδικευμένες εφαρμογές χρησιμοποιεί μόνο το 5% (κυρίως πληροφοριακά συστήματα υγείας και ηλεκτρονικούς φακέλους ασθενών)
- Το 76% από αυτούς που δεν γνωρίζουν να χρησιμοποιούν ηλεκτρονικό υπολογιστή απάντησε ότι επιθυμεί να αποκτήσει σχετικές γνώσεις

Τα βασικά συμπεράσματα εδώ είναι

- Η πλειονότητα των εργαζομένων γνωρίζει να χρησιμοποιεί ηλεκτρονικό υπολογιστή λόγω της προσωπικής τους ενασχόλησης με το αντικείμενο
- Υπάρχει σύγχυση και έλλειψη κατανόησης σχετικά με εξειδικευμένες εφαρμογές
- Χρειάζεται βασική εκπαίδευση σε αρκετούς χρήστες αλλά και εξειδικευμένη εκπαίδευση στο μεγαλύτερο σύνολο του προσωπικού

Από τη μέχρι τώρα χρήση τεχνολογιών πληροφορικής σε δημόσια νοσοκομεία βρέθηκε ότι

- Μειώθηκαν τα λάθη
- Βελτίωση ποιότητας υπηρεσιών
- Μείωση του χρόνου εξυπηρέτησης

Συνεπώς, η υλοποίηση έργων μέσω της Κοινωνίας της Πληροφορίας κρίνεται θετική, αλλά πρέπει να διορθωθούν/βελτιωθούν τα κάτωθι

- Καλύτερος συντονισμός με φορείς υγείας
- Καθυστερήσεις από έλλειψη προσωπικού

- Έλλειψη σωστού χρονοπρογραμματισμού σε αρκετά έργα
- Καλύτερη εκπαίδευση (και σε βασικό επίπεδο)
- Κίνητρα για την εκπαίδευση του προσωπικού

Τέλος, η στρατηγική που θα μπορούσε να χαραχτεί σε αυτό τον τομέα στη χώρα μας θα μπορούσε να περιλαμβάνει τα παρακάτω

- Πλήρης εφαρμογή του AMKA
- Μεγαλύτερη χρήση ηλεκτρονικών συναλλαγών
- Συστήματα με διαλειτουργικότητα (χρήση υπηρεσιών διαδικτύου)
- Εφαρμογή πληροφοριακών συστημάτων οικονομικής διαχείρισης σε όλους τους φορείς
- Σταθερή χρηματοδότηση
- Δια βίου εκπαίδευση
- Υλοποίηση ενιαίων κανόνων και προτύπων [18]

## 8. Βιβλιογραφία

---

1. **Skonnard, Aaron.** Understanding SOAP. *Microsoft Developer Network*. [Ηλεκτρονικό] DevelopMentor, March 2003. <http://msdn.microsoft.com/en-us/library/ms995800.aspx>.
2. **F. Curbera, Y. Golland, J. Klein, F. Leymann, D. Roller, S. Thatte, S. Weerawarana.** Business Process Execution Language for Web Services. *Microsoft Developer Network*. [Ηλεκτρονικό] IBM, Microsoft, BEA, August 2002. [http://msdn.microsoft.com/en-US/library/ee251592\(v=BTS.10\).aspx](http://msdn.microsoft.com/en-US/library/ee251592(v=BTS.10).aspx).
3. **Μαρίνος Θεμιστοκλέους, Βασιλική Μαντζάνα.** *Υπηρεσίες Παγκοσμίου Ιστού και Υπηρεσιοστρεφείς Αρχιτεκτονικές*. 2010.
4. **Microsoft.** Windows Azure Development Center. *Microsoft Developer Network*. [Ηλεκτρονικό] <http://msdn.microsoft.com/el-gr/windowsazure/>.
5. —. SQL Azure Developer Center. *Microsoft Developer Network*. [Ηλεκτρονικό] <http://msdn.microsoft.com/el-gr/windowsazure/sqlazure/default.aspx>.
6. —. Windows Azure AppFabric Developer Center. *Microsoft Developer Network*. [Ηλεκτρονικό] <http://msdn.microsoft.com/el-gr/windowsazure/cc994381.aspx>.

7. **Chappell, David.** David Chappell blog. *OPINARI*. [Ηλεκτρονικό] March 2009.  
[http://download.microsoft.com/download/0/8/7/087A3AE1-2880-4452-88DD-09398D0A522A/Introducing\\_Windows\\_Azure.doc](http://download.microsoft.com/download/0/8/7/087A3AE1-2880-4452-88DD-09398D0A522A/Introducing_Windows_Azure.doc).
8. **Microsoft.** About Windows Azure. *Microsoft Developer Network*. [Ηλεκτρονικό]  
<http://msdn.microsoft.com/library/dd179442.aspx>.
9. **Davis, Mike και Garets, Dave.** *Electronic Patient Records*. October 2005.
10. **Committee, HIMSS Electronic Health Record.** *HIMSS Electronic Health Record Definitional Model*.
11. *Electronic Patient Records*. **D. Garets, M. Davis.** s.l. : Healthcare Informatics, 2005.
12. **Εργαστήριο Ιατρικής Πληροφορικής, ΑΠΘ.** Ηλεκτρονικός Φάκελος Υγείας και Διαχείριση της Ιατρικής Πληροφορίας. Θεσσαλονίκη : ΑΠΘ.
13. *Use and Acceptance of an Electronic Health Record: Factors Affecting Physician Attitudes*. **Morton, Mary Elizabeth.** 2008.
14. **Burns, D.** *Systemic Action Research: A strategy for whole system change*. Bristol : Policy Press.
15. **Carr, W. & Kemmis, S.** *Becoming Critical*. s.l. : Falmer Press, 1986.
16. BPEL for Windows Workflow Foundation March CTP. [Ηλεκτρονικό]  
<http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=20222>.
17. <http://nhapi.sourceforge.net/home.php>. [Ηλεκτρονικό]
18. **Κουντζέρης, Δρ. Α.** *Η χρήση τεχνολογιών πληροφορικής και επικοινωνιών στην υγεία - συμπεράσματα από περιπτώσεις υλοποίησης έργων*. s.l. : Παρατηρητήριο για την κοινωνία της πληροφορίας, 2008.
19. **OASIS.** [Ηλεκτρονικό] <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>.
20. —. [Ηλεκτρονικό] <http://www.w3.org/TR/wsdl20-primer/>.
21. **W3C.** [Ηλεκτρονικό] <http://www.w3.org/TR/wsdl20/>.
22. W3C Web Services Activity. [Ηλεκτρονικό] <http://www.w3.org/2002/ws>.
23. **Gudgin, M. και Hadley, M.** *Web Services Addressing - SOAP Binding*. s.l. : W3C Working Draft, December 2004.
24. **Nadalin, A., και συν., και συν.** *SOAP Message Security 1.0*. s.l. : OASIS, March 2004.

25. **Obasanjo, Dare.** Understanding XML. *Microsoft Developer Network*. [Ηλεκτρονικό] Microsoft, July 2003. <http://msdn.microsoft.com/en-us/library/aa468558.aspx>.
26. **Aziz, Atif και Mitchell, Scott.** An introduction to JSON. *Microsoft Developer Network*. [Ηλεκτρονικό] Microsoft, February 2007. <http://msdn.microsoft.com/en-us/library/bb299886.aspx>.
27. **Microsoft.** Overview of REST in WCF. *Microsoft Developer Network*. [Ηλεκτρονικό] Microsoft. <http://msdn.microsoft.com/en-us/netframework/dd547388>.
28. **WCF team @ Microsoft.** WCF Community Site. *Codeplex, open source repository*. [Ηλεκτρονικό] Microsoft. <http://wcf.codeplex.com>.
29. **OASIS.** *Web Services Business Process Execution Language version 2.0*. s.l. : OASIS, 2007.
30. **Tao Tao, Fethi Rabhi, Hairong Yu, Fan Xiong.** *Service Oriented Business Process Development using BPEL: A case study*. Sydney, Australia and Sichuan, China : s.n.

## 9. ΠΑΡΑΡΤΗΜΑ

---

### 9.1 Νομικό πλαίσιο για απόρρητο και ασφάλεια της επεξεργασίας

Άρθρο 10

Απόρρητο και ασφάλεια της επεξεργασίας

1. Η επεξεργασία δεδομένων προσωπικού χαρακτήρα είναι απόρρητη. Διεξάγεται αποκλειστικά και μόνο από πρόσωπα που τελούν υπό τον έλεγχο του υπεύθυνου επεξεργασίας ή του εκτελούντος την επεξεργασία και μόνο κατ' εντολή του.
2. Για τη διεξαγωγή της επεξεργασίας ο υπεύθυνος επεξεργασίας οφείλει να επιλέγει πρόσωπα με αντίστοιχα επαγγελματικά προσόντα που παρέχουν επαρκείς εγγυήσεις από πλευράς τεχνικών γνώσεων και προσωπικής ακεραιότητας για την τήρηση του απορρήτου.
3. Ο υπεύθυνος επεξεργασίας οφείλει να λαμβάνει τα κατάλληλα οργανωτικά και τεχνικά μέτρα για την ασφάλεια των δεδομένων και την προστασία τους από τυχαία ή αθέμιτη καταστροφή, τυχαία απώλεια, αλλοίωση, απαγορευμένη διάδοση ή πρόσβαση και κάθε άλλη μορφή αθέμιτης επεξεργασίας. Αυτά τα μέτρα πρέπει να εξασφαλίζουν επίπεδο ασφάλειας ανάλογο προς τους κινδύνους που συνεπάγεται η επεξεργασία και η φύση των δεδομένων που είναι αντικείμενο της επεξεργασίας. Η Αρχή παρέχει εκάστοτε οδηγίες για το βαθμό ασφάλειας των δεδομένων, καθώς και για τα μέτρα προστασίας που

είναι αναγκαίο να λαμβάνονται για κάθε κατηγορία δεδομένων, εν όψει και των τεχνολογικών εξελίξεων.

4. Αν η επεξεργασία διεξάγεται για λογαριασμό του υπευθύνου από πρόσωπο μη εξαρτώμενο από αυτόν, η σχετική ανάθεση γίνεται υποχρεωτικά εγγράφως. Η ανάθεση προβλέπει υποχρεωτικά ότι ο ενεργών την επεξεργασία τη διεξάγει μόνο κατ' εντολή του υπευθύνου και ότι οι λοιπές υποχρεώσεις του παρόντος άρθρου βαρύνουν αναλόγως και αυτόν.

## 9.2 Παραδοχές για την ανάπτυξη της εφαρμογής

Εδώ θα παρουσιάσουμε ορισμένες παραδοχές για την εφαρμογή που σαν στόχο θα έχουν να κάνουν πιο κατανοητό το πεδίο δράσης κάθε χρήστη αλλά και να κάνει πιο ξεκάθαρο τον τρόπο συνεργασίας τους. Πιο συγκεκριμένα σαν παραδοχές δεχόμαστε:

- Κάθε νοσοκομείο - ιατρικό κέντρο διαθέτει μια κεντρική γραμματεία.
- Κάθε ακτινολογικό τμήμα νοσοκομείου - ιατρικού κέντρου διαθέτει μια γραμματεία ακτινολογικού.
- Όλοι οι γιατροί ακτινολόγοι κάθε νοσοκομείου-ιατρικού κέντρου έχουν πρόσβαση στην εφαρμογή και στις επιμέρους καρτέλες των ασθενών.
- Όλοι οι ακτινολόγοι γιατροί είναι εξειδικευμένοι και μπορούν να κάνουν όλες τις ακτινολογικές εξετάσεις, καθώς επίσης και να ελέγχουν στοιχεία και παρατηρήσεις συναδέλφων τους ακτινολόγων στο πεδίο παρατηρήσεων αρμόδιων γιατρών ανά ασθενή.
- Ο διαχειριστής (administrator) του συστήματος δεν μπορεί να έχει πρόσβαση σε καρτέλες ασθενών, καθώς επίσης και σε κανενός είδους προσωπικά τους στοιχεία.
- Οποιοδήποτε στοιχείο προσβασιμότητας χρηστών στην εφαρμογή που ισχύει για το περιφερειακό νοσοκομείο ισχύει και για τα ιατρικά κέντρα τα οποία υπάγονται ή απλά συνεργάζονται μέσω του πληροφοριακού συστήματος με το περιφερειακό νοσοκομείο.