

## Πρόσκληση Σπουδαστών για την ανάληψη Πτυχιακής εργασίας.

Εισηγητής: Δρ. Νικήτας Ν. Καρανικόλας, Αναπληρωτής Καθηγητής  
Χρόνος εισήγησης: Μάρτιος 2012

### Θέμα 1<sup>ο</sup>:

**Εφαρμογή αλγορίθμου C4.5 ή άλλης γνωστής μεθόδου για Rule/Tree induction το οποίο θα χρησιμοποιηθεί για document/text classification.**

Ο αλγόριθμος C4.5 (ή όποιο πακέτο βασίζεται σε αυτόν) χρησιμοποιείται για να παράγει (εξορύξει, mine) κανόνες συμπερασμού από ένα Training Set (εκπαιδευτικό σύνολο). Για το σκοπό αυτό πρέπει να έχουν επιλεγεί τα χαρακτηριστικά (ιδιότητες) του Training Set στα οποία θα βασισθεί ο αλγόριθμος για να παράγει τους κανόνες συμπερασμού. Στην εργασία αυτή τα χαρακτηριστικά του Training Set θα έχουν επιλεγεί από προηγούμενο στάδιο και δεν θα είναι στις υποχρεώσεις των σπουδαστών που θα την αναλάβουν. Κατά αναλογία η δημιουργία του Training Set δεν ανήκει στις υποχρεώσεις των σπουδαστών που θα αναλάβουν την παρούσα εργασία. Τα αντικείμενα του Training Set καθώς και τα αντικείμενα που θα κατατάσσουν (classify) οι κανόνες που θα προκύψουν, θα είναι έγγραφα/κείμενα. Τα χαρακτηριστικά (ιδιότητες) του Training Set που θα χρησιμοποιηθούν είναι φράσεις (ακολουθίες από ρίζες λέξεων).

### Θέμα 2<sup>ο</sup>:

**Αποθήκευση HTML εγγράφων και εξαγωγή πολυγλωσσικού κειμένου.**

Οι browsers δίνουν τη δυνατότητα αποθήκευσης του περιεχομένου μιας ιστοσελίδας στον τοπικό δίσκο του υπολογιστή του χρήστη. Για το σκοπό αυτό δημιουργούν ένα αρχείο με όνομα που προσδιορίζει ο χρήστης και κατάληξη html. Επίσης δημιουργούν έναν κατάλογο με το ίδιο όνομα και κατάληξη files. Στον κατάλογο αυτό αποθηκεύουν τα σημαντικότερα από πλευράς παρουσίασης συνοδευτικά έγγραφα (κυρίως images) στα οποία κάνει άμεση αναφορά η κύρια σελίδα που αποθηκεύεται. Τα source paths των συνοδευτικών εγγράφων αντικαθίστανται στο .html αρχείο με paths στον .files κατάλογο.

Στα πλαίσια της παρούσας εργασίας θα πρέπει να κατασκευασθεί πρόγραμμα (υπό μορφή ρουτίνας σε DLL file) η οποία θα δέχεται ως παράμετρο την URL διεύθυνση μιας κύριας ιστοσελίδας και το πλήρες όνομα (path και file name) ενός αρχείου και στη συνέχεια θα αποθηκεύει με ανάλογο τρόπο τα περιεχόμενα της κύριας ιστοσελίδας όπως γίνεται και από τους browsers (θα χρησιμοποιεί κατάληξη .htm και .dir για τη σελίδα και τον κατάλογο, αντίστοιχα). Η ρουτίνα αυτή θα πρέπει επίσης να ξεχωρίζει μόνο το κείμενο και να το αποθηκεύει σε ένα αρχείο με το ίδιο όνομα και κατάληξη .mht (Multi-National Text). Το .mht αρχείο θα περιέχει μόνο το απλό κείμενο και ορισμένες γραμμές ένδειξης αλλαγής της γλώσσας. Οι ενδείξεις αλλαγής της γλώσσας είναι γραμμές με τη μορφή:

```
lang=xxxx<CR>
```

όπου το xxxx είναι ένας αριθμός σαν και αυτόν που χρησιμοποιούν τα αρχεία .RTF (Rich Text Format) με τη μορφή:

\langxxxx

Για παράδειγμα ο επόμενος πίνακας δείχνει ορισμένες γλώσσες και τον κωδικό τους όπως τον συναντάμε στα RTF αρχεία, δείχνει επίσης ενδεικτικό font και Codepage:

Lang code	Language	ενδεικτικό font	Codepage
1061	Estonian	Times New Roman	1252 (Latin 1)
1033	English	Times New Roman	1252 (Latin 1)
1052	Alban	Times New Roman	1252 (Latin 1)
1038	Hungarian	Times New Roman CE	1250 (Latin 2)
1050	Croatian	Times New Roman CE	1250 (Latin 2)
1045	Polish	Times New Roman CE	1250 (Latin 2)
1060	Slovenian	Times New Roman CE	1250 (Latin 2)
1058	Ukrainian	Times New Roman Cyr	1251
1059	Byelorussian	Times New Roman Cyr	1251
1049	Russian	Times New Roman Cyr	1251
1026	Bulgarian	Times New Roman Cyr	1251
3098	Serbian	Times New Roman Cyr	1251
1032	Greek	Times New Roman Greek	1253
1055	Turkish	Times New Roman Turkish	1254 (Latin 5)
1062	Leetonian	Times New Roman Baltic	1257
1063	Lithuanian	Times New Roman Baltic	1257
1048	Romanian		
1029	Chez		
1051	Slovakian		
		Times New Roman Hebrew	1255
		Times New Roman Arabic	1256

Στα html έγγραφα η γλώσσα (όταν ορίζεται) ορίζεται με tags σαν τα δύο επόμενα:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1253">  
<meta http-equiv="Content-Language" content="greek">
```

Όταν δεν υπάρχει ένδειξη γλώσσας (δεν υπάρχουν τα δύο προηγούμενα tags) ή ορισμένοι χαρακτήρες γράφονται με κάποια κωδικοσελίδα διαφορετική από αυτή που ορίζουν τα παραπάνω tags τότε χρησιμοποιούνται ακολουθίες διαφυγής. Το επόμενο παράδειγμα περιέχει την κωδικοποίηση της φράσης «Το καλύτερο» με ακολουθίες διαφυγής:

```
&#932;&#959; &#954;&#945;&#955;&#973;&#964;&#949;&#961;&#959;  
Τ ο κ α λ ύ τ ε ρ ο
```

Από τα παραπάνω γίνεται αντιληπτό ότι δεν είναι πάντα δυνατό να εξαχθεί πολύ εύκολα η γλώσσα και να δημιουργηθεί η κατάλληλη ένδειξη στο .mht αρχείο. Ένα ακόμα παράδειγμα (που ενισχύει αυτή την άποψη) θα μπορούσε να αποτελέσει ένα html αρχείο που θα περιείχε το επόμενο tag:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

αλλά δεν θα περιείχε κάτι σαν:

```
<meta http-equiv="Content-Language" content="....">
```

Στην περίπτωση αυτή θα έπρεπε να γίνει μια στατιστική ανάλυση η οποία να «έβλεπε» ποιες χαρακτήρες του Κυριλλικού αλφαβήτου δεν χρησιμοποιούνται στο html έγγραφο προκειμένου να επιλέξει την πιθανότερη μεταξύ των υποψηφίων γλωσσών: Ukrainian, Byelorussian, Russian, Bulgarian και Serbian.

### **Θέμα 3<sup>ο</sup>: Αποσύνθεση συνθετικών ιατρικής εικόνας DICOM.**

Το πρωτόκολλο DICOM έχει σχεδιασθεί για την ανταλλαγή ιατρικής εικόνας. Η βασική του ιδέα είναι ότι μαζί με την εικόνα συσκευάζονται και στοιχεία ασθενούς. Στο πρωτόκολλο αυτό οι συμμετέχοντες μπορεί να έχουν ρόλο χρήστη (Service Class User – SCU) ή ρόλο server (Service Class Provider – SCP). Στα πλαίσια της παρούσας πτυχιακής εργασίας θα πρέπει να αναπτυχθεί λογισμικό το οποίο θα εντάσσεται σε μια DICOM επικοινωνία σε ρόλο SCU, θα αλληλεπιδρά με ένα Dicom server (SCP) για να ζητήσει και να παραλάβει Dicom δεδομένα και τελικά θα αποσυνθέτει τα στοιχεία αυτά σε εικόνα και στοιχεία ασθενούς (συνήθως πεδία κειμένου). Το αποτέλεσμα από την αποσύνθεση θα παρουσιάζεται στον χρήστη και θα αποθηκεύεται σε δύο αρχεία. Το ένα θα είναι ένα αρχείο εικόνας (το πλέον απλό και συνηθισμένο format που θα μπορούσε να χρησιμοποιηθεί είναι το Bitmap – BMP) και το άλλο θα είναι αρχείο κειμένου με την ακολουθεί γραμμογράφηση:

<όνομα πεδίου>:<τιμή πεδίου><CR>

<όνομα πεδίου>:<τιμή πεδίου><CR>

<όνομα πεδίου>:<τιμή πεδίου><CR>

...

### **Θέμα 4<sup>ο</sup>: Αποθήκευση PDF εγγράφων και εξαγωγή πολυγλωσσικού κειμένου.**

Τα αρχεία PDF (Portable Document Format) παρέχουν τη δυνατότητα μεταφοράς εγγράφων στη τελική τους μορφή. Τα έγγραφα αυτά εμφανίζονται και εκτυπώνονται πάντα με τον ίδιο ακριβώς τρόπο και δεν επιδέχονται μεταβολές. Η ανάγνωση τους γίνεται συνήθως με τον Acrobat Reader που διατίθεται δωρεάν.

Στα πλαίσια της παρούσας εργασίας θα πρέπει να κατασκευασθεί πρόγραμμα (υπό μορφή ρουτίνας σε DLL file) η οποία θα δέχεται δύο παραμέτρους. Η πρώτη παράμετρος θα είναι το πλήρες όνομα (path και file name) ενός αρχείου PDF. Από αυτό θα εξάγει το κείμενο που περιέχει, χωρίς να μας ενδιαφέρουν άλλα χαρακτηριστικά όπως εικόνες, στοιχεία μορφοποίησης, κλπ. Το κείμενο θα έχει μορφή Multi-National Text (mht) και θα αποθηκεύεται σε ένα αρχείο με το όνομα που προσδιορίζει ο χρήστης (στη δεύτερη παράμετρο). Το mht αρχείο θα περιέχει μόνο το απλό κείμενο και ορισμένες γραμμές ένδειξης αλλαγής της γλώσσας. Οι ενδείξεις αλλαγής της γλώσσας είναι γραμμές με τη μορφή:

lang=xxxx<CR>

όπου το xxxx είναι ένας αριθμός σαν και αυτόν που χρησιμοποιούν τα αρχεία .RTF (Rich Text Format) με τη μορφή. Περισσότερα για τους κωδικούς γλώσσας υπάρχουν στην πτυχιακή με θέμα «Αποθήκευση HTML εγγράφων και εξαγωγή πολυγλωσσικού κειμένου».

Είναι προφανές ότι η εύρεση της γλώσσας δεν είναι πάντα μια προφανής εργασία και θα πρέπει να αναζητηθούν κατάλληλες λύσεις.

## Θέμα 5<sup>ο</sup>:

### Ψηφιοποίηση εγγράφων σε Linux με τεχνολογία SANE.

Ανάπτυξη εφαρμογής σε περιβάλλον Linux, για την ψηφιοποίηση εγγράφων. Η εφαρμογή πρέπει να χρησιμοποιεί την τεχνολογία SANE για την επικοινωνία με τις συσκευές ψηφιοποίησης (scanners, ψηφιακές φωτογραφικές μηχανές).

Η τεχνολογία SANE μπορεί να χαρακτηριστεί ως ανάλογη της τεχνολογίας TWAIN που έχει επικρατήσει στα MS Windows και στα MAC. Η τεχνολογία SANE έχει το πλεονέκτημα ότι επιτρέπει την επικοινωνία με απομακρυσμένες (remote) συσκευές ψηφιοποίησης. Μια άλλη σημαντική παράμετρος του SANE είναι ότι δεν βασίζεται σε κάποια διεπαφή (interface) που ενσωματώνει ο κάθε κατασκευαστής στον οδηγό (driver) που προσφέρει μαζί με τη συσκευή του. Αντίθετα επιτρέπει στον προγραμματιστή (developer) να επικοινωνήσει, με τη χρήση API, με τη συσκευή προκειμένου να προβεί σε ρυθμίσεις και να εκκινήσει εργασίες της συσκευής (όπως για παράδειγμα scanning). Έτσι ο προγραμματιστής (developer) έχει τη δυνατότητα να χτίσει τη δική του διεπαφή για το περιβάλλον που θέλει (console (text mode), GNOME, KDE ή X11 Window) και στη συνέχεια αυτή η διεπαφή να αξιοποιηθεί για την επικοινωνία με κάθε πιθανή συσκευή ψηφιοποίησης. Το αντίθετο συμβαίνει με το TWAIN όπου ο χρήστης χρησιμοποιώντας την ίδια εφαρμογή βλέπει διαφορετικό παράθυρο όταν επικοινωνεί με διαφορετική συσκευή (διαφορετικό driver) ψηφιοποίησης.

Ακολουθούν διευθύνσεις όπου μπορεί να ευρεθούν πληροφορίες για τις τεχνολογίες SANE και TWAIN.

#### SANE

<http://www.sane-project.org/intro.html>  
<http://www.sane-project.org/lj98/doc000.html>

#### TWAIN

<http://www.twain.org/>

## Θέμα 6<sup>ο</sup>:

### Σύλληψη ακίνητης εικόνας από Video σε Linux με τεχνολογία V4L.

Ανάπτυξη εφαρμογής σε περιβάλλον Linux, για σύλληψη ακίνητης εικόνας από πηγές Video (S-Video, Composite, RGB) με τη χρήση ενός εκ των API V4L ή V4L2.

Το Video for Linux (V4L) είναι η αντίστοιχη τεχνολογία με την τεχνολογία Video for Windows (VfW) που χρησιμοποιείται στο ΛΣ MS Windows. Το Video for Linux είναι το πρωτότυπο API για capture/overlay που εμπεριέχεται στους Linux kernels. Εμφανίσθηκε στην έκδοση 2.1.x του Linux kernel. Το Video for Linux Two (V4L2) είναι η δεύτερη γενιά του video for Linux API. Ολοκληρώθηκε στον kernel 2.5.x του Linux. Δυστηχώς, το v4l2 δεν είναι συμβατό προς τα πίσω και κατά συνέπεια οι drivers που έχουν γραφτεί για v4l πρέπει να προσαρμοσθούν. Αυτό σημαίνει ότι μια εφαρμογή που χρησιμοποιεί κλήσεις v4l2 δεν μπορεί να αξιοποιήσει (δεν μπορεί να μιλήσει σε) ένα v4l driver. Αντίθετα, υπάρχει ένα επίπεδο συμβατότητας προς τα πίσω ενσωματωμένο στον v4l2 high-level driver. Αυτό το επίπεδο επιτρέπει σε εφαρμογές που έχουν γραφτεί για το v4l API να συνεχίζουν να ομιλούν με v4l2 drivers.

Η εφαρμογή θα πρέπει να υποστηρίζει ένα από τα δύο APIs αλλά θα ήταν πρωτιμότερο να υποστηρίζει αμφότερα. Ακολουθούν διευθύνσεις όπου μπορεί να ευρεθούν πληροφορίες για V4L, V4L2 και Vfw.

#### Video for Linux

<http://www.thedirks.org/v4l2>

<http://linux.bytesex.org/v4l2>

<http://www.video4linux.net/>

#### Video for Windows (Vfw)

<http://www.videoforwindows.com/samplecode/>

<http://www.videoforwindows.com/articles/>

<http://ej.bantz.com/video>

### **Θέμα 7<sup>ο</sup>:**

#### **Επικοινωνία με mail server για λήψη και αρχειοθέτηση μηνυμάτων (emails)**

Ανάπτυξη εφαρμογής, με γραφικό περιβάλλον, που θα επικοινωνεί απευθείας με mail servers με σκοπό τη λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου. Τα ληφθέντα μηνύματα θα μπορούν να αρχειοθετηθούν αδιάσπαστα (δηλαδή μήνυμα και προσαρτήματα μαζί) σε αρχεία .eml και .zip ή θα μπορούν να αρχειοθετηθούν τα επιμέρους συνθετικά τους (κυρίως κείμενο και προσαρτήματα) στα native format που αυτά έχουν δημιουργηθεί.

Η εφαρμογή θα «μιλάει» απευθείας με τους mail servers και δεν θα βασίζεται στα API (Application Programming Interfaces) που παρέχουν οι Mail Clients. Δηλαδή δεν θα χρησιμοποιηθεί κάτι ανάλογο με το MAPI των Windows. (Το MAPI – Mail ή Messaging API – επιτρέπει σε μια εφαρμογή να επικοινωνήσει με τον email client – Eudora, Outlook, κλπ – για την αποστολή και τη λήψη μηνυμάτων ηλεκτρονικού ταχυδρομείου.)

Η εφαρμογή θα υποστηρίζει δύο τουλάχιστον διαδεδομένους mail Servers (έναν για Windows – π.χ. exchange – και έναν για Linux – π.χ. Postfix ή SendMail).

Υπάρχει ένας σημαντικός αριθμός από scripting languages, βιβλιοθήκες και καθορισμένες διαδικασίες (Requests For Comments, RFCs) που υποστηρίζουν την άμεση επικοινωνία με τον Mail Server για την αποστολή και τη λήψη email.

Ακολουθούν διευθύνσεις όπου μπορεί να ευρεθούν οι ως άνω βιβλιοθήκες και τα RFCs που διέπουν την επικοινωνία για αποστολή και λήψη email.

#### **Email libraries**

*Python:*

<http://www.eskimo.com/~jet/python/examples/mail/smtp1.html>

*Perl:*

<http://www.cs.cf.ac.uk/Dave/PERL/node180.html#SECTION00193250000000000000>

*PHP:*

<http://www.phpmail.com>

*C:*

<http://www.stafford.uklinux.net/libesmtplib>

*JavaMail API:*

<http://java.sun.com/products/javamail/>

#### **Email Relevant RFCs**

*SMTP*

<http://www.faqs.org/ftp/rfc/pdf/rfc821.txt.pdf>  
*POP v 3*  
<http://www.faqs.org/ftp/rfc/pdf/rfc1939.txt.pdf>  
*IMAP version 4*  
<http://www.faqs.org/ftp/rfc/pdf/rfc1730.txt.pdf>  
*IMAP version 4.1*  
<http://www.faqs.org/ftp/rfc/pdf/rfc2060.txt.pdf>  
*IMAP4 implementation recommendations*  
<http://www.faqs.org/ftp/rfc/pdf/rfc2683.txt.pdf>

## **Θέμα 8<sup>ο</sup>: Αξιοποίηση της κουλτούρας των μηχανών αναζήτησης σε βάσεις ακαδημαϊκής βιβλιογραφίας**

Προηγούμενες μελέτες έχουν καταγράψει την αποκρυσταλλωμένη (πρακτικά επιβεβλημένη μέσα από εφαρμογές) μέθοδο αλληλεπίδρασης με τους χρήστες των μηχανών αναζήτησης. Την αποκρυσταλλωμένη αυτή αντίληψη την ονομάζουμε *search culture* (κουλτούρα αναζητήσεων). Σε άλλες προηγούμενες εργασίες (μελέτες) έχει αξιοποιηθεί η κουλτούρα αναζητήσεων για την εκτίμηση του ερευνητικού έργου ενός ακαδημαϊκού/ερευνητή. Στις τελευταίες μελέτες έχει γίνει εστίαση σε συνδυασμούς από συγκεκριμένα εργαλεία (Google Scholar και Scopus) και σε συγκεκριμένα templates ερωτημάτων. Αυτά τα templates ερωτημάτων βασίζονται στην πλήρη εγγραφή (ολόκληρη τη δομημένη βιβλιογραφική εγγραφή) αλλά και στο πλήρες κείμενο (*full text*) που μπορεί να τη συνοδεύει (*abstract* και *full text*).

Τα templates ερωτημάτων που έχουν χρησιμοποιηθεί επιδιώκουν να βρουν:

- αναφορές ενός συγγραφέα  
(Citations of an author),
- αναφορές ενός συγγραφέα χωρίς αυτοαναφορές  
(Citations of an author, without self-citations),
- αναφορές ενός συγγραφέα χωρίς αυτοαναφορές από τον ίδιο ή από συνεργάτη του  
(Citations of an author, without self-citations from the author or specific collaborator),
- αναφορές που έχουν λάβει δύο ερευνητές στις κοινές τους δημοσιεύσεις  
(Citations that have received two (or more) researchers in their common publications),
- ετεροαναφορές που έχουν λάβει δύο ερευνητές στις κοινές τους δημοσιεύσεις  
(Citations that have received two (or more) researchers in their common publications, without self-citations),
- αναφορές σε συγκεκριμένη δημοσίευση χωρίς αυτοαναφορές από τον πρώτο συγγραφέα  
(Citations of a specific article title, without self-citations from the first author),
- ετεροαναφορές σε συγκεκριμένη δημοσίευση  
(Citations of a specific article title, without self-citations from any of the contributing authors).

Στην εργασία αυτή, οι βασικές επιδιώξεις είναι:

(α) Να γίνει σύγκριση με άλλες ανάλογες (ή συναφείς προσπάθειες). Ενδεικτικά αναφέρεται η σύγκριση με το σύστημα "Publish or Perish".

(β) Να γίνει διεύρυνση και σε άλλα εργαλεία, επιπλέον από αυτά που έχουν χρησιμοποιηθεί (επιπλέον από τα Google Scholar και Scopus).

(γ) Να επεκταθεί η διάσταση του πειράματος και του συνόλου των αξιολογούμενων ερευνητών.

Περισσότερο προηγμένες επιδιώξεις είναι:

(δ) Αυτόματος εντοπισμός των ερευνητικών περιοχών ενός ερευνητή.

(ε) Υπολογισμό της ατομικής συνεισφοράς του κάθε ερευνητή σε μια ομαδική (με περισσότερους από έναν συγγραφείς) εργασία.

(στ) Στην εύρεση των ομάδων συνεργαζόμενων ερευνητών (communities).

Ένα καλό σημείο εκκίνησης για την παρούσα εργασία είναι το επόμενο άρθρο:

[http://www.cs.teiath.gr/nnk/papers/B30\\_cr.pdf](http://www.cs.teiath.gr/nnk/papers/B30_cr.pdf)

### **Θέμα 9<sup>ο</sup>:**

#### **Προσωπικά και Οικιακά Ρομπότ (Personal and Domestic Robots) καθοδηγούμενα από κατά περίπτωση κατασκευασμένο λογισμικό**

Στην εργασία αυτή θα αποτυπωθούν οι δυνατότητες των προσωπικών και οικιακών Ρομπότ (που διαθέτουν περιορισμένες λειτουργικές δυνατότητες και ταυτόχρονα έχουν χαμηλό κόστος). Θα δοθεί έμφαση σε εκείνα που δεν επιτελούν προκαθορισμένες λειτουργίες αλλά είναι εφοδιασμένα με δυνατότητες καθορισμού της λειτουργικότητάς τους με βάση λογισμικό που μπορεί να εγκαθίσταται και να αντικαθίσταται σε αυτά. Για την πρακτική διάσταση του ζητήματος θα γίνει επιλογή ενός προσωπικού/οικιακού ρομπότ και για αυτό θα αναπτυχθούν δύο ή περισσότερες ανεξάρτητες λειτουργικές εφαρμογές που θα επιδεικνύουν τις διαφορετικές του χρήσεις.

Επιδίωξη της εργασίας είναι ο καθορισμός της λειτουργικότητας (προγραμματισμός) με τη βοήθεια γνωστών γλωσσών προγραμματισμού (C, C++, Java) και όχι με προσανατολισμένες (proprietary) γλώσσες.

### **Θέμα 10<sup>ο</sup>:**

#### **Πλατφόρμες κατασκευής πρωτοτύπων ηλεκτρονικών συσκευών**

Στην εργασία αυτή θα αποτυπωθούν οι δυνατότητες των πλατφορμών κατασκευής πρωτοτύπων ηλεκτρονικών συσκευών (electronics prototyping platform). Θα δοθεί έμφαση στις πλατφόρμες που «αισθάνονται» το περιβάλλον (sense the environment) και μπορούν να «επιδρούν» σε αυτό (affect its surroundings). Λέμε ότι «αισθάνονται» το περιβάλλον γιατί μπορούν να λαμβάνουν δεδομένα από μια ποικιλία από αισθητήρες (receiving input from a variety of sensors). Λέμε ότι «επιδρούν» στο περιβάλλον γιατί μπορούν να καθοδηγούν φώτα, μοτέρ και άλλες παρόμοιες συσκευές (controlling lights, motors and other actuators). Οι πλατφόρμες στις οποίες προασανατολίζεται η εργασία αυτή είναι εκείνες που διαθέτουν δυνατότητες καθορισμού της λειτουργικότητάς τους με βάση λογισμικό που μπορεί να εγκαθίσταται και να αντικαθίσταται σε αυτές. Για την πρακτική διάσταση του ζητήματος θα γίνει επιλογή μίας τέτοιας πλατφόρμας και θα αναπτυχθούν δύο ή περισσότερες ανεξάρτητες λειτουργικές εφαρμογές (προγράμματα) που θα επιτυγχάνουν τη λειτουργία δύο διαφορετικών ηλεκτρονικών συσκευών με την ίδια (την επιλεγμένη) πλατφόρμα. Οι επόμενες δύο εφαρμογές αναφέρονται για την πληρέστερη κατανόηση των ζητούμενων:

- Αναγνώριση πτηνών σε ένα πεδίο (π.χ. ένα οικόπεδο) και δημιουργία ήχων (π.χ. από μία κόρνα) που (θα τρομάζουν και) θα διώχνουν τα πτηνά.
- Κατανόηση της θέσεως του ήλιου και περιστροφή (σε έναν άξονα, με ένα μοτέρ) ενός δείκτη (μίας ράβδου). Δηλαδή η ράβδος θα δείχνει πάντα προς το σημείο από το οποίο έρχεται το φως (θα δείχνει πάντα στην εστία του φωτός).

Επιδίωξη της εργασίας είναι ο καθορισμός της λειτουργικότητας (προγραμματισμός της

πλατφόρμας) με τη βοήθεια γνωστών γλωσσών προγραμματισμού (C, C++, Java) και όχι με προσαρμοσμένες (proprietary) γλώσσες.

### **Θέμα 11<sup>ο</sup>:**

#### **Αυτόματη μετατροπή από ER σε ανώτερες από τις σχεσιακές λογικές δομές**

Κατά τη σχεδίαση των Πληροφοριακών Συστημάτων, οι εννοιολογικές δομές των απαιτούμενων δεδομένων συνηθίζεται να αναπαρίστανται με Entity Relationship (ER) diagrams. Κατά την υλοποίηση των συστημάτων και όταν τα διαθέσιμα συστήματα διαχείρισης δεδομένων είναι σχεσιακά, οι εννοιολογικές δομές μετασχηματίζονται σε σχεσιακές λογικές δομές. Οι σχεσιακές λογικές δομές γίνονται αντιληπτές (από τους πληροφορικούς) ως πίνακες. Όμως, υπάρχουν αρκετές προσπάθειες για συστήματα διαχείρισης δεδομένων που υποστηρίζουν περισσότερο σύνθετα λογικά δεδομένα. Μία τέτοια προσπάθεια είναι η "Conceptual Universal Database Language" (σύντομα CUDL). Η CUDL μας επιτρέπει να εκλαμβάνουμε τα δεδομένα ως πίνακες των οποίων όμως τα κελιά μπορεί να δέχονται απλές τιμές ή πολλαπλές τιμές ή να είναι σύνθετα (να μπορούν να διασπασθούν σε απλά υποπεδία - επιμέρους απλά συνθετικά) ή να δέχονται πολλαπλές τιμές και ταυτόχρονα να είναι σύνθετα (το τελευταίο είναι πίνακας που φωλιάζει στη θέση απλού πεδίου). Αυτός ο τρόπος αντίληψης των δεδομένων που υποστηρίζει η CUDL ονομάζεται CUDL Abstraction Level (CAL). Σκοπός της παρούσας εργασίας είναι να κατασκευασθεί εργαλείο που θα μετασχηματίζει αυτόματα από ER diagrams σε CAL structures. Πρακτικά αυτό σημαίνει ότι το εργαλείο θα διαβάζει ER διαγράμματα (συγκεκριμένα τα ER διαγράμματα που γίνονται με το εργαλείο Dia και τα οποία είναι XML αρχεία) και θα παράγει CAL structures (ή ισοδύναμα τις data definition εντολές της CUDL).

Ένα καλό σημείο εκκίνησης για την παρούσα εργασία είναι το επόμενο άρθρο (A9):

[http://www.scieiv.com/publications/ijim/index.php?option=com\\_k2&view=item&task=download&d=12&Itemid=61](http://www.scieiv.com/publications/ijim/index.php?option=com_k2&view=item&task=download&d=12&Itemid=61)

Συμπληρωματικά άρθρα για την παρούσα εργασία είναι τα άρθρα (B14, A6 και B17):

<http://www.cs.teiath.gr/nnk/papers/B14.pdf>

<http://dx.doi.org/10.1016/j.jss.2008.12.031>

<http://www.cs.teiath.gr/nnk/papers/B17.pdf>

### **Θέμα 12<sup>ο</sup>:**

#### **Σύστημα παρακολούθησης στόλου οχημάτων.**

Θα αναπτυχθεί ολοκληρωμένο σύστημα καταγραφής σημάτων από οχήματα. Το σύστημα θα παρέχει στους τελικούς χρήστες του (πελάτες) τη δυνατότητα: (α) παρακολούθησης της παρούσας θέσης του κάθε αυτοκινήτου που ανήκει στον στόλο των οχημάτων του, (β) εμφάνισης της ακολουθίας του συνόλου ή των πιο πρόσφατων ληφθέντων μηνυμάτων και (γ) την προβολή ιστορικού θέσεων (μετακινήσεως) του κάθε οχήματος. Συμπληρωματική στην (α) δυνατότητα θα είναι η προβολή, για ένα κάθε φορά επιλεγμένο όχημα εκ του στόλου, της εκτελούμενης πορείας επάνω σε ψηφιακό χάρτη (τη δυνατότητα αυτή θα την ονομάζουμε "live"). Οι τεχνολογίες που θα συνδυάζει το σύστημα θα είναι: καταγραφικός εξοπλισμός αυτοκινήτων, GPRS (για την επικοινωνία του καταγραφικού εξοπλισμού αυτοκινήτου με τον Server καταγραφής), GPS (General Positioning System), Γεωγραφικές βάσεις δεδομένων (GIS database, όπως είναι η POST GIS), ψηφιακούς χάρτες (όπως είναι το Google Map ή το Open Street Map). Η αρχιτεκτονική του συστήματος θα βασίζεται σε δύο Servers. Ο πρώτος Server θα αποθηκεύει τα μηνύματα με τις συντεταγμένες των οχημάτων σε κλασσική σχεσιακή βάση δεδομένων, θα ανακτά (από άλλη Γεωγραφική βάση δεδομένων) την οδό και την περιοχή και με

τα τελευταία στοιχεία θα εμπλουτίζει τις καταγραφές που γίνονται στην κλασσική σχεσιακή βάση δεδομένων. Ο δεύτερος Server θα παρέχει υπό μορφή Web Services και με γραφικές διεπαφές (GUI) τις δυνατότητες (α), (β), (γ) και "live" που αναφέρθηκαν παραπάνω. Στην ανάπτυξη θα χρησιμοποιηθούν λογισμικά και συστήματα ανοιχτού κώδικα (ενδεικτικά αναφέρουμε: Linux, Web Services / Restfull, Tomcat/Glassfish, Post GIS, Postgress, Open Street Map, Java).

### **Θέμα 13<sup>ο</sup>:**

#### **Σύστημα «Ηλεκτρονικής συνταγογράφησης εξωτερικών ασθενών» και "Γενόσημα προϊόντα"**

Η ηλεκτρονική συνταγογράφηση επιτρέπει σε έναν γιατρό, νοσοκόμο, ή βοηθό γιατρού να διαβιβάσει ή να ανανεώσει ηλεκτρονικά μια συνταγή σε ένα φαρμακείο. Έχει ως στόχο να μειώσει τους κινδύνους που συνδέονται με την συμβατική συγγραφή συνταγής. Βασικοί περιορισμοί της ηλεκτρονικής συνταγογράφησης:

1. Η χρήση της εφαρμογής γίνεται μόνο από εξουσιοδοτημένους χρήστες
2. Εάν ο ασφαλισμένος δεν έχει ΑΜΚΑ είναι αδύνατη η καταχώριση ηλεκτρονικής συνταγής άρα και η εκτέλεση συνταγών φαρμάκων.
3. Το σύστημα δεν επιτρέπει την καταχώριση συνταγής εάν δεν έχουν καταγραφεί από τον Ιατρό σε μία συνταγή τα στοιχεία του ασθενή, η διάγνωση, η θεραπεία (δηλαδή τα φάρμακα που θα πρέπει να πάρει ασθενής) και η δοσολογία.
4. Το σύστημα καταγράφει αυτόματα την ημερομηνία έναρξης μιας συνταγής και υπολογίζει την ημερομηνία λήξης της αυτόματα.
5. Το σύστημα ακυρώνει αυτόματα μία συνταγή αν παρέλθει ο χρόνος λήξης της.
6. Το σύστημα δεν επιτρέπει στον Φαρμακοποιό να δώσει οποιοδήποτε άλλο φάρμακο πλην αυτών που καταχώρησε ο Ιατρός.

Για την καταχώριση της διάγνωσης του Ιατρού το σύστημα δίνει τη δυνατότητα η καταχώριση να γίνει είτε μέσω κωδικοποίησης ICD-10 (International Classification of Diseases - 10) είτε με ελεύθερο κείμενο. Το σύστημα θα πρέπει να σχεδιαστεί έτσι ώστε να λαμβάνει υπ' όψιν τις επιπρόσθετες πληροφορίες που απαιτούνται (σύμφωνα με τον Ν. 4052/2012) για τα Γενόσημα φάρμακα. Οι πληροφορίες αυτές, είναι οι εξής:

1. Η συνταγογράφηση πρέπει να γίνεται με βάση τη δραστική ουσία του φαρμάκου.
2. Το σύστημα παρέχει την δυνατότητα στον φαρμακοποιό να μπορεί να χορηγεί το φθηνότερο φαρμακευτικό προϊόν (γενόσημο) όμοιας δραστικής περιεκτικότητας και φαρμακοτεχνικής μορφής με αυτό που έχει συνταγογραφηθεί από τον Ιατρό.

Σχετικές Πληροφορίες παρέχονται στα επόμενα:

- ΦΕΚ 545/τ.Β/1.3.2012 (είναι διαθέσιμο από το [www.et.gr](http://www.et.gr)),
- Νόμος 4052, ΦΕΚ 41/τ.Α/1.3.2012, με έμφαση στα άρθρα 16 μέχρι 26 (είναι διαθέσιμο από το [www.et.gr](http://www.et.gr)),
- [http://www.eof.gr/web/guest/home?p\\_p\\_id=62\\_INSTANCE\\_Cu81&p\\_p\\_lifecycle=0&p\\_p\\_state=maximized&p\\_p\\_mode=view&p\\_p\\_col\\_id=column-2&p\\_p\\_col\\_pos=1&p\\_p\\_col\\_count=12&\\_62\\_INSTANCE\\_Cu81\\_struts\\_action=%2Fjournal\\_articles%2](http://www.eof.gr/web/guest/home?p_p_id=62_INSTANCE_Cu81&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&p_p_col_id=column-2&p_p_col_pos=1&p_p_col_count=12&_62_INSTANCE_Cu81_struts_action=%2Fjournal_articles%2Fview&_62_INSTANCE_Cu81_groupId=12225&_62_INSTANCE_Cu81_articleId=88476&_62_INSTANCE_Cu81_version=1.0)
- [http://www.eof.gr/web/guest/home?p\\_p\\_id=62\\_INSTANCE\\_Cu81&p\\_p\\_lifecycle=0&p\\_p\\_state=maximized&p\\_p\\_mode=view&p\\_p\\_col\\_id=column-2&p\\_p\\_col\\_pos=1&p\\_p\\_col\\_count=12&\\_62\\_INSTANCE\\_Cu81\\_struts\\_action=%2Fjournal\\_articles%2](http://www.eof.gr/web/guest/home?p_p_id=62_INSTANCE_Cu81&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&p_p_col_id=column-2&p_p_col_pos=1&p_p_col_count=12&_62_INSTANCE_Cu81_struts_action=%2Fjournal_articles%2)

Fview&\_62\_INSTANCE\_Cu81\_groupId=12225&\_62\_INSTANCE\_Cu81\_articleId=104086&\_62\_INSTANCE\_Cu81\_version=1.0

#### **Θέμα 14<sup>ο</sup>:**

#### **Διαδικτυακή εφαρμογή διαχείρισης αποθήκης νοσοκομειακού εργαστηρίου**

Η εφαρμογή θα διαχειρίζεται τα ΑΝΤΙΔΡΑΣΤΗΡΙΑ, τα στοιχεία ελέγχου (CONTROLS), τα στοιχεία διαμόρφωσης (CALIBRATORS), τα BUFFERS και τα ΑΝΑΛΩΣΙΜΑ ενός νοσοκομειακού εργαστηρίου. Η ανάπτυξη θα γίνει σε PHP/javascript/mysql ή σε ASP/javascript/access και θα παρέχει πλήρη ασφάλεια σε εξουσιοδότηση πρόσβασης (χρήστες), σε μη άρνηση ενέργειας (καταγραφή ενεργειών χρηστών – log) και σε κακόβουλες προσπάθειες (με SSL). Η σχεδίαση και υλοποίηση της εφαρμογής θα πρέπει να έχει επαγγελματικό χαρακτήρα.

Αιγάλεω 12-3-2012

Δρ. Νικήτας Ν. Καρανικόλας