



**ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ**

**Υλοποίηση Συστήματος Σύγκρισης Σχεδίων και  
Επιλογής Βέλτιστης Λύσης (CVR) Για Την  
Αρχιτεκτονική Εσωτερικών Χώρων.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**του**

**ΒΟΥΚΑΤΟΝΑΣΙΟΥ ΑΛΕΞΑΝΔΡΟΥ**

**Επιβλέπων :** Μαλάμος Γ. Αθανάσιος  
Αναπληρωτής Καθηγητής

Ηράκλειο, Ιούνιος 2010

Η σελίδα αυτή είναι σκόπιμα κενή



**ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

ΤΕΧΝΟΛΟΓΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ  
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΠΟΛΥΜΕΣΩΝ

**Υλοποίηση Συστήματος Σύγκρισης Σχεδίων και  
Επιλογής Βέλτιστης Λύσης (CBR) Για Την  
Αρχιτεκτονική Εσωτερικών Χώρων.**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

του

**ΒΟΥΚΑΤΟΝΑΣΙΟΥ ΑΛΕΞΑΝΔΡΟΥ**

**Επιβλέπων :** Μαλάμος Γ. Αθανάσιος  
Αναπληρωτής Καθηγητής

Ηράκλειο, Ιούνιος 2010

Η σελίδα αυτή είναι σκόπιμα κενή

*(Υπογραφή)*

.....

**ΒΟΥΚΑΤΟΝΑΣΙΟΣ ΑΛΕΞΑΝΔΡΟΣ**

ΠΤΥΧΙΟΥΧΟΣ ΤΕΧΝΟΛΟΓΟΣ ΜΗΧΑΝΙΚΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΠΟΛΥΜΕΣΩΝ

© 2010 – All rights reserved

## Περίληψη

Η εργασία αυτή ασχολείται με τον τομέα της τεχνητής νοημοσύνης και ειδικότερα με το κομμάτι της συλλογιστικής βασισμένης σε περιπτώσεις (CBR – Case Based Reasoning). Ο τομέας αυτός έχει τραβήξει το ενδιαφέρον της ερευνητικής κοινότητας εξαιτίας της ομοιότητάς του με τον ανθρώπινο τρόπο σκέψης. Το CBR προσπαθεί να αποφασίσει για μια πράξη, υπολογίζοντας μνήμες του παρελθόντος, από σχετικές πράξεις και καταστάσεις στις οποίες βρέθηκε. Επιπλέον, προσφέρει τις κατάλληλες διαδικασίες για να ενισχύσει την αρχική δοθείσα επιλογή, με την εφαρμογή επαναληπτικής βελτίωσης των λύσεων, προκειμένου να αντιμετωπίσει τις απαιτήσεις των χρηστών. Εδώ, εφαρμόστηκε το CBR πάνω στο θέμα της διακόσμησης εσωτερικών χώρων. Σχεδιάστηκε και υλοποιήθηκε ένα σύστημα με πολύ ικανοποιητικά αποτελέσματα.

**Λέξεις Κλειδιά:** << Τεχνητή Νοημοσύνη, Συλλογιστική Βάσει Περιπτώσεων, CBR αλγόριθμος, Περιγραφή Διακόσμησης Εσωτερικών Χώρων, XML >>

Η σελίδα αυτή είναι σκόπιμα κενή

## Abstract

Our work deals with the field of artificial intelligence and more specifically with the case based reasoning domain (CBR - Case Based Reasoning). This area is attractive to the research community since it resembles the human way of thinking. CBR aims to provide decision support about a transaction by retracing memories of the past in the corresponding acts and cases. Moreover, offers the appropriate procedures to enhance the initial choice of actions by applying recursive refinement of solutions in order to meet the initial user requirements. In our thesis, CBR is applied in the interior decoration domain. A system has been designed and tested with very satisfactory results.

**Keywords:** Artificial Intelligence, Case Based Reasoning, CBR algorithm, Interior Design Room Description, XML

Η σελίδα αυτή είναι σκόπιμα κενή



## Περιεχόμενα

Περίληψη.....	5
Abstract .....	7
Κεφάλαιο 1.....	11
1.1 Αντικείμενο τις πτυχιακής εργασίας .....	12
1.2 Οργάνωση Κειμένου .....	12
Κεφάλαιο 2.....	14
2.1 Γενικά.....	15
2.3 Συλλογιστική μέσα στις συλλογιστικές συγκρίσεις του CBR.....	18
2.3.1 Εργασία αναπροσαρμογής.....	18
2.3.2 Κατάρτιση δεδομένων Προσαρμογής για τη μάθηση .....	20
2.3.3 Προσαρμοστικές Δράσεις.....	22
2.4 Εκμάθηση περιπτώσεων στο CBR .....	24
2.5 Σύγκριση Case-Based Reasoning με άλλες μεθόδους.....	25
2.6 Κριτική .....	26
2.7 Ιστορικά Στοιχεία .....	26
2.8 Γνωστά CBR Συστήματα .....	27
2.9 Ταίριασμα βάσει προτύπου (Pattern Matching) .....	33
2.10 Βασικές μορφές προτύπων(primitive patterns) .....	35
2.11 Πρότυπα δέντρων (Tree Patterns) .....	36
Κεφάλαιο 3.....	38
3.1 Γενικά περί του αλγορίθμου επίλυσης του προβλήματος διακόσμησης εσωτερικών χώρων. ....	39
3.2 Δομή XML εγγράφων περιγραφής Δωματίου.....	43
3.2.1 Δομή XML pattern εγγράφου.....	43
Κεφάλαιο 4.....	89
4.1 Αναλυτική παρουσίαση κλάσεων και μεθόδων αλγορίθμου.....	90
Κεφάλαιο 5.....	96
5.1 Παράθυρο ρυθμίσεων.....	97
5.2 Παράθυρο Ρύθμισης Βάρους Κόμβων .....	99

5.3 Παράθυρο εφαρμογής .....	101
Κεφάλαιο 6.....	107
6.1 Εγκατάσταση Προαπαιτούμενων Εφαρμογών .....	108
Επίλογος .....	113
ΠΑΡΑΡΤΗΜΑ .....	114
1. Javados Εφαρμογής.....	115
2. XML pattern έγγραφο που χρησιμοποιείται για να περιγραφεί ένα δωμάτιο .....	138
3. Τελικός Πίνακας XML Διαδρομών.....	148
Βιβλιογραφία.....	155

## **Κεφάλαιο 1**

### **Εισαγωγή**

## 1.1 Αντικείμενο τις πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής είναι η ανάπτυξη ενός συστήματος διακόσμησης εσωτερικών χώρων, με τη χρήση της συλλογιστικής βασισμένης σε περιπτώσεις (CBR – Case Based Reasoning).

Ο αλγόριθμος συλλογιστικής βάση περιπτώσεων ανήκει στην κατηγορία των αλγορίθμων τεχνητής νοημοσύνης και χαρακτηριστικό του είναι ότι προσομοιάζει τον ανθρώπινο τρόπο σκέψης και λήψης αποφάσεων, αφού χρειάζεται προηγούμενες γνώσεις για να αποφασίσει. Οι προηγούμενες αυτές γνώσεις αποθηκεύονται και ανακτώνται μέσω μιας βάσης (βάση γνώσης του αλγορίθμου).

Έγινε χρήση ενός τέτοιου είδους τεχνικής επειδή θέλαμε αυτός που παίρνει την τελική απόφαση να είναι το σύστημα εν τέλει και συγκεκριμένα χρησιμοποιήσαμε συλλογιστική βάση περιπτώσεων λόγω του ότι η τεχνική αυτή είναι αρκετά γρήγορη υπολογιστικά, σε σχέση με άλλες τεχνικές του είδους της.

Η λύση που προτείνεται θα βοηθήσει κάποιον ο οποίος θέλει να διακοσμήσει το χώρο του, του χωρίς να χρειαστεί να καταφύγει σε διακοσμητή η να ξοδέψει χρήματα. Σκοπός είναι να προσφερθεί μια ευκολία στη διακόσμηση στον καθένα ακόμα και αν δεν την έχει ανάγκη, εφόσον δεν έχει να χάσει κάτι όπως χρόνο και χρήμα.

## 1.2 Οργάνωση Κειμένου

Στην πορεία του παρόντος κειμένου θα δούμε αναλυτικά τη λύση του προβλήματος που αντιμετωπίσαμε, για να κατανοήσουμε πως καταλήξαμε εκεί.

Στο κεφάλαιο 2 θα προσεγγίσουμε γενικά το τι είναι ένας αλγόριθμος συλλογιστικής βασισμένος σε περιπτώσεις θα μιλήσουμε για ιστορικά στοιχεία και για κάποια ήδη υπάρχοντα και πολύ γνωστά άλλα συστήματα που λειτουργούν με την ίδια λογική.

Στη συνέχεια στο κεφάλαιο 3 θα γίνει λόγος για τον συγκεκριμένο αλγόριθμο στην πρώτη ενότητα και στη συνέχεια θα γίνει μια εκτενής παρουσίαση του xml εγγράφου που χρησιμοποιείται.

Εν συνεχεία, στο επόμενο κεφάλαιο δηλαδή το 4<sup>ο</sup>, έχουμε την τεχνική περιγραφή του αλγορίθμου, παρουσίαση των μεθόδων που τον απαρτίζουν και επεξήγηση αυτών όπως και διαγράμματα που κάνουν εμφανείς τις μεταξύ τους σχέσεις.

Στο 5<sup>ο</sup> κεφάλαιο γίνεται παρουσίαση της διεπαφής ώστε να γίνει κατανοητός ο τρόπος χρήσης του οπτικού περιβάλλοντός της από το χρήστη.

Το κεφάλαιο 6 μιλά για το πώς κάποιος μπορεί να εγκαταστήσει στον υπολογιστή του την εφαρμογή και να κάνει χρήση αυτής εύκολα και γρήγορα χωρίς προβλήματα.

Τέλος είναι ο επίλογος όπου αναλύεται το τι έχει γίνει έως και σήμερα αλλά και πως μπορεί να χρησιμοποιηθεί ο αλγόριθμος αυτός στο μέλλον και ακόμα και να αναπτυχθεί περαιτέρω.

## **Κεφάλαιο 2**

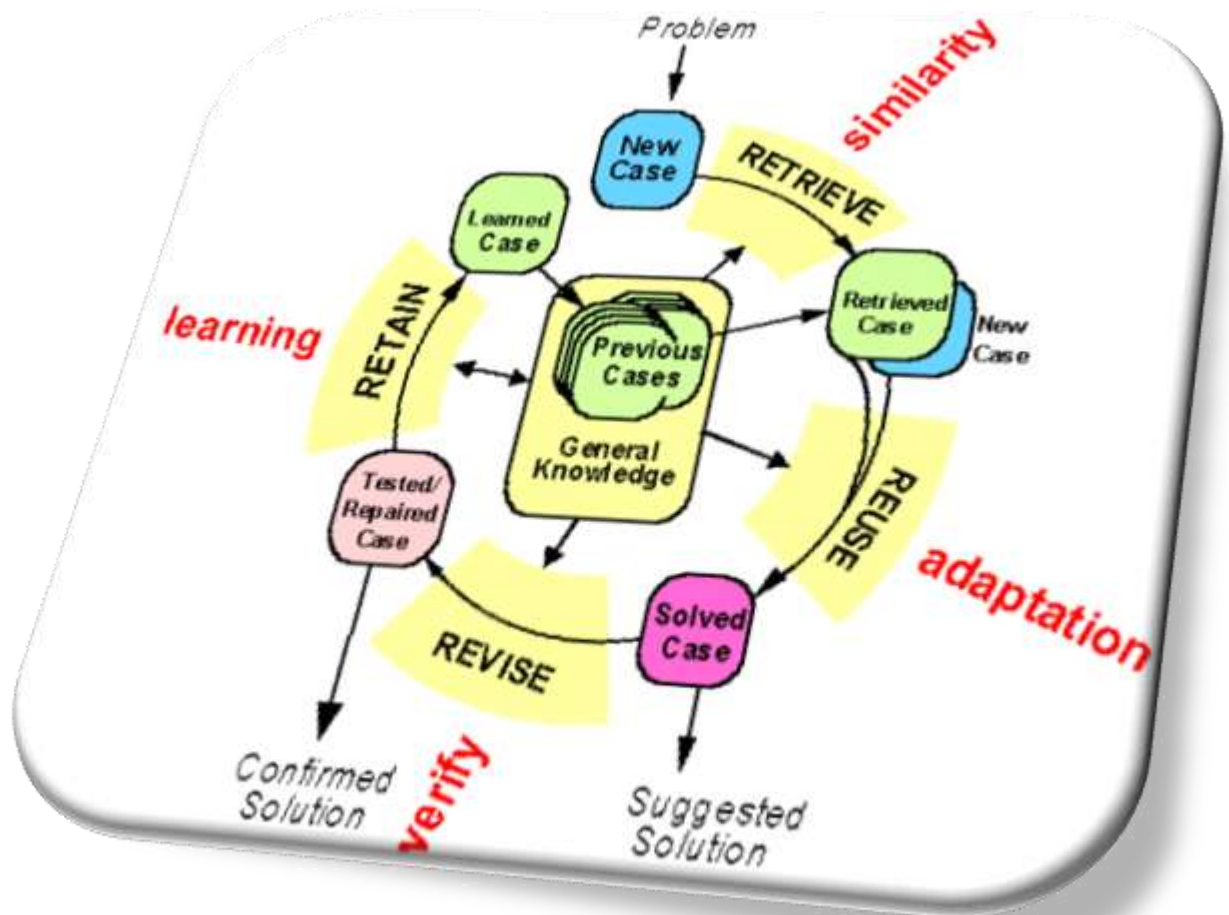
### **Προσέγγιση ενός Case-Based Reasoning συστήματος.**

## 2.1 Γενικά

Με τον όρο Case-Based Reasoning (CBR), υπό την ευρεία έννοια, εννοούμε τη διαδικασία επίλυσης των νέων προβλημάτων που εμπνέεται από τις λύσεις του παρελθόντος παρόμοιων προβλημάτων. Ένας μηχανικός αυτοκινήτων που διορθώνει ένα κινητήρα θυμάται κάποια άλλα αυτοκίνητα που παρουσίασαν παρόμοια συμπτώματα επομένως χρησιμοποιεί τη μέθοδο της απόφασης με βάση την περίπτωση. Το ίδιο και ένας δικηγόρος ο οποίος υποστηρίζει ένα συγκεκριμένο αποτέλεσμα, σε μια μελέτη, που βασίζεται σε νομικά προηγούμενα ή δικαστή που δημιουργεί νομολογία χρησιμοποιεί (CBR), υπό ευρεία έννοια, είναι η διαδικασία επίλυσης των νέων προβλημάτων που εμπνέεται από τις λύσεις του παρελθόντος παρόμοιων προβλημάτων. Έτσι, επίσης, μια μηχανικός η οποία ασχολείται με την αντιγραφή στοιχείων της φύσης (εξασκεί την βιομίμηση), είναι η αντιμετώπιση της φύσης ως μια βάση δεδομένων, με λύσεις σε προβλήματα. Η μέθοδος απόφασης με βάση την περίπτωση είναι ένα σημαντικό είδος λήψης αποφάσεων δημιουργώντας αναλογίες.

Έχει υποστηριχθεί ότι η απόφαση με βάση την περίπτωση δεν είναι μόνο μια ισχυρή μέθοδος για τη συλλογιστική του υπολογιστή, αλλά και μια διάχυτη συμπεριφορά στην καθημερινή ανθρώπινη επίλυση προβλημάτων. Ή πιο συγκεκριμένα, ότι όλη η συλλογιστική αυτή στηρίζεται σε προηγούμενες περιπτώσεις, προσωπικές εμπειρίες. Η άποψη αυτή συνδέεται με την θεωρία του πρωτοτύπου, που είναι το πιο βαθιά διερευνηθέν αντικείμενο στη γνωσιακή επιστήμη.

## 2.2 Διαδικασίες Case-Based Reasoning



Εικόνα 2.1. Οι βασικοί κύκλοι ενός Case based reasoning συστήματος.

Πηγή: <http://www.peerscience.com>.



Η συλλογιστική βασισμένη σε περιπτώσεις έχει επισημοποιηθεί για τους σκοπούς της συλλογιστικής του υπολογιστή ως μια διαδικασία τεσσάρων βημάτων οι οποίες εξηγούνται παρακάτω με τη χρήση ενός παραδείγματος:

### 1. Ανάκτηση:

Δεδομένου ενός προβλήματος, ανακτώνται οι περιπτώσεις από τη μνήμη που είναι συναφείς με την επίλυσή του. Μια περίπτωση αποτελείται από ένα πρόβλημα, τη λύση του, και συνήθως, σχόλια για το πώς η λύση αυτή προέκυψε. Για παράδειγμα, ας υποθέσουμε ότι ο Fred θέλει να προετοιμάσει τηγανίτες βατόμουρου. Όντας αρχάριος ως μάγειρας, η σημαντικότερη εμπειρία που μπορεί να θυμηθεί είναι εκείνη κατά την οποία έκανε με επιτυχία απλές τηγανίτες. Η διαδικασία που ακολούθησε για την κατασκευή της απλής τηγανίτας, μαζί με τις αιτιολογήσεις για αποφάσεις που λαμβάνονται στην πορεία, αποτελεί ανακτηθείσα περίπτωση του Fred.

### 2. Επαναχρησιμοποίηση:

Χρήση της λύσης από προηγούμενη υπόθεση-περίπτωση στο νέο πρόβλημα. Αυτό μπορεί να συνεπάγεται την προσαρμογή της λύσης, όπως απαιτείται για να ταιριάζει στη νέα κατάσταση. Στο παράδειγμα με τις τηγανίτες, ο Fred πρέπει να προσαρμόσει την ανακτημένη λύση του και να περιλάβει και την προσθήκη των βατόμουρων.

### 3. Αναθεώρηση:

Έχοντας χρησιμοποιήσει και προσαρμόσει την προηγούμενη λύση στην κατάσταση-στόχο, δοκιμάζουμε τώρα τη νέα λύση στον πραγματικό κόσμο (ή μια προσομοίωση) και, εφόσον κριθεί αναγκαίο, αυτή αναθεωρείται. Ας υποθέσουμε ότι Fred προσάρμοσε τη λύση του με την προσθήκη βατόμουρου στο μίγμα του. Μετά την ανάμιξη, ανακαλύπτει ότι το μίγμα έχει μετατραπεί σε μπλε, το οποίο είναι ένα ανεπιθύμητο αποτέλεσμα. Αυτό υποδηλώνει την ακόλουθη αναθεώρηση: καθυστέρηση της προσθήκης των βατόμουρων έως ότου το μίγμα έχει απλωθεί στο ταψί.

#### 4. Διατήρηση:

Αμέσως μετά την προσαρμογή της λύσης στο πρόβλημα του, το αποτέλεσμα αποθηκεύεται σαν εμπειρία, ως μια νέα περίπτωση στην μνήμη. Ο Φρέντ, κατά συνέπεια, καταγράφει την πρωτόγνωρη διαδικασία για την κατασκευή τηγανιτών βατόμουρου, με αποτέλεσμα να εμπλουτίζει το σετ των αποθηκευμένων εμπειριών και να προετοιμάζεται καλύτερα για μια μελλοντική περίπτωση λήψης απόφασης για την κατασκευή τηγανίτας.

### 2.3 Συλλογιστική μέσα στις συλλογιστικές συγκρίσεις του CBR.

#### 2.3.1 Εργασία αναπροσαρμογής

Το CBR ανακτά περιπτώσεις που αντιστοιχούν σε παρόμοια προβλήματα από τη βάση περιπτώσεων-γνώσης της. Το βήμα της προσαρμογής πρέπει να αναγνωρίσει τις διαφορές μεταξύ των νέων και των ανακτηθέντων προβλημάτων, και να τελειοποιήσουν τις ανακτημένες λύσεις ώστε να αντικατοπτρίζουν τις διαφορές, ανάλογα με την περίπτωση. Για παράδειγμα, σε μια τιμολόγηση ενός σπιτιού το CBR σύστημα, ανακτά ένα σπίτι με λιγότερα υπνοδωμάτια θα πρέπει να έχει στην ανάκτηση των τιμών μια αύξηση ώστε να αντικατοπτρίζει τα επιπλέον υπνοδωμάτια στο νέο σπίτι. Ο Kolodner διακρίνει τρεις τύπους προσαρμογής:

- **Αντικατάσταση:** αντικαθιστά τιμές στην ανακτηθείσα λύση με νέες αξίες κατάλληλες για το νέο πρόβλημα (π.χ. αλλαγή μιας σύστασης των τιμών των κατοικιών),

- Ο **μετασχηματισμός** αλλοιώνει την ανακτημένη λύση με την προσθήκη, διαγραφεί ή αντικαθιστά τμήματα της ανακτηθείσας λύσης ώστε να ταιριάζει στο νέο πρόβλημα (π.χ. μεταβολή βημάτων σε ένα σχέδιο) και
- **Ειδικές μέθοδοι** εφαρμόζονται εξειδικευμένες γνώσεις heuristic για την επισκευή της ανακτηθείσας λύσης, ή επανάληψη της μεθόδου που χρησιμοποιήθηκε για την ανάκτηση της λύσης για το νέο πρόβλημα.

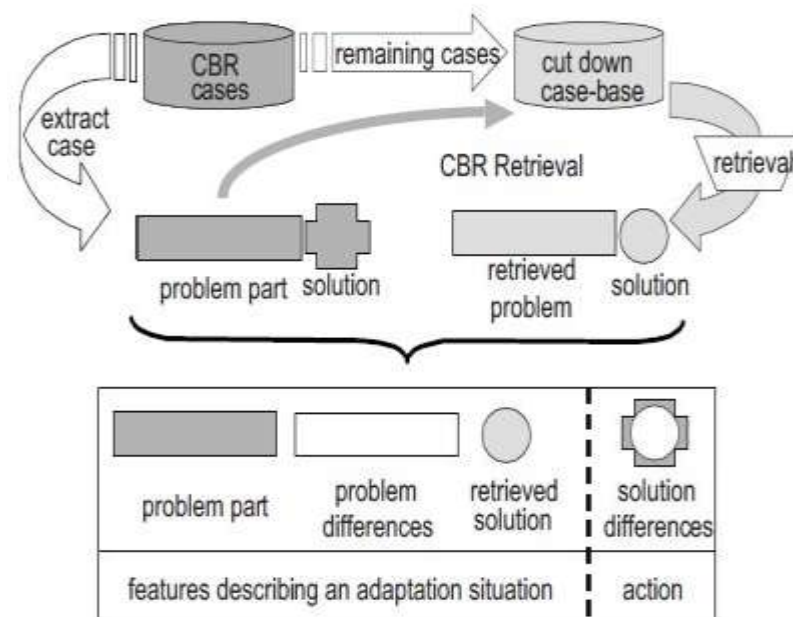
Για την προσαρμογή, ο στόχος είναι να αναγνωρίζεται από το σύστημα πότε η προσαρμογή θα πρέπει να εφαρμόζονται επειδή τα νέα και αντληθέντα προβλήματα είναι αρκετά διαφορετικά με κάποιο σχετικό τρόπο, και να εκτελέσει κάποια αλλαγή (ες) για την ανάκτηση της λύσης. Μια προσαρμογή μπορεί να θεωρηθεί ως μια *κατάσταση/δράση* ζεύγος. Η *κατάσταση* περιλαμβάνει τις διαφορές μεταξύ των νέων και των αντληθέντων προβλημάτων. Η *δράση* αιχμαλωτίζει την ενημέρωση για την ανακτημένη λύση: νέες τιμές για την επαναχρησιμοποιημένη λύση (υποκατάσταση), συνιστώσες λύσης που πρέπει να προστεθούν, διαγραφούν ή να αλλάξει (μετασχηματισμός), ή περισσότερο εξειδικευμένες γνώσεις για να επιτευχθεί η ενημέρωση (ειδικά).

Το παρόν έγγραφο εξετάζει την προσαρμογή από την υποκατάσταση, αλλά παρόμοιες τεχνικές ενδέχεται να είναι κατάλληλες για τις πιο πολύπλοκες μεθόδους μετατροπής, όπου τη δομή της λύσης είναι η μεταβολή με οποιοδήποτε τρόπο. Οι ειδικές μέθοδοι είναι αυτές που χρειάζεται η πλέον έντονη γνώση για την προσαρμογή επειδή μπορεί να είναι πιο δύσκολο να συλληφθεί, ιδιαίτερα για την αχνή γνώση των μεθοδολογίας, δεδομένου ότι η ενημέρωση που πρέπει να εκτελεστεί θα είναι πιο πολύπλοκη και πρέπει να εκπροσωπείται κατά κάποιον τρόπο, για το τμήμα δράσης.

## 2.3.2 Κατάρτιση δεδομένων Προσαρμογής για τη μάθηση

Η βάση γνώσεων είναι μια χρήσιμη πηγή γνώσης για τον τομέα επίλυσης των προβλημάτων. Περιλαμβάνει μια σειρά από υποθέσεις που περιλαμβάνουν το πρόβλημα, μαζί με τη λύση του. Από τη φύση της, η βάση γνώσης θεωρείται ως ένα αντιπροσωπευτικό δείγμα της επίλυσης προβλημάτων στον τομέα, και έτσι τα δεδομένα κατάρτισης της προσαρμογής, που αποκτήθηκαν από την πηγή αυτή, είναι πιθανόν να δημιουργήθηκαν αντιπροσωπευτικά της γνώσης προσαρμογής.

Έχουμε υιοθετήσει μια προσέγγιση άφησε-ένα-εκτός για τη δημιουργία παραδειγμάτων προσαρμογής, όπως φαίνεται στο πάνω μέρος της εικόνας 2.3.2.1. Μία περίπτωση, που αποτελείται από ένα τμήμα προβλήματος και τη λύση του, αφαιρείται από την CBR βάση γνώσης, και οι υπόλοιπες περιπτώσεις χρησιμοποιούνται για το σχηματισμό μιας μικρότερης, κομμένης, εκδοχής της βάσης. Ως αποτέλεσμα, η εξαγόμενη περίπτωση και οι πιο παρόμοιες περιπτώσεις, που ανακτώνται από την κομμένη βάση γνώσης, είναι διαθέσιμες να παρέχουν στοιχεία για να δημιουργήσουν το παράδειγμα της κατάρτισης προσαρμογής, όπως φαίνεται στο κάτω μέρος της εικόνας 2.3.2.1. Κάθε παράδειγμα προσαρμογής συλλαμβάνει την κατάσταση προσαρμογής που το εξαγόμενο πρόβλημα και οι ανακτημένες περιπτώσεις ανέκτησαν καθώς και τη δράση προσαρμογής που θα πρέπει να εφαρμοστεί. Η κατάσταση της προσαρμογής αποτελείται από τις τιμές παραμέτρων στο προβληματικό τμήμα της εξαγόμενης περίπτωσης, τις διαφορές (διαφοροποιήσεις) μεταξύ αυτών που έχουν αφαιρεθεί και άντλησε τις περιπτώσεις για κάθε ένα από αυτά τα προβληματικά χαρακτηριστικά και την ανακτημένη λύση. Η δράση προσαρμογής συλλαμβάνει την ενημερωμένη έκδοση που είναι απαραίτητο να αλλάξει την ανακτημένη λύση στη λύση της υπόθεσης που θα εξαχθεί.



**Εικόνα 2.3.2.1 Παράδειγμα Προσαρμογής (Adaptation) Assembling Adaptation Training Data from the Case-base**

Είναι σημαντικό να σημειωθεί ότι τα παραδείγματα της προσαρμογής είναι η κατάρτιση των δεδομένων στα οποία η δράσεις καταγράφονται μαζί με τις καταστάσεις. Οι προσαρμογές γενικεύονται αργότερα από όπου ο αλγόριθμος εφαρμόζεται πάνω. Δεδομένου ότι κάθε ένα από τους  $k$  πλησιέστερους γείτονες που ανακτήθηκαν για μια εξαχθείσα υπόθεση από την κομμένη βάση γνώσης, μπορεί να γίνει η βάση για ένα παράδειγμα προσαρμογής, το σύνολο των παραδειγμάτων αναπροσαρμογής μπορεί να αρκετές φορές μεγαλύτερο από την αρχική βάση γνώσης.

### 2.3.3 Προσαρμοστικές Δράσεις

Η δράση κρατάει την ενημερωμένη έκδοση που μετατρέπει την ανακτημένη λύση στη λύση στόχο για την περίπτωση που θα εξαχθεί. Για τις αριθμητικές λύσεις, το μέρος δράση του παραδείγματος της προσαρμογής είναι ο υπολογισμός, ίσως για να προστεθεί η αριθμητική διαφορά μεταξύ του στόχου και της ανακτηθείσας λύσης. Η προσαρμογή των συμβολικών λύσεων είναι πιο ενδιαφέρον δύσκολο για πολλούς λόγους.

- Τα χαρακτηριστικά αριθμητικά έχουν έναν φυσικό τρόπο για τον υπολογισμό ανομοιότητας και αυτό μπορεί να χρησιμοποιείται για να παράγει την ενημερωμένη έκδοση σε μια αριθμητική λύση. Οι συμβολικές ιδιότητες στη λύση μπορεί να μην έχουν ορισμένη ομοιότητα από το CBR μιας και ομοιότητα χρησιμοποιείται μόνο στα χαρακτηριστικά του προβληματικού μέρους.
- Η αριθμητική ανομοιότητα μπορεί να χρησιμοποιηθεί για να παραγάγει την ενημερωμένη λύση. Σε Αντίθεση μπορεί να μην υπάρξει μια συμβολική λύση που έχει τη σωστή ανομοιότητα στην ανακτημένη λύση όπως καθορίζεται από τη δράση προσαρμογής.
- Υπάρχει ένας φυσικός υπολογισμός που βασίζεται στην αριθμητική ανομοιότητα που μπορεί να εφαρμοστεί στην ανακτημένη αριθμητική λύση για να υπολογίσει μια ενημερωμένη λύση, αλλά δεν υπάρχει αντίστοιχη για τις συμβολικές ανομοιότητες.

Παρ'όλα αυτά, τα στοιχεία της προσαρμογής για αριθμητικές λύσεις παρέχουν πρόσβαση σε ό, τι είναι επιθυμητό για την προσαρμογή των συμβολικών λύσεων. Η εικόνα 2.3.2.2 δίνει παραδείγματα των αρχικών δράσεων προσαρμογής που προτείνουμε σε αυτήν την ενότητα. Για μια αριθμητική λύση όπως η ηλικία, όταν ο στόχος και ανακτημένες λύσεις είναι και οι δύο 40, τότε η δράση που σημειώνεται είναι να προστεθεί το μηδέν, αλλά όταν οι τιμές είναι 40 και 21, αντίστοιχα, στη συνέχεια, τότε το

-19 σημειώνεται ως η διόρθωση στην ανακτημένη λύση. Η πρώτη μέθοδος που αναπτύχθηκε εδώ για συμβολικές λύσεις χρησιμοποιεί την πρότυπη δυαδική ομοιότητα (ή διαφορετικά ίδιο) να ενημερώσει μια άτεχνη προσαρμογή (είτε η προσαρμογή είναι αναγκαία ή όχι). Για μια συμβολική λύση όπως το χρώμα, όταν ο στόχος είναι «κόκκινο», και «κόκκινο» ανακτάται, τότε η τιμή «ok» σημαίνει ότι δεν χρειάζεται προσαρμογή. Όταν είναι διαφορετικά, ας πούμε μια ότι στόχος είναι ένα «κόκκινο» και «κίτρινο» ανακτάται, η τιμή «¬ ok», δηλώνει ότι μια διαφορετική λύση θα πρέπει να χρησιμοποιηθεί, π.χ. από την επαναλαμβανόμενη ανάκτηση CBR. Στη δεύτερη προσέγγιση, η προσαρμογή της «¬ok» αντικαθίσταται από μια απλοϊκή multi-class προσαρμογή η οποία διορίζει τη λύση στόχο ως την ανακτημένη λύση. Για το παράδειγμα του χρώματος, αν ανακτηθεί κίτρινο δημιουργεί ένα παράδειγμα όπου η προσαρμογή η δράση προσαρμογής δείχνει ότι μια νέα κόκκινη λύση θα πρέπει να χρησιμοποιείται.

Numeric	Target	Retrieved	Adaptation	Action
Age	40	40	0	
	21	40	-19	

Symbolic	Target	Retrieved	Adaptation	Action
Colour	red	red	ok	
	red	yellow	¬ok	(binary)
	red	yellow	red	(multi-class)

Εικόνα 2.3.2.2 Translating Solution Differences into Adaptation Actions

### 2.4 Εκμάθηση περιπτώσεων στο CBR

Ένα πολύ σημαντικό χαρακτηριστικό της συλλογιστικής βάση περιπτώσεων είναι η σύνδεσή της στη μάθηση. Η κινητήρια δύναμη πίσω από συλλογιστική μέθοδο προέρχεται σε μεγάλο βαθμό από την κοινότητα μηχανικής μάθησης, και η συλλογιστική με βάση περιπτώσεις θεωρείται επίσης ένα υποπεδίο της μηχανικής μάθησης. Έτσι, η έννοια του CBR δεν προσδιορίζει μόνον μια συγκεκριμένη μέθοδο συλλογισμού, ανεξάρτητα από τον τρόπο απόκτησης των περιπτώσεων, προσδιορίζει επίσης ένα παράδειγμα μηχανικής μάθησης που δίνει τη δυνατότητα εκμάθησης από την ενημέρωση της βάσης περιπτώσεων (βάση γνώσης) μετά από τη λύση ενός προβλήματος. Η μάθηση στην CBR παρουσιάζεται ως ένα φυσικό υποπροϊόν της επίλυσης προβλημάτων. Όταν ένα πρόβλημα αντιμετωπισθεί με επιτυχία, η εμπειρία έχει διατηρηθεί για την επίλυση παρόμοιων προβλημάτων στο μέλλον. Όταν μια προσπάθεια για την επίλυση ενός προβλήματος αποτύχει, ο λόγος της αποτυχίας είναι αναγνωρισμένος και μνήμη ενημερώνεται για να αποφευχθεί το ίδιο λάθος στο μέλλον.

Το CBR ευνοεί τη μάθηση από την εμπειρία, δεδομένου ότι είναι συνήθως πιο εύκολο να μάθουν με τη διατήρηση της εμπειρίας της συγκεκριμένης επίλυσης προβλήματος από το να γενικεύουμε από αυτήν. Ακόμα, η αποτελεσματική μάθηση σε CBR απαιτεί μια καλά επεξεργασμένη σειρά μεθόδων, προκειμένου να εξαχθεί σχετική γνώση από την εμπειρία, για την ένταξη μιας υπόθεσης σε μια υπάρχουσα δομή γνώσης, και ταξινόμηση αυτής για να ταιριάζει αργότερα με άλλες παρόμοιες υποθέσεις.



## 2.5 Σύγκριση Case-Based Reasoning με άλλες μεθόδους

Με την πρώτη ματιά, η CBR μπορεί να φαίνεται παρόμοια με τους αλγόριθμους επαγωγής κανόνων της μηχανικής μάθησης. Όπως και σε έναν αλγόριθμο επαγωγής κανόνων, ο CBR ξεκινά με μια σειρά περιπτώσεων ή εκπαιδευτικών παραδειγμάτων. Αποτελεί γενικεύσεις των παραδειγμάτων αυτών, έστω και σιωπηρή, εντοπίζοντας ομοιότητες μεταξύ ανακτηθείσας υπόθεσης και προβλήματος στόχου.

Αν, για παράδειγμα, μια διαδικασία για τις απλές τηγανίτες αντιστοιχίζεται στις τηγανίτες βατόμουρου, θα ληφθεί απόφαση να χρησιμοποιούν το ίδιο βασικό μίγμα και τηγάνισμα, έτσι εμμέσως γενικεύεται το σύνολο των καταστάσεων σύμφωνα με τις οποίες οι μέθοδοι του μίγματος και του τηγανίσματος μπορούν να χρησιμοποιηθούν. Η βασική διαφορά, ωστόσο, μεταξύ του έμμεσης γενίκευσης του CBR και γενίκευσης σε κανόνες επαγωγής έγκειται στο πότε γίνεται η γενίκευση. Ένας αλγόριθμος κανόνων-επαγωγής αντλεί τις γενικεύσεις του από ένα σύνολο παραδειγμάτων εκπαίδευσης, πριν το πρόβλημα στόχος γίνει ακόμη γνωστό. Δηλαδή, εκτελεί πρόωμη γενίκευση.

Για παράδειγμα, σε έναν αλγόριθμο κανόνων-επαγωγής δοθούν συνταγές για απλές κρέπες, τηγανίτες ολλανδικού μήλου, και κρέπες μπανάνας ως παραδείγματα εκμάθησης, θα οδηγήσουν, κατά το χρόνο της εκμάθησης, ένα σύνολο γενικών κανόνων για τη λήψη όλων των τύπων τηγανίτας. Δεν θα δίνονταν μέχρι το στάδιο της δοκιμής, για παράδειγμα, το μαγείρεμα τηγανίτας βατόμουρου. Η δυσκολία για τον αλγόριθμο επαγωγής κανόνων είναι να προληφθούν οι διαφορετικές κατευθύνσεις στις οποίες θα πρέπει να προσπαθήσει να γενικεύσει τα παραδείγματα εκπαίδευσης. Αυτό έρχεται σε αντίθεση με την CBR, η οποία καθυστερεί (έμμεση) γενίκευση των περιπτώσεων της μέχρι το στάδιο της δοκιμής- μια στρατηγική τεμπέλικης γενίκευσης. Στο παράδειγμα με τις τηγανίτες, στο CBR έχει δοθεί ήδη το πρόβλημα στόχος του μαγειρέματος τηγανίτας βατόμουρου. Έτσι μπορεί να γενικεύσει τις περιπτώσεις του όπως ακριβώς χρειάζεται για να καλυφθεί αυτή η κατάσταση. Το CBR τείνει συνεπώς να είναι μια καλή προσέγγιση για πλούσιους σε χαρακτηριστικά, πολύπλοκους τομείς στους οποίους υπάρχουν μυριάδες τρόποι να γενίκευσης μιας υπόθεσης.

### 2.6 Κριτική

Οι επικριτές του CBR υποστηρίζουν ότι είναι μια προσέγγιση που δέχεται ανεπίσημα στοιχεία, ως βασική αρχή λειτουργίας του. Χωρίς στατιστικά στοιχεία που αφορούν την υποστηρικτική και αφανή του γενίκευση, δεν υπάρχει καμία εγγύηση ότι η γενίκευση είναι σωστή. Ωστόσο, όλη η επαγωγική λογική, όπου τα στοιχεία είναι πολύ σπάνια για τη στατιστική σημασία γίνεται εγγενώς με βάση ανεπίσημα στοιχεία.

### 2.7 Ιστορικά Στοιχεία

Ο αλγόριθμος CBR έχει τις ρίζες του στις εργασίες του Roger Schank και των μαθητών του στο Πανεπιστήμιο Yale στις αρχές του 1980. Το μοντέλο Schank της δυναμικής μνήμης ήταν η βάση για την ταχύτητα στα συστήματα CBR: Cyrus Janet Kolodner και Michael Lebowitz.

Άλλα σχολεία του CBR και συναφή πεδία αναδύθηκαν στη δεκαετία του 1980, και διερεύνησαν θέματα όπως CBR σε νομική επιχειρηματολογία, που βασίζεται σε αιτολόγηση βάσει μνήμης (ένας τρόπος συλλογισμού με χρήση παραδειγμάτων που προέρχονται μαζικά από πολλές παράλληλα συνδεδεμένες μηχανές), καθώς και συνδυασμοί των CBR με άλλες μεθόδους συλλογιστικής. Στη δεκαετία του 1990, το ενδιαφέρον για CBR μεγάλωσε στη διεθνή κοινότητα, όπως αποδεικνύεται από τη δημιουργία μιας διεθνούς διάσκεψης σχετικά με το Case-Based Reasoning το 1995, καθώς και από ευρωπαϊκά, γερμανικά, βρετανικά, ισπανικά, ιταλικά, και άλλα εργαστήρια CBR.

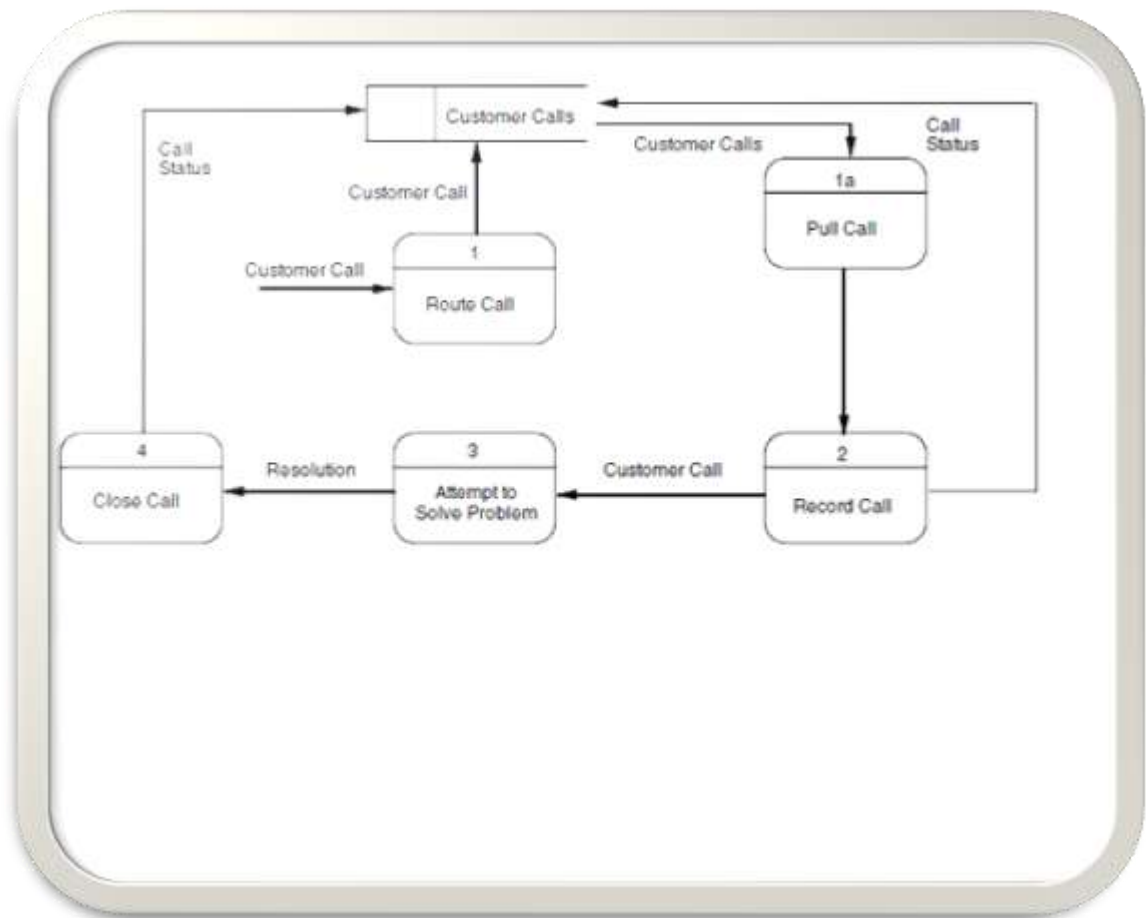
Η CBR τεχνολογία έχει δημιουργήσει μια σειρά επιτυχώς εγκατεστημένων συστημάτων, όπως για παράδειγμα το σύστημα Compaq SMART.

### 2.8 Γνωστά CBR Συστήματα

Εδώ παρουσιάζονται τα πιο γνωστά CBR συστήματα τα οποία έχουν παρουσιαστεί στη διεθνή κοινότητα.

- **SMART: Support management automated reasoning technology for Compaq customer service**

Εξαιτίας της ολοένα και πιο ανταγωνιστικής φύσης της βιομηχανίας κατασκευής υπολογιστικών συστημάτων, η Compaq Computer Corporation σημειώνει κάποιες καθοριστικές αλλαγές στον τρόπο που κάνει τις εργασίες της. Μία από αυτές τις αλλαγές είναι η επέκταση του συστήματος καταγραφής κλήσεων της Compaq να περιλαμβάνει μια συνιστώσα ανάλυσης προβλημάτων, που θα βοηθήσει το προσωπικό υποστήριξης πελατών για τον καθορισμό του ψηφίσματος στα ερωτήματα και τα προβλήματα του πελάτη. Ένα από τα εργαλεία που κάνει αυτό το κατόρθωμα είναι δυνατό είναι το σύστημα SMART (διαχείριση - στήριξη αυτοματοποιημένη τεχνολογία συλλογιστικής). Το SMART αποτελεί μέρος μιας στρατηγικής της Compaq για την αύξηση της αποτελεσματικότητας του προσωπικού εξυπηρέτησης των πελατών και τη μείωση του συνολικού κόστους για τον οργανισμό, με τη συγκράτηση επίλυση των προβλημάτων βάση της γνώσης και τη διάθεσή τους στο σύνολο του προσωπικού υποστήριξης.

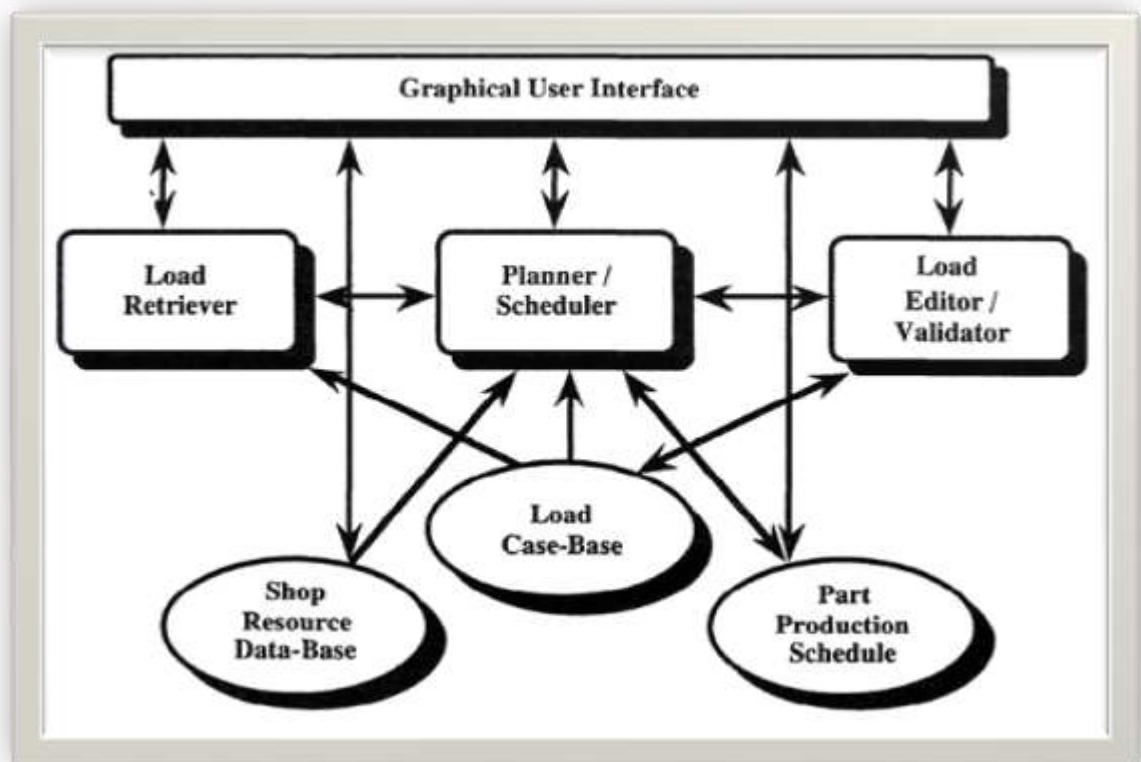


Εικόνα 2.5.1 Smart cbr System Problem Resolution Workflow

### ▪ CLAVIER: Applying case-based reasoning on to composite part fabrication

Το Clavier είναι ένα σύστημα συλλογιστικής βασισμένης σε περιπτώσεις (CBR), το οποίο βοηθά για τον αποτελεσματικό καθορισμό φορτίων των σύνθετων υλικών μερών να καούν. Κεντρικός σκοπός του Clavier είναι η εξεύρεση των πιο κατάλληλων ομάδων και συνθέσεις του μέρη (ή φορτία), προκειμένου να μεγιστοποιηθεί ο ρυθμός καύσης (throughput) ενώ παράλληλα διασφαλίζεται ότι τα μέρη είναι σωστά αναμιγμένα. Ο Clavier χρησιμοποιεί περίπτωση με βάση το σκεπτικό για την αντιστοιχία του καταλόγου των συστατικών μερών που πρέπει να αναμιχθούν αντί μια βιβλιοθήκη με προηγούμενες επιτυχημένες χρήσεις φορτίων και προτείνει το πιο κατάλληλο επόμενο φορτίο. Ο Clavier χρησιμοποιεί επίσης ένα heuristic scheduler για να παράγει μια ακολουθία των φορτίων που ανταποκρίνεται καλύτερα στο σκοπό της παραγωγής, ενώ ικανοποιεί επιχειρησιακούς περιορισμούς. Το σύστημα αυτό χρησιμοποιείται καθημερινά

στην παραγωγή και έχει εξαλείψει σχεδόν την παραγωγή των χαμηλής ποιότητας εξαρτημάτων που θα πρέπει να είναι σε αχρηστία, έτσι εξοικονομούνται χιλιάδες δολάρια κάθε μήνα. Ως μία από τις πρώτες χρήσεις της συλλογιστικής με βάση περιπτώσεις, το Clavier αποδεικνύει ότι το CBR να είναι μια πρακτική τεχνολογία που μπορεί να χρησιμοποιηθεί με επιτυχία σε τομείς όπου παραδοσιακές προσεγγίσεις είναι δύσκολο να εφαρμόζονται.



Εικόνα 2.5.2 The Clavier high-level functional architecture.

- **Appliance Call Center automation at General Electric**

Η General Electric δημιούργησε ένα σύστημα συλλογιστικής βασισμένη σε περιπτώσεις για την εξ αποστάσεως διάγνωση, την αυτοματοποίηση του τηλεφωνικού κέντρου, καθώς και τα εσωτερικά έργα παραγωγικότητας. CBR συστήματα έχουν χρησιμοποιηθεί για την εξ αποστάσεως διάγνωση μηχανημάτων ακτίνων X, μηχανές (κινητήρων), και κινητήρων αεροσκαφών. Τηλεφωνικά κέντρα έχουν CBR συστήματα που βοηθούν τους εργαζομένους να βοηθήσουν αυτοί με τη σειρά τους τους πελάτες, την

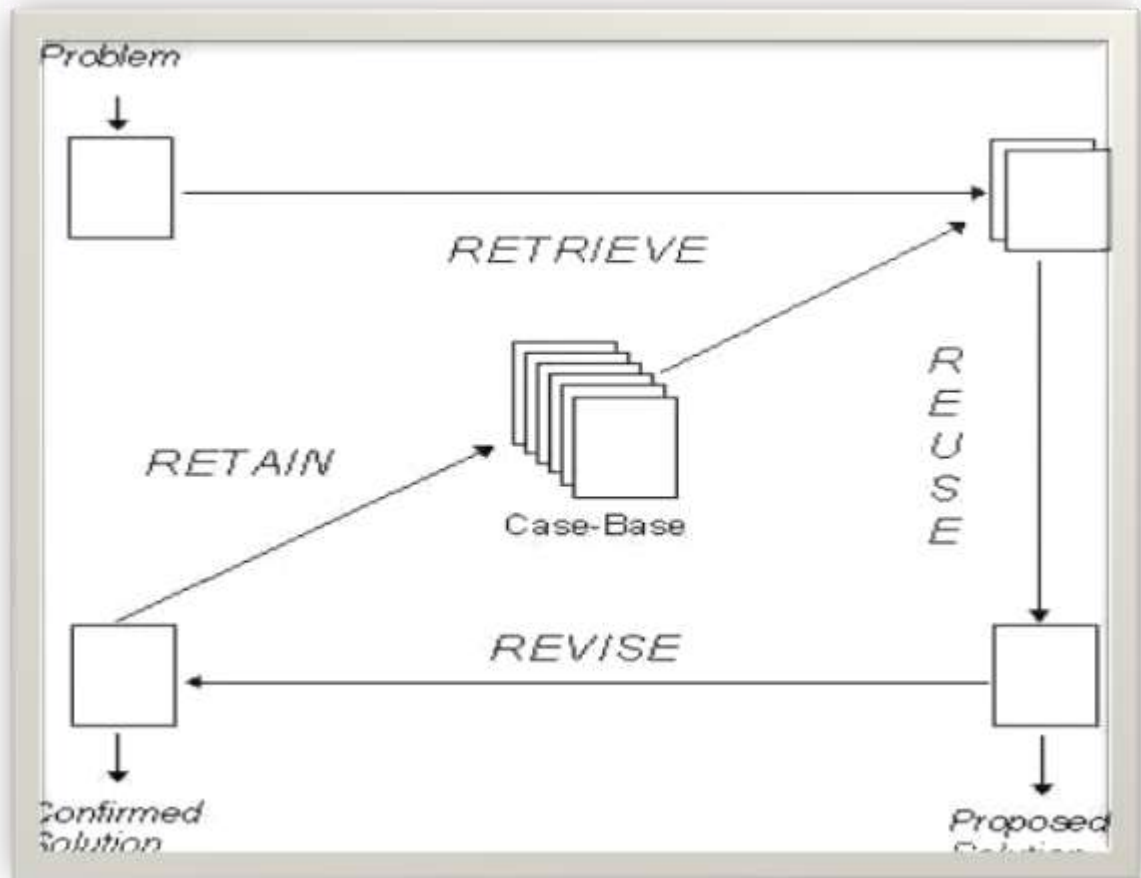
παροχή web-based αυτοεξηγηρέτηση πελατών, και να προτείνει τυποποιημένες απαντήσεις στα e-mail μηνύματα. Παραγωγικά εργαλεία που βασίζονται στο CBR έχουν αυτοματοποιήσει την διαδικασία αξιοποίησης ακινήτου για την GE ιδιοκτησία, και βοήθησε τη GE Plastics να καθορίσει τη σωστή χρωστική ουσία για τη δημιουργία ενός προσαρμοσμένου πλαστικού χρώματος, και επέτρεψε στη GE Plastics να παρέχει ένα web-based εργαλείο για την επιλογή πλαστικού χρώματος.

- **FormTool: Plastics Color Matching**

Ένα σύστημα συλλογιστικής βασισμένης σε περιπτώσεις, για τον προσδιορισμό των χρωστικών ουσιών που θα χρησιμοποιηθούν για την παραγωγή ενός ιδιαίτερου χρώματος των πλαστικών, δημιουργήθηκε. Η επιλογή των χρωστικών ουσιών πρέπει να λαμβάνει πολλούς παράγοντες υπόψη. Μια τεχνική που έκανε χρήση της ασαφούς λογικής χρησιμοποιήθηκε για να συγκρίνει την ποιότητα του χρώματος που ταιριάζουν για κάθε παράγοντα. Το σύστημα ήταν σε χρήση για δύο χρόνια σε έναν αυξανόμενο αριθμό των τοποθεσιών της GE Plastics και έδειξε σημαντική εξοικονόμηση κόστους.

- **CoolAir: HVAC specification and pricing system**

Αναπτύχθηκε ως συμπλήρωμα σε ένα υπάρχον σύστημα που χρησιμοποιεί συλλογιστική βασισμένη σε περιπτώσεις για την επαναχρησιμοποίηση προηγούμενων προδιαγραφών για την εγκατάσταση HVAC και σχεδιασμού. Το σύστημα που περιγράφεται αφήνει την μηχανική ανάκληση λεπτομέρειων της εγκατάστασης, την ανάθεση και λειτουργικά προβλήματα με τα συστήματα HVAC.



Εικόνα 2.5.3 CoolAir

- **Vidur - A CBR based intelligent advisory system, by C-DAC Mumbai, for farmers of North-East India.**

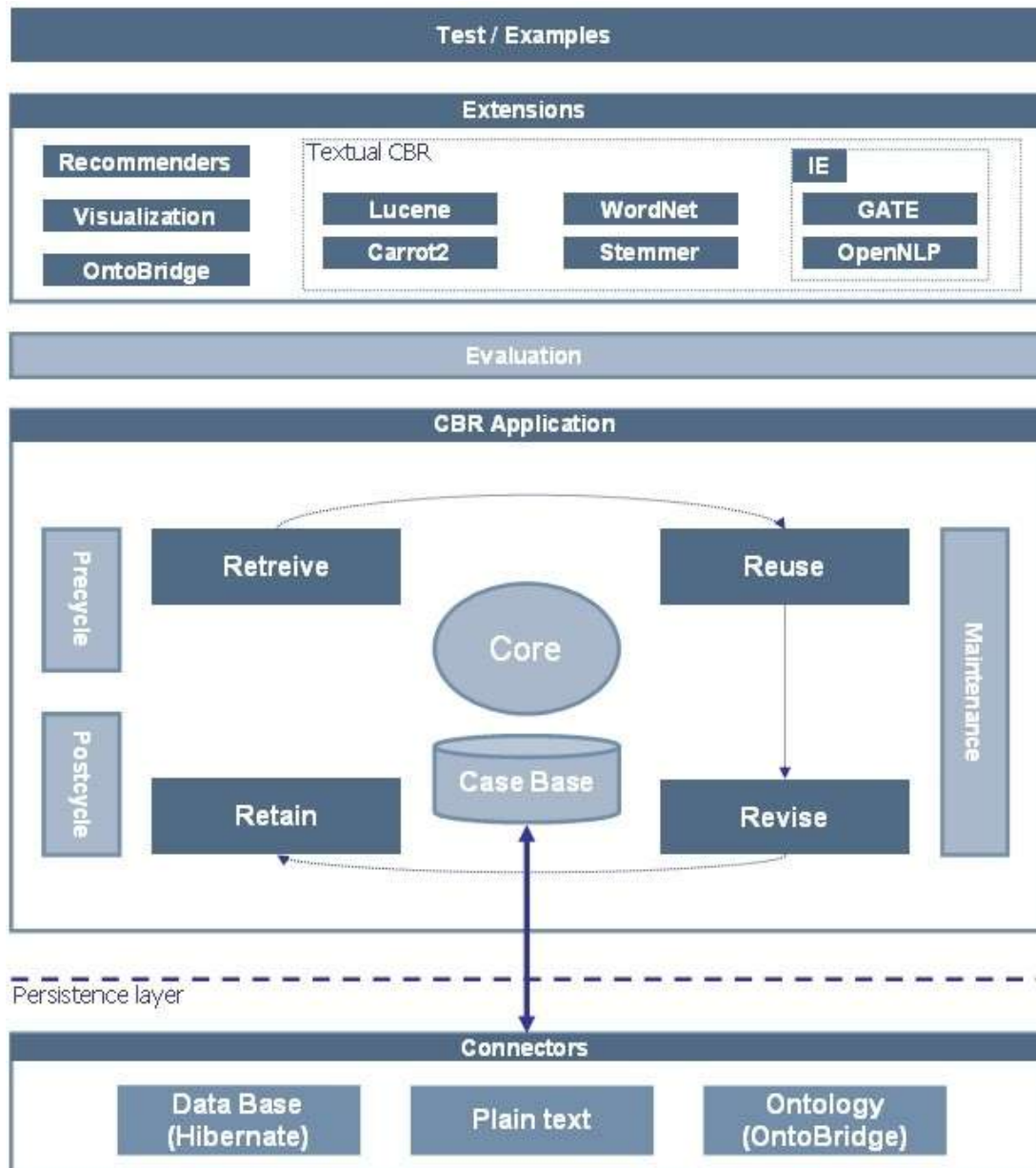
Η C-DAC στη Βομβάη έχει αναπτύξει ένα συμβουλευτικό σύστημα βασισμένο σε CBR που ονομάζεται Vidur, για τους αγρότες της κρατικής Μανιπούρ. Το Vidur αρχικά αναπτύχθηκε για την καταπολέμηση των ζιζανίων και για την επιλογή αναποφλοιώτης ποικιλίας, με τη βοήθεια των γεωργικών εμπειρογνομόνων από το Κεντρικό Γεωπονικό Πανεπιστήμιο (Cau), Iroisemba, Ιμπχάλ.

- **jColibri - A CBR framework that can be used to build your own CBR system.**

Το jCOLIBRI1 είναι ένα open source Java Framework που βοηθά το σχεδιασμό Συλλογιστικών Βασισμένων σε Περιπτώσεις συστημάτων, που διαθέτει επίσης

## Συλλογιστική βασισμένη σε περιπτώσεις

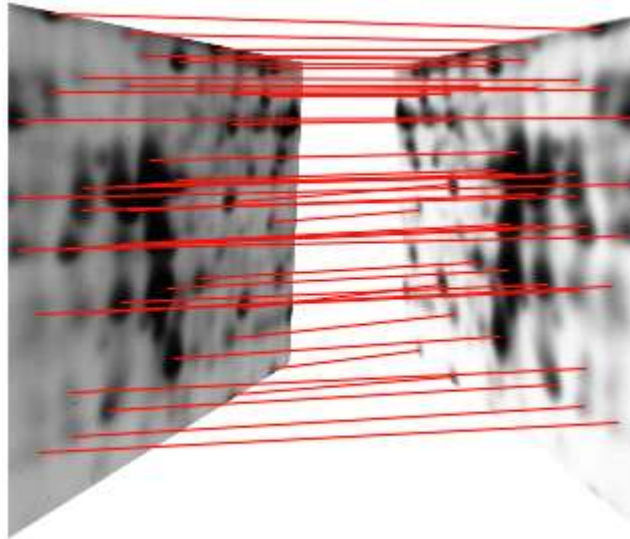
ημιαυτόματα εργαλεία ρύθμισης που επιτρέπει τη δημιουργία ενός CBR συστήματος χωρίς να γράφετε μια γραμμή κώδικα. Σχετίζεται επίσης με τη βασική έρευνα σχετικά με μηχανική συστημάτων βασισμένων στη γνώση και την εφαρμογή τεχνικών Τεχνητής Νοημοσύνης για τη διευκόλυνση της μη προγραμματιστών στην κατασκευή προγραμμάτων.



Εικόνα 2.5.4 jColibri



## 2.9 Ταίριασμα βάσει προτύπου (Pattern Matching)



Εικόνα 2.7.1 Pattern Matching

Πηγή: [www.cs.helsinki.fi](http://www.cs.helsinki.fi)

Στην επιστήμη των υπολογιστών, το ταίριασμα βάσει προτύπου είναι η πράξη του ελέγχου ορισμένων ελέγχων ακολουθίας για την παρουσία των συστατικών κάποιου προτύπου. Σε αντίθεση με την αναγνώριση προτύπων, το ταίριασμα πρέπει συνήθως να είναι ακριβές. Τα πρότυπα έχουν συνήθως τη μορφή είτε ακολουθιών ή δομών δέντρο. Χρήσεις του ταίριασματος προτύπου περιλαμβάνουν απόδοση της θέσης (εάν υπάρχει) ενός προτύπου σε μια συμβολική ακολουθία, να εμφανίζει κάποιο συστατικό στοιχείο του ταιριασμένου προτύπου και να αντικατασταθούν τα ταιριασμένα πρότυπα, με επιλογή, με κάποια άλλη συμβολική ακολουθία (Δηλαδή, αναζήτηση και αντικατάσταση).

Ακολουθία προτύπων (π.χ., μια συμβολοσειρά κειμένου) συχνά περιγράφονται

χρησιμοποιώντας τις συνήθεις εκφράσεις και ταιριάζονται με τη χρήση τεχνικών όπως η υπαναχώρηση (backtracking).

Πρότυπα δέντρων χρησιμοποιούνται σε ορισμένες γλώσσες προγραμματισμού σαν ένα γενικό εργαλείο για την επεξεργασία δεδομένων βάσει της δομής τους, π.χ., Haskell, ML και η συμβολική γλώσσα των μαθηματικών Mathematica έχουν ειδική σύνταξη για την έκφραση των μοντέλων δέντρων και μια δομή της γλώσσας για την υπό όρους την εκτέλεση και την ανάκτηση τιμών. Για λόγους απλότητας και αποτελεσματικότητας, από αυτά τα πρότυπα δέντρων λείπουν μερικά χαρακτηριστικά που είναι διαθέσιμα σε συνήθεις εκφράσεις.

Συχνά είναι δυνατό να δίνονται διαφορετικά σχέδια που δοκιμάζονται το ένα μετά το άλλο, πράγμα που δίνει ένα ισχυρό προγραμματιστικό κατασκεύασμα.

Γλώσσες αναδιαμόρφωσης όρων και αναδημιουργίας Γράφων βασίζονται σε ταίριασμα των προτύπων όπου ένα πρόγραμμα καταλήγει σε ένα αποτέλεσμα.

Τα πρώτα προγράμματα υπολογιστή που χρησιμοποίησαν ταίριασμα προτύπου ήταν αυτά της επεξεργασίας κειμένου. Στα Bell Labs, ο Ken Thompson επέκτεινε τα χαρακτηριστικά της αναζήτησης και αντικατάστασης του QED editor για να δεχθεί τις συνήθεις εκφράσεις. Πρώιμες γλώσσες προγραμματισμού που είχαν pattern matching δομές ήταν οι SNOBOL από το 1962, SASL από το 1976, NPL από το 1977, και KRC από το 1981. Η πρώτη γλώσσα προγραμματισμού με tree matching στοιχεία ήταν το extension του Fred McBride, για την LISP, το 1970.

## 2.10 Βασικές μορφές προτύπων(primitive patterns)

Η απλούστερη μορφή προτύπου στο ταίριασμα προτύπων είναι η ρητή τιμή ή μια μεταβλητή. Για παράδειγμα, σκεφτείτε έναν απλό ορισμό της συνάρτησης στο συντακτικό Haskell (οι παράμετροι λειτουργίας δεν είναι σε παρενθέσεις, αλλά είναι χωρισμένες με κενά, = δεν είναι παραχώρηση αλλά ορισμός):

$f\ 0 = 1$
------------

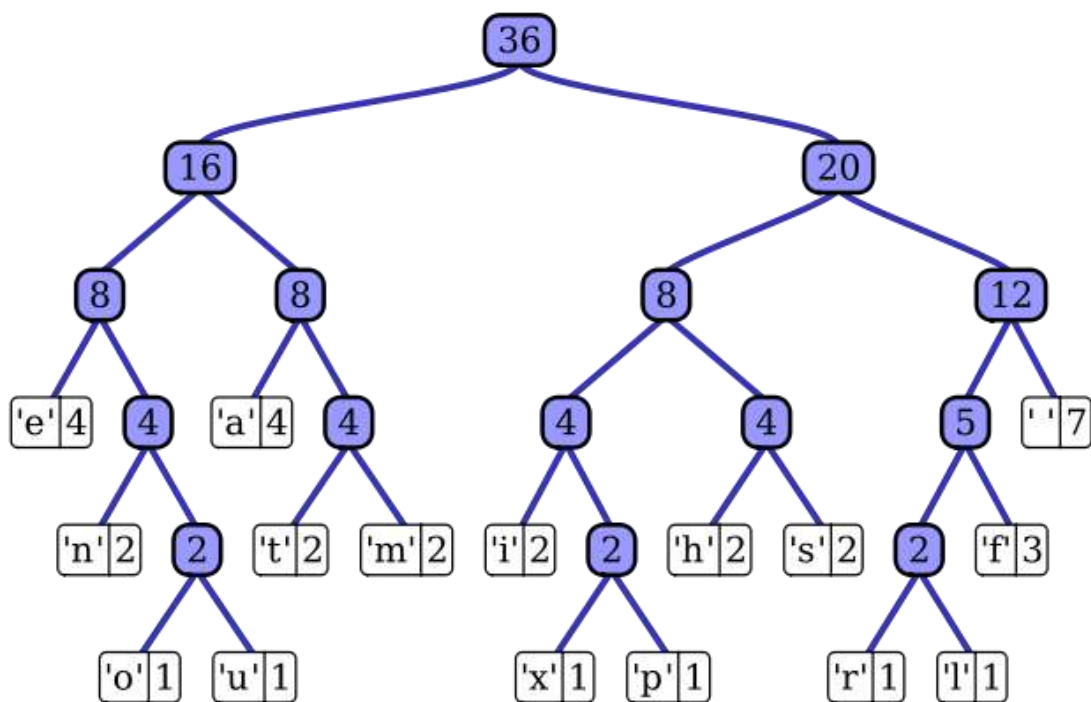
Εδώ, το 0 είναι ένα και μοναδικό πρότυπο τιμής. Τώρα, όταν το f δίνεται 0 ως παράμετρος το πρότυπο ταιριάζει και η συνάρτηση επιστρέφει 1. Με οποιοδήποτε άλλο επιχειρήμα, το ταίριασμα και η λειτουργία αποτυγχάνουν. Εφόσον το συντακτικό υποστηρίζει εναλλακτικά πρότυπα σαν ορισμούς σε συναρτήσεις, μπορούμε να συνεχίσουμε με τον ορισμό ώστε η παράσταση αυτή να λάβει γενικότερες τιμές:

$n = f * n\ f\ (n-1)$
-----------------------

Εδώ, το πρώτο n είναι ένα και μοναδικό πρότυπο μεταβλητής, η οποία θα ταιριάζει απολύτως με οποιοδήποτε επιχειρήμα και θα το συνδέσει με τη μεταβλητή n που θα χρησιμοποιείται στο υπόλοιπο του ορισμού. Στο Haskell (σε αντίθεση με τουλάχιστον το Hope), τα πρότυπα δοκιμάζονται με τη σειρά έτσι ο πρώτος ορισμός ισχύει ακόμη στην πολύ ειδική περίπτωση της εισόδου να είναι 0, ενώ για κάθε άλλο επιχειρήμα των συνάρτησης επιστρέφει  $n * f\ (n-1)$  με το «n» το επιχειρήμα.

Το πρότυπο μπαλαντέρ (συχνά γράφεται ως `_`) είναι επίσης απλό: όπως μια μεταβλητή όνομα, ταιριάζει με κάθε τιμή, αλλά δεν δεσμεύει την τιμή σε οποιοδήποτε όνομα.

## 2.11 Πρότυπα δέντρων (Tree Patterns)



Εικόνα 2.9.1 Δέντρο Huffman

Πηγή: <http://sector0.dk>

Πιο περίπλοκα σχέδια μπορούν να χτιστούν απ' ότι τα πρωτόγονα εκείνα της προηγούμενης ενότητας, συνήθως με τον ίδιο τρόπο όπως και οι τιμές είναι χτισμένες με συνδυασμό άλλων τιμών. Η διαφορά είναι ότι στη συνέχεια με μεταβλητά και μπαλαντέρ μέρη, ένα πρότυπο που δεν δημιουργείται με ενιαία τιμή, αλλά ταιριάζει με μια ομάδα τιμών που είναι ο συνδυασμός των διακριτών στοιχείων και των στοιχείων που μπορούν να ποικίλλουν μέσα στη δομή του προτύπου.

## Συλλογιστική βασισμένη σε περιπτώσεις

---

Ένα πρότυπο δέντρου περιγράφει ένα μέρος του δέντρου, ξεκινώντας με ένα κόμβο και διευκρινίζει ορισμένους κλάδους και κόμβους και αφήνει κάποιους αόριστους με μια μεταβλητή ή πρότυπο μπαλαντέρ. Μπορεί να βοηθήσει αν σκεφτούμε το αφηρημένο συντακτικό δέντρου μιας γλώσσας προγραμματισμού και αλγεβρικών τύπων δεδομένων.

Στο Haskell η ακόλουθη γραμμή καθορίζει ένα αλγεβρικό τύπο χρώματος δεδομένων που έχει ένα και μόνο κατασκευαστή `ColorConstructor` δεδομένα που περιέχει έναν ακέραιο και μια συμβολοσειρά.

```
data Color = ColorConstructor Integer String
```

Ο κατασκευαστής είναι ένας κόμβος σε ένα δέντρο και ο ακέραιος και η συμβολοσειρά είναι τα φύλλα σε κλαδιά.

Όταν θέλουμε να γράψουμε συναρτήσεις για να κάνουμε το Χρώμα ένα αφηρημένο τύπο δεδομένων, θα θέλαμε να γράψουμε τις λειτουργίες για διασύνδεση με το είδος των δεδομένων, και έτσι θέλουμε να εξάγουμε ορισμένα στοιχεία από τον τύπο δεδομένων, για παράδειγμα, μόνο τη συμβολοσειρά ή απλά το ακέραιο μέρος του Χρώματος.

Αν περάσει μια μεταβλητή που είναι τύπου `Color`, πώς μπορούμε να παίρνουμε τα δεδομένα από αυτή τη μεταβλητή; Για παράδειγμα, για μια λειτουργία για να πάρει το μέρος ακέραιο του χρώματος, μπορούμε να χρησιμοποιήσουμε ένα απλό σχήμα δέντρου και να γράψει:

```
integerPart (ColorConstructor _ theInteger) = theInteger
```

Καθώς επίσης και:

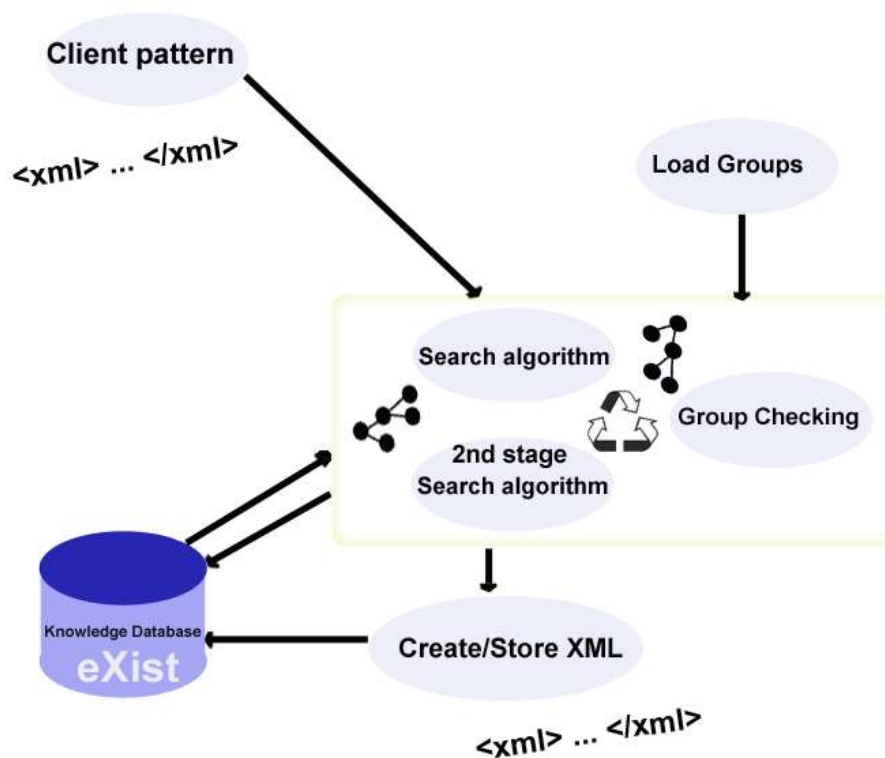
```
stringPart (ColorConstructor _ theString) = theString
```

Η δημιουργίες από αυτές τις συναρτήσεις μπορούν να αυτοματοποιηθούν με το συντακτικό εγγραφής δεδομένων του Haskell.

## Κεφάλαιο 3

### Περιγραφή του συστήματος διακόσμησης εσωτερικών χώρων

### 3.1 Γενικά περί του αλγορίθμου επίλυσης του προβλήματος διακόσμησης εσωτερικών χώρων.

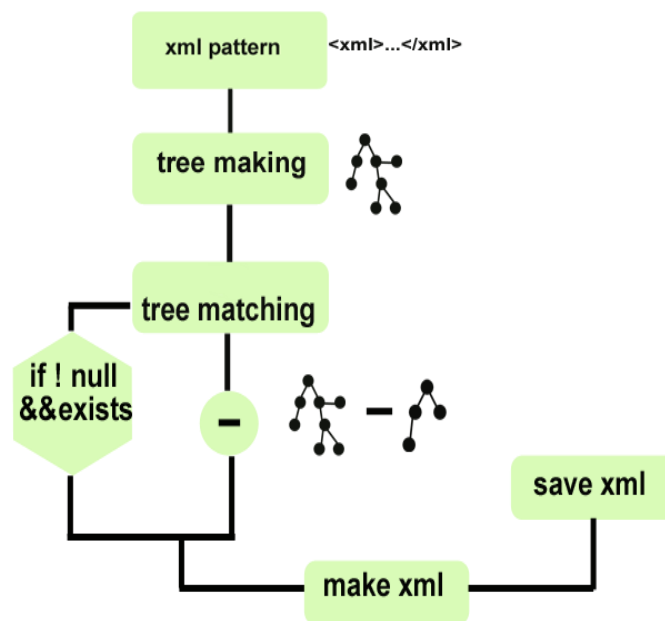


Εικόνα 3.1.1 Σχηματική αναπαράσταση αλγορίθμου.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η δημιουργία και υλοποίηση ενός αλγορίθμου ο οποίος θα δέχεται σαν είσοδο ένα xml έγγραφο και θα προσπαθεί να υλοποιήσει ένα νέο xml στην έξοδό του το οποίο να του μοιάζει όσο το δυνατόν

περισσότερο. Γι αυτό το σκοπό χρησιμοποιεί μια βάση γνώσης από την οποία αντλεί πληροφορίες συνεχώς. Αυτή η βάση κρατάει όλη τη μνήμη του συστήματος μας.

Η αρχιτεκτονική πάνω στην οποία είναι υλοποιημένο το σύστημά μας είναι case-based reasoning, δηλαδή αποφασίζει βάσει προηγούμενων περιπτώσεων. Πράγμα που σημαίνει ότι το σύστημά μας, όσο χρησιμοποιείται μαθαίνει περιπτώσεις, άρα με κάθε χρήση του, γίνεται πιο ακριβές και αποτελεσματικό. Σκοπός αυτού είναι να χρησιμοποιηθεί αργότερα σαν τμήμα ενός συνόλου εφαρμογών οι οποίες θα λειτουργήσουν στον τομέα της εσωτερικής διακόσμησης χώρων (interior design).



Εικόνα 3.1.2 Διάγραμμα ροής αλγορίθμου μετατροπής και αναζήτησης.

Ουσιαστικά ο αλγόριθμος μας αφού πάρει σαν είσοδο το ζητούμενο xml, το αναλύει σε xml paths (διαδρομές xml), δημιουργώντας ένα δέντρο. Στη συνέχεια γίνεται αναζήτηση στη βάση γνώσης για το πιο όμοιο δέντρο, σε σχέση με αυτό που απαιτείται



από το ζητούμενο (pattern xml), μέσω ενός αλγορίθμου αναζήτησης. Εφόσον βρεθεί ένα σετ από όμοια δέντρα, τότε αυτό με την μεγαλύτερη ομοιότητα κρατείτε σε ένα πίνακα και δημιουργείται ακόμα ένας δυναμικός πίνακας ο οποίος κρατάει τα paths (φύλλα ή ακόμα και κλαδιά στο δέντρο) τα οποία δεν ταίριαζαν. Επίσης σε αυτό το στάδιο γίνεται και ένας έλεγχος ομαδοποίησης των δεδομένων για να αποφευχθούν λάθη κατά την αναζήτηση διαδρομών.

Σε επόμενο στάδιο τα paths τα οποία δεν βρέθηκαν όμοια μπαίνουν στον αλγόριθμο αναζήτησης και αναζητείται πάλι στη βάση γνώσης ένα έγγραφο που να τα περιέχει. Επίσης από το σετ που θα προκύψει κρατείται σε ένα πίνακα το καλύτερο (αυτό δηλαδή με την μεγαλύτερη ομοιότητα) και δημιουργείται ακόμα ένας πίνακας με τα paths που δεν βρέθηκαν. Επίσης ο πίνακας με τις διαδρομές οι οποίες είναι όμοιες προστίθεται σε αυτόν του προηγούμενου κύκλου. Η διαδικασία στη συνέχεια είναι η ίδια και κάθε φορά περιορίζονται οι διαδρομές (paths) που πρέπει να βρεθούν, έως ότου αυτές είτε βρεθούν όλες είτε να μην υπάρχουν.

Εφόσον φτάσουμε στο σημείο όπου ο πίνακας με τις αταίριαστες διαδρομές τελειώσει τότε ο αλγόριθμος αναζήτησης σταματά και ο πίνακας με τις διαδρομές που ταίριαζαν περνάει στη φάση δημιουργίας xml εγγράφου.

Στο στάδιο δημιουργίας xml εγγράφου χρησιμοποιείται ένα κενό xml, δηλαδή ένα έγγραφο το οποίο περιέχει τη σωστή δομή αλλά στη θέση τις τιμές για την εκάστοτε μεταβλητή έχουμε βάλει null(κάτι σαν καλούπι). Τότε διαβάζουμε ένα ένα τα πεδία του πίνακα που εισήχθη και πηγαίνουμε στην κατάλληλη θέση του κενού εγγράφου και τοποθετούμε την τιμή που του αναλογεί.

Στη συνέχεια περνάμε στο στάδιο της αποθήκευσης στη βάση γνώσης. Το νέο έγγραφο που μόλις δημιουργήθηκε πρέπει να αποθηκευτεί στη βάση έτσι ώστε να αναπτυχθεί και η γνώση του συστήματος μας. Στο στάδιο αυτό λοιπόν απλά γίνεται η αποθήκευση του νέου εγγράφου στη βάση δίνοντας του ένα τυχαίο όνομα. Αυτό είναι το στάδιο το οποίο κάνει τον αλγόριθμό μας να «θυμάται», γιατί αν υποθέσουμε πως στη βάση μας μέχρι τώρα υπήρχαν πέντε (5) xml έγγραφα για αναζήτηση, την επόμενη φορά που θα ζητηθεί κάτι θα υπάρχουν πλέον  $5+1=6$  έγγραφα για συνδυασμό. Άρα εδώ έγκειται και η τεχνητή νοημοσύνη του όλου συστήματος εφόσον ο αλγόριθμος μαθαίνει άλλη μια περίπτωση, αυτή που μόλις δημιούργησε.

Τέλος το έτοιμο πλέον νέο έγγραφο xml μπορεί να παραδοθεί στον χρήστη από τον οποίο προήλθε το pattern.

## 3.2 Δομή XML εγγράφων περιγραφής Δωματίου

### 3.2.1 Δομή XML pattern εγγράφου

#### Ιεραρχία - επίπεδο 1:

Στην κορυφή της ιεραρχίας, υπάρχει μόνο η το στοιχείο της xml.

#### Ιεραρχία - Επίπεδο 2

Underneath the xml element, there are two sub elements: Κάτω από το xml στοιχείο, υπάρχουν δύο επιμέρους στοιχεία:

header


room



Διάγραμμα 1

#### Ιεραρχία - Επίπεδο 3

Όπως φαίνεται στο διάγραμμα 1, σε πρώτη φάση, έχουμε ορίσει το στοιχείο κεφαλίδα, room. Το στοιχείο κεφαλίδα περιλαμβάνει πληροφορίες που περιγράφει το decoXML του συγκεκριμένου έργου.



xml	
▲ header	
File	filepath
Date	
Designer	
Company	
Project	
config	
▲ room	
model3D	
shape	SVG something
dimension	number, number, number, -[x],[y],[z] bounding box
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box) inside the schema
style	styleIndexName=style index name st...
ColorPalette	ColorPaletteIndexName=pale...
functionality	
RoomCoveringF...	percentage of the covered space (by furnitures etc) bounding box of the room - Sum(bounding boxes of the furnitures)/bounding box of the room
lighting	in lux the light volume of the room (artificial and daylight as well)
ceiling	
wall	
floor	
stair	type=stair type

Διάγραμμα 2

Όπως μπορούμε να δούμε από το διάγραμμα 2, η **επικεφαλίδα στοιχείου** περιλαμβάνει τα ακόλουθα επιμέρους στοιχεία:

**File:** είναι το στοιχείο όπου ορίζεται η διαδρομή αρχείου.

**Date:** είναι το στοιχείο που αναφέρεται στην ημερομηνία που το decoXML δημιουργήθηκε .

**Designer:** είναι το στοιχείο που αναφέρεται στο σχεδιαστή του συγκεκριμένου εσωτερικού χώρου.

Company: είναι το στοιχείο που αναφέρεται στην εταιρεία η οποία παρέχει την συγκεκριμένη εσωτερική διακόσμηση

Όλα τα παραπάνω δεν έχουν τα επιμέρους στοιχεία (child elements).

Project: "Είναι το στοιχείο που αναφέρεται σε πληροφορίες σχετικά με το έργο για το οποίο εργαζόμαστε .

Config: Είναι το στοιχείο που αναφέρεται στον ορισμό των μονάδων που πρόκειται να χρησιμοποιηθούν.

Το **δωμάτιο στοιχείο** αποτελείται από τα εξής επιμέρους στοιχεία:

Model3D: είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση.

Shape: Είναι το στοιχείο που αναφέρεται στο σχήμα του δωματίου. Το σχήμα αυτό παρουσιάζεται με τη βοήθεια του SVG (Scalable Vector Graphics) το οποίο μπορεί να περιγράψει 2 διαστάσεων διανυσματικά γραφικά.

Dimension: Είναι το στοιχείο που αναφέρεται στις διαστάσεις του δωματίου και πρέπει να είναι σε συγκεκριμένη μορφή (αριθμός, αριθμός, αριθμός-| x |, | y |, | z | bounding box)

Orientation: Είναι το στοιχείο που αφορά στον προσανατολισμό της χώρα, ο οποίος και πάλι θα πρέπει να έχει τη μορφή που περιγράφεται στο περιεχόμενο του στοιχείου.

Style: Είναι το στοιχείο που παραπέμπει στο ύφος του δωματίου. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

- styleIndexName: Είναι ο δείκτης του ονόματος του στυλ που χρησιμοποιείται.
- styleIndexUrl: Είναι το URL του στυλ που χρησιμοποιείται.

Ως περιεχόμενο, το style στοιχείο έχει τον κωδικό του στυλ που χρησιμοποιείτε.

ColorPalette: Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται σε ένα δωμάτιο. Το ColorPalette στοιχείο μπορεί να περιλαμβάνει

περισσότερα από ένα χρώματα, δεδομένου ότι ένα δωμάτιο μπορεί να έχει ένα συνδυασμό χρωμάτων. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

- `ColorPaletteIndexName`:: Είναι ο δείκτης του το όνομα του χρώματος της παλέτας που χρησιμοποιείται.
- `ColorPaletteIndexURL`: Είναι η διεύθυνση URL της παλέτας χρωμάτων.

Το στοιχείο `ColorPalette` έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιείται.

`Functionality`: Είναι το στοιχείο που αναφέρεται στη λειτουργικότητα του χώρου.

`RoomCoveringFactor`: Είναι το στοιχείο που αναφέρεται στο ποσοστό του καλυμμένου χώρου του δωματίου από έπιπλα κ.λπ. Αυτός ο παράγοντας εμφανίζεται όταν αφαιρούμε το άθροισμα των bounding boxes των επίπλων από το bounding box του δωματίου και το διαιρέσουμε με το bounding box του δωματίου.

`Lighting`: Είναι το στοιχείο που αναφέρεται στην lux (SI unit of illuminance ,μονάδα φωτισμού) του εσωτερικού χώρου.

`Ceiling`: Είναι το στοιχείο που αναφέρεται στην οροφή του δωματίου.

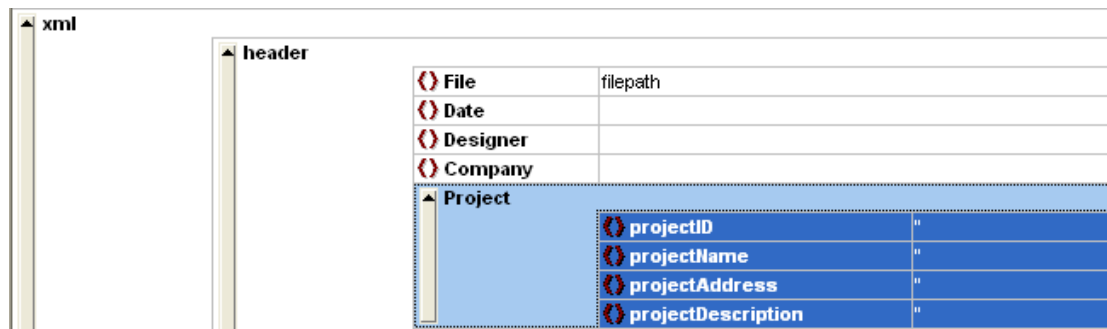
`Wall`: Είναι το στοιχείο που αναφέρεται στους τοίχους του δωματίου.

`Όροφος`: Είναι το στοιχείο που αναφέρεται στο πάτωμα του δωματίου.

`Stair`: Είναι το στοιχείο που αναφέρεται στη σκάλα / ες του δωματίου.

### Ιεραρχία - Επίπεδο 4

Στο επίπεδο 4 που έχουμε:



Διάγραμμα 3

Το **Project** στοιχείο αποτελείται από τα ακόλουθα επιμέρους στοιχεία:

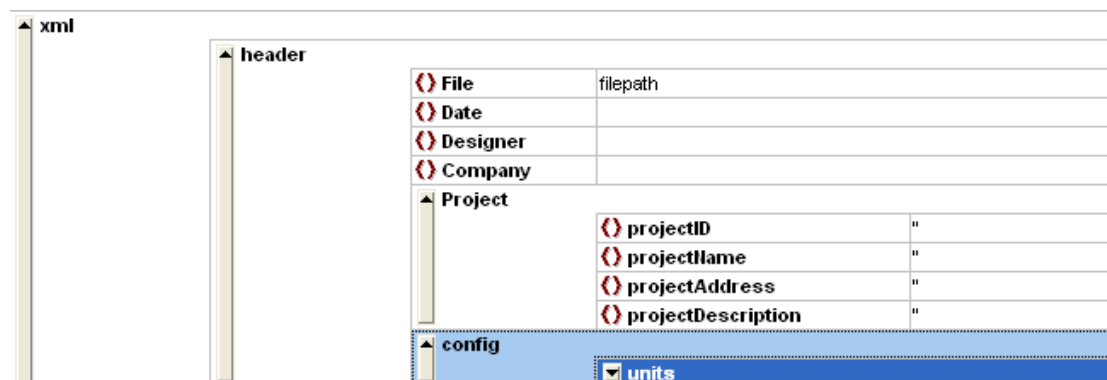
ProjectID: Είναι το στοιχείο που παραπέμπει στο ID του project.

ProjectName:: Είναι το στοιχείο που αναφέρεται το όνομα του project.

Projectaddress:: Είναι το στοιχείο που αναφέρεται στη διεύθυνση του έργου

projectDescription:: Είναι το στοιχείο που αναφέρεται στην περιγραφή του project, μπορεί να περιλαμβάνει μια περιληπτική παρουσίαση της σχετικά με το project.

Το στοιχείο **config** αποτελείται από τα εξής επιμέρους στοιχεία:

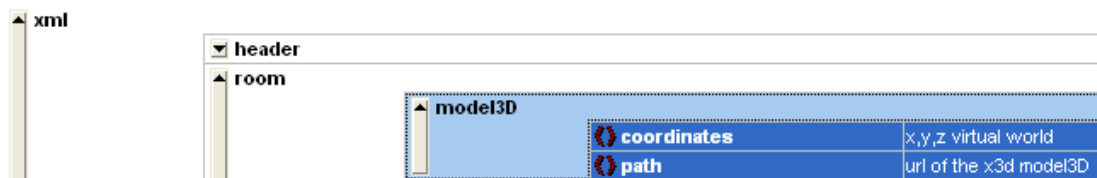


Διάγραμμα 4

Units: Είναι το στοιχείο που αναφέρεται στις μονάδες που πρόκειται να χρησιμοποιηθούν στο έγγραφο σχετικά με τα ποσοτικά στοιχεία του decoXML.

Στο ίδιο επίπεδο, αλλά κάτω από το room στοιχείο που έχουμε:

Το στοιχείο **model3D** αποτελείται από τα εξής επιμέρους στοιχεία:



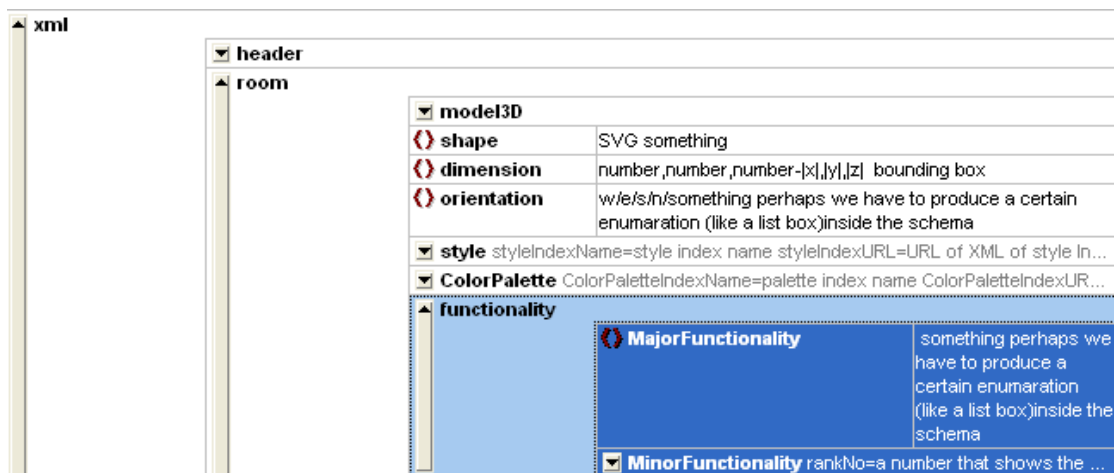
xml	header	
	room	
	model3D	
	coordinates	x,y,z virtual world
	path	url of the x3d model3D

Διάγραμμα 5

Coordinates: Είναι το στοιχείο που αναφέρεται στις συντεταγμένες του δωματίου και το περιεχόμενο που ορίζει το σύστημα συντεταγμένων.

Path: Είναι το στοιχείο που αναφέρεται στη διεύθυνση URL του x3d μοντέλου για τις περιπτώσεις όπου προχωρούμε στην οπτικοποίηση του δωματίου

Το **functionality** στοιχείο αποτελείται από τα εξής επιμέρους στοιχεία:



xml	header	
	room	
	model3D	
	shape	SVG something
	dimension	number,number,number-[x],[y],[z] bounding box
	orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
	style	styleIndexName=style index name styleIndexURL=URL of XML of style In...
	ColorPalette	ColorPaletteIndexName=palette index name ColorPaletteIndexUR...
	functionality	
	MajorFunctionality	something perhaps we have to produce a certain enumeration (like a list box)inside the schema
	MinorFunctionality	rankNo=a number that shows the ...

Διάγραμμα 6

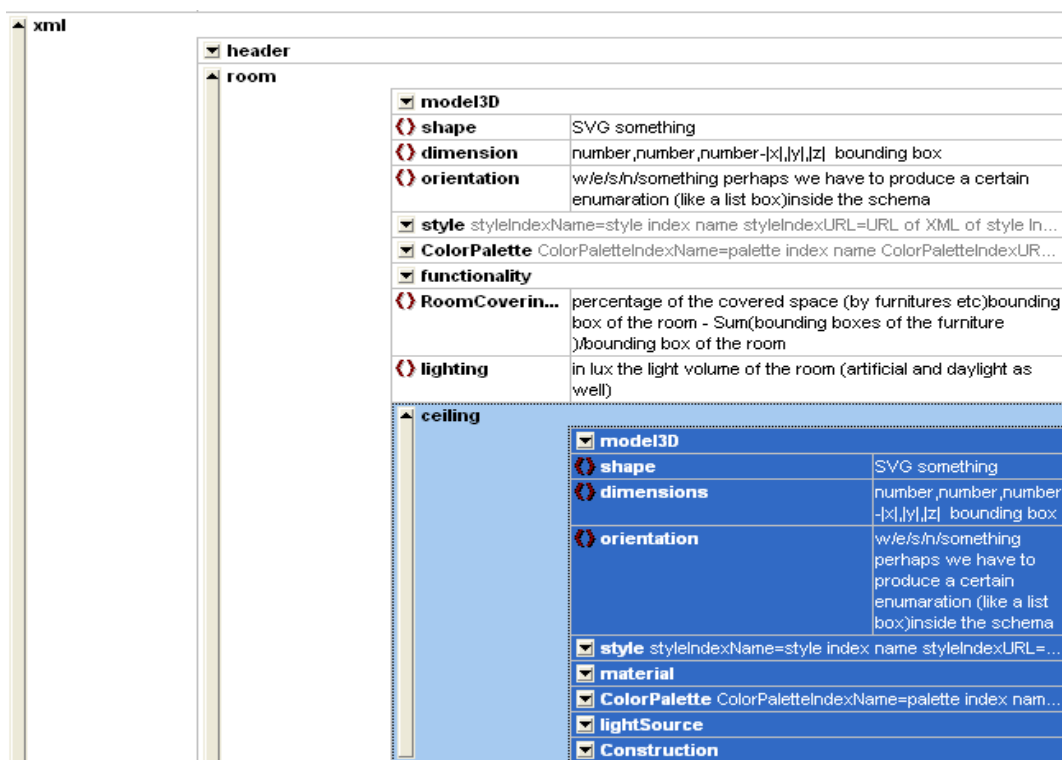


**MajorFunctionality:** Είναι το στοιχείο που αναφέρεται στην κύρια functionality του δωματίου. Μπορεί να υπάρχει μια λίστα, όπου ο χρήστης να μπορεί να επιλέξει μία από τις προτεινόμενες αίθουσες.

**MinorFunctionality:** Είναι το στοιχείο που αναφέρεται στη λιγότερο σημαντική λειτουργικότητα του χώρου. Μπορεί επίσης να συνδιαστεί με μια λίστα με προτάσεις. Για παράδειγμα, ένα δωμάτιο μπορεί να έχει MajorFunctionality υποδομάτιο και MinorFunctionality γραφείο.

Το MinorFunctionality έχει το χαρακτηριστικό rankNo το οποίο είναι ένας αριθμός που δείχνει την προτεραιότητα σε ήσσονος σημασίας λειτουργίες.

Το **ceiling** στοιχείο αποτελείται από τα εξής επιμέρους στοιχεία:



ceiling	
model3D	
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
style	styleIndexName=style index name styleIndexURL=URL of XML of style In...
ColorPalette	ColorPaletteIndexName=palette index name ColorPaletteIndexUR...
material	
lightSource	
Construction	

Διάγραμμα 7

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση.

**Shape:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG του ceiling.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις του ceiling, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** It is the element which refers to the orientation of the ceiling (w/e/s/n) Προσανατολισμός: Είναι το στοιχείο που αναφέρεται στον προσανατολισμό του ceiling (w / e / s / n)

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος του ceiling. Το στοιχείο Style έχει δύο χαρακτηριστικά:

**styleIndexName:** Είναι ο δείκτης του το όνομα του στυλ που χρησιμοποιείται.

**styleIndexUrl:** Είναι το URL του στυλ που χρησιμοποιείται.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το ceiling.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για το ceiling. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

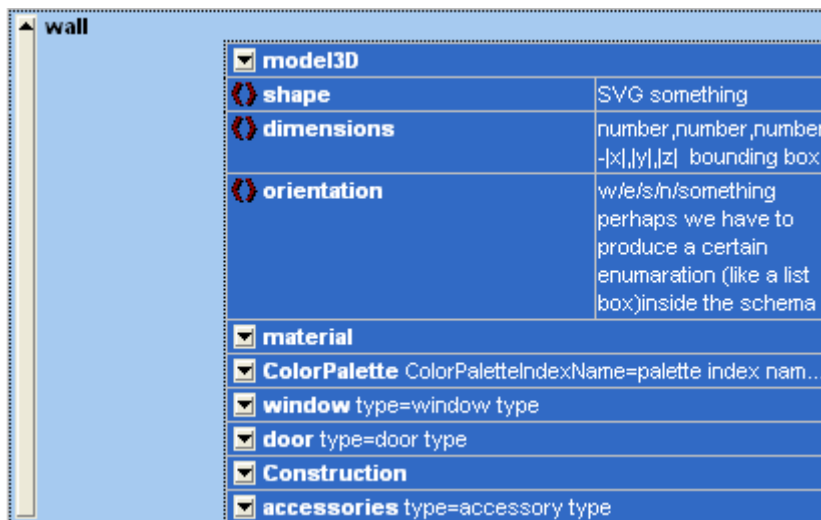
- **ColorPaletteIndexName::** Είναι ο δείκτης στο όνομα του χρώματος που χρησιμοποιείται.
- **ColorPaletteIndexURL:** Είναι η διεύθυνση URL της παλέτας χρωμάτων.

Το ColorPalette στοιχείο έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

**Lightsource:** Είναι το στοιχείο που αναφέρεται στα χαρακτηριστικά των φωτεινών πηγών που χρησιμοποιούνται στο δωμάτιο

**Construction:** It is the element which refers to the construction that the ceiling might have depending on the style. Κατασκευές: Είναι το στοιχείο που αναφέρεται στην κατασκευή του ceiling που θα μπορούσε να έχει, ανάλογα με το στυλ.

Το **wall** στοιχείο αποτελείται από τα εξής επιμέρους στοιχεία:



wall	
model3D	
shape	SVG something
dimensions	number ,number ,number - x , y , z  bounding box
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
material	
ColorPalette	ColorPaletteIndexName=palette index nam...
window	type=window type
door	type=door type
Construction	
accessories	type=accessory type

Διάγραμμα 8

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση του wall.

**Shape:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG του wall.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις του wall, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αναφέρεται στον προσανατολισμό του wall (w / e / s / n).

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το wall.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για τον τοίχο. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων

Το ColorPalette στοιχείο έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

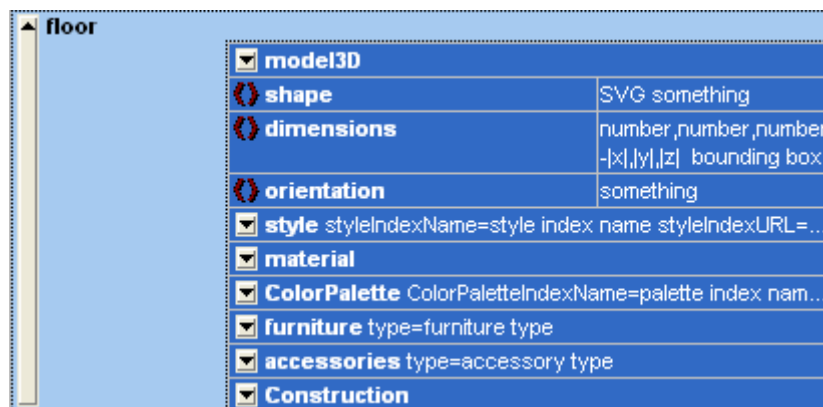
Window: Είναι το στοιχείο που αναφέρεται στα χαρακτηριστικά του παραθύρου / α από το σπίτι.

Door: Είναι το στοιχείο που αναφέρεται στα χαρακτηριστικά της πόρτας / ες του δωματίου

Construction: Είναι το στοιχείο που αναφέρεται στην κατασκευή που θα μπορούσε να γίνει στον τοίχο.

Accessories: Είναι το στοιχείο που αναφέρεται σε όλα τα στοιχεία που θα μπορούσαν να χρησιμοποιηθούν ως αξεσουάρ σε έναν τοίχο, όπως πίνακες ζωγραφικής, καθρέπτες, κλπ.

Το Floor στοιχείο αποτελείται από τα εξής επιμέρους στοιχεία:



floor	
<input type="checkbox"/> model3D	
shape	SVG something
dimensions	number,number,number -[x],[y],[z] bounding box
orientation	something
<input type="checkbox"/> style	styleIndexName=style index name styleIndexURL=...
<input type="checkbox"/> material	
<input type="checkbox"/> ColorPalette	ColorPaletteIndexName=palette index nam...
<input type="checkbox"/> furniture	type=furniture type
<input type="checkbox"/> accessories	type=accessory type
<input type="checkbox"/> Construction	

Διάγραμμα 9

Model3D: Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση του δαπέδου.

**Shape:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG του δαπέδου

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις του δαπέδου, υπολογίζεται ως η απόλυτη τιμή των  $x$ ,  $y$ ,  $z$  τιμών

**Orientation:** Είναι το στοιχείο που αναφέρεται στον προσανατολισμό του δαπέδου ( $w / e / s / n$ ).

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος του δαπέδου. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης του το όνομα του στυλ που χρησιμοποιείται .

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο Style έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείται.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το δάπεδο.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιούνται για το πάτωμα. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το ColorPalette στοιχείο έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

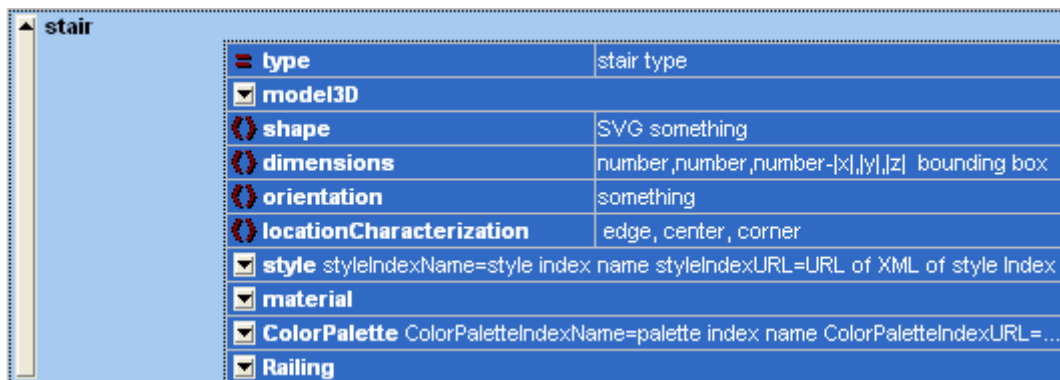
**Furnitures:** Είναι το στοιχείο που αναφέρεται στα έπιπλα που υπάρχουν στον εσωτερικό χώρο.

**Accessories:** Είναι το στοιχείο που αναφέρεται σε όλα τα στοιχεία που θα μπορούσαν να

χρησιμοποιηθούν ως αξεσουάρ σε όροφο, όπως κεριά, καλάθια, χαλιά etc.

Construction: Είναι το στοιχείο που αναφέρεται στην κατασκευή που θα μπορούσε να γίνει στο πάτωμα..

Το στοιχείο Stair έχει σαν χαρακτηριστικό το είδος της σκάλας και αποτελείται από τα εξής επιμέρους στοιχεία:



Property	Value
type	stair type
model3D	
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	something
locationCharacterization	edge, center, corner
style	styleIndexName=style index name styleIndexURL=URL of XML of style Index
material	
ColorPalette	ColorPaletteIndexName=palette index name ColorPaletteIndexURL=...
Railing	

Διάγραμμα 10

Model3D: Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση της σκάλας.

Shape: Είναι το στοιχείο που αναφέρεται στη μορφή SVG της σκάλας.

Dimensions: Είναι το στοιχείο που αναφέρεται στις διαστάσεις της σκάλας, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

Orientation: Είναι το στοιχείο που αφορά στον προσανατολισμό της σκάλας (w / e / s / n).

LocationCharacterization: Είναι το στοιχείο που αναφέρεται στη θέση της σκάλας στο χώρο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μία πρέπει να επιλέγεται: άκρη, στο κέντρο, γωνία.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος της σκάλας. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης του στο όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο Style έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό της σκάλας.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για τη σκάλα. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

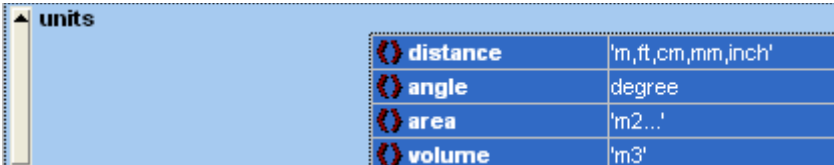
**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το ColorPalette στοιχείο έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

**Railing:** Είναι το στοιχείο που αναφέρεται στο είδος των κιγκλιδωμάτων της σκάλας.

### Ιεραρχία - Επίπεδο 5

Το στοιχείο units αποτελείται από τα εξής επιμέρους στοιχεία:



units	
distance	'm,ft,cm,mm,inch'
angle	degree
area	'm2...'
volume	'm3'

Διάγραμμα 11

**Distance:** Είναι το στοιχείο που αναφέρεται στις μονάδες μέτρησης που μπορούν να χρησιμοποιηθούν για τον καθορισμό της απόστασης και λαμβάνει ως περιεχόμενο m, ft, cm, mm, ίντσες.

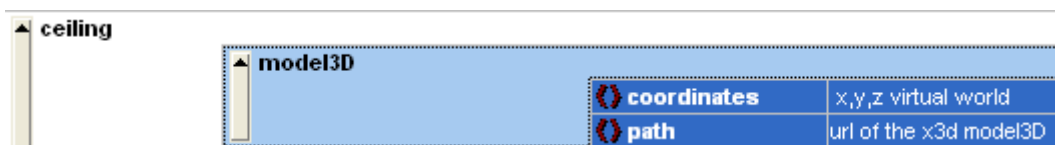
**Angle:** Είναι το στοιχείο που αναφέρεται στις μονάδες μέτρησης που μπορούν να χρησιμοποιηθούν για τον ορισμό της γωνίας και λαμβάνει ως περιεχόμενο βαθμούς γωνία.

**Area:** Είναι το στοιχείο που αναφέρεται στις μονάδες μέτρησης που μπορούν να χρησιμοποιηθούν για τον καθορισμό μιας περιοχής και να λαμβάνει ως περιεχόμενο  $m^2$ .

**Volume:** Είναι το στοιχείο που αναφέρεται στις μονάδες μέτρησης που μπορούν να χρησιμοποιηθούν για τον ορισμό του όγκου των αντικειμένων και λαμβάνει ως περιεχόμενο  $m^3$ .

Μπορούμε να ορίσουμε το σύνολο των ανωτέρω αντικειμένων πάνω από μία φορά. Για παράδειγμα, μπορούμε να ορίσουμε όλους τους τοίχους του δωματίου.

Το στοιχείο `model3D` αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 12

**Coordinates:** Είναι το στοιχείο που αναφέρεται στις συντεταγμένες του wall και έχει ως περιεχόμενο τις συντεταγμένες x, y, z του εικονικού κόσμου, όπου το 3dmodel πρέπει να παρουσιάζεται.

**Path:** Είναι το στοιχείο που αναφέρεται στη διεύθυνση URL του x3d μοντέλου του wall.



Το Material στοιχείο συνίσταται στα εξής επιμέρους στοιχεία:

▲ ceiling	
▲ model3D	
○ coordinates	x,y,z virtual world
○ path	url of the x3d model3D
○ shape	SVG something
○ dimensions	number,number,number-[x],[y],[z] bounding box
○ orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
▲ style	
= styleIndexName	style index name
= styleIndexURL	URL of XML of style Index
abc Text	style code
▲ material	
▲ materialType	
= materialIndexName	material index name
= materialIndexURL	URL of Material Index
abc Text	code
▲ pattern	
= patternIndexName	style index name
= patternIndexURL	URL of XML of pattern Index
abc Text	pattern index ID that expresses the details shapes

Διάγραμμα 13

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού που χρησιμοποιείται για την κατασκευή της οροφής και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το Material στοιχείο έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

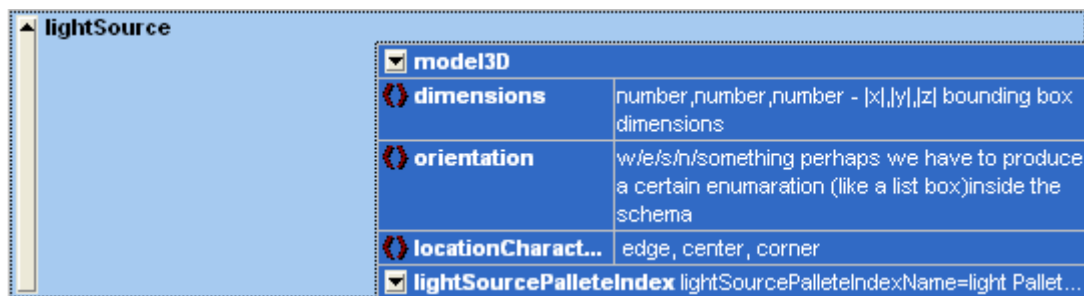
**Pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του προτύπου που χρησιμοποιείται για την κατασκευή της οροφής και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιούνται στο μοντέλο.

Το lightsource στοιχείο αποτελείται από τα εξής επιμέρους στοιχεία:



lightSource	
model3D	
dimensions	number ,number ,number - [x ,y ,z] bounding box dimensions
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box) inside the schema
locationCharact...	edge , center , corner
lightSourcePaletteIndex	lightSourcePaletteIndexName=light Palette...

Διάγραμμα 14

**Model3D:** Είναι το στοιχείο που αναφέρεται στην 3d μοντέλο της πηγής φωτισμού.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις της πηγής φωτισμού, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αναφέρεται στα σημεία προσανατολισμού της πηγής φωτισμού (p / e / s / n).

**LocationCharacterization:** Είναι το στοιχείο που αναφέρεται στη θέση της πηγής φωτισμού στο δωμάτιο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μία πρέπει να επιλέγεται: άκρη, στο κέντρο, γωνία.

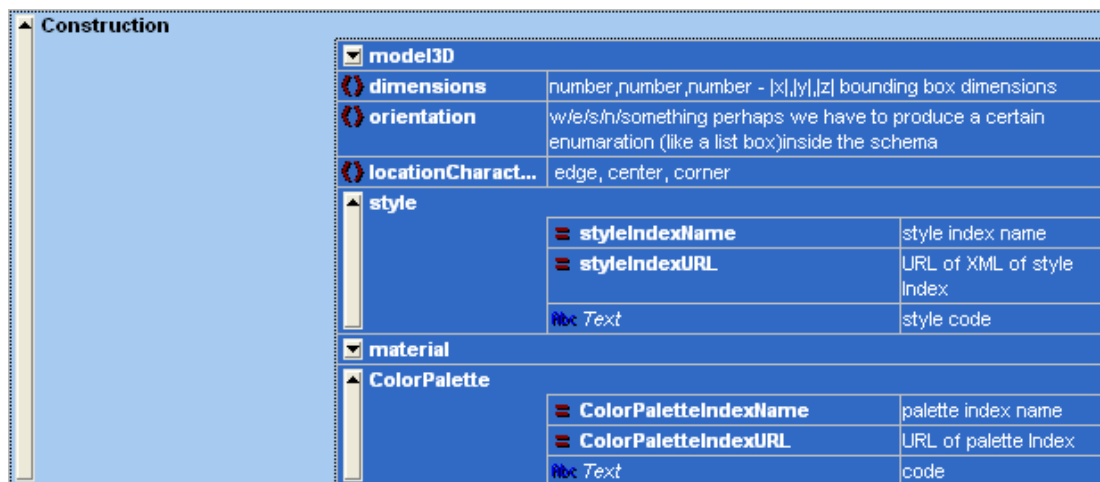
**lightSourcePaletteIndex:** Είναι το στοιχείο που αναφέρεται στο χρώμα του φωτός και έχει δύο χαρακτηριστικά:

**LightSourcePaletteIndexName:** αναφέρεται το όνομα του δείκτη light πηγής palte

LightSourcePaletteIndexURL: παραπέμπει στη διεύθυνση URL της φωτεινής πηγής.

Το lightSourcePaletteIndex στοιχείο έχει ως περιεχόμενο τον αριθμό του δείκτη της φωτεινής πηγής.

Το στοιχείο Construction αποτελείται από τα εξής επιμέρους στοιχεία:



Construction	
model3D	
dimensions	number,number,number - [x],[y],[z] bounding box dimensions
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
locationCharacter...	edge, center, corner
style	
styleIndexName	style index name
styleIndexURL	URL of XML of style index
ibc Text	style code
material	
ColorPalette	
ColorPaletteIndexName	palette index name
ColorPaletteIndexURL	URL of palette Index
ibc Text	code

Διάγραμμα 15

Model3D: Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση του ceiling των κατασκευών.

Dimensions: Είναι το στοιχείο που αναφέρεται στις διαστάσεις της κατασκευής, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

Orientation: Είναι το στοιχείο που αφορά στον προσανατολισμό της κατασκευής (w / e / s / n)

LocationCharacterization: Είναι το στοιχείο που αναφέρεται στη θέση της κατασκευής στο χώρο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μία πρέπει να επιλέγεται: άκρη, στο κέντρο, γωνία.

Style: Είναι το στοιχείο που παραπέμπει στο ύφος της σκάλας. Το στοιχείο στυλ έχει δύο

χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης του το όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο **Style** έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείται.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό της κατασκευής.

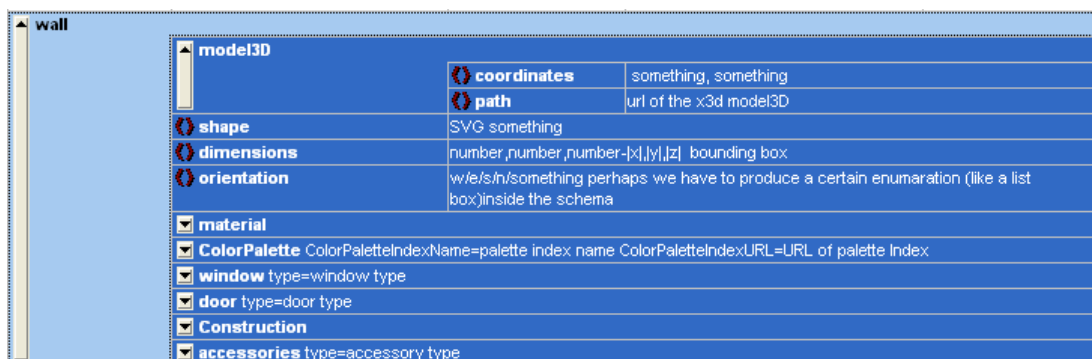
**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για την κατασκευή. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων

Το στοιχείο **ColorPalette** έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το στοιχείο **Model3d** του **wall** αποτελείται από τα εξής επιμέρους στοιχεία:



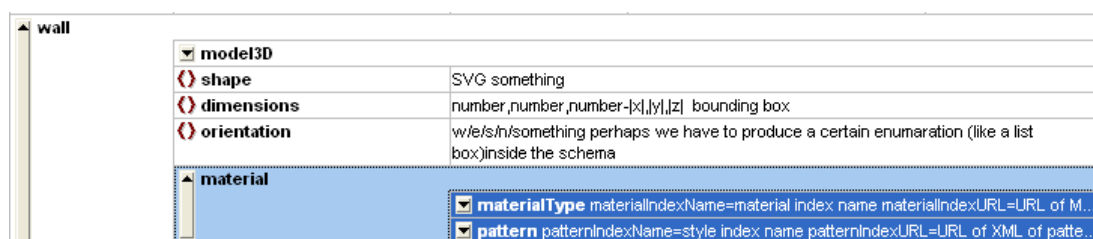
wall	
model3D	
coordinates	something, something
path	url of the x3d model3D
shape	SVG something
dimensions	number ,number ,number -[x],[y],[z] bounding box
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
material	
ColorPalette	ColorPaletteIndexName=palette index name ColorPaletteIndexURL=URL of palette Index
window	type=window type
door	type=door type
Construction	
accessories	type=accessory type

Διάγραμμα 16

**Coordinates:** Είναι το στοιχείο που αναφέρεται στις συντεταγμένες του 3d μοντέλου του τείχου.

**Path:** Είναι το στοιχείο που αναφέρεται στη διεύθυνση URL του x3d προτύπου για τις περιπτώσεις όπου προχωρούμε στην οπτικοποίηση του τείχου.

Το στοιχείο Material του τείχου αποτελείται από τα εξής επιμέρους στοιχεία:



wall	
model3D	
shape	SVG something
dimensions	number ,number ,number -[x , y , z ] bounding box
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
material	
materialType	materialIndexName=material index name materialIndexURL=URL of M...
pattern	patternIndexName=style index name patternIndexURL=URL of XML of patte...

Διάγραμμα 17

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού που χρησιμοποιείται για την κατασκευή του τείχου και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το MaterialType στοιχείο έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

**Pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του προτύπου που χρησιμοποιείται για την κατασκευή του τείχους και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

PatternIndexURL: παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το στοιχείο Pattern λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιούνται στο μοντέλο.

Το στοιχείο window του τείχους έχει ως χαρακτηριστικό το είδος του παραθύρου και αποτελείται από τα εξής επιμέρους στοιχεία:

▼ model3D	
⊗ shape	SVG something
⊗ dimensions	number ,number ,number - x , y , z  bounding box
⊗ orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
▼ material	
▼ ColorPalette	ColorPaletteIndexName=palette index name ColorPaletteIndexURL=URL of palette Index
▲ window	
▢ type	window type
▼ model3D	
⊗ shape	SVG something
⊗ dimensions	number ,number ,number - x , y , z  bounding box
⊗ orientation	something
⊗ locationCharact...	center, corner
▼ style	styleIndexName=style index name styleIndexURL=URL of XML of style Index
▼ material	
▼ frame	

Διάγραμμα 18

Model3D: Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση του παραθύρου.

Shape: Είναι το στοιχείο που αναφέρεται στη μορφή SVG του παραθύρου.

Dimensions: Είναι το στοιχείο που αναφέρεται στις διαστάσεις του παραθύρου, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

Orientation: Είναι το στοιχείο που αναφέρεται στον προσανατολισμό του παραθύρου (w / e / s / n)

**LocationCharacterization:** Είναι το στοιχείο που επικαλείται τη θέση του παραθύρου στο δωμάτιο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μία πρέπει να επιλέγεται: κέντρο, γωνία.

**Style:** Είναι το στοιχείο που αναφέρεται στο στυλ του παραθύρου. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

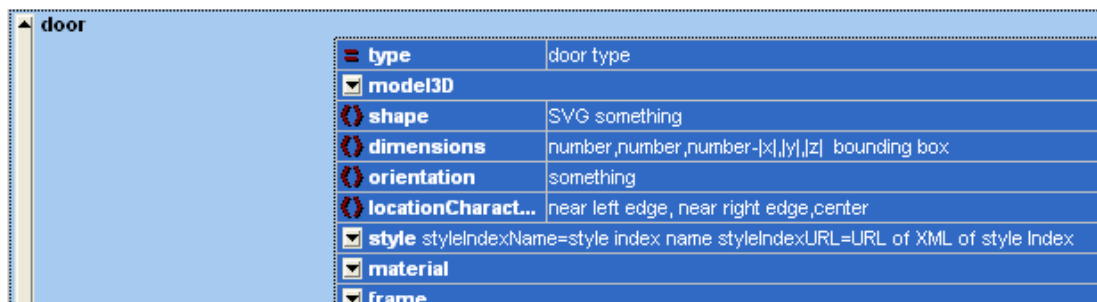
**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο στυλ έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το παράθυρο.

**Frame:** Είναι το στοιχείο που αναφέρεται με το πλαίσιο του παραθύρου.

Το στοιχείο Door του τείχους έχει ως χαρακτηριστικό το είδος της πόρτας και αποτελείται από τα εξής επιμέρους στοιχεία:



door	
type	door type
model3D	
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	something
locationCharact...	near left edge, near right edge,center
style	styleIndexName=style index name styleIndexURL=URL of XML of style Index
material	
frame	

Διάγραμμα 19

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση της πόρτας.

**Shape:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG της πόρτας.

## Συλλογιστική βασισμένη σε περιπτώσεις

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις της πόρτας, υπολογίζεται ως η απόλυτη τιμή των  $x$ ,  $y$ ,  $z$  τιμών.

**Orientation:** Είναι το στοιχείο που αφορά στον προσανατολισμό της θύρας ( $w / e / s / n$ ).

**LocationCharacterization:** Είναι το στοιχείο που επικαλείται τη θέση του παραθύρου σε σχέση με τον τοίχο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μία πρέπει να επιλέγεται: κοντά στο αριστερό άκρο, κοντά στην δεξιά άκρη, στο κέντρο.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος της πόρτας. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

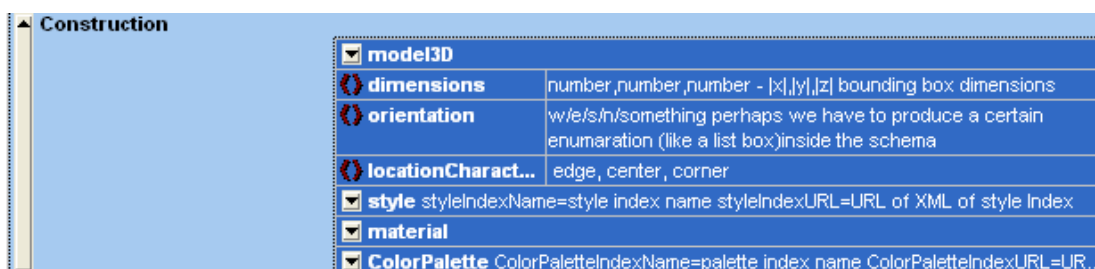
**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο στυλ έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για την πόρτα.

**Frame:** Είναι το στοιχείο που αναφέρεται στο πλαίσιο της πόρτας.

Το στοιχείο Construction του τείχους αποτελείται από τα εξής επιμέρους στοιχεία:



Construction	
<input checked="" type="checkbox"/> <b>model3D</b>	
<input checked="" type="checkbox"/> <b>dimensions</b>	number_number_number - [x],[y],[z] bounding box dimensions
<input checked="" type="checkbox"/> <b>orientation</b>	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
<input checked="" type="checkbox"/> <b>locationCharact...</b>	edge, center, corner
<input checked="" type="checkbox"/> <b>style</b>	styleIndexName=style index name styleIndexURL=URL of XML of style Index
<input checked="" type="checkbox"/> <b>material</b>	
<input checked="" type="checkbox"/> <b>ColorPalette</b>	ColorPaletteIndexName=palette index name ColorPaletteIndexURL=UR...

Διάγραμμα 20



**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση της κατασκευής του τείχους.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις της κατασκευής, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αφορά στον προσανατολισμό της κατασκευής (w / e / s / n).

**LocationCharacterization:** Είναι το στοιχείο που αναφέρεται στη θέση της κατασκευής στο χώρο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μία πρέπει να επιλέγεται: άκρη, στο κέντρο, γωνία.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος της σκάλας. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο Style έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό της κατασκευής.

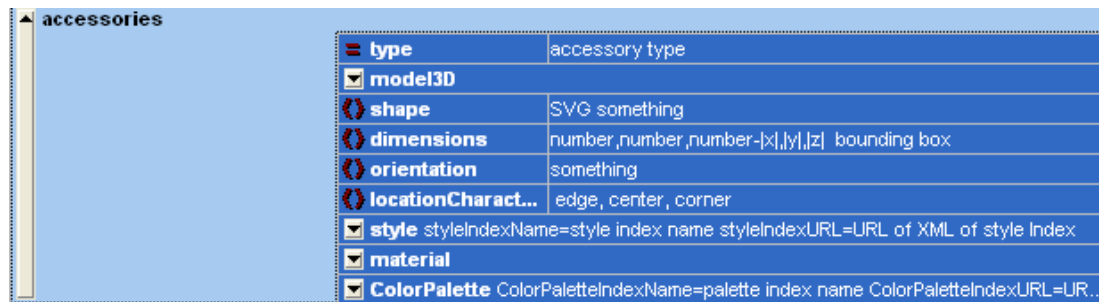
**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για την κατασκευή. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο ColorPalette έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το στοιχείο Accessories του τείχους έχει την ιδιότητα accessory και αποτελείται από τα ακόλουθα επιμέρους στοιχεία:



Property	Value
type	accessory type
model3D	<input checked="" type="checkbox"/>
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	something
locationCharact...	edge, center, corner
style	styleIndexName=style index name styleIndexURL=URL of XML of style index
material	<input checked="" type="checkbox"/>
ColorPalette	<input checked="" type="checkbox"/> ColorPaletteIndexName=palette index name ColorPaletteIndexURL=UR...

Διάγραμμα 21

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση του αξεσουάρ.

**Shape:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG του αξεσουάρ.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις του αξεσουάρ υπολογίζεται, ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αναφέρεται στον προσανατολισμό του εξαρτήματος (w / e / s / n).

**LocationCharacterization:** Θα είναι το στοιχείο που αναφέρεται στη θέση του εξάρτηματος στον τοίχο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες μια πρέπει να επιλέγεται: ακμή, κέντρο, γωνία.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος της σκάλας. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

## Συλλογιστική βασισμένη σε περιπτώσεις

---

Το στοιχείο Style έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

Material: Είναι το στοιχείο που αναφέρεται στην ύλη του αξεσουάρ.

ColorPalette: Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για το αξεσουάρ. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

ColorPaletteIndexName:: Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

ColorPaletteIndexURL: Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο ColorPalette έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το στοιχείο model3D του δαπέδου αποτελείται από τα εξής επιμέρους στοιχεία:



▲ floor	
▲ model3D	
coordinates	something, something
path	url of the x3d model3D

Διάγραμμα 22

Coordinates: Είναι το στοιχείο που αναφέρεται στην συντεταγμένες του δαπέδου και στο περιεχόμενο που καθορίζει το σύστημα συντεταγμένων.

Path: Είναι το στοιχείο που αναφέρεται στη διεύθυνση URL του x3d προτύπου για τις περιπτώσεις όπου προχωρούμε στην οπτικοποίηση του δαπέδου.

Το στοιχείο material του δαπέδου αποτελείται από τα εξής επιμέρους στοιχεία:

▲ floor	▼ model3D
	○ shape SVG something
	○ dimensions number,number,number- x , y , z  bounding box
	○ orientation something
	▼ style styleIndexName=style index name styleIndexURL=URL of XML of style Index
	▲ material
	▼ materialType materialIndexName=material index name mat...
	▼ pattern patternIndexName=style index name patternIndexU...

Διάγραμμα 23

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού που χρησιμοποιείται για την κατασκευή του δαπέδου και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το στοιχείο **MaterialType** έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

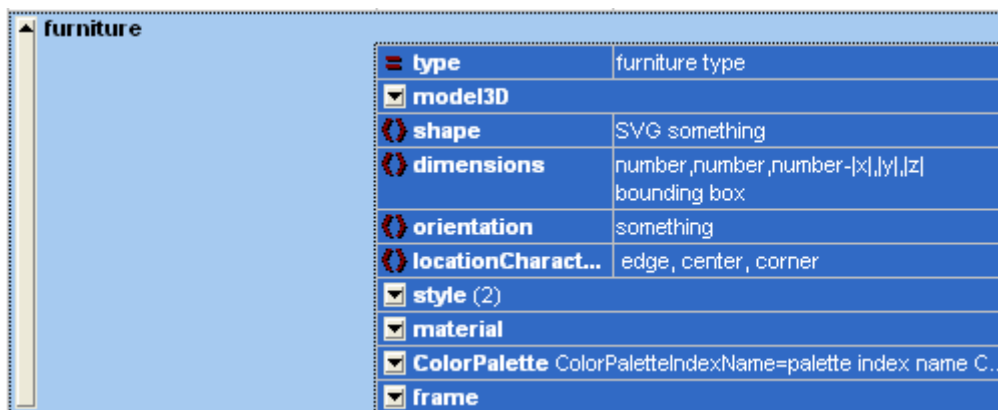
**Pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του προτύπου που χρησιμοποιείται για την κατασκευή του δαπέδου και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το **pattern** στοιχείο λαμβάνει ως περιεχόμενο του το δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιούνται στο μοντέλο.

Το στοιχείο **Furniture** του ορόφου έχει το χαρακτηριστικό **FurnitureType** και αποτελείται από τα ακόλουθα επιμέρους στοιχεία:



Property	Value
type	furniture type
model3D	
shape	SVG something
dimensions	number ,number ,number - x , y , z  bounding box
orientation	something
locationCharacter...	edge , center , corner
style (2)	
material	
ColorPalette	ColorPaletteIndexName=palette index name C...
frame	

Διάγραμμα 24

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση των επίπλων.

**Shape:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG των επίπλων.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις των επίπλων, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αναφέρεται στον προσανατολισμό των επίπλων (w / e / s / n).

**LocationCharacterization:** Είναι το στοιχείο που αναφέρεται στη θέση των επίπλων στο χώρο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες ένα πρέπει να επιλεγεί: ακμή, κέντρο, γωνία.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος των επίπλων. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο Style έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για τα έπιπλα.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για τα έπιπλα. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

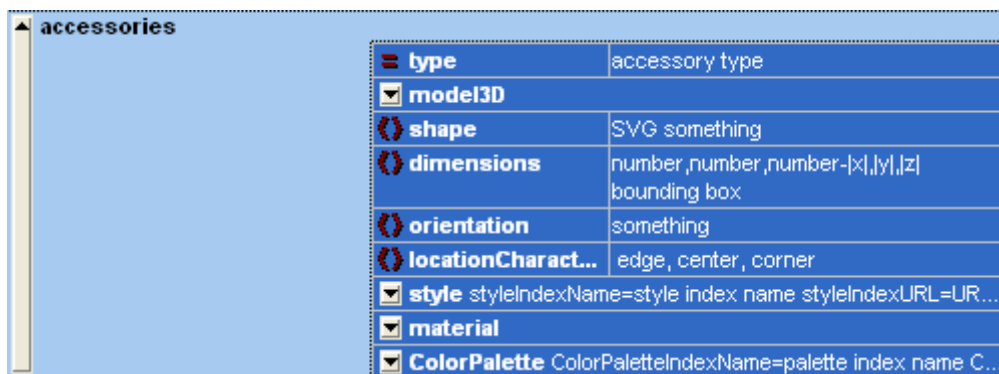
**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο **ColorPalette** έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

**Frame:** Είναι το στοιχείο που αναφέρεται στο πλαίσιο των επίπλων.

Το στοιχείο **Accessories** του δαπέδου έχει την ιδιότητα **Accessory type** και αποτελείται από τα ακόλουθα επιμέρους στοιχεία:



Property	Value
<b>type</b>	accessory type
<b>model3D</b>	
<b>shape</b>	SVG something
<b>dimensions</b>	number ,number ,number - x , y , z  bounding box
<b>orientation</b>	something
<b>locationCharact...</b>	edge , center , corner
<b>style</b>	styleIndexName=style index name styleIndexURL=UR...
<b>material</b>	
<b>ColorPalette</b>	ColorPaletteIndexName=palette index name C...

Διάγραμμα 25

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση του αξεσουάρ.

**Style:** Είναι το στοιχείο που αναφέρεται στη μορφή SVG του αξεσουάρ.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις του αξεσουάρ υπολογίζεται, ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αναφέρεται στον προσανατολισμό του αξεσουάρ (w / e / s / n).

**LocationCharacterization:** Θα είναι το στοιχείο που αναφέρεται στη θέση του αξεσουάρ στον τοίχο και έχει ως περιεχόμενο τρεις τιμές από τις οποίες μια πρέπει να επιλέγεται: ακμή, κέντρο, γωνία.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος της σκάλας. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο στυλ έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στην ύλη του αξεσουάρ.

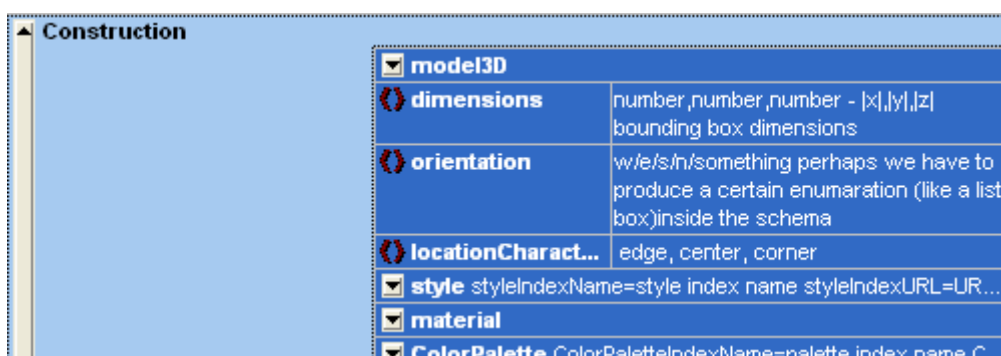
**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για το αξεσουάρ. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων

Το στοιχείο ColorPalette έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το στοιχείο Construction του δαπέδου αποτελείται από τα εξής επιμέρους στοιχεία:



Construction	
model3D	
dimensions	number,number,number - [x],[y],[z] bounding box dimensions
orientation	w/e/s/n/something perhaps we have to produce a certain enumeration (like a list box)inside the schema
locationCharact...	edge, center, corner
style	styleIndexName=style index name styleIndexURL=UR...
material	
ColorPalette	ColorPalettelIndexName=palette.index.name C...

Διάγραμμα 26

**Model3D:** Είναι το στοιχείο που αναφέρεται στο 3D μοντέλο που μπορεί να χρησιμοποιηθεί για την απεικόνιση της κατασκευής δαπέδου.

**Dimensions:** Είναι το στοιχείο που αναφέρεται στις διαστάσεις της κατασκευής, υπολογίζεται ως η απόλυτη τιμή των x, y, z τιμών.

**Orientation:** Είναι το στοιχείο που αφορά στον προσανατολισμό της κατασκευής (w / e / s / n).

**LocationCharacterization:** Είναι το στοιχείο που αναφέρεται στη θέση της κατασκευής στο πάτωμα και έχει ως περιεχόμενο τρεις τιμές από τις οποίες η μια πρέπει να επιλεγεί: άκρη, στο κέντρο, γωνία.

**Style:** Είναι το στοιχείο που παραπέμπει στο ύφος της σκάλας. Το στοιχείο στυλ έχει δύο χαρακτηριστικά:

**StyleIndexName:** Είναι ο δείκτης στο όνομα του στυλ που χρησιμοποιείται.

**StyleIndexUrl:** Είναι το URL του χρησιμοποιούμενου στυλ.

Το στοιχείο στυλ έχει ως περιεχόμενο τον κωδικό του στυλ που χρησιμοποιείτε.

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό της κατασκευής.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για την κατασκευή. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:



**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο **ColorPalette** έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το στοιχείο **model3d** της σκάλας αποτελείται από τα εξής επιμέρους στοιχεία:

▲ stair	
type	stair type
▲ model3D	
coordinates	something, something
path	url of the x3d model

Διάγραμμα 27

**Coordinates:** Είναι το στοιχείο που αναφέρεται στις συντεταγμένες του μοντέλου της σκάλας και καθορίζει το σύστημα συντεταγμένων.

**Path:** Είναι το στοιχείο που αναφέρεται στη διεύθυνση URL του x3d προτύπου για τις περιπτώσεις όπου προχωρούμε στην οπτικοποίηση της σκάλας.

Το **Material** στοιχείο της σκάλας αποτελείται από τα εξής επιμέρους στοιχεία:

▲ stair	
type	stair type
▼ model3D	
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	something
locationCharacterization	edge, center, corner
▼ style styleIndexName=style index name styleIndexURL=URL of XML of style Index	
▲ material	
materialType	materialIndexName=material index name mat...
pattern	patternIndexName=style index name patternIndexU...

Διάγραμμα 28

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού που χρησιμοποιείται για την κατασκευή της σκάλας και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Ο τύπος υλικού στοιχείου έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

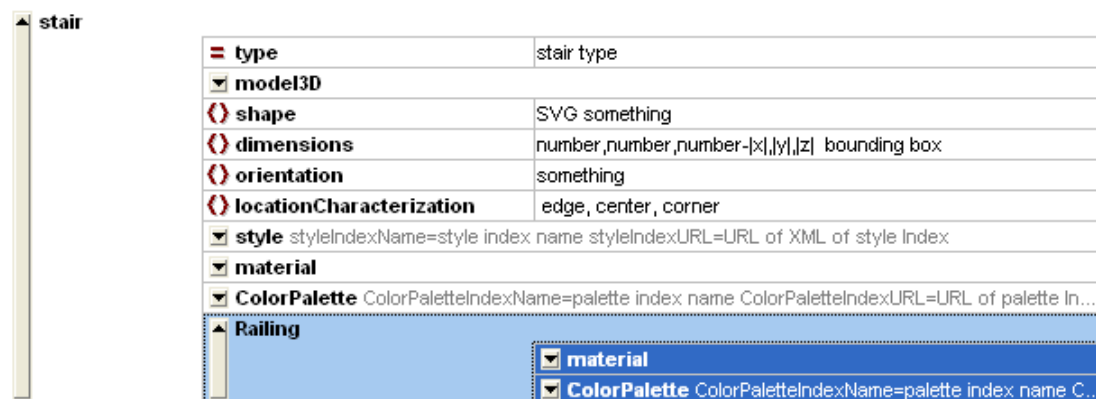
**Pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του προτύπου που χρησιμοποιείται για την κατασκευή της σκάλας και έχει δύο χαρακτηριστικά.

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιούνται στο μοντέλο.

Το στοιχείο Railing της σκάλας αποτελείται από τα εξής επιμέρους στοιχεία:



stair	
type	stair type
model3D	
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	something
locationCharacterization	edge, center, corner
style	styleIndexName=style index name styleIndexURL=URL of XML of style index
material	
ColorPalette	ColorPaletteIndexName=palette index name ColorPaletteIndexURL=URL of palette In...
Railing	
material	
ColorPalette	ColorPaletteIndexName=palette index name C...

Διάγραμμα 29

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού που χρησιμοποιείται για την κατασκευή των κιγκλιδωμάτων και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το στοιχείο **Material type** έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για τα κάγκελα. Το στοιχείο **ColorPalette** μπορεί να περιλαμβάνει περισσότερα από ένα χρώματα, δεδομένου ότι ένα δωμάτιο μπορεί να έχει ένα συνδυασμό χρωμάτων. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο **ColorPalette** έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

### **Ιεραρχία - Επίπεδο 6**

Το στοιχείο **Material** του παραθύρου αποτελείται από τα εξής επιμέρους στοιχεία:

type	window type
model3D	
shape	SVG something
dimensions	number_number_number-[x ,y ,z] bounding box
orientation	something
locationCharacter...	center, corner
style	styleIndexName=style index name styleIndexURL=URL of XML of style Index
material	
materialType	
materialIndexName	material index name
materialIndexURL	URL of Material Inde
ribc Text	code
pattern	
patternIndexName	style index name
patternIndexURL	URL of XML of patte
ribc Text	pattern index ID that
	expresses the detail
	shapes

Διάγραμμα 30

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού του παραθύρου και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το στοιχείο **materialType** έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

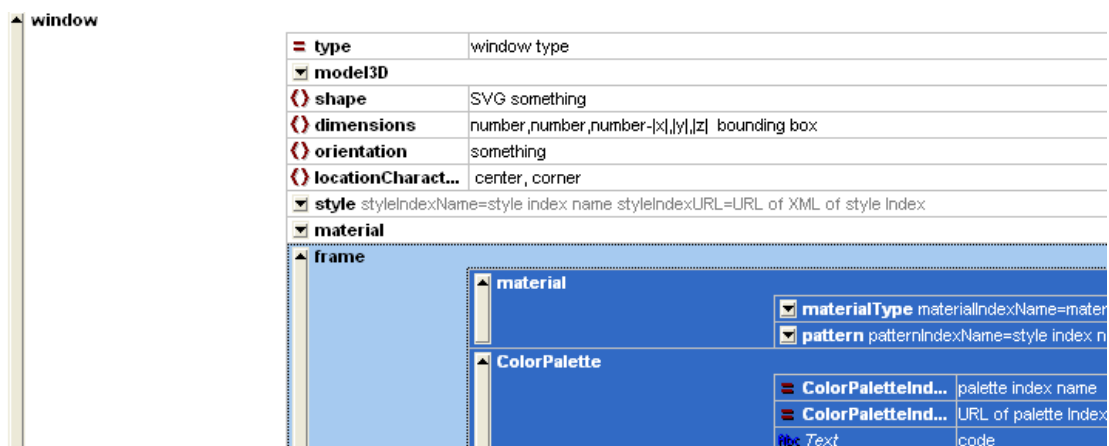
**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου του παραθύρου (π.χ. vitro) και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το **pattern** στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το στοιχείο **Frame** του παραθύρου αποτελείται από τις ακόλουθες επιμέρους στοιχεία:



Διάγραμμα 31

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το πλαίσιο του παραθύρου.

**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για το πλαίσιο του παραθύρου. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο **ColorPalette** έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το στοιχείο **Frame** της θύρας αποτελείται από τα εξής επιμέρους στοιχεία:

door	
type	door type
model3D	
shape	SVG something
dimensions	number,number,number- x , y , z  bounding box
orientation	something
locationCharacter...	near left edge, near right edge,center
style	styleIndexName=style index name styleIndexURL=URL of XML of style Index
material	
materialType	
materialIndexNa...	material index name
materialIndexURL	URL of Material Ind
code	code
pattern	
patternIndexNa...	style index name
patternIndexURL	URL of XML of patt
pattern index ID the	expresses the deta
shapes	shapes

Διάγραμμα 32

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού από της πόρτας και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το στοιχείο **materialType** έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

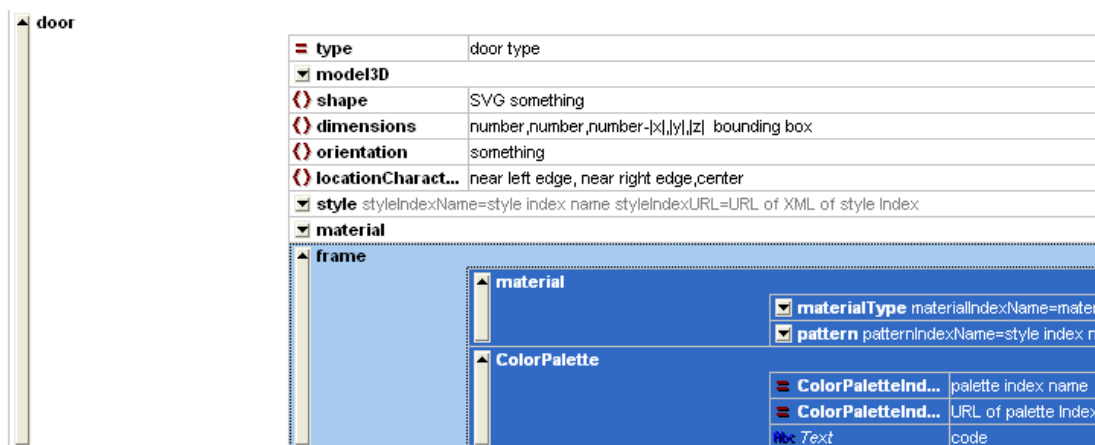
**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου της πόρτας και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το **pattern** στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το στοιχείο **Frame** της θύρας αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 33

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το πλαίσιο της πόρτας

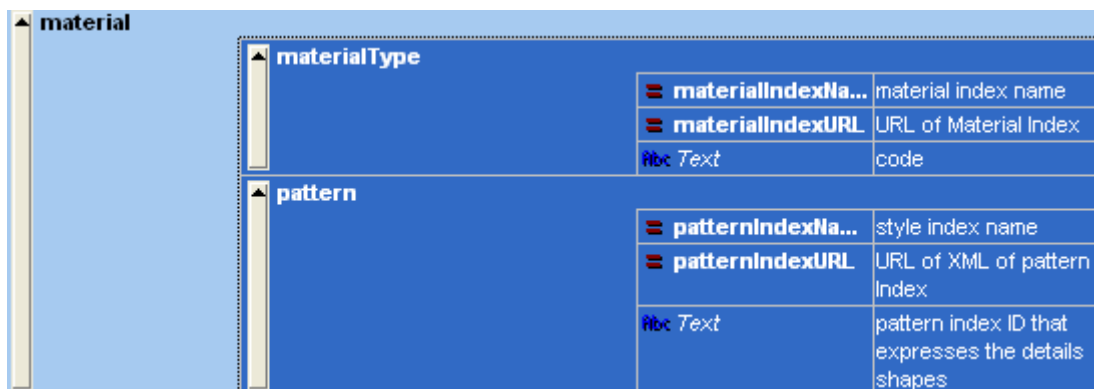
**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για το πλαίσιο της πόρτας. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το στοιχείο **ColorPalette** έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το **Material** στοιχείο της κατασκευής του τείχους αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 34

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού για την κατασκευή του τείχους και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών

Το στοιχείο **materialType** έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου της κατασκευής του τείχους και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το **pattern** στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το **material** στοιχείο της επίπλωσης αποτελείται από τα εξής επιμέρους στοιχεία:



type	furniture type
model3D	
shape	SVG something
dimensions	number,number,number-[x],[y],[z] bounding box
orientation	something
locationCharact...	edge, center, corner
style (2)	
material	
materialType	
materialIndexNa...	material index name
materialIndexURL	URL of Material Index
pattern	
patternIndexNa...	style index name
patternIndexURL	URL of XML of pattern index

Διάγραμμα 35

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού των επίπλων και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το στοιχείο **materialType** έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου των επίπλων και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το **pattern** στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το Frame στοιχείο της επίπλωσης αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 36

**Material:** Είναι το στοιχείο που αναφέρεται στο υλικό που χρησιμοποιείται για το πλαίσιο των επίπλων.

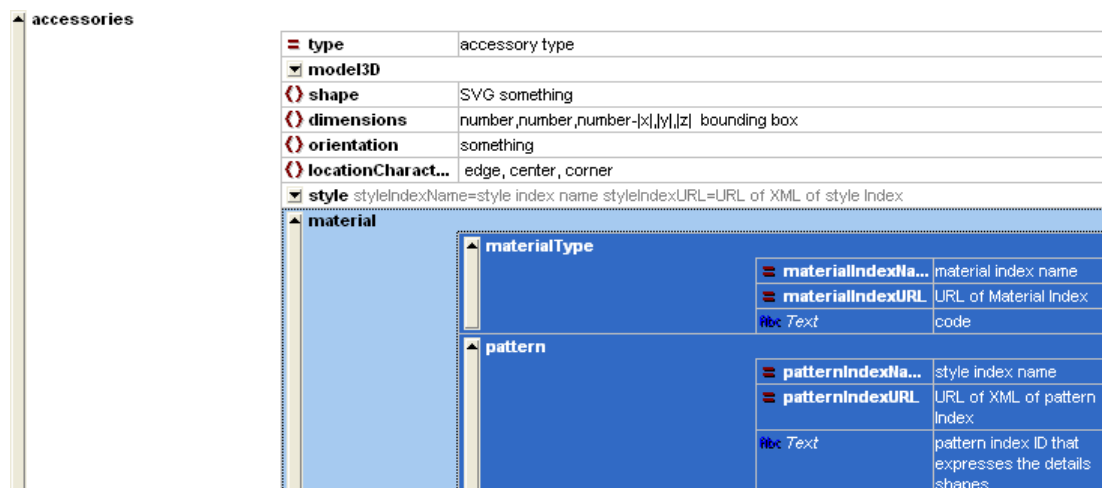
**ColorPalette:** Είναι το στοιχείο που αναφέρεται στην παλέτα χρωμάτων που χρησιμοποιείται για τον σκελετό του επίπλου. Το στοιχείο αυτό έχει επίσης 2 χαρακτηριστικά:

**ColorPaletteIndexName::** Είναι ο δείκτης του ονόματος που χρησιμοποιείται στη χρωματική παλέτα.

**ColorPaletteIndexURL:** Είναι το URL της παλέτας χρωμάτων.

Το ColorPalette στοιχείο έχει ως περιεχόμενο τον κωδικό της παλέτας χρωμάτων που χρησιμοποιούνται.

Το Material στοιχείο του αξεσουάρ περιλαμβάνει τα ακόλουθα επιμέρους στοιχεία:



Διάγραμμα 37

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού του αξεσουάρ και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού  
**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το materialType στοιχείο έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

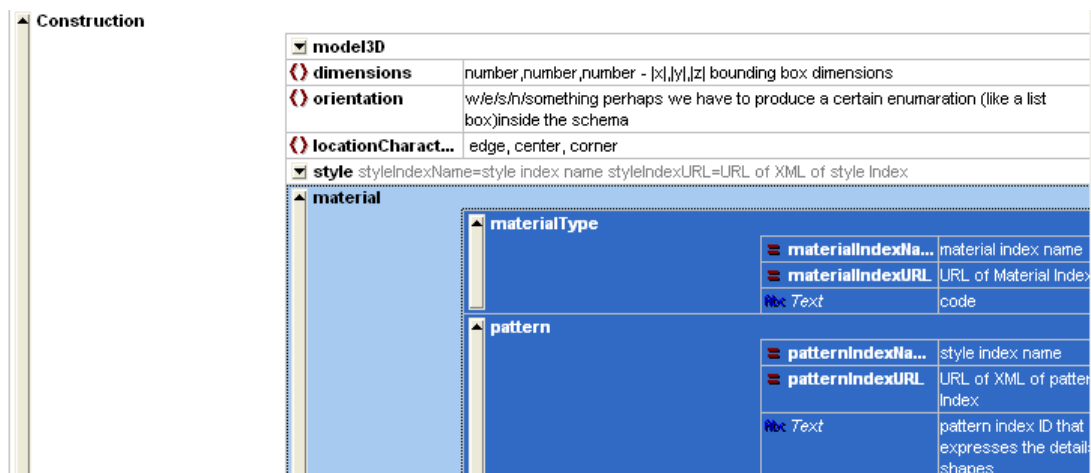
**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου των αξεσουάρ και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το material στοιχείο της κατασκευής του δαπέδου αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 38

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού κατασκευής του δαπέδου και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το materialType στοιχείο έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

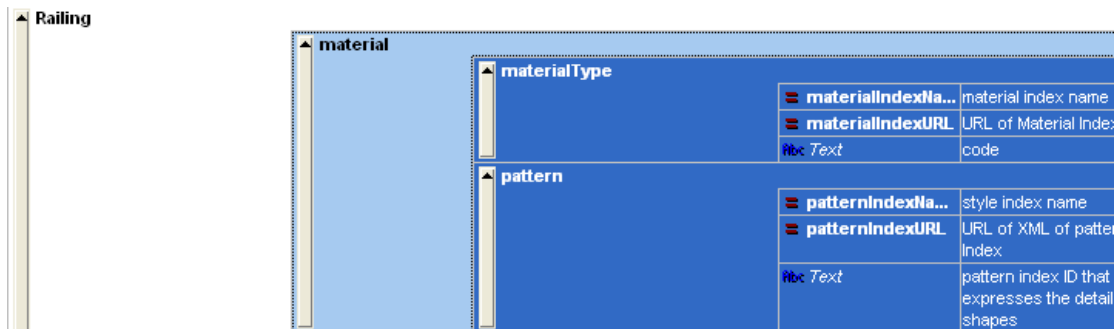
**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου της κατασκευής του δαπέδου και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το material στοιχείο του κιγκλιδώματος αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 39

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού του κιγκλιδώματος και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το materialType στοιχείο έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου των κιγκλιδωμάτων και έχει δύο χαρακτηριστικά:

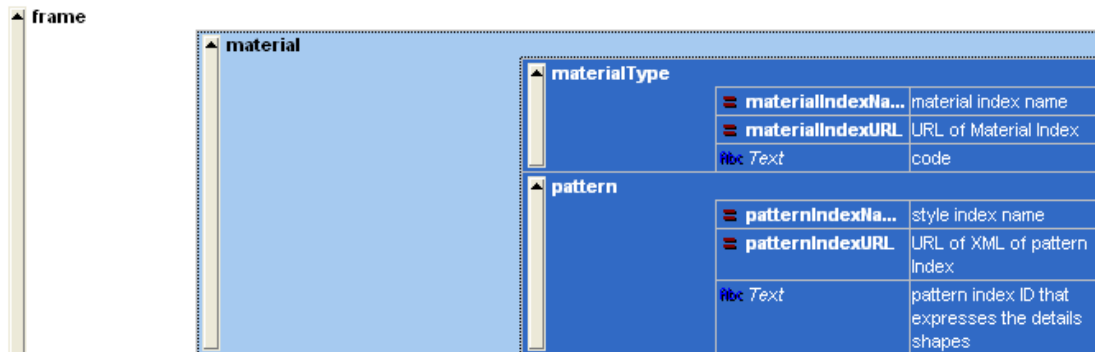
**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

## Ιεραρχία - Επίπεδο 7

Το material στοιχείο του πλαισίου του παραθύρου αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 40

**MaterialType:** Είναι το στοιχείο που αναφέρεται στο είδος του υλικού του πλαισίου του παραθύρου και έχει δύο χαρακτηριστικά:

**MaterialIndexName:** αναφέρεται το όνομα δείκτη του υλικού.

**MaterialIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το **materialType** στοιχείου έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

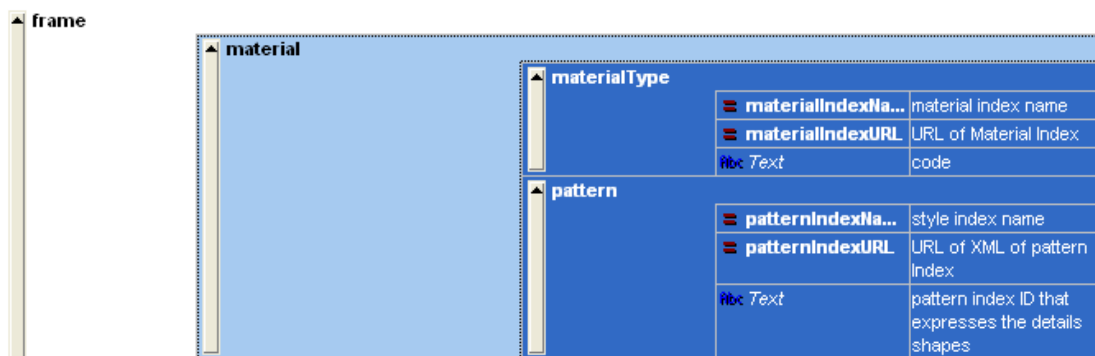
**pattern:** Είναι το στοιχείο που αναφέρεται στο είδος του τύπου του πλαισίου του παραθύρου και έχει δύο χαρακτηριστικά:

**PatternIndexName:** αναφέρεται το όνομα δείκτη του προτύπου.

**PatternIndexURL:** παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.

Το material στοιχείο του πλαισίου της πόρτας αποτελείται από τα εξής επιμέρους στοιχεία:



Διάγραμμα 41

MaterialType: Είναι το στοιχείο που αναφέρεται στο είδος του υλικού του πλαισίου της πόρτας και έχει δύο χαρακτηριστικά:

MaterialIndexName: αναφέρεται το όνομα δείκτη του υλικού.

MaterialIndexURL: παραπέμπει στη διεύθυνση URL του δείκτη υλικών.

Το materialType στοιχείο έχει ως περιεχόμενο τον κωδικό του υλικού που χρησιμοποιείται.

pattern: Είναι το στοιχείο που αναφέρεται στο είδος του τύπου του πλαισίου της πόρτας και έχει δύο χαρακτηριστικά:

PatternIndexName: αναφέρεται το όνομα δείκτη του προτύπου.

PatternIndexURL: παραπέμπει στη διεύθυνση URL του δείκτη προτύπου.

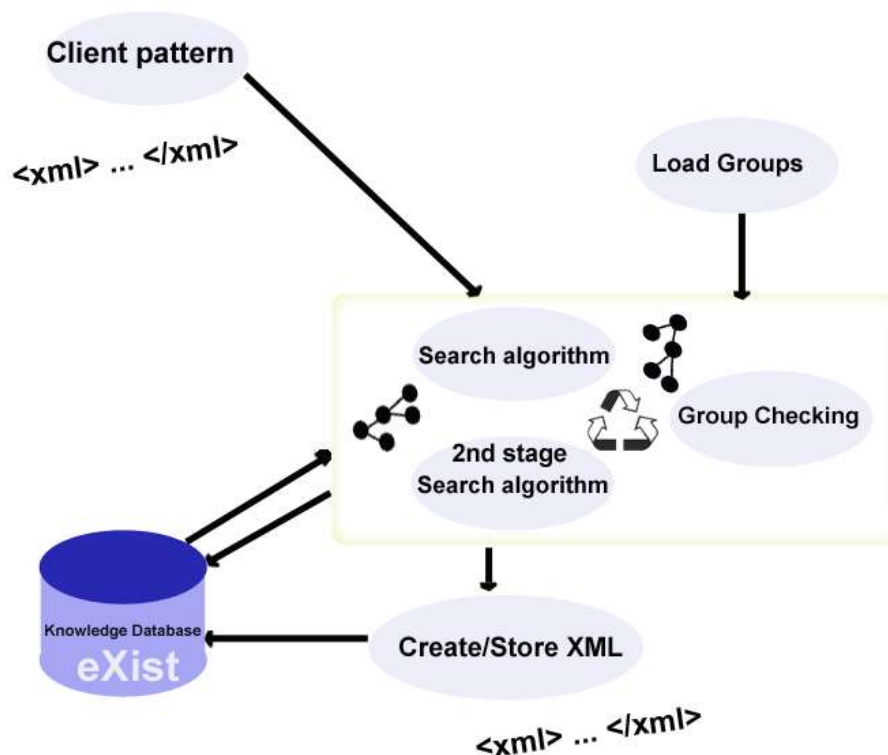
Το pattern στοιχείο λαμβάνει ως περιεχόμενο του δείκτη αναγνωριστικό του προτύπου που εκφράζει τις λεπτομέρειες των σχημάτων που χρησιμοποιείται στο μοντέλο.



## Κεφάλαιο 4

### Τεχνική Περιγραφή Του Αλγορίθμου

## 4.1 Αναλυτική παρουσίαση κλάσεων και μεθόδων αλγορίθμου



Εικόνα 4.1.1 Σχηματική αναπαράσταση συστήματος.

### 4.1.1 Ανάλυση της SecondPanel

Το σύστημα μας ξεκινάει με ένα αλγόριθμο αναζήτησης δέντρων, ο οποίος έχει αναληθεί σε προηγούμενη πτυχακή εργασία, οπότε και δε θα τον αναλύσουμε περαιτέρω εδώ. Εδώ μας ενδιαφέρει η επέκταση αυτού του αλγορίθμου σε ένα ολοκληρωμένο σύστημα τεχνητής νοημοσύνης το οποίο αποφασίζει βάση προηγούμενων περιπτώσεων.

Στη συνέχεια λοιπόν στο τέλος της αναζήτησης εκτελείται η κλάση second panel η οποία είναι μια φόρμα και είναι υπεύθυνη για το εξαντλητικό ψάξιμο κόμβων του

δέντρου, οι οποίοι δεν βρέθηκαν δε προηγούμενες αναζητήσεις. Η κλάση αυτή καλείται τόσες φορές όσοι είναι και οι κόμβοι που δεν βρέθηκαν. Η διαφορά της σε σχέση με την κλάση `matchingApp` είναι ότι πλέον αναζητείται συγκριμένο υποδέντρο διαφορετικό κάθε φορά ανάλογα με το `root` με το οποίο «ταΐζουμε» τον αλγόριθμο αναζήτησης.

Ένας σημαντικός έλεγχος που γίνεται εδώ με τη μέθοδο `CheckVec` είναι να μην μπαίνουν προς αναζήτηση διαδρομές οι οποίες ήδη έχουν αναζητηθεί και δεν βρέθηκαν. Έτσι αποφεύγεται η περίπτωση του ατέρμονου βρόγχου, όπου διαδρομές που είναι προς αναζήτηση αφού αναζητηθούν και δε βρεθούν να μπαίνουν ξανά προς αναζήτηση με αποτέλεσμα ποτέ να μην βρίσκονται και πάντα να ψάχνονται.

Αυτό που κάνει ουσιαστικά η `CheckVec` είναι να κρατάει το διάνυσμα αναζήτησης που ταΐζει την `Rse` και να το συγκρίνει κάθε φορά με αυτό το οποίο παράγεται από αυτή. Αν αυτά τα δύο είναι ίδια τότε το δεύτερο διαγράφεται μιας και αν αφηθεί διατρέχει ο κίνδυνος του ατέρμονος βρόγχου.

Καλείται λοιπόν, σε πρώτη φάση, η μέθοδος `Research2` η οποία κρατάει τις διαδρομές που είναι να αναζητηθούν και «ταΐζει» με αυτές την `Rse` η οποία αναλαμβάνει να τις ψάχνει.

Μόλις κάποιο διάνυσμα της `Rse` γεμίσει με διαδρομές τότε γίνεται ο έλεγχος των `Groups` καλώντας την κλάση `GroupCheck`. Ο έλεγχος αυτός ομαδοποιεί κάποιες διαδρομές και επιβάλλει αυτές να είναι ομοιογενείς. Αυτό σημαίνει πως αν κάποια διαδρομή περιγράφει για παράδειγμα μια τιμή αλλά στο `tag` της περιέχει κάποια `attributes` τότε πρέπει αυτά τα `attributes` να είναι του τύπου αυτής της διαδρομής. Είναι ένα είδος κανόνων για να παρέχετε ποιο ποιοτικό τελικό αποτέλεσμα. Στην κλάση `GroupCheck` θα γίνει εκτενέστερη αναφορά πιο κάτω.

Όταν μια αναζητούμενη διαδρομή βρεθεί τότε ο αλγόριθμος βρίσκει και την τιμή αυτής της διαδρομής και την προσκολλά στο τέλος χωρίζοντας τις δύο πληροφορίες με «!-!». Εν συνεχεία η διαδρομή επιστρέφεται για να μπει στο υποψήφιο για δημιουργία `xml` εγγράφου διάνυσμα μαζί με τις υπόλοιπες διαδρομές που προορίζονται να το απαρτίσουν. Οι διαδρομές αυτές χωρίζονται μεταξύ τους με «,,» για να είναι δυνατή η επεξεργασία τους σε κάθε φάση του αλγορίθμου όπου και όταν αυτό κριθεί αναγκαίο.

Στο τέλος τις Rse και εφόσον έχουν δημιουργηθεί όλα τα υπονήφια διανύσματα υπολογίζεται το καλύτερο από αυτά το οποίο και προστίθεται σε εκείνο της προηγούμενης αναζήτησης. Υπολογίζεται επίσης και το διάνυσμα με τις υπολειπόμενες διαδρομές που θα περάσει στο επόμενο στάδιο.

### 4.1.2 Ανάλυση της GroupCheck

Η GroupCheck είναι η μέθοδος η οποία είναι υπεύθυνη να πάρει το εκάστοτε διάνυσμα διαδρομών που έχει σχηματιστεί και να επιβάλλει τον κανόνα ομαδοποίησης της. Παίρνει λοιπόν ως είσοδο το υπονήφιο διάνυσμα και φορτώνει το αρχείο με τους κανόνες, διαβάσει το αρχείο αυτό και ψάχνει το διάνυσμα διαδρομών για να δει αν συμβαδίζει με τους κανόνες. Αν κάποιες διαδρομές είναι διαφορετικές από ότι ορίζετε στους κανόνες της τότε οι διαδρομές αυτές διαγράφονται από το διάνυσμα διαδρομών. τελικά το διάνυσμα επιστρέφεται για περαιτέρω επεξεργασία.

### 4.1.3 Ανάλυση της GettingTheXml

Η GettingTheXml κλάση είναι αυτή που είναι υπεύθυνη να διαβάσει το αρχείο των groups και να το φορτώσει σωστά σε ένα διάνυσμα. Με τον όρο σωστά εννοούμε ότι τα groups φορτώνονται στο διάνυσμα σε ομάδες ανά θέση.

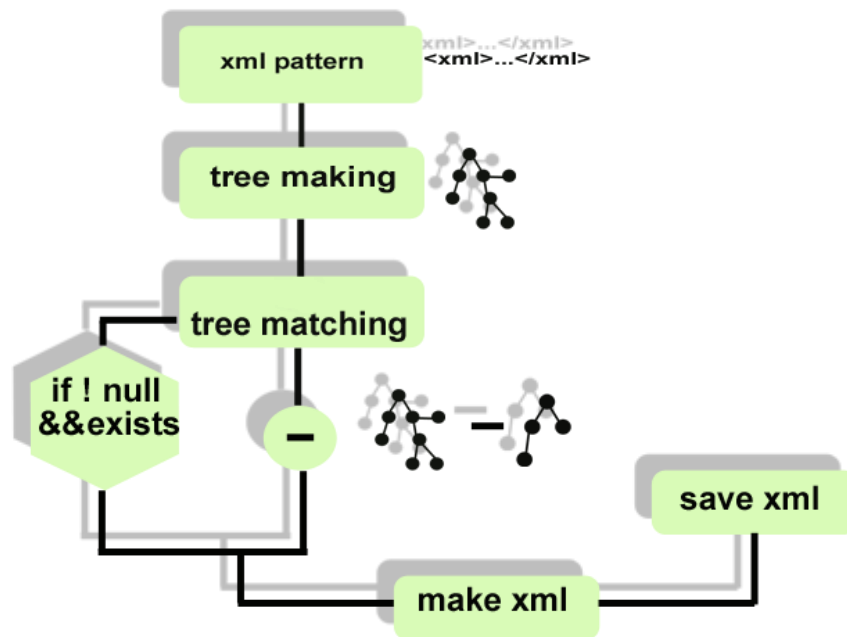
### 4.1.4 Ανάλυση της XmlCreator

Η κλάση αυτή με τη modifyDocument μέθοδο, παίρνει από την matchingApp σαν είσοδο το τελικό διάνυσμα διαδρομών. Φορτώνει από την xml βάση, στη μνήμη, το null xml και αρχίζει να το διαβάσει και να το συμπληρώνει βάση των τιμών των διαδρομών. Παράγει έξοδο ένα xml έγγραφο και τροφοδοτεί με αυτό την store κλάση.

Το null xml έγγραφο είναι ένα xml έγγραφο του οποίου η δομή είναι πανομοιότυπη με εκείνη των υπολοίπων εγγράφων δωματίων, απλά όπου υπάρχει τιμή ή ακόμα και attribute αυτή έχει αλλαχθεί και στη θέση της έχει μπει μια τιμή την οποία ο αλγόριθμός μας αναγνωρίζει ως null.

## 4.1.5 Ανάλυση της Store

Η κλάση Store κάνει αυτό που λέει ουσιαστικά και το όνομα της δηλαδή παίρνει ως είσοδο ένα xml έγγραφο και αναλαμβάνει να το αποθηκεύσει στη xml βάση δεδομένων αφού πρώτα του δώσει ένα τυχαίο όνομα ώστε να μην συμπέσει με το όνομα κάποιου άλλου από τα έγγραφα που είναι ήδη αποθηκευμένα εκεί.



Εικόνα 4.1.2 flowchart διεργασιών



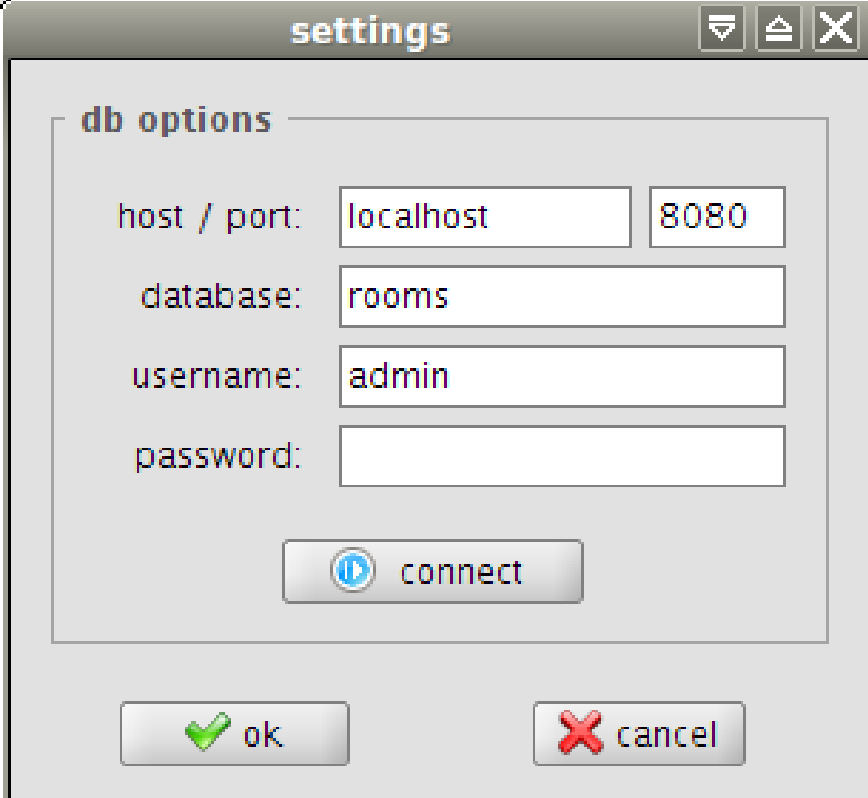


## Κεφάλαιο 5

### Περιγραφή της διεπαφής



## 5.1 Παράθυρο ρυθμίσεων



The image shows a 'settings' dialog box with the following fields and controls:

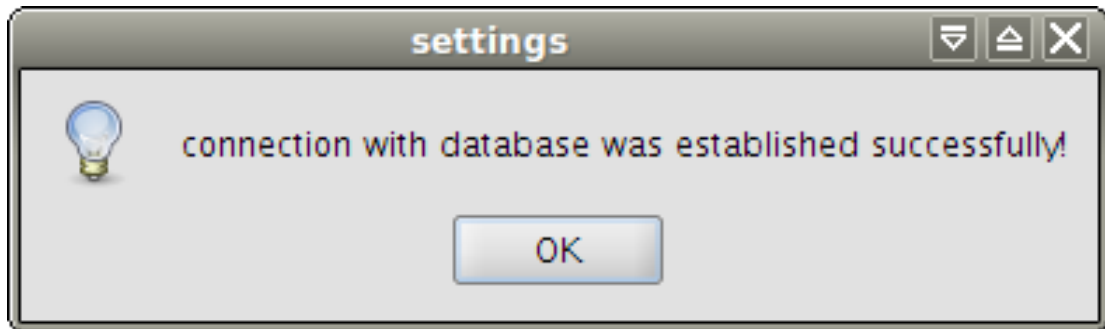
- host / port: localhost 8080
- database: rooms
- username: admin
- password: (empty)
- connect button (with play icon)
- ok button (with checkmark icon)
- cancel button (with X icon)

Εικόνα 5.1.1 settings form

Το παράθυρο ρυθμίσεων (settings) εμφανίζεται αυτόματα κατά την πρώτη εκτέλεση της εφαρμογής και μας ζητά να ορίσουμε τα απαραίτητα στοιχεία για την εγκαθίδρυση της σύνδεσης με τη βάση γνώσης που χρησιμοποιεί η εφαρμογή. Τα στοιχεία αυτά είναι:

- Το όνομα και η πόρτα του ηλεκτρονικού υπολογιστή στην οποία τρέχει ο server της βάσης.
- Το όνομα της βάσης γνώσης.
- Το όνομα του χρήστη μέσω του οποίου θα γίνει η σύνδεση (ο οποίος θα πρέπει να έχει δικαιώματα εγγραφής και ανάγνωσης πάνω στη βάση).

Και τέλος ο κωδικός του χρήστη αυτού.



Εικόνα 5.1.2 connection established

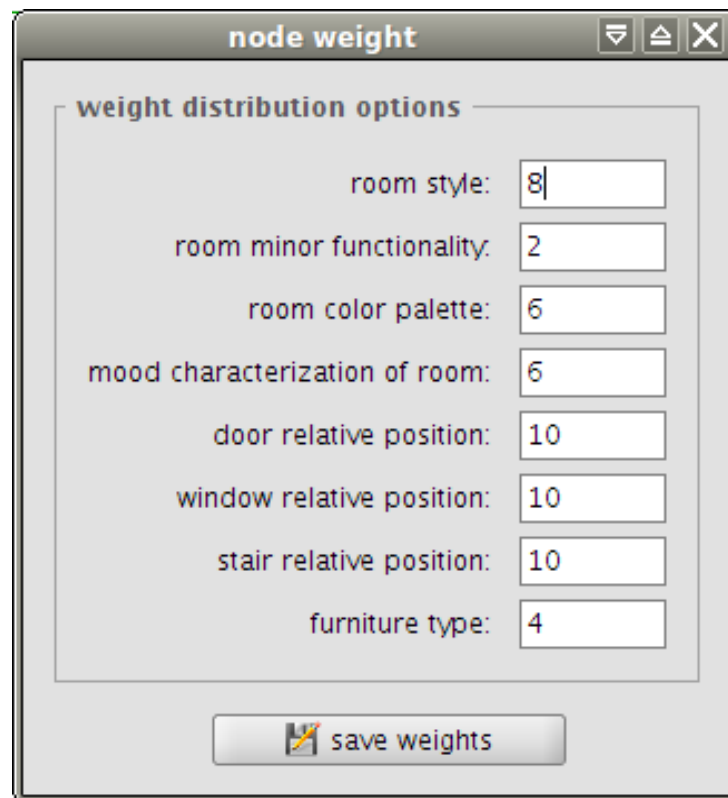
Αν όλα τα στοιχεία συμπληρωθούν σωστά τότε εμφανίζεται το παραπάνω μήνυμα το οποίο ενημερώνει για την ορθότητα των δεδομένων που πληκτρολογήθηκαν από το χρήστη και για να μας ενημερώσει ότι η σύνδεση με τη βάση μας έχει πραγματοποιηθεί.



Εικόνα 5.1.3 settings error

Όταν πατηθεί τη κουμπί connect γίνεται ένας έλεγχος ορθότητας των τιμών των στοιχείων που πληκτρολογήθηκαν. Αν κάποιο από τα στοιχεία είναι λάθος ή παραλείψαμε να το πληκτρολογήσουμε τότε εμφανίζεται το παραπάνω μήνυμα.

## 5.2 Παράθυρο Ρύθμισης Βάρους Κόμβων



The image shows a software dialog box titled "node weight". It contains a section titled "weight distribution options" with several input fields for different attributes. The values entered in the fields are: room style: 8, room minor functionality: 2, room color palette: 6, mood characterization of room: 6, door relative position: 10, window relative position: 10, stair relative position: 10, and furniture type: 4. At the bottom of the dialog is a button labeled "save weights" with a floppy disk icon.

Attribute	Weight
room style	8
room minor functionality	2
room color palette	6
mood characterization of room	6
door relative position	10
window relative position	10
stair relative position	10
furniture type	4

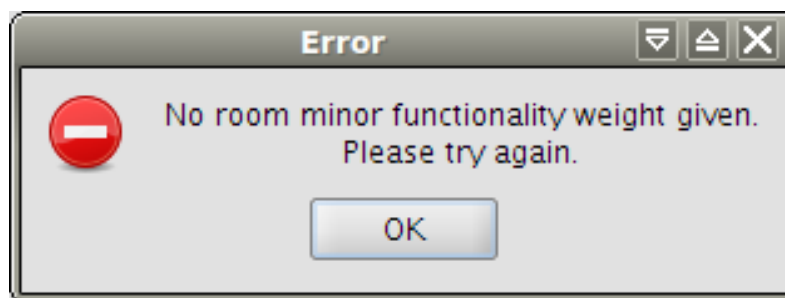
Εικόνα 5.2.1 node weight form

Το παράθυρο ρύθμισης βαρών κόμβων εμφανίζεται και αυτό αυτόματα κατά την πρώτη εκτέλεση της εφαρμογής μετά τη σύνδεση της εφαρμογής με τη βάση γνώσης η οποία έχει πραγματοποιηθεί μέσω του παραθύρου ρυθμίσεων (βλ. ενότητα 5.1). Αυτό το παράθυρο ζητά να οριστούν τα βάρη για τους παρακάτω κόμβους:

- Room style
- Room minor functionality

- Room color palette
- Mood characterization of room
- Door relative position
- Window relative position
- Stair relative position
- Furniture type

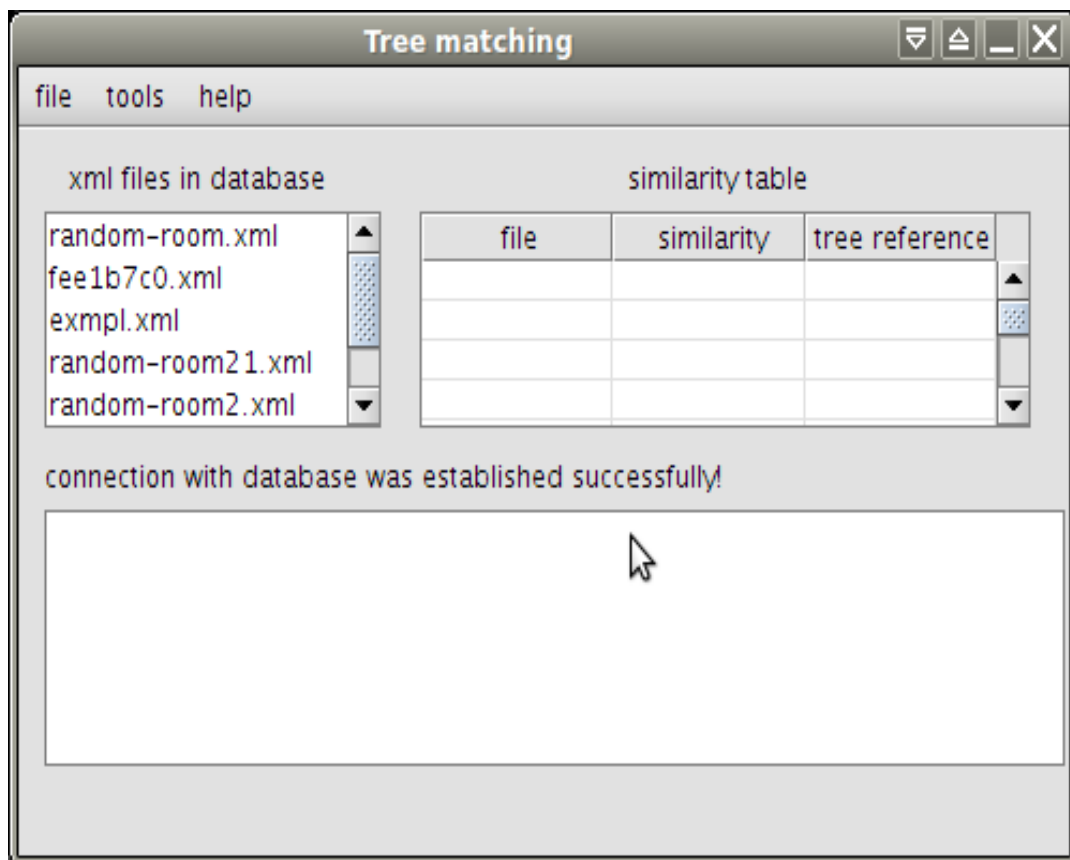
Αφού συμπληρωθούν τα στοιχεία αυτά τότε η εφαρμογή τα αποθηκεύει σε ένα xml έγγραφο και έως ότου χρειαστεί να αλλαχθεί κάποιο από τα βάρη δε χρειάζεται να ξαναπειράξουμε αυτή τη ρύθμιση. Με κάθε εκκίνηση πλέον η εφαρμογή θα βρίσκει και θα διαβάζει το xml αρχείο που δημιούργησε και θα παίρνει τις ίδιες ρυθμίσεις.



Εικόνα 5.2.2 weight error

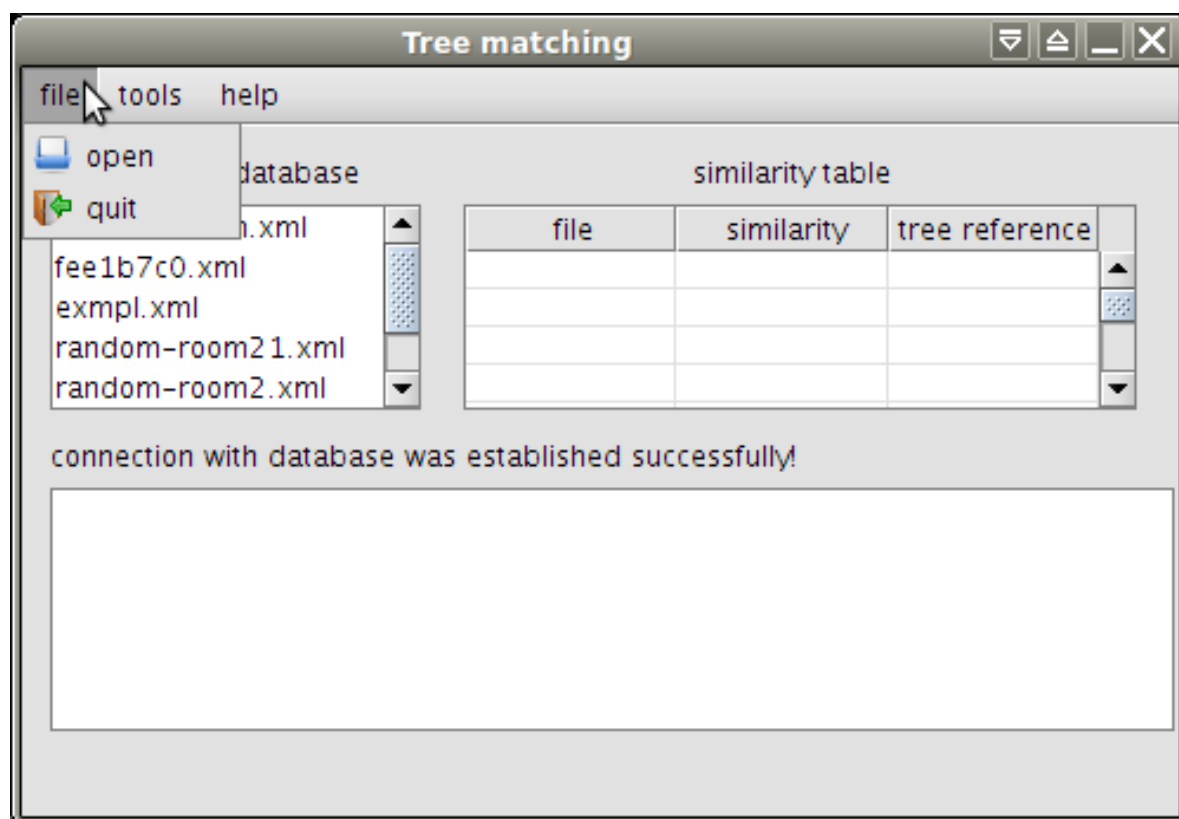
Ωστόσο αν κατά την πρώτη φορά κάποιο πεδίο βάρους μείνει κενό τότε εμφανίζεται το παραπάνω προειδοποιητικό μήνυμα και μας λέει ποιο είναι το στοιχείο που χρειάζεται συμπλήρωμα.

### 5.3 Παράθυρο εφαρμογής



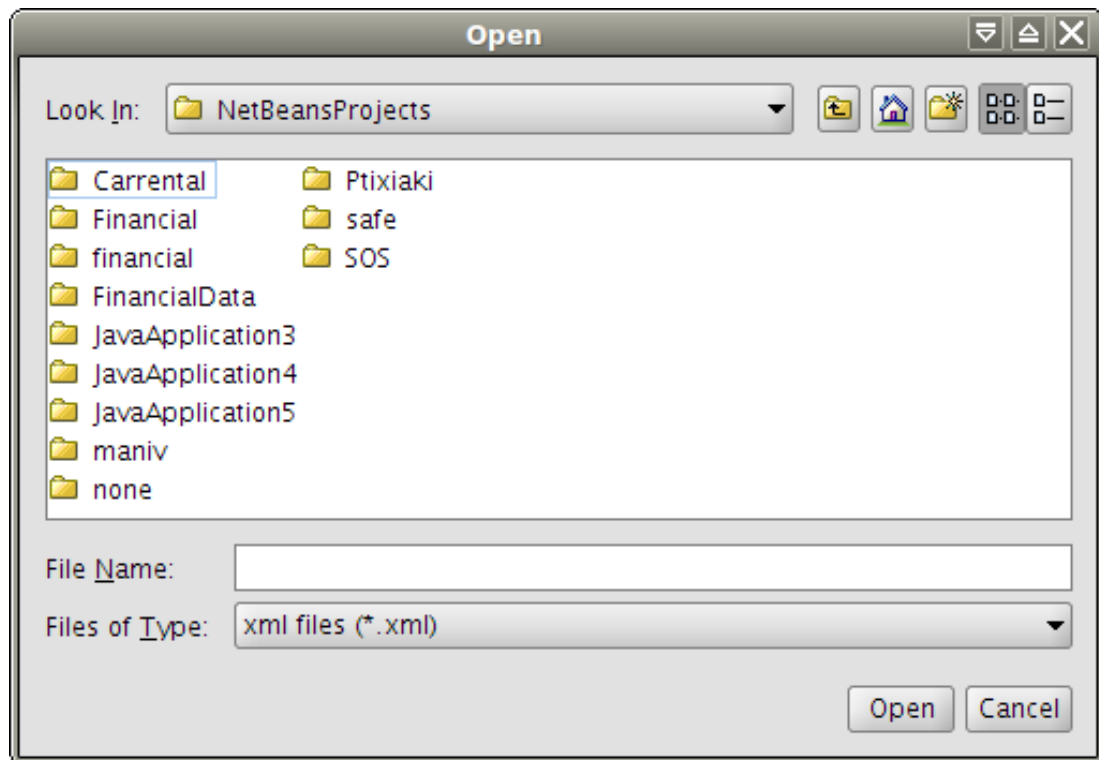
Εικόνα 5.3.1 application form

Όταν η εφαρμογή μας εκτελεστεί για πρώτη φορά και λόγω του ότι δεν έχουν οριστεί ακόμα οι πρωταρχικές, πολύ σημαντικές, πληροφορίες λειτουργικότητας, ανοίγει αυτόματα το παράθυρο settings και το παράθυρο ρύθμισης βαρών. Αφού οριστούν τα απαραίτητα αυτά στοιχεία τότε η εφαρμογή εκκινεί κανονικά.



Εικόνα 5.3.2 file menu

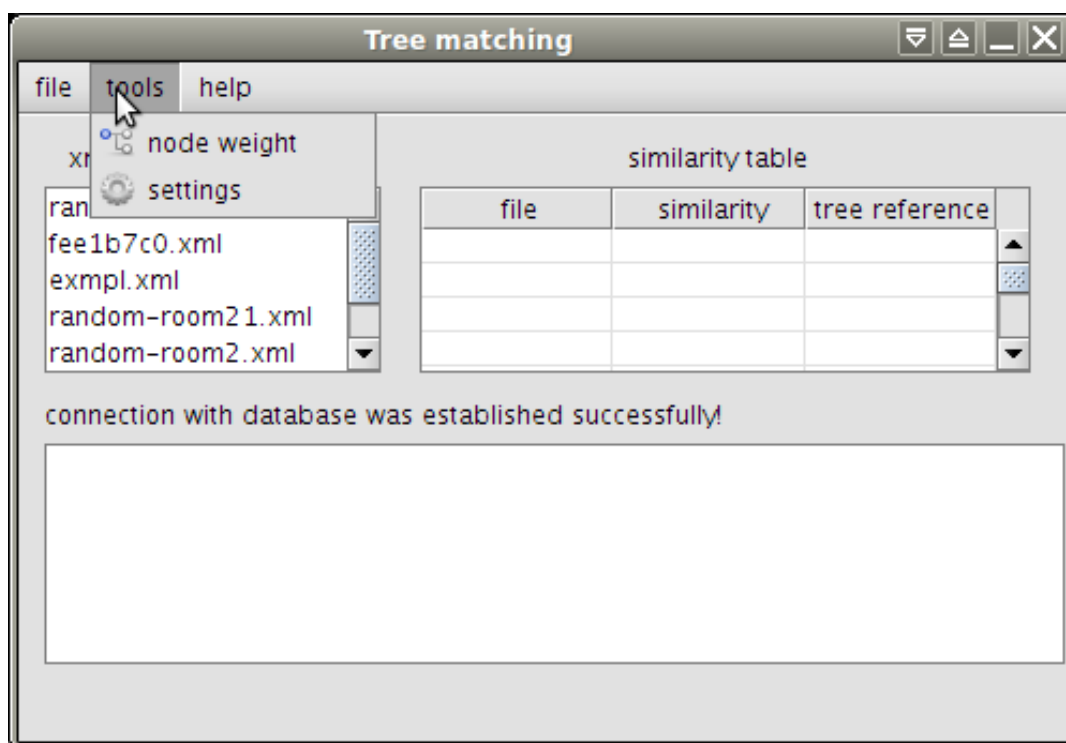
Στο menu file της εφαρμογής υπάρχει η δυνατότητα να κάνουμε αναζήτηση (**OPEN**) κάποιου xml αρχείου βάση του οποίου θα τρέξει ο αλγόριθμος. Επίσης έχουμε και την επιλογή **QUIT** η οποία τερματίζει την εφαρμογή μας.



Εικόνα 5.3.3 browse menu

Όταν πατηθεί η επιλογή OPEN από το μενού file τότε ανοίγει ένα καινούριο παράθυρο το οποίο δίνει την επιλογή στο χρήστη να αναζητήσει το αρχείο που θέλει να φορτώσει στην εφαρμογή.

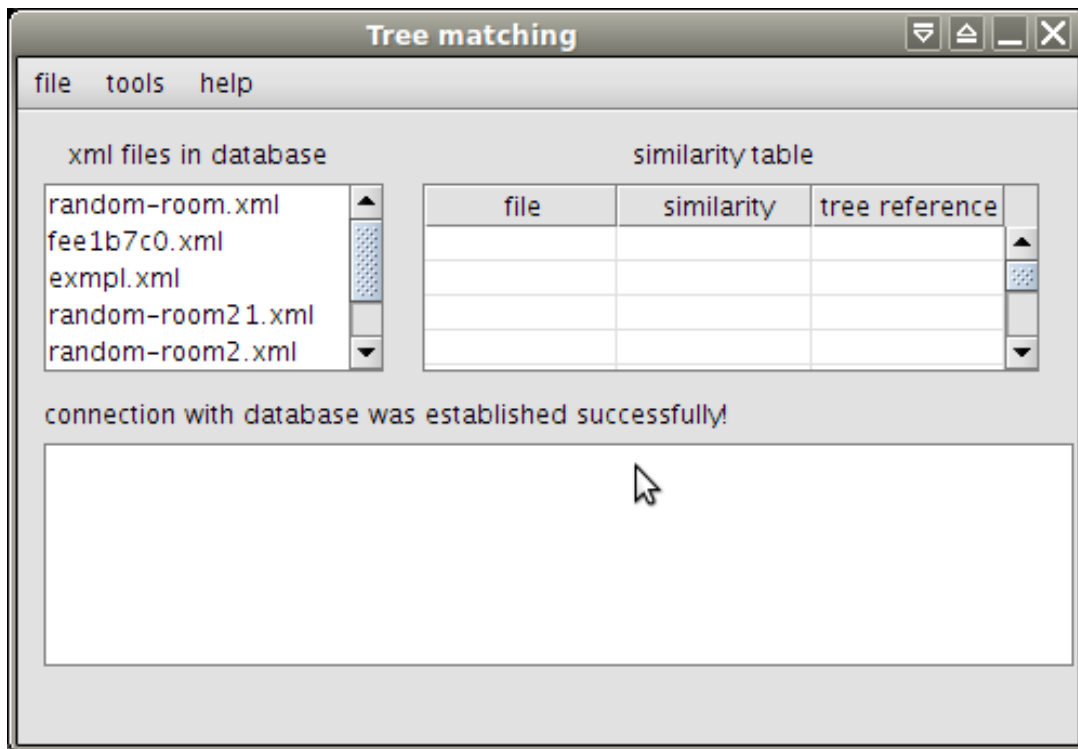
Η εφαρμογή διαθέτει φίλτρο ώστε να βλέπει τα αρχεία με κατάληξη **.xml**.



Εικόνα 5.3.4 tools menu

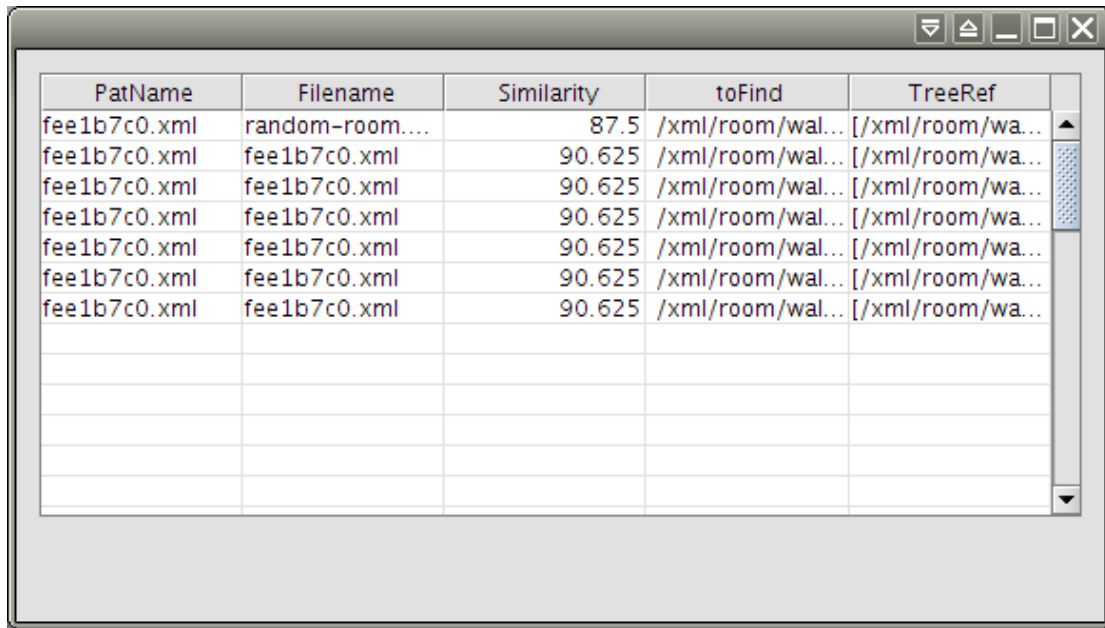
Στο μενού **tools** υπάρχει η επιλογή **node weight**, η οποία είναι υπεύθυνη για το άνοιγμα του παραθύρου ρύθμισης βαρών κόμβων και η επιλογή **settings** η οποία είναι υπεύθυνη με τη σειρά της για το άνοιγμα του παραθύρου εισαγωγής ρυθμίσεων της εφαρμογής.





Εικόνα 5.3.5 application form

Όταν έχουν οριστεί όλες οι ρυθμίσεις και γίνει η σύνδεση με τη βάση δεδομένων (βάση γνώσης της εφαρμογής), θα δούμε στο αριστερό μισό της εφαρμογής τα xml έγγραφα που περιέχονται στη βάση γνώσης και στο δεξί μισό τα αποτελέσματα που παρήγαγε ο αλγόριθμος.



PatName	Filename	Similarity	toFind	TreeRef
fee1b7c0.xml	random-room...	87.5	/xml/room/wal...	[/xml/room/wa...
fee1b7c0.xml	fee1b7c0.xml	90.625	/xml/room/wal...	[/xml/room/wa...
fee1b7c0.xml	fee1b7c0.xml	90.625	/xml/room/wal...	[/xml/room/wa...
fee1b7c0.xml	fee1b7c0.xml	90.625	/xml/room/wal...	[/xml/room/wa...
fee1b7c0.xml	fee1b7c0.xml	90.625	/xml/room/wal...	[/xml/room/wa...
fee1b7c0.xml	fee1b7c0.xml	90.625	/xml/room/wal...	[/xml/room/wa...
fee1b7c0.xml	fee1b7c0.xml	90.625	/xml/room/wal...	[/xml/room/wa...

Εικόνα 5.3.6 result panel

Μετά το πέρας των εργασιών του αλγορίθμου θα εμφανιστεί το παραπάνω παράθυρο το οποίο μας πληροφορεί σχετικά με τις συσχετίσεις xml εγγράφων που έκανε στην πορεία της λειτουργίας του ώσπου να καταλήξει στο αποτέλεσμα που παρήγαγε.

## Κεφάλαιο 6

### Προετοιμασία Εκτέλεσης της Εφαρμογής

## 6.1 Εγκατάσταση Προαπαιτούμενων Εφαρμογών

### **Βήμα 1: Java Installation and Setup**

Αρχικά τρέχουμε το αρχείο `jdk-6u16-nb-6_7_1-windows-ml.exe` το οποίο εγκαθιστά τις βασικές λειτουργίες τις `java` (`jdk` και `jre`) . Μαζί με τη `java` εγκαθίσταται και το `Netbeans IDE 6.7.1` το οποίο όμως (από το επίσημο site τις `Sun`) δεν εγκαθιστά τη λειτουργία `Servises>Servers>Apache Tomcat 6.0.20`. Επομένως αφού εγκατασταθεί πλήρως η `Java` και το `Netbeans`, απεγκαθιστούμε το `Netbeans` και τρέχουμε το δεύτερο αρχείο `Netbeans-6.7.1-ml-windows.exe` το οποίο εγκαθιστά ολοκληρωμένα `Netbeans IDE 6.7.1` σε συνδυασμό με τον `Apache Tomcat (Application Server) 6.0.20` .

Το μόνο που χρειάζεται να προσέξει κανείς κατά την εγκατάσταση του `Netbeans` είναι να τσεκάρει την επιλογή `customize` στο πεδίο (`tick box`) `Apache Tomcat 6.0.20` και όχι τις επιλογές `Glassfish`.

Στις μεταβλητές περιβάλλοντος (`Environment Variables`) των `Windows` δημιουργούμε δύο νέες μεταβλητές συστήματος:

Όνομα (`Name`) : `JAVA_HOME` και τιμή (`Value`) : `C:\Program Files\Java\jdk1.6.0_16\bin` (ή όποιο άλλο `path` που βρίσκεται η `Java` αρκεί να δείχνει το `bin` φάκελο του `jdk`).

Όνομα (`Name`) : `CLASSPATH` και τιμή (`Value`) : `C:\Program Files\Java\jdk1.6.0_16\jre\lib\ext` (ή όποιο άλλο `path` βρίσκεται στη βιβλιοθήκη `lib\ext` της `Java` αρκεί να δείχνει στον `ext` φάκελο του `jdk\jre`).

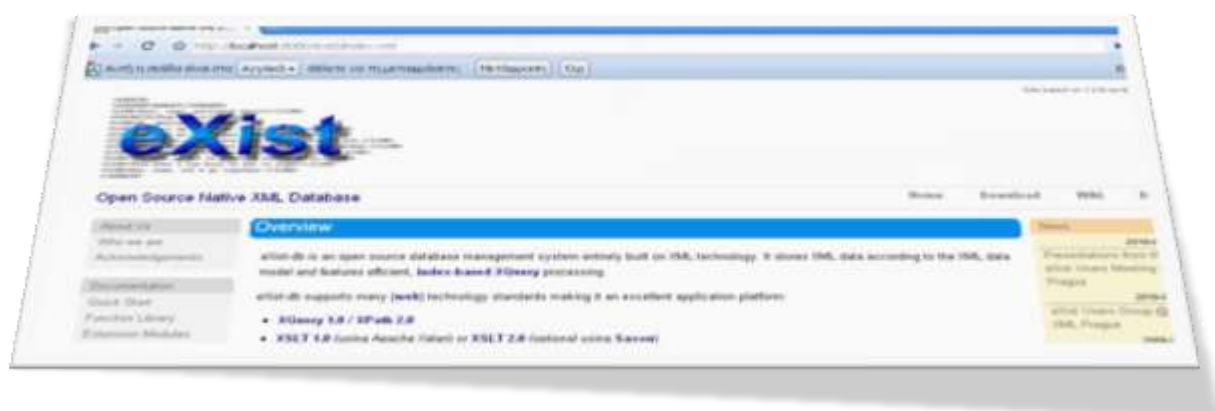
### **Βήμα 2: Project Setup**

Για να τρέξουμε τώρα την εφαρμογή μας κάνουμε αντιγραφή στο φάκελο treematching και τον επικολλούμε στο φάκελο Netbeans projects όπου αυτός υπάρχει (συνήθως στο MyDocuments), και εκκινούμε το Netbeans IDE.

Όταν το Netbeans ανοίξει βλέπουμε το project αλλά με errors (ένα κόκκινο θαυμαστικό δίπλα από το εικονίδιο του project). Τότε επιλέγουμε το project μας, κάνουμε δεξί click και επιλέγουμε properties. Στην καρτέλα που εμφανίζεται επιλέγουμε από το δέντρο αριστερά libraries και με το Add Jar/Folder εισάγουμε στο project τα Jar αρχεία από το φάκελο Jars, και όταν τελειώσουμε πατάμε το ok.

Αν αυτό το βήμα έχει γίνει σωστά τότε έχουμε τελειώσει με το setup τις εφαρμογής και το κόκκινο σημάδι θα έχει εξαφανιστεί.

### **Βήμα 3 : eXist Setup**



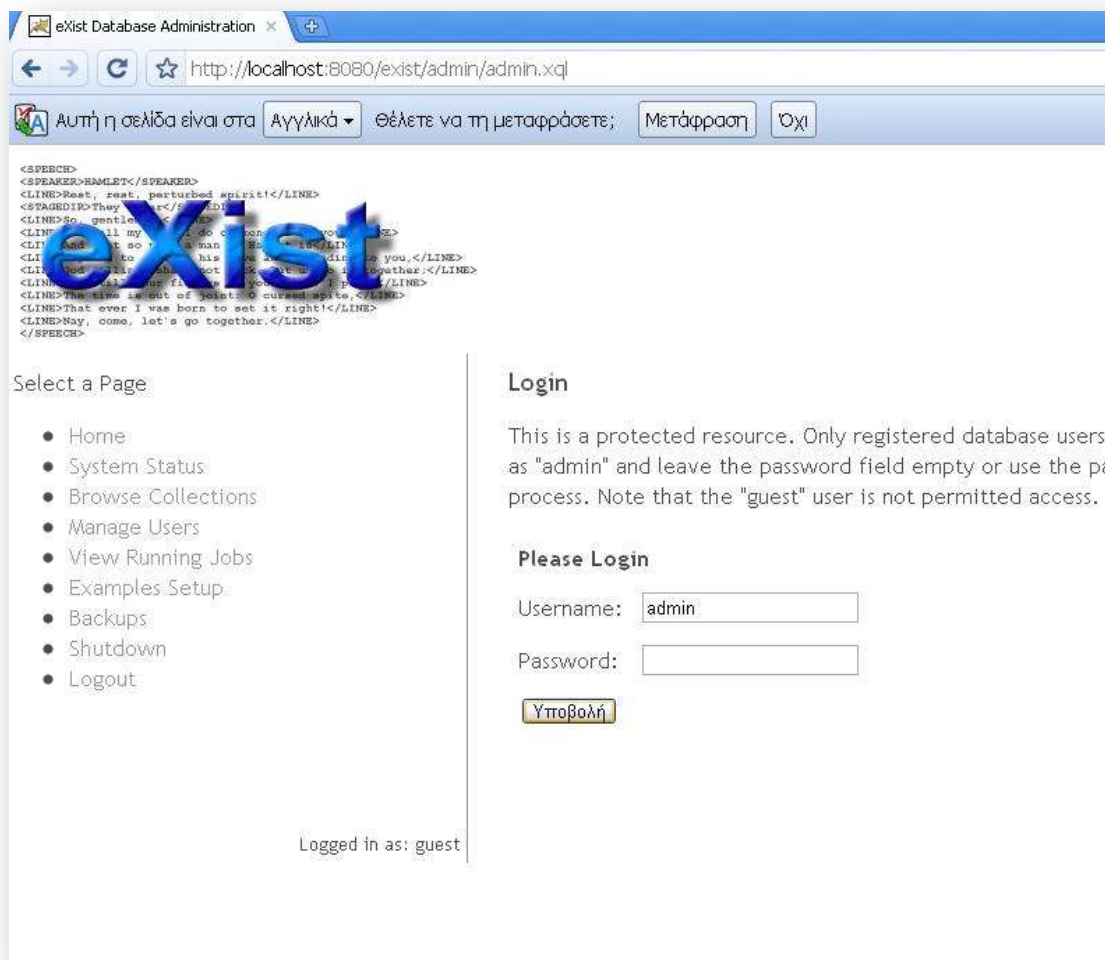
Εικόνα 6.1 Exist XML Database

Αντιγράφουμε το αρχείο eXist.war από το φάκελο existdb και το επικολλούμε στο φάκελο του Apache Tomcat Application Server στο path : C:\Users\\.netbeans\6.7\apache-tomcat-6.0.20\_base\webapps . Στη συνέχεια από την πλατφόρμα Netbeans επιλέγουμε Tab Services > Servers > Apache Tomcat

6.0.18 > (right click) Start. Ξεκινώντας τη λειτουργία του ο Server, αρχικοποιεί τη βάση eXist και δημιουργεί το φάκελο της βάσης (eXist) .

Στη συνέχεια ανοίγουμε τον browser και γράφουμε σαν uri : <http://localhost:8080/exist> όπου και βλέπουμε το περιβάλλον της βάσης.

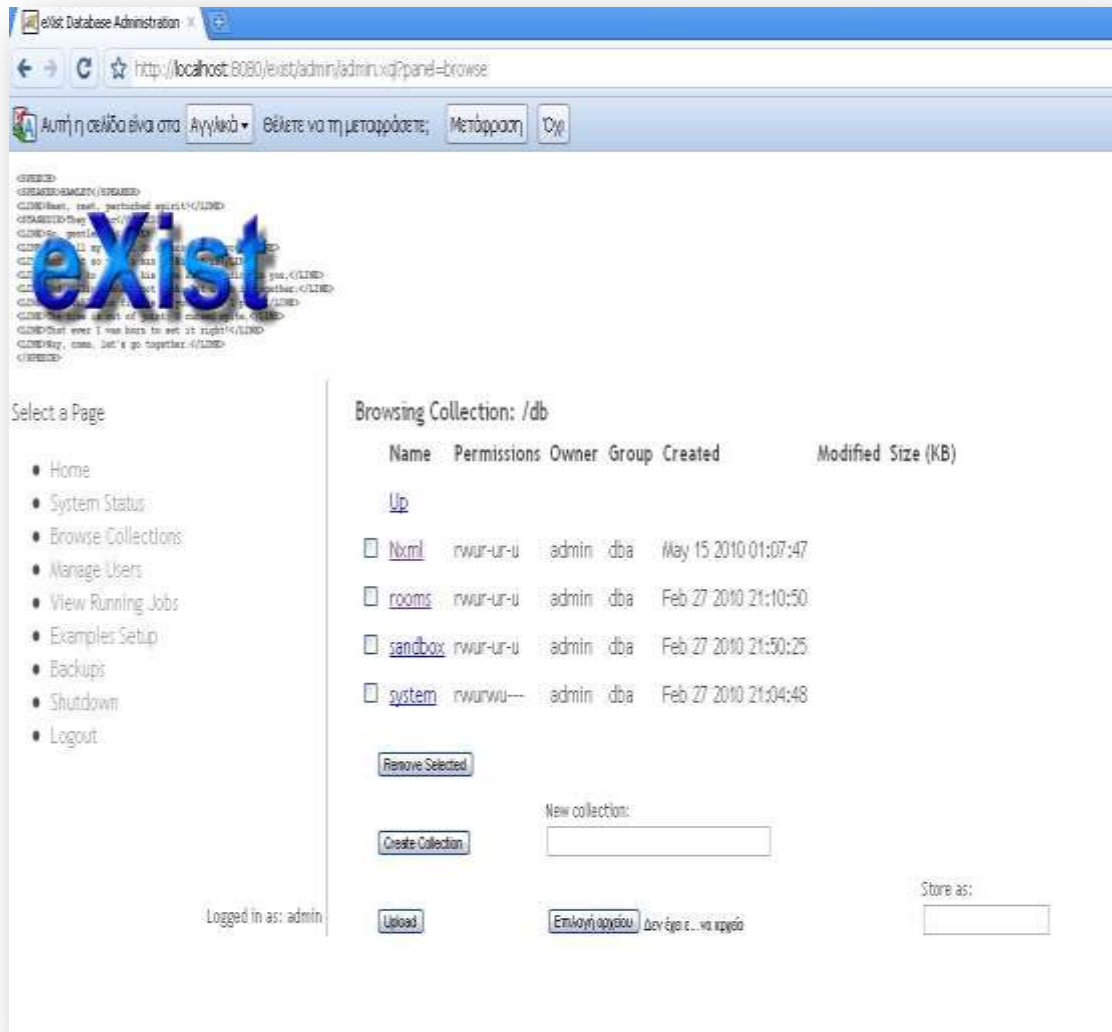
Στο περιβάλλον της βάσης κάνουμε scroll down και κάτω αριστερά κάνουμε κλικ στην επιλογή Admin και βάζουμε username: Admin και password: (κενό αρχικά).



Εικόνα 6.2 Exist Login Screen

Παταμε υποβολή και οδηγούμαστε στο περιβάλλον του διαχειριστή. Από εκεί επιλέγουμε Browse Collections και μπορούμε πλέον να δούμε τις συλλογές μας. Δημιουργούμε δύο collections, η μια με όνομα Nxml και η δεύτερη με όνομα rooms.

Σε αυτή με όνομα Nxml θα αποθηκεύσουμε το αρχείο με όνομα nulXml.xml που είναι το κένο xml και στη δεύτερη δηλαδή την rooms θα είναι αποθηκευμένο το repository μας δηλαδή όλα τα υπόλοιπα xml δωματίων.



Εικόνα 6.3 exist Collections

Αν μέχρι εδώ έχουν γίνει όλα σωστά τότε είμαστε έτοιμοι να τρέξουμε την εφαρμογή και αυτή θα συνεργάζεται άμεσα με την βάση μας.



### Επίλογος

Μετά το πέρας της παρούσας πτυχιακής εργασίας μπορούμε να πούμε πως δημιουργήθηκε επιτυχώς ένας αλγόριθμος, τεχνητής νοημοσύνης, ικανός να διακοσμήσει εσωτερικούς χώρους.

Σκοπός αυτής της εργασίας είναι να ενωθεί στο μέλλον, με ένα σύνολο άλλων εφαρμογών, οι οποίες έχουν πραγματοποιηθεί σε άλλες πτυχιακές εργασίες, ώστε να σχηματίσουν ένα ολοκληρωμένο σύστημα διακόσμησης εσωτερικών χώρων. Από αυτές τις εφαρμογές άλλες δημιουργούν xml έγγραφα που παίρνονται ως είσοδος στην παρούσα εργασία, και άλλες οπτικοποιούν το αποτέλεσμα του διακοσμημένου δωματίου. Έτσι στο μέλλον θα μπορεί οποιοσδήποτε ακόμα και από τον καναπέ του σπιτιού του να διακοσμήσει το χώρο του εύκολα, γρήγορα, δωρεάν και πάνω απ' όλα έγκυρα.

Ο αλγόριθμος αυτός θα μπορούσε να αναπτυχθεί στο μέλλον και άλλο, αν προσθεθούν για παράδειγμα περισσότεροι κανόνες για τη λήψη αποφάσεων απ' ότι έχουν χρησιμοποιηθεί έως τώρα, ώστε τα αποτελέσματα να είναι ακόμα πιο ακριβή.

## ΠΑΡΑΡΤΗΜΑ

## 1. Javadoc Εφαρμογής

---

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>	<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>	

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)      DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

adaptation

### Class gettingTheXml

```
java.lang.Object  
  adaptation.gettingTheXml
```

```
public class gettingTheXml  
  extends java.lang.Object
```

### Constructor Summary

<a href="#">gettingTheXml</a> () gettingTheXml empty constructor.
--

### Method Summary

int	<a href="#">getIt</a> ()
-----	--------------------------

<code>java.util.List</code>	<a href="#"><u>getNodes</u></a> ()
<code>java.util.List</code>	<a href="#"><u>getNodes1</u></a> ()
<code>java.util.Vector&lt;java.lang.String&gt;</code>	<a href="#"><u>load</u></a> () this method is used to load the group xml
<code>static void</code>	<a href="#"><u>main</u></a> (java.lang.String[] args)
<code>void</code>	<a href="#"><u>setIt</u></a> (int it)
<code>void</code>	<a href="#"><u>setNodes</u></a> (java.util.List nodes)
<code>void</code>	<a href="#"><u>setNodes1</u></a> (java.util.List nodes1)

### Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### gettingTheXml

```
public gettingTheXml()  
    gettingTheXml empty constructor.
```

## Method Detail

## getIt

```
public int getIt()
```

---

## setIt

```
public void setIt(int it)
```

---

## getNodes

```
public java.util.List getNodes()
```

---

## setNodes

```
public void setNodes(java.util.List nodes)
```

---

## getNodes1

```
public java.util.List getNodes1()
```

---

## setNodes1

```
public void setNodes1(java.util.List nodes1)
```

---

## load

```
public java.util.Vector<java.lang.String> load()  
this method is used to load the group xml
```

### Returns:

SumVec Vector String AlexV

---

## main

```
public static void main(java.lang.String[] args)
```

---

[Overview](#) [Package](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

---

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use</a>	<a href="#">Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>	
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>				<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)      DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

adaptation

## Class SecondPanel

```
java.lang.Object
  java.awt.Component
    java.awt.Container
      java.awt.Window
        java.awt.Frame
          javax.swing.JFrame
            adaptation.SecondPanel
```

### All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable,  
javax.accessibility.Accessible, javax.swing.RootPaneContainer,  
javax.swing.WindowConstants

---

```
public class SecondPanel
  extends javax.swing.JFrame
```

### See Also:

[Serialized Form](#)

---

## Nested Class Summary

### Nested classes/interfaces inherited from class javax.swing.JFrame

```
javax.swing.JFrame.AccessibleJFrame
```

### Nested classes/interfaces inherited from class java.awt.Frame

java.awt.Frame.AccessibleAwtFrame

### Nested classes/interfaces inherited from class java.awt.Window

java.awt.Window.AccessibleAwtWindow

### Nested classes/interfaces inherited from class java.awt.Container

java.awt.Container.AccessibleAwtContainer

### Nested classes/interfaces inherited from class java.awt.Component

java.awt.Component.AccessibleAwtComponent,  
java.awt.Component.BaselineResizeBehavior,  
java.awt.Component.BltBufferStrategy,  
java.awt.Component.FlipBufferStrategy

## Field Summary

protected static org.xmldb.api.base.Collection	<a href="#">collection</a>
static <a href="#">NodeWeight</a>	<a href="#">nodeWeight</a>
protected static <a href="#">Settings</a>	<a href="#">settings</a>

static <a href="#">WeightHashMap</a>	<a href="#">weightHashMap</a>
protected static java.lang.String[]	<a href="#">xmlFiles</a>

### Fields inherited from class javax.swing.JFrame

accessibleContext, EXIT\_ON\_CLOSE, rootPane, rootPaneCheckingEnabled

### Fields inherited from class java.awt.Frame

CROSSHAIR\_CURSOR, DEFAULT\_CURSOR, E\_RESIZE\_CURSOR, HAND\_CURSOR, ICONIFIED, MAXIMIZED\_BOTH, MAXIMIZED\_HORIZ, MAXIMIZED\_VERT, MOVE\_CURSOR, N\_RESIZE\_CURSOR, NE\_RESIZE\_CURSOR, NORMAL, NW\_RESIZE\_CURSOR, S\_RESIZE\_CURSOR, SE\_RESIZE\_CURSOR, SW\_RESIZE\_CURSOR, TEXT\_CURSOR, W\_RESIZE\_CURSOR, WAIT\_CURSOR

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

### Fields inherited from interface javax.swing.WindowConstants

DISPOSE\_ON\_CLOSE, DO\_NOTHING\_ON\_CLOSE, HIDE\_ON\_CLOSE



## Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

## Constructor Summary

### [SecondPanel](#) ()

Creates new form SecondPanel

## Method Summary

java.lang.String	<a href="#">BalanceVectors</a> (java.util.Vector<java.lang.String> balancer) this method is used to give out the best graph that found in the whole search
boolean	<a href="#">checkVec</a> (java.util.Vector<java.lang.String> Vec) this method is used to prevent infinite loops by checking if the previous "to be searched" vector is the same as the present one
java.lang.String	<a href="#">getFilePat</a> h ()
static void	<a href="#">main</a> (java.lang.String[] args)
java.util.Vector<java.lang.String>	<a href="#">REsearch2</a> (java.lang.String FileName, java.util.Vector<java.lang.String> SearchPaths, java.lang.String Fpath, java.util.Vector<java.lang.String> XmlPaths) this method feeds the Rse method with paths to be searched again
java.util.Vector<java.lang.String>	<a href="#">Rse</a> ( <a href="#">TreeModeling</a> patternLoader1,

<code>ng.String&gt;</code>	<code>java.lang.String xxpin, java.util.Vector&lt;java.lang.String&gt; XmlPath s1)</code> <p>this method is to re-check the database for the paths that didn't matched in previous search this method is being fed by the research method</p>
----------------------------	---

### Methods inherited from class `javax.swing.JFrame`

`addImpl, createRootPane, frameInit, getAccessibleContext,  
getContentPane, getDefaultCloseOperation, getGlassPane, getGraphics,  
getJMenuBar, getLayeredPane, getRootPane, getTransferHandler,  
isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled,  
paramString, processWindowEvent, remove, repaint, setContentPane,  
setDefaultCloseOperation, setDefaultLookAndFeelDecorated,  
setGlassPane, setIconImage, setJMenuBar, setLayeredPane, setLayout,  
setRootPane, setRootPaneCheckingEnabled, setTransferHandler, update`

### Methods inherited from class `java.awt.Frame`

`addNotify, getCursorType, getExtendedState, getFrames, getIconImage,  
getMaximizedBounds, getMenuBar, getState, getTitle, isResizable,  
isUndecorated, remove, removeNotify, setCursor, setExtendedState,  
setMaximizedBounds, setMenuBar, setResizable, setState, setTitle,  
setUndecorated`

### Methods inherited from class `java.awt.Window`

`addPropertyChangeListener, addPropertyChangeListener,  
addWindowFocusListener, addWindowListener, addWindowStateListener,  
applyResourceBundle, applyResourceBundle, createBufferStrategy,  
createBufferStrategy, dispose, getBufferStrategy,  
getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner,  
getFocusTraversalKeys, getGraphicsConfiguration, getIconImages,  
getInputContext, getListeners, getLocale, getModalExclusionType,  
getMostRecentFocusOwner, getOwnedWindows, getOwner,  
getOwnerlessWindows, getToolkit, getWarningString,`

```
getWindowFocusListeners, getWindowListeners, getWindows,  
getWindowStateListeners, hide, isActive, isAlwaysOnTop,  
isAlwaysOnTopSupported, isFocusableWindow, isFocusCycleRoot,  
isFocused, isLocationByPlatform, isShowing, pack, postEvent,  
processEvent, processWindowFocusEvent, processWindowStateEvent,  
removeWindowFocusListener, removeWindowListener,  
removeWindowStateListener, reshape, setAlwaysOnTop, setBounds,  
setBounds, setCursor, setFocusableWindowState, setFocusCycleRoot,  
setIconImages, setLocationByPlatform, setLocationRelativeTo,  
setMinimumSize, setModalExclusionType, setSize, setSize, setVisible,  
show, toBack, toFront
```

### Methods inherited from class java.awt.Container

```
add, add, add, add, add, addContainerListener,  
applyComponentOrientation, areFocusTraversalKeysSet, countComponents,  
deliverEvent, doLayout, findComponentAt, findComponentAt,  
getAlignmentX, getAlignmentY, getComponent, getComponentAt,  
getComponentAt, getComponentCount, getComponents, getComponentZOrder,  
getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout,  
getMaximumSize, getMinimumSize, getMousePosition, getPreferredSize,  
insets, invalidate, isAncestorOf, isFocusCycleRoot,  
isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout,  
list, list, locate, minimumSize, paint, paintComponents,  
preferredSize, print, printComponents, processContainerEvent, remove,  
removeAll, removeContainerListener, setComponentZOrder,  
setFocusTraversalKeys, setFocusTraversalPolicy,  
setFocusTraversalPolicyProvider, setFont, transferFocusBackward,  
transferFocusDownCycle, validate, validateTree
```

### Methods inherited from class java.awt.Component

```
action, add, addComponentListener, addFocusListener,  
addHierarchyBoundsListener, addHierarchyListener,  
addInputMethodListener, addKeyListener, addMouseListener,  
addMouseMotionListener, addMouseWheelListener, bounds, checkImage,  
checkImage, coalesceEvents, contains, contains, createImage,  
createImage, createVolatileImage, createVolatileImage, disable,  
disableEvents, dispatchEvent, enable, enable, enableEvents,  
enableInputMethods, firePropertyChange, firePropertyChange,  
firePropertyChange, firePropertyChange, firePropertyChange,
```

```
firePropertyChange, firePropertyChange, firePropertyChange,
firePropertyChange, getBackground, getBaseline,
getBaselineResizeBehavior, getBounds, getBounds, getColorModel,
getComponentListeners, getComponentOrientation, getCursor,
getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled,
getFont, getFontMetrics, getForeground, getHeight,
getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint,
getInputMethodListeners, getInputMethodRequests, getKeyListeners,
getLocation, getLocation, getLocationOnScreen, getMouseListeners,
getMouseMotionListeners, getMousePosition, getMouseWheelListeners,
getName, getParent, getPeer, getPropertyChangeListeners,
getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth,
getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside,
isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered,
isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet,
isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet,
isOpaque, isPreferredSizeSet, isValid, isVisible, keyDown, keyUp,
list, list, list, location, lostFocus, mouseDown, mouseDrag,
mouseenter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll,
prepareImage, prepareImage, printAll, processComponentEvent,
processFocusEvent, processHierarchyBoundsEvent,
processHierarchyEvent, processInputMethodEvent, processKeyEvent,
processMouseEvent, processMouseMotionEvent, processMouseWheelEvent,
removeComponentListener, removeFocusListener,
removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener, removeMouseListener,
removeMouseMotionListener, removeMouseWheelListener,
removePropertyChangeListener, removePropertyChangeListener, repaint,
repaint, repaint, requestFocus, requestFocus, requestFocusInWindow,
requestFocusInWindow, resize, resize, setBackground,
setComponentOrientation, setDropTarget, setEnabled, setFocusable,
setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint,
setLocale, setLocation, setLocation, setMaximumSize, setName,
setPreferredSize, show, size, toString, transferFocus,
transferFocusUpCycle
```

### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait,
wait, wait
```

### Methods inherited from interface java.awt.MenuContainer

getFont, postEvent

### Field Detail

#### collection

protected static org.xmldb.api.base.Collection **collection**

---

#### xmlFiles

protected static java.lang.String[] **xmlFiles**

---

#### settings

protected static [Settings](#) **settings**

---

#### nodeWeight

public static [NodeWeight](#) **nodeWeight**

---

#### weightHashMap

public static [WeightHashMap](#) **weightHashMap**

### Constructor Detail

#### SecondPanel

public **SecondPanel**()  
Creates new form SecondPanel

### Method Detail

#### REsearch2

```
public java.util.Vector<java.lang.String>  
RResearch2(java.lang.String FileName,  
  
java.util.Vector<java.lang.String> SearchPaths,  
  
java.lang.String Fpath,  
  
java.util.Vector<java.lang.String> XmlPaths)  
    this method feeds the Rse method with paths to be searched again
```

### Parameters:

FileName - String

Searchpaths - Vector String the Vector who contains the paths to be searched

Fpath - String The filepath

XmlPaths - Vector String previous paths that are ready to be placed in the new xml  
AlexV

---

### getFilePath

```
public java.lang.String getFilePath()
```

---

### Rse

```
public java.util.Vector<java.lang.String>  
Rse(TreeModeling patternLoader1,  
  
java.lang.String xxpin,  
  
java.util.Vector<java.lang.String> XmlPaths1)  
    this method is to re-check the database for the paths that didn't matched in previous  
    search this method is being fed by the research method
```

### Parameters:

patternloader1 - treemodeling

xxpin - String that contains the path to be searched

XmlPaths1 - Vector string the vector that contains the previous searched paths that  
are ready to be xml AlexV

---

### BalanceVectors

```
public java.lang.String
```

```
BalanceVectors (java.util.Vector<java.lang.String> balancer)
```

this method is used to give out the best graph that found in the whole search

### Parameters:

balancer - Vector String is the vector containing the paths

### Returns:

String AlexV

---

## checkVec

```
public boolean checkVec (java.util.Vector<java.lang.String> Vec)
```

this method is used to prevent infinite loops by checking if the previous "to be searched" vector is the same as the present one

### Parameters:

Vec - Vector string the vector containing the paths that come from research method AlexV

---

## main

```
public static void main (java.lang.String[] args)
```

### Parameters:

args - the command line arguments

---

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>	
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>			<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>
SUMMARY: <a href="#">NESTED</a>   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>				DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>		

---

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>	
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>			<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>
SUMMARY: <a href="#">NESTED</a>   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>				DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>		

---

adaptation

## Class deepSearch

```
java.lang.Object  
  adaptation.deepSearch
```

---

```
public class deepSearch  
  extends java.lang.Object
```

---

### Field Summary

protected static org.xmldb.api.base.Collection	<a href="#">collection</a>
static <a href="#">NodeWeight</a>	<a href="#">nodeWeight</a>
protected static <a href="#">Settings</a>	<a href="#">settings</a>
protected static java.lang.String[]	<a href="#">xmlFiles</a>

### Constructor Summary

[deepSearch](#) ()

### Method Summary

```
void REsearch(java.lang.String FileName,  
             java.util.Vector<java.lang.String> SearchPaths)
```



--	--

### Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Field Detail

### collection

`protected static org.xmldb.api.base.Collection collection`

---

### xmlFiles

`protected static java.lang.String[] xmlFiles`

---

### settings

`protected static Settings settings`

---

### nodeWeight

`public static NodeWeight nodeWeight`

## Constructor Detail

### deepSearch

`public deepSearch()`

## Method Detail

### REsearch

`public void REsearch(java.lang.String fileName,`

java.util.Vector<java.lang.String> SearchPaths)

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>	<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>	
SUMMARY: NESTED   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>			DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>		

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>	<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>	
SUMMARY: NESTED   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>			DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>		

adaptation

## Class GroupCheck

java.lang.Object  
adaptation.GroupCheck

```
public class GroupCheck  
extends java.lang.Object
```

## Constructor Summary

[GroupCheck](#) ()

## Method Summary

java.util.Vector<java.lang.String>	<a href="#">GroupChecker1</a> (java.util.Vector<java.lang.String> CheckV1) this method is used to check the xml for groups
------------------------------------	---

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GroupCheck

```
public GroupCheck()
```

## Method Detail

### GroupChecker1

```
public java.util.Vector<java.lang.String>  
GroupChecker1 (java.util.Vector<java.lang.String> CheckV1)  
this method is used to check the xml for groups
```

#### Parameters:

CheckV1 - Vector String is the vector with tha paths

#### Returns:

CheckV Vector String is the vector checked for groups alexV

---

[Overview](#) [Package](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

[Overview](#) [Package](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

adaptation

## Class XmlCreator

```
java.lang.Object  
  adaptation.XmlCreator
```

```
public class XmlCreator  
  extends java.lang.Object
```

### Field Summary

protected static org.xmldb.api.base.Collection	<a href="#">collection</a>
protected static <a href="#">Settings</a>	<a href="#">settings</a>
protected static java.lang.String[]	<a href="#">xmlFiles</a>

### Constructor Summary

[XmlCreator](#) ()

### Method Summary

java.util.Vector<java.lang.String>	<a href="#">checkDuplicate</a> (java.util.Vector<java.lang.String> Vec)
------------------------------------	---

void	<a href="#">makeXml</a> (java.util.Vector<java.lang.String > Vec4) this method is used to create the new final xml
void	<a href="#">modifyDocument</a> (java.util.Vector<java.lang .String> Vec)

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### collection

protected static org.xmldb.api.base.Collection **collection**

---

### xmlFiles

protected static java.lang.String[] **xmlFiles**

---

### settings

protected static [Settings](#) **settings**

## Constructor Detail

### XmlCreator

public **XmlCreator**()

## Method Detail

## makeXml

```
public void makeXml(java.util.Vector<java.lang.String> Vec4)
    throws java.io.FileNotFoundException,
           java.io.IOException,
           org.xmldb.api.base.XMLDBException
```

this method is used to create the new final xml

### Parameters:

Vec4 - Vector String is the vector with the final paths AlexV

### Throws:

java.io.FileNotFoundException

java.io.IOException

org.xmldb.api.base.XMLDBException

---

## modifyDocument

```
public void modifyDocument(java.util.Vector<java.lang.String> Vec)
    throws org.xmldb.api.base.XMLDBException
```

### Throws:

org.xmldb.api.base.XMLDBException

---

## checkDuplicate

```
public java.util.Vector<java.lang.String>
checkDuplicate(java.util.Vector<java.lang.String> Vec)
```

---

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>	
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>	<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>		
SUMMARY: NESTED   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>			DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>			

---

<a href="#">Overview</a>	<a href="#">Package</a>	<a href="#">Use Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>	
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>	<a href="#">FRAMES</a>	<a href="#">NO FRAMES</a>	<a href="#">All Classes</a>		

---

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

**retain**

## Class store

```
java.lang.Object  
  retain.store
```

```
public class store  
  extends java.lang.Object
```

## Field Summary

static java.lang.String	<a href="#">URI</a>
-------------------------	---------------------

## Constructor Summary

<a href="#">store</a> ()	
--------------------------	--

## Method Summary

void	<a href="#">storeXml</a> (org.dom4j.Document doc) this method is used to store the newly produced xml into the database
------	--

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### URI

```
public static final java.lang.String URI
```

#### See Also:

[Constant Field Values](#)

## Constructor Detail

### store

```
public store()
```

## Method Detail

### storeXml

```
public void storeXml(org.dom4j.Document doc)
    throws org.xmldb.api.base.XMLDBException,
           java.lang.ClassNotFoundException,
           java.lang.InstantiationException,
           java.lang.IllegalAccessException
```

this method is used to store the newly produced xml into the database

#### Parameters:

Doc - Document is the document produced by the XmlCreator AlexV

#### Throws:

org.xmldb.api.base.XMLDBException

java.lang.ClassNotFoundException

java.lang.InstantiationException

java.lang.IllegalAccessException

---

[Overview](#) [Package](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)



## Συλλογιστική βασισμένη σε περιπτώσεις

---

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## 2. XML pattern έγγραφο που χρησιμοποιείται για να περιγραφεί ένα δωμάτιο

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2008 rel. 2 sp1 (http://www.altova.com) by MCLab (EMBRACE)
-->
<xml>
  <header>
    <File>filepath</File>
      <Date/>
      <Designer>"</Designer>
      <Company>'bhj'</Company>
      <Project>
        <projectID>"</projectID>
          <projectName>"</projectName>
          <projectAddress>"</projectAddress>
          <projectDescription>"</projectDescription>
        </Project>
      <config>
        <units>
          <distance>'m,ft,cm,mm,inch'</distance>
            <angle>degree</angle>
            <area>'m2...'</area>
            <volume>'m3'</volume>
          </units>
        </config>
      </header>
      <room MoodCharacterization="Energetic">
        <model3D>
          <coordinates>x,y,z virtual world</coordinates>
            <path>url of the x3d model3D</path>
          </model3D>
          <shape>SVG something</shape>
```

```
<dimension>number,number,number-|x|,|y|,|z| bounding box</dimension>
<style styleIndexName="modern" styleIndexURL="URL of XML of style
Index">3</style>
<ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
<functionality>
<MajorFunctionality>Kitchen</MajorFunctionality>
<MinorFunctionality rankNo="a number that shows the priority in
minor functionalities">Dining room</MinorFunctionality>
</functionality>
<RoomCoveringFactor>percentage of the covered space (by furniture etc)
bounding box of the room - Sum (bounding boxes of the furniture)/bounding box of the room
</RoomCoveringFactor>
<lighting>in lux the light volume of the room (artificial and daylight as
well)</lighting>
<ceiling>
<model3D>
<coordinates> x,y,z virtual world</coordinates>
<path>url of the x3d model3D</path>
</model3D>
<shape>square</shape>
<dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
<style styleIndexName="classic" styleIndexURL="URL of XML of
style Index">1</style>
<material>
<materialType materialIndexName="Wood" materialIndexURL="URL of Material
Index">1</materialType>
<pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
</material>
<ColorPalette ColorPaletteIndexName="white"
ColorPaletteIndexURL="URL of palette Index">#ffffff</ColorPalette>
```

```
<lightSource>
  <model3D>
    <coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  <dimensions>number,number,number - |x|,|y|,|z| bounding
box dimensions</dimensions>
  <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
  <lightSourcePalletteIndex
lightSourcePalletteIndexName="light Pallette index name"
lightSourcePalletteIndexURL="URL of XML of style Index">light source index
number</lightSourcePalletteIndex>
</lightSource>
<Construction>
  <model3D>
    <coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
  </model3D>
  <dimensions>number,number,number - |x|,|y|,|z| bounding
box dimensions</dimensions>
  <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
  <style styleIndexName="style index name"
styleIndexURL="URL of XML of style Index">style code</style>
  <material>
    <materialType materialIndexName="material index name"
materialIndexURL="URL of Material Index">code</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
  </material>
  <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
```

```

        </Construction>
    </ceiling>
    <wall>
<model3D>
    <coordinates> something, something</coordinates>
        <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
        <RelativePosition ReferenceObject="another object">left right above
below across in front behind center top bottom abut</RelativePosition>
    <material>
        <materialType materialIndexName="wood" materialIndexURL="URL of Material
Index">1</materialType>
            <pattern patternIndexName="wallpaper"
patternIndexURL="URL of XML of pattern Index">3</pattern>
        </material>
        <ColorPalette ColorPaletteIndexName="white"
ColorPaletteIndexURL="URL of palette Index">#ffffff</ColorPalette>
        <window type="window type">
            <model3D>
                <coordinates> something, something</coordinates>
                    <path>url of the x3d model3D</path>
                </model3D>
                <shape>SVG something</shape>
                <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
                    <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
                    <style styleIndexName="modern" styleIndexURL="URL of
XML of style Index">3</style>
                    <material>
                        <materialType materialIndexName="glass" materialIndexURL="URL of
```

```
Material Index">4</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
    </material>
    <frame>
    <material>
        <materialType materialIndexName="metal" materialIndexURL="URL of
Material Index">6</materialType>
            <pattern patternIndexName="style index
name" patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses
the details shapes</pattern>
            </material>
            <ColorPalette ColorPaletteIndexName="black"
ColorPaletteIndexURL="URL of palette Index">#000000</ColorPalette>
            </frame>
        </window>
        <door type="door type">
            <model3D>
                <coordinates> something, something</coordinates>
                <path>url of the x3d model3D</path>
            </model3D>
            <shape>SVG something</shape>
            <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
            <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
            <style styleIndexName="modern" styleIndexURL="URL of
XML of style Index">3</style>
            <material>
                <materialType materialIndexName="glass" materialIndexURL="URL of
Material Index">4</materialType>
                    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
```

```
details shapes</pattern>
    </material>
    <frame>
        <material>
            <materialType materialIndexName="metal" materialIndexURL="URL of
Material Index">6</materialType>
                <pattern patternIndexName="style index
name" patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses
the details shapes</pattern>
                    </material>
                    <ColorPalette ColorPaletteIndexName="black"
ColorPaletteIndexURL="URL of palette Index">#000000</ColorPalette>
                </frame>
            </door>
            <Construction>
                <model3D>
                    <coordinates> x,y,z virtual world</coordinates>
                        <path>url of the x3d model3D</path>
                    </model3D>
                    <dimensions>number,number,number - |x|,|y|,|z| bounding
box dimensions</dimensions>
                        <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
                            <style styleIndexName="style index name"
styleIndexURL="URL of XML of style Index">style code</style>
                                <material>
                                    <materialType materialIndexName="material index name"
materialIndexURL="URL of Material Index">code</materialType>
                                        <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
                                            </material>
                                            <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
```

```
</Construction>
  <accessories type="lighting">
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
    <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
    <style styleIndexName="modern" styleIndexURL="URL of
XML of style Index">3</style>
    <material>
      <materialType materialIndexName="glass" materialIndexURL="URL of
Material Index">6</materialType>
      <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="white"
ColorPaletteIndexURL="URL of palette Index">#ffffff</ColorPalette>
  </accessories>
  </wall>
  <floor>
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
    <style styleIndexName="style index name" styleIndexURL="URL of
XML of style Index">style code</style>
```



```
<material>
  <materialType materialIndexName="marble" materialIndexURL="URL of Material
Index">7</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
  </material>
  <ColorPalette ColorPaletteIndexName="black"
ColorPaletteIndexURL="URL of palette Index">#000000</ColorPalette>
  <furniture type="chair">
    <model3D>
      <coordinates> something, something</coordinates>
      <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
    <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
    <style styleIndexName="modern" styleIndexURL="URL of
XML of style Index">3</style>
    <style styleIndexName="style index name"
styleIndexURL="URL of XML of style Index">style code</style>
  </material>
  <materialType materialIndexName="plastic" materialIndexURL="URL of
Material Index">8</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
  </material>
  <ColorPalette ColorPaletteIndexName="grey"
ColorPaletteIndexURL="URL of palette Index">#CCCCCC</ColorPalette>
  <frame>
</material>
```

```
<materialType materialIndexName="plastic" materialIndexURL="URL of
Material Index">8</materialType>
    <pattern patternIndexName="style index
name" patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses
the details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="grey"
ColorPaletteIndexURL="URL of palette Index">#CCCCCC</ColorPalette>
    </frame>
</furniture>
<accessories type="carpet">
    <model3D>
    <coordinates> something, something</coordinates>
        <path>url of the x3d model3D</path>
    </model3D>
    <shape>SVG something</shape>
    <dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
        <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
        <style styleIndexName="modern" styleIndexURL="URL of
XML of style Index">3</style>
    <material>
    <materialType materialIndexName="material index name"
materialIndexURL="URL of Material Index">code</materialType>
        <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="grey"
ColorPaletteIndexURL="URL of palette Index">#cccccc</ColorPalette>
    </accessories>
    <Construction>
    <model3D>
```

```
<coordinates> x,y,z virtual world</coordinates>
    <path>url of the x3d model3D</path>
</model3D>
<dimensions>number,number,number - |x|,|y|,|z| bounding
box dimensions</dimensions>
    <RelativePosition ReferenceObject="another object">left
right above below across in front behind center top bottom abut</RelativePosition>
    <style styleIndexName="style index name"
styleIndexURL="URL of XML of style Index">style code</style>
    <material>
    <materialType materialIndexName="material index name"
materialIndexURL="URL of Material Index">code</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="palette index name"
ColorPaletteIndexURL="URL of palette Index">code</ColorPalette>
    </Construction>
</floor>
<stair type="stair type">
    <model3D>
<coordinates> something, something</coordinates>
    <path>url of the x3d model</path>
</model3D>
<shape>SVG something</shape>
<dimensions>number,number,number-|x|,|y|,|z| bounding
box</dimensions>
    <RelativePosition ReferenceObject="another object">left right above
below across in front behind center top bottom abut</RelativePosition>
    <style styleIndexName="modern" styleIndexURL="URL of XML of
style Index">3</style>
    <material>
    <materialType materialIndexName="metal" materialIndexURL="URL of Material
```

```
Index">4</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="grey"
ColorPaletteIndexURL="URL of palette Index">#cccccc</ColorPalette>
    <Railing>
    <material>
    <materialType materialIndexName="metal" materialIndexURL="URL of
Material Index">4</materialType>
    <pattern patternIndexName="style index name"
patternIndexURL="URL of XML of pattern Index">pattern index ID that expresses the
details shapes</pattern>
    </material>
    <ColorPalette ColorPaletteIndexName="grey"
ColorPaletteIndexURL="URL of palette Index">#cccccc</ColorPalette>
    </Railing>
    </stair>
</room>
</xml>
```

### 3. Τελικός Πίνακας XML Διαδρομών

Παρακάτω μπορούμε να δούμε τη μορφή στη οποία έχει έρθει το xml πριν δημιουργηθεί το νέο αρχείο από τον αλγόριθμο.

```
/xml/room!-!n/a
/xml/room/@MoodCharacterization!-!calm
/xml/room/ColorPalette!-!#fffccf
/xml/room/ColorPalette/@ColorPaletteIndexName!-!white
/xml/room/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index
/xml/room/RoomCoveringFactor!-!percentage of the covered space (by furniture
```

etc)bounding box of the room - Sum(bounding boxes of the furniture)/bounding box of the room  
/xml/room/ceiling!-!n/a  
/xml/room/ceiling/ColorPalette!-!#ffffcv  
/xml/room/ceiling/ColorPalette/@ColorPaletteIndexName!-!whiter  
/xml/room/ceiling/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/ceiling/Construction!-!n/a  
/xml/room/ceiling/Construction/ColorPalette!-!code  
/xml/room/ceiling/Construction/ColorPalette/@ColorPaletteIndexName!-!palette index name  
/xml/room/ceiling/Construction/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/ceiling/Construction/RelativePosition!-!left right above below across in front behind center top bottom abut  
/xml/room/ceiling/Construction/RelativePosition/@ReferenceObject!-!another object  
/xml/room/ceiling/Construction/dimensions!-!number,number,number - |x|,|y|,|z| bounding box dimensions  
/xml/room/ceiling/Construction/material!-!n/a  
/xml/room/ceiling/Construction/material/materialType!-!Glass  
/xml/room/ceiling/Construction/material/materialType/@materialIndexName!-!3  
/xml/room/ceiling/Construction/material/materialType/@materialIndexURL!-!URL of Material Index  
/xml/room/ceiling/Construction/material/pattern!-!pattern index ID that expresses the details shapes  
/xml/room/ceiling/Construction/material/pattern/@patternIndexName!-!style index name  
/xml/room/ceiling/Construction/material/pattern/@patternIndexURL!-!URL of XML of pattern Index  
/xml/room/ceiling/Construction/style!-!1  
/xml/room/ceiling/Construction/style/@styleIndexName!-!Classic  
/xml/room/ceiling/Construction/style/@styleIndexURL!-!URL of XML of style Index  
/xml/room/ceiling/dimensions!-!2,5,9  
/xml/room/ceiling/lightSource!-!n/a  
/xml/room/ceiling/lightSource/RelativePosition!-!left right above below across in front behind center top bottom abut  
/xml/room/ceiling/lightSource/RelativePosition/@ReferenceObject!-!another object  
/xml/room/ceiling/lightSource/dimensions!-!number,number,number - |x|,|y|,|z| bounding box dimensions  
/xml/room/ceiling/lightSource/lightSourcePalletIndex!-!light source index number  
/xml/room/ceiling/lightSource/lightSourcePalletIndex/@lightSourcePalletIndexName!-!light PalletIndex name  
/xml/room/ceiling/lightSource/lightSourcePalletIndex/@lightSourcePalletIndexURL!-!URL of XML of style Index  
/xml/room/ceiling/lightSource/model3D!-!n/a  
/xml/room/ceiling/lightSource/model3D/coordinates!-!x,y,z virtual world  
/xml/room/ceiling/lightSource/model3D/path!-!url of the x3d model3D  
/xml/room/ceiling/material!-!n/a  
/xml/room/ceiling/material/materialType!-!Woody  
/xml/room/ceiling/material/materialType/@materialIndexName!-!8  
/xml/room/ceiling/material/materialType/@materialIndexURL!-!URL of Material Index  
/xml/room/ceiling/material/pattern!-!pattern index ID that expresses the details shapes  
/xml/room/ceiling/material/pattern/@patternIndexName!-!style index name  
/xml/room/ceiling/material/pattern/@patternIndexURL!-!URL of XML of pattern Index

/xml/room/ceiling/model3D!-!n/a  
/xml/room/ceiling/model3D/coordinates!-!x,y,z virtual world  
/xml/room/ceiling/model3D/path!-!url258 of the x3d model3D  
/xml/room/ceiling/shape!-!Squareijk  
/xml/room/ceiling/style!-!3  
/xml/room/ceiling/style/@styleIndexName!-!Classical  
/xml/room/ceiling/style/@styleIndexURL!-!URL of XML of style Index  
/xml/room/dimension!-!number,number,number-|x|,|y|,|z| bounding box  
/xml/room/floor!-!n/a  
/xml/room/floor/ColorPalette!-!#3cccc  
/xml/room/floor/ColorPalette/@ColorPaletteIndexName!-!Beige  
/xml/room/floor/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/floor/Construction!-!n/a  
/xml/room/floor/Construction/ColorPalette!-!code  
/xml/room/floor/Construction/ColorPalette/@ColorPaletteIndexName!-!palette index name  
/xml/room/floor/Construction/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/floor/Construction/RelativePosition!-!left right above below across in front behind  
center top bottom abut  
/xml/room/floor/Construction/RelativePosition/@ReferenceObject!-!another object  
/xml/room/floor/Construction/dimensions!-!number,number,number - |x|,|y|,|z| bounding box  
dimensions  
/xml/room/floor/Construction/material!-!n/a  
/xml/room/floor/Construction/material/materialType!-!code  
/xml/room/floor/Construction/material/materialType/@materialIndexName!-!material index  
name  
/xml/room/floor/Construction/material/materialType/@materialIndexURL!-!URL of Material  
Index  
/xml/room/floor/Construction/material/pattern!-!pattern index ID that expresses the details  
shapes  
/xml/room/floor/Construction/material/pattern/@patternIndexName!-!style index name  
/xml/room/floor/Construction/material/pattern/@patternIndexURL!-!URL of XML of pattern  
Index  
/xml/room/floor/Construction/model3D!-!n/a  
/xml/room/floor/Construction/model3D/coordinates!-!x,y,z virtual world  
/xml/room/floor/Construction/model3D/path!-!url of the x3d model3D  
/xml/room/floor/Construction/style!-!style code  
/xml/room/floor/Construction/style/@styleIndexName!-!style index name  
/xml/room/floor/Construction/style/@styleIndexURL!-!URL of XML of style Index  
/xml/room/floor/accessories!-!n/a  
/xml/room/floor/accessories/@type!-!chair  
/xml/room/floor/accessories/ColorPalette!-!#000000  
/xml/room/floor/accessories/ColorPalette/@ColorPaletteIndexName!-!black  
/xml/room/floor/accessories/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/floor/accessories/RelativePosition!-!left right above below across in front behind  
center top bottom abut  
/xml/room/floor/accessories/RelativePosition/@ReferenceObject!-!another object  
/xml/room/floor/accessories/dimensions!-!number,number,number-|x|,|y|,|z| bounding box  
/xml/room/floor/accessories/material!-!n/a  
/xml/room/floor/accessories/material/materialType!-!2  
/xml/room/floor/accessories/material/materialType/@materialIndexName!-!Wood  
/xml/room/floor/accessories/material/materialType/@materialIndexURL!-!URL of Material

### Index

/xml/room/floor/accessories/material/pattern!-!pattern index ID that expresses the details shapes

/xml/room/floor/accessories/material/pattern/@patternIndexName!-!style index name

/xml/room/floor/accessories/material/pattern/@patternIndexURL!-!URL of XML of pattern

### Index

/xml/room/floor/accessories/model3D!-!n/a

/xml/room/floor/accessories/model3D/coordinates!-!something, something

/xml/room/floor/accessories/model3D/path!-!url of the x3d model3D

/xml/room/floor/accessories/shape!-!SVG something

/xml/room/floor/accessories/style!-!2

/xml/room/floor/accessories/style/@styleIndexName!-!Modern

/xml/room/floor/accessories/style/@styleIndexURL!-!URL of XML of style Index

/xml/room/floor/dimensions!-!rturtfg fgf

/xml/room/floor/furniture!-!n/a

/xml/room/floor/furniture/@type!-!furniture type

/xml/room/floor/furniture/ColorPalette!-!#ffffff

/xml/room/floor/furniture/ColorPalette/@ColorPaletteIndexName!-!white

/xml/room/floor/furniture/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index

/xml/room/floor/furniture/RelativePosition!-!left right above below across in front behind center top bottom abut

/xml/room/floor/furniture/RelativePosition/@ReferenceObject!-!another object

/xml/room/floor/furniture/dimensions!-!number,number,number-|x|,|y|,|z| bounding box

/xml/room/floor/furniture/material!-!n/a

/xml/room/floor/furniture/material/materialType!-!1

/xml/room/floor/furniture/material/materialType/@materialIndexName!-!Wood

/xml/room/floor/furniture/material/materialType/@materialIndexURL!-!URL of Material

### Index

/xml/room/floor/furniture/material/pattern!-!pattern index ID that expresses the details shapes

/xml/room/floor/furniture/material/pattern/@patternIndexName!-!style index name

/xml/room/floor/furniture/material/pattern/@patternIndexURL!-!URL of XML of pattern

### Index

/xml/room/floor/furniture/model3D!-!n/a

/xml/room/floor/furniture/model3D/coordinates!-!something, something

/xml/room/floor/furniture/model3D/path!-!url of the x3d model3D

/xml/room/floor/furniture/shape!-!SVG something

/xml/room/floor/furniture/style!-!2

/xml/room/floor/furniture/style/@styleIndexName!-!Modern

/xml/room/floor/furniture/style/@styleIndexURL!-!URL of XML of style Index

/xml/room/floor/material!-!n/a

/xml/room/floor/material/materialType!-!11115

/xml/room/floor/material/materialType/@materialIndexName!-!Tilesresdt

/xml/room/floor/material/materialType/@materialIndexURL!-!URL of Material Index

/xml/room/floor/material/pattern!-!pattern index ID that expresses the details shapes

/xml/room/floor/material/pattern/@patternIndexName!-!style index name

/xml/room/floor/material/pattern/@patternIndexURL!-!URL of XML of pattern Index

/xml/room/floor/model3D!-!n/a

/xml/room/floor/model3D/coordinates!-!something, something

/xml/room/floor/model3D/path!-!url of the x3d model3D

/xml/room/floor/shape!-!SVGdfhg something

/xml/room/floor/style!-!78895

/xml/room/floor/style/@styleIndexName!-!Class  
/xml/room/floor/style/@styleIndexURL!-!URLfdghfd of XML of style Index  
/xml/room/functionality!-!n/a  
/xml/room/functionality/MajorFunctionality!-!Bedroom  
/xml/room/functionality/MinorFunctionality/@rankNo!-!a number that shows the priority in minor functionalities  
/xml/room/lighting!-!in lux the light volume of the room (artificial and daylight as well)  
/xml/room/model3D!-!n/a  
/xml/room/model3D/coordinates!-!x,y,z virtual world  
/xml/room/model3D/path!-!url of the x3d model3D  
/xml/room/shape!-!Square  
/xml/room/stair!-!n/a  
/xml/room/stair/@type!-!stair type234  
/xml/room/stair/ColorPalette!-!#ffffff  
/xml/room/stair/ColorPalette/@ColorPaletteIndexName!-!white  
/xml/room/stair/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/stair/Railing!-!n/a  
/xml/room/stair/Railing/ColorPalette!-!#ffffff  
/xml/room/stair/Railing/ColorPalette/@ColorPaletteIndexName!-!White  
/xml/room/stair/Railing/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/stair/Railing/material!-!n/a  
/xml/room/stair/Railing/material/materialType!-!7  
/xml/room/stair/Railing/material/materialType/@materialIndexName!-!Wood  
/xml/room/stair/Railing/material/materialType/@materialIndexURL!-!URL of Material Index  
/xml/room/stair/Railing/material/pattern!-!pattern index ID that expresses the details shapes  
/xml/room/stair/Railing/material/pattern/@patternIndexName!-!style index name  
/xml/room/stair/Railing/material/pattern/@patternIndexURL!-!URL of XML of pattern Index  
/xml/room/stair/RelativePosition!-!left right above below across in front behind center top bottom abut  
/xml/room/stair/RelativePosition/@ReferenceObject!-!another object  
/xml/room/stair/dimensions!-!2,2,3  
/xml/room/stair/material!-!n/a  
/xml/room/stair/material/materialType!-!7  
/xml/room/stair/material/materialType/@materialIndexName!-!Wood  
/xml/room/stair/material/materialType/@materialIndexURL!-!URL of Material Index  
/xml/room/stair/material/pattern!-!pattern index ID that expresses the details shapes  
/xml/room/stair/material/pattern/@patternIndexName!-!style index name  
/xml/room/stair/material/pattern/@patternIndexURL!-!URL of XML of pattern Index  
/xml/room/stair/model3D!-!n/a  
/xml/room/stair/model3D/coordinates!-!something, something  
/xml/room/stair/model3D/path!-!url of the x3d model  
/xml/room/stair/shape!-!SVG 34something  
/xml/room/stair/style!-!7  
/xml/room/stair/style/@styleIndexName!-!Moderning  
/xml/room/stair/style/@styleIndexURL!-!URL of XML of style Index  
/xml/room/wall!-!n/a  
/xml/room/wall/ColorPalette!-!code  
/xml/room/wall/ColorPalette/@ColorPaletteIndexName!-!palette index name  
/xml/room/wall/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index  
/xml/room/wall/RelativePosition!-!dfhgfnh  
/xml/room/wall/RelativePosition/@ReferenceObject!-!another objective



```
/xml/room/wall/accessories!-!n/a
/xml/room/wall/accessories/@type!-!accessory type
/xml/room/wall/accessories/ColorPalette!-!#000000
/xml/room/wall/accessories/ColorPalette/@ColorPaletteIndexName!-!black
/xml/room/wall/accessories/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index
/xml/room/wall/accessories/RelativePosition!-!left right above below across in front behind
center top bottom abut
/xml/room/wall/accessories/RelativePosition/@ReferenceObject!-!another object
/xml/room/wall/accessories/dimensions!-!number,number,number-|x|,|y|,|z| bounding box
/xml/room/wall/accessories/material!-!n/a
/xml/room/wall/accessories/material/materialType!-!6
/xml/room/wall/accessories/material/materialType/@materialIndexName!-!Plastic
/xml/room/wall/accessories/material/materialType/@materialIndexURL!-!URL of Material
Index
/xml/room/wall/accessories/material/pattern!-!pattern index ID that expresses the details
shapes
/xml/room/wall/accessories/material/pattern/@patternIndexName!-!style index name
/xml/room/wall/accessories/material/pattern/@patternIndexURL!-!URL of XML of pattern
Index
/xml/room/wall/accessories/model3D!-!n/a
/xml/room/wall/accessories/model3D/coordinates!-!something, something
/xml/room/wall/accessories/model3D/path!-!url of the x3d model3D
/xml/room/wall/accessories/shape!-!SVG something
/xml/room/wall/accessories/style!-!1
/xml/room/wall/accessories/style/@styleIndexName!-!Classic
/xml/room/wall/accessories/style/@styleIndexURL!-!URL of XML of style Index
/xml/room/wall/dimensions!-!5,6,7
/xml/room/wall/door!-!n/a
/xml/room/wall/door/@type!-!Internal
/xml/room/wall/door/RelativePosition!-!left right above below across in front behind center
top bottom abut
/xml/room/wall/door/RelativePosition/@ReferenceObject!-!another object
/xml/room/wall/door/frame!-!n/a
/xml/room/wall/door/frame/ColorPalette!-!#A78233
/xml/room/wall/door/frame/ColorPalette/@ColorPaletteIndexName!-!Brown
/xml/room/wall/door/frame/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index
/xml/room/wall/door/frame/material!-!n/a
/xml/room/wall/door/frame/material/materialType!-!2
/xml/room/wall/door/frame/material/materialType/@materialIndexName!-!Wood
/xml/room/wall/door/frame/material/materialType/@materialIndexURL!-!URL of Material
Index
/xml/room/wall/door/frame/material/pattern!-!pattern index ID that expresses the details
shapes
/xml/room/wall/door/frame/material/pattern/@patternIndexName!-!style index name
/xml/room/wall/door/frame/material/pattern/@patternIndexURL!-!URL of XML of pattern
Index
/xml/room/wall/door/material!-!n/a
/xml/room/wall/door/material/materialType!-!2
/xml/room/wall/door/material/materialType/@materialIndexName!-!Wood
/xml/room/wall/door/material/materialType/@materialIndexURL!-!URL of Material Index
/xml/room/wall/door/material/pattern!-!pattern index ID that expresses the details shapes
```

```
/xml/room/wall/door/material/pattern/@patternIndexName!-!style index name
/xml/room/wall/door/material/pattern/@patternIndexURL!-!URL of XML of pattern Index
/xml/room/wall/door/model3D!-!n/a
/xml/room/wall/door/model3D/coordinates!-!something, something
/xml/room/wall/door/model3D/path!-!url of the x3d model3D
/xml/room/wall/door/shape!-!SVG something
/xml/room/wall/door/style!-!2
/xml/room/wall/door/style/@styleIndexName!-!Modern
/xml/room/wall/door/style/@styleIndexURL!-!URL of XML of style Index
/xml/room/wall/material!-!n/a
/xml/room/wall/material/materialType!-!2
/xml/room/wall/material/materialType/@materialIndexName!-!Wood
/xml/room/wall/material/materialType/@materialIndexURL!-!URL of Material Index
/xml/room/wall/material/pattern!-!pattern index ID that expresses the details shapes
/xml/room/wall/material/pattern/@patternIndexName!-!style index name
/xml/room/wall/material/pattern/@patternIndexURL!-!URL of XML of pattern Index
/xml/room/wall/model3D!-!n/a
/xml/room/wall/model3D/coordinates!-!something, something
/xml/room/wall/model3D/path!-!url of the x3d model3D
/xml/room/wall/shape!-!SVG somethingggg
/xml/room/wall/window!-!n/a
/xml/room/wall/window/@type!-!windowhjc type
/xml/room/wall/window/RelativePosition!-!left right above below across in front behind
center top bottom abut
/xml/room/wall/window/RelativePosition/@ReferenceObject!-!another object
/xml/room/wall/window/dimensions!-!ghjg
/xml/room/wall/window/frame!-!n/a
/xml/room/wall/window/frame/ColorPalette!-!#ffffff
/xml/room/wall/window/frame/ColorPalette/@ColorPaletteIndexName!-!white
/xml/room/wall/window/frame/ColorPalette/@ColorPaletteIndexURL!-!URL of palette Index
/xml/room/wall/window/frame/material!-!n/a
/xml/room/wall/window/frame/material/materialType!-!2
/xml/room/wall/window/frame/material/materialType/@materialIndexName!-!Wood
/xml/room/wall/window/frame/material/materialType/@materialIndexURL!-!URL of
Material Index
/xml/room/wall/window/frame/material/pattern!-!pattern index ID that expresses the details
shapes
/xml/room/wall/window/frame/material/pattern/@patternIndexName!-!style index name
/xml/room/wall/window/frame/material/pattern/@patternIndexURL!-!URL of XML of
pattern Index
/xml/room/wall/window/model3D!-!n/a
/xml/room/wall/window/model3D/coordinates!-!something, something
/xml/room/wall/window/model3D/path!-!url of the x3d model3D
/xml/room/wall/window/shape!-!SVG hhjhjsomething
/xml/room/wall/window/style!-!3
/xml/room/wall/window/style/@styleIndexName!-!modern
/xml/room/wall/window/style/@styleIndexURL!-!URL of XML of style Index
```

## Βιβλιογραφία

- Case-based reasoning in color matching Springer Berlin / Heidelberg 1997
- SMART: Support Management Automated Reasoning Technology for Compaq Customer Service *Timothy L. Acorn, Compaq Computer Corporation, and Sherry H. Walden, Inference Corporation*
- Clavier: Applying Case-Based Reasoning to Composite Part Fabrication David Hinkle Christopher N.Toomey
- Case-Based Reasoning at General Electric William Cheetham ( [cheetham@crd.ge.com](mailto:cheetham@crd.ge.com) ) Anil Varma ( [varma@crd.ge.com](mailto:varma@crd.ge.com) ) Kai Goebel ( [goebelk@crd.ge.com](mailto:goebelk@crd.ge.com) ) 1 Research Circle Niskayuna, New York 12309
- Lessons Learned During HVAC Installation Ian Watson Dept. of Computer Science University of Auckland Auckland New Zealand [ian@ai-cbr.org](mailto:ian@ai-cbr.org) [www.cs.auckland.ac.nz/~ian](http://www.cs.auckland.ac.nz/~ian)
- Xml annotation of conceptual characteristics in interior decoration.
- Wikipedia: <http://en.wikipedia.org/>
- [http://cbrwiki.fdi.ucm.es/wiki/index.php/Main\\_Page](http://cbrwiki.fdi.ucm.es/wiki/index.php/Main_Page)
- Υλοποίηση αλγορίθμου συνταιριάσματος γράφων. Μαρκάκης Κυριάκος, Δρ. Μαλάμος Αθανάσιος.
- Learning Adaptation Knowledge to Improve Case-Based Reasoning Susan Craw, Nirmalie Wiratung Ray C. Rowe School of Computing, The Robert Gordon University, Aberdeen AB25 1HG, UK Institute of Pharmaceutical Innovation, University of Bradford, Bradford BD7 1DP, UK
- Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, Agnar Aamodt, Enric Plaza.