



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Υλοποίηση της Ιεραρχικής Ανάλυσης Αποφάσεων (ΑΗΡ) για Joomla CMS
Όνοματεπώνυμο Φοιτητή	Θεοδόσης Ιωάννης
Αριθμός Μητρώου	ΜΠΠΛ08061
Επιβλέπων	Δημήτρης Δεσπότης

Ημερομηνία Παράδοσης

Μάιος 2012



Τριμελής Εξεταστική Επιτροπή

Δ. Δεσπότης
Καθηγητής

Δ. Αποστόλου
Επικ. Καθηγητής

Κ. Μεταξιώτης
Επικ. Καθηγητής

1. Περίληψη – Abstract

Η Ιεραρχική Ανάλυση Αποφάσεων είναι μια πολυκριτηριακή μέθοδος η οποία περιγράφει το πρόβλημα ως μια ιεραρχία στόχων και, με βάση αυτή μετατρέπει τις αφηρημένες προτιμήσεις σε αριθμούς. Αναλύεται σε τέσσερα στάδια τα οποία είναι η Ιεραρχική Ανάλυση του προβλήματος, η συλλογή των προτιμήσεων, ο υπολογισμός των επιμέρους προτεραιοτήτων και, τέλος η σύνθεσή τους στις καθολικές προτεραιότητες των διαθέσιμων λύσεων. Η μεθοδολογία, προβλέπει τον έλεγχο της συνέπειας των προτιμήσεων, κατά τον υπολογισμό των αποτελεσμάτων, ενώ καλό είναι να συμπληρώνεται από μια ανάλυση ευαισθησίας, η οποία θα δώσει μια εικόνα της σταθερότητας των προτεραιοτήτων.

Στην εργασία αυτή, η μεθοδολογία υλοποιείται ως επέκταση του δημοφιλούς συστήματος διαχείρισης περιεχομένου Joomla. Αρχικά, εξετάζονται οι απαιτήσεις της υλοποίησης: είναι οι απαιτήσεις που προκύπτουν από τη φύση της μεθοδολογίας κι αυτές που προκύπτουν από το σύγχρονο περιβάλλον του διαδικτύου. Οι προδιαγραφές εξετάζονται κι από λειτουργική σκοπιά.

Οι επεκτάσεις Joomla αναπτύσσονται κατά το πρότυπο Model-View-Controller. Για να αποφευχθεί ο εγκλωβισμός του κώδικα στη συγκεκριμένη πλατφόρμα, επιλέγεται, πέραν αυτού του προτύπου, η μεθοδολογία να υλοποιηθεί σε ένα σχετικά αυτόνομο κομμάτι κώδικα-πυρήνα το οποίο θα επικοινωνεί με το Joomla χωρίς να καταστρατηγεί το γενικότερο πρότυπο.

Τεκμηριώνονται συνοπτικά οι κλάσεις οι οποίες αποτελούν το αυτόνομο αυτό κομμάτι κώδικα: ο εσωτερικός τρόπος λειτουργίας τους, το πως επικοινωνούν μεταξύ τους και το πως επικοινωνούν με το υπόλοιπο σύστημα. Επίσης, τεκμηριώνεται ο τρόπος υλοποίησης των ενεργειών οι οποίες είναι διαθέσιμες στο χρήστη.

Τέλος, παρέχεται ένας λεπτομερής οδηγός χρήσης της επέκτασης.

Analytical Hierarchy Process (AHP) is a multi-criteria decision analysis method (MCDA) which decomposes a problem to a hierarchy of sub-problems, and, thus, becomes able to convert abstract preferences to numbers. It comprises of four steps, these being: hierarchical analysis of the problem, collection of preferences, calculation of partial priorities and, last, composition of partial priorities to the global priorities of available alternatives. AHP specifies a way to measure consistency of preferences, in calculation step, while it should be accompanied by a sensitivity analysis process in order to ensure priorities' stability.

In this project, AHP is implemented as a component of the widespread Joomla CMS. Initially, we focus on implementation requirements; not only those who derive from the methodology itself, but also those that would suit a contemporary web application. Specifications are examined also from a functional point of view.

Joomla components follow the Model-View-Controller (MVC) pattern. A slightly modified implementation path is followed, to avoid platform lock-in: AHP's core is implemented as a semi-separate library of code (a «core») which communicates with Joomla, without compromising, though, the MVC pattern.

Documentation consists of an overview of the core, including code internals, the way the classes communicate between them, and with Joomla. Also, we document implementation and flow of main actions that are available to the user.

Finally, a detailed user-manual is provided.

2. Γενική περιγραφή

2.1. Ιεραρχική Ανάλυση Αποφάσεων

Η Ιεραρχική Ανάλυση Αποφάσεων είναι μια πολυκριτηριακή μέθοδος λήψης αποφάσεων. Η ιδιαιτερότητά της είναι πως δεν αναζητά την τέλεια λύση ανάμεσα σε ένα σύνολο εναλλακτικών, αλλά,

περιγράφει το πρόβλημα ως μια ιεραρχία στόχων· κάθε στόχος, εκπληρώνεται, σε διαφορετικούς βαθμούς, από ένα μικρότερο σύνολο υπο-στόχων, κ.ο.κ. έως ότου βρεθούν οι στόχοι εκείνοι οι οποίοι δεν είναι σκόπιμο να αναλυθούν περαιτέρω. Κάθε διαθέσιμη εναλλακτική αξιολογείται ως προς το κατά πόσο εκπληρώνουν τους θεμελιώδεις στόχους και οι επιμέρους αυτές προτιμήσεις συντίθεται, αναδεικνύοντας έτσι τη λύση η οποία εκπληρώνει καλύτερα τον τελικό στόχο.

Η διαδικασία, αναλύεται σε τέσσερα στάδια:

- η Ιεραρχική Ανάλυση του προβλήματος στην οποία, ο γενικός στόχος αναλύεται σε μικρότερους, πιο εύκολα αντιληπτούς στόχους
- η συλλογή των προτιμήσεων, όπου καθορίζονται, συγκριτικά, τα βάρη μεταξύ στόχων και οι προτεραιότητες μεταξύ εναλλακτικών, κατά ζεύγη
- ο υπολογισμός των επιμέρους προτεραιοτήτων, όπου, τα ζεύγη συντίθενται σε τοπικές προτεραιότητες ή βάρη και, τέλος
- η σύνθεσή τους στις καθολικές προτεραιότητες των διαθέσιμων λύσεων.

Καθώς πρόκειται για μια μεθοδολογία η οποία βασίζεται σε επι μέρους υποκειμενικές κρίσεις, υπάρχει το ενδεχόμενο οι υποκειμενικές κρίσεις αυτές να είναι αντιφατικές μεταξύ τους. Γι' αυτό το λόγο, η μεθοδολογία, προβλέπει τον έλεγχο της συνέπειας των προτιμήσεων, με ένα σαφώς καθορισμένο μέγεθος το οποίο υπολογίζεται στο τρίτο στάδιο, στον υπολογισμό των επι μέρους προτεραιοτήτων.

Τέλος, καμμία μεθοδολογία λήψης αποφάσεων δεν είναι πλήρης, αν δε συνοδεύεται από μια διαδικασία ανάλυσης ευαισθησίας. Πρέπει δηλ. ο αποφασίζων να έχει πληροφόρηση γύρω από το αν η τελική κατάταξη των προτιμήσεων έχει σταθερότητα ή θα μπορούσε να ανατραπεί εύκολα από μια μικρή αλλαγή σε κάποιο κριτήριο. Αυτό είναι ένα στάδιο το οποίο δεν προβλέπεται τυπικά από τη μεθοδολογία, στην πράξη όμως παρέχεται πάντα σε οποιοδήποτε πακέτο λογισμικού την υλοποιεί.

2.2. Ανάλυση

Από προγραμματιστικής πλευράς, γίνεται ένας διαχωρισμός ανάμεσα σε στάδια τα οποία απαιτούν την αλληλεπίδραση του χρήστη και υπολογιστικά στάδια. Τα στάδια τα οποία απαιτούν τη διεπαφή του χρήστη είναι ο ορισμός του προβλήματος, η συλλογή των προτιμήσεων και η ανάλυση ευαισθησίας. Τα υπόλοιπα τμήματα της διαδικασίας, είναι καθαρά υπολογιστικά.

Πριν τη δυνατότητα δημιουργίας προβλήματος όμως, ο χρήστης έχει τη δυνατότητα να επεξεργαστεί κάποιο ήδη υπάρχον πρόβλημα. Επομένως, υπάρχει άλλη μία φάση η οποία δεν έχει να κάνει με τη μεθοδολογία, αλλά με την υλοποίηση, κι αυτή είναι η παρουσίαση των προβλημάτων, η οποία πρέπει να λαμβάνει υπόψη και τα χαρακτηριστικά ασφαλείας.

Ο ορισμός του προβλήματος, με τη σειρά του, αναλύεται σε ενέργειες οι οποίες έχουν να κάνουν με το πρόβλημα, ενέργειες αναφορικές με την ιεραρχία των στόχων και, τέλος, ενέργειες σχετικές με τις εναλλακτικές λύσεις.

Κατά τον ορισμό του προβλήματος, θα πρέπει να τεθούν περιορισμοί ως προς την έκτασή του, δηλ. μέγιστος αριθμός κριτηρίων και εναλλακτικών.

Η συλλογή των προτιμήσεων, μπορεί να γίνει από μια σειρά από sliders τα οποία ο χρήστης θα μπορεί να σύρει προς το μέρος της προτίμησής του. Παράλληλα, μπορεί να υπολογίζεται σε πραγματικό χρόνο ο πίνακας συγκρίσεων, δηλ. να πραγματοποιείται το στάδιο του υπολογισμού των επιμέρους προτεραιοτήτων, ενημερώνοντας παράλληλα το χρήστη για το δείκτη συνέπειας των προτιμήσεων που έχει εισάγει.

Τα υπολογιστικά μέρη πρέπει να γίνεται ανεξάρτητα από τη συλλογή των προτιμήσεων.

Η ανάλυση ευαισθησίας τέλος, θα υλοποιηθεί μέσω ενός γραφήματος το οποίο θα παρουσιάζει τη διακύμανση της τελικής λύσης σε συνάρτηση με τη διακύμανση μιας επιμέρους προτεραιότητας (ή βάρους) που επιλέγει ο χρήστης.

Για λόγους διευκόλυνσης του χρήστη, θα παρασχεθούν κι επιπλέον λειτουργίες, όπως εισαγωγή/εξαγωγή από/σε φύλλο εργασίας, κλωνοποίηση, διαγραφή προβλημάτων κ.ο.κ.

Σε ό,τι αφορά το σχεδιασμό της η λύση θα πρέπει να ακολουθεί μεν τις προδιαγραφές του Joomla, θα πρέπει όμως παράλληλα να μπορούν να επαναχρησιμοποιηθούν μεγάλα της τμήματα. Επίσης, θα πρέπει να δοθεί σημασία στην ασφάλειά της.

Το Joomla CMS, στην έκδοση 1.7, επιτρέπει την επέκτασή του μέσω εξαρτημάτων (components) τα οποία είναι κομμάτια κώδικα, γραμμένα υποχρεωτικά σε PHP τα οποία συνδέονται με το κυρίως σύστημα μέσω του παρεχόμενου Joomla API, και τα οποία, επίσης, ακολουθούν μια τυποποιημένη δομή MVC (Model – View – Controller)

Προκειμένου να εκπληρωθεί ο στόχος της φορητότητας του κώδικα, παράλληλα με την υποχρέωση να ακολουθούνται τα standards του Joomla, το κομμάτι του Μοντέλου από το πρότυπο MVC, θα σπάσει σε δύο τμήματα: το ένα τμήμα θα είναι αυτό που ορίζει το Joomla, ενώ ένα άλλο, ένθετο κομμάτι, θα υλοποιεί τον κορμό της διαδικασίας AHP. Ο κορμός αυτός, θα είναι, σε γενικές γραμμές μεταφέρσιμος και σε άλλα συστήματα.

Η ασφάλεια, σε γενικές γραμμές, διαχειρίζεται από το Joomla. Με αυτό το δεδομένο, θα ακολουθηθούν στην ανάπτυξη γενικοί παραδεδεγμένοι κανόνες ασφάλειας όπως το «Never trust the client». Επίσης, θα υπάρξει φροντίδα ώστε να μην επιβαρύνεται ο εξυπηρετητής από μεγάλο αριθμό υπολογισμών.

Για την ανάπτυξη, είναι φυσικό να χρησιμοποιηθούν τμήματα έτοιμου κώδικα, τα οποία εν προκειμένω, αφορούν το χειρισμό πινάκων, το χειρισμό αρχείων excel και csv, την παραγωγή γραφημάτων και, τέλος, frameworks που εμπλουτίζουν τη διεπαφή του χρήστη.

Επίσης, για την ανάπτυξη χρησιμοποιήθηκε ως σκελετός, έτοιμος κώδικας ο οποίος παρήχθηκε αυτόματα, με βάση τη δομή των πινάκων.

2.3. Ανάπτυξη

Χρησιμοποιούνται 4 πίνακες στη βάση δεδομένων οι οποίοι αφορούν τα προβλήματα, τα κριτήρια, τις εναλλακτικές και τα ζεύγη συγκρίσεων. Από αυτούς, ο πιο περίπλοκος είναι ο τελευταίος, αφού τα πεδία του αναφέρονται, κατά περίπτωση, είτε σε εναλλακτικές είτε σε κριτήρια και, η τιμή που περιέχει κάθε εγγραφή, μπορεί να είναι είτε αριθμητική τιμή είτε συμβολική σταθερά η οποία παραπέμπει στην τιμή μιας κλίμακας.

Βασικό κομμάτι της ανάπτυξης είναι ο υπολογισμός ιδιοτιμής και ιδιοδιανύσματος, υπολογισμός ο οποίος δεν παρέχεται από τον έτοιμο κώδικα. Στην υλοποίηση, επιλέχθηκε η μέθοδος των επαναλήψεων (Power method) με την οποία υπολογίζονται ταυτόχρονα η ιδιοτιμή και το ιδιοδιάνυσμα.

Για την καλύτερη διασφάλιση των δεδομένων οι πίνακες σύγκρισης δε θεωρούνται απλοί πίνακες, αλλά έχει υλοποιηθεί μια κλάση η οποία παρέχει τα ιδιαίτερα χαρακτηριστικά που έχουν οι πίνακες συγκρίσεων (τετράγωνοι πίνακες, στοιχεία διαγωνίου φέρουν την τιμή 1 κ.α.)

Οι εναλλακτικές και τα κριτήρια είναι τα στοιχεία του προβλήματος. Κάθε στοιχείο του προβλήματος φέρει ένα μοναδικό αριθμητικό αναγνωριστικό και ένα όνομα. Οι εναλλακτικές δε χρειάζεται να έχουν κάποια ιδιότητα πέρα από αυτές, οπότε, οι εναλλακτικές είναι η βασική κλάση που αντιπροσωπεύει ένα στοιχείο του προβλήματος. Τα κριτήρια του προβλήματος έχουν υλοποιηθεί κληρονομώντας από τις εναλλακτικές και προσθέτοντας αναφορές σε γονεϊκά και θυγατρικά στοιχεία.

Ωστόσο, η ιεραρχία των κριτηρίων παρουσιάζει κάποια σημεία τα οποία χρειάζονται ιδιαίτερη προσοχή. Κατά την ανάπτυξη, χρειάστηκε να υπάρχει μια μέθοδος επιλογής τους. Επίσης, να υπάρχει συνέπεια στις αναφορές γονέων και τέκνων και να είναι δυνατόν οι κόμβοι του δέντρου να οριστούν με οποιαδήποτε σειρά. Τα προβλήματα αυτά αντιμετωπίζονται, αντίστοιχα, με έναν κατάλογο όλων των κόμβων, και με τη μεσολάβηση ψευδο-κόμβων, οι οποίοι κρατάνε τις σχέσεις των γονέων που δεν έχουν δημιουργηθεί ακόμα. Η απαραίτητη υποχώρηση που έγινε, ήταν, το δέντρο να μην μπορεί να αλλάξει μετά τη δημιουργία του.

Τα μοναδικά αναγνωριστικά χρησιμεύουν, όχι μόνο στην εσωτερική αναγνώριση των στοιχείων, αλλά και στη διεπαφή του χρήστη. Κατά κανόνα, ταυτίζονται με τα μοναδικά αναγνωριστικά της βάσης δεδομένων, αλλά αυτό δεν ισχύει αναγκαστικά. Το κορυφαίο κριτήριο, το οποίο δεν αποθηκεύεται στη βάση δεδομένων, είναι ειδική περίπτωση κι έχει πάντα το αναγνωριστικό 0 (μηδέν).

Η αποθήκευση των τιμών σύγκρισης είναι ένας πολύ ευέλικτος μηχανισμός ο οποίος επιτρέπει την αποθήκευση ζευγών σύγκρισης αλλά και τον απευθείας ορισμό βαρών, με τιμές, οι οποίες, αντίστοιχα, είτε αναφέρονται σε μια κλίμακα, είτε είναι ελεύθερες δεκαδικές τιμές.

3. Ιεραρχική Ανάλυση Αποφάσεων – Γενική περιγραφή

3.1. Εισαγωγικά

Η Ιεραρχική Ανάλυση Αποφάσεων (Δεσπότης, 2002) (Saaty T. , The Analytic Hierarchy and Analytic Network Processes for the Measurement of Intangible Criteria and for Decision-Making, 2005) (Analytic Hierarchy Process) είναι μια πολυκριτηριακή μέθοδος λήψης αποφάσεων. Παρουσιάστηκε το 1977 από τον Thomas L. Saaty και μετά από ένα μεγάλο αριθμό δημοσιεύσεων, θεμελιώθηκε μαθηματικά το 1980, όταν την περιέγραψε στο βιβλίο του «The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation»

Στόχος της μεθόδου δεν είναι να βρεθεί η «σωστή» απόφαση, αλλά να βρεθεί η λύση εκείνη η οποία εκπληρώνει στο μέγιστο βαθμό τους επιθυμητούς στόχους οι οποίοι έχουν τεθεί από τον αποφασίζοντα. Η μέθοδος έχει δεχτεί κριτική, ωστόσο η σαφήνεια κι η απλότητά της την έχουν καταστήσει ως μία από τις ευρύτερα εφαρμοζόμενες μεθόδους λήψης αποφάσεων.

Η ονομασία «Ιεραρχική Ανάλυση Αποφάσεων» αναδεικνύει τα τρία βασικά χαρακτηριστικά της μεθόδου αυτής δηλαδή:

- Είναι Ιεραρχική, αφού το πρόβλημα περιγράφεται ως μια ιεραρχία από γενικούς στόχους, καθένas από τους οποίους αναλύεται σε ειδικότερους υπο-στόχους.
- Είναι Αναλυτική, αφού δομεί το πρόβλημα σε μια λογική βάση και μετατρέπει τις αφηρημένες προτιμήσεις σε αριθμούς.
- Ορίζει συγκεκριμένη διαδικασία Λήψης Αποφάσεων.

Η Ιεραρχική Ανάλυση Αποφάσεων ορίζει τέσσερα στάδια:

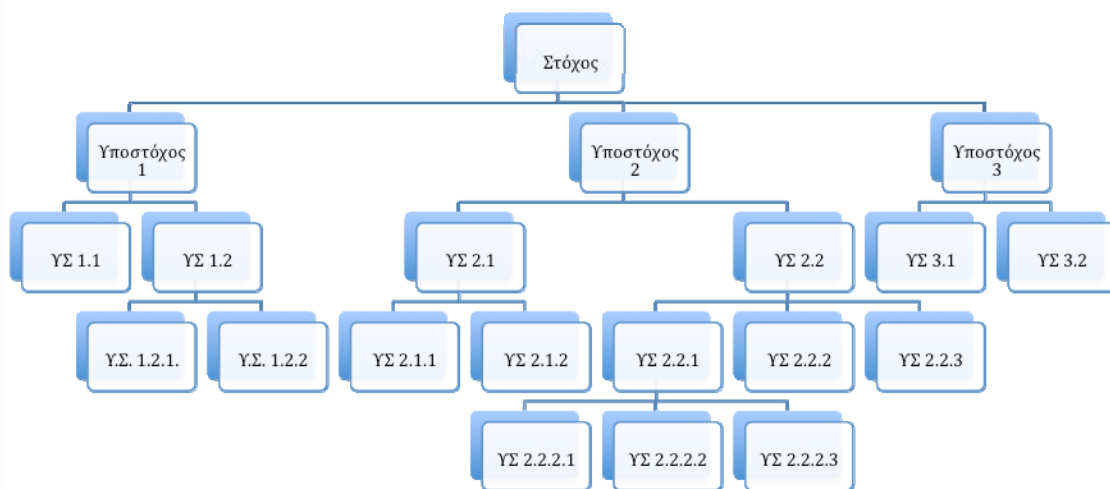
- Ιεραρχική Ανάλυση του Προβλήματος Απόφασης σε Στοιχεία Απόφασης, όπου το πρόβλημα περιγράφεται ως ένα δέντρο από στόχους.
- Συλλογή Προτιμήσεων, όπου ο/οι αποφασίζων/αποφασίζοντες καταγράφουν το βαθμό προτίμησης της κάθε λύσης
- Υπολογισμός Επιμέρους Προτεραιοτήτων, όπου υπολογίζονται οι επιμέρους προτεραιότητες των εναλλακτικών λύσεων στο χαμηλότερο επίπεδο που έχει προσδιοριστεί
- Σύνθεση των Επιμέρους Προτεραιοτήτων σε Γενικές προτεραιότητες, όπου υπολογίζονται οι γενικές προτεραιότητες και προκρίνεται η καταλληλότερη εναλλακτική.

Σημειώνεται ωστόσο πως σε κάθε σύστημα λήψης αποφάσεων είναι χρήσιμο να ακολουθήσει μια Ανάλυση Ευαισθησίας η οποία θα βρει το κατά πόσο η τελική κατάταξη των επιλογών είναι σταθερή ή ευμετάβλητη, κατά πόσο δηλαδή μικρές αλλαγές στην κατανομή των βαρών μπορούν να προκαλέσουν αλλαγές στην τελική κατάταξη. Η Ανάλυση Ευαισθησίας είναι επομένως το πέμπτο στάδιο της μεθόδου η οποία θα ακολουθηθεί.

3.2. Ιεραρχική Ανάλυση του Προβλήματος Απόφασης σε Στοιχεία Απόφασης

Το πρώτο στάδιο της Ιεραρχικής Αναλυτικής Μεθόδου είναι το σημαντικότερο, αφού είναι αυτό το οποίο θα καθορίσει τελικά και την ποιότητα της απόφασης η οποία θα ληφθεί.

Κατά το πρώτο στάδιο, ορίζεται ο απώτερος στόχος του προβλήματος Λήψης Απόφασης. Στη συνέχεια, ο στόχος αυτός αναλύεται σε μια ομάδα από επί μέρους στόχους και, στη συνέχεια το ίδιο γίνεται για κάθε επί μέρους στόχο κι η διαδικασία αυτή επαναλαμβάνεται για κάθε επί μέρους στόχο στα αμέσως χαμηλότερα επίπεδα, μέχρις ότου να διαπιστωθεί πως το πρόβλημα έχει αναλυθεί σε ένα επαρκές επίπεδο.



Εικόνα 1 - Ιεραρχική Ανάλυση Αποφάσεων: μια αφηρημένη ιεραρχία στόχου-υποστόχων

Είναι εύλογο πως ο γενικότερος στόχος θα είναι αρκετά περίπλοκος ώστε να μην μπορεί να προσδιοριστεί άμεσα. Αν π.χ. στόχος είναι η αγορά του «καλύτερου» αυτοκινήτου, τότε θα πρέπει να προσδιοριστεί σε τι ακριβώς συνίσταται η ποιότητα για τον αποφασίζοντα. Στην περίπτωση όπου αποφασίζων είναι ένα άτομο, το σύνολο των κριτηρίων είναι, σαφώς, υποκειμενικό. Κάποιος καταναλωτής μπορεί να ενδιαφέρεται, στο παράδειγμά μας, για το οικολογικό αποτύπωμα ενός αυτοκινήτου και να συμπεριλάβει παράγοντες όπως οι εκπομπές ρύπων, ενώ κάποιος άλλος θα τους παραλείψει τελείως. Η πολυπλοκότητα των προβλημάτων ωστόσο αναδεικνύεται σε προβλήματα όπου ο αποφασίζων είναι μια μεγάλη ομάδα, π.χ. όταν μια εταιρεία θέλει να ανανεώσει το στόλο των αυτοκινήτων της. Εκεί, το οικονομικό τμήμα θα επιμείνει σε παράγοντες κόστους ενώ το τμήμα Δημοσίων Σχέσεων μπορεί να θέσει παράγοντες τελείως άυλους, όπως π.χ. την εικόνα της εταιρείας όπως αυτή προβάλεται από το εταιρικό όχημα. Επομένως, η εμπλοκή διαφορετικών ομάδων ατόμων κατά την ιεραρχική ανάλυση του προβλήματος, αναδεικνύει τις ξεχωριστές πλευρές τις οποίες μπορεί να έχει το πρόβλημα.

3.3. Συλλογή προτιμήσεων για τα Στοιχεία Απόφασης

3.3.1. Δημιουργία πινάκων σύγκρισης

Το στάδιο αυτό μπορούμε να το χωρίσουμε σε δύο φάσεις.

Ο σκοπός της πρώτης φάσης, είναι να συλλεχθούν στοιχεία ώστε να εκτιμηθούν τα βάρη του κάθε κριτηρίου. Προκειμένου να γίνει αυτό, πρέπει πρώτα ο χρήστης να καταχωρίσει τις προτιμήσεις του. Έτσι λοιπόν, δημιουργούνται συγκριτικοί πίνακες σε κάθε επίπεδο του δέντρου των στόχων (χωρίς τις εναλλακτικές λύσεις). Κάθε συγκριτικός πίνακας περιέχει, και στους δύο άξονες, τα στοιχεία απόφασης τα οποία είναι άμεσοι απόγονοι του ίδιου στοιχείου. Με αυτόν τον τρόπο, προκύπτουν ζεύγη σύγκρισης μεταξύ των στοιχείων αυτών. Τα στοιχεία του πίνακα αρχικοποιούνται με την τιμή 1. Ο τρόπος με τον οποίο συμπληρώνονται οι πίνακες αυτοί αναλύεται παρακάτω.

	ΥΣ 2.2.2.1	ΥΣ 2.2.2.2	ΥΣ 2.2.2.3
ΥΣ 2.2.2.1	1	1	1
ΥΣ 2.2.2.2	1	1	1

ΥΣ 2.2.2.3	1	1	1
------------	---	---	---

Πίνακας 1: Παράδειγμα πίνακα προτιμήσεων μεταξύ στόχων

Ο σκοπός της δεύτερης φάσης είναι να αξιολογήσει τις εναλλακτικές λύσεις. Για αυτόν το σκοπό, δημιουργούνται πίνακες οι οποίοι περιέχουν όλες τις εναλλακτικές λύσεις, για κάθε φύλλο του δέντρου των στόχων. Στους πίνακες αυτούς τους πίνακες καταγράφονται οι προτιμήσεις μεταξύ των εναλλακτικών λύσεων, για κάθε κριτήριο (του τελευταίου επιπέδου). Τα στοιχεία του πίνακα αρχικοποιούνται με την τιμή 1.

	<i>Εναλ. Λύση Α</i>	<i>Εναλ. Λύση Β</i>	<i>Εναλ. Λύση Γ</i>
<i>Εναλ. Λύση Α</i>	1	1	1
<i>Εναλ. Λύση Β</i>	1	1	1
<i>Εναλ. Λύση Γ</i>	1	1	1

Πίνακας 2: Παράδειγμα πίνακα προτιμήσεων μεταξύ εναλλακτικών λύσεων

3.3.2. Συμπλήρωση των πινάκων

Οι πίνακες συμπληρώνονται ως εξής:

- σε κάθε στοιχείο του πίνακα συμπληρώνεται η προτίμηση του στοιχείου του άξονα Y έναντι του στοιχείου του άξονα X, ως λόγος.
- Κατά συνέπεια, για κάθε στοιχείο του πίνακα, ισχύει πως, το αντίστροφο στοιχείο, περιέχει την αντίστροφη τιμή (δηλ. $\forall A(x,y): A(y,x) = 1 / A(x,y)$)
- Δεδομένου πως η διαγώνιος του πίνακα (δηλ. όπου $x=y$) συγκρίνει κάθε στοιχείο με τον εαυτό του, η τιμή των στοιχείων αυτών δεν μπορεί να είναι άλλη από 1.
- Συνολικά, επομένως, ο αριθμός των συγκρίσεων που πρέπει να γίνει από τον αποφασίζοντα, είναι $n(n-1)/2$

3.3.3. Κλίμακες

Οι τιμές που αποδίδονται στα στοιχεία εξαρτώνται από την κλίμακα η οποία επιλέγεται. Η κλίμακα, σύμφωνα με τον Saaty, πρέπει να ορίζει τις παρακάτω διαβαθμίσεις προτίμησης:

- I (Ισοδυναμία)
- WP (ασθενής προτίμηση / weak preference)
- SP (ισχυρή προτίμηση / strong preference)
- DP (πολύ ισχυρή προτίμηση / very strong preference)
- AP (απόλυτη προτίμηση / absolute preference)

Το σύνολο των αριθμητικών προτιμήσεων, επομένως, θα είναι το εξής:

$$P = \{I, WP, SP, DP, AP, I/WP, I/SP, I/DP, I/AP\}$$

Η κλίμακα η οποία προτείνεται από τον Saaty, κι η οποία χρησιμοποιείται ευρύτερα, αντιστοιχίζει τις παραπάνω τιμές ως εξής:

- I: 1
- WP: 3
- SP: 5

- DP: 7
- AP: 9

Σημειώνεται πως, στην παραπάνω κλίμακα, είναι δυνατή και η χρήση των αριθμών 2,4,6 και 8, προκειμένου να εκφράσουν ενδιάμεσες προτιμήσεις. Κατά συνέπεια, το σύνολο των διαβαθμίσεων στην κλίμακα αυτή, είναι:

$$P=\{1,2,3,4,5,6,7,8,9,1/2,1/3,1/4,1/5,1/6,1/7,1/8,1/9\}$$

	<i>Εναλ. Λύση Α</i>	<i>Εναλ. Λύση Β</i>	<i>Εναλ. Λύση Γ</i>
<i>Εναλ. Λύση Α</i>	1	1/3	5
<i>Εναλ. Λύση Β</i>	3	1	9
<i>Εναλ. Λύση Γ</i>	1/5	1/9	1

Πίνακας 3: Παράδειγμα συμπλήρωσης πίνακα προτιμήσεων

Η κλίμακα αυτή, όπως περιγράφεται από τον Saaty, είναι η πλέον διαδεδομένη. Ωστόσο, στη βιβλιογραφία έχουν προταθεί και άλλες κλίμακες, με χαρακτηριστικά διαφοροποίησης όπως:

- Μικρότερο ή μεγαλύτερο εύρος επιλογών
- Προσθήκη σταθεράς σε κάθε τιμή του αρχικού συνόλου των αριθμητικών προτιμήσεων στην κλίμακα Saaty.
- Ύψωση σε δύναμη του αρχικού συνόλου των αριθμητικών προτιμήσεων της κλίμακας Saaty
- Εκθετικές κλίμακες

3.4. Εκτίμηση προτεραιοτήτων

Η εκτίμηση των προτεραιοτήτων είναι το πρώτο υπολογιστικό στάδιο. Σκοπός του σταδίου είναι να υπολογιστούν οι σχετικές προτεραιότητες των στοιχείων απόφασης.

Αρχικά, υπολογίζεται η μέγιστη ιδιοτιμή και το αντίστοιχο ιδιοδιάνυσμα του πίνακα. Στο παράδειγμά μας:

	<i>Εναλ. Λύση Α</i>	<i>Εναλ. Λύση Β</i>	<i>Εναλ. Λύση Γ</i>	<i>Ιδιοδιάνυσμα</i>
<i>Εναλ. Λύση Α</i>	1	1/3	5	0,2655
<i>Εναλ. Λύση Β</i>	3	1	9	0,6716
<i>Εναλ. Λύση Γ</i>	1/5	1/9	1	0,0629
Μέγιστη ιδιοτιμή: 3,029				

Το ιδιοδιάνυσμα του πίνακα, είναι οι τιμές των σχετικών βαρών μεταξύ των στοιχείων της απόφασης. Προτού όμως το χρησιμοποιήσουμε, θα πρέπει να γίνει μια εκτίμηση της συνέπειας των στοιχείων που έχει καταχωρήσει ο αποφασίζων.

3.4.1. Συνέπεια των προτιμήσεων

Ιδανικά, κάθε πίνακας καταχώρισης προτιμήσεων, θα έπρεπε να εμφανίζει τις παρακάτω ιδιότητες:

1. Αν $a_{[i,j]} > 1$ και $a_{[j,k]} > 1$ τότε $a_{[i,k]} > 1$ (μεταβατική ιδιότητα, αν δηλ. Το i προτιμάται έναντι του j και το j έναντι του k , τότε και το i πρέπει να είναι προτιμητέο έναντι του k .)
2. $a_{[i,j]}a_{[j,k]}=a_{[i,k]}$: η προτίμηση του i έναντι του k πρέπει να είναι το γινόμενο των προτιμήσεων του i έναντι του j και του j έναντι του k . Αυτό, πρακτικά, είναι αδύνατο, μια και η κλίμακα είναι φραγμένη.

Προκειμένου να υπολογιστεί η συνέπεια του πίνακα με ευκολία, η μέθοδος της Ιεραρχικής Ανάλυσης Αποφάσεων, ορίζει τα εξής μεγέθη:

A Δείκτης Συνέπειας (Consistency Index - CI) ο οποίος υπολογίζεται ως εξής:

$$CI = (\lambda_{\max} - n) / (n-1)$$

Όπου λ_{\max} είναι η τιμή του μέγιστου ιδιοδιανύσματος και n ο αριθμός των στοιχείων απόφασης που συμμετέχουν στον πίνακα.

B Λόγος Συνέπειας (Consistency Ratio - CR) ο οποίος υπολογίζεται από τον τύπο:

$$CR = (CI/ACI) * 100$$

Όπου ACI είναι μια μέση τιμή δεικτών συνέπειας η οποία έχει υπολογιστεί σε τυχαίες προσομοιώσεις όπου τίθενται τυχαία βάρη. Η τιμή η οποία προκύπτει είναι ποσοστό. Γενικά, θεωρείται αποδεκτό το CR να είναι μικρότερο ή ίσο του 10%. Σε αντίθετη περίπτωση, ο αποφασίζων πρέπει να επανεξετάσει τις επιλογές του ώστε να ελαττωθούν οι ασυνέπειες σε αποδεκτά επίπεδα.

Στον παρακάτω πίνακα, δίνονται οι τιμές ACI για αριθμούς στοιχείων από 1 έως 15, όπως έχουν υπολογιστεί από τον Saaty (Saaty T. , The Analytic Hierarchy and Analytic Network Processes for the Measurement of Intangible Criteria and for Decision-Making, 2005) καθώς και από διάφορους μελετητές (Alonso, 2006):

	Saaty	Oak Ridge	Wharton	Golden Wang	Lane, Verдини	Forman	Noble	Tumala, Wan	Aguaron et al	Alonso, Lamata
3	0.52	0.382	0.58	0.5799	0.52	0.5233	0.49	0.500	0.525	0.5245
4	0.89	0.946	0.90	0.8921	0.87	0.8860	0.82	0.834	0.882	0.8815
5	1.11	1.220	1.12	1.1159	1.10	1.1098	1.03	1.046	1.115	1.1086
6	1.25	1.032	1.24	1.2358	1.25	1.2539	1.16	1.178	1.252	1.2479
7	1.35	1.468	1.32	1.3322	1.34	1.3451	1.25	1.267	1.341	1.3417
8	1.40	1.402	1.41	1.3952	1.40		1.31	1.326	1.404	1.4056
9	1.45	1.350	1.45	1.4537	1.45		1.36	1.369	1.452	1.4499
10	1.49	1.464	1.49	1.4882	1.49		1.39	1.406	1.484	1.4854
11		1.576	1.51	1.5117			1.42	1.433	1.513	1.5141
12		1.476		1.5356	1.54		1.44	1.456	1.535	1.5365
13		1.564		1.5571			1.46	1.474	1.555	1.5551
14		1.568		1.5714	1.57		1.48	1.491	1.570	1.5713
15		1.586		1.5831			1.49	1.501	1.583	1.5838

Κατά συνέπεια, στο παράδειγμά μας, οι δύο δείκτες υπολογίζονται ως εξής:

$$CI = (3,029 - 3) / (3-1) = 0,014$$

$$CR = (0,014 / 0,52) * 100 = 2,79\%$$

Άρα, το παράδειγμα μας είναι συνεπές και άρα αποδεκτό.

3.4.2. Υπολογισμός του ιδιοδιανύσματος

Ο υπολογισμός του ιδιοδιανύσματος γίνεται με την επαναληπτική (iterative) μέθοδο, ως εξής:

- Αθροίζονται τα στοιχεία του πίνακα κατά γραμμή
- Το άθροισμα κάθε γραμμής, διαιρείται από το σύνολο των στοιχείων του πίνακα. Έτσι, το διάνυσμα που προκύπτει, έχει ως άθροισμα των στοιχείων του τη μονάδα.
- Υψώνεται ο πίνακας στο τετράγωνο.

- Η διαδικασία επαναλαμβάνεται ώσπου το τελευταίο διάνυσμα που έχει προκύψει, να μη διαφέρει σημαντικά από το προηγούμενο.

3.4.3. Αντικειμενικά δεδομένα

Η χρήση της μεθόδου της Ιεραρχικής Ανάλυσης Αποφάσεων χρησιμοποιείται, γενικά, για προβλήματα τα οποία εμπλέκουν κριτήρια για τα οποία δεν υπάρχει αντικειμενική κρίση. Π.χ. η αισθητική ενός αυτοκινήτου ή η ευχρηστία κάποιου λογισμικού, είναι κριτήρια τα οποία είναι, σε μεγαλύτερο ή μικρότερο βαθμό, υποκειμενικά.

Όταν ένα πρόβλημα εξαρτάται από αντικειμενικά κριτήρια, η Ιεραρχική Ανάλυση Αποφάσεων πιθανότατα να μην είναι η ενδεδειγμένη μεθοδολογία, καθώς μια απλή γραμμική ανάλυση κόστους/οφέλους να μπορούσε να οδηγήσει σε αποφάσεις.

Ωστόσο, δεν μπορεί να αποκλειστεί η εισαγωγή κάπου κριτηρίου το οποίο περιέχει αντικειμενικά δεδομένα. Π.χ. η κατανάλωση ενός αυτοκινήτου ή η τιμή ενός λογισμικού είναι αντικειμενικά μεγέθη, τα οποία – παρ’ όλο που δεν είναι ενδεδειγμένη η εισαγωγή τους – μπορούν να συμπληρωθούν απευθείας.

Τα αντικειμενικά δεδομένα, μπορούν να χρησιμοποιηθούν απευθείας ως βάρη, μόνο όταν έχουν γραμμική σχέση με το όφελος που προκύπτει από αυτά. Π.χ. στο παράδειγμα του αυτοκινήτου, για έναν αποφασίζοντα μπορεί να έχει σημασία η επιτάχυνση του αυτοκινήτου από 0 στα 100 χλμ/ώρα. Εκεί, η μικρότερη τιμή είναι καλύτερη, αλλά και πάλι η σχέση δεν είναι γραμμική, αλλά αντιστρόφως ανάλογη προς το τετράγωνο του χρόνου. Σε αυτήν την περίπτωση, ο αποφασίζων θα πρέπει ο ίδιος να μετατρέψει τα αντικειμενικά δεδομένα σε βάρη με γραμμική, ευθέως ανάλογη σχέση με το όφελος, προτού τα εισάγει.

3.5. Σύνθεση επιμέρους βαρών

Το τελευταίο στάδιο της μεθόδου, είναι επίσης καθαρά υπολογιστικό. Στην πράξη, αυτό που γίνεται είναι συνεχείς πολλαπλασιασμοί πινάκων, από το κατώτερο προς το ανώτερο επίπεδο, μέχρις ότου να προκύψει το τελικό διάνυσμα της σχετικής προτεραιότητας κάθε εναλλακτικής λύσης.

Αναλυτικά, για κάθε φύλλο στο δέντρο των κριτηρίων, προκύπτει ένα διάνυσμα, το οποίο αντιπροσωπεύει τη σχετική εκπλήρωση του κάθε κριτηρίου από την κάθε εναλλακτική λύση. Ένα άλλο διάνυσμα έχει επίσης προκύψει σε κάθε κόμβο τελευταίου επιπέδου του δέντρου των κριτηρίων.

Συγghωνεύοντας τα διανύσματα των εναλλακτικών, για κάθε κριτήριο το οποίο είναι άμεσος απόγονος σε κόμβο τελευταίου επιπέδου, προκύπτει ένας πίνακας του οποίου οι γραμμές είναι όσες και οι εναλλακτικές λύσεις, και οι στήλες όσες και τα κριτήρια που ανήκουν στον κόμβο. Δεδομένου πως το διάνυσμα που έχει προκύψει από τα κριτήρια θα έχει στοιχεία όσα και τα κριτήρια, είναι δυνατός ο πολλαπλασιασμός των δύο πινάκων, από τον οποίο θα προκύψει ένα διάνυσμα με αριθμό στοιχείων όσες και οι εναλλακτικές.

Με αυτόν τον τρόπο, το κατώτατο επίπεδο του δέντρου των κριτηρίων, έχει συγghωνευτεί σε έναν αριθμό διανυσμάτων, τόσα όσα οι κόμβοι του αμέσως ανώτερου επιπέδου. Η διαδικασία επαναλαμβάνεται, μέχρι να φτάσουμε στη ρίζα του δέντρου, όπου και θα προκύψει το τελικό διάνυσμα των σχετικών προτιμήσεων μεταξύ των εναλλακτικών.

3.5.1. Κατανεμημένος και ιδεατός τρόπος

Κατά την εξέλιξη της μεθόδου, και μετά από την κριτική που δέχθηκε η μέθοδος, προτάθηκε ένας νέος τρόπος υπολογισμού με τον οποίο αποφεύγεται το πρόβλημα της διατάραξης της τελικής κατάταξης (rank reversal) από την εισαγωγή ή τη διαγραφή μιας εναλλακτικής λύσης. Ο νέος τρόπος υπολογισμού ονομάστηκε “ideal mode” (ιδεατός τρόπος) ενώ ο αρχικός (ιστορικός) τρόπος, ονομάστηκε «κατανεμημένος».

Ο κατανεμημένος τρόπος, είναι ο τρόπος ο οποίος μόλις εκτέθηκε· κατ’ αυτόν, το κάθε ιδιοδιάνυσμα του πίνακα έχει κανονικοποιηθεί ως προς το άθροισμα των στοιχείων του πίνακα.

Κατά τον ιδεατό τρόπο, το κάθε ιδιοδιάνυσμα, κατά το στάδιο της σύνθεσης, κανονικοποιείται ως προς τη μέγιστη τιμή της στήλης.

Στο παράδειγμά μας, το ιδιοδιάνυσμα τροποποιείται ως εξής:

<i>Κατανεμημένος τρόπος</i>	<i>Ιδεατός τρόπος</i>
0,2655	0,395210367
0,6716	1
0,0629	0,093714741

Σύμφωνα με τον Saaty, (Saaty T. L., 2001), η κατανεμημένη μέθοδος πρέπει να προτιμάται οποτεδήποτε τα κριτήρια επιλογής έχουν δομική σχέση με τις εναλλακτικές λύσεις, ενώ η ιδεατή μέθοδος, όταν τα κριτήρια είναι ανεξάρτητα από τις επιλογές.

3.6. Ανάλυση Ευαισθησίας

Η Ανάλυση Ευαισθησίας, δεν είναι μια τυποποιημένη διαδικασία σε ότι αφορά την Ιεραρχική Ανάλυση Αποφάσεων. Το λογισμικό Expert Choice, προκειμένου να διευκολύνει το χρήστη να εξάγει συμπεράσματα για τη σταθερότητα ενός αποτελέσματος, παρέχει γραφικές παραστάσεις οι οποίες παρουσιάζουν την κατάταξη βάσει της διακύμανσης μιας παραμέτρου.

4. Ανάλυση

4.1. Απαιτήσεις

Οι προδιαγραφές που τέθηκαν για την εφαρμογή ήταν οι εξής:

- Να υλοποιεί τη μεθοδολογία της Ιεραρχικής Ανάλυσης Αποφάσεων για έναν αποφασίζοντα.
- Συγκεκριμένα, ένας χρήστης, δημιουργεί το πρόβλημα απόφασης (πρώτο στάδιο της μεθόδου) και καταχωρεί τις προτιμήσεις του (υπόλοιπα στάδια της μεθόδου).
- Από πλευράς δικαιωμάτων χρηστών:
 - ο δημιουργός του προβλήματος έχει όλες τις δυνατότητες τροποποίησης στο πρόβλημα.
 - οι υπόλοιποι χρήστες μπορούν να δουν το πρόβλημα
- Τα προβλήματα, θα πρέπει να μπορούν να εξαχθούν σε μορφή CSV και Microsoft Excel καθώς επίσης και να εισαχθούν σε αυτήν τη μορφή.
- Η εφαρμογή πρέπει να παρέχει οπωσδήποτε κάποιον τρόπο ανάλυσης ευαισθησίας.
- Δεν υπάρχει ανάγκη καταγραφής «καταστάσεων» του προβλήματος (όπως «οριστικοποίηση» προβλήματος, κριτηρίων, εναλλακτικών).
- Να λειτουργεί μέσα από το λογισμικό CMS Joomla, στην έκδοση 1.7

4.2. Λειτουργικές Προδιαγραφές

Ακολουθεί η ανάλυση της μεθόδου, όπως εφαρμόζονται με γνώμονα τις απαιτήσεις που τέθηκαν. Πιο συγκεκριμένα, εξετάζεται η εφαρμογή από λειτουργικό τρόπο, χωρίς να προχωράμε σε τεχνικές λεπτομέρειες, τις οποίες θα εξετάσουμε στο επόμενο στάδιο, αυτό του σχεδιασμού.

Από λειτουργικής πλευράς, η εφαρμογή διαιρείται στα εξής αυτόνομα στάδια:

1. *Δημιουργία προβλήματος*, η οποία περιλαμβάνει δημιουργία και τροποποίηση των στοιχείων απόφασης (πρώτο στάδιο της μεθόδου) και άλλες λειτουργίες.
2. *Εκτίμηση προβλήματος*, η οποία περιλαμβάνει την καταχώριση των προτιμήσεων (δεύτερο στάδιο) και τον έλεγχο συνέπειας (μέρος του τρίτου σταδίου).
3. *Υπολογιστικό μέρος*: εκτίμηση και σύνθεση των προτεραιοτήτων απόφασης.

4. Ανάλυση ευαισθησίας.

4.2.1. Δημιουργία προβλήματος

Ο χρήστης αρχίζει τη δημιουργία ενός προβλήματος και έχει ως διαθέσιμες ενέργειες:

- Ορισμό/τροποποίηση του τίτλου του προβλήματος. Αυτό θα αποτελεί τη ρίζα του δέντρου των στοιχείων απόφασης
- Επιλογή κατανεμημένου ή ιδεατού τρόπου υπολογισμού. Δεδομένου πως είναι ο δημιουργός του προβλήματος αυτός ο οποίος ορίζει τα κριτήρια και τις εναλλακτικές, είναι εκείνος που γνωρίζει αν τα κριτήρια συνδέονται οργανικά με τις διαθέσιμες εναλλακτικές λύσεις (οπότε πρέπει να επιλεγεί ο κατανεμημένος τρόπος) ή όχι (οπότε πρέπει να επιλεγεί ο οργανικός τρόπος).
- Ενέργειες στο δέντρο στόχων – υποστόχων, οι οποίες είναι:
 - Προσθήκη θυγατρικού κόμβου. Η προσθήκη αδελφού (sibling) κόμβου δεν είναι παρά προσθήκη θυγατρικού κόμβου στο αμέσως ανώτερο επίπεδο.
 - Τροποποίηση κόμβου. Αυτό με τη σειρά του αναλύεται σε:
 - Τροποποίηση ονόματος θυγατρικού κόμβου.
 - Αλλαγή σειράς κόμβων στο ίδιο επίπεδο: πρόκειται καθαρά για θέμα εμφάνισης, αλλά είναι καλό ο χρήστης να μπορεί να ορίσει τη σειρά εμφάνισης των στόχων στην οριζόντια κατεύθυνση.
 - Αλλαγή γονικού κόμβου: επειδή κατά τη δημιουργία του προβλήματος, μπορεί ο χρήστης να αλλάξει γνώμη, καλό είναι να μπορεί να αλλάξει το γονικό κόμβο.
 - Διαγραφή κόμβου. Σε περίπτωση που ο κόμβος περιέχει θυγατρικούς κόμβους, θα εμφανίζεται μήνυμα προειδοποίησης. Μετά από την επιβεβαίωση του χρήστη, μαζί με τον κόμβο, διαγράφονται και οι θυγατρικοί κόμβοι. Σε περίπτωση που ο χρήστης δε θέλει τη διαγραφή των θυγατρικών κόμβων, μπορεί να αποτρέψει την ενέργεια (μήνυμα επιβεβαίωσης) και να αλλάξει το γονικό κόμβο των θυγατρικών του κόμβων (βλπ. ενέργεια «αλλαγή γονικού κόμβου»).
- Ενέργειες σχετικές με τις εναλλακτικές λύσεις, οι οποίες είναι:
 - Προσθήκη εναλλακτικής λύσης.
 - Αλλαγή του ονόματος της εναλλακτικής λύσης
 - Ενδεχομένως, αλλαγή της σειράς εμφάνισης της εναλλακτικής λύσης (βλπ. παρακάτω)
 - Διαγραφή της εναλλακτικής λύσης

Ως προς τη δημιουργία του δέντρου των στόχων, δεν είναι απαραίτητο να υπάρχει κάποιος περιορισμός του αριθμού των επιπέδων στην ιεραρχία, εκτός κι αν αυτό επιβάλλεται από τεχνικούς παράγοντες. Επιπλέον, δεν είναι ανάγκη να συμπίπτει το βάθος ανάλυσης (επίπεδα του δέντρου) για όλους τους κύριους κλάδους (κόμβοι πρώτου επιπέδου) του προβλήματος· τα τυπικά παραδείγματα Ιεραρχικής Ανάλυσης Αποφάσεων δείχνουν προβλήματα στα οποία όλοι οι στόχοι αναλύονται στον ίδιο αριθμό επιπέδων, ωστόσο ο Saaty (Saaty T., 2005) δίνει παραδείγματα στόχων οι οποίοι αναλύονται σε διαφορετικό βάθος.

Ως προς τις εναλλακτικές λύσεις, και ειδικότερα, στη σειρά εμφάνισης, είναι αυτονόητο πως ο χρήστης θα πρέπει να μπορεί να ορίσει τη σειρά με την οποία εμφανίζονται.

Σε ότι αφορά τον αριθμό των στοιχείων απόφασης στο ίδιο επίπεδο, είτε αυτά είναι στόχοι είτε εναλλακτικές λύσεις, σύμφωνα με μελέτες που έχουν γίνει, (Saaty & Ozdemir, 2003), συνίσταται να μην είναι περισσότερα από επτά. Κι ενώ δεν υπάρχει λειτουργικός λόγος να τεθεί κάποιο όριο – τεχνικοί ίσως προκύψουν στη συνέχεια – από λειτουργικής πλευράς δεν είναι απαραίτητο να ληφθούν υπόψη σενάρια στα οποία ο αριθμός των στοιχείων σε ένα επίπεδο να υπερβαίνει κατά πολύ τον αριθμό επτά.

4.2.2. Συμπλήρωση προτεραιοτήτων

Το σύστημα, θα πρέπει να παρέχει στο χρήστη ένα συνοπτικό κατάλογο με τα προβλήματα απόφασης για τα οποία θα πρέπει να συμπληρώσει τις προτιμήσεις του. Αυτά περιλαμβάνουν:

- Προβλήματα τα οποία έχει δημιουργήσει κάποιος άλλος χρήστης, αλλά επιτρέπει να είναι προσβάσιμα από άλλους χρήστες.
- Προβλήματα τα οποία έχει δημιουργήσει ο ίδιος

Ο κατάλογος αυτός, θα πρέπει να περιλαμβάνει τις εξής ελάχιστες πληροφορίες προς διευκόλυνση του χρήστη:

- Τίτλο του προβλήματος
- Όνομα χρήστη του δημιουργού του προβλήματος

Εμφάνιση προβλήματος

Ανοίγοντας ένα συγκεκριμένο πρόβλημα από τον κατάλογο, ο χρήστης βλέπει την εικόνα του προβλήματος. Στην κορυφή, ο τίτλος του προβλήματος θα εμφανίζεται ως ρίζα του προβλήματος.

Από κάτω, εμφανίζονται οι κλάδοι του δέντρου των κριτηρίων. Οι κόμβοι είναι δύο τύπων: οι κατώτατοι κόμβοι (φύλλα του δέντρου των κριτηρίων), οι οποίοι έχουν το απλούστερο κριτήριο (δηλ. κριτήριο το οποίο δεν αναλύεται σε περαιτέρω κριτήρια). Σε αυτούς τους κόμβους, οι συγκρίσεις γίνονται μεταξύ εναλλακτικών. Στους υπόλοιπους κόμβους του δέντρου, οι συγκρίσεις γίνονται μεταξύ των υποκριτηρίων του κόμβου.

Στους κατώτατους κόμβους (φύλλα) παρουσιάζονται τα ζεύγη συγκρίσεων μεταξύ των εναλλακτικών, οι τοπικές προτεραιότητες της κάθε εναλλακτικής καθώς και οι καθολικές προτεραιότητες. Επειδή όμως πρέπει να γνωρίζουμε και τις προτεραιότητες της κάθε εναλλακτικής στο επίπεδο του πατρικού κριτηρίου (το κριτήριο το οποίο προκύπτει από τη σύνθεση των υποκριτηρίων δηλαδή), πάνω από τον εν λόγω πίνακα, εμφανίζεται ένας δεύτερος πίνακας που παρουσιάζει τις τοπικές και καθολικές προτεραιότητες της κάθε εναλλακτικής.

Παράδειγμα πίνακα με ζεύγη σύγκρισης

Κριτήριο: ΥΣ 2.2.2.1						
[-] (Σύμπτυξη)	EA 1	EA 2	EA 3	EA 4	Τοπ. Προτερ.	Καθ. Προτ.
EA 1	1,0000	0,3333	1,0000	5,0000	0,2046	0,2046
EA 2	3,0000	1,0000	3,0000	9,0000	0,5439	0,5439
EA 3	1,0000	0,3333	1,0000	5,0000	0,2046	0,2046
EA 4	0,2000	0,1111	0,2000	1,0000	0,0469	0,0469

Πίνακας 4: Σκαρίφημα ανεπτυγμένης μορφής

Ορισμός προτεραιοτήτων

Ο ορισμός προτεραιοτήτων πρέπει να γίνεται με εύληπτο και κατανοητό για το χρήστη τρόπο, ο οποίος να εγγυάται την ακεραιότητα των δεδομένων. Πιο συγκεκριμένα:

- Ο χρήστης δε θα πρέπει να καλείται να κάνει περισσότερες συγκρίσεις από τις απαραίτητες $(n(n-1)/2)$
- θα πρέπει να εισάγει τιμές οι οποίες ανήκουν στην κλίμακα που χρησιμοποιείται
- θα πρέπει να εμποδίζεται να αλλάξει αριθμούς για τους οποίους δεν έχει νόημα η αλλαγή τους (π.χ. τους αριθμούς της διαγωνίου, ή να διορθώσει το βάρος w_{13} και μετά το βάρος w_{31}).
- Επίσης, θα πρέπει να είναι σαφής η σημασία του αριθμού: ο χρήστης μπορεί να μερδευτεί και να αγνοήσει προς στιγμήν το ότι, αριθμός μεγαλύτερος του 1 σημαίνει προτίμηση για το σημείο στον άξονα Y.

Για τους παραπάνω λόγους, η απευθείας εισαγωγή αριθμών δεν είναι η κατάλληλη μέθοδος. Θα ήταν πολύ καλύτερο να υπάρχει κάποιο interface το οποίο θα βοηθούσε τον χρήστη να δηλώσει με όσο το δυνατόν μεγαλύτερη ακρίβεια την επιλογή του.

Η λύση, θα μπορούσε να είναι μια σειρά από sliders τα οποία ο χρήστης σύρει προς τη μεριά της προτίμησής του.

Σύρετε τον δείκτη προς το μέρος της λύσης που προτιμάτε.

Πλησιάστε το δείκτη τόσο κοντά, όσο προτιμότερη είναι η λύση κατά τη γνώμη σας.

	Προτιμάται η Α				Ισοδύναμες	Προτιμάται η Β					
	Απόλυτη Προτίμηση	Μεγάλη Προτίμηση	Μέση Προτίμηση	Μικρή προτίμηση	Ισοδύναμες Λύσεις	Μικρή προτίμηση	Μέση Προτίμηση	Μεγάλη Προτίμηση	Απόλυτη Προτίμηση		
A					^					B	
A					^					Γ	
A	^									Δ	
B					^					Γ	
B	^										Δ
Γ					^					Δ	

[Λόγος Συνέπειας: 1,23%]

Πίνακας 5: Σκαρίφημα οθόνης εισαγωγής προτιμήσεων

Σε ό,τι αφορά το Λόγο Συνέπειας, θα φαινόταν μια καλή ιδέα να γίνεται υπολογισμός του σε πραγματικό χρόνο, δηλ. να υπολογίζεται καθώς ο χρήστης σύρει τον δείκτη πάνω στο slider. Ωστόσο, από πλευράς φιλικότητας προς το χρήστη, αυτό μπορεί να μην είναι τόσο καλό, διότι μπορεί να αποσπάσει την προσοχή του χρήστη από το πρόβλημα και ο χρήστης να προσέχει την τιμή του Λόγου Συνέπειας.

Επίσης, η αποτροπή της αποθήκευσης του πίνακα συγκρίσεων, λόγω υπερβολικά υψηλού Λόγου Συνέπειας, μπορεί να δημιουργήσει προβλήματα στην ευχρηστία του συστήματος: σε ένα ρεαλιστικό σενάριο όπου ο χρήστης έχει 9 στοιχεία απόφασης, άρα να κάνει 36 συγκρίσεις, η πιθανότητα να πέσει ο χρήστης σε ασυνέπεια είναι μεγαλύτερη. Αν το σύστημα όμως δεν επιτρέπει στο χρήστη να αποθηκεύσει τις προτιμήσεις του, σε έκτακτες περιπτώσεις οι οποίες δε γίνεται να ελεγχθούν από το σύστημα (διακοπή σύνδεσης δικτύου, πτώση ρεύματος κ.α.) ο χρήστης μπορεί να χάσει ολόκληρο τον πίνακα συγκρίσεων και θα πρέπει να κάνει όλες τις συγκρίσεις από την αρχή.

Για τους παραπάνω λόγους, είναι προτιμότερο ο Λόγος Συνέπειας να επιστρέφεται ως αποτέλεσμα κατά την αποθήκευση του πίνακα, με ειδική επισήμανση (όπως κόκκινο χρώμα) όταν αυτός ξεπερνά τα αποδεκτά επίπεδα.

4.2.3. Υπολογιστικό Μέρος

Το υπολογιστικό μέρος δε χρειάζεται την αλληλεπίδραση του χρήστη.

Από λειτουργικής πλευράς, θα πρέπει πάντα τα δεδομένα που εμφανίζονται στο χρήστη να είναι επικαιροποιημένα: δεν είναι δυνατόν, π.χ. να έχουν τροποποιηθεί οι τιμές σε έναν πίνακα συγκρίσεων και στη συνολική εικόνα του προβλήματος να εμφανίζονται οι προτεραιότητες που είχαν γίνει πριν την τροποποίηση αυτή.

Το άλλο άκρο, ο υπολογισμός σε πραγματικό χρόνο, δεν είναι καλή ιδέα. Από τη μία, θα δώσει μεγαλύτερη ευχέρεια στο χρήστη να αλλοιώσει τις προτιμήσεις κατά το δοκούν (συνεχείς αλλαγές προτεραιοτήτων μέχρι να βγει η επιθυμητή επιλογή) αφ' ετέρου (και πιο σημαντικό) σε μεγάλα προβλήματα είναι πιθανό να φορτώσει τον υπολογιστή με μεγάλο αριθμό υπολογισμών.

4.3. Ανάλυση Ευαισθησίας

Η Ανάλυση Ευαισθησίας δεν είναι μια τυποποιημένη διαδικασία. Δεδομένου πως ο σκοπός της είναι να βοηθήσει το χρήστη να εντοπίσει τα πιο ευαίσθητα κριτήρια (κριτήρια τα οποία με μικρή αλλαγή μπορούν να προκαλέσουν αλλαγή της σειράς κατάταξης, για την πρώτη επιλογή ή για όλες τις επιλογές), μια καλή μέθοδος είναι να επιλέγει ο χρήστης το κριτήριο που τον ενδιαφέρει και να παρουσιάζονται οι αλλαγές στην καθολική (ή στην τοπική) κατάταξη, για διαφορετικές τιμές στο βάρος του κριτηρίου, στο διάστημα $[0,1]$, με ανάλογη αύξηση ή μείωση των υπολοίπων τοπικών βαρών.

Π.χ., έστω πως έχουμε το παρακάτω διάνυσμα βαρών σε έναν κόμβο:

A (w_1)	0,2046
B (w_2)	0,5439
Γ (w_3)	0,2046
Δ (w_4)	0,0469

Και θέλουμε να ελέγξουμε την ευαισθησία ως προς τις αυξομειώσεις του κριτηρίου B. Οι τιμές που θα ληφθούν υπόψη για τα άλλα βάρη, αν το βάρος του κριτηρίου B κυμανθεί στο διάστημα $[0,1]$, θα τηρούν την αναλογία ανάμεσα στα υπόλοιπα κριτήρια, για τη διαφορά της τιμής του B από τη μονάδα. Θα έχουμε δηλαδή:

$$w'_1 = (1 - w_2) \cdot \frac{w_1}{1 - w_2}$$

$$w'_2 \in [0,1]$$

$$w'_3 = (1 - w_2) \cdot \frac{w_3}{1 - w_2}$$

$$w'_4 = (1 - w_2) \cdot \frac{w_4}{1 - w_2}$$

Δίνεται το παράδειγμα των τιμών αυτών για το παραπάνω διάνυσμα:

A	0,4486	0,4037	0,3589	0,3140	0,2691	0,2243	0,1794	0,1346	0,0897	0,0449	0,0000
B	0,0000	0,1000	0,2000	0,3000	0,4000	0,5000	0,6000	0,7000	0,8000	0,9000	1,0000
Γ	0,4486	0,4037	0,3589	0,3140	0,2691	0,2243	0,1794	0,1346	0,0897	0,0449	0,0000
Δ	0,1029	0,0926	0,0823	0,0720	0,0617	0,0514	0,0411	0,0309	0,0206	0,0103	0,0000

Έστω πως το διάνυσμα των τοπικών βαρών των εναλλακτικών λύσεων, είναι το εξής:

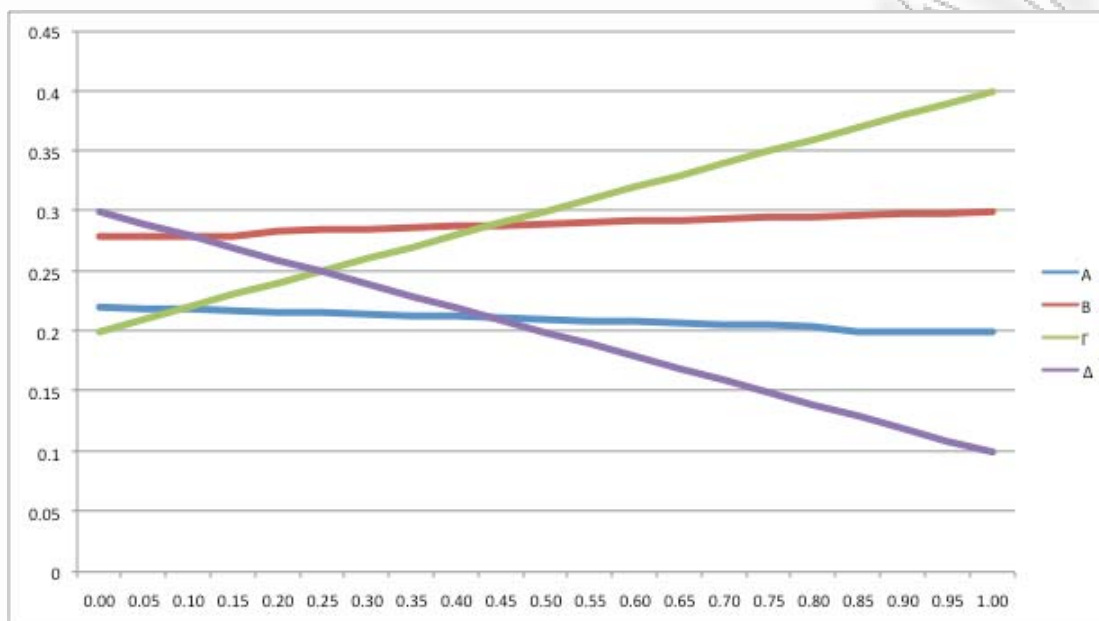
	A	B	Γ	Δ
ΕΛ 1	0,10	0,20	0,30	0,40
ΕΛ 2	0,20	0,30	0,40	0,10
ΕΛ 3	0,30	0,40	0,10	0,20
ΕΛ 4	0,40	0,10	0,20	0,30

Θα προκύψουν τα παρακάτω βάρη για τις λύσεις:

Τιμή κριτηρίου B	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
A	0,2206	0,2185	0,2165	0,2144	0,2123	0,2103	0,2082	0,2062	0,2041	0,2021	0,2000
B	0,2794	0,2815	0,2835	0,2856	0,2877	0,2897	0,2918	0,2938	0,2959	0,2979	0,3000
Γ	0,2000	0,2200	0,2400	0,2600	0,2800	0,3000	0,3200	0,3400	0,3600	0,3800	0,4000

Δ	0,3000	0,2800	0,2600	0,2400	0,2200	0,2000	0,1800	0,1600	0,1400	0,1200	0,1000
---	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Τα οποία μπορούν να απεικονιστούν σε ένα γράφημα όπως το παρακάτω:



Σχήμα 1: Σκαρίφημα γραφήματος απεικόνισης Ανάλυσης Ευαισθησίας

Από τον πίνακα αυτόν προκύπτει πως, εάν το βάρος του κριτηρίου μεταβληθεί, από 0,54 σε 0,45, θα προκύψουν σημαντικές αλλαγές στην προτεραιότητα προτιμήσεων ανάμεσα στις εναλλακτικές λύσεις.

4.3.1. Άλλες Λειτουργίες

Εισαγωγή / εξαγωγή σε Φύλλο Εργασίας: Οποιοσδήποτε χρήστης, θα πρέπει να μπορεί να έχει τη δυνατότητα εξαγωγής του προβλήματος σε φύλλο εργασίας, ώστε να το επεξεργαστεί περαιτέρω σε κάποιο πρόγραμμα όπως Microsoft Excel, OpenOffice Calc, κ.α. Η εισαγωγή από φύλλο εργασίας, θα γίνεται αφού πρώτα ελεγχθεί το πρόβλημα που περιγράφεται στο φύλλο εργασίας το οποίο εισάγεται· εάν συμπίπτει με το πρόβλημα το οποίο εξετάζει ο χρήστης, αυτή θα είναι δυνατή. Εάν όχι, τότε ο χρήστης θα ερωτάται για το αν θέλει να εισαχθεί το φύλλο σε νέο πρόβλημα.

Διαγραφή προβλημάτων: Επειδή, με το πέρασμα του χρόνου, θα συσσωρευτούν αρκετά προβλήματα τα οποία μπορεί να καταλαμβάνουν χώρο ή να δημιουργούν προβλήματα στους χρήστες (πολυπληθής κατάλογος προβλημάτων), θα πρέπει να δοθεί στο διαχειριστή του συστήματος η δυνατότητα να διαγράφει προβλήματα. Μία τέτοια οθόνη διαχείρισης, θα πρέπει να περιέχει την ονομασία του προβλήματος, το χρήστη που το δημιούργησε, μια περιγραφή της κατάστασης του προβλήματος (η οποία θα προκύπτει από την εξέταση των status στον κάθε χρήστη ο οποίος ασχολείται με το πρόβλημα) και, τέλος, την ημερομηνία τελευταία φορά που κάποιος χρήστης έκανε την οποιαδήποτε ενέργεια σε αυτό το πρόβλημα. Η διαγραφή μπορεί να είναι είτε για μεμονωμένα προβλήματα, είτε μαζικά, π.χ. διαγραφή όλων των προβλημάτων τα οποία δεν έχουν ανοιχτεί για 90 ημέρες (όπου την τιμή «90» θα την επιλέγει ο χρήστης παραμετρικά).

Κλωνοποίηση προβλήματος: Αν και δεν υπάρχει στις απαιτήσεις, είναι ωφέλιμο, οι χρήστες, εφ' όσον μπορούν να δουν το πρόβλημα ενός άλλου χρήστη, να μπορούν να το αντιγράψουν και να δώσουν τις δικές τους απαντήσεις ή να το τροποποιήσουν όπως θέλουν

5. Σχεδιασμός

5.1. Γενικές αρχές σχεδιασμού

Πριν προχωρήσουμε στον τεχνικό σχεδιασμό, θα θέσουμε τους στόχους της υλοποίησης την οποία πρόκειται να αναπτύξουμε, κι οι οποίοι είναι οι παρακάτω:

1. Η λύση θα υλοποιηθεί για το λογισμικό CMS Joomla, στην έκδοση 1.7. Η υλοποίηση πρέπει επομένως να ακολουθεί πιστά τις προδιαγραφές ανάπτυξης επεκτάσεων του Joomla, ώστε να εντάσσεται αρμονικά σε αυτό.
2. Φορητότητα: ο κώδικας θα πρέπει να μπορεί να επαναχρησιμοποιηθεί όσο το δυνατόν ευρύτερα. Γι' αυτό το λόγο, η λύση θα αρθρώνεται σε όσο το δυνατόν πιο αυτόνομα τμήματα, τα οποία θα μπορούν να επαναχρησιμοποιηθούν σε άλλες πλατφόρμες.
3. Δεδομένου πως μιλάμε για ένα σύστημα το οποίο θα είναι διαθέσιμο μέσω internet, θα πρέπει να δοθεί ιδιαίτερη σημασία στην ασφάλεια του προγράμματος.

Οι αρχές αυτές κι οι συνέπειές τους αναλύονται αμέσως παρακάτω:

5.1.1. Απαιτήσεις πλατφόρμας Joomla CMS

Το λογισμικό Joomla είναι ένα Σύστημα Διαχείρισης Περιεχομένου ιστοτόπων. (Content Management System – CMS). Παρέχει τη δυνατότητα διαχείρισης ενός ιστοτόπου και παρουσίασης των σελίδων του, σύμφωνα με τις ανάγκες του ιδιοκτήτη του ιστοτόπου. (Lanham & Kennard, 2010)

Για τη διαχείριση του περιεχομένου, χρησιμοποιεί τη βάση δεδομένων MySQL με συνιστώμενη έκδοση την 4.1.x

Είναι γραμμένο στη γλώσσα προγραμματισμού PHP. Για την έκδοση 1.7, συνιστάται η έκδοση 5.3 της γλώσσας αυτής.

Η γλώσσα προγραμματισμού PHP πρέπει να συνεργάζεται με έναν εξυπηρετητή ιστοσελίδων (http server) προκειμένου να εκπληρώσει το σκοπό του. Το Joomla έχει επιβεβαιωθεί πως δουλεύει σωστά με τους εξυπηρετητές Apache και Microsoft IIS.

Το Joomla είναι επεκτάσιμο. Εσωτερικά, είναι αρθρωμένο σε διαφορετικά στρώματα: την υποδομή, η οποία χωρίζεται στις βιβλιοθήκες και την κυρίως υποδομή (framework), το στρώμα της εφαρμογής και το στρώμα των επεκτάσεων. Δε χρειάζεται να επεκταθούμε στις λεπτομέρειες, αρκεί να αναφερθεί πως η εφαρμογή παρέχει το απαραίτητο υπόστρωμα για την λειτουργία των επεκτάσεων.

Οι επεκτάσεις, οι οποίες μας ενδιαφέρουν, είναι προγράμματα τυποποιημένα σύμφωνα με συγκεκριμένες προδιαγραφές, τα οποία επικοινωνούν με την εφαρμογή και εμπλουτίζουν τη λειτουργικότητα του CMS.

Οι κυριότεροι τύποι των επεκτάσεων, είναι οι εξής:

- Components (εξαρτήματα): είναι ο κυριότερος τύπος επεκτάσεων, κι εμφανίζονται στον κύριο χώρο της σελίδας. Είναι αυτά τα οποία, κατά κανόνα, παρέχουν την επιπλέον λειτουργικότητα.
- Modules (ενθέματα): καταλαμβάνουν μικρό χώρο, έξω από τον κύριο χώρο της σελίδας κι έχουν συνήθως πληροφοριακό χαρακτήρα ή παρέχουν περιορισμένη λειτουργικότητα.
- Plugins (πρόσθετα): πρόκειται για τύπο επεκτάσεων το οποίο ενεργοποιείται σε συγκεκριμένα συμβάντα (events) για να τροποποιήσουν τη λειτουργία κάποιου εξαρτήματος ή κάποιου ενθέματος.

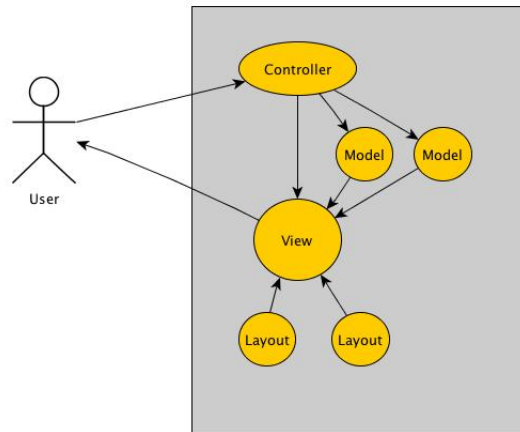
Οι επεκτάσεις του Joomla χτίζονται πάνω στο πρότυπο MVC (Model-View-Controller), ένα από τα πλέον διαδεδομένα πρότυπα στην ανάπτυξη λογισμικού. Σύμφωνα με αυτό το πρότυπο, μια εφαρμογή χωρίζεται σε τρεις κύριες λειτουργίες:

- Το Μοντέλο (**Model**) το οποίο διαχειρίζεται την ανάκτηση και την αποθήκευση των δεδομένων, ανεξαρτήτως του μέσου που χρησιμοποιείται για την αποθήκευσή τους (βάσεις δεδομένων, αρχεία κ.α.). Αποστολή του είναι να μετατρέψει τα αποθηκευμένα δεδομένα σε οντότητες. Αυτό σημαίνει πως, δεν ασχολείται απλώς με τη μετατροπή των δεδομένων, αλλά και αντιστοιχίζει τις μεθόδους των οντοτήτων σε ενέργειες πάνω στα αποθηκευμένα δεδομένα.
- Η Όψη (**View**) αναλαμβάνει την παρουσίαση των δεδομένων στην επιθυμητή μορφή. Μια όψη μπορεί να παρουσιάσει τα ίδια δεδομένα με διαφορετικό τρόπο (κατάλογος πολλών αντικειμένων ή ένα αντικείμενο) ή σε διαφορετικές μορφές (έγγραφο HTML, αρχείο XML ή

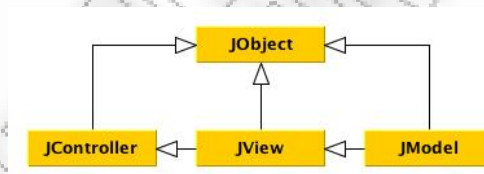
έγγραφο PDF). Σε κάθε περίπτωση όμως, αναλαμβάνει μόνο την παρουσίαση των δεδομένων, όχι την τροποποίησή τους.

- Ο Ελεγκτής (**Controller**) είναι ο κυβερνήτης των άλλων δύο. Παραλαμβάνει την είσοδο του χρήστη και τη δρομολογεί προς την κατάλληλη απάντηση, καλώντας τις απαραίτητες μεθόδους του Μοντέλου είτε δημιουργώντας μια νέα Όψη. Ο Ελεγκτής δεν εκτελεί απευθείας ενέργειες στα δεδομένα. Απλώς καλεί τις μεθόδους του μοντέλου.

Οι σχέσεις μεταξύ των οντοτήτων, όπως τις ορίζει το πρότυπο MVC, φαίνονται στο ακόλουθο Use Case διάγραμμα:



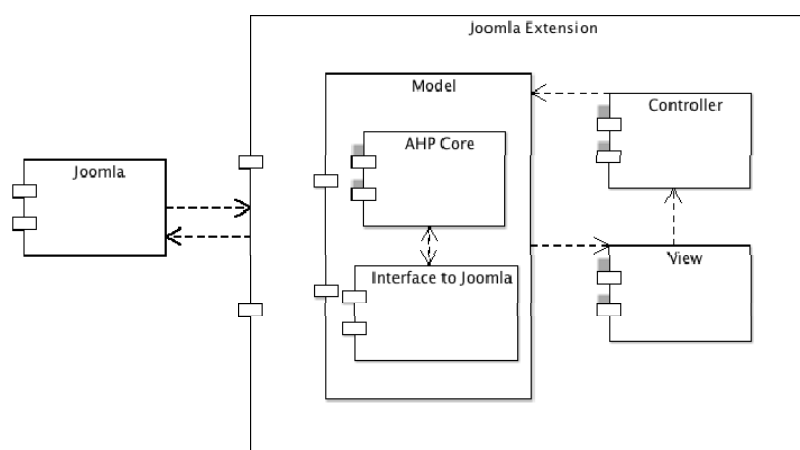
Η υλοποίηση του προτύπου στο Joomla, φαίνεται στο παρακάτω διάγραμμα, το οποίο δείχνει τη σχέση μεταξύ των αφηρημένων κλάσεων τις οποίες παρέχει το Joomla:



5.1.2. Φορητότητα

Ο στόχος αυτός είναι, σε γενικές γραμμές, αντιφατικός ως προς το γενικό στόχο, ο οποίος είναι η υλοποίηση μιας εφαρμογής η οποία θα τρέχει αποκλειστικά μέσω Joomla. Ωστόσο, έχουμε το περιθώριο να απομονώσουμε τον πυρήνα της εφαρμογής σε ένα αυτόνομο κομμάτι, μέσα στο Μοντέλο.

Με άλλα λόγια, θα διαχωρίσουμε το Μοντέλο, σε δύο τμήματα. Το ένα τμήμα, θα υλοποιεί τη μέθοδο της Ιεραρχικής Ανάλυσης Αποφάσεων, χωρίς να υλοποιεί τη διεπαφή με το χρήστη, ούτε με την αποθήκευση των δεδομένων. Για την αποθήκευση των δεδομένων, θα επαφίεται στο δεύτερο τμήμα του Μοντέλου, το οποίο θα αλληλεπιδρά με τη Βάση Δεδομένων από τις τυποποιημένες βιβλιοθήκες του Joomla, ή θα εισάγει/εξάγει δεδομένα σε μορφή φύλλου εργασίας. Η διεπαφή με το χρήστη, θα γίνεται μέσω της όψης και του ελεγκτή.



Σχήμα 2: Γενική περιγραφή αρχιτεκτονικής συστήματος

Η φορητότητα επιτυγχάνεται με αυτόν τον τρόπο ως εξής: οποιοσδήποτε θα μπορεί να χρησιμοποιήσει τον πυρήνα της εφαρμογής, καλώντας ένα συγκεκριμένο API (το ίδιο που θα χρησιμοποιεί ο controller της εφαρμογής μας) και θα πρέπει επίσης να παρέχει μια κλάση η οποία θα υλοποιεί μεθόδους αποθήκευσης / ανάκτησης των οντοτήτων που ορίζει ο πυρήνας.

5.1.3. Ασφάλεια

Το Joomla, παρέχει επαρκείς λειτουργίες για απλά, αλλά ουσιαστικά θέματα ασφάλειας όπως η πρόσβαση των χρηστών, το να μην μπορεί κανείς να καλέσει απευθείας ένα αρχείο php αν αυτό προηγουμένως δεν έχει δρομολογηθεί από το σύστημα κ.α. Ωστόσο, κάθε νέα εφαρμογή θα πρέπει να μεριμνεί για τα ειδικά προβλήματα ασφάλειας τα οποία μπορεί να δημιουργήσει.

Στην παρούσα υλοποίηση, θα προσέξουμε ιδιαίτερα:

- Την υποχρεωτική επικύρωση των δεδομένων, πριν την οποιαδήποτε ενέργεια, στον εξυπηρετητή (αρχή γνωστή ως «Never trust the client»)
- Την αποφυγή δημιουργίας υπερβολικού φόρτου εργασίας στον εξυπηρετητή

Η αρχή «Never trust the client» προκύπτει από το γεγονός πως, ακόμα και πιστοποιημένοι χρήστες, αν είναι κακόβουλοι, μπορούν να παρακάμψουν το φυλλομετρητή (και, κατά συνέπεια, οσεςδήποτε διαδικασίες επικύρωσης έχουν υλοποιηθεί σε αυτόν μέσω, π.χ. Javascript) και να στείλουν αιτήματα τα οποία δεν μπορούν να εκτελεστούν, με σκοπό να μπλοκάρουν το σύστημα.

Προκειμένου να αποφευχθεί αυτό, κάθε είσοδος από πλευράς του χρήστη, θα πρέπει να επικυρώνεται, με απλούς τρόπους, στην πλευρά του εξυπηρετητή την οποία ο κακόβουλος χρήστης δεν μπορεί να παρακάμψει, ώστε να αποφεύγεται η δρομολόγηση εργασιών από πλευράς του υπολογιστή οι οποίες μπορούν να υπερφορτώσουν το σύστημα.

6. Ανάπτυξη

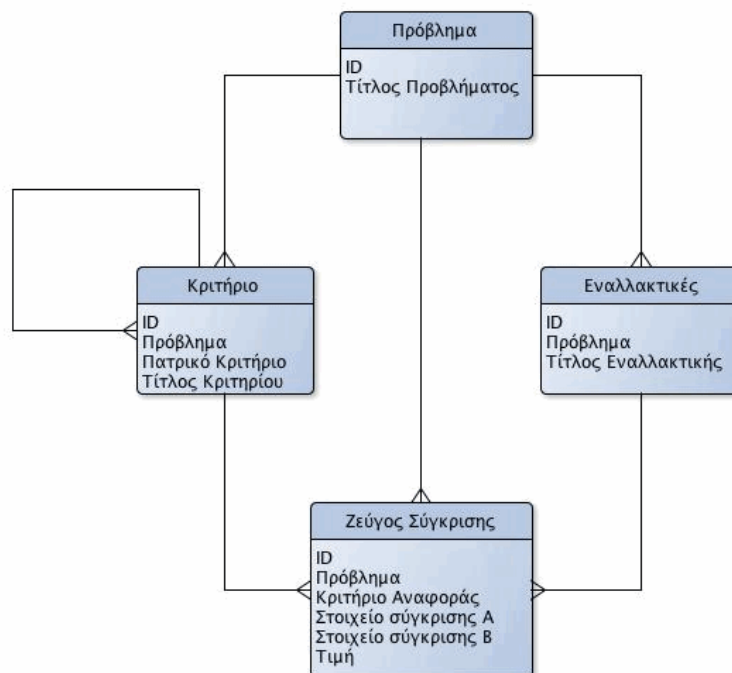
6.1. Χρήση έτοιμου κώδικα

Για την ανάπτυξη αυτής της εφαρμογής, χρησιμοποιήθηκαν τα παρακάτω στοιχεία έτοιμου κώδικα, κάποια από αυτά, με μικρές τροποποιήσεις:

- Matrix: κλάση η οποία υλοποιεί μήτρες στη γλώσσα προγραμματισμού php, του Pier-André Bouchard St-Amant (άδεια χρήσης BSD, <http://www.phpclasses.org/package/5793-PHP-Perform-operations-to-manipulate-matrices.html>)
- Php-csv-utils: μια συλλογή από κλάσεις για ανάγνωση και εγγραφή αρχείων σε μορφή csv (comma separated values), σε γλώσσα php, του Luke Visinoni (<http://code.google.com/p/php-csv-utils/>)
- Excel Reader V2, του Matt Kruse, κλάση για την ανάγνωση αρχείων Microsoft Excel, σε γλώσσα php (άδεια χρήσης MIT, <http://code.google.com/p/php-excel-reader/>)
- Libchart, του Jean-Marc Trémeaux, βιβλιοθήκη για την εμφάνιση γραφημάτων, σε γλώσσα php (άδεια χρήσης GNU GPL, <http://naku.dohcrew.com/libchart/pages/introduction/>)
- Matrix: υλοποίηση μήτρας σε Javascript, από το Rosetta Code (http://rosettacode.org/wiki/Matrix_multiplication)
- Mootools-tree: κώδικας ο οποίος χειρίζεται τη διεπαφή ενός δέντρου σε html. Σε γλώσσα javascript, με προϋπόθεση το mootools framework. Του Cristopher Rojer (άδεια χρήσης MIT, <https://github.com/crojer/mootools-tree>)
- Επίσης, ως βάση-σκελετός για την ανάπτυξη, χρησιμοποιήθηκε ο αυτόματος κώδικας που παρήχθη από την υπηρεσία J-Cook (<http://www.j-cook.pro/>). Η υπηρεσία αυτή παράγει ένα component το οποίο παρέχει τις βασικές λειτουργίες διαχείρισης πάνω σε ένα σχήμα πινάκων το οποίο δίνεται από το χρήστη.
- Γίνεται τέλος χρήση του Joomla Framework, το οποίο παρέχει έτοιμο κώδικα για τη δημιουργία στοιχείων HTML, για την ασφάλεια, την πρόσβαση στη βάση δεδομένων κλπ.

6.2. Πίνακες βάσης δεδομένων

Έχουμε 4 οντότητες: προβλήματα, κριτήρια, εναλλακτικές και ζεύγη συγκρίσεων. Συνεπώς, στη βάση δεδομένων, δημιουργούνται 4 πίνακες. Στο παρακάτω διάγραμμα, φαίνονται οι πίνακες, τα κυριότερα πεδία τους και οι μεταξύ τους σχέσεις:



Πίνακες βάσης δεδομένων και οι σχέσεις τους

Αναλυτικότερα:

Πίνακας προβλημάτων: είναι ο πίνακας # __ahrjoomla_ahpproblems¹. Ο πίνακας περιέχει τα πεδία:

- Id: ακέραιος, μοναδικό αναγνωριστικό (κλειδί) της βάσης (η τιμή του διαχειρίζεται από τη βάση δεδομένων)
- Params: τύπου text, υποχρεωτικό από τις προδιαγραφές του Joomla (κρατά ένα αλφαριθμητικό με παραμετρικές τιμές)
- Problem_title: αλφαριθμητικό 50 χαρακτήρων, ο τίτλος του προβλήματος
- Ordering: ακέραιος, τυποποιημένο πεδίο του joomla στο οποίο καταχωρείται η σειρά εμφάνισης του στοιχείου.
- Creation_date: τύπου datetime, ημερομηνία δημιουργίας της εγγραφής
- modification_date: τύπου datetime, ημερομηνία τελευταίας τροποποίησης της εγγραφής
- checked_out: ακέραιος, τυποποιημένο πεδίο Joomla. Εάν κάποιος χρήστης έχει δεσμεύσει την εγγραφή, το πεδίο παίρνει την τιμή του αναγνωριστικού του χρήστη, αλλιώς, 0
- checked_out_time: τύπου datetime, τυποποιημένο πεδίο Joomla. Εάν κάποιος χρήστης έχει δεσμεύσει την εγγραφή, λαμβάνει τιμή τη στιγμή εισόδου του χρήστη, αλλιώς, την τιμή 0.

Πίνακας εναλλακτικών: # __ahrjoomla_ahpalternatives. Πεδία:

- Id: ακέραιος, μοναδικό αναγνωριστικό (κλειδί) της βάσης (η τιμή του διαχειρίζεται από τη βάση δεδομένων)
- Params: τύπου text, υποχρεωτικό από τις προδιαγραφές του Joomla (κρατά ένα αλφαριθμητικό με παραμετρικές τιμές)
- ahp_problem: ακέραιος, περιέχει το μοναδικό αναγνωριστικό του προβλήματος στο οποίο αναφέρεται η εγγραφή. (foreign key στον πίνακα # __ahrjoomla_ahpproblems)
- alternative_name: αλφαριθμητικό 255 χαρακτήρων, ο τίτλος της εναλλακτικής
- Ordering: ακέραιος, τυποποιημένο πεδίο του joomla στο οποίο καταχωρείται η σειρά εμφάνισης του στοιχείου.
- Creation_date: τύπου datetime, ημερομηνία δημιουργίας της εγγραφής
- modification_date: τύπου datetime, ημερομηνία τελευταίας τροποποίησης της εγγραφής

Πίνακας κριτηρίων: # __ahrjoomla_ahrcriteria

Id: ακέραιος, μοναδικό αναγνωριστικό (κλειδί) της βάσης (η τιμή του διαχειρίζεται από τη βάση δεδομένων)

Params: τύπου text, υποχρεωτικό από τις προδιαγραφές του Joomla (κρατά ένα αλφαριθμητικό με παραμετρικές τιμές)

ahp_problem: ακέραιος, περιέχει το μοναδικό αναγνωριστικό του προβλήματος στο οποίο αναφέρεται η εγγραφή. (foreign key στον πίνακα # __ahrjoomla_ahpproblems)

parent_criterion: ακέραιος, περιέχει το μοναδικό αναγνωριστικό του πατρικού κριτηρίου, αν υπάρχει, αλλιώς παίρνει την τιμή 0 ή NULL. (το πεδίο αναφέρεται σε εγγραφή του ίδιου πίνακα)

criteria_name: αλφαριθμητικό 255 χαρακτήρων, ο τίτλος του κριτηρίου

Ordering: ακέραιος, τυποποιημένο πεδίο του joomla στο οποίο καταχωρείται η σειρά εμφάνισης του στοιχείου.

Creation_date: τύπου datetime, ημερομηνία δημιουργίας της εγγραφής

modification_date: τύπου datetime, ημερομηνία τελευταίας τροποποίησης της εγγραφής

Πίνακας Ζευγών σύγκρισης

- Id: ακέραιος, μοναδικό αναγνωριστικό (κλειδί) της βάσης (η τιμή του διαχειρίζεται από τη βάση δεδομένων)

¹ Κατά σύμβαση, οι πίνακες στο joomla συμβολίζονται με το πρόθεμα #__ το οποίο αντικαθίσταται από το σύστημα με ένα μεταβλητό πρόθεμα, διαφορετικό σε κάθε εγκατάσταση.

- Params: τύπου text, υποχρεωτικό από τις προδιαγραφές του Joomla (κρατά ένα αλφαριθμητικό με παραμετρικές τιμές)
- problem: ακέραιος, περιέχει το μοναδικό αναγνωριστικό του προβλήματος στο οποίο αναφέρεται η εγγραφή. (foreign key στον πίνακα #__ahrjoomla_ahpproblems)
- criterion: ακέραιος, περιέχει το μοναδικό αναγνωριστικό του κριτηρίου στο οποίο αναφέρεται το ζεύγος. (foreign key στον πίνακα #__ahrjoomla_ahpcriteria)
- alternative_a: ακέραιος, περιέχει το μοναδικό αναγνωριστικό του πρώτου στοιχείου το οποίο συγκρίνεται, είτε αυτό το στοιχείο είναι εναλλακτική, είτε είναι κριτήριο (διαχειρίζεται από την εφαρμογή)
- alternative_b: ακέραιος, ακριβώς όπως το αμέσως προηγούμενο πεδίο, για το δεύτερο στοιχείο που συγκρίνεται.
- Value: αλφαριθμητικό 255 χαρακτήρων, η τιμή της σύγκρισης. Μπορεί να περιέχει, είτε ένα αλφαριθμητικό το οποίο είναι ο βαθμός μιας σκάλας σύγκρισης (AP, I, SP κλπ.) είτε μια τιμή, όταν πρόκειται για απευθείας σύγκριση.

Από τους παραπάνω πίνακες, χρειάζεται να δώσουμε ιδιαίτερη σημασία στον τελευταίο (#__ahrjoomla_comparisonpairs). Ο πίνακας αυτός χρειάζεται να καλύψει ιδιαίτερες απαιτήσεις.

Πρώτον, τα στοιχεία για τα οποία καταγράφει τη σύγκριση, μπορεί να είναι, είτε κριτήρια, είτε εναλλακτικές. Το τελευταίο όμως, εξαρτάται από το κριτήριο στο οποίο αναφέρεται η σύγκριση: εάν πρόκειται για κριτήριο τελευταίου επιπέδου, η σύγκριση αφορά εναλλακτικές. Εάν όχι, η σύγκριση αφορά υποκριτήρια. Επομένως, η εφαρμογή, αρκεί να γνωρίζει το κριτήριο αναφοράς και, με βάση αυτό, να αναζητήσει τα στοιχεία αναφοράς στους αντίστοιχους πίνακες. Επομένως, αρκεί ένα πεδίο για να καταγράψει κάθε στοιχείο σύγκρισης, είτε αυτό είναι εναλλακτική, είτε είναι κριτήριο.

Δεύτερον, ο πίνακας καλείται να καταγράψει είτε τιμές οι οποίες ανήκουν σε μία βαθμονομημένη σκάλα, και μόνο αυτές τις τιμές, χωρίς αποκλίσεις, είτε τιμές που έχουν εισαχθεί απευθείας. Στην περίπτωση των τιμών που ανήκουν σε σκάλα, είναι προτιμότερη η σήμανσή τους με το λεκτικό της βαθμίδας (AP, DP, κλπ.) παρά η απευθείας αναγραφή της τιμής για δύο λόγους: για να αποφευχθούν ανακρίβειες οι οποίες συμβαίνουν πολλές φορές κατά την αποθήκευση αριθμών κινητής υποδιαστολής, αλλά και για την εύκολη μετατροπή ενός προβλήματος, από τη μια χρησιμοποιούμενη κλίμακα στην άλλη. Οπότε, χρειάζεται αλφαριθμητικό πεδίο, στο οποίο, όταν χρειαστεί, μπορεί να αποθηκευθεί και αριθμός.

Βέβαια, η εφαρμογή πρέπει να γνωρίζει τότε η αποθηκευμένη τιμή αφορά πίνακα από ζεύγη συγκρίσεων ή απευθείας εισαγωγή βαρών από το χρήστη. Γι' αυτό το λόγο, χρησιμοποιείται η εξής σύμβαση: όταν ο χρήστης εισάγει δεδομένα σε πίνακα συγκρίσεων, τα πεδία alternative_a και alternative_b περιέχουν τα μοναδικά αναγνωριστικά των συγκρινόμενων στοιχείων. Όταν ο χρήστης εισάγει απευθείας τιμές, τα πεδία alternative_a και alternative_b περιέχουν την ίδια τιμή, η οποία αναφέρεται στο στοιχείο για το οποίο εισάγεται το βάρος.

Τρίτο σημείο προσοχής: δε χρειάζεται να εισαχθούν όλα τα ζεύγη σύγκρισης. Δεδομένου πως, σε έναν πίνακα σύγκρισης, ισχύει πως

$$w[x][y]=1/w[y][x] \text{ και} \\ w[x][x]=1$$

αποθηκεύονται μόνο οι τιμές για τις οποίες ισχύει $y > x$. Επίσης, θεωρούμε πως προεπιλεγμένη τιμή σύγκρισης σε ένα ζεύγος, είναι η τιμή I της κλίμακας και πως, κατά συνέπεια, αν δε βρεθεί τιμή στον πίνακα για ένα ζεύγος σύγκρισης, τότε η τιμή αυτή θα είναι I. Με αυτόν τον τρόπο ελαχιστοποιούνται στο μέγιστο δυνατό οι εγγραφές που χρειάζονται για να περιγράψουν έναν πίνακα συγκρίσεων.

6.3. Ανάπτυξη πυρήνα (επίλυση προβλήματος)

Στην ενότητα αυτή, περιγράφεται η υλοποίηση του πυρήνα, του κομματιού δηλ. του κώδικα, ο οποίος ασχολείται με την προγραμματιστική μοντελοποίηση και επίλυση ενός προβλήματος AHP και μόνο.

6.3.1. Υπολογισμοί πινάκων

Το πιο σημαντικό τμήμα υπολογισμών για την Ιεραρχική Ανάλυση Αποφάσεων, είναι η εξαγωγή ιδιοτιμής και ιδιοδιάνυσματος. Ο πιο απλός αλγόριθμος για τον υπολογισμό, είναι η επαναληπτική μέθοδος κατά την οποία:

- υπολογίζεται ο μέσος όρος κάθε γραμμής,
- το ιδιοδιάνυσμα που προκύπτει, κανονικοποιείται, δηλ. κάθε στοιχείο του διαιρείται προς το άθροισμα των στοιχείων
- ο αρχικός πίνακας, υψώνεται στο τετράγωνο και υπολογίζεται εκ νέου το κανονικοποιημένο ιδιοδιάνυσμα
- εάν η διαφορά του νέου κανονικοποιημένου ιδιοδιάνυσματος από το προηγούμενο, είναι μικρότερη από το αποδεκτό όριο που έχουμε θέσει, σταματάμε τη διαδικασία. Αλλιώς, την επαναλαμβάνουμε υψώνοντας στο τετράγωνο τον ήδη υψωμένο πίνακα.

Όμως, η παραπάνω μέθοδος βρίσκει μόνο το κανονικοποιημένο ιδιοδιάνυσμα, ενώ στους υπολογισμούς, χρειαζόμαστε και την ιδιοτιμή, για τον υπολογισμό του Λόγου Συνέπειας (CR). Γι' αυτό, χρησιμοποιήθηκε η παραλλαγή της μεθόδου αυτής, η οποία βρίσκει και την ιδιοτιμή.

- Ως αρχικό ιδιοδιάνυσμα, χρησιμοποιείται το ιδιοδιάνυσμα που προκύπτει από το μέσο όρο των γραμμών.
- Το ιδιοδιάνυσμα πολλαπλασιάζεται με τον πίνακα.
- Το ιδιοδιάνυσμα κανονικοποιείται ως προς την τετραγωνική ρίζα του αθροίσματος των στοιχείων του.
- Εάν το νέο ιδιοδιάνυσμα έχει αποδεκτή διαφορά από το προηγούμενο, η διαδικασία διακόπτεται. Αλλιώς, επαναλαμβάνεται, πολλαπλασιάζοντας το νέο κανονικοποιημένο ιδιοδιάνυσμα.
- Στο τέλος της διαδικασίας, η τετραγωνική ρίζα του αθροίσματος των τετραγώνων του ιδιοδιάνυσματος, πριν την κανονικοποίηση, είναι η ιδιοτιμή, και το κανονικοποιημένο ιδιοδιάνυσμα, το ιδιοδιάνυσμα.

6.3.2. Εγκυρότητα των πινάκων

Πριν φτάσουμε στο πεδίο των υπολογισμών, πρέπει να περιορίσουμε την πιθανότητα σφαλμάτων από εσφαλμένα δεδομένα. Το ιδιοδιάνυσμα κι η ιδιοτιμή υπολογίζονται μόνο για τετράγωνους πίνακες. Εκτός αυτού, ο πίνακας συγκρίσεων της Ιεραρχικής Ανάλυσης Αποφάσεων έχει κάποια χαρακτηριστικά τα οποία πρέπει να είναι διασφαλισμένα. Αυτά τα χαρακτηριστικά είναι:

- Τα στοιχεία της διαγωνίου είναι πάντα 1
- Για κάθε στοιχείο του πίνακα ισχύει πως $a_{ij} = a_{ji}^{-1}$

Επιπλέον, ο πίνακας συγκρίσεων της Ι.Α.Α. ορίζει μερικές πράξεις (υπολογισμένα μεγέθη) οι οποίες είναι χαρακτηριστικές. Τα μεγέθη αυτά είναι ο Λόγος Συνέπειας κι ο Δείκτης Συνέπειας.

Για όλους τους παραπάνω λόγους, ο πίνακας συγκρίσεων έπρεπε να υλοποιηθεί ως ένα ξεχωριστό αντικείμενο.

Θεωρητικά, το νέο αυτό αντικείμενο θα έπρεπε να κληρονομεί από τον πίνακα. Μια τέτοια λύση όμως μπορούσε να δημιουργήσει προβλήματα, μια και, η πρόσβαση του χρήστη στις μεθόδους και στα πεδία του πίνακα, είναι δυνατόν να ακυρώσει την εγκυρότητα των δεδομένων του. Προτιμήθηκε λοιπόν, το αντικείμενο του πίνακα συγκρίσεων (AHPMatrix) να υλοποιηθεί ως wrapper του αντικειμένου Matrix, ούτως ώστε η προγραμματιστική πρόσβαση στον πίνακα να είναι πλήρως ελεγχόμενη.

6.3.3. Στοιχεία προβλήματος (Εναλλακτικές και κριτήρια)

Παρ' ότι οι Εναλλακτικές και τα Κριτήρια είναι απολύτως διακριτά στη διαδικασία της Ι.Α.Α., από προγραμματιστικής πλευράς έχουν το κοινό πως και τα δύο μπορούν να είναι αντικείμενο συγκρίσεων.

Άρα, αφού μιλάμε για δύο αντικείμενα που μοιράζονται κάποια ιδιότητα, είναι επόμενο πως μιλάμε δύο αντικείμενα που κληρονομούν από έναν κοινό πρόγονο.

Στην πράξη, το μόνο στο οποίο διαφέρουν τα δύο αντικείμενα μεταξύ τους, είναι πως τα κριτήρια έχουν και ιεραρχία. Άρα, μια εναλλακτική, είναι ένα στοιχείο προβλήματος. Οι ιδιότητες που φέρει είναι, ένα μοναδικό αριθμητικό αναγνωριστικό (το οποίο δε συμπίπτει απαραίτητα με αυτό της βάσης δεδομένων) και ένα όνομα. Ένα κριτήριο, είναι ένα στοιχείο προβλήματος το οποίο, επιπλέον, μπορεί να έχει γονέα ή/και τέκνα.

Τελικά, ένα κριτήριο, είναι απόγονος μιας εναλλακτικής (η οποία, με αυτόν τον τρόπο, ταυτίζεται με το στοιχείο προβλήματος). Συγκρίσεις μπορούν να γίνουν μόνο μεταξύ εναλλακτικής ή απογόνων της κλάσης εναλλακτικής.

6.3.4. Ιεραρχία κριτηρίων – κόμβοι

Η μοντελοποίηση της ιεραρχίας των κριτηρίων παρουσιάζει τα δικά της προβλήματα.

- **Χρειάζεται να μπορεί να επιλεγεί ένας συγκεκριμένος κόμβος:** προκειμένου να μπορεί να επιλεγεί ο οποιοσδήποτε κόμβος, δημιουργήθηκε μια στατική μεταβλητή τύπου πίνακα, η οποία περιέχει αναφορές σε όλους τους κόμβους που έχουν δημιουργηθεί. Κάθε φορά που δημιουργείται ένα νέο κριτήριο, στον πίνακα προστίθεται ως στοιχείο η αναφορά στο κριτήριο με δείκτη το μοναδικό αναγνωριστικό του κριτηρίου (το οποίο είναι απαραίτητα μοναδικό, όχι όμως απαραίτητα αυτό της βάσης δεδομένων). Ο μηχανισμός συμπληρώνεται από μια στατική μέθοδο η οποία παίρνει ως όρισμα το μοναδικό αναγνωριστικό κι επιστρέφει το στοιχείο του πίνακα με δείκτη το αναγνωριστικό αυτό, ή null αν δεν υπάρχει.
- **Πρέπει να υπάρχει πάντα συνέπεια στους δείκτες γονέων και τέκνων.** Δηλαδή, αν το A λέει πως έχει τέκνο του το B, τότε και το B πρέπει να λέει πως έχει γονέα του το A και να αποτραπεί το ενδεχόμενο να σπάσει κάπου αυτή η συνέπεια. Η αρχική προσέγγιση ήταν, όταν τίθεται ο γονέας ενός κόμβου, να τίθεται ταυτόχρονα και το τέκνο του πατρικού κόμβου και το αντίστροφο. Με άλλα λόγια, ενώ, προγραμματιστικά, θα ορίζεται η σχέση ενός στοιχείου, εσωτερικά, θα ορίζονται οι σχέσεις σε όλα τα στοιχεία που εμπλέκονται. Η προσέγγιση αυτή ήταν εύκολο να οδηγήσει σε ατέρμονους βρόχους (τέκνο καλεί πατέρα που καλεί τέκνο κ.ο.κ.). Για να αποφευχθεί αυτό το ενδεχόμενο, ορίστηκε πως οι γονικές σχέσεις κάθε κριτηρίου θα ορίζονται άπαξ, στον ορισμό του. Έτσι, όταν δημιουργείται ένα κριτήριο, θα ορίζεται και ο γονέας του και το κριτήριο θα προστίθεται στα τέκνα του γονέα του. Με αυτόν τον τρόπο, η σειρά ορισμού είναι πάντα ορισμένη και δεν προκύπτουν ατέρμονοι βρόχοι. Το παρεπόμενο αυτής της προσέγγισης είναι πως, με αυτόν τον τρόπο, δεν μπορεί να αλλάξει δυναμικά η δομή του δέντρου. Είναι όμως ένα παρεπόμενο που δε δημιουργεί προβλήματα, αφού το δέντρο δεν χρειάζεται να αλλάζει μορφή κατά τη διάρκεια μιας αίτησης στον εξυπηρετητή. Η προσέγγιση αυτή όμως δημιούργησε το επόμενο πρόβλημα
- **Οι κόμβοι του δέντρου των κριτηρίων πρέπει να μπορούν να οριστούν με οποιαδήποτε σειρά:** έχοντας υπόψη πως η ιεραρχία αποθηκεύεται σε μια βάση δεδομένων και πως, η σειρά με την οποία θα έρθουν οι εγγραφές από τη βάση δεδομένων, δεν μπορεί να μας εγγυηθεί πως θα ακολουθεί την ιεραρχία του δέντρου, βρισκόμαστε μπροστά στο ενδεχόμενο να δημιουργηθεί πρώτα το τέκνο και μετά ο γονέας. Πρώτα θα δημιουργηθεί το τέκνο το οποίο θα δηλώνει ως γονέα του έναν κόμβο ο οποίος δεν υπάρχει, άρα δε θα δηλωθεί πουθενά η αντίστροφη σχέση. Μετά, θα δημιουργηθεί ο γονέας, αλλά δε θα φαίνονται από πουθενά τα τέκνα του, αφού ο γονέας δεν υπήρχε όταν δημιουργήθηκε το τέκνο. Πρέπει να υπάρχει ένας τρόπος να διατηρηθεί αυτή η πληροφορία.
Ο τρόπος με τον οποίο επιλύθηκε το πρόβλημα αυτό, ήταν η δημιουργία ψευδο-κριτηρίων. Όταν δηλ. δημιουργείται ένα κριτήριο το οποίο δηλώνει ως πατρικό κάποιο κριτήριο το οποίο δεν υπάρχει ακόμα, δημιουργείται ένας ψευδο-κόμβος ο οποίος έχει το ίδιο μοναδικό αναγνωριστικό με τον πατρικό κόμβο, δεν έχει όνομα και κρατά την αναφορά προς το τέκνο που δημιουργείται. Αν δημιουργηθούν και άλλα τέκνα του ίδιου (ανύπαρκτου ακόμα) κόμβου, αυτά προστίθενται στα τέκνα του ίδιου ψευδο-κόμβου. Όταν, τελικά, έρθει η σειρά του γονέα να δημιουργηθεί, αυτός δημιουργείται κανονικά και η πληροφορία των τέκνων αντιγράφεται από τον ψευδο-κόμβο κι ο ψευδοκόμβος καταστρέφεται.
Ως επιπλέον δικλείδα ασφαλείας, ελέγχεται πάντα πως το δέντρο των κριτηρίων, πριν ενταχθεί

σε κάποιο πρόβλημα, δεν περιέχει κανέναν ψευδοκόμβο, έχει δηλ. ολοκληρωθεί ο ορισμός των κριτηρίων.

Ειδική περίπτωση κριτηρίου, είναι το κορυφαίο κριτήριο το οποίο αντιπροσωπεύει τη ρίζα της ιεραρχίας, δηλ. το γενικό στόχο. Ο κόμβος αυτός αντιπροσωπεύει ολόκληρο το πρόβλημα, είναι το μοναδικό σημείο αφητηρίας σε ό,τι αφορά τα κριτήρια, πλην όμως, δεν αποθηκεύεται ποτέ στη βάση δεδομένων.

6.3.5. Ιεραρχία Κριτηρίων – Δέντρο

Η παρουσίαση των κόμβων του προβλήματος έπρεπε να ακολουθεί τον κανόνα από το γενικότερο, στο ειδικότερο. Αυτό συνεπάγεται πως πρέπει να διατρέξουμε το δέντρο κατά μήκος (length-first). Ο αλγόριθμος αυτός υλοποιήθηκε, κατά τα standards της γλώσσας php ως iterator, ως μια δομή δηλ. η οποία επιτρέπει στο χρήστη να καλέσει το αντικείμενο μέσω ενός βρόχου.

6.3.6. Μοναδικά αναγνωριστικά

Λίγα λόγια για τα μοναδικά αναγνωριστικά: η υιοθέτησή τους κατά την ανάπτυξη ήταν απαραίτητη, όχι μόνο για να ξεχωρίζουν μεταξύ τους τα στοιχεία από προγραμματιστικής πλευράς, αλλά και για την απεικόνισή τους στο χρήστη. Πράγματι, στην παρουσίαση των τιμών στο χρήστη, χρειάζεται να δημιουργηθούν πολλά πανομοιότυπα στοιχεία (σύνδεσμοι ή input boxes) τα οποία πρέπει να έχουν ένα ξεχωριστό αναγνωριστικό. Τα μοναδικά χαρακτηριστικά, ως συνθετικά των ονομάτων των στοιχείων HTML, βοηθούν στο να είναι μοναδικά και τα στοιχεία HTML.

Ως μοναδικά αναγνωριστικά παίρνονται, κατά κανόνα, τα id όπως αυτά λαμβάνονται από τη βάση δεδομένων. Εξαιρέση όμως αποτελούν οι περιπτώσεις import και copy, όπου το μοναδικό αναγνωριστικό παράγεται εσωτερικά από το σύστημα, αφού οι λειτουργίες αυτές δεν ξεκινάνε από τη βάση δεδομένων.

Στην ειδική περίπτωση του κορυφαίου κριτηρίου, έχει ληφθεί υπόψη μια ιδιότητα της MySQL. Συγκεκριμένα, το πεδίο id έχει την ιδιότητα auto increment, και η τιμή του ορίζεται αυτόματα από τη MySQL. Η MySQL όμως αριθμεί τις εγγραφές, εξ ορισμού, από το 1. Γι' αυτό, στο κορυφαίο κριτήριο δίνεται η τιμή 0, η οποία δεν πρόκειται ποτέ να συμπίπτει με αριθμό εγγραφής που υπάρχει ήδη στη βάση.

Το ότι το κορυφαίο κριτήριο έχει σταθερό αναγνωριστικό, προσθέτει τον περιορισμό πως, από την υλοποίηση, μπορεί να εξεταστεί μόνο ένα πρόβλημα κάθε φορά.

6.3.7. Αποθήκευση τιμών σύγκρισης

Οι τιμές σύγκρισης είναι το κομμάτι που πρέπει να έχει τη μεγαλύτερη ευελιξία, αλλά πρέπει ταυτόχρονα να έχει και μεγάλη αξιοπιστία. Τα στοιχεία σύγκρισης, ανάλογα με το κριτήριο στο οποίο αναφέρονται, μπορούν να είναι είτε κριτήρια είτε εναλλακτικές. Ωστόσο, κάτω από το ίδιο κριτήριο θα συγκρίνονται είτε αποκλειστικά κριτήρια, είτε αποκλειστικά εναλλακτικές. Η παραπάνω απαίτηση, τελικά, μπορεί να υλοποιηθεί λιγότερο περίπλοκα από όσο ακούγεται: δύο sql statements που έχουν οριστεί στο μοντέλο και καλούνται οποτεδήποτε χρειάζεται να επιβεβαιωθεί η εγκυρότητα των δεδομένων σύγκρισης, διαγράφουν οποιοδήποτε στοιχείο δεν υπακούει σε αυτούς τους κανόνες.

Οι τιμές που αποθηκεύονται, εξ ορισμού, πρέπει να ανήκουν σε μια κλίμακα. Ο κανόνας αυτός γίνεται υποχρεωτικός αποθηκεύοντας, όχι την τιμή απευθείας, αλλά μια συμβολική σταθερά η οποία παραπέμπει στην τιμή. Ξεκινώντας από τις σταθερές I, WP, SP, DP και AP, βάζουμε το πρόθεμα I για να δημιουργήσουμε τις αντίστροφές τους (IWP κλπ) και με την κατάληξη M (medium) σηματοδοτούμε τις ενδιάμεσες τιμές. Η τακτική αυτή, μας δίνει το πλεονέκτημα να μπορούμε να αλλάξουμε την κλίμακα ενός προβλήματος αφού έχουμε ορίσει ήδη τα ζεύγη συγκρίσεων.

Ωστόσο, οι τιμές πρέπει να επιτρέπουν και την απευθείας εισαγωγή βαρών. Εκεί έπρεπε να υπάρχει μια σύμβαση η οποία να ξεχωρίζει εγγραφές που αφορούν απευθείας ανάθεση βάρους από τιμή που αφορά ζεύγος σύγκρισης. Η λύση του να μπει flag που να ξεχωρίζει τέτοιες εγγραφές μεταξύ τους απορρίφθηκε καθώς θα μπορούσε να δημιουργήσει το ενδεχόμενο να συνυπάρχουν, κάτω από το ίδιο κριτήριο, εγγραφές που αφορούν ζεύγη σύγκρισης και τιμές απευθείας ανάθεσης, ταυτόχρονα. Αντ' αυτής της

λύσης, προτιμήθηκε, οι εγγραφές απευθείας ανάθεσης, να ξεχωρίζουν από το ότι, το στοιχείο σύγκρισης A είναι το ίδιο με το στοιχείο σύγκρισης B.

Η εγκυρότητα των δεδομένων του πίνακα αυτού ενισχύεται από τη γενικότερη τακτική στην αποθήκευση των τιμών σύγκρισης: οι τιμές στον πίνακα δεν ενημερώνονται. Διαγράφονται όλες οι τιμές που αφορούν κάποιο κριτήριο και επανεισάγονται.

6.4. Τεκμηρίωση βασικών κλάσεων

Σε αυτήν την ενότητα, γίνεται τεκμηρίωση των κλάσεων του πυρήνα (όπως αυτό περιγράφηκε στην ανάλυση) τα οποία υλοποιούν το πρόβλημα της Ιεραρχικής Ανάλυσης Αποφάσεων. Η πρόσβαση στη βάση δεδομένων, η παρουσίαση κι η δρομολόγηση των ενεργειών, γίνεται με τον τυποποιημένο τρόπο του Joomla, από τη δομή Model-View-Controller.

Περιγράφονται μόνο οι βασικές μέθοδοι και πεδία των κλάσεων, καθώς κάθε κλάση περιέχει εσωτερική τεκμηρίωση, η οποία συμφωνεί με το καθιερωμένο standard του phpdocumentor.

6.4.1. AhpConstants

Η κλάση αυτή περιλαμβάνει σταθερές και τιμές οι οποίες αντιμετωπίζονται ως σταθερές. Πρόκειται για τιμές οι οποίες καθορίζουν τη συμπεριφορά του προγράμματος σε κρίσιμα σημεία και οι οποίες είναι καλύτερα να είναι δύσκολο να αλλάξουν (να μην μπορούν δηλ. να αλλάξουν εύκολα μέσω του διαχειριστικού interface).

Οι τιμές οι οποίες ορίζονται και η σημασία τους, είναι οι παρακάτω:

MAX_ALTERNATIVES: Ορίζει το μέγιστο αριθμό εναλλακτικών που μπορεί να έχει ένα πρόβλημα. Επειδή η τιμή αυτή σχετίζεται άμεσα με το μέγεθος των πινάκων που προκύπτουν και, κατ' επέκταση, με τον υπολογισμό του Δείκτη Συνέπειας, η τιμή αυτή πρέπει να είναι ίση με το μέγιστο δείκτη για τον οποίο έχει καταχωρηθεί τιμή ACI (βλπ. παρακάτω)

MAX_CRITERIA_PER_NODE: αντίστοιχα, ο μέγιστος αριθμός των υποκριτηρίων σε έναν κόμβο. Σημειώνεται πως κορυφαίος κόμβος είναι πάντα το πρόβλημα. Ισχύουν τα ίδια με την αμέσως προηγούμενη τιμή, με την οποία καλό είναι να είναι ίσες.

MIN_ALTERNATIVES: Ο ελάχιστος αριθμός εναλλακτικών. Τιμή μικρότερη από 2 δεν έχει νόημα και δεν πρέπει να χρησιμοποιείται.

MIN_CRITERIA_PER_NODE: Ο ελάχιστος αριθμός κριτηρίων. Τιμή μικρότερη από 2 θα προκαλέσει σφάλμα στην εφαρμογή.

CR_LIMIT: Η μέγιστη αποδεκτή τιμή του λόγου συνέπειας (Consistency Ratio) ως ποσοστό. Έχει τεθεί η, γενικώς αποδεκτή τιμή, 10%.

PRECISION: ο αριθμός των δεκαδικών ψηφίων που θέλουμε να εμφανίζονται

TREE_ROOT_ID: Ο κορυφαίος κόμβος των κριτηρίων, είναι το ίδιο το πρόβλημα. Δεδομένου πως δεν αποθηκεύεται στη βάση των δεδομένων ως κριτήριο, θα πρέπει να έχει ένα σταθερό αναγνωριστικό. Χρησιμοποιείται λοιπόν ο αριθμός 0, αφού η MySQL ξεκινάει την αρίθμηση των εγγραφών από το 1 και, άρα, δεν υπάρχει περίπτωση να συμπέσει με κριτήριο που έχει αποθηκευτεί στη βάση δεδομένων. Αυτή η τιμή δεν πρέπει να αλλάξει, ή, εάν παραστεί ανάγκη, να πάρει μόνο αρνητικές τιμές.

SENS_ANALYSIS_STEPS: Χρησιμοποιείται κατά την ανάλυση ευαισθησίας. Η Ανάλυση Ευαισθησίας, παίρνει τα τοπικά βάρη ενός κριτηρίου (ή τις προτεραιότητες μιας εναλλακτικής) και αλλάζει την τιμή τους στο διάστημα [0..1]. Η τιμή αυτή, ελέγχει σε πόσα βήματα θα γίνει αυτή η εναλλαγή, δηλ. πόσο λεπτομερής θα είναι. Έχει τεθεί η τιμή 20 η οποία είναι αρκετά υψηλή, αφού αυτό σημαίνει πως σε μία αίτηση ανάλυσης ευαισθησίας, ολόκληρο το πρόβλημα θα υπολογιστεί 20 φορές. Επομένως, η τιμή αυτή πρέπει να χειρίζεται με προσοχή, αφού μπορεί να δημιουργήσει φόρτο στον εξυπηρετητή.

ACI_VALUE: Ο πίνακας αυτός περιέχει τις τιμές ACI για τον υπολογισμό του Consistency Index. Πρώτος δείκτης είναι το 3, αφού ACI για δείκτη 1 δεν έχει νόημα, και ζεύγη συγκρίσεων για 2 στοιχεία είναι εξ ορισμού συνεπή. Η τιμή αυτή είναι private. Η πρόσβαση στον πίνακα γίνεται από την μέθοδο `getAciArray()`

SCALE_PARAMETERS: Πίνακας ο οποίος περιέχει τις διαθέσιμες κλίμακες που υπάρχουν, με δείκτη του πίνακα ένα μοναδικό αναγνωριστικό για κάθε σκάλα. Σημειώνεται πως οι δείκτες του πίνακα πρέπει οπωσδήποτε να συμφωνούν με αυτές του αρχείου config.xml του component.

6.4.2. AhpScale

Αφηρημένη κλάση η οποία υλοποιεί μια κλίμακα AHP. Παρέχει μεθόδους για την αντιστοίχιση κλειδιών (SP, DP, I κλπ.) με πραγματικές τιμές (5,7,1 αντίστοιχα) καθώς και πληροφόρηση για την κλίμακα που χρησιμοποιείται.

Με τη χρήση αυτής της κλάσης, απλοποιείται η πρόσθεση κλιμάκων προς χρήση. Οποιαδήποτε κλίμακα AHP πρέπει να είναι απόγονος της AHP Scale και, το μόνο που χρειάζεται για να υλοποιηθεί, είναι να οριστούν οι μεταβλητές _scale9, name και id, οι οποίες περιγράφονται παρακάτω:

Η μεταβλητή _scale9 είναι ένας πίνακας που περιέχει τις τιμές της κλίμακας που είναι μεγαλύτερες ή ίσες του ένα. Πρέπει υποχρεωτικά να περιέχει και τις ενδιάμεσες τιμές της κλίμακας, συνεπώς πρέπει να έχει ακριβώς 9 στοιχεία, αριθμημένα από 0 (που αντιστοιχεί στο I) μέχρι και 8 (που αντιστοιχεί στο δείκτη AP).

Η μεταβλητή name είναι το όνομα της κλίμακας το οποίο θα εμφανίζεται στο χρήστη. Αυτό, καλό είναι να αντιστοιχεί επίσης με το όνομα που ορίζεται στο αρχείο config.xml

Τέλος, η μεταβλητή id είναι το μοναδικό αναγνωριστικό της κλίμακας, το οποίο πρέπει υποχρεωτικά να συμπίπτει με το αναγνωριστικό στον πίνακα SCALE_PARAMETERS της κλάσης AhpConstants και τις τιμές του αρχείου config.xml

6.4.3. AhpElement

Κοινός πρόγονος όλων των στοιχείων του προβλήματος, δηλ. των κλάσεων που αναπαριστούν κριτήρια και εναλλακτικές. Πρόκειται για μια απλούστατη κλάση η οποία κρατάει το αναγνωριστικό της βάσης δεδομένων και τον τίτλο του στοιχείου, και παρέχει μεθόδους για την προσπέλασή τους.

Είναι η κλάση που χρησιμοποιείται για να απεικονίσει προγραμματιστικά τις εναλλακτικές.

6.4.4. AhpCriteriaNode

Υλοποιεί έναν κόμβο στο δέντρο των κριτηρίων, αλλά εμπεριέχει και λειτουργικότητα δέντρου. Απόγονος της AhpElement, κρατά πληροφορίες για τα γονικά και τα θυγατρικά κριτήρια. Περιέχει στατική λίστα, η οποία με τη σειρά της περιέχει όλους τους κόμβους που έχουν δημιουργηθεί. Παρέχει μεθόδους για την εγκυρότητα των κόμβων, για την επιλογή συγκεκριμένου κόμβου, για την πρόσβαση στο γονικό ή στα θυγατρικά στοιχεία και, τέλος, μεθόδους που ελέγχουν αν ο κόμβος είναι ρίζα ή φύλλο του δέντρου.

Ένα πρόβλημα που έπρεπε να αντιμετωπιστεί στον constructor της κλάσης, είναι πως υπάρχει η πιθανότητα να ζητηθεί να δημιουργηθεί ένας θυγατρικός κόμβος χωρίς προηγουμένως να έχει δημιουργηθεί ο πατρικός. Σε αυτήν την περίπτωση, υπάρχει κίνδυνος να χαθεί η σύνδεση μεταξύ πατρικού και θυγατρικών στοιχείων. Γι' αυτό το λόγο, εάν διαπιστωθεί κάτι τέτοιο, δημιουργείται ένας ψευδο-κόμβος ο οποίος κρατά τις πληροφορίες σύνδεσης και, όταν ζητηθεί τελικά η δημιουργία του πατρικού κόμβου, οι πληροφορίες μεταφέρονται από τον ψευδο-κόμβο στον πραγματικό κόμβο.

Ισχύει η σύμβαση, πως η κλάση αυτή θα κληθεί μαζικά μια φορά κατά τη διάρκεια της εκτέλεσης και θα δημιουργήσει όλα τα στοιχεία του δέντρου. Αφού ολοκληρωθεί η δημιουργία του δέντρου, δε θα πρέπει να υπάρχουν ψευδο-κόμβοι και αυτό ελέγχεται από τη μέθοδο isTreeComplete.

Επίσης, έχει ακολουθηθεί η σύμβαση πως, μετά τη δημιουργία του, το δέντρο δεν τροποποιείται. Αυτό εξυπηρετεί κατά την εκτέλεση και δεν έχει καμία σχέση με τη δυνατότητα του χρήστη να προσθαφαιρέσει κόμβους από το δέντρο.

6.4.5. AhpWeights

Η κλάση αυτή κρατάει τα βάρη (ή τις προτεραιότητες) που συνδέονται με ένα σύνολο υποκριτηρίων (του ίδιου γονέα) ή εναλλακτικών. Συγκεκριμένα, κρατά τα βάρη και τη σύνδεσή τους με το μοναδικό αναγνωριστικό της βάσης δεδομένων. Η κλάση αυτή προέκυψε από την ανάγκη να μπορεί να γίνει προσπέλαση των βαρών και με τη σειρά (0,1,2...) και με το αναγνωριστικό της βάσης δεδομένων. Παρέχει μεθόδους που επιστρέφουν ένα βάρος μεμονωμένα ή όλα τα βάρη μαζί. Τα βάρη επιστρέφονται ως αντικείμενο AhpMatrix, το οποίο περιγράφεται παρακάτω.

6.4.6. AhpMatrix

Η κλάση αυτή υλοποιεί το υπολογιστικό μέρος ενός πίνακα συγκρίσεων. Κληρονομεί από την κλάση Matrix η οποία υλοποιεί μια μήτρα και προσαρμόζει μια μήτρα στις ανάγκες ενός προβλήματος Ιεραρχικής Ανάλυσης Αποφάσεων, δηλ. φροντίζει οι πίνακες να είναι πάντα τετράγωνοι, τα στοιχεία της διαγωνίου να έχουν πάντα την τιμή 1, και, όταν τίθεται μια τιμή σε κάποιο στοιχείο του πίνακα, να τίθεται ταυτόχρονα και η συμμετρική του τιμή. Ακόμα η μήτρα του προβλήματος δεν εκτίθεται (είναι private) ώστε η κλάση να ελέγχει πλήρως τα δεδομένα που περιέχει.

Επίσης, παρέχει μεθόδους για τον υπολογισμό, όχι μόνο των προτεραιοτήτων, αλλά και του δείκτη συνέπειας και του λόγου συνέπειας. Σημειώνεται πως τα αποτελέσματα των υπολογισμών αποθηκεύονται και πως οι υπολογισμοί δεν επαναλαμβάνονται, παρά μόνο αν έχει αλλάξει κάποια τιμή στον πίνακα, αποφεύγοντας έτσι την επιβάρυνση του εξυπηρετητή.

Διευκρινίζεται πως η κλάση αυτή ασχολείται μόνο με το υπολογιστικό μέρος, χωρίς να γνωρίζει τη σύνδεση με στοιχεία του προβλήματος. Η σύνδεση αυτή υλοποιείται σε ανώτερο επίπεδο.

6.4.7. AhpObjectMatrix

Περικλείει την AhpMatrix, και υλοποιεί τη σύνδεση μεταξύ ενός πίνακα συγκρίσεων και των στοιχείων τα οποία συγκρίνει. Η κλάση περικλείει τις κυριότερες μεθόδους της AhpMatrix, αλλά παρέχει επιπλέον μεθόδους που διασφαλίζουν πως τα συγκρινόμενα στοιχεία θα είναι ομοειδή (μόνο κριτήρια μεταξύ τους, μόνο εναλλακτικές μεταξύ τους).

Η κλάση, παρέχει επίσης μια παράκαμψη στο μηχανισμό υπολογισμού. Όταν ο χρήστης θέλει να θέσει απευθείας τα βάρη, ακυρώνει τον πίνακα συγκρίσεων που περιέχει, και αποθηκεύει τις τιμές απευθείας στον εσωτερικό πίνακα που περιέχει τα βάρη, θέτοντας – εσωτερικά – την απαραίτητη σήμανση πως τα βάρη έχουν τεθεί απευθείας. Ως αποτέλεσμα, όταν ζητηθεί κάποιο βάρος από το αντικείμενο, δεν έχει σημασία το πως έχει προκύψει, επιστρέφει πάντα το σωστό.

6.4.8. AhpProblemData

Η κλάση αυτή χρησιμεύει ως γέφυρα μεταξύ της βάσης δεδομένων και της προγραμματιστικής απεικόνισης ενός πλήρους προβλήματος Ιεραρχικής Ανάλυσης Αποφάσεων και έχει σκοπό να εξασφαλίσει την εγκυρότητα των δεδομένων που θα αποτελέσουν το πρόβλημα.

Κατά τη δημιουργία του αντικειμένου, ο constructor λαμβάνει ως ορίσματα τις γραμμές της βάσης δεδομένων που αφορούν το πρόβλημα, από όλους τους πίνακες και τις περνά από διαδοχικές δοκιμές εγκυρότητας. Συγκεκριμένα, ελέγχει:

- Πως τα ονόματα των στοιχείων είναι μοναδικά
- Πως όλα τα στοιχεία του δέντρου των κριτηρίων είναι κανονικά συνδεδεμένα μεταξύ τους, χωρίς αμφισημίες και χωρίς κυκλικές αναφορές
- Πως ο αριθμός των κριτηρίων σε κάθε κόμβο δεν υπερβαίνει τα επιτρεπτά όρια
- Πως τα ζεύγη συγκρίσεων αναφέρονται σε ομοειδή αντικείμενα, με ίδιο πατρικό στοιχείο.

Ο constructor δεν αποτυγχάνει, ούτε προκαλεί εξαίρεση κατά την εκτέλεση. Αντίθετα, σε περίπτωση σφάλματος, γεμίζει έναν εσωτερικό πίνακα με μηνύματα τα οποία μετά είναι προσπελάσιμα μέσω μεθόδου.

6.4.9. AhpCriteriaTree

Παρέχει έναν iterator ο οποίος προσπελαύνει το δέντρο των κριτηρίων κατά πλάτος (length-first). Επίσης, φροντίζει ώστε κάθε κόμβος, να είναι συνδεδεμένος με έναν πίνακα συγκρίσεων.

6.4.10. AhpComputations

Παρέχει στατικές μεθόδους οι οποίες επιστρέφουν τοπικά και καθολικά βάρη κριτηρίων και προτεραιότητες εναλλακτικών.

6.4.11. AhpProblem

Υλοποιεί προγραμματιστικά το πρόβλημα Ιεραρχικής Ανάλυσης Απόφασης. Λαμβάνει ως όρισμα, ένα αντικείμενο της κλάσης AhpProblemData το οποίο προηγουμένως πρέπει οπωσδήποτε να έχει ελεγχθεί ώστε να μην επιστρέφει λάθη επαλήθευσης.

Φροντίζει για τη δημιουργία ολόκληρου του δέντρου των κριτηρίων, των εναλλακτικών και αποδίδει τιμές στους πίνακες συγκρίσεων.

Παρέχει μεθόδους πρόσβασης στα στοιχεία του προβλήματος, αλλά και στους πίνακες συγκρίσεων.

6.4.12. SensAnalysisHelper

Παρέχει τη μέθοδο getVariations η οποία επιστρέφει μια σειρά αποτελεσμάτων για χρήση από την Ανάλυση Ευαισθησίας. Συγκεκριμένα, παρέχει όλες τις λύσεις του προβλήματος για διαδοχικές τιμές του βάρους του κριτηρίου που ζητήθηκε, ή της προτεραιότητας της εναλλακτικής.

6.4.13. DbAlternative, DbCriterion, DbComparisonPair

Οι τρεις αυτές κλάσεις, προσομοιώνουν μια εγγραφή από τους πίνακες των εναλλακτικών, των κριτηρίων και των ζευγών σύγκρισης αντίστοιχα. Χρησιμοποιούνται από τη διαδικασία εισαγωγής αρχείου CSV ή Excel.

6.4.14. AhpCsvParser

Λαμβάνει έναν πίνακα από blocks (βλπ. παρακάτω) και δημιουργεί αντικείμενα που μοιάζουν με εγγραφές στη βάση δεδομένων, ώστε να είναι έτοιμα για εισαγωγή. Κατά την κλήση του constructor, τα blocks δρομολογούνται στους κατάλληλους parsers.

6.4.15. AhpCsvOperations

Παρέχει μεθόδους σχετικές με την εισαγωγή αρχείων CSV ή Excel.

Η γενική διαδικασία έχει ως εξής: διαβάζεται το αρχείο και μετατρέπεται σε έναν πίνακα δύο διαστάσεων. Στη συνέχεια, όλες οι συνεχείς γραμμές του αρχείου, ενώνονται σε μια ενότητα που ονομάζεται, εσωτερικά, block. Τα blocks δρομολογούνται στην κλάση AhpCsvParser για να δημιουργηθούν οι ψευδοεγγραφές. Οι ψευδοεγγραφές δημιουργούν ένα νέο αντικείμενο AhpProblemData και το αντικείμενο αυτό δρομολογείται στη διαδικασία κλωνοποίησης, σαν να αντιγράφεται ένα πρόβλημα από τη βάση δεδομένων.

6.5. Περιγραφή κυρίων διαδικασιών

Σε αυτήν την ενότητα, τεκμηριώνεται συνοπτικά ο τρόπος υλοποίησης των κυρίων διαδικασιών του component.

6.5.1. Αποθήκευση προβλήματος

Αποθηκεύει νέα ή ήδη υπάρχοντα προβλήματα στη βάση δεδομένων, από τη λειτουργία Save & Apply της θόνης ορισμού της δομής ενός προβλήματος. Χρησιμοποιείται επίσης από τη διαδικασία εισαγωγής προβλήματος. Ο κώδικας της διαδικασίας βρίσκεται στο αρχείο `site\controllers\ahpproblems.php`, στη μέθοδο `AhpJoomlaControllerAhpproblems::_save_internal`.

Για την αποθήκευση των κριτηρίων, έχει να αντιμετωπιστεί το παρακάτω πρόβλημα: πρέπει να μεταφερθεί η πληροφορία για την ιεραρχία τους με αξιόπιστο τρόπο, ακόμα κι όταν πρόκειται για νέα κριτήρια τα οποία δεν έχουν λάβει μοναδικό αναγνωριστικό από τη βάση δεδομένων. Γι' αυτό το λόγο, το σύστημα αποδίδει σε κάθε κριτήριο ένα ψευδο-αναγνωριστικό (στον κώδικα αναφέρεται ως dummy ID) το οποίο είναι μοναδικό, στα πλαίσια του προβλήματος.

Η αποθήκευση των κριτηρίων, επομένως, γίνεται σε δύο φάσεις: πρώτα αποθηκεύονται τα κριτήρια χωρίς την πληροφορία της ιεραρχίας. Καθώς αποθηκεύονται, συσχετίζεται το αναγνωριστικό της βάσης δεδομένων, με το ψευδο-αναγνωριστικό. Όταν έχουν αποθηκευθεί όλα, τότε έχουμε το μοναδικό αναγνωριστικό βάσης για όλες τις εγγραφές, το οποίο μπορούμε να το βρούμε μέσω του ψευδο-αναγνωριστικού. Τότε, οι εγγραφές ενημερώνονται (αποθηκεύονται για δεύτερη φορά), αυτή τη φορά περιέχοντας την πληροφορία της ιεραρχίας.

1. Ελέγχονται τα δικαιώματα του χρήστη. Εάν ο χρήστης δεν μπορεί να δημιουργήσει ή να τροποποιήσει το πρόβλημα, γίνεται έξοδος.
2. Αποθηκεύεται η εγγραφή του προβλήματος
3. Βρίσκονται οι τιμές των παραμέτρων του προβλήματος (κλίμακα και τρόπος υπολογισμού).
4. Λαμβάνονται τα δεδομένα των κριτηρίων όπως έχουν παραληφθεί από τη φόρμα html. Τα δεδομένα αυτά, έρχονται από τη φόρμα στη μορφή παράλληλων πινάκων, όπου π.χ. ο πίνακας `critNodeName` κρατά τα ονόματα των κριτηρίων και ο πίνακας `critid` τα αναγνωριστικά τους. Τα στοιχεία του πρώτου κριτηρίου βρίσκονται στα στοιχεία 0 των πινάκων, του δεύτερου στα στοιχεία 1 κ.ο.κ.
5. Μετά τη συλλογή, τα δεδομένα περνάνε από ελέγχους για την εγκυρότητά τους. Αν αποτύχει ο έλεγχος, γίνεται έξοδος.
6. Διαγράφονται, εάν υπάρχουν ζεύγη συγκρίσεων ή υπο-κριτήρια που αναφέρονται σε κριτήρια τα οποία πρόκειται να διαγραφούν (cascade delete – γίνεται πριν τη διαγραφή των ίδιων των κριτηρίων, για λόγους ακεραιότητας των δεδομένων)
7. Διαγράφονται τα κριτήρια που έχει διαγράψει ο χρήστης, αν υπάρχουν.
8. Τα κριτήρια αποθηκεύονται σε δύο φάσεις. Στην πρώτη φάση, αποθηκεύονται τα κριτήρια χωρίς να τιμή στο πεδίο που προσδιορίζει το γονικό κριτήριο. Καθώς αποθηκεύονται, ενημερώνεται ο πίνακας που περιέχει την αντιστοίχιση ψευδο-αναγνωριστικού και πραγματικού αναγνωριστικού.
9. Τα κριτήρια αποθηκεύονται για δεύτερη φορά, περιέχοντας την πληροφορία του γονικού κριτηρίου, όπου υπάρχει.
10. Λαμβάνονται τα δεδομένα των εναλλακτικών.
11. Γίνεται έλεγχος της εγκυρότητας των δεδομένων των εναλλακτικών.
12. Αποθηκεύονται οι εναλλακτικές.
13. Αποθηκεύεται η σειρά εμφάνισης των εναλλακτικών
14. Διαγράφονται οι εναλλακτικές που έχει διαγράψει ο χρήστης.
15. Διαγράφονται ζεύγη συγκρίσεων που αναφέρονται στις διαγραμμένες εναλλακτικές.
16. Γίνεται εκκαθάριση των δεδομένων που δεν είναι πλέον έγκυρα στον πίνακα ζευγών συγκρίσεων.
17. Όταν γίνεται αποθήκευση ενός προβλήματος, η διαδικασία τερματίζει ομαλά σε αυτό το σημείο. Η διαδικασία προχωράει στα επόμενα βήματα, μόνο στις ενέργειες κλωνοποίησης ή εισαγωγής προβλήματος από αρχείο.
18. Επειδή, αν είμαστε σε αυτό το σημείο, κλωνοποιούμε ένα πρόβλημα, θεωρείται, ως σύμβαση, πως όλα τα αναγνωριστικά που περιέχουν τα δεδομένα (είτε αναφέρονται σε εναλλακτικές είτε σε κριτήρια) αναφέρονται στα ψευδο-αναγνωριστικά των εναλλακτικών ή των κριτηρίων.

Οπότε, πριν την αποθήκευση, γίνεται μετατροπή των ψευδο-αναγνωριστικών σε πραγματικά αναγνωριστικά.

19. Τέλος διαδικασίας.

6.5.2. Αντιγραφή προβλήματος

Η διαδικασία αντιγραφής ενός προβλήματος, διαβάζει τα στοιχεία ενός προβλήματος από τη βάση δεδομένων και δημιουργεί τις δομές δεδομένων που περιμένει η διαδικασία αποθήκευσης, την οποία καλεί. Στην ουσία, ξεγελάει τη διαδικασία αποθήκευσης, δίνοντάς της την εντύπωση πως έχει να αποθηκεύσει ένα νέο πρόβλημα που έχει εισάγει ο χρήστης από τη φόρμα.

Αυτό, το πετυχαίνει με τον παρακάτω τρόπο: αφού διαβάσει τα στοιχεία του προβλήματος από τη βάση δεδομένων, δηλώνει το αναγνωριστικό της βάσης δεδομένων ως ψευδο-αναγνωριστικό και μηδενίζει το πραγματικό αναγνωριστικό στη βάση. Αυτό, κάνει τις διαδικασίες αποθήκευσης να θεωρούν όλες τις εγγραφές ως νέες εγγραφές, αλλά, από το ψευδο-αναγνωριστικό, κρατούν την πληροφορία της σύνδεσης.

6.5.3. Εισαγωγή προβλήματος από csv/excel

Το αρχείο που φορτώνεται στο διακομιστή, διαχειρίζεται από το Joomla, το οποίο φροντίζει:

- το αρχείο να μην υπερβαίνει το μέγιστο επιτρεπτό μέγεθος
- να αποθηκευθεί με διαφορετικό όνομα και
- σε διαδρομή μη προσβάσιμη από τον client.

Η διαδικασία, ελέγχει μόνο την επέκταση του αρχείου, ώστε να επιτρέπεται να φορτωθούν μόνο αρχεία csv και excel.

Κατόπιν, το αρχείο δρομολογείται στις διαδικασίες εισαγωγής, οι οποίες περιγράφονται παραπάνω, ανάλογα με την κατάληξή του. Οι διαδικασίες αυτές, δημιουργούν πίνακες από αντικείμενα που προσομοιώνουν εγγραφές στη βάση δεδομένων. Οι ψευδο-εγγραφές αυτές, δίνονται ως όρισμα στη δημιουργία ενός αντικειμένου AhpProblemData και γίνεται έλεγχος για την εγκυρότητά τους. Εάν οι εγγραφές βρεθούν έγκυρες, τότε μόνο προωθούνται στην τελική φάση της διαδικασίας, όπου τα δεδομένα και πάλι ανασυντάσσονται ώστε να μοιάζουν σαν να τα έχει εισάγει χρήστης από τη φόρμα δημιουργίας νέου προβλήματος και παραπέμπονται στη διαδικασία αποθήκευσης που περιγράφεται παραπάνω.

Με άλλα λόγια, η διαδικασία ανάγνωσης του αρχείου, ξεγελά το σύστημα, κάνοντάς το να νομίζει πως πρόκειται για εγγραφές από τη βάση δεδομένων, προκειμένου να τα επαληθεύσει. Με αυτόν τον τρόπο, κάθε πρόβλημα που ζητείται να εισαχθεί από αρχείο, περνάει τους ίδιους ελέγχους που περνάνε όλα τα προβλήματα για την εγκυρότητά τους. Στη συνέχεια, ακολουθείται η ίδια διαδικασία που χρησιμοποιείται στην αντιγραφή.

6.5.4. Εξαγωγή προβλήματος σε csv / excel

Η διαδικασία εκμεταλλεύεται το χαρακτηριστικό του Joomla, το οποίο επιτρέπει την παραγωγή, όχι μόνο τυποποιημένης σελίδας του, αλλά και ελεύθερο περιεχόμενο, εάν αυτό ζητηθεί με την παράμετρο `format=raw`.

Επομένως, η εξαγωγή σε csv και excel, έχουν υλοποιηθεί ως διαφορετικό layout του view `comparisonpairs`. Πρόκειται για το view το οποίο παρουσιάζει τα ζεύγη σύγκρισης και τις προτεραιότητες του προβλήματος (επιλογή View/Edit comparison pairs). Ο κώδικας, και στις δύο περιπτώσεις, είναι ο ίδιος, απλώς έχει προσαρμοστεί στις ιδιαίτερες απαιτήσεις της κάθε περίπτωσης.

6.5.5. Αποθήκευση ζευγών συγκρίσεων

Οι ιδιαιτερότητες της διαδικασίας είναι οι εξής:

- Πριν αποθηκεύσει οποιαδήποτε τιμή, διαγράφει από τη βάση δεδομένων οποιεσδήποτε τιμές αναφέρονται στο ίδιο κριτήριο.

- αποθηκεύει είτε ζεύγη συγκρίσεων, είτε τιμές βαρών που έχουν εισαχθεί απευθείας.
- αποθηκεύει μόνο τα ζεύγη συγκρίσεων που βρίσκονται άνω και δεξιά της διαγωνίου
- αποθηκεύει μόνο ζεύγη συγκρίσεων που δηλώνουν προτίμηση (δηλ. δεν αποθηκεύει τις τιμές I)

7. Οδηγός Χρήσης

7.1. Εγκατάσταση

7.1.1. Προαπαιτούμενα

Η εγκατάσταση της επέκτασης προϋποθέτει:

- Μια λειτουργική εγκατάσταση Joomla έκδοσης 1.7
- Η έκδοση PHP να είναι 5.3

7.2. Διαδικασία εγκατάστασης

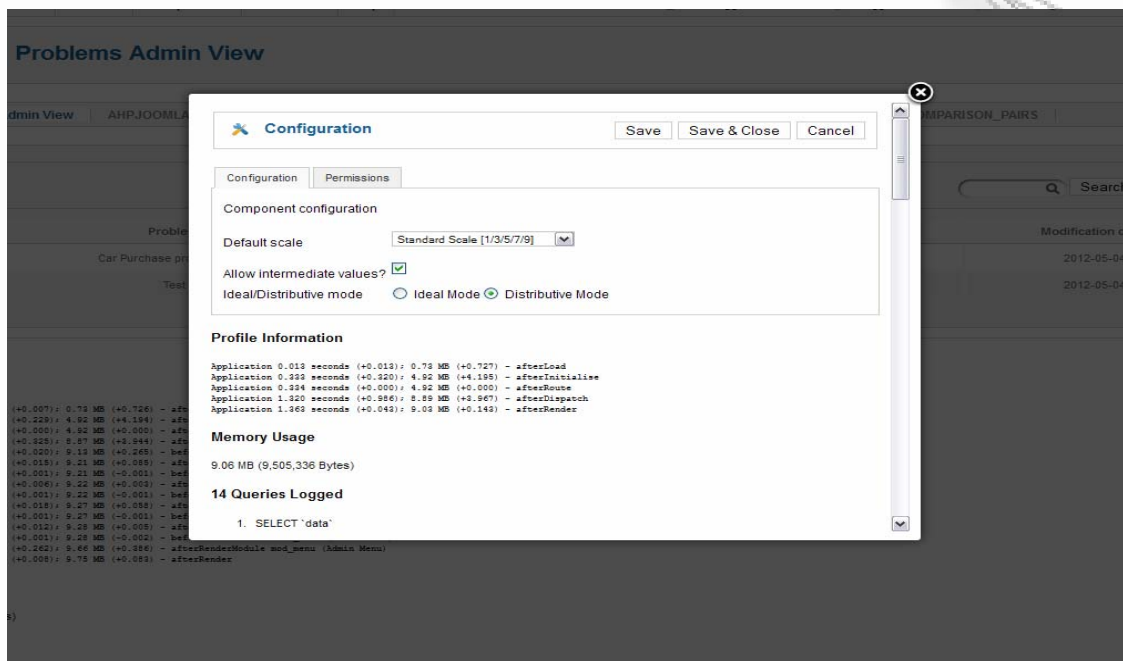
Η εγκατάσταση, γίνεται με την τυποποιημένη διαδικασία του Joomla, από το διαχειριστή επεκτάσεων (extension manager) και την επιλογή εγκατάστασης, επιλέγοντας το αρχείο com_ahp_joomla.tar.gz για φόρτωση.

Σημείωση: έχει παρατηρηθεί, μερικές φορές, το ενδεχόμενο, η πρώτη αίτηση εγκατάστασης να «μπλοκάρει» και να χρειάζεται επανάληψη της αίτησης. Αυτό δεν έχει να κάνει με την επέκταση, αλλά με άλλες συνθήκες (όπως π.χ. ο browser που χρησιμοποιείται).

7.2.1. Μετά την εγκατάσταση

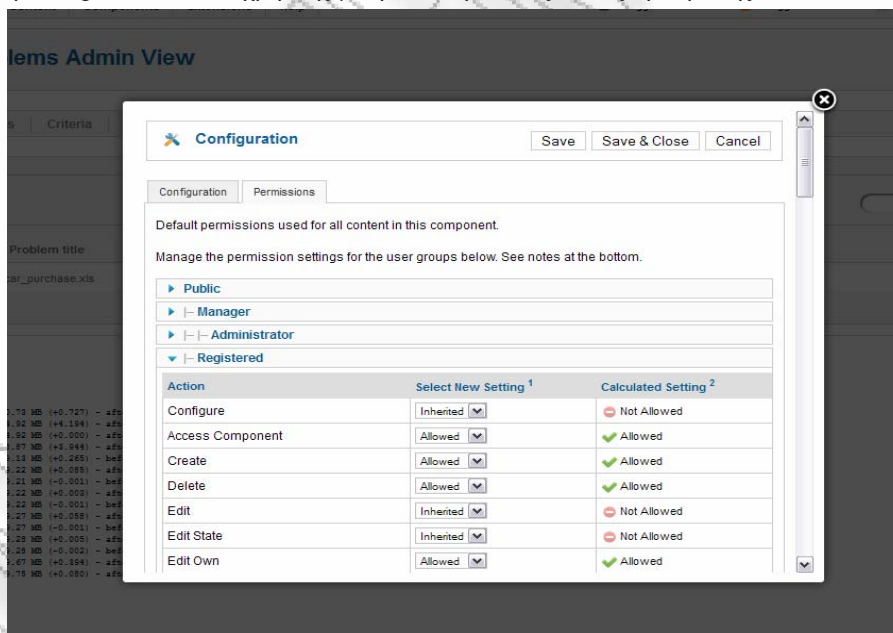
Μετά την εγκατάσταση, ο διαχειριστής της ιστοσελίδας πρέπει να ορίσει τις άδειες (permissions) των χρηστών και την πρόσβαση από το menu.

Οι άδειες, ρυθμίζονται επιλέγοντας τη διαχείριση του component (Extensions > Ahp for Joomla) και, στην οθόνη διαχείρισης που θα εμφανιστεί, επιλέγοντας options. Εμφανίζεται μια οθόνη με δύο καρτέλες: Configuration και Permissions.



Από την καρτέλα Configuration, ο διαχειριστής μπορεί να ορίσει ποια θα είναι η προεπιλεγμένη κλίμακα για τους υπολογισμούς, αν θα επιτρέπονται οι ενδιάμεσες τιμές σε αυτήν, και τον προεπιλεγμένο τρόπο υπολογισμού. Διευκρινίζεται πως αυτές είναι οι προεπιλεγμένες τιμές, τις οποίες ο χρήστης μπορεί να ορίσει χωριστά σε κάθε πρόβλημα

Από την καρτέλα permissions, ο χρήστης μπορεί να ορίσει τις άδειες πρόσβασης.



Οι εξ ορισμού άδειες, επιτρέπουν μόνο στο διαχειριστή να έχει πρόσβαση στην επέκταση. Επομένως, ο διαχειριστής πρέπει οπωσδήποτε να ορίσει άδειες για τους υπόλοιπους χρήστες. Προτείνεται, για τους εγγεγραμμένους χρήστες (registered) να οριστούν οι άδειες Create, Delete και Edit Own στην τιμή Allowed. Με αυτόν τον τρόπο, κάθε χρήστης θα μπορεί να έχει πρόσβαση στην επέκταση, να δει όλα τα προβλήματα, να τροποποιήσει μόνο τα δικά του, να δημιουργήσει, νέα προβλήματα, και να διαγράψει μόνο τα δικά του.

Συγκεκριμένα, κάθε άδεια ελέγχει τα εξής:

- **Configure:** δυνατότητα να θέσει παραμέτρους και άδειες πρόσβασης στην επέκταση. Αυτήν την άδεια πρέπει να την έχουν μόνο οι διαχειριστές.
- **Access component:** δίνει την άδεια στο χρήστη να δει τη λίστα των προβλημάτων, αλλά και να δει (χωρίς να μπορεί να τροποποιήσει) τα προβλήματα που έχουν δημιουργήσει άλλοι χρήστες.
- **Create:** δυνατότητα δημιουργίας νέων προβλημάτων. Συμπεριλαμβάνει τη δυνατότητα αντιγραφής προβλήματος και εισαγωγής.
- **Delete:** δυνατότητα διαγραφής προβλημάτων. Ο χρήστης μπορεί να διαγράψει μόνο τα δικά του προβλήματα, εκτός αν έχει και την άδεια Edit, οπότε μπορεί να διαγράψει και προβλήματα άλλων χρηστών.
- **Edit:** δυνατότητα τροποποίησης προβλημάτων άλλων χρηστών. Εάν ο χρήστης μπορεί να διαγράψει, τότε μπορεί να διαγράψει και προβλήματα άλλων χρηστών.
- **Edit State:** δυνατότητα να μπορεί ο χρήστης να τροποποιήσει προβλήματα που έχουν δεσμευτεί (check-out) από άλλους χρήστες.
- **Edit Own:** δυνατότητα του χρήστη να τροποποιήσει δικά του προβλήματα.

Τέλος, ο διαχειριστής πρέπει να ορίσει ένα σημείο πρόσβασης της εφαρμογής από το μενού. Αυτό μπορεί να δημιουργηθεί με την τυποποιημένη διαδικασία του Joomla.

7.3. Λειτουργίες διαχείρισης

Η επέκταση, από την πλευρά της διαχείρισης, δεν προσφέρει τίποτε παραπάνω από έναν τρόπο άμεσης διαγραφής εγγραφών οι οποίες ενδεχομένως να δημιουργούν πρόβλημα στα δεδομένα.— ενδεχόμενο απίθανο, το οποίο όμως ποτέ δεν μπορεί να αποκλειστεί. Ουσιαστικά λοιπόν μιλάμε για ένα κέλυφος της βάσης δεδομένων, το οποίο όμως, σημειώνεται, δεν προσφέρει cascade delete. Κατά τα άλλα, παρέχει έναν καλό μηχανισμό φιλτραρίσματος των εγγραφών ώστε να εντοπίζονται γρήγορα οι εγγραφές που θέλουμε να διαγραφούν.

7.4. Λειτουργίες χρήστη

7.4.1. Κατάλογος προβλημάτων

Είναι η αρχική οθόνη από την οποία γίνεται πρόσβαση στα δεδομένα των προβλημάτων. Παρουσιάζει τον τίτλο του προβλήματος, τον δημιουργό του, το χρόνο δημιουργίας του προβλήματος και το χρόνο της τελευταίας τροποποίησής του. Ο χρήστης, από αυτήν την οθόνη, μπορεί:

- Να δημιουργήσει νέο πρόβλημα, από την επιλογή New.
- Να δει ή/και να τροποποιήσει τη δομή ενός επιλεγμένου προβλήματος (εναλλακτικές, ιεραρχία κριτηρίων), από την επιλογή View/Edit Structure.
- Να διαγράψει τα επιλεγμένα προβλήματα.
- Να αντιγράψει ένα υπάρχον επιλεγμένο πρόβλημα
- Να δει την τρέχουσα λύση ενός προβλήματος, ή και να τροποποιήσει τα ζεύγη συγκρίσεων (από την επιλογή View/Edit Comparison Pairs)
- Να εισάγει ένα πρόβλημα από αρχείο CSV ή Excel (επιλογή Import from Excel/CSV)
- Να εξάγει το επιλεγμένο πρόβλημα σε αρχείο Excel (επιλογή Export to Excel)
- Να εξάγει το επιλεγμένο πρόβλημα σε αρχείο CSV (επιλογή Export to CSV)
- Να αναπροσδιορίσει τη σειρά εμφάνισης των προβλημάτων.

Η οθόνη, επίσης, παρέχει τη δυνατότητα φιλτραρίσματος. Πάνω από τη λίστα προβλημάτων, αριστερά, ο χρήστης μπορεί να επιλέξει τον χρήστη του οποίου τα προβλήματα επιθυμεί να εμφανίζονται, ενώ, δεξιά, μπορεί να πραγματοποιήσει αναζήτηση ενός προβλήματος με βάση τον τίτλο του. Η αναζήτηση που γίνεται, κάνει διάκριση σε πεζά και κεφαλαία, π.χ. αν αναζητηθεί ο όρος "PURCHASE", δε θα επιστραφούν προβλήματα που περιέχουν το αλφαριθμητικό «purchase».

Στο αριστερό μέρος του καταλόγου, υπάρχει ένα τετραγωνίδιο επιλογής. Αυτό είναι διαθέσιμο μόνο για προβλήματα τα οποία δεν έχουν δεσμευτεί από κάποιον άλλο χρήστη. Αν δηλαδή, κατά τη στιγμή της εμφάνισης της λίστας, κάποιος χρήστης τροποποιεί κάποιο πρόβλημα, αυτό δε θα έχει αριστερά του τετραγωνίδιο επιλογής.

AHP Problems List

Filter

<input type="checkbox"/>	Problem title	Order	Owner	Creation date	Modification date
<input checked="" type="checkbox"/>	Car Purchase (Wikipedia)	1	Super User	2012-05-05 22:49	2012-05-05 22:51
<input type="checkbox"/>	Test 1	2	Super User	2012-05-08 12:47	2012-05-08 12:47

Display #

7.4.2. New

Εμφανίζεται στο χρήστη μια οθόνη με τρεις επιλογές: Cancel (επιστροφή στην προηγούμενη οθόνη χωρίς αλλαγές), Apply (αποθήκευση των αλλαγών και παραμονή στην ίδια οθόνη) και Save (αποθήκευση των αλλαγών κι επιστροφή στην προηγούμενη οθόνη). Ο χρήστης μπορεί να εισάγει τις παρακάτω πληροφορίες:

Στο πεδίο Problem Title, μπορεί να εισάγει τον τίτλο του προβλήματος, μέχρι 50 χαρακτήρες.

Πατώντας στο σύνδεσμο “Advanced Parameters”, μπορεί να καθορίσει την κλίμακα που θα χρησιμοποιήσει στο πρόβλημα, και τον τρόπο υπολογισμού, συγκεκριμένα, από το πεδίο Scale, επιλέγει την κλίμακα, από το πεδίο allow intermediate values, αν θα είναι διαθέσιμες στο χρήστη οι ενδιάμεσες τιμές ή όχι και, από το calculation mode, τον τρόπο υπολογισμού.

Edit AHP Problem

New

Problem title:

[Advanced Parameters](#)

Scale:

Allow intermediate values: No Yes

Calculation mode: Ideal Mode Distributive Mode

Criteria Tree

- Criteria 1
- Criteria 2
- Criteria 3
- Criteria 1.1

[Add new criteria](#)

Alternatives List

[Add new alternative](#)

Εισάγοντας κείμενο στο πεδίο αριστερά από το σύνδεσμο «Add new criteria», ο χρήστης εισάγει νέα κριτήρια. Το πεδίο εισάγεται, είτε πατώντας στο σύνδεσμο «Add new criteria», είτε πατώντας το πλήκτρο enter. Κατόπιν, ο χρήστης μπορεί να ορίσει την ιεραρχία με drag & drop, όπως φαίνεται στην εικόνα.

Η εισαγωγή των εναλλακτικών, γίνεται με τον ίδιο ακριβώς τρόπο, μέσω του πεδίου αριστερά από το λεκτικό «Add new alternative». Ο χρήστης μπορεί να ελέγξει τη σειρά εμφάνισης των εναλλακτικών μέσω drag & drop.

Ο χρήστης μπορεί να διαγράψει εναλλακτικές ή κριτήρια, πατώντας πάνω στο εικονίδιο της διαγραφής το οποίο βρίσκεται πάντα δεξιά από το λεκτικό κάθε εναλλακτικής ή κριτηρίου.

Μπορεί επίσης να διορθώσει το λεκτικό κάποιου κριτηρίου ή εναλλακτικής, κάνοντας διπλό κλικ πάνω σε αυτό, και πατώντας enter όταν ολοκληρώσει τη διόρθωση. (Βλπ. Παρακάτω εικόνα)

Ο χρήστης, μπορεί, όποτε θέλει, να αποθηκεύσει τις αλλαγές του, πατώντας Apply ή Save, ή να ακυρώσει τις αλλαγές και, επιλέγοντας cancel να επιστρέψει στην προηγούμενη οθόνη (κατάλογος προβλημάτων).

7.4.3. View/Edit structure

Είναι η ίδια οθόνη που καλείται από την επιλογή «New», με τη διαφορά πως φορτώνει το ήδη επιλεγμένο πρόβλημα.

Ωστόσο, η αλλαγή της δομής ενός προβλήματος, αφού έχουν εισαχθεί προηγουμένως ζεύγη σύγκρισης, μπορεί να προκαλέσει απώλεια δεδομένων στις συγκρίσεις. Πιο συγκεκριμένα:

- Διαγραφή εναλλακτικής θα προκαλέσει διαγραφή των ζευγών σύγκρισης γι' αυτήν την εναλλακτική
- Διαγραφή ή μετακίνηση κριτηρίων, θα προκαλέσει τη διαγραφή των ζευγών σύγκρισης που αφορούν το συγκεκριμένο κριτήριο και τα θυγατρικά του
- Αλλαγή της κλίμακας υπολογισμού, και ειδικότερα αφαίρεση των ενδιάμεσων τιμών, δε θα αλλοιώσει τις τιμές όταν εμφανιστεί ο πίνακας σύγκρισης, αλλά θα τις αλλοιώσει (προς ένα από τα πλησιέστερα σημεία της κλίμακας) κατά την επεξεργασία.

Γι' αυτό το λόγο, εάν υπάρχουν αποθηκευμένα δεδομένα στα ζεύγη σύγκρισης, η εφαρμογή προειδοποιεί με μήνυμα το χρήστη.

7.4.4. Delete

Η ενέργεια επιτρέπεται μόνο εάν ο χρήστης έχει το δικαίωμα Delete. Εν συνέχεια, αν έχει το δικαίωμα Edit Own, μπορεί να διαγράψει μόνο προβλήματα που έχει δημιουργήσει ο ίδιος, ενώ αν έχει το δικαίωμα Edit, μπορεί να διαγράψει οποιοδήποτε πρόβλημα.

Ο χρήστης μπορεί να διαγράψει μαζικά προβλήματα, επιλέγοντας παραπάνω από ένα. Η ενέργεια, διαγράφει όχι μόνο το πρόβλημα, αλλά και τα στοιχεία του (εναλλακτικές, κριτήρια, ζεύγη σύγκρισης) από τους αντίστοιχους πίνακες (cascade delete).

7.4.5. Copy

Η ενέργεια γίνεται σε ένα μόνο πρόβλημα. Εάν έχουν επιλεγεί πάνω από ένα προβλήματα, θα γίνει στο πρώτο από αυτά. Η ενέργεια αντιγράφει το πρόβλημα πλήρως, δηλ. το πρόβλημα, τη δομή

View Comparison Pairs

Alternative Priorities for Car Purchase (Wikipedia)			
Alternatives	Local Priority	Global Priority	
Accord Sedan	0.1990	0.1990	
Accord Hybrid	0.1405	0.1405	
Pilot SUV	0.1082	0.1082	
CR-V SUV	0.1754	0.1754	
Element SUV	0.1486	0.1486	
Odyssey Minivan	0.2264	0.2264	

Comparison Matrix under criteria Car Purchase (Wikipedia)							
Set AHP Comparison Pairs	Cost	Safety	Style	Capacity	Local Priority	Global Priority	S.A.
Cost	1.0000	1.0000	7.0000	3.0000	0.5101	0.5101	
Safety	0.3333	1.0000	9.0000	1.0000	0.2344	0.2344	
Style	0.1429	0.1111	1.0000	0.1429	0.0405	0.0405	
Capacity	0.3333	1.0000	7.0000	1.0000	0.2191	0.2191	

Alternative Priorities for Cost			
Alternatives	Local Priority	Global Priority	
Accord Sedan	0.2003	0.1002	
Accord Hybrid	0.1032	0.0516	
Pilot SUV	0.0808	0.0310	
CR-V SUV	0.2550	0.1301	
Element SUV	0.2514	0.1262	
Odyssey Minivan	0.1213	0.0619	

Comparison Matrix under criteria Cost						
Set AHP Comparison Pairs	Purchase Price	Fuel Economy	Resale Value	Local Priority	Global Priority	S.A.
Purchase Price	1.0000	1.0000	6.0000	0.4862	0.3926	

Μεταπτυχιακή Διατριβή

ωάννης

(εναλλακτικές και κριτήρια) και τα στοιχεία σύγκρισης. Το όνομα του αντεγραμμένου προβλήματος είναι το «Copy of» συμπληρωμένο από το όνομα του αρχικού προβλήματος.

7.4.6. View/Edit Comparison Pairs

Από την ενέργεια αυτή ο χρήστης μπορεί να δει τη συνολική εικόνα του προβλήματος, με τα τοπικά και καθολικά βάρη των κριτηρίων, τις τοπικές και καθολικές προτεραιότητες των λύσεων και τα στοιχεία των ζευγών σύγκρισης. Επίσης, παρέχει στο χρήστη τη δυνατότητα:

- Να ορίσει τα ζεύγη σύγκρισης
- Να θέσει απευθείας τα βάρη ή τις προτεραιότητες
- Να κάνει ανάλυση ευαισθησίας ενός κριτηρίου ή μιας εναλλακτικής.

Η οθόνη, παρουσιάζει μια εικόνα του προβλήματος ξεκινώντας από τη ρίζα του και προχωρώντας στο δέντρο των κριτηρίων κατά πλάτος. Δηλ. τα κριτήρια εμφανίζονται ανάλογα με την απόστασή τους από τη ρίζα. Στην κορυφή, βρίσκεται ένας πίνακας με τις τελικές προτεραιότητες των εναλλακτικών. Ακολουθούν οι πίνακες σύγκρισης των κριτηρίων, και, τέλος, οι πίνακες σύγκρισης μεταξύ των εναλλακτικών, σε όποια κριτήρια δεν αναλύονται περαιτέρω.

Επίσης, εμφανίζεται πίνακας με τις προτεραιότητες των εναλλακτικών (χωρίς πίνακες σύγκρισης) για κάθε κριτήριο το οποίο αναλύεται σε μικρότερα.

Σε κάθε πίνακα που παρουσιάζει ζεύγη σύγκρισης μεταξύ κριτηρίων, οι ονομασίες των κριτηρίων στην αριστερή στήλη, είναι σύνδεσμοι που οδηγούν στην ίδια σελίδα, στον πίνακα σύγκρισης για το συγκεκριμένο υπο-κριτήριο (Σημ. Το συγκεκριμένο χαρακτηριστικό λειτουργεί μόνο όταν δεν έχουν ενεργοποιηθεί οι φιλικοί σύνδεσμοι προς μηχανές αναζήτησης).

Ο χρήστης μπορεί να ορίσει τα ζεύγη σύγκρισης ή να θέσει απευθείας τα βάρη/τις προτεραιότητες κάνοντας κλικ πάνω στο σύνδεσμο «Set AHP Comparison Pairs», ή «Set Weights Directly» ο οποίος βρίσκεται στην άνω αριστερά γωνία κάθε πίνακα σύγκρισης. Σημειώνεται πως προεπιλογή είναι η εισαγωγή ζευγών σύγκρισης, δηλ. σε πρόβλημα στο οποίο δεν έχουν οριστεί στοιχεία σύγκρισης, η εφαρμογή θα παρουσιάσει την επιλογή «Set AHP Comparison Pairs». Καλώντας αυτήν την ενέργεια, μετά, ο χρήστης μπορεί να επιλέξει τον απευθείας ορισμό βαρών προτεραιοτήτων.

Μπορεί επίσης να εμφανίσει ένα γράφημα ανάλυσης ευαισθησίας για κάθε κριτήριο ή εναλλακτική, πατώντας το εικονίδιο του μεγεθυντικού φακού στην αριστερή στήλη, στη σειρά του κριτηρίου ή της εναλλακτικής που τον ενδιαφέρει.

Τέλος, εάν κάποιος από τους πίνακες σύγκρισης υπερβαίνει τα επιτρεπτά όρια συνέπειας (ο δείκτης συνέπειας δηλαδή είναι άνω του 10%), εμφανίζεται ενημερωτικό μήνυμα στην κορυφή της σελίδας το οποίο τον ειδοποιεί για τον πίνακα που εμφανίζει την ασυνέπεια.

7.4.7. View/Edit Comparison Pairs -> Set AHP Comparison Pairs

Η ενέργεια εμφανίζει τις εξής επιλογές στο χρήστη:

- Αλλαγή σε απευθείας ορισμό βαρών/προτεραιοτήτων (Set Weights Directly)
- Ακύρωση (Cancel)
- Εφαρμογή αλλαγών (Apply)
- Αποθήκευση (Save)

Set AHP Comparison Pairs

Set priorities for criteria Car Purchase (Wikipedia)

Cost	<input type="range"/>	Safety
Cost	<input type="range"/>	Style
Cost	<input type="range"/>	Capacity
Safety	<input type="range"/>	Style
Safety	<input type="range"/>	Capacity
Style	<input type="range"/>	Capacity

	Cost	Safety	Style	Capacity	Priorities
Cost	1.0000	3.0000	7.0000	3.0000	0.5101
Safety	0.3333	1.0000	9.0000	1.0000	0.2344
Style	0.1429	0.1111	1.0000	0.1429	0.0405
Capacity	0.3333	1.0000	7.0000	1.0000	0.2151
CR:					7.4600

Κάτω από τις επιλογές, του παρουσιάζονται όλα τα διαθέσιμα ζεύγη σύγκρισης, με έναν slider. Ο χρήστης, προκειμένου να ορίσει την προτίμησή του σε κάθε ζεύγος, σέρνει το δείκτη του slider προς το μέρος της προτιμότερης λύσης (ή του ισχυρότερου κριτηρίου), τόσο, όσο μεγαλύτερη είναι η προτίμηση. Καθώς το κάνει αυτό, ενημερώνεται σε πραγματικό χρόνο ένας πίνακας από κάτω ο οποίος αναπαριστά τον πίνακα σύγκρισης, υπολογίζει τα βάρη ή τις προτεραιότητες σε τοπικό επίπεδο, κι εμφανίζει, στην τελευταία του γραμμή, το δείκτη συνέπειας. Εάν ο δείκτης υπερβεί το επιτρεπόμενο όριο, εμφανίζεται με κόκκινο χρώμα. Τυχόν υπέρβαση δε θα αποτρέψει την αποθήκευση των ζευγών. Απλώς, θα εμφανίσει ένα πληροφοριακό μήνυμα, στην εικόνα του προβλήματος (βλπ. επιλογή View/Edit Comparison Pairs). Ο χρήστης δεν είναι υποχρεωμένος να εισάγει ζεύγη σύγκρισης. Επιλέγοντας την ενέργεια «Set Weights Directly», μεταφέρεται σε οθόνη απευθείας ορισμού βαρών / προτεραιοτήτων. Στην περίπτωση που επιλέξει αυτήν την ενέργεια, εάν έχει εισάγει ήδη κάποια ζεύγη σύγκρισης, τα δεδομένα δε θα χαθούν παρά μόνο όταν αποθηκεύσει τα στοιχεία της απευθείας σύγκρισης (στην νέα οθόνη).

7.4.8. View/Edit Comparison Pairs -> Set Weights Directly

Η ενέργεια εμφανίζει τις εξής επιλογές στο χρήστη:

- Αλλαγή σε ορισμό ζευγών συγκρίσεων (Set AHP Comparison Pairs)
- Ακύρωση (Cancel)
- Εφαρμογή αλλαγών (Apply)
- Αποθήκευση (Save)

Set AHP Comparison Pairs

Set priorities for criteria Fuel Costs

Higher value means better

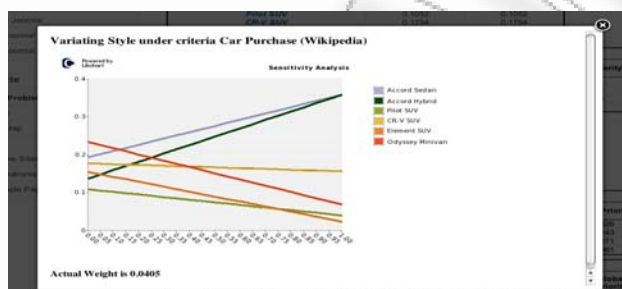
Alternative A > Alternative Name	Value
Accord Sedan	<input type="text" value="31"/>
Accord Hybrid	<input type="text" value="35"/>
Pilot SUV	<input type="text" value="22"/>
CR-V SUV	<input type="text" value="27"/>
Element SUV	<input type="text" value="25"/>
Odyssey Minivan	<input type="text" value="26"/>

Κάτω από τις επιλογές, εμφανίζονται οι εναλλακτικές για τις οποίες ο χρήστης μπορεί να εισάγει μια πραγματική τιμή. Μεγαλύτερη τιμή σημαίνει μεγαλύτερο βάρος / υψηλότερη προτεραιότητα.

Όπως και στην αμέσως προηγούμενη επιλογή, ο χρήστης μπορεί να επιστρέψει στον ορισμό ζευγών συγκρίσεων. Αν έχει ορίσει ήδη απευθείας βάρη, τα δεδομένα δε θα χαθούν παρά αφού αποθηκεύσει τα στοιχεία των ζευγών σύγκρισης (στην νέα οθόνη).

7.4.9. View/Edit Comparison Pairs-> Ανάλυση Ευαισθησίας

Η επιλογή ενεργοποιείται επιλέγοντας το μεγεθυντικό φακό σε κάποιο κριτήριο ή κάποια εναλλακτική, στην εικόνα του προβλήματος (βλπ. View/Edit Comparison Pairs). Εμφανίζει ένα γράφημα, το οποίο δείχνει την τιμή που θα είχε η τελική λύση του προβλήματος, εάν το βάρος ή η προτεραιότητα της επιλογής κυμαίνονταν μεταξύ [0..1], με τις άλλες εναλλακτικές να διατηρούν την ίδια αναλογία μεταξύ τους. Στην κορυφή, διευκρινίζεται το στοιχείο του προβλήματος το οποίο αφορά η ανάλυση ευαισθησίας. Στο δεξιό μέρος του γραφήματος, εμφανίζεται υπόμνημα το οποίο συνδέει τα χρώματα με τα στοιχεία που παρουσιάζονται. Στο κάτω μέρος, εμφανίζεται το πραγματικό βάρος του επιλεγμένου στοιχείου, ώστε ο χρήστης να γνωρίζει σε ποιο σημείο πρέπει να εστιάσει..



7.4.10. Import from Excel/CSV

Η ενέργεια, εμφανίζει στο χρήστη μια οθόνη με τα παρακάτω πεδία:

- Επιλογή αρχείου: επιλέγεται το αρχείο το οποίο θα φορτωθεί
- Τίτλο προβλήματος: (προαιρετικό) Ο τίτλος του προβλήματος που θα εισαχθεί. Σε περίπτωση που δε δοθεί, το πρόβλημα θα λάβει ως τίτλο το όνομα του αρχείου που φορτώθηκε (μαζί με την κατάληξη)

Οι επόμενες παράμετροι χρησιμοποιούνται μόνο για την εισαγωγή αρχείων CSV

- Delimiter: ο χαρακτήρας που διαχωρίζει τα πεδία μεταξύ τους. Έχει προεπιλεγμένη την τιμή το χαρακτήρα «;» που είναι ο χαρακτήρας με τον οποίο διαχωρίζονται τα πεδία μεταξύ τους σε αρχεία που έχουν παραχθεί από Excel.
- Quote Char: ο χαρακτήρας που περικλείει τα αλφαριθμητικά. Έχει προεπιλεγμένη τιμή το χαρακτήρα «"» που είναι ο χαρακτήρας ο οποίος περικλείει τα αριθμητικά σε αρχεία που έχουν παραχθεί από Excel.

Επιλέγοντας «Submit», το αρχείο υποβάλλεται στην εφαρμογή και ο χρήστης επιστρέφει στον κατάλογο των προβλημάτων. Επιτυχία στην εισαγωγή θα εμφανίσει ένα σχετικό μήνυμα, ενώ τυχόν αποτυχία θα εμφανίσει ένα ή περισσότερα μηνύματα τα οποία θα εξηγούν τους λόγους της αποτυχίας.

Set AHP Comparison Pairs

Set priorities for criteria Fuel Costs

Higher value means better

Alternative A > Alternative Name	Value
Accord Sedan	31
Accord Hybrid	35
Pilot SUV	22
CR-V SUV	27
Element SUV	25
Odyssey Minivan	26

7.4.11. Export to Excel, Export to CSV

Με την επιλογή αυτή, μεταφορτώνεται στο χρήστη το πλήρες πρόβλημα σε μορφή Excel ή CSV.

7.5. Μορφή του αρχείου προς εισαγωγή (Excel/CSV)

Το αρχείο που δίνεται προς εισαγωγή πρέπει να συμμορφώνεται στους παρακάτω κανόνες.

- Όλα τα δεδομένα αρχίζουν από την πρώτη στήλη
- Το αρχείο χωρίζεται σε blocks, δηλ. σε σύνολα από συνεχείς σειρές. Τα blocks είναι οι ενότητες που διαχωρίζονται από κενές γραμμές.
- Από την πρώτη γραμμή, ξεκινάει η περιγραφή των κριτηρίων και των εναλλακτικών. Στην πρώτη στήλη, εμφανίζονται τα κριτήρια πρώτου επιπέδου, στη δεύτερη του δεύτερου επιπέδου κ.ο.κ. Μετά τη στήλη των κριτηρίων τελευταίου επιπέδου, και χωρίς ενδιάμεση στήλη, βρίσκεται η στήλη των εναλλακτικών.
- Κάτω από κάθε κριτήριο ανώτερου επιπέδου, εισάγονται κενά κελιά, μέχρι να έχουν αναπτυχθεί όλα τα υποκριτήρια, μέχρι το τελευταίο επίπεδο.

Παράδειγμα: έστω η δομή με το δέντρο κριτηρίων:

- Κριτήριο 1
 - Κριτήριο 1.1
 - Κριτήριο 1.2
 - Κριτήριο 1.2.1
 - Κριτήριο 1.2.2
 - Κριτήριο 1.3
- Κριτήριο 2
- Κριτήριο 3

Και τις εναλλακτικές

- Εναλλακτική Α
- Εναλλακτική Β
- Εναλλακτική Γ

Θα αναπαρασταθεί ως εξής:

Κριτήριο 1	Κριτήριο 1.1		Εναλλακτική Α
	Κριτήριο 1.2	Κριτήριο 1.2.1	Εναλλακτική Β
		Κριτήριο 1.2.2	Εναλλακτική Γ
	Κριτήριο 1.3		
	Κριτήριο 1.4		
Κριτήριο 2			
Κριτήριο 3			

Η παραπάνω δομή, θα πρέπει υποχρεωτικά να ακολουθείται από μια κενή γραμμή.

Οι κανόνες των blocks, είναι οι εξής:

- Το άνω αριστερά γωνιακό κελί του block, ορίζει υποχρεωτικά το κριτήριο στο οποίο αναφέρεται ο πίνακας σύγκρισης και πρέπει να είναι το ίδιο ακριβώς με την ονομασία ενός κριτηρίου που έχει οριστεί από το κορυφαίο block. Μόνη εξαίρεση είναι το block των κορυφαίων κανόνων, το οποίο μπορεί να έχει κενό άνω αριστερό κελί.
- Blocks των οποίων το άνω αριστερά γωνιακό κελί περιέχει αλφαριθμητικό το οποίο ξεκινά με το χαρακτήρα # αγνοούνται πλήρως. Αυτή η δυνατότητα υπάρχει ώστε να μπορούν να υπάρχουν blocks με συγκεντρωτικές προτεραιότητες εναλλακτικών.
- Τα blocks τα οποία ορίζουν ζεύγη σύγκρισης, πρέπει να ορίζουν στοιχεία σύγκρισης με τον ίδιο γονέα, και πρέπει όλα τα στοιχεία της πρώτης στήλης (πλην του κορυφαίου) να ταυτίζονται με τα πρώτα στοιχεία της πρώτης σειράς. Οι επόμενες στήλες αγνοούνται (αυτό γίνεται ώστε να μπορούν να συμπληρωθούν οι προτεραιότητες που προκύπτουν).
- Blocks τα οποία δε συμμορφώνονται στον παραπάνω κανόνα, θεωρούνται πως ορίζουν απευθείας προτεραιότητες και λαμβάνεται υπόψη μόνο η πρώτη τους στήλη.
- Ειδικά τα αρχεία csv, πρέπει να τελειώνουν οπωσδήποτε με μια κενή γραμμή, αλλιώς δε θα εισαχθεί η τελευταία (πρόκειται για bug της βιβλιοθήκης που χρησιμοποιείται).

Ο χρήστης μπορεί να έχει ένα παράδειγμα της μορφής που πρέπει να έχει ένα πρόβλημα, κάνοντας εξαγωγή ένα πρόβλημα σε μορφή excel ή csv. Το αρχείο csv μπορεί να επανα-εισαχθεί, όπως είναι. Το αρχείο Excel, πρέπει προηγουμένως να αποθηκευθεί σε μορφή xls. (Αυτό συμβαίνει διότι η εφαρμογή παράγει ένα αρχείο xml το οποίο συμμορφώνεται στα standards του Microsoft Excel, δεν είναι όμως η μορφή την οποία αναμένει η εφαρμογή.)

7.6. Περιορισμοί

Οι περιορισμοί της υλοποίησης αυτής είναι οι παρακάτω:

- Το πρόβλημα που ορίζεται, πρέπει να αποτελείται οπωσδήποτε από, το λιγότερο δύο κριτήρια και δύο εναλλακτικές. Λιγότερα κριτήρια ή εναλλακτικές θα προκαλέσουν σφάλμα.
- Επειδή το Joomla δεν υποστηρίζει transactions, υπάρχει περίπτωση, τα στοιχεία κάποιου προβλήματος να μην αποθηκευτούν στο σύνολό τους. Έχει ληφθεί κάθε δυνατό μέτρο να αποφευχθεί αυτό το σενάριο και είναι αρκετά απίθανο, ωστόσο δεν μπορεί να αποκλειστεί.
- Όταν ο χρήστης εισάγει απευθείας βάρη ή προτεραιότητες, υψηλότερη τιμή θεωρείται υψηλότερη προτεραιότητα. Δεν έχει υλοποιηθεί τρόπος ώστε να υπολογίζεται αντιστρόφως ανάλογη σχέση, ούτε μη γραμμική σχέση.

7.7. Εκκρεμότητες

Υπάρχουν οι παρακάτω εκκρεμότητες οι οποίες αφορούν τον υπολογισμό του δείκτη συνέπειας.

Στον ιδεατό τρόπο υπολογισμού, ο λόγος συνέπειας είναι διαφορετικός αν υπολογιστεί στον αρχικό πίνακα, ή εάν υπολογιστεί πάνω στον πίνακα με τις αρχικές τιμές. Αυτό έχει ως αποτέλεσμα, πίνακες συγκρίσεων, να φαίνονται πως υπερβαίνουν τα όρια στον έναν τρόπο υπολογισμού και πως δεν τα υπερβαίνουν στον άλλο τρόπο. Για να λυθεί αυτή η αντίφαση, ο λόγος συνέπειας υπολογίζεται πάντα πάνω στις αρχικές τιμές.

Όταν χρησιμοποιούνται άλλες κλίμακες, πέραν της τυποποιημένης (1/3/5/7/9), είναι πρακτικά αδύνατον ο δείκτης συνέπειας να παραμείνει μέσα στα αποδεκτά όρια. Δεν έχει εφαρμοστεί λύση σε αυτό.

8. Βιβλιογραφία

- Δεσπότης, Δ. (2002). Ιεραρχική Ανάλυση Αποφάσεων, σημειώσεις διαλέξεων για το μάθημα «Υποστηρίξη Αποφάσεων». Πειραιάς: Πανεπιστήμιο Πειραιώς.

- Alonso, J. A.-L. (2006). Consistency in the Analytic Hierarchy Process: A new approach. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* , 14 (4), 445-449.
- Lanham, C., & Kennard, J. (2010). *Mastering Joomla! 1.5 Extension and Framework development* (2nd edition εκδ.). Birmingham - Mumbai: Packt Publishing.
- Saaty, T. (1977). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology* , 15 (3), 234-281.
- Saaty, T. L. (2001). *Fundamentals of Decision Making and Priority Theory*. Pittsburgh, Pennsylvania: RWS Publications.
- Saaty, T. L., & Ozdemir, M. S. (2003). Why the Magic Number Seven Plus or Minus two. *Mathematical and Computer Modelling* , 38, 233-244.
- Saaty, T. (2005). The Analytic Hierarchy and Analytic Network Processes for the Measurement of Intangible Criteria and for Decision-Making. Στο Σ. τόμος, J. Figueira, S. Greco, & M. Ehrgott (Επιμ.), *Multiple Criteria Decision Analysis: State of the art surveys* (σφ. 346-408). Boston: Springer Science + Business Media, Inc.