

ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΗΣΙΜΑ ΣΥΣΤΑΤΙΚΑ ΒΑΣΙΣΜΕΝΑ ΣΕ
AJAX ΓΙΑ ΔΙΑΔΙΚΤΥΑΚΕΣ ΕΦΑΡΜΟΓΕΣ

ΚΩΝΣΤΑΝΤΙΝΟΥ ΓΕΩΡΓΙΟΣ

ΜΕΤΑΤΠΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων Καθηγητής: Χατζηγεωργίου Αλέξανδρος

Εξεταστές: Γεωργιάδης Χρήστος

Τμήμα Εφαρμοσμένης Πληροφορικής

Πανεπιστήμιο Μακεδονίας

Θεσσαλονίκη

Ιούνιος 2008

REUSABLE AJAX BASED COMPONENTS FOR WEB
APPLICATIONS

KONSTANTINOU GEORGIOS

MASTER THESIS

Supervisor: Chatzigeorgiou Alexandros

Examiners: Georgiadis Christos

Department of Applied Informatics

University of Macedonia

Thessaloniki, Greece

June 2008

Στους γονείς μου

Copyright © Κωνσταντίου Γεώργιος (<http://www.csd.uoc.gr/~gconst>), 2008-05-25

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Περίληψη

Έχει γίνει μεγάλη προσπάθεια από την κοινότητα του λογισμικού για να δημιουργηθούν εφαρμογές που βασίζονται σε συστατικά. Παράλληλα, το διαδίκτυο έχει εξελιχθεί σε ένα μέσο κατάλληλο για την φιλοξενία εφαρμογών. Ως εκ τούτου, εμφανίστηκε η ανάγκη για διαδικτυακές εφαρμογές που βασίζονται σε συστατικά και ένας νέος επιστημονικός τομέας γεννήθηκε. Στην παρούσα μεταπτυχιακή εργασία, πραγματοποιείται μια έρευνα επάνω σε αυτόν τον τομέα. Τα αποτελέσματα της έρευνας χρησιμοποιούνται για να οριστεί ένα μοντέλο συστατικών που αποτελείται από μια πλατφόρμα συστατικών πελάτη (client), μια πλατφόρμα συστατικών εξυπηρετή (server) και ένα σύνολο από προδιαγραφές συστατικών. Η πλατφόρμα συστατικών υλοποιείται και χρησιμοποιείται έτσι ώστε να αναπτυχθεί ένα ρεαλιστικό, λειτουργικό παράδειγμα ηλεκτρονικού καταστήματος, που είναι συνεπές με τις προδιαγραφές του μοντέλου συστατικών. Η πλατφόρμα συστατικών επεκτείνεται έτσι ώστε να επιτρέπει την υιοθέτηση συστατικών σε διαδικτυακές εφαρμογές που φιλοξενούνται σε εξυπηρετές τρίτων. Αυτό δίνει την δυνατότητα αναδιανομής λειτουργικών κομματιών της εφαρμογής. Μέσα από το παράδειγμα, παρουσιάζεται ο κύκλος ζωής μιας διαδικτυακής εφαρμογής που βασίζεται σε συστατικά καθώς και όλα τα πλεονεκτήματα που αυτή παρουσιάζει (εύκολη συντήρηση, εύκολη επέκταση, εύκολη υιοθέτηση λειτουργιών σε νέα συστήματα).

Abstract

There has been a major effort in the software community to create component based applications. In the meanwhile, the web has evolved into a perfect medium for hosting application. Thus, the need for component based web applications has risen and a new science field was born. In this master thesis, an exploration and research on this field is made. The research results are used in order to define a component model which comprises from a client side component framework, a server side component framework and set of component specifications. The component framework is being implemented and used in order to develop a functional, realistic example of an e-shop. The example web application is comprised of reusable, redistributable components that follow component model's specifications. Component framework is extended in order to allow adaptation of components in web applications that are hosted in third party servers. This gives the ability to redistribute functional parts of the application. Through this example, the lifecycle of a component based web applications is presented and all benefits are exposed (easy maintenance, easy extensibility, easy adaptability in new systems).

Περιεχόμενα

1. Περιγραφή Εργασίας.....	19
1.1 Εισαγωγή.....	19
1.2 Σκοπός – Στόχοι.....	20
Κεφάλαιο 2.....	21
2. Έρευνα.....	21
2.1 Βιβλιογραφική Ανασκόπηση.....	21
2.1.1 Ορισμοί Συστατικών.....	21
2.1.2 Συνολικά Συμπεράσματα.....	22
2.2 Διερεύνηση στο χώρο του Διαδικτύου.....	23
2.3 Επαναχρησιμοποίηση.....	24
2.4 Μοντέλο Συστατικών (κατά Bachmann).....	25
2.5 Προβλήματα Παραδοσιακών Αρχιτεκτονικών.....	27
2.6 Λύσεις.....	28
Κεφάλαιο 3.....	29
3. Μοντέλο Συστατικών.....	29
3.1 Εισαγωγή – Ορισμός.....	29
3.2 Πλατφόρμα Συστατικών.....	30
3.2.1 Εισαγωγή.....	30

3.2.2	Λειτουργικές Απαιτήσεις Πλατφόρμας.....	30
3.2.3	Χαρακτηριστικά Πλατφόρμας.....	31
3.2.4	Υλοποίηση.....	32
3.2.5	Περιορισμοί.....	32
3.2.6	Τεχνικές Λεπτομέρειες.....	33
3.3	Συστατικά.....	34
3.3.1	Περιγραφή.....	34
3.3.2	Χαρακτηριστικά Συστατικών.....	34
3.3.3	Δομή Αρχείων Συστατικών.....	35
3.3.4	Περιορισμοί.....	36
3.4	Σενάριο Εκτέλεσης.....	37
3.4.1	Περιγραφή - Προεργασία.....	37
3.4.2	Βήμα 1 – Υλοποίηση Συστατικού.....	37
3.4.3	Βήμα 2 – Αρχικοποίηση Συστατικού.....	40
3.4.4	Βήμα 3 – Έλεγχος Συστατικού.....	41
3.5	Επαναχρησιμοποίηση.....	42
3.5.1	Τεχνικές Υιοθέτησης Συστατικών σε Απομακρυσμένους Εξυπηρετές.....	42
3.5.1.1	Τεχνική Proxy class.....	43
3.5.1.2	Τεχνική iframes.....	43
3.5.2	Επέκταση της Πλατφόρμας για Υποστήριξη Συστατικών που Αρχικοποιούνται σε Απομακρυσμένους Εξυπηρετές.....	44
3.6	Συμπεράσματα.....	45
	Κεφάλαιο 4.....	47
4	Ένα λειτουργικό Παράδειγμα.....	47
4.1	Εισαγωγή.....	47
4.2	Απαιτήσεις.....	47
4.3	Λειτουργίες.....	47

4.4Σχεδίαση Συστατικών.....	48
4.4.1Συστατικό Αναζήτησης.....	48
4.4.2Συστατικό Καλάθι Αγορών.....	50
4.4.3Επικοινωνία Συστατικών.....	51
4.4.4Γεγονότα.....	51
4.5Επεκτείνοντας το Σύστημα.....	52
4.5.1Ανάγκη Επέκτασης του Συστήματος.....	52
4.5.2Συστατικό Προβολής Λεπτομερών Προϊόντος	53
4.5.3Εισαγωγή Συστατικού στο Σύστημα.....	53
4.5.4Συστατικό Προβολής Νέων Προϊόντων.....	55
4.5.5Επεκτείνοντας Ένα Υπάρχον Συστατικό.....	56
4.6Υιοθέτηση Συστατικών σε Νέα Συστήματα.....	57
4.6.1Εφαρμογές που Στεγάζονται στον Ίδιο Εξυπηρέτη.....	57
4.6.2Σενάριο νέας σελίδας.....	57
4.6.3Τροποποίηση του Συστατικού Αναζήτησης	59
4.6.4Εισαγωγή του Συστατικού Επεξεργασίας Εισαγωγής.....	59
4.6.5Εφαρμογές που Στεγάζονται σε Απομακρυσμένους Εξυπηρέτες (Τρίτων).....	60
4.7Συμπεράσματα.....	62
Κεφάλαιο 5.....	63
5. Συμπεράσματα.....	63
5.1Διαφορές από Παραδοσιακές Διαδικτυακές Εφαρμογές.....	63
5.2Μελλοντική Δουλειά.....	63
Βιβλιογραφία.....	66

Κατάλογος Εικόνων

Εικόνα 1 Πρότυπο Σχεδίασης Υψηλού Επιπέδου (πηγή: [10]).....	26
Εικόνα 2 Γραφική Αναπαράσταση Μοντέλου Συστατικού.....	29
Εικόνα 3 Αντιστοίχιση λειτουργικού συστήματος και εφαρμογής με την πλατφόρμα συστατικών και το συστατικό.....	30
Εικόνα 4 demoComponent μέθοδος init.....	41
Εικόνα 5 demoComponent μέθοδος hello.....	42
Εικόνα 6 Επικοινωνία Συστατικών.....	51
Εικόνα 7 Έναρξη Σελίδας με το Συστατικό Προβολής Λεπτομερειών Προϊόντος.....	54
Εικόνα 8 Το σύστημα αφού έχει εισαχθεί το συστατικό προβολής νέων προϊόντων .	55
Εικόνα 9 Προετοιμασία συστατικού επεξεργασίας προϊόντων.....	58
Εικόνα 10 Φόρμα επεξεργασίας προϊόντος	58
Εικόνα 11 Ενεργοποίηση της μεταβλητής embed για το συστατικό προβολής νέων προϊόντων.....	61
Εικόνα 12 Συστατικό προβολής νέων προϊόντων σε εφαρμογές τρίτων	62

Κατάλογος Πινάκων

Πίνακας 1 Σύγκριση στοιχείων προτύπου σχεδίασης με το μοντέλο συστατικών.....	45
Πίνακας 2 Περιγραφή Μεθόδων του Συστατικού Αναζήτησης.....	49
Πίνακας 3 Περιγραφή Μεθόδων του Συστατικού Καλάθι Αγορών.....	50
Πίνακας 4 Αντιστοίχιση Μεθόδων και Γεγονότων.....	52
Πίνακας 5 Νέοι μέθοδοι του συστατικού «καλάθι αγορών».....	56
Πίνακας 6 Μέθοδοι του συστατικού επεξεργασίας.....	60

Κεφάλαιο 1

1. Περιγραφή Εργασίας

1.1 Εισαγωγή

Το θέμα της σύνθεσης λογισμικού βασισμένο σε συστατικά είναι στο προσκήνιο για τουλάχιστον τριάντα χρόνια και αποτελεί ανεξάρτητο κλάδο της επιστήμης των υπολογιστών (http://en.wikipedia.org/wiki/Component-based_software_engineering). Η ιδέα για ένα λογισμικό αποτελούμενο από συστατικά δημοσιοποιήθηκε για πρώτη φορά από τον Douglas McIlroy σε συνέδριο του NATO που έγινε το 1968 με θέμα «Μαζική παραγωγή συστατικών λογισμικού» [6]. Εκ τότε το πεδίο στον χώρο του λογισμικού άλλαξε δραματικά και ως συνέπεια άλλαξε και η συζήτηση γύρω από τα συστατικά λογισμικού.

Ήδη από τα μέσα της δεκαετίας του 90 το διαδίκτυο άρχισε να εμφανίζεται ως ένα μέσο φιλοξενίας εφαρμογών. Η πάροδος των χρόνων απέδειξε ότι το μοντέλο πελάτη εξυπηρέτη είναι ιδανικό για μια μεγάλη πλειάδα εφαρμογών καθώς παρέχει αρκετά πλεονεκτήματα (όπως κεντρικοποιημένη διαχείριση του λογισμικού). Στα πλαίσια αυτής της εξέλιξης, οι μεγάλες εταιρίες λογισμικού (πχ Google, Adobe) μεταφέρουν και διαθέτουν προς χρήση μέσω του διαδικτύου, παραδοσιακά προϊόντα λογισμικού πελάτη (πχ Google Docs¹, Adobe Photoshop Express²) αποδεικνύοντας έτσι, την δεδομένη πρόθεση των εταιριών για μεταφορά της λειτουργικότητας στην πλευρά του εξυπηρέτη.

Εξάλλου, αρχιτεκτονικές βασισμένες σε συστατικά παρουσιάζουν πλεονεκτήματα έναντι σε παραδοσιακούς τρόπους σχεδίασης όπως:

- Επαναχρησιμοποίηση λογισμικού
- Ξεκάθαρη διαίρεση του λογισμικού και καταμερισμός εργασιών

Ως εκ τούτου, κρίνεται αναγκαία η μελέτη της σύνθεσης λογισμικού που βασίζεται σε συστατικά για εφαρμογές διαδικτύου.

Η παρούσα εργασία χωρίζεται σε πέντε κεφάλαια. Αρχικά, γίνεται μια εισαγωγή στο θέμα, με μια γρήγορη ιστορική αναδρομή και μια αναφορά στα πλεονεκτήματα των αρχιτεκτονικών βασισμένων σε συστατικά. Επίσης, αναφέρονται ο σκοπός και οι στόχοι της εργασίας. Στο δεύτερο κεφάλαιο, παρουσιάζεται μια έρευνα γύρω από τις

¹ <http://docs.google.com/> online εργασία γραφείου, όπως επεξεργαστής κειμένου.

² <https://www.photoshop.com/express/landing.html> online επεξεργαστή φωτογραφιών.

αρχιτεκτονικές βασισμένες σε συστατικά τόσο σε επίπεδο βιβλιογραφίας όπως και σε εμπειρικό επίπεδο στο διαδίκτυο. Παράλληλα, γίνεται αναφορά σε θέματα επαναχρησιμοποίησης λογισμικού και τέλος παρουσιάζονται προβλήματα παραδοσιακών αρχιτεκτονικών ανάπτυξης διαδικτυακών εφαρμογών τα οποία επιλύονται με την υιοθέτηση αρχιτεκτονικών βασισμένων σε συστατικά. Στο τρίτο κεφάλαιο, ορίζεται και παρουσιάζεται ένα μοντέλο συστατικών για την ανάπτυξη εφαρμογών διαδικτύου. Στο τέταρτο κεφάλαιο, αναπτύσσεται μια εφαρμογή βασισμένη σε συστατικά, η οποία στην συνέχεια εμπλουτίζεται με περισσότερες λειτουργίες και συστατικά. Στο πέμπτο κεφάλαιο, παρουσιάζονται προτάσεις για μελλοντική εργασία επάνω στο θέμα, παράλληλα με τα τελικά συμπεράσματα που προκύπτουν από την εργασία.

1.2 Σκοπός - Στόχοι

Έχοντας ως συνισταμένες την μεταφορά της λειτουργικότητας του λογισμικού στο διαδίκτυο και την διαρκή ανάγκη για λογισμικό βασισμένο σε συστατικά διερευνώνται οι δυνατότητες ανάπτυξης δικτυακών εφαρμογών βασισμένες σε συστατικά.

Σκοπός της παρούσας εργασίας είναι να αποδείξει και να επιδείξει την δυνατότητα ανάπτυξης διαδικτυακών εφαρμογών βασισμένες σε συστατικά.

Οι στόχοι της εργασίας είναι η όσο δυνατόν ρεαλιστικότερη προσέγγιση του θέματος της ανάπτυξης διαδικτυακών εφαρμογών βασισμένες σε συστατικά έτσι ώστε να προκύψει ένα λειτουργικό αποτέλεσμα που θα δίνει την δυνατότητα ανάπτυξης πραγματικών εφαρμογών διαδικτύου βασισμένες σε συστατικά. Επίσης, στόχο αποτελεί ο ορισμός ενός μοντέλου συστατικών και η ανάπτυξη μιας λειτουργικής πλατφόρμας για την ανάπτυξη εφαρμογών βασισμένες σε συστατικά. Τέλος, στόχο αποτελεί η ανάπτυξη μιας εφαρμογής που θα χρησιμοποιεί την πλατφόρμα και θα είναι αποτέλεσμα σύνθεσης συστατικών. Η εφαρμογή θα αποτελέσει παράδειγμα ανάπτυξης λογισμικού με συστατικά.

Κεφάλαιο 2

2. Έρευνα

2.1 Βιβλιογραφική Ανασκόπηση

Στην βιβλιογραφία υπάρχει πλειάδα ορισμών για τον όρο: «συστατικό λογισμικού». Επίσης, υπάρχουν διαφορετικές οπτικές από τις οποίες αντιμετωπίζεται το συστατικό λογισμικού. Ορισμένοι επιστήμονες θεωρούν ως συστατικά στοιχεία υλοποίησης λογισμικού, ενώ άλλοι θεωρούν οποιοδήποτε τέχνημα (artifact). Δεν υπάρχει έλλειμμα ορισμών στην βιομηχανία λογισμικού για τον όρο συστατικό [10].

Αξίζει να παρατεθούν τέσσερις ορισμοί έτσι ώστε να γίνει ξεκάθαρό ότι υπάρχουν ομάδες επιστημόνων με διαφορετικό όραμα όσον αφορά τον ορισμό της έννοιας του συστατικού λογισμικού. Ταυτόχρονα, στους ορισμούς παρατηρείται και σύγκληση απόψεων σε ορισμένα σημεία. Επίσης, μέσα από τους ορισμούς θα επισημανθούν και τα κοινά σημεία που υπάρχουν.

2.1.1 Ορισμοί Συστατικών

Ο **Szyperski**, ορίζει το συστατικό λογισμικού ως εξής: «μια μονάδα σύνθεσης με συμβατικά ορισμένες διεπαφές και σαφή εξάρτηση από το περιβάλλον. Ένα συστατικό λογισμικού μπορεί να εγκατασταθεί ανεξάρτητα και να γίνει τμήμα σύνθεσης από τρίτους.» [5].

Στον παραπάνω ορισμό το συστατικό παρουσιάζεται ως ένα στοιχείο ανεξάρτητο και επαναχρησιμοποιήσιμο που αλληλεπιδρά με άλλα συστατικά και με το περιβάλλον του.

Ο **Bachmann** ορίζει το συστατικό ως εξής: «μια αδιαφανής υλοποίηση λειτουργικότητας, η οποία μπορεί να γίνει τμήμα σύνθεσης τρίτων και είναι συμμορφωμένη με ένα μοντέλο συστατικών.» [10].

Στον ορισμό του Bachmann, το συστατικό λογισμικού παρουσιάζεται ως στοιχείο υλοποίησης που αποτελεί τμήμα μιας σύνθεσης, η οποία είναι συμμορφωμένη με ένα μοντέλο συστατικών. Συνεπώς το συστατικό είναι στοιχείο

ενός γενικότερου μοντέλου συστατικών και κάθε υλοποίηση βασισμένη σε συστατικά πρέπει να είναι συμμορφωμένη με ένα μοντέλο.

Κατά τον **Brown**, « το συστατικό ορίζεται σαν ένα σύνολο από ανεξάρτητες επαναχρησιμοποιήσιμες υπηρεσίες. Η ανεξαρτησία είναι ένα χαρακτηριστικό που σημαίνει ότι δίνεται η δυνατότητα χρήσης σε διαφορετικά περιβάλλοντα χωρίς να υπάρχει εξάρτηση από οποιουδήποτε είδους άλλο συστατικό. Ο όρος επαναχρησιμοποιήσιμες υπηρεσίες υποδηλώνει ότι ένα συστατικό εφοδιάζεται με λειτουργικότητες στις οποίες άλλα συστατικά απαιτούν πρόσβαση.» [9].

Ο ορισμός του Brown, παρουσιάζει ομοιότητες με αυτόν του Szyperski, στο θέμα της ύπαρξης του συστατικού σε ένα περιβάλλον. Ο Brown, θεωρεί ότι το επίπεδο επαναχρησιμοποίησης του συστατικού βρίσκεται στις υπηρεσίες που αυτό διαθέτει.

Κατά τον **Sametinger** το συστατικό ορίζεται ως εξής: «επαναχρησιμοποιήσιμα συστατικά λογισμικού είναι αυτό-ελεγχόμενα (*self-contained*), ξεκάθαρα αναγνωρίσιμα τεχνουργήματα (*artifact*) που περιγράφουν η/και εκτελούν συγκεκριμένες λειτουργίες και έχουν ξεκάθαρη διεπαφή, κατάλληλη τεκμηρίωση και ορίζουν μια κατάσταση επαναχρησιμοποίησης.» [8].

Ο Sametinger, στον ορισμό του διευρύνει την έννοια του συστατικού και σε αυτήν περιλαμβάνει τεχνήματα (*artifacts*) που παράγονται κατά την διαδικασία ανάπτυξης του λογισμικού και μπορεί να περιλαμβάνουν: πηγαίο κώδικα, κείμενα (τεκμηρίωσης) ή εκτελέσιμα αρχεία [1].

2.1.2 Συνολικά Συμπεράσματα

Το πρώτο συμπέρασμα που μπορεί να εξαχτεί από τους παραπάνω ορισμούς είναι ότι τα συστατικά μπορούν να υπάρχουν σε δυο μορφές: σαν στοιχεία (υλοποίησης) λογισμικού ή σαν οποιοδήποτε τέχνημα που παράγεται κατά την διαδικασία της ανάπτυξης του λογισμικού.

Στα πλαίσια της παρούσας εργασίας συστατικά θεωρούνται μόνο τα στοιχεία λογισμικού.

Στους ορισμούς που παρατέθηκαν, παρατηρούνται ομοιότητες σε τρία σημεία:

- ανεξάρτητη ύπαρξη του συστατικού (από την εφαρμογή στην οποία ανήκει)

- δυνατότητα επαναχρησιμοποίησης του συστατικού
- εξάρτηση του συστατικού από το περιβάλλον που αυτό μπορεί να εγκατασταθεί και να λειτουργήσει

2.2 Διερεύνηση στο χώρο του Διαδικτύου

Διερευνώντας πακέτα λογισμικού και πλατφόρμες που υπάρχουν στο διαδίκτυο διαπιστώνεται ότι ο όρος «συστατικό» χρησιμοποιείται ευρέως και με διαφορετικές έννοιες πολλές φορές. Αυτό οφείλεται, στο ότι είναι ένας όρος με γενική σημασία, αρκετά κατανοητός (σε μια αφηρημένη του μορφή) και είναι εμπορικά ελκυστικός. Χαρακτηριστικά, αναφέρονται τα παρακάτω πακέτα λογισμικού και πλατφόρμες μέσω των οποίων οι προγραμματιστές παράγουν (κάποιας συγκεκριμένης και διαφορετικής μορφής) συστατικά λογισμικού:

- Java Beans™
- EJB™ – Enterprise Java Beans
- JSF – Java Server Faces
- CakePHP Framework

Τα παραπάνω αποδεικνύουν ότι είναι έντονη η ανάγκη για μια ενδεδειγμένη καταγραφή της τρέχουσας κατάστασης, τόσο σε ερευνητικό επίπεδο (βιβλιογραφικό) όσο και σε επίπεδο εξελίξεων στον εμπορικό τομέα (στο διαδίκτυο).

Η αντιμετώπιση των συστατικών στο διαδίκτυο μπορεί να γίνει από την οπτική του χρήστη έτσι ώστε να αποφευχθούν τυχόν εγκλωβισμοί σε αυστηρούς ορισμούς και να υπάρξει ευελιξία στην αναγνώριση συστατικών.

Η έννοια του «χρήστη» έχει διττή υπόσταση, γιατί αναφέρεται ταυτόχρονα σε τελικούς χρήστες αλλά και σε χρήστες συστατικών (προγραμματιστές) που έχουν ως σκοπό να τα προσαρμόσουν στην εφαρμογή/σελίδα τους.

Έτσι, ως συστατικό λογισμικού σε εφαρμογές διαδικτύου ορίζεται οποιοδήποτε στοιχείο λογισμικού της εφαρμογής μπορεί να λειτουργήσει αυτόνομα (μη εξάρτηση από άλλα συστατικά ή λειτουργίες της εφαρμογής) και να προσαρμοστεί σε μια τρίτη εφαρμογή που θέλει να το ενσωματώσει (επαναχρησιμοποίηση). Ο ορισμός υιοθετεί και τα τρία σημεία στα οποία παρουσίαζαν ομοιότητες οι ορισμοί που παρουσιάστηκαν στην βιβλιογραφική ανασκόπηση (παράγραφος 2.1). Το στοιχείο της εξάρτησης του συστατικού από το περιβάλλον ισχύει και αυτό, εφόσον τα συστατικά εκτελούνται στο περιβάλλον ενός φυλλομετρητή ιστού (web browser).

Εμπειρικά διαπιστώνεται ότι ακόμα και σε επίπεδο χρήστη, οι εφαρμογές στο διαδίκτυο έχουν μια φυσική τάση αποσύνθεσης σε συστατικά (ορισμένα εξ αυτών επαναχρησιμοποιήσιμα). Η δημιουργία συστατικών με δυνατότητα επαναχρησιμοποίησης (λειτουργιών ή κομματιών λειτουργικότητας) αποτελεί κίνητρο για τους προγραμματιστές γιατί με αυτόν τον τρόπο είναι εύκολο να εξαπλωθεί μια εφαρμογή. Οι εφαρμογές ευρείας αποδοχής, έχουν ανάγκη την αναδιανομής των λειτουργιών τους από άλλους ιστοτόπους έτσι ώστε να διαδίδονται πιο εύκολα. Ένα εξαιρετικό (βάση της επιτυχίας του) παράδειγμα που αναδείχτηκε τα τελευταία χρόνια είναι το αυτό του YouTube³, ενός ιστότοπος που προσφέρει την υπηρεσία διαμοιρασμού βίντεο. Η καινοτομία που εισήγαγε το Youtube και του έδωσε την ώθηση για να επικρατήσει στον χώρο, είναι η ευκολία διαμοιρασμού του κάθε βίντεο. Αν και στην οπτική του χρήστη γίνεται διαμοιρασμός των βίντεο στην πραγματικότητα γίνεται διαμοιρασμός του προγράμματος αναπαραγωγής (Player) που αναπαράγει το βίντεο. Το πρόγραμμα αναπαραγωγής (Player) αποτελεί ένα επαναχρησιμοποιήσιμο συστατικό και μπορεί να ενσωματώνεται με ελάχιστο κόπο σε μια άλλη εφαρμογή ή σελίδα του διαδικτύου. Από το παραπάνω παράδειγμα αποδεικνύεται ότι ένα εύκολα επαναχρησιμοποιήσιμο συστατικό αποτελεί παράγοντα επιτυχίας για μια διαδικτυακή εφαρμογή.

Με την έλευση και εισαγωγή συστατικών στον χώρο του διαδικτύου είναι πλέον δυνατό να οριστεί μια οπτική συστατικόστρεφής μέσα από την οποία μπορούν να αξιολογηθούν οι ιστοσελίδες.

Όσον αφορά το τεχνικό κομμάτι, η έλευση πολλών νέων τεχνολογιών που εμφανίστηκαν κάτω από την ομπρέλα του Web 2.0⁴, έχουν αλλάξει σε κάποιο βαθμό τον τρόπο προγραμματισμού διαδικτυακών εφαρμογών όπως και τις συνθήκες αλληλεπίδρασης με τον χρήστη. Αν και δεν ορίζεται αυστηρά, υπάρχει μια εμφανής τάση ανάπτυξης συστατικών που ενθαρρύνεται έντονα από τις τεχνολογίες όσο και την φιλοσοφία γύρω από το Web 2.0. Το μεγαλύτερο κίνητρο είναι η επαναχρησιμοποίηση και ο διαμοιρασμός λειτουργιών από ιστότοπο σε ιστότοπο, κάτι το οποίο επιτυγχάνεται με την υιοθέτηση συστατικόστρεφών αρχιτεκτονικών.

2.3 Επαναχρησιμοποίηση

Στους ορισμούς των συστατικών λογισμικού γίνεται σαφής αναφορά στο θέμα της επαναχρησιμοποίησης. Απαραίτητη προϋπόθεση για ένα στοιχείο λογισμικού έτσι

³ <http://www.youtube.com>

⁴ <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html> - O'Reilly What is Web 2.0

http://en.wikipedia.org/wiki/Web_2 - wikipedia

ώστε να χαρακτηριστεί συστατικό λογισμικού είναι η δυνατότητα επαναχρησιμοποίησης του.

Η επαναχρησιμοποίηση ορίζεται στο επίπεδο των λειτουργιών του συστατικού. Αυτό πρακτικά σημαίνει ότι ένα συστατικό με τις συγκεκριμένες λειτουργίες του μπορεί να ενσωματωθεί σε τρίτες εφαρμογές (third party applications) και να παρέχει τις λειτουργίες που ήδη έχει στα υπόλοιπα συστατικά της εφαρμογής.

Ως εκ τούτου, έτοιμα συστατικά λογισμικού μπορούν να αποτελέσουν δομικά στοιχεία μιας νέας εφαρμογής, όχι όμως προσαρμόζοντας τις λειτουργίες τους στις ανάγκες της εφαρμογής αλλά ενσωματώνοντας (σε αυτήν) τις λειτουργίες που ήδη παρείχαν.

Για να δημιουργηθεί μια εφαρμογή από έτοιμα συστατικά, αυτά θα πρέπει να είναι παραμετροποιήσιμα σε τέτοιο βαθμό, ώστε να παρέχουν την λειτουργία τους προσαρμοσμένη στις ανάγκες της εκάστοτε εφαρμογής. Η παραμετροποίηση θα πρέπει να γίνεται από το επίπεδο της παρουσίασης (presentation tier) μέχρι αυτό της βάσης δεδομένων (database tier). Στον τομέα των εφαρμογών διαδικτύου, είναι τεχνικά εφικτή η δημιουργία τέτοιων συστατικών λογισμικού αλλά ταυτόχρονα είναι και μη ρεαλιστική για τους εξής λόγους:

1. Είναι αδύνατο να υπάρξει ένα συστατικό λογισμικού το οποίο θα έχει προβλέψει και θα καλύπτει τις ανάγκες που θα έχει η κάθε εφαρμογή η οποία θα το ενσωματώσει.
2. Η ενσωμάτωση τέτοιων συστατικών θα ήταν δεσμευτική για την τεχνολογία που η εφαρμογή θα έπρεπε να χρησιμοποιήσει (κυρίως στο επίπεδο της βάσης δεδομένων). Αυτό αποτελεί έναν ιδιαίτερα δεσμευτικό παράγοντα γιατί οι προδιαγραφές επιλογής τεχνολογίας για το επίπεδο της βάσης δεδομένων είναι πολλές φορές προκαθορισμένη.

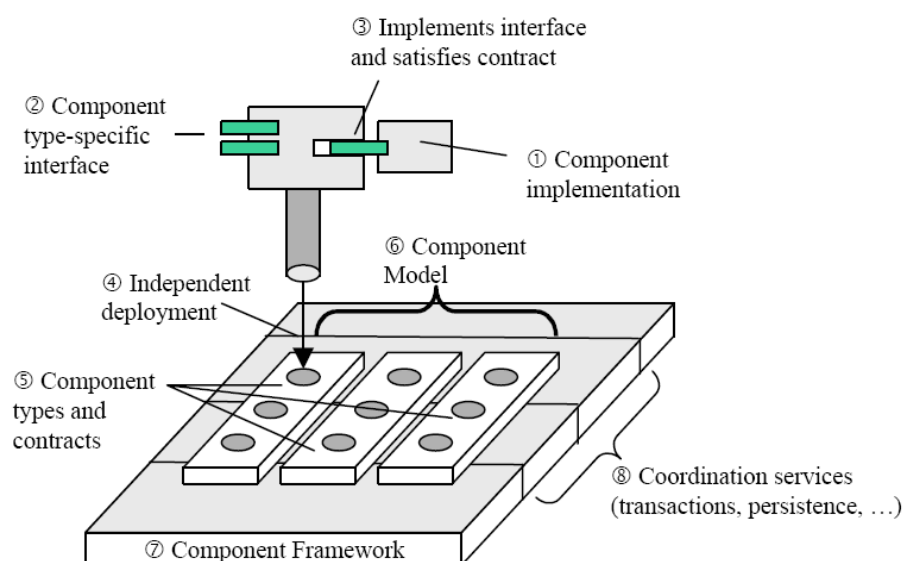
Στα πλαίσια της παρούσας εργασίας, επαναχρησιμοποίηση θα θεωρείται η ιδιότητα ενός συστατικού λογισμικού να προσαρμοστεί σε διαφορετικά περιβάλλοντα και να παρέχει τις λειτουργίες του σε άλλα συστατικά.

2.4 Μοντέλο Συστατικών (κατά Bachmann)

Συμφώνα με τον ορισμό του Bachmann, υφίσταται ένα μοντέλο συστατικών με το οποίο πρέπει να είναι συμμορφωμένα τα συστατικά.

Συστήματα βασισμένα σε συστατικά θεωρούνται αυτά που βασίζονται σε ένα πρότυπο σχεδίασης υψηλού επιπέδου (Εικόνα 1) [10]. Το συστατικό λογισμικού αποτελεί μέρος αυτού του προτύπου σχεδίασης. Το πρότυπο είναι αρκετά αφηρημένο

έτσι ώστε να επιτρέπει ευελιξία στις δυνατότητες ορισμού συστημάτων βασισμένα σε συστατικά.



Εικόνα 1 Πρότυπο Σχεδίασης Υψηλού Επιπέδου (πηγή: [10])

Το πρότυπο αποτελείται από οκτώ στοιχεία:

1. **Υλοποίηση Συστατικού (component implementation)** (όπως ορίστηκε στην παράγραφο 2.1)
2. **Διεπαφή (interface)**. Το κάθε συστατικό υλοποιεί μια ή περισσότερες διεπαφές που διαθέτει.
3. **Συμβόλαιο (contract)**. Με αυτόν τον όρο περιγράφονται οι υποχρεώσεις που ένα συστατικό έχει.
4. **Ανεξάρτητη εγκατάσταση (independent deployment)**. Πληρώνοντας τις προϋποθέσεις ένα συστατικό μπορεί να εγκατασταθεί σε ένα περιβάλλον.
5. **Τύποι συστατικών (component types)**. Ένα σύστημα βασισμένο σε συστατικά βασίζεται σε έναν μικρό αριθμό από διαφορετικούς τύπους συστατικών.
6. **Μοντέλο συστατικών (component model)**. Το περιβάλλον στο οποίο φιλοξενούνται οι τύποι συστατικών.
7. **Πλατφόρμα συστατικών (component framework)**. Η πλατφόρμα συστατικών παρέχει μια πληθώρα από υπηρεσίες,

8. **Υπηρεσίες (runtime services).** Η πλατφόρμα συστατικών παρέχει ένα πλήθος από υπηρεσίες στα συστατικά.

2.5 Προβλήματα Παραδοσιακών Αρχιτεκτονικών

Οι σύγχρονες διαδικτυακές εφαρμογές είναι διαρκώς εξελισσόμενοι ζωντανοί οργανισμοί. Αυτό, πρακτικά σημαίνει ότι κατά τον κύκλο ζωής τους προστίθενται αρκετά νέα χαρακτηριστικά και λειτουργίες. Με δεδομένο ότι αυτό γίνεται σε διάστημα χρόνων και ταυτόχρονα οι απαιτήσεις είναι τεράστιες, υπάρχει πολλές φορές το ενδεχόμενο οι προγραμματιστές να παρεκκλίνουν της αρχικής σχεδίασης είτε γιατί δεν υπήρχε η υποδομή για να υποστηριχθεί τέτοια λειτουργικότητα (πχ λόγω της εξέλιξης της τεχνολογίας) είτε γιατί η χρονική πίεση δεν επιτρέπει την σωστή σχεδίαση και ανάλυση των απαιτήσεων.

Επιπλέον, σε εφαρμογές μεγάλης κλίμακας υπάρχει η ανάγκη για εμφάνιση των υπηρεσιών και λειτουργιών τους σε πολλαπλά διαφορετικά σημεία της. Υπάρχουν περιπτώσεις που η αλλαγή, μετατροπή ή επέκταση σημείων της εφαρμογής που έχουν πολλαπλές εμφανίσεις μπορεί να προκαλέσει δυσλειτουργίες σε άλλα σημεία στα οποία επανεμφανίζονται και ενδεχόμενος ο προγραμματιστής δεν εντόπισε. Συνεπώς, σε περιπτώσεις επέκτασης της εφαρμογής αυξάνεται ο χρόνος που χρειάζεται ο προγραμματιστής για να φέρει εις πέρας την «αποστολή» του (task), αφού προστίθεται ο χρόνος διερεύνησης του κώδικα για επαναλαμβανόμενες εμφανίσεις της συγκεκριμένης λειτουργίας, όπως επίσης και ο χρόνος του ελέγχου (testing time). Ο συνολικός χρόνος που προκύπτει συνοψίζεται στον παρακάτω τύπο:

Συνολικός Χρόνος Αποστολής =

(αρχικός χρόνος υλοποίησης αποστολής) +

(χρόνος διερεύνησης επαναλήψεων) +

(τυχόν αλλαγές σε επανεμφανίσεις της λειτουργίας) +

(έλεγχος αρχικού αποστολής) +

(έλεγχος επανεμφανίσεων)

2.6Λύσεις

Η υιοθέτηση μια αρχιτεκτονικής βασισμένης σε συστατικά δίνει μια ξεκάθαρη προσέγγιση στο θέμα και λύνει τα προβλήματα που παρουσιάζονται σε εφαρμογές βασισμένες σε παραδοσιακές αρχιτεκτονικές. Σε μια τέτοια περίπτωση το σύστημα είναι ένα σύνολο από συστατικά που αλληλεπιδρούν. Τα συστατικά είναι ανεξάρτητα μεταξύ τους και συνεπή ως προς την διεπαφή που προσφέρουν. Τα συστατικά αλληλεπιδρούν μεταξύ τους στέλνοντας μηνύματα μέσω των διεπαφών τους. Σε ένα σύστημα βασισμένο σε συστατικά οι προγραμματιστές είναι επιφορτισμένοι με συγκεκριμένα εξ αυτών και δεν επιφορτίζονται με όλες τις λειτουργίες της εφαρμογής. Παράλληλα, η επέκταση ενός συστατικού επηρεάζει μονό τα συστατικά με τα οποία επικοινωνεί το τροποποιημένο συστατικό εάν και εφόσον έχει αλλάξει η διεπαφή του. Οπότε ο προγραμματισμός και οι αρμοδιότητες του κάθε προγραμματιστή είναι ξεκάθαρες και συνδεδεμένες με τα συστατικά τα οποία έχει επωμιστεί, δίνοντας έτσι ξεκάθαρο καταμερισμό ευθυνών και αρμοδιοτήτων.

Κεφάλαιο 3

3. Μοντέλο Συστατικών

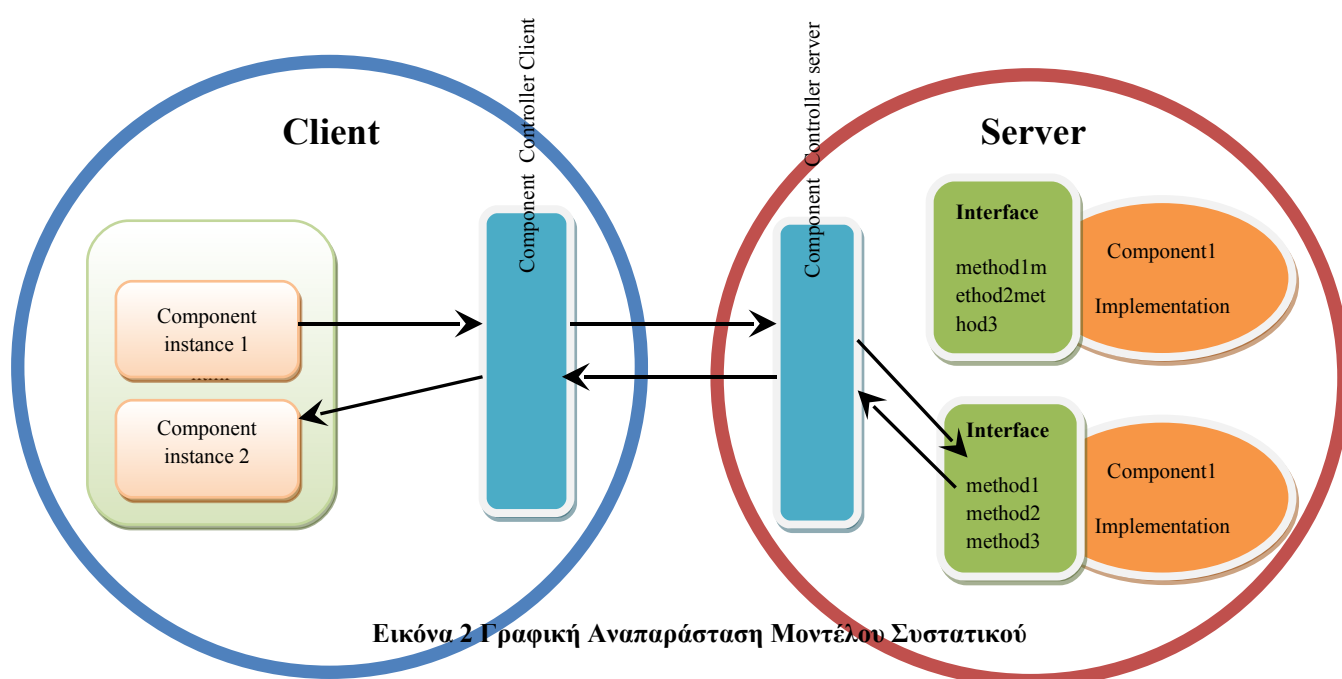
3.1 Εισαγωγή – Ορισμός

Με βάση το πρόβλημα που παρουσιάστηκε (παράγραφος 2.5) και την θεωρητική λύση που δόθηκε (παράγραφος 2.6), θα περιγράψει ένα μοντέλο συστατικών που είναι συνεπές με τις αρχές που τέθηκαν κατά την βιβλιογραφική ανασκόπηση. Το μοντέλο θα αποτελέσει βάση για την ανάπτυξη συστατικών.

Ως μοντέλο συστατικών θεωρείται ένα σύστημα το οποίο ακολουθεί το πρότυπο σχεδίασης υψηλού επιπέδου (παράγραφος 2.4).

Το μοντέλο συστατικών είναι συνεπές με το μοντέλο πελάτη-εξυπηρετή και βρίσκεται ένα επίπεδο επάνω από αυτό. Μπορεί να αποτελέσει βάση για την ανάπτυξη εφαρμογών διαδικτύου βασισμένες σε συστατικά.

Το μοντέλο αποτελείται από δυο στοιχεία, την πλατφόρμα συστατικών (component framework) και τα συστατικά. Σχηματικά αναπαρίσταται στην **Εικόνα 2**. Το μοντέλο επιτρέπει την επικοινωνία (αλληλεπίδραση) των συστατικών μεταξύ τους και με τον ίδιο τους τον εαυτό.



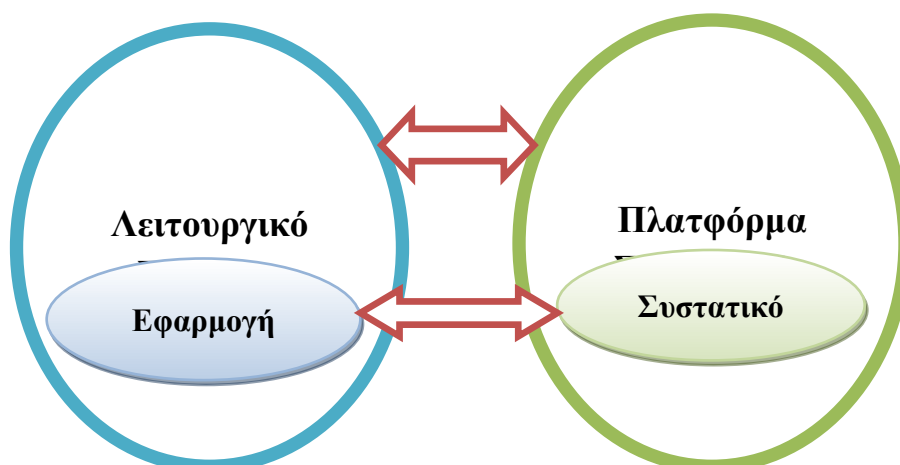
Εικόνα 2 Γραφική Αναπαράσταση Μοντέλου Συστατικού

3.2 Πλατφόρμα Συστατικών

3.2.1 Εισαγωγή

Σκοπός της πλατφόρμας είναι να παράσχει όλες τις υπηρεσίες που χρειάζονται έτσι ώστε να είναι εφικτή η λειτουργία συστατικών και κατ' επέκταση η λειτουργία εφαρμογών βασισμένες σε συστατικά. Επίσης η πλατφόρμα είναι υπεύθυνη για την επικοινωνία συστατικό με συστατικό.

Ένας παραλληλισμός που μπορεί να χρησιμοποιηθεί σαν παράδειγμα κατανόησης της πλατφόρμας συστατικών είναι αυτό της σχέσης μεταξύ λειτουργικού συστήματος και εφαρμογής σε αντιστοιχία με την πλατφόρμα συστατικών και τα συστατικά (Εικόνα 3). Έτσι, όπως ένα λειτουργικό σύστημα φιλοξενεί εφαρμογές και τους παρέχει υπηρεσίες (πχ ΙΟ) έτσι και η πλατφόρμα συστατικών φιλοξενεί συστατικά και τους παρέχει υπηρεσίες (πχ component to component communication).



Εικόνα 3 Αντιστοιχία λειτουργικού συστήματος και εφαρμογής με την πλατφόρμα συστατικών και το συστατικό

3.2.2 Λειτουργικές Απαιτήσεις Πλατφόρμας

Απαιτήσεις πλατφόρμας:

- Η πλατφόρμα πρέπει να λειτουργεί ανεξάρτητα από τα συστατικά (Οι προγραμματιστές των συστατικών θα αγνοούν τις λεπτομέρειες υλοποίησης της πλατφόρμας).

- Η πλατφόρμα πρέπει να ανακαλύπτει τα συστατικά (θα αποκρύπτεται από τον πελάτη η θέση του συστατικού)
- Η πλατφόρμα πρέπει να αποστέλλει την απόκριση στον πελάτη

3.2.3 Χαρακτηριστικά Πλατφόρμας

Η πλατφόρμα θα έχει τα παρακάτω χαρακτηριστικά:

- Πλευρά Εξυπηρέτη (Server – Side)
 - Δυνατότητα ανάγνωσης και δημιουργίας καθολικών ιδιοτήτων, διαθέσιμων σε όλα τα συστατικά (Definition of global properties)
 - Καταγραφή κλήσεων/γεγονότων σε αρχεία (Keep log files)
 - Υποστήριξη σε περιβάλλον με περισσότερους του ενός εξυπηρέτες που φιλοξενούν συστατικά (clustering)
 - Πακετάρισμα των αποκρίσεων στον πελάτη σε αρχεία XML (pure XML responses)
 - Δυναμική ή στατική ανακάλυψη των συστατικών. Η ανακάλυψη μπορεί να γίνει με έναν από τους παρακάτω τρόπους:
 - μέσο του ονόματος του συστατικού και του URL κλήσης
 - μέσο μιας αντιστοίχισης που ορίζεται σε αρχεία κειμένου (XML mappings)
 - μέσο αντιστοίχισης από βάση δεδομένων
 - Διαχείριση λανθασμένων κλήσεων
 - Εγκατάσταση και απεγκατάσταση συστατικών κατά την εκτέλεση (Runtime deployment/undeployment of components)
 - Υποστήριξη λειτουργίας αποσφαλμάτωσης (Debugging mode)
- Πλευρά Πελάτη (Client – Side)
 - Παροχή υπηρεσιών για την δημιουργία κλήσεων και χρήσης των συστατικών
 - Υπηρεσίες διαχείρισης των XML αποκρίσεων (XML response handling)

Σημείωση: το θέμα υποστήριξης συστάδων από εξυπηρέτες (clustering) δεν υλοποιήθηκε (καθώς ξεφεύγει από τους σκοπούς της παρούσας εργασίας) και ως εκ τούτου μπορεί να αποτελέσει αντικείμενο έρευνας μελλοντικών εργασιών.

3.2.4 Υλοποίηση

Στα πλαίσια της παρούσας εργασίας υλοποιήθηκε μια πλατφόρμα συστατικών που πληροί τις απαιτήσεις οι οποίες τέθηκαν στην παράγραφο 3.2.2 και διαθέτει τα χαρακτηριστικά που περιγράφηκαν στην παράγραφο 3.2.3.

Η πλατφόρμα συστατικών αποτελείται από δυο ελεγκτές συστατικών (Εικόνα 2). Έναν στην πλευρά του πελάτη και έναν στην πλευρά του εξυπηρέτη. Ο κάθε ελεγκτής υλοποιεί τα χαρακτηριστικά που αναφέρθηκαν στην παράγραφο 3.2.3.

Ο ελεγκτής στην πλευρά του πελάτη είναι υλοποιημένος σε γλώσσα σεναρίων πελάτη (client side scripting language), την Javascript που αποτελεί την πιο καθιερωμένη γλώσσα αυτού του είδους και υποστηρίζεται από όλους τους φυλλομετρητές ιστού.

Ο ελεγκτής στην πλευρά του εξυπηρέτη είναι υλοποιημένος σε γλώσσα σεναρίων εξυπηρέτη (server side scripting language), την PHP⁵. Η επιλογή της PHP έγινε με το κριτήριο της απλότητας που αυτή έχει στον τρόπο γραφής και κατανόησης της, επίσης αποτελεί μια ιδιαίτερα δημοφιλή λύση για την υλοποίηση εφαρμογών διαδικτύου. Πάραυτα, η γλώσσα που είναι υλοποιημένος ο ελεγκτής δεν είναι δεσμευτική και ως εκ τούτου μπορούν να υπάρξουν και εκδόσεις του ελεγκτή σε άλλες γλώσσες έτσι ώστε να υποστηριχτούν και άλλες τεχνολογίες (πχ Java).

3.2.5 Περιορισμοί

Η παραπάνω αρχιτεκτονική πλατφόρμας συστατικών παρουσιάζει έχει ορισμένους περιορισμούς:

- Τα συστατικά οφείλουν να είναι υλοποιημένα στην ίδια γλώσσα προγραμματισμού με την πλατφόρμα.
- Τα συστατικά οφείλουν να φιλοξενούνται στον ίδιο εξυπηρέτη με την πλατφόρμα.

⁵ <http://www.php.net>

- Τα συστατικά δεσμεύονται από την ομαλή λειτουργία της πλατφόρμας. Τυχών λανθασμένες καταστάσεις της πλατφόρμας θα έχουν ως αποτέλεσμα την προβληματική απόκριση/λειτουργία των συστατικών.
- Τα συστατικά οφείλουν να είναι συνεπή με τις απαιτήσεις τις πλατφόρμας και να τοποθετούνται στους ανάλογους φακέλους στον εξυπηρέτη.

3.2.6 Τεχνικές Λεπτομέρειες

Η πλατφόρμα συστατικών είναι βασισμένη αποκλειστικά σε κλήσεις AJAX⁶ (Asynchronous Javascript And XML) μεταξύ πελάτη και εξυπηρέτη και δεν υποστηρίζει παραδοσιακές υποβολές φορμών (form submit). Η επιλογή αυτή έγινε βάση της τάσης που καταγράφηκε κατά την ερεύνα και διαπιστώθηκε μια συγκεκριμένη στροφή προς πλούσιους (σε λειτουργικότητα) πελάτες (rich clients) σε αντίθεση με το παρελθόν όπου όλη η λειτουργικότητα ήταν ενσωματωμένη στην πλευρά του εξυπηρέτη. Η τάση αυτή καταγράφεται τόσο σε μεγάλες εμπορικές εφαρμογές όσο και στην έντονη δραστηριότητα που υπάρχει στην εμφάνιση πακέτων που υποστηρίζουν εφαρμογές βασισμένες σε κλήσεις AJAX. Χαρακτηριστικότερο παράδειγμα ανάλογης διαδικτυακής εφαρμογής ευρείας αποδοχής είναι αυτό του Yahoo MailTM. Συγκεκριμένα η εταιρία που παρέχει το πιο διαδεδομένο διαδικτυακό πρόγραμμα αλληλογραφίας (256,3 εκατομμύρια λογαριασμοί – 2^η δημοφιλέστερη διαδικτυακή εφαρμογή αλληλογραφίας⁷), η Yahoo, προέβη σε μια ριζική ανανέωση της εφαρμογής με σκοπό να παρέχει σε μορφή διαδικτυακής εφαρμογής ότι πιο κοντινό σε έναν παραδοσιακό πελάτη αλληλογραφίας (πχ Thunderbird, MS Outlook)⁸. Το αποτέλεσμα είναι μια διαδικτυακή εφαρμογή βασισμένη αποκλειστικά σε κλήσεις AJAX, στην οποία ο χρήστης δεν συναντά παραδοσιακές ανανεώσεις όλης της σελίδα, παρά μόνο επιμέρους σημείων της.

Το θέμα των πλούσιων (σε λειτουργικότητα) AJAX πελατών, αποτελεί ένα θέμα έντονης ερευνητικής δραστηριότητας και η περαιτέρω ανάλυση του ξεφεύγει από τα πλαίσια της παρούσας εργασίας.

Το συμπέρασμα που απορρέει από τα παραπάνω και σχετίζεται με τεχνολογίες βασισμένες σε συστατικά είναι ότι η χρήση σε κλήσεων AJAX ενθαρρύνει την σύνθεση και αποσύνθεση της εφαρμογής από συστατικά.

⁶ <http://en.wikipedia.org/wiki/AJAX>

⁷ <http://www.email-marketing-reports.com/metrics/email-statistics.htm> - Email and webmail statistics

⁸ <http://ajaxian.com/archives/yahoo-mail-ajax-edition-goes-live> Yahoo mail (AJAX based) goes live announcement

3.3 Συστατικά

3.3.1 Περιγραφή

Τα συστατικά αρχικοποιούνται στον πελάτη αλλά ο πυρήνας της υλοποίησης τους βρίσκεται στον εξυπηρέτη και έχει XML μορφή. Η κάθε σελίδα μπορεί να φιλοξενήσει έναν αυθαίρετο αριθμό συστατικών χωρίς να υπάρχει περιορισμός.

Τα συστατικά αποτελούν δομικούς λίθους της εφαρμογής και ταυτόχρονα αποτελούν και αυτόνομες οντότητες εγκατεστημένες στον εξυπηρέτη.

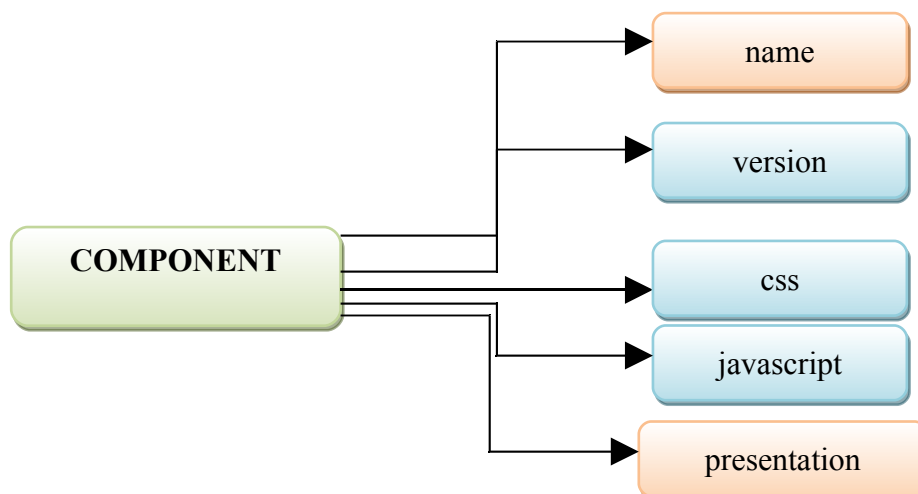
3.3.2 Χαρακτηριστικά Συστατικών

Τα συστατικά παρουσιάζουν τα εξής χαρακτηριστικά:

- Τα συστατικά αρχικοποιούνται στο πελάτη με απλό τρόπο (μέσο HTML ετικετών).
- Το κάθε συστατικό μπορεί να έχει ή όχι γραφική υπόσταση.
- Τα συστατικά παρέχουν μια διεπαφή με τις μεθόδους τους.
- Κάθε συστατικό έχει υποχρεωτικά μια μέθοδο αρχικοποίησης (init) και προαιρετικά μια μέθοδο καταστροφής (destroy).
- Δεν υπάρχει περιορισμός στον αριθμό των μεθόδων που ένα συστατικό μπορεί να έχει.
- Τα συστατικά επικοινωνούν μεταξύ τους στέλνοντας μηνύματα μέσω των μεθόδων τους.
- Υπάρχουν συστατικά που διατηρούν την κατάσταση τους (statefull) και συστατικά που δεν την διατηρούν (stateless). Τα μεν (statefull) λαμβάνουν υπ' όψη την κάθε κλήση και ανανεώνουν την κατάσταση τους βάση της κάθε κλήσης, ενώ τα δε (stateless) επιστρέφουν όμοια αποτελέσματα για κλήσεις με όμοια δεδομένα εισόδου.
- Το κάθε συστατικό επιστρέφει μια απόκριση XML στην οποία πακετάρεται η γραφική απεικόνιση του συστατικού μαζί με άλλες μετά-πληροφορίες που αφορούν το συστατικό (πχ έκδοση, όνομα, αναφορές σε αρχεία πελάτη που

συσχετίζεται). Η XML δομή παρέχει ευελιξία και μπορεί να επεκταθεί με περισσότερες ετικέτες έτσι ώστε να καλύψει τυχόν άλλες απαιτήσεις.

- Τα συστατικά έχουν ένα βασικό αρχείο υλοποίησης στο εξυπηρέτη το οποίο έχει την εξής XML μορφή:



```
<component>
  <name>myComponent</name>
  <version>1.15</version>
  <css>
    <file>file1.css</file>
  </css>
  <javascript>
    <file>file1.js</file>
  </javascript>
  <presentation>
    <![CDATA[ implementation script goes here! ]]>
  </presentation>
</component>
```

Η παραπάνω XML δομή χρησιμοποιείται και για τις αποκρίσεις στον πελάτη.

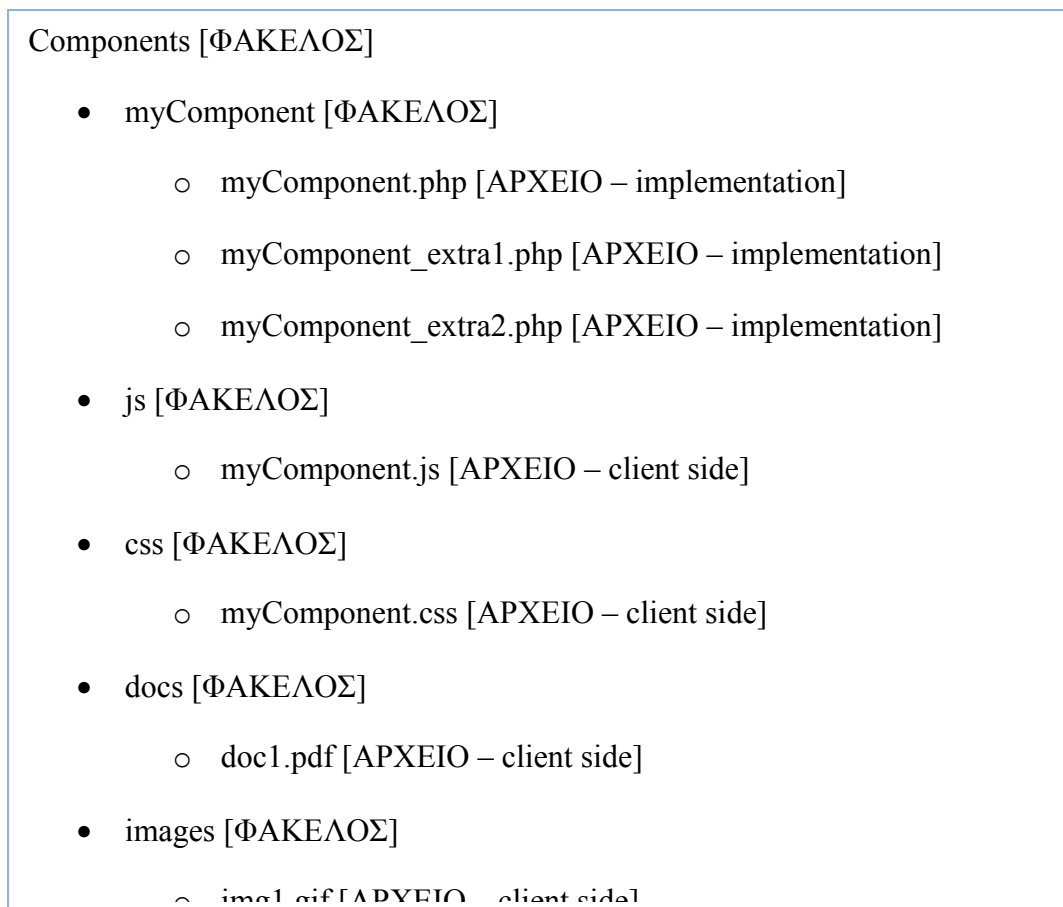
3.3.3 Δομή Αρχείων Συστατικών

Τα συστατικά αποτελούνται από δύο κομμάτια: το κομμάτι που χρησιμοποιεί ο πελάτης και το κομμάτι που χρησιμοποιεί εξυπηρέτης. Το κομμάτι του πελάτη περιλαμβάνει αρχεία σεναρίων πελάτη (πχ Javascript), αρχεία μορφοποίησης σελίδας

(πχ stylesheet), αρχεία εικόνων και άλλα αρχεία (πχ doc, pdf, zip κλπ). Το κομμάτι του εξυπηρέτη περιλαμβάνει ένα ή περισσότερα αρχεία σεναρίων εξυπηρέτη (πχ αρχεία PHP) στα οποία βρίσκεται η υλοποίηση του συστατικού. Αν και η πλατφόρμα παρέχει γενικές υπηρεσίες για ανάγνωση των (XML) αποκρίσεων υπάρχει η περίπτωση ένα συστατικό να τις παρακάμψει και να χρησιμοποιήσει δικές του συγκεκριμένες υπηρεσίες.

Όλα τα αρχεία του συστατικού πακετάρονται κάτω από τον ίδιο φάκελο, έτσι ώστε να είναι ανεξάρτητη η δομή και τα αρχεία του. Ο φάκελος έχει το όνομα του συστατικού.

Έτσι σύμφωνα με τα παραπάνω προκύπτει η παρακάτω δομή αρχείων για ένα συστατικό με όνομα myComponent:



3.3.4 Περιορισμοί

Η επικοινωνία των συστατικών γίνεται μόνο μεταξύ δυο εξ αυτών. Το μήνυμα στέλνεται από ένα συστατικό A σε ένα άλλο B (ή στον ίδιο του τον εαυτό A) και

αυτό θα επιστρέφει μια απόκριση (response) στο Β. Συνεπώς, η απόκριση του συστατικού που κλήθηκε θα είναι πάντα στον ίδιο του τον εαυτό, ανεξάρτητος από το ποιο συστατικό έστειλε το μήνυμα. Ως εκ τούτου η απόκριση μπορεί να επηρεάσει ένα και μόνο ένα συστατικό.

3.4 Σενάριο Εκτέλεσης

3.4.1 Περιγραφή - Προεργασία

Έχοντας εγκατεστημένη την πλατφόρμα συστατικών σε έναν εξυπηρέτη θα περιγραφεί η διαδικασία ανάπτυξης, εγκατάστασης και χρήσης ενός απλού συστατικού. Μέσα από το παράδειγμα θα επιδειχθεί ο βασικός κύκλος λειτουργίας του συστατικού μέσα στην εφαρμογή.

Το συστατικό είναι ιδιαίτερα απλό, έχει γραφικό περιβάλλον και κατά την αρχικοποίηση του επιστρέφει το αρχικό γραφικό περιβάλλον του, το οποίο περιέχει έναν μόνο υπερσύνδεσμο (hyperlink). Ο υπερσύνδεσμος κατά το πάτημα του θα ενεργοποιεί μια μέθοδο του συστατικού (την hello) η οποία θα επιστρέφει το ανάλογο γραφικό περιβάλλον. Στο γραφικό περιβάλλον της hello, θα υπάρχει ένας υπερσύνδεσμος κλήσης της init.

3.4.2 Βήμα 1 - Υλοποίηση Συστατικού

Όπως αναφέρθηκε παραπάνω το βασικό αρχείο υλοποίησης του συστατικού έχει συγκεκριμένη XML μορφή και η γλώσσα υλοποίησης είναι η PHP. Με βάση τα παραπάνω το αρχείο υλοποίησης του συστατικού είναι το εξής:


```

<component>
<name>demoComponent</name>
<version>1.00</version>
<presentation><![CDATA[
<h2> this is demoComponent! </h2>
<br/><br/>
<?php
if ( $method == 'init' ) {
echo "<a
href=\"javascript:sendMessageWithParameters('demoComponent','hello','GET',demoResponseHandler)\
">invoke hello method</a>";
} else if ( $method == 'hello' ) {
echo "<a
href=\"javascript:sendMessageWithParameters('demoComponent','init','GET',demoResponseHandler)\
">invoke init method</a>";
} ?>
]]>
</presentation>
</component>

```

Τα περιεχόμενα της ετικέτας *<presentation>* είναι αυτά που θα καταλήξουν στην οθόνη του πελάτη αφού εκτελέσει ο εξυπηρετής όλες τις εντολές σεναρίου PHP. Η προκαθορισμένη μεταβλητή “\$method” περιέχει ήδη το όνομα της μεθόδου που κλήθηκε ενώ αντίστοιχα η μεταβλητή “\$componentName” περιέχει το όνομα του συστατικού. Τις δυο αυτές μεταβλητές φροντίζει να ενημερώνει σε κάθε κλήση ο ελεγκτής εξυπηρετή (από την πλατφόρμα συστατικών).

Στον κώδικα παρατηρούνται το δυο υπερσύνδεσμοι μέσω των οποίων στέλνεται μήνυμα από το demoComponent σε μεθόδους του. Οι υπερσύνδεσμοι έχουν την παρακάτω μορφή:

```

<a
href="javascript:
sendMessageWithParameters('demoComponent','init','GET',demoResponseHandler)">
invoke init method
</a>

```

Μέσω του υπερσυνδέσμου καλείται μια από τις υπηρεσίες της πλατφόρμας (η **sendMessageWithParameters**).

Η `sendMessageWithParameters`, λειτουργεί με πέντε παραμέτρους (η πέμπτη είναι προαιρετική):

- Όνομα συστατικού στο οποίο θα σταλεί μήνυμα (πχ `demoComponent`)
- Όνομα μεθόδου του συστατικού που θα κληθεί (πχ `init`)
- HTTP μέθοδος κλήσης (πχ `GET` ή `POST`)
- Συνάρτηση (Javascript) που θα διαχειριστεί την απόκριση του εξυπηρέτη (πχ `demoResponseHandler`)
- Παράμετροι (δεν εμφανίζεται στο παραπάνω παράδειγμα καθώς το συστατικό δεν χρειάζεται να στείλει παραμέτρους μαζί με την κλήση)

Μέσω της `sendMessageWithParameters`, γίνεται η επικοινωνία και η αλληλεπίδραση των συστατικών. Οι κλήσεις στα συστατικά δεν είναι άμεσες αλλά έμμεσες. Καλείται η υπηρεσία `sendMessageWithParameters` της πλατφόρμας στην πλευρά του πελάτη. Στην συνέχεια γίνεται μια κλήση `AJAX` στην πλατφόρμα στην πλευρά του εξυπηρέτη. Αυτή με την σειρά της, εντοπίζει και ζητά από το κατάλληλο συστατικό να εκτελεστεί. Το αποτέλεσμα επιστρέφεται στον πελάτη.

3.4.3 Βήμα 2 – Αρχικοποίηση Συστατικού

Σε μια σελίδα (HTML ή PHP) που ενσωματώνει την πλατφόρμα συστατικών (συμπεριλαμβάνοντας τις κατάλληλες αναφορές σε Javascript αρχεία) η αρχικοποίηση του συστατικού γίνεται με τον ορισμό μιας ετικέτας (`div`). Το `div` εξ ορισμού, δεν έχει σημασιολογικό νόημα, αποκτά όμως ορίζοντας το να ανήκει στην κλάση “`components`”. Πλέον θεωρείται στιγμιότυπο ενός συστατικού. Ο ορισμός του συστατικού στην σελίδα είναι ο εξής:

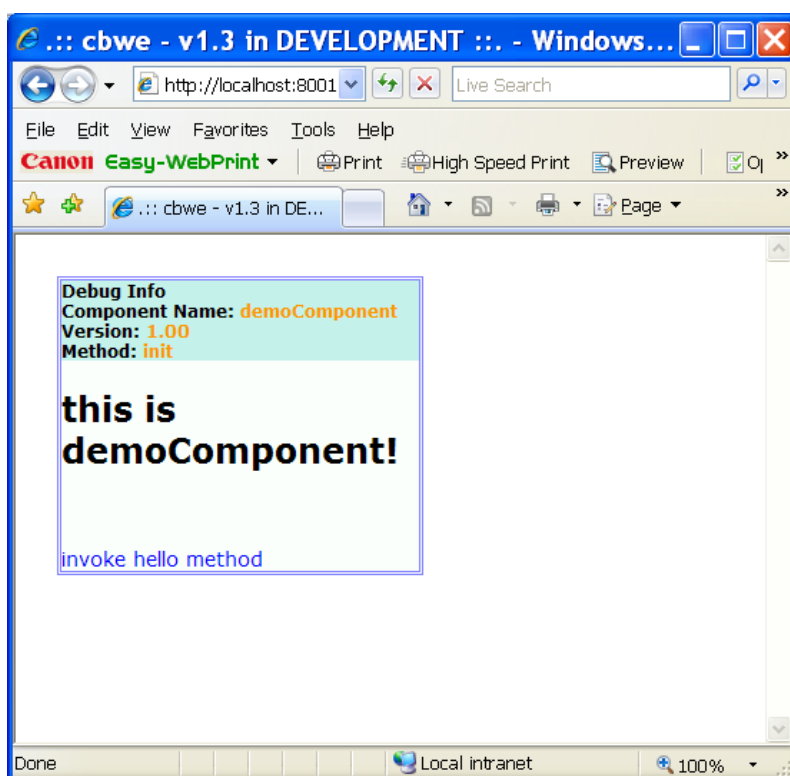
```
<div
  id="demoComponent"
  class="components"
  loadfunction="demoResponseHandler">
  Your browser does not support components!
</div>
```

Παρατήρηση: στο περιεχόμενο της ετικέτας `<div>` υπάρχει ένα κείμενο προειδοποίησης σε περίπτωση που ο φυλλομετρητής ιστού του χρήστη δεν υποστηρίζει κάποιες από τις τεχνολογίες που χρησιμοποιεί η πλατφόρμα συστατικών. Εάν, όλες οι

τεχνολογίες υποστηρίζονται σωστά και η εκτέλεση του συστατικού είναι επιτυχής τότε το κείμενο προειδοποίησης δεν θα είναι ορατό στον χρήστη.

3.4.4 Βήμα 3 – Έλεγχος Συστατικού

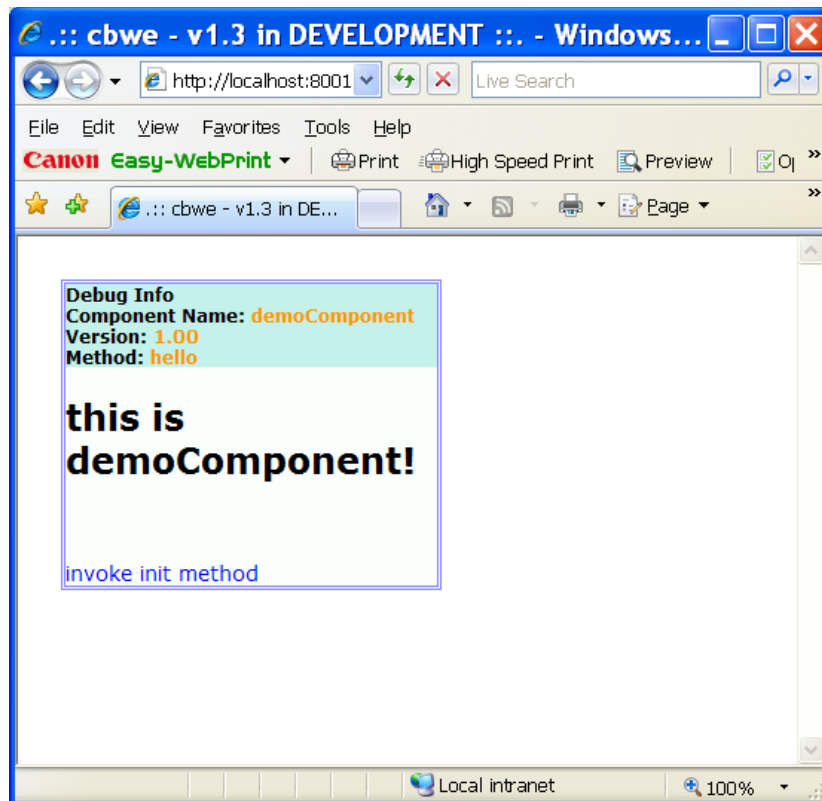
Σε έναν φυλλομετρητή ιστού, ζητείται η σελίδα στην οποία έχει οριστεί το στιγμιότυπο του συστατικού. Το αποτέλεσμα φαίνεται στην Εικόνα 3/Εικόνα 4.



Εικόνα 4 demoComponent μέθοδος init

Σημειώνεται ότι η πλατφόρμα συστατικών έχει ενεργοποιημένη την λειτουργία αποσφαλμάτωσης και έτσι εμφανίζονται στο πάνω μέρος του συστατικού πληροφορίες γι αυτό (όνομα συστατικού, έκδοση, μέθοδος που κλήθηκε). Επίσης γύρο από το συστατικό υπάρχει ένα πλαίσιο έτσι ώστε να είναι ξεκάθαρα τα όρια του. Το πλαίσιο αφαιρείται σε συνθήκες πραγματικής λειτουργίας.

Μετά το πάτημα του υπερσυνδέσμου από τον χρήστη, ένα μήνυμα στέλνεται στο συστατικό, αυτό εκτελεί την μέθοδο hello και επιστρέφει την ανάλογη γραφική αναπαράσταση (Εικόνα 5).



Εικόνα 5 demoComponent μέθοδος hello

3.5 Επαναχρησιμοποίηση

3.5.1 Τεχνικές Υιοθέτησης Συστατικών σε Απομακρυσμένους Εξυπηρέτες

Βασικό ζητούμενο σε αρχιτεκτονικές βασισμένες σε συστατικά είναι η δυνατότητα επαναχρησιμοποίησης των συστατικών και ενσωμάτωσής τους σε εφαρμογές τρίτων (third party application).

Τα συστατικά δεν γνωρίζουν σε ποια εφαρμογή φιλοξενούνται, μοναδική απαίτηση είναι να υπάρχει η πλατφόρμα συστατικών εγκατεστημένη. Ως εκ τούτου, η υιοθέτηση ενός συστατικού σε μια τρίτη εφαρμογή που έχει εγκατεστημένη την πλατφόρμα συστατικών, μπορεί να γίνει ακριβώς με τον ίδιο τρόπο όπως και στην πρωτογενή εφαρμογή, ορίζοντας δηλαδή μια HTML ετικέτα (div) συγκεκριμένης κλάσης (class=components) με αναγνωριστικό το αναγνωριστικό του συστατικού (id=demoComponent).

Ο παραπάνω τρόπος αναμένεται να λειτουργήσει σε εφαρμογές που φιλοξενούνται στον ίδιο εξυπηρετή με αυτόν που είναι εγκατεστημένα τα συστατικά, όχι όμως για εφαρμογές/σελίδες που φιλοξενούνται σε άλλους εξυπηρετές. Ο λόγος είναι ένας περιορισμός ασφάλειας που υπάρχει σε όλους τους φυλλομετρητές ιστού όσον αφορά σε κλήσεις AJAX μεταξύ διαφορετικών εξυπηρετών (γνωστό στην βιβλιογραφία ως: cross domain AJAX calls security restriction ⁹).

Για να υπερκεραστεί ο περιορισμός ασφάλειας υπάρχουν δυο καθιερωμένες λύσεις που χρησιμοποιούνται στο διαδίκτυο και έχουν εφαρμογή στην προκειμένη περίπτωση:

- Χρήση <iframes> για την αρχικοποίηση των συστατικών
- Δημιουργία και χρήση μιας κλάσης proxy για τις κλήσεις στην πλατφόρμα

3.5.1.1 Τεχνική Proxy class

Μια τεχνική για να ξεπεραστεί ο περιορισμός ασφάλειας που περιγράφηκε είναι η δημιουργία μιας proxy κλάσης στον εξυπηρετή στον οποίο θα φιλοξενηθούν τα συστατικά. Η κλάση πρέπει να είναι σε μια γλώσσα που υποστηρίζει ο εξυπηρετής.

Αφού δημιουργηθεί η κλάση τότε οι κλήσεις του στιγμιότυπου του συστατικού θα γίνονται προς αυτήν, η οποία στην συνέχεια θα πραγματοποιεί την κλήση στον απομακρυσμένο εξυπηρετή.

3.5.1.2 Τεχνική iframes

Η iframe είναι μια HTML ετικέτα και ορίζεται ως ένα στοιχείο που δημιουργεί ένα πλαίσιο (frame) που περιέχει ένα άλλο κείμενο (The iframe element creates an inline frame that contains another document – ορισμός από w3schools¹⁰) Συνήθως μια iframe περιέχει μια άλλη HTML σελίδα. Οι iframes δεν είναι έγκυρη XHTML σύμφωνα με τις XHTML προδιαγραφές 1.0¹¹.

Η τεχνική χρησιμοποίησης iframes για να ξεπεραστεί ο περιορισμός ασφαλείας κλήσεων AJAX μεταξύ διαφορετικών εξυπηρετών έγινε γνωστή μέσα από την εφαρμογή Google Maps TM. Σε αυτό το παράδειγμα κάθε σελίδα/εφαρμογή στο διαδίκτυο μπορεί να ενσωματώσει έναν χάρτη με δυνατότητες αλληλεπίδρασης (interactive map) ορίζοντας μια iframe. Όσο ο χρήστης κινείται στον χάρτη η

⁹ <http://getahead.org/dwr/ajax/cross-domain-xhr> άρθρο σχετικά με cross domain AJAX

http://snook.ca/archives/javascript/cross_domain_aj/ προτάσεις σχετικά με cross domain AJAX

¹⁰ http://www.w3schools.com/TAGS/tag_iframe.asp ορισμός iframe

¹¹ <http://www.w3.org/TR/xhtml1/>

πληροφορία φορτώνεται δυναμικά μέσω κλήσεων AJAX στην αρχική εφαρμογή της Google.

Συγκρίνοντας τις δυο παραπάνω τεχνικές προκύπτει ότι η τεχνική των iframes υπερτερεί των proxy κλάσεων σε δύο σημεία:

- Δεν χρειάζεται η υλοποίηση και εγκατάσταση νέου λογισμικού (όπως η proxy κλάση)
- Η ενσωμάτωση ενός συστατικού είναι ιδιαίτερα απλή, καθώς απαιτείται ο ορισμός μόνο μιας HTML ετικέτας.

3.5.2 Επέκτασης της Πλατφόρμας για Υποστήριξη Συστατικών που Αρχικοποιούνται σε Απομακρυσμένους Εξυπηρέτες

Η πλατφόρμα όπως περιγράφηκε δεν παρέχει υποστήριξη σε συστατικά που αρχικοποιούνται μέσω μιας ετικέτας iframe. Αν και η πλατφόρμα θα απαντήσει σε μια κλήση αρχικοποίησης ενός τέτοιου συστατικού, η απάντηση της θα είναι μορφής XML. Είναι στις αρμοδιότητες της πλατφόρμας που είναι εγκατεστημένη στον πελάτη να εξάγει την πληροφορία για το γραφικό περιβάλλον του συστατικού (που βρίσκεται μέσα στην ετικέτα <presentation>). Η ενσωμάτωση της πλατφόρμας πελάτη στην εφαρμογή του απομακρυσμένου εξυπηρέτη κάνει πιο πολύπλοκη την διαδικασία ενσωμάτωσης συστατικών και καταργεί τα πλεονεκτήματα που αρχικά διακρίθηκαν στην τεχνική των iframes. Συνεπώς, χρειάζεται να επεκταθούν οι λειτουργίες της πλατφόρμας συστατικών στον εξυπηρέτη έτσι ώστε να καλυφθούν τέτοιες περιπτώσεις.

Η επέκταση σημαίνει την εισαγωγή ενός νέου ελεγκτή συστατικών (εξωτερικού ελεγκτή – external controller) για τα συστατικά που βρίσκονται σε απομακρυσμένους εξυπηρέτες ο οποίος θα έχει όλες τις λειτουργίες του ελεγκτή που ήδη υπάρχει με την διαφορά ότι θα επιστρέφει το HTML κείμενο της γραφικής αναπαράστασης του συστατικού και όχι όλη την XML πληροφορία. Επίσης στο HTML κομμάτι του συστατικού θα προσαρτά τυχόν αναφορές σε άλλα αρχεία, όπως μορφοποίησης ή σεναρίων (πχ Javascript, CSS). Ο εξωτερικός ελεγκτής θα χρησιμοποιεί τον εσωτερικό για να καλέσει το συστατικό και ουσιαστικά θα συμπεριφέρεται σαν proxy ελεγκτής με αυξημένες λειτουργίες. Αν και η λύση που δίνεται φαίνεται να συγκλίνει στην λύση με την proxy κλάση, εντούτοις παρουσιάζει μια σημαντική διαφορά που αποτελεί και πλεονέκτημα: Με την εισαγωγή του εξωτερικού ελεγκτή, υπάρχει ένας μόνο proxy ελεγκτής ο οποίος αποκρύπτεται από την απομακρυσμένη εφαρμογή.

Συνεπώς η λύση μπορεί να θεωρηθεί υβριδική καθώς περιέχει τόσο στοιχεία από την τεχνική των iframes, όσο και από την τεχνική της proxy κλάσης.

Με βάση τα παραπάνω, αποδεικνύεται ότι συστατικά που αναπτύσσονται βασισμένα στο μοντέλο που περιγράφηκε είναι δυνατόν να επαναχρησιμοποιηθούν με απλό τρόπο από εφαρμογές που φιλοξενούνται σε οποιονδήποτε εξυπηρέτη.

3.6 Συμπεράσματα

Ένα σύστημα βασισμένο σε συστατικά θεωρείται αυτό που υιοθετεί μια αρχιτεκτονική βασισμένη σε συστατικά [10]. Ως εκ τούτου, οποιαδήποτε εφαρμογή αναπτυχθεί χρησιμοποιώντας το μοντέλο συστατικών που περιγράφηκε, μπορεί να θεωρηθεί εφαρμογή βασισμένη σε συστατικά.

Συγκρίνοντας το μοντέλο που ορίστηκε, με το πρότυπο σχεδίασης που παρουσιάστηκε στην βιβλιογραφική ανασκόπηση, παρατηρείται ότι το μοντέλο πληροί τις προϋποθέσεις του προτύπου σχεδίασης. Συγκεκριμένα, στο πρότυπο σχεδίασης υπάρχουν οκτώ στοιχεία τα οποία διακρίνονται και στο μοντέλο συστατικών που ορίστηκε:

Πίνακας 1 Σύγκριση στοιχείων προτύπου σχεδίασης με το μοντέλο συστατικών

Στοιχείο	Περιγραφή Bachmann [10]	Αντιστοίχιση με το μοντέλο συστατικών
Συστατικό	Το μοντέλο συστατικών αποτελείται από συστατικά .	Υποστηρίζεται από το μοντέλο (παράγραφος 3.1).
Διεπαφή	Το κάθε συστατικό υλοποιεί μια ή περισσότερες διεπαφές που διαθέτει.	Υποστηρίζεται από το μοντέλο (παράγραφος 3.3.2).
Συμβόλαιο	Με αυτόν τον όρο περιγράφονται οι υποχρεώσεις που ένα συστατικό έχει.	Στο μοντέλο συστατικών που ορίστηκε το κάθε συστατικό οφείλει να έχει μια XML μορφή με συγκεκριμένη δομή. Επίσης, πρέπει να παρέχει μια μέθοδο αρχικοποίησης (την init).

Ανεξάρτητη εγκατάσταση	Πληρώντας τις προϋποθέσεις ένα συστατικό μπορεί να εγκατασταθεί σε ένα περιβάλλον.	Υποστηρίζεται από το μοντέλο, αρκεί ένα συστατικό να έχει συγκεκριμένη δομή φακέλων και το βασικό αρχείο της υλοποίησής τους να είναι στην προκαθορισμένη XML μορφή. Η παρουσία των αρχείων στον εξυπηρέτη αρκεί για την εγκατάσταση του συστατικού.
Τύποι συστατικών	Ένα σύστημα βασισμένο σε συστατικά βασίζεται σε έναν μικρό αριθμό από διαφορετικούς τύπους συστατικών.	Υποστηρίζεται από το μοντέλο, η ύπαρξη διαφορετικών τύπων συστατικών που θα αποτελέσουν το σύστημα.
Μοντέλο συστατικών	Το περιβάλλον στο οποίο φιλοξενούνται οι τύποι συστατικών.	Ορίστηκε στο κεφάλαιο 3.
Πλατφόρμα συστατικών	Η πλατφόρμα συστατικών παρέχει μια πληθώρα από υπηρεσίες,	Ορίστηκε στην παράγραφο 3.2.
Υπηρεσίες (runtime services)	Η πλατφόρμα συστατικών παρέχει ένα πλήθος από υπηρεσίες στα συστατικά.	Όπως περιγράφονται στην παράγραφο 3.2.3.

Κεφάλαιο 4

4. Ένα λειτουργικό Παράδειγμα

4.1 Εισαγωγή

Στα πλαίσια της παρούσας εργασίας αναπτύχθηκε ένα δικτυακό ηλεκτρονικό κατάστημα ως παράδειγμα εφαρμογής βασισμένη σε συστατικά. Το ηλεκτρονικό κατάστημα αγορών υποστηρίζει τις πολύ βασικές λειτουργίες ενός καταστήματος (αναζήτηση προϊόντων, αγορές κλπ). Μέσα από το παράδειγμα αναδεικνύονται τα πλεονεκτήματα ανάπτυξης συστημάτων βασισμένα σε συστατικά και κυρίως τα πλεονεκτήματα επέκτασης τέτοιων συστημάτων.

4.2 Απαιτήσεις

Το σύστημα «ηλεκτρονικό κατάστημα» πρέπει να έχει ένα γραφικό περιβάλλον φιλικό προς τον χρήστη και να του δίνει την δυνατότητα, μέσα από τις λειτουργίες του, να περιηγηθεί μεταξύ των διαθέσιμων προϊόντων και να επιλέξει ποια από αυτά θα βάλει στο καλάθι αγορών έτσι ώστε μετέπειτα να αγοράσει. Ο χρήστης πρέπει να είναι ενήμερος για τις τιμές των προϊόντων καθώς επίσης και για το άθροισμα των τιμών των προϊόντων που έχει προσθέσει στο καλάθι.

Το σύστημα πρέπει να βασίζεται σε επαναχρησιμοποιήσιμα συστατικά.

4.3 Λειτουργίες

Το σύστημα θα υποστηρίζει τις παρακάτω λειτουργίες:

1. Ο χρήστης θα έχει την δυνατότητα αναζήτησης προϊόντος. Η αναζήτηση βασίζεται σε τέσσερα κριτήρια:
 - Κωδικός προϊόντος
 - Όνομα προϊόντος
 - Κατηγορία (Βιβλίο, Ηλεκτρονική συσκευή, Μουσική, Δίσκος DVD)
 - Τιμή (μεγαλύτερο/μικρότερο από κάποιο ποσό, ίσο με κάποιο ποσό)

2. Ο χρήστης θα έχει την δυνατότητα επιλογής προϊόντων από τα αποτελέσματα της αναζήτησης και αποστολής τους στο καλάθι αγορών
3. Ο χρήστης θα έχει την δυνατότητα αφαίρεσης ενός προϊόντος από το καλάθι αγορών
4. Ο χρήστης θα έχει την δυνατότητα αφαίρεσης όλων των προϊόντων με ένα κλικ από το καλάθι αγορών
5. Το καλάθι αγορών θα προβάλλει συνοπτικές πληροφορίες για το κάθε προϊόν που περιέχει (Όνομα, Κατηγορία, Τιμή)
6. Το καλάθι αγορών θα προβάλλει την συνολική τιμή των προϊόντων

4.4 Σχεδίαση Συστατικών

Είναι απαραίτητο να αναγνωριστούν τα συστατικά από τα οποία θα αποτελείται το σύστημα. Με βάση τις απαιτήσεις που υπάρχουν για το σύστημα αναγνωρίζονται δυο συστατικά στα οποία μοιράζονται οι λειτουργίες του συστήματος:

- Συστατικό αναζήτησης προϊόντων (Search Component) (λειτουργίες 1,2)
- Συστατικό καλάθι αγορών (Basket Component) (λειτουργίες 3,4,5,6)

4.4.1 Συστατικό Αναζήτησης

Δεδομένου ενός συνόλου δεδομένων θα ορίζεται ένα συστατικό αναζήτησης σε αυτά. Ως σύνολο δεδομένων θεωρούνται τα περιεχόμενα του σχεσιακού πίνακα **products** (id, code, name, category, price) στην βάση δεδομένων. Η αναζήτηση γίνεται βάση ορισμένων προαιρετικών κριτηρίων.

Το συστατικό αναζήτησης υποστηρίζει την λειτουργία αναζήτησης προϊόντων.

Το συστατικό αναζήτησης δεν κρατάει την κατάσταση του (stateless) και έτσι για κάθε κλήση μεθόδων του με όμοιες παραμέτρους εισόδου, επιστρέφει όμοια αποτελέσματα.

Στον Πίνακα 2 παρατίθεται η περιγραφή μεθόδων για το Συστατικό Αναζήτησης.

Πίνακας 2 Περιγραφή Μεθόδων του Συστατικού Αναζήτησης

Μέθοδος (method)	Παράμετροι εισόδου (Input Parameters)	Περιγραφή (Description)
init	-	Μέθοδος αρχικοποίησης του συστατικού. Επιστρέφει ένα γραφικό περιβάλλον με τον πίνακα κριτηρίων αναζήτησης και την επιλογή search (αναζήτηση) η οποία στέλνει μήνυμα εκτέλεσης στην μέθοδο “search” του ίδιου του συστατικού αναζήτησης.
search	<ul style="list-style-type: none"> • t1 – Κωδικός προϊόντος • t2 – Όνομα προϊόντος • t3 – κατηγορία προϊόντος • t4 – Τιμή προϊόντος • t4op – Τελεστής σύγκρισης τιμής προϊόντος (>=, <=, =) 	<p>Με βάση τις παραμέτρους εισόδου το συστατικό αναζήτησης επιχειρεί αναζήτηση στην βάση δεδομένων και επιστρέφει έναν πίνακα με όλα τα προϊόντα που ταιριάζουν στα κριτήρια αναζήτησης. Επίσης, θέτει τις κατάλληλες τιμές στην συνεδρία (session) του χρήστη έτσι ώστε να διατηρηθούν οι τιμές της αναζήτησης σε περίπτωση επιστροφής στην σελίδα κριτηρίων.</p> <p>Η σελίδα αποτελεσμάτων περιέχει υπερσυνδέσμους μέσω των οποίων υπάρχει η δυνατότητα αποστολής μηνυμάτων στο συστατικό «καλάθι αγορών» έτσι ώστε να ενημερωθεί για την προσθήκη νέου προϊόντος.</p>

4.4.2 Συστατικό Καλάθι Αγορών

Το συστατικό αναζήτησης διατηρεί την κατάσταση του (statefull) και έτσι για κάθε κλήση μεθόδων του με όμοιες παραμέτρους εισόδου, δεν επιστρέφει πάντα όμοια αποτελέσματα. Τα αποτελέσματα εξαρτώνται από την προηγούμενη κατάσταση του συστατικού καθώς και από την μέθοδο που κλήθηκε.

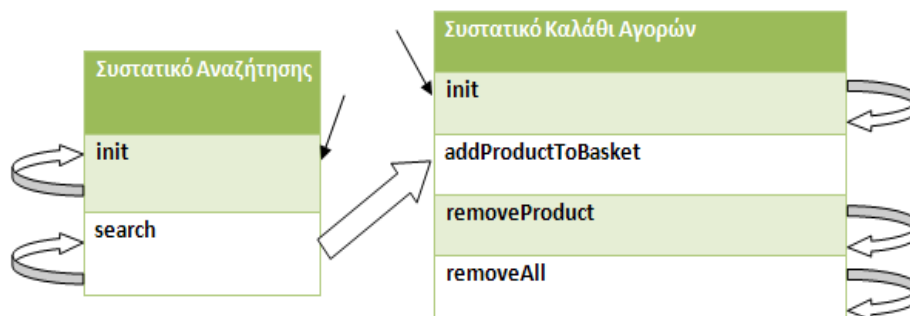
Στον Πίνακα 3 παρατίθεται η περιγραφή μεθόδων για το Συστατικό Καλάθι Αγορών.

Πίνακας 3 Περιγραφή Μεθόδων του Συστατικού Καλάθι Αγορών

Μέθοδος (method)	Παράμετροι εισόδου (Input parameters)	Περιγραφή (Description)
<code>init</code>	-	Μέθοδος αρχικοποίησης του καλάθι αγορών. Το καλάθι αρχικοποιείται και ανασύρει από το αντικείμενο συνεδρίας (session object) όλα τα προϊόντα που βρίσκονται σε αυτό.
<code>addProductToBasket</code>	<code>id</code> – το αναγνωριστικό σε επίπεδο βάσης δεδομένων του προϊόντος που θα εισαχτεί στο καλάθι αγορών.	Το καλάθι προσπαθεί να προσθέσει το προϊόν με αναγνωριστικό <code>id</code> (σε επίπεδο βάσης δεδομένων). Το καλάθι ενημερώνει τον χρήστη ανάλογα για το αν εισήχθη επιτυχώς το προϊόν σε αυτό.
<code>removeProduct</code>	<code>id</code> – το αναγνωριστικό σε επίπεδο βάσης δεδομένων του προϊόντος που θα αφαιρεθεί από το καλάθι.	Το συστατικό αφαιρεί από την λίστα με τα προϊόντα που διατηρεί το προϊόν με αναγνωριστικό <code>id</code> .
<code>removeAll</code>	-	Το συστατικό αφαιρεί από την λίστα με τα προϊόντα που διατηρεί όλα τα προϊόντα.

4.4.3 Επικοινωνία Συστατικών

Κάθε λειτουργία (αντικατοπτρίζεται σε μια μέθοδο) των συστατικών υποκρύπτει την επικοινωνία μεταξύ δυο συστατικών ή ενός με τον εαυτό του. Όλες οι μέθοδοι των δυο συστατικών του συστήματος αφορούν επικοινωνία συστατικού με τον εαυτό του, εκτός από την μέθοδο `addProductToBasket`, της οποίας η κλήση γίνεται από το συστατικό αναζήτησης προς το συστατικό καλάθι αγορών. Τα κάθε συστατικό δεν εξετάζει αν το μήνυμα (κλήση της μεθόδου) στάλθηκε από το ίδιο ή από ένα τρίτο. Ο ρόλος του είναι να συλλέξει τις παραμέτρους εισόδου και να επιστρέψει τα ανάλογα αποτελέσματα τα οποία θα διαχειριστεί η ανάλογη υπηρεσία της πλατφόρμας ή συνάρτηση του συστατικού. Στην Εικόνα 6 Επικοινωνία Συστατικών συνοψίζεται γραφικά η επικοινωνία μεταξύ των δυο συστατικών. Οι μέθοδοι `init`, παρουσιάζουν την ιδιαιτερότητα ότι καλούνται κατά το φόρτωμα της σελίδας από την ίδια την πλατφόρμα συστατικών.



Εικόνα 6 Επικοινωνία Συστατικών

4.4.4 Γεγονότα

Από την σκοπιά του προγραμματισμού βασισμένου σε γεγονότα παρατηρείται ότι η λειτουργία και η επικοινωνία των συστατικών βασίζεται σε γεγονότα.

Για παράδειγμα, το γεγονός έναρξης της σελίδας (onload) προκαλεί την αρχικοποίηση των συστατικών μέσω αποστολής μηνύματος για την λειτουργία `init` (από την πλατφόρμα συστατικών).

Στον Πίνακα 4 συνοψίζονται τα γεγονότα σε συνδυασμό με τις πράξεις (actions) που προκαλούν.

Πίνακας 4 Αντιστοίχιση Μεθόδων και Γεγονότων

Γεγονός	Συστατικό αναζήτησης – Μέθοδος	Συστατικό Καλάθι Αγορών - Μέθοδος
onload	init	init
Click on search button	search	
Click on “+” on search results of search component		addProductToBasket
Click on remove		removeProduct
Click on remove all product		removeAll

4.5 Επεκτείνοντας το Σύστημα

4.5.1 Ανάγκη Επέκτασης του Συστήματος

Όπως επισημάνθηκε και στην θεωρητική ανάλυση του προβλήματος, τα πλεονεκτήματα του προγραμματισμού βασισμένου σε συστατικά εμφανίζονται σε περιπτώσεις επέκτασης του συστήματος.

Στο σύστημα ηλεκτρονικό κατάστημα υπάρχει η ανάγκη για λεπτομερέστερη προβολή πληροφοριών σχετικά με ένα προϊόν. Συγκεκριμένα όταν ένα προϊόν είναι στην λίστα αποτελεσμάτων ή βρίσκεται στο καλάθι αγορών τότε είναι απαραίτητο για τον χρήστη να μπορεί εύκολα να δει λεπτομέρειες για αυτό το προϊόν (όπως αναλυτική περιγραφή, εικόνα προϊόντος).

Τα υπάρχοντα συστατικά δεν έχουν ως σκοπό την υποστήριξη τέτοιας λειτουργικότητας και ως εκ τούτου θα πρέπει να εισαχθεί ένα νέο συστατικό, το συστατικό προβολής λεπτομερειών προϊόντος (viewProductDetailsComponent). Για κάθε προϊόν που θα προβάλλεται από το συστατικό προβολής λεπτομερειών θα υπάρχει ένας υπερσύνδεσμος στο τέλος για αγορά του προϊόντος (μεταφορά του στο καλάθι αγορών).

Επίσης, είναι χρήσιμο για την προώθηση των προϊόντων να υπάρχει ένας χώρος στην σελίδα όπου θα εμφανίζονται τα νέα προϊόντα. Γι αυτό τον σκοπό θα εισαχθεί στο σύστημα ένα επιπλέον συστατικό, το προβολή νέων προϊόντων

(viewLatestProductsComponent) με σκοπό την συνοπτική προβολή τριών προϊόντων επιλεγμένα τυχαία μέσα από τα δέκα πιο πρόσφατα εισαγμένα στην βάση δεδομένων. Το συστατικό θα συνδέεται με το συστατικό προβολής λεπτομερειών προϊόντος. Σε ενδεχόμενο κλικ του χρήστη επάνω σε ένα προϊόν τότε το συστατικό θα στέλνει μήνυμα στο συστατικό προβολής λεπτομερειών έτσι ώστε να προβάλλει τις λεπτομέρειες του προϊόντος που επιλέχθηκε.

4.5.2 Συστατικό Προβολής Λεπτομερών Προϊόντος

Το συστατικό προβολής λεπτομερών προϊόντος μπορεί να αναπτυχθεί σε ένα περιβάλλον ανάπτυξης, ανεξάρτητο από την εφαρμογή που τελικά θα το φιλοξενήσει και έτσι να μην επηρεάσει σε πρώτο βαθμό τα άλλα συστατικά του συστήματος. Έτσι, ο προγραμματιστής καταναλώνει λιγότερο χρόνο γιατί σε πρώτη φάση δεν ασχολείται με το πώς θα εισάγει μέσα στην εφαρμογή το συστατικό. Αυτό θα γίνει αφού ολοκληρωθεί η ανάπτυξη του και με τρόπο συγκεκριμένο.

Το συστατικό προβολής λεπτομερών προϊόντος έχει μια μέθοδο, την `viewDetails` με παραμέτρους εισόδου ένα αναγνωριστικό βάσης δεδομένων προϊόντων (`products.id`). Η μέθοδος του `init`, δεν παρέχει καμία λειτουργικότητα.

Το συστατικό αναζήτησης δεν κρατάει την κατάσταση του (`stateless`) και έτσι για κάθε κλήση μεθόδων του με όμοιες παραμέτρους εισόδου, επιστρέφει όμοια αποτελέσματα.

Για την ανάπτυξη του συστατικού είναι απαραίτητο να τροποποιηθεί η βάση δεδομένων και συγκεκριμένα ο πίνακας `products`, στον οποίο θα προστεθούν δυο επιπλέον πεδία (`description`, `image_url`).

4.5.3 Εισαγωγή Συστατικού στο Σύστημα

Οι προγραμματιστές που είναι επωμισμένοι με την συντήρηση των συστατικών αναζήτησης και καλαθιού αρχικά απαντούν στο κατά πόσο οι αλλαγές στην βάση δεδομένων επηρεάζουν την λειτουργία των συστατικών τους. Στην προκειμένη περίπτωση η απάντηση κα για τα δύο συστατικά είναι αρνητική, οι αλλαγές στην βάση δεδομένων δεν τα επηρεάζουν.

Εν συνεχεία, εισάγεται το συστατικό στο σύστημα. Αρχικά, εγκαθίσταται στο εξυπηρέτη μεταφέροντας τα αρχεία του στους κατάλληλους φακέλους και στην

συνέχεια αρχικοποιείται στο γραφικό περιβάλλον με έναν ορισμό <div> (βλ. παράγραφο 3.4.3).

Κατά την έναρξη της σελίδας το συστατικό δεν είναι ορατό. Είναι μόνο οι πληροφορίες αποσφαλμάτωσης του (έχοντας ενεργοποιημένη την λειτουργία παροχής πληροφοριών αποσφαλμάτωσης στην πλατφόρμα συστατικών -Εικόνα 7).

Το συστατικό προβολής λεπτομερειών προϊόντος είναι πλέον μέρος του συστήματος και απομένει η σύνδεση του με τα άλλα συστατικά. Οι προγραμματιστές που έχουν αναλάβει την συντήρηση και επέκταση του συστατικού αναζήτησης προϊόντος μετατρέπουν σε υπερσύνδεσμο τον κωδικό του κάθε προϊόντος στην σελίδα των αποτελεσμάτων. Το πάτημα του υπερσυνδέσμου προκαλεί την αποστολή μηνύματος στην μέθοδο viewProductDetails του συστατικού προβολής λεπτομερειών προϊόντος με το εσωτερικό αναγνωριστικό της βάσης δεδομένων για το προϊόν αυτό. Το συστατικό προβολής λεπτομερειών επιστρέφει στο γραφικό κομμάτι που του αναλογεί τις λεπτομέρειες του προϊόντος. Ομοίως, μετατρέπονται σε υπερσυνδέσμους, από τους υπεύθυνους προγραμματιστές, οι κωδικοί των προϊόντων που βρίσκονται στο καλάθι αγορών.

Στην Εικόνα 7 φαίνεται το σύστημα κατά την εκτέλεση του.

The screenshot displays a web application interface with three main panels. The left panel shows search criteria and a table of search results. The middle panel displays the details of a selected product, including its cover and description. The right panel shows the contents of a shopping basket.

Code	Name	Price €	Category	Add
00002001	Love in the Time of Cholera by Gabriel Garcia Marquez	11.3	Books	to Basket
00002000	One Hundred Years of Solitude by Gabriel Garcia Marquez	12.5	Books	to Basket
00002002	Memories of My Melancholy Whores by Gabriel Garcia Marquez	15.22	Books	+

Product Details for 'Memories of My Melancholy Whores by Gabriel Garcia Marquez':

- Code: 00002002
- Title: Memories of My Melancholy Whores by Gabriel Garcia Marquez
- Category: Books
- Price(€): 15.22
- Description: ..one of twentieth-century literature?s great figures pushes back the years and gives us fiction of the very highest order.

Shopping Basket:

- 1. Code: 00002001, Category: Books, Title: Love in the Time of Cholera by Gabriel Garcia Marquez, Price: 11.3€, remove
- 2. Code: 00002000, Category: Books, Title: One Hundred Years of Solitude by Gabriel Garcia Marquez, Price: 12.5€, remove

Total Cost: 23.86€

remove all products, reload basket

Εικόνα 7 Έναρξη Σελίδας με το Συστατικό Προβολής Λεπτομερειών Προϊόντος

4.5.4 Συστατικό Προβολής Νέων Προϊόντων

Ομοίως με το συστατικό προβολής λεπτομερειών, αναπτύσσεται και εγκαθίσταται στον εξυπηρέτη το συστατικό προβολής νέων προϊόντων.

Το συστατικό έχει μόνο την μέθοδο `init` η οποία καλείται αυτόματα κατά το φόρτωμα της σελίδας. Το συστατικό επικοινωνεί με το συστατικό προβολής λεπτομερειών. Όταν ο χρήστης κάνει κλικ επάνω σε ένα από τα τρία προϊόντα που προβάλλονται στέλνεται ένα μήνυμα στο συστατικό προβολής λεπτομερειών έτσι ώστε να προβάλλει το προϊόν που επιλέχθηκε.

Η εισαγωγή του συστατικού στο σύστημα γίνεται ομοίως με αυτή του συστατικού προβολής λεπτομερειών προϊόντων.

Στην Εικόνα 8 φαίνεται το σύστημα κατά την εκτέλεση του με τα δύο νέα συστατικά.

The screenshot displays a web browser window titled "cbwe - v1.4 in DEVELOPMENT :: - Windows Internet Explorer". The address bar shows "http://localhost:8001/cbwe/1.4/". The page content is organized into several panels:

- Search Products Form:** A form with fields for "Name", "Code", "Category", and "Price" (with a Euro symbol). It includes "clear" and "search" buttons.
- cb e-shop ** Just Arrived ** cb - e-shop:** A section displaying three product thumbnails with their respective titles and prices.
- Basket:** A section showing a list of 3 items in the basket, including their codes, categories, titles, and prices. The total cost is 205.52€. There are buttons for "remove all products", "reload basket", and "checkout basket".
- productDetailsComponent:** A section showing details for a selected product, including its code (00001004), title (INTEL PENTIUM DUAL CORE E2140 1.60 GHZ), and category (Electronics).

Εικόνα 8 Το σύστημα αφού έχει εισαχθεί το συστατικό προβολής νέων προϊόντων

4.5.5 Επεκτείνοντας Ένα Υπάρχον Συστατικό

Για να θεωρείται πλήρες το ηλεκτρονικό κατάστημα είναι απαραίτητο για τον πελάτη να ολοκληρώνει τις αγορές του με τα προϊόντα του καλαθιού αγορών. Στο σύστημα, αυτό θα υποστηριχθεί εισάγοντας την παραγγελία σε ένα νέο πίνακα στην βάση δεδομένων:

```
[orders (id, order_code, creation_date, product_ids, process_status, first_name, surname, street_name, street_no, city, phone_no) ]
```

Η νέα λειτουργία θα εισαχθεί στο συστατικό καλάθι αγορών καθώς σημασιολογικά συνδέεται με αυτό και τους σκοπούς του.

Η δημιουργία του νέου πίνακα στην βάση δεδομένων δεν επηρεάζει τα άλλα δυο συστατικά.

Η επέκταση του συστατικού καλάθι αγορών επιφέρει τον εμπλουτισμό της διεπαφής του με δυο καινούργιες μεθόδους, οι υπόλοιπες μέθοδοι του δεν αλλάζουν επομένως δεν επηρεάζεται το υπόλοιπο σύστημα. Οι δυο νέες μέθοδοι είναι:

Πίνακας 5 Νέοι μέθοδοι του συστατικού «καλάθι αγορών»

Μέθοδος (method)	Παράμετροι εισόδου (Input parameters)	Περιγραφή (Description)
checkout_prepare	-	Επιστρέφει την φόρμα παραγγελίας την οποία ο χρήστης καλείται να συμπληρώσει.
Checkout_finish	Καλείται με παράμετρο εισόδου την φόρμα παραγγελίας (checkout_form) που συμπλήρωσε ο χρήστης.	Καταχωρεί την παραγγελία στην βάση δεδομένων και δημιουργεί έναν μοναδικό κωδικό παραγγελίας και τον επιστρέφει στον χρήστη.

4.6 Υιοθέτηση Συστατικών σε Νέα Συστήματα

4.6.1 Εφαρμογές που Στεγάζονται στον Ίδιο Εξυπηρέτη

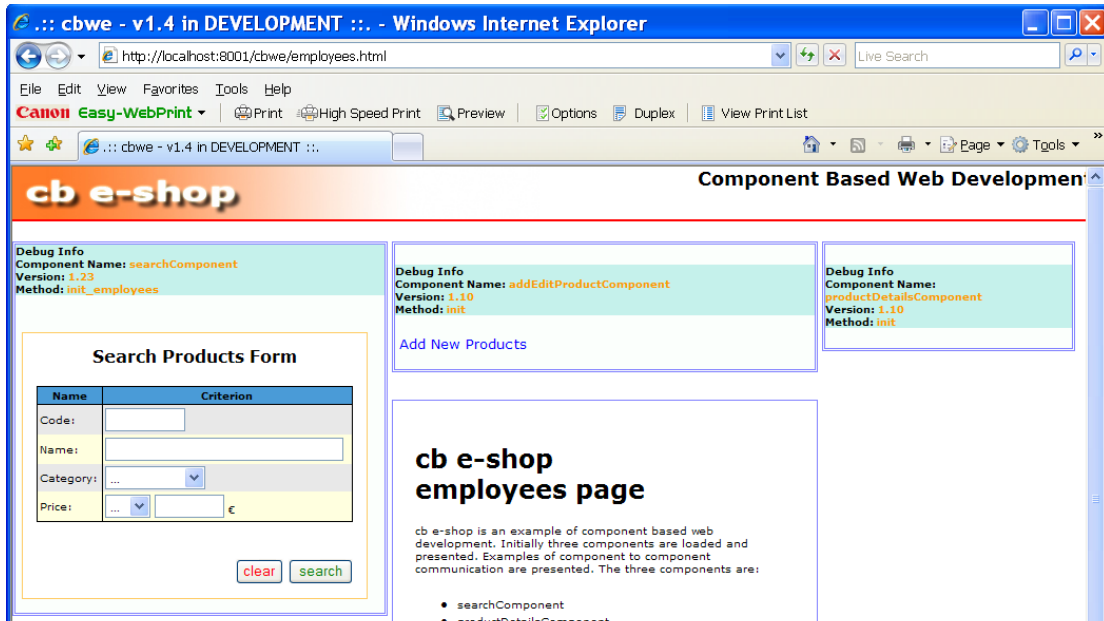
Καθώς το ηλεκτρονικό κατάστημα μεγαλώνει τόσο αυξάνει η ανάγκη για αυτοματοποιημένη λειτουργία του. Στην ως τώρα υλοποίηση δεν έχει προβλεφθεί γραφικό περιβάλλον με τις ανάλογες λειτουργίες για την εισαγωγή, επεξεργασία και διαγραφή προϊόντων από την βάση δεδομένων. Για τον σκοπό αυτό θα εισαχθεί μια νέα εφαρμογή στην οποία θα έχουν πρόσβαση μόνο οι υπάλληλοι της εταιρίας μέσα από τοπικό δίκτυο της εταιρίας. Η εφαρμογή θα φιλοξενηθεί στον ίδιο εξυπηρέτη με την εφαρμογή ηλεκτρονικό κατάστημα και θα έχει πρόσβαση στην ίδια βάση δεδομένων. Μέσω αυτής της σελίδας οι υπάλληλοι θα έχουν την δυνατότητα εύρεσης, επεξεργασίας, διαγραφής προϊόντων καθώς και την εισαγωγή νέων.

Το συστατικό εύρεσης προϊόντων φαίνεται να είναι χρήσιμο στο κομμάτι της εύρεσης των προϊόντων ενώ ένα νέο συστατικό (addEditProductsComponent) θα δίνει την δυνατότητα επεξεργασίας των υπαρχόντων συστατικών και την δυνατότητα εισαγωγής νέων.

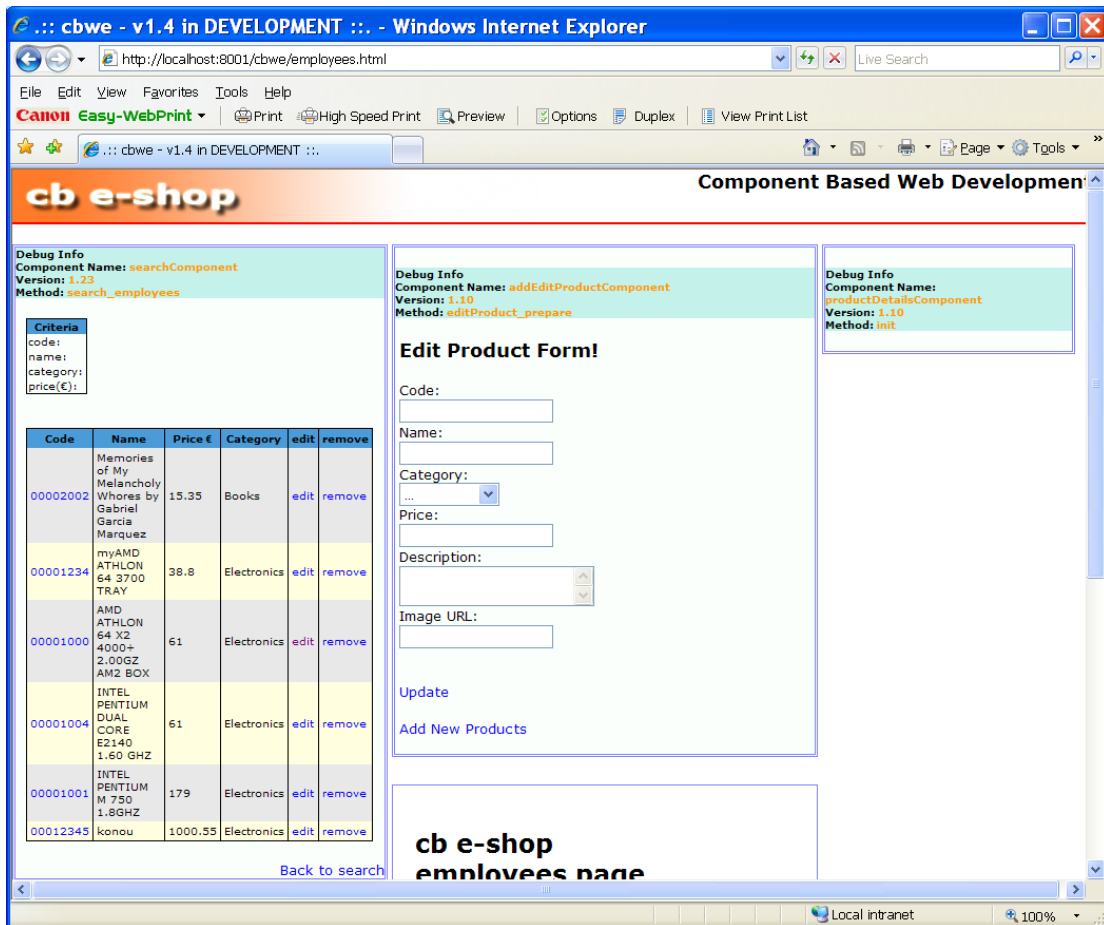
4.6.2 Σενάριο νέας σελίδας

Η νέα σελίδα των υπαλλήλων θα έχει δυο συστατικά (το συστατικό εύρεσης και το συστατικό επεξεργασίας/εισαγωγής) που θα φορτώνουν κατά το άνοιγμα της. Κατά το άνοιγμα του, το συστατικό εύρεσης θα φορτώνει την βασική (default) φόρμα αναζήτησης. Μετά το συμπλήρωμα των κριτηρίων και το πάτημα του κουμπιού “search” το συστατικό θα ανανεώνεται προβάλλοντας τα αποτελέσματα. Στην τελευταία στήλη των αποτελεσμάτων έχει αντικατασταθεί ο υπερσύνδεσμος που προσθέτει το προϊόν στο καλάθι αγορών με έναν υπερσύνδεσμο που μεταφέρει το προϊόν στο συστατικό επεξεργασίας για ανανέωση των στοιχείων του. Επίσης, υπάρχει μια επιπλέον στήλη που περιέχει έναν υπερσύνδεσμο για την διαγραφή του προϊόντος.

Το συστατικό επεξεργασίας εισαγωγής, φορτώνεται περιέχοντας έναν μόνο υπερσύνδεσμο για την εισαγωγή νέου προϊόντος (Εικόνα 9). Σε περίπτωση που λάβει ένα μήνυμα επεξεργασίας προϊόντος τότε ανανεώνεται προβάλλοντας την φόρμα επεξεργασίας συστατικού (Εικόνα 10).



Εικόνα 9 Προετοιμασία συστατικού επεξεργασίας προϊόντων



Εικόνα 10 Φόρμα επεξεργασίας προϊόντος

4.6.3 Τροποποίηση του Συστατικού Αναζήτησης

Το συστατικό αναζήτησης προϊόντων χρειάζεται να τροποποιηθεί έτσι ώστε να υποστηρίζει τον νέο τρόπο εμφάνισης των αποτελεσμάτων στην περίπτωση που πρόκειται για την σελίδα των υπαλλήλων. Για τον λόγο αυτό προστέθηκαν δυο νέες μέθοδοι στην διεπαφή του συστατικού, οι : `init_employees` και `search_employees`. Κατά το άνοιγμα της σελίδας, το συστατικό αναζήτησης φορτώνεται με την μέθοδο `init_employees`. Για να γίνει αυτό πρέπει να παρακαμφθεί η default μέθοδος `init`. Η πλατφόρμα συστατικών υποστηρίζει αυτήν τη λειτουργία προσθέτοντας μια επιπλέον ιδιότητα (την `constructor`) στον ορισμό της ετικέτας στιγμιότυπου του συστατικού, στην HTML σελίδα. Η πλατφόρμα αναζήτα αυτήν την ιδιότητα και σε περίπτωση που την βρει τότε εκτελεί αυτήν την μέθοδο που ορίζεται εκεί και όχι την `init`. Έτσι, ο ορισμός του συστατικού στην νέα σελίδα είναι ο εξής:

```
<div
  id="searchComponent"
  class="components"
  loadmethod="loadSearchComponent"
  constructor="init_employees">
  Your browser does not support components!
</div>
```

4.6.4 Εισαγωγή του Συστατικού Επεξεργασίας Εισαγωγής

Το συστατικό επεξεργασίας εισαγωγής έχει ως σκοπό την διαχείριση των προϊόντων που διατίθενται μέσω της εφαρμογής ηλεκτρονικό κατάστημα. Συνεπώς, υποστηρίζει λειτουργίες εισαγωγής νέου προϊόντος, διαγραφής ενός προϊόντος και επεξεργασίας των στοιχείων ενός προϊόντος.

Η διαγραφή προϊόντος ξεκινά από το συστατικό αναζήτησης (μέσω του υπερσυνδέσμου που υπάρχει στον πίνακα των αποτελεσμάτων), άλλα η λειτουργία της γίνεται από το συστατικό επεξεργασίας εισαγωγής γιατί το συστατικό επεξεργασίας είναι υπεύθυνο για την διαχείριση των προϊόντων.

Στον Πίνακα 6 παρατίθεται η περιγραφή μεθόδων για το Συστατικό Επεξεργασίας.

Πίνακας 6 Μέθοδοι του συστατικού επεξεργασίας

Μέθοδος (method)	Παράμετροι εισόδου (Input parameters)	Περιγραφή (Description)
<code>addProduct_prepare</code>	-	Επιστρέφει την φόρμα εισαγωγής προϊόντος την οποία ο χρήστης καλείται να συμπληρώσει.
<code>addProduct_add</code>	Καλείται με παράμετρο εισόδου την φόρμα εισαγωγής προϊόντος (addProduct_form) που συμπλήρωσε ο χρήστης.	Καταχωρεί το νέο προϊόν στην βάση δεδομένων και επιστρέφει ένα μήνυμα επιτυχίας στον χρήστη.
<code>removeProduct</code>	<code>id</code> – το αναγνωριστικό σε επίπεδο βάσης δεδομένων του προϊόντος που θα διαγραφεί από το σύστημα.	Διαγραφεί το προϊόν από την βάση δεδομένων.
<code>editProduct_prepare</code>	<code>id</code> – το αναγνωριστικό σε επίπεδο βάσης δεδομένων του προϊόντος που θα διαγραφεί από το σύστημα.	Επιστρέφει την φόρμα επεξεργασίας στοιχείων προϊόντος την οποία ο χρήστης καλείται να συμπληρώσει.
<code>editProduct_edit</code>	Καλείται με παράμετρο εισόδου την φόρμα επεξεργασίας στοιχείων προϊόντος (editProduct_form) που συμπλήρωσε ο χρήστης.	Ανανεώνει τα στοιχεία του προϊόντος σε όσα από τα πεδία συμπληρώθηκαν και επιστρέφει ένα μήνυμα επιτυχίας στον χρήστη.

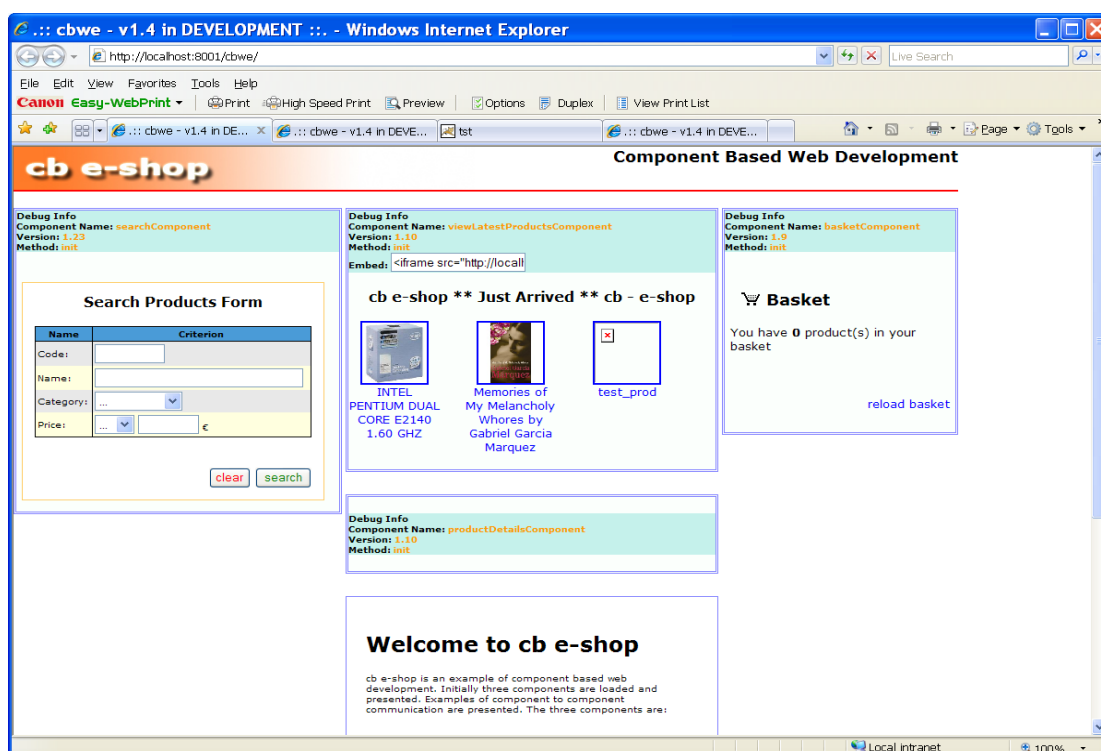
4.6.5 Εφαρμογές που Στεγάζονται σε Απομακρυσμένους Εξυπηρέτες (Τρίτων)

Η πλατφόρμα συστατικών υποστηρίζει την δυνατότητα χρησιμοποίησης των συστατικών σε απομακρυσμένους εξυπηρέτες με την χρήση της τεχνικής των iframes (βλ. παράγραφο 3.5.2).

Από το παράδειγμα του ηλεκτρονικού καταστήματος θα ήταν χρήσιμο, για διαφημιστικούς λόγους, να δίνεται η δυνατότητα επαναχρησιμοποίησης, σε εφαρμογές τρίτων, του συστατικού προβολής νέων προϊόντων. Η μόνη τροποποίηση που απαιτείται στο συστατικό, είναι κατά στην περίπτωση διαχείρισης του γεγονότος

κλικ επάνω σε ένα προϊόν, από τον χρήστη. Τότε, αντί να στέλνεται ένα μήνυμα στο συστατικό προβολής λεπτομερειών, ο χρήστης πρέπει να μεταφέρεται στην αρχική σελίδα του ηλεκτρονικού καταστήματος. Συνεπώς, το συστατικό πρέπει να υποστηρίζει μια νέα μέθοδο (init_external).

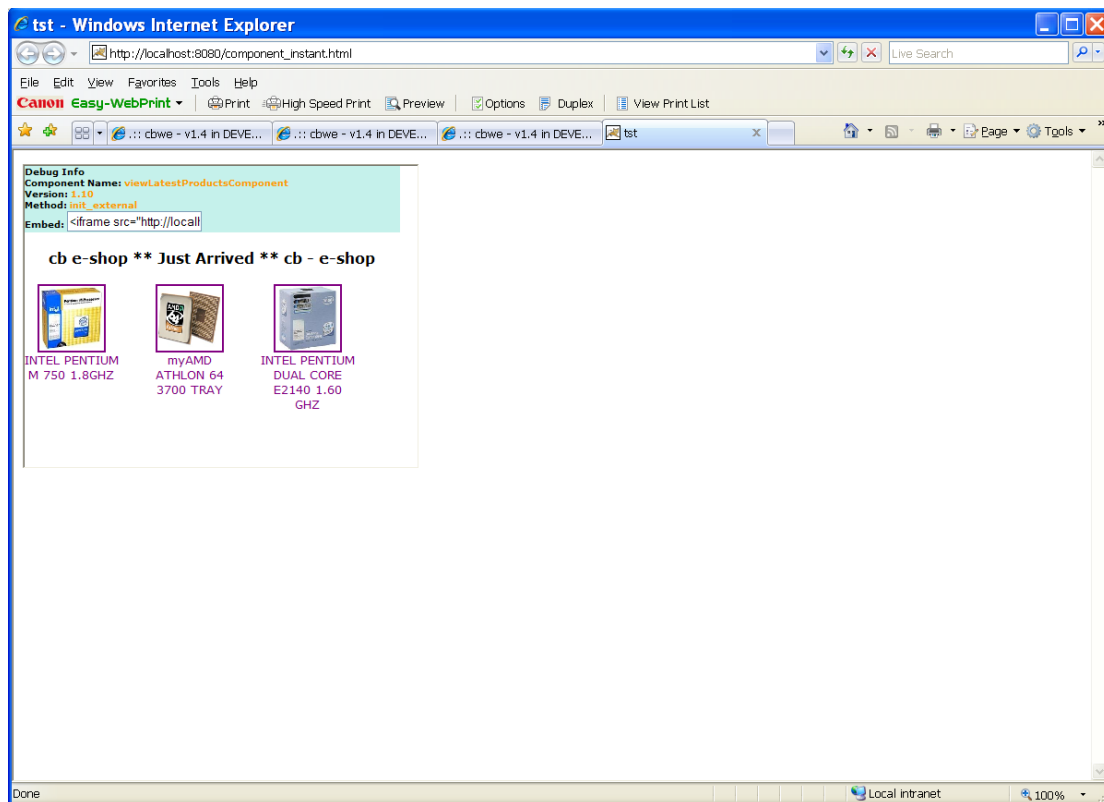
Επίσης στις λειτουργίες αποσφαλμάτωσης είναι δυνατόν να οριστεί, μέσα στο συστατικό, μια μεταβλητή \$embed, η οποία θα περιέχει το κείμενο της iframe ετικέτας. Με αυτόν τον τρόπο, θα εμφανίζεται στο πλαίσιο πληροφοριών αποσφαλμάτωσης του συστατικού η HTML ετικέτα μέσω της οποίας μπορεί το συστατικό να αρχικοποιηθεί σε εξυπηρέτες τρίτων (Εικόνα 11). Αρκεί να αντιγραφεί και να επικολληθεί στην/στις σελίδα/ες του απομακρυσμένου εξυπηρέτη έτσι ώστε να ενσωματωθεί σε αυτήν (Εικόνα 12).



Εικόνα 11 Ενεργοποίηση της μεταβλητής embed για το συστατικό προβολής νέων προϊόντων

Ο κώδικας της ετικέτας iframe στο παραπάνω παράδειγμα είναι ο εξής:

```
<iframe
  src="http://localhost:8001/cbwe/components/componentControllerExternal.php?componentName=viewLatestProductsComponent&method=init_external"
  name="comp"
  id="comp"
  width="430"
  height="300" >
</iframe>
```



Εικόνα 12 Συστατικό προβολής νέων προϊόντων σε εφαρμογές τρίτων

4.7 Συμπεράσματα

Με το παραπάνω παράδειγμα είναι σαφής ο καταμερισμός των ευθυνών που υπάρχει ανάμεσα στους προγραμματιστές.

Επίσης φαίνεται ότι υπάρχει ένας συγκεκριμένος τρόπος διερεύνησης του κατά ποσό επηρεάζεται συνολικά το σύστημα όταν εισάγεται σε αυτό ένα νέο συστατικό. Ο έλεγχος γίνεται με δυο βήματα:

1. Έλεγχος και ερώτηση για κάθε συστατικό αν οι αλλαγές στην βάση δεδομένων το επηρεάζουν.
2. Οι αλλαγές σε ένα συστατικό επηρεάζουν τα άλλα αν και μόνο αν αλλάξει η υπογραφή μιας ή περισσότερων μεθόδων του.

Κεφάλαιο 5

5. Συμπεράσματα

5.1 Διαφορές από Παραδοσιακές Διαδικτυακές Εφαρμογές

Οι εφαρμογές που βασίζονται στο μοντέλο συστατικών που περιγράφηκε (κεφάλαιο 3) παρουσιάζουν «ασυμβατότητα» με ορισμένες συνήθειες χρηστών όπως η χρήση των κουμπιών «πίσω» (back) και «μπροστά» (forward) του φυλλομετρητή ιστού (browser). Όπως και με την λειτουργία ανανέωσης της σελίδας (refresh). Η χρήση αυτών των λειτουργιών μπορεί να προκαλέσει προβλήματα στην λειτουργία της εφαρμογής και ο χρήστης να βρεθεί αντιμέτωπος με δυσάρεστες καταστάσεις. Συνεπώς, η εφαρμογή και οι σελίδες της πρέπει να είναι σχεδιασμένες με τέτοιο τρόπο ώστε να αποθαρρύνουν με έμμεσο τρόπο τον χρήστη να πατήσει κάποια από αυτά τα κουμπιά. Επίσης, κάθε συστατικό θα πρέπει να παρέχει λειτουργίες επαναφοράς (back) και ανανέωσης (refresh) όταν αυτό χρειάζεται.

Μια άλλη διαφορά σε σχέση με τις συνηθισμένες διαδικτυακές εφαρμογές είναι ότι το περιεχόμενο της σελίδας είναι 100% δυναμικό με αποτέλεσμα να μην έχουν εφαρμογή παραδοσιακές τεχνικές διερεύνησης και αποσφαλμάτωσης (όπως προβολής πηγαίου κώδικα σελίδας ή χρήση κονσόλας Javascript για την εμφάνιση λαθών).

Όλες οι λειτουργίες είναι κλήσεις σε Javascript συναρτήσεις με αποτέλεσμα να μην είναι δυνατό το άνοιγμα λειτουργιών σε νέο παράθυρο (τακτική που ακολουθούν πολλοί χρήστες).

Επιπλέον, εφόσον η υιοθέτηση του μοντέλου συστατικών οδηγεί σε πλούσιους πελάτες (rich clients), αυξάνονται οι απαιτήσεις σε υπολογιστική ισχύ σε σχέση με τις παραδοσιακές διαδικτυακές εφαρμογές.

5.2 Μελλοντική Δουλειά

Το μοντέλο συστατικών που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας είναι πλήρες, με δεδομένο ότι παρέχει όλες τις υπηρεσίες για την δημιουργία πραγματικών εφαρμογών διαδικτύου που στηρίζονται σε αυτό. Ωστόσο, οι ανάγκες εξέλιξης του μοντέλου είναι δεδομένες μια και πρέπει να υποστηρίζει όλο και περισσότερες απαιτήσεις των χρηστών του (τους προγραμματιστές).

Παρακάτω αναφέρονται προσθήκες που θα ήταν χρήσιμες να γίνουν στο μοντέλο συστατικών:

- Ανάπτυξη της πλατφόρμας και σε άλλες διαδεδομένες γλώσσες προγραμματισμού (πχ Java).
- Ανάπτυξη IDE που να υποστηρίζει το μοντέλο συστατικών έτσι ώστε να επιταχύνεται η ανάπτυξη των συστατικών.
- Υποστήριξη περιβάλλοντος με περισσότερους εξυπηρετές.
- Πακετάρισμα των αρχείων των συστατικών σε ένα συμπιεσμένο αρχείο και εγκατάσταση αυτού του αρχείου στην πλατφόρμα.
- Δημιουργία γραφικού περιβάλλοντος για την διαχείριση των λειτουργιών της πλατφόρμας (πχ ενεργοποίηση/απενεργοποίηση λειτουργίας αποσφαλμάτωσης).
- Υποστήριξη επικοινωνίας συστατικού προς συστατικά.

Επίσης, σε επίπεδο σχεδίασης της εφαρμογής, το κομμάτι της ανάπτυξης διαδικτυακών εφαρμογών που βασίζονται σε συστατικά παρουσιάζει ενδιαφέρον για έρευνα. Ορισμένοι συγκεκριμένοι τομείς έρευνας είναι:

- Πρότυπα σχεδίασης για αρχιτεκτονικές βασισμένες στο μοντέλο συστατικών που ορίστηκε
- Σχεδίαση εφαρμογών μεγάλης κλίμακας και διαμοιρασμός των λειτουργιών τους σε συστατικά

Τέλος, μέσω της ιδιότητας της επαναχρησιμοποίησης που παρουσιάζουν τα συστατικά, αναδεικνύεται η δυνατότητα διανομής λειτουργικών κομματιών της εφαρμογής από ιστοσελίδες και τρίτων. Οι εμπορικές και λειτουργικές επιπτώσεις που επιφέρει η δυνατότητα αναδιανομής μπορούν να αποτελέσουν πεδίο έρευνας.

Βιβλιογραφία

- [1] Redolfi G, Spagnoli L, Hemesath P, Bastos R M, Riberiro M B, Cristal M. & Espindola A. (2005). “A Reference Model for Reusable Components Description”. In: *Proceedings of the 38th Hawaii International Conference on System Science*, Hawaii, US.

- [2] Plakosh D., Smith D., Kurt C. W. (1999). *Builder’s Guide for WaterBeans Components*. Tech Report, Carnegie Mellon Software Engineering Institute. Ανάκτηση 01/02/2008 από World Wide Web:
<http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr024.pdf>

- [3] Hofer E. (2007), *Component-Based Web Development*. Master Thesis. University of Technology. Graz. Austria.

- [4] Maurer P. (2003). *Component-Level Programming*. New Jersey. Prentice Hall.

- [5] Szyperski, C. (1998). *Component Software: Beyond Object-Oriented Programming*. New York, NY: ACM Press and Addison-Wesley.

- [6] McIlroy, D. (1968). *Mass Produced Software Components*. Ανάκτηση 01/02/2008 από World Wide Web:
<http://www.cs.dartmouth.edu/~doug/components.txt>

- [7] Brown A. W. & Short K. (1997). On components and objects: The foundation of component-based development. In: *International Symposium on assessments of software tools and technologies*, 5, Pittsburg, PA, IEEE Press, 112-121.

- [8] Sametinger J. (1997). *Software Engineering with Reusable Components*. New York, NY: Springer-Verlag.

- [9] Gaedke M. & Rehse J. (1999). *Supporting Compositional Reuse in Component Based Web Engineering*. Ανάκτηση 01/02/2008 από World Wide Web: <http://www.teco.edu/~gaedke/paper/2000-sac2000-gr.pdf>
- [10] Bachmann, F. (2000). *Volume II: Technical concepts of component-based software engineering*. Tech Report. 008, Carnegie Mellon Software Engineering Institute. Ανάκτηση 01/02/2008 από World Wide Web: <http://www.sei.cmu.edu/pub/documents/00.reports/pdf/00rt008.pdf>