Μελετόντας το pServer -User Guide

PServer χρησιμοποιεί RDBMS
requests σε HTTP (με τις παραμέτρους)
results σε XML

ΛΟΓΙΚΌ ΕΠΙΠΕΔΟ:
κεντρικό σημείο το User Modeling
παρέχωνται 3 models

-> Personal User Models
Περιλαμβάνουν τις πληροφορίες που αποθηκεύονται για κάθε χρηστη (π.χ. ποια links έχει χτυπήσει, πόσες φορές, κτλ)

Ο user καθορίζεται από συγκεκριμένα 'attributes' και 'features'

-Τα 'attributes' είναι προσωπικές πληροφορίες που δεν αλλάζουν (πχ γλώσσα, ηλικία, κτλ)
-Τα 'features' έχουν να κάνουν με τα χαρακτηριστικά της κάθε περίπτωσης χρήσης του pserver. Είναι αριθμητικά που δείχνουν πόσο σημαντικό είναι τι για τον συγκεκριμένο χρήστη (πχ πόσο σημαντικό είναι γι αυτόν η βιολογία (3), η τεχνολογία (10), κτλ)

-> Stereotypes
Είναι group χρηστών με κοινά 'attributes' και δεν είναι πεπερασμένου αριθμού. Έχουν και αυτά 'attributes' και 'features'
(πχ group με ενήλικες, η group ανθρώπων που τους ενδιαφέρει η τεχνολογία, κτλ)
Χρησιμοποιήτε για το collaboration filtering. Το κάθε στερεότυπο έχει χρήστες που ανήκουν σε αυτό και το προφίλ του υπολογίζεται απο μια μαθιματική φόρμουλα που σαν input έχει τα προφίλ όλων τον χρηστών που το αποτελούν.

-> User ή features Communities
Έίναι 'group χρηστών' ή 'group features' που εξαρτώνται από την αλληλεπίδραση του χρήστη με το σύστημα. Δημιουργούνται με την χρήση machine learning αλγορίθμων.
Τα 'user groups' δημιουργούνται από την εύρεση κοινών 'features' μεταξύ των user (με common likes κτλ) κάνοντας clustering
και για τα 'feature groups' μαθένουμε ποιά απο αυτά 'εχουν περίπου τις ίδιες τιμές για τους χρήστες, δημιουργούνται δηλαδή από "features" που αρέσουν σε πολλούς user ταυτόχρονα. Αυτό συμένει πως αν ένα feature είναι συμαντικό για πολλούς user ταφτόχρονα πρέπει να το ξέρουμε για να μπορούμε να το προτήνοουμε σε νέους χρήστες. Τα features οργανώνονται σε γράφο.

ΦΥΣΙΚΌ ΕΠΙΠΕΔΟ:
-> Ο pServer μπορεί να είναι σε άλλο μηχάνημα σαν app και να υλοποιείται σαν web server που "ακούει" σε συγκεκριμένο port
Αφού παίρνει http requests οι web browsers μπορούν να χρησιμοποιηθούν σας clients.
(πχ: http://server:port/<clnt=client_name|client_pass><mode_id>?<query_string>)
-> Τα responce από τον pServer είναι σε XML (XSL style για να μπορούν να γίνουν displayed σε

browsers.
-> επίσης για την διευκόλυνση υπάρχουν client-side βιβλιοθήκες που μπορούν να ενσοματωθούν στην εφαρμογή για τον χηρισμό των low-level communication details.

ΣΥΣΤΑΤΙΚΑ ΚΑΙ ΕΠΕΞΗΓΗΣΗ:

Το κύριο συστατικό είναι ένα data-model, το οποίο προσφέρει έναν abstract τρόπο παρουσίασης των ενδιαφερόντων του χρήση. Και στο οποίο μπορούν να εφαρμοστούν data mining αλγόριθμοι για να εξάγουμε πληροφορίες.

Με αυτόν το τρόπο ο pServer ανεξαρτιτοποιήτε απο το εκάστοτε app αφού δεν αποθηκεύονται σημιολογίες και οι διαφορετική τύποι περιεχομένου εκφράζονται με γενικούς τύπους.

Ενσοματώνοντας τον pServer στο site ή app:
-> διατύπωση του User Model

Ο admin αποφασίζει ποιά features θέλει να ενσοματώσει στο site του σύμφωνα με τις ανάγκες και απετήσεις
Συγκεκριμένα αν θέλει δεδομένα που αφορούν User Models, Stereotypes, User Communities ή όλα τα παραπάνω

Στη συνέχια πρέπει να συγκεκριμενοποιήσει τα δεδομένα που πρέπει να αποθηκεύονται απο τον pServer για να καληφθούν οι απετήσεις του User model.

2 προτηνόμενοι τρόποι για να δήνονται δεδομένα στον pServer: real time ή periodicl
(εξαρτάτε συνήθος απο trafic του site)

Τέλος πρέπει να υλοποιηθεί ένας μηχανισμός που θα δέχεται τις απαντήσεις του pServer

Επίπεδο Java κώδηκα:

Στο πακλέτο pserver.pservlets βρήσκονται οι μέθοδοι που χρηάζονται για τον χηρισμό των 3 modes αλλά και για την εισαγογή δεδομένων στον pServer.

Cluto (http://glaros.dtc.umn.edu/gkhome/views/cluto)
Το Cluto προσφέρεται για clustering ΒΔ και για την ανάληση των αποτελεσμάτων και των χαρακτηριστικών των cluster. Μπορεί να χρησιμοποιηθεί και σαν βιβλιοθήκη απο την οποία μπορούμε να αντλίσουμε αλγορίθμους clustering και ανάλισης. Στην δική μας περίπτωση είναι χρήσιμο εργαλίο επιδή μπορεί να χηριστεί γράφους (πχ γράφους φιλίας)

Επίσεις υπάρχει το εργαλίο wCluto που είναι ένα web-enabled clustering app βασισμένο στο cluto στο οποίο μπορύμε ανεβάζοντας τα datasets να κάνουμε on-line clustering και ανάληση.
(το link που οδηγεί στον server δεν λειτουργεί επομένος πρέπει να επικοινωνήσουμε με τον developer)

Περετέρο λεπτομέριες για το πως χηρίζεται το cluto γράφους:

Graph File

The primary input of C LUTO's scluster program is the adjacency matrix of the graph that specifies the similarity between the objects to be clustered. Each row/column of this matrix represents a single object, and a value at the (i, j) location of this matrix indicates the similarity between the ith and the jth object.

C LUTO understands two different input graph formats. The first format is suitable for sparse graphs and the second format is suitable for storing dense graphs (i.e., graphs whose adjacency matrix contain mostly non-zeros). The format of these files are very similar to the corresponding formats for matrices, and the only difference is that they now store adjacency matrices which are square.

Sparse Graph Format The adjacency matrix A of a sparse graph with n vertices is stored in a plain text file that contains n + 1 lines. The first line contains information about the size of the graph, while the remaining n lines contain information for each row of A (i.e., adjacency structure of the corresponding vertex). In C LUTO's sparse graph format only the non-zero entries of the adjacency matrix are stored.

The first line of the file contains exactly two numbers, all of which are integers. The first integer is the number of vertices in the graph (n) and the second integer is the number of edges in the graph (i.e., the total number of non-zeros entries in A).

The remaining n lines store information about the actual non-zero structure of A. In particular, the (i + 1)st line of the file contains information about the adjacency structure of the ith vertex (i.e., the non-zero entries of the ith row of the adjacency matrix). The adjacency structure of each vertex is specified as a space-separated list of pairs. Each pair contains the number of the adjacent vertex followed by the similarity of the corresponding edge. The vertex numbers are assumed to be integers and their similarity values are assumed to be floating point numbers.

Note that the vertices are numbered starting from 1 (not from 0 as is often done in C). Furthermore, C LUTO's graph format does not require the vertex-pairs (vertex-number — similarity-value) to be sorted in any order.

Dense Graph Format The adjacency matrix of a dense graph with n vertices is stored in a plain text file that contains n + 1 lines. The first line stores information about the size of the graph, while the remaining n lines contain information for each row of the adjacency matrix. The first line of the file contains exactly one number, which is the number of vertices n of the graph. The remaining n lines store the values of the n columns of the adjacency matrix for each one of the vertices. In particular, each line contains exactly n space-separated floating point values, such that the ith value corresponds to the similarity to the ith vertex of the graph.