ΕΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

Εργασία στο μάθημα Θέματα Εφαρμογών Βάσεων Δεδομένων

Θέμα: Σύγκριση Γλωσσών Επερωτήσεων

Καθηγητής: κος Χατζόπουλος Μ.

Φοιτητές:
Μαρκομανώλης Γεώργιος (Μ781)
Πατσουράκης Νικόλαος (Μ713)

29/6/2006

Περιεχόμενα

Πρόλογος

Η απήχηση του World-Wide Web (WWW) το έχει κάνει κύριο όχημα για την διάδοση πληροφοριών. Η σχέση των εννοιών της βάσης με τα προβλήματα διαχείρισης και επερώτησης πληροφοριών, οδήγησε σε ένα ενδεικτικό σώμα πρόσφατης έρευνας για αυτά τα προβλήματα. Ακόμα και η πρόκληση από την κοινότητα των βάσεων δεδομένων για την διαχείριση μεγάλων όγκων δεδομένων, μας προτρέπουν στην επέκταση των τεχνικών. Η τεχνολογία των βάσεων δεδομένων δεν είναι η μαγική σφαίρα που θα λύσει όλα αυτά τα προβλήματα. Η λύση σε αυτό το θέμα ήρθε από τις γλώσσες αναζήτησης στο web. Είναι γλώσσες που μας δίνουν την δυνατότητα να αναζητούμε πληροφορίες από ιστοσελίδες με βάση τα κριτήρια που θέλουμε εμείς.

Έχουμε τις γλώσσες επερώτησης πρώτης γενιάς, που έχουν στόχο να συνδυάσουν ερωτήματα σχετικά με περιεχόμενο με ερωτήματα σχετικά με την δομή. Αυτές οι γλώσσες που περιλαμβάνουν τις W3QL, WebSQL και WebLog συνδυάζουν συνθήκες με text patterns που εμφανίζονται μέσα σε έγγραφα, με graph patterns που περιγράφουν την δομή των links.

Οι γλώσσες επερώτησης δεύτερης γενιάς, που ονομάζονται “Web data manipulation languages”, πηγαίνουν πιο πέρα από της πρώτης γενιάς, με δύο σημαντικούς τρόπους. Αρχικά προσφέρουν πρόσβαση στην δομή των web-αντικειμένων που διαχειρίζονται. Σε αντίθεση με τις γλώσσες πρώτης γενιάς, μοντελοποιούν τόσο την εσωτερική δομή των web-εγγράφων, όσο και τα εξωτερικά links που τα συνδέουν. Επίσης παρέχουν την δυνατότητα για δημιουργία νέων πολύπλοκων δομών ως αποτέλεσμα σε ερώτημα. Αφού τα δεδομένα στο web είναι κυρίως ημιδομημένα, αυτές οι γλώσσες δίνουν έμφαση στην υποστήριξη για ημιδομημένα χαρακτηριστικά.

Σε αυτή την εργασία, γράφουμε λίγα λόγια για κάποιες γλώσσες επερώτησης στο Web.

WebSQL

Η WebSQL είναι μια γλώσσα αναζήτησης που χρησιμοποιεί το σχεσιακό μοντέλο με τύπο γλώσσας SQL για εξαγωγή πληροφοριών από το web. Οι δυνατότητές της για την πλοήγησή της σε υπερκείμενα στο web, την κάνει χρήσιμο εργαλείο για την αυτοματοποίηση αρκετών διεργασιών που σχετίζονται με το web που απαιτούν την συστηματική επεξεργασία είτε όλων των συνδέσμων μιας σελίδας, όλων των σελίδων που μπορούν να βρεθούν από ένα δεδομένο URL μέσω μονοπατιών που ταυτίζονται με αυτά που θέλουμε να βρούμε. Επίσης η WebSQL παρέχει πρόσβαση σε index servers στους οποίους μπορούμε να κάνουμε ερωτήματα μέσω του Common Gateway Interface.

Μια από τις δυσκολίες, στην δημιουργία γλωσσών αναζητήσεων στο web με μορφή sql γλώσσας, είναι η απουσία ενός σχήματος βάσης για αυτή την τεράστια αποθήκη ετερογενών πληροφοριών. Αν όμως μας ενδιαφέρουν μόνο html αρχεία, μπορούμε να δημιουργήσουμε μόνο εικονικά σχήματα των δομών αυτών των αρχείων. Κάθε έγγραφο αναγνωρίζεται από το URL του, τον τίτλο του και το κείμενό του. Επίσης οι web servers παρέχουν πληροφορίες όπως τον τύπο, το μήκος και ημερομηνία τελευταίας τροποποίησης του εγγράφου. Επομένως από πλευράς data mining μπορούμε να θεωρήσουμε ένα σύνολο όλων των html εγγράφων σαν μια σχέση:

Document(url, title, text, type, length, modif)

Όπου κάθε γνώρισμα είναι συμβολοσειρά. Φυσικά αν κάποιο έγγραφο δεν έχει κάποια πληροφορία, το γνώρισμα μένει κενό. Αφού ορίσαμε αυτή την εικονική σχέση μπορούμε να εκφράσουμε ερωτήματα. Τώρα πρέπει να προσθέσουμε ένα κατηγόρμα *mentions* για ταίριασμα συμβολοσειρών. Για παράδειγμα η έκφραση *x metions y* είναι αληθής, αν το y υπάρχει κάπου στο έγγραφο x.

Παράδειγμα
Να βρούμε έγγραφα σχετικά με πληροφορική.

Select d.url,d.title from Document d such that d mentions “πληροφορική”;

Σημείωση: Το αποτέλεσμα από αυτό το WebSQL ερωτήματος δημιουργείται με την αποστολή της λέξης πληροφορική σε έναν index server. Εφόσον μας ενδιαφέρουν όχι μόνο τα περιεχόμενα του εγγράφου αλλά και η hypertext δομή, χρειαζόμαστε τους συνδέσμους μεταξύ των εγγράφων. Ένας σύνδεσμος χαρακτηρίζεται με το anchor URL του εγγράφου, την επιγραφή του συνδέσμου και τον προορισμό του εγγράφου. Άρα μπορούμε να θεωρήσουμε μια σχέση των συνδέσμων:

Anchor(base, label, href)

Όπου base είναι το URL του anchor εγγράφου, label είναι η επιγραφή του συνδέσμου και href είναι το URL του εγγράφου-προορισμού. Επομένως τώρα μπορούμε να κάνουμε ερωτήματα που αναφέρονται σε συνδέσμους εγγράφων.

Παράδειγμα

Να βρεθούν όλες οι ειδήσεις από το site www.in.gr που περιέχουν την λέξη παιδεία.

select x.url from document x such that http://www.in.gr/index.html=>|->x,anchor y such that base=x where y.label contains “παιδεία”;

Σημείωση: Η λέξη contains στην πρόταση where συγκρίνει την λέξη και βρίσκει αυτά που ταιριάζουν.

Για να μελετήσουμε την τοπολογία του Web πρέπει να διευκρινίσουμε τις διαφορές στα είδη συνδέσμων. Επομένως έχουμε:

- Interior αν ο προορισμός συμπίπτει με το anchor έγγραφο.
- Local αν ο προορισμός και το anchor έγγραφο είναι διαφορετικά αλλά βρίσκονται στον ίδιο server.
- Global αν ο προορισμός και το anchor έγγραφο βρίσκονται σε διαφορετικούς servers.

Χρησιμοποιούμε ένα είδος διανύσματος για το κάθε είδος συνδέσμου.

- #> για interior.
- -> για local.
- => για global.
- = άδειο μονοπάτι.

Οι μορφές του μονοπατιού περιλαμβάνουν αλυσίδα χαρακτήρων για εναλλαγή (|) και επανάληψη (*). Για παράδειγμα το = | =>->* είναι μια έκφραση που απεικονίζει το σύνολο που περιέχει το μηδενικό μονοπάτι, και όλα τα μονοπάτια που αρχίζουν με global σύνδεσμο και συνεχίζουν με μηδενικό ή περισσότερους local συνδέσμους.

Παράδειγμα

Να βρούμε όλα τα έγγραφα από το site της σχολής που αναφέρονται στους μεταπτυχιακούς.

select x.url from document x such that http://www.di.uoa.gr/ ->|=> x where x.url contains “μεταπτυχιακοί”;

Το ->|=> σημαίνει local σύνδεσμος ή global σύνδεσμος.

W3QS

Αν δούμε το Web σαν μια τεράστια βάση δεδομένων μπορούμε να κάνουμε ερωτήματα για επεξεργασία και εύρεση δεδομένων από τους servers. Έχει υλοποιηθεί ένα σύστημα που ονομάζεται W3QS για την εκτέλεση W3QL ερωτημάτων. Η αρχιτεκτονική του W3QS επιτρέπει στους χρήστες να σταματούν τα ερωτήματα καθώς και υλοποίηση άλλων εργαλείων για επεξεργασία δεδομένων. Η W3QL είναι μία γλώσσα υψηλού επιπέδου σαν SQL για το WWW.

Τα κύρια χαρακτηριστικά της W3QL είναι:

- W3QL επιτρέπει και ερωτήματα για το περιεχόμενο και ερωτήματα δομής.(hypertext organization)
- Είναι επεκτάσιμη και μπορεί να συνδυαστεί με προγράμματα χρηστών.
- Είναι καλύτερη από τα υπάρχον WWW ευρετήρια και υπηρεσίας αναζήτησης.
- Παρέχει μία εύκολα ενημερώσιμη όψη

Το W3QS είναι ένα πρωτότυπο σύστημα που εκτελεί W3QL ερωτήματα. Τα κύρια χαρακτηριστικά του W3QS είναι:

- Είναι προσβάσιμο από κάθε WWW browser.
- Παρέχει ένα application program interface (API) που μπορεί να χρησιμοποιηθεί από κάθε πρόγραμμα που τρέχει στο internet.
- Παρέχει εργαλεία για ανάλυση των περιεχομένων αρχείων και εκμάθησης online forms.
- Απλοποιεί την υλοποίηση προγραμμάτων αναζήτησης στο WWW με την παροχή Perl κλάσεων.

Παραδείγματα ερωτήσεων

Θα γράψουμε ένα ερώτημα για να βρούμε όλες τις εικόνες που βρίσκονται στο site της σχολής (http://www.di.uoa.gr).

1. Αρχικά, ψάχνουμε για ένα μονοπάτι που ξεκινάει από την αρχική σελίδα και τελειώνει σε ένα αρχείο εικόνας ακολουθώντας μόνο hypertext συνδέσμους που μένουν μέσα στο αρχικό site. Στην W3QL αυτά εκφράζονται με: n1,l1,(n2,l2),l3,n3 όπου n1,n2.n3 είναι μεταβλητές WWW κόμβων και l1,l2,l3 είναι μεταβλητές WWW συνδέσμων. (n2,l2) είναι ένα μη φραγμένο μονοπάτι σελίδων, προσβάσιμες από την n1.

2. n1 είναι ο πρώτος κόμβος του μονοπατιού και πρέπει να αντιστοιχή στην αρχική σελίδα. Δηλαδή γίνεται αντιστοίχιση συγκεκριμένης σελίδας σε μεταβλητή κόμβου. Αυτό γράφεται ως εξής:
   n1 in {http://www.di.uoa.gr};

3. Οι σύνδεσμοι μεταξύ αυτών των σελίδων, πρέπει να παραμείνουν μέσα στο αρχικό site. Πρέπει η διεύθυνση του στόχου να περιέχει την συμβολοσειρά

"di.uoa.gr". Άρα έχουμε l2 in {/di\.uoa\.gr/}. Τα \,/ χρειάζονται για την Perl.

4. Ο τελευταίος κόμβος n3 πρέπει να είναι εικόνα, που βρίσκεται μέσα στο site: n3 in {/di\.uoa\.gr/}. Η W3QL επιτρέπει σε εξωτερικά προγράμματα (παράδειγμα το PERLCOND) να αναλύουν τα περιεχόμενα των αρχείων. Επομένως n3: PERLCOND 'n3.format=~/image/';

5. Τέλος περιορίζουμε το βάθος αναζήτησης σε 1000 και το μέγιστο μήκος μονοπατιών σε 5, για να μην επιβαρύνουμε τον server. Αυτό γίνεται περνώντας ορίσματα σε ένα απομακρυσμένο πρόγραμμα αναζήτησης (RSP) που ονομάζεται ISEARCHd.

Επομένως έχουμε:

```
Select
From n1,l1,(n2,l2),l3,n3
Where
n1 in {http://www.di.uoa.gr};
l1 in {/di\.uoa\.gr/};
l2 in {/di\.uoa\.gr/};
n3: PERLCOND 'n3.format=~/image/';
n3 in {/di\.uoa\.gr/}
Using ISEARCHd –d 5 –l 1000
```

Το W3QS μπορεί να συμπληρώσει φόρμες αυτόματα με τις κατάλληλες οδηγίες.

Όψεις

Οι όψεις επιτρέπουν στο W3QS να έχουμε ενημερωμένα αποτελέσματα. Για παράδειγμα για να έχουμε μία λίστα με sites για ταινίες χρησιμοποιώντας το Yahoo Entertainment Cool Sites List μπορούμε να γράψουμε:

```
Select
From n1,l1,n2
Where
n1 in {http://www.yahoo.com/Entertainment/Cool_Links/Entertainment/};
n2: PERLCOND '(n2.content=~/movie/i) || (n2.content=~/cinema/i)'
Using ISEARCHd
Evaluated every day
```

Η παραπάνω όψη ενημερώνεται κάθε 24 ώρες.

Τα κύρια χαρακτηριστικά της W3QL είναι:

- Τύπος γλώσσας, SQL.
- Τα ερωτήματα αφορούν και την δομή και το περιεχόμενο.
- Επεκτάσιμη.
- Παρέχει όψεις για μέρη του WWW.

Από το WWW πρέπει να παίρνουμε τα ακόλουθα χαρακτηριστικά για να βρίσκουμε πληροφορίες:

- Το περιεχόμενο των πληροφοριών
- Δυνατότητα για αναζήτηση στην δομή του γράφου των hypertext.
- Πρόσβαση σε service providers μέσω html forms.

Θεωρούμε το WWW σαν έναν κατευθυντικό γράφο. Η τοπολογία του γράφου είναι άγνωστη, αλλά μπορούμε να συμπεράνουμε μερικά πράγματα από την πλοήγηση στο WWW. Οι κόμβοι και οι ακμές του γράφου ορίζονται από κάθε δυνατή δραστηριότητα πλοήγησης στο WWW.

Γλώσσα W3QL

Αρχές

- Ερωτήματα περιεχομένου: Αυτά τα ερωτήματα επιλέγουν κόμβους από hypertext με βάση το περιεχόμενό τους. Ένας κόμβος πρέπει να ικανοποιεί τις συνθήκες που περιέχονται στο ερώτημα, ώστε να επιλεχθεί.

- Ερωτήματα δομής: Επιλέγονται ένα σύνολο κόμβων και ακμών από την δομή των hypertext που ικανοποιούν το υπόδειγμα του γράφου που υπάρχει στο ερώτημα.

Υπάρχει η δυνατότητα συνδυασμό των παραπάνω για πιο πολύπλοκα ερωτήματα.

Ερωτήματα περιεχομένου

Οι πληροφορίες που βρίσκονται στο WWW είναι κυρίως σε μη δομημένα αρχεία σε αντίθεση με αυτό που έχουμε στις βάσεις. Για να μπορέσουμε όμως να βρούμε αυτό που θέλουμε, κάνουμε ερωτήσεις στα αρχεία που έχουν τις πληροφορίες. Τα αρχεία στο WWW περιέχουν συνήθως μετα-πληροφορίες που κωδικοποιούνται με συγκεκριμένες εντολές, για παράδειγμα <title> στην Html. Φυσικά είναι δύσκολο, να ορίσουμε εξαρχής μια γλώσσα για τον κόμβο-περιεχόμενο για κάθε φόρμα αρχείων. Η λύση που προσφέρει η W3QL είναι να συμπεριφέρεται στις συνθήκες περιεχομένου σαν οδηγίες εκτέλεσης και αναλαμβάνουν εξωτερικά προγράμματα την εκτέλεση. Οι συνθήκες περιεχομένου ορίζονται με τον καθορισμό προγραμμάτων για την διαχείριση των αρχείων.
Για παράδειγμα η συνθήκη,

node_format eq “Doc” && node_author eq “John”

Θα βρει όλα τα αρχεία doc με συγγραφέα τον John

Ερωτήματα δομής

Ερωτήματα συγκεκριμένης δομής στην W3QL είναι ένας κατευθυνόμενος γράφος, στον οποίο οι κόμβοι και οι ακμές ακολουθούν τις συνθήκες. Οι συνθήκες είναι αυτές

από το ερώτημα για το περιεχόμενο. Η απάντηση σε ένα ερώτημα δομής είναι ένα σύνολο υπο-γράφων του WWW όπου:
Κάθε WWW γράφος πρέπει να είναι παρόμοιος με τον γράφο ερωτήματος. Η τοπολογία ενός γράφου εκφράζεται από ένα σύνολο ορισμού μονοπατιών. Δηλαδή n1,l1,n2,l2,n3 ορίζει ένα μονοπάτι τριών κόμβων από τον n1 στον n2 μέσω της ακμής l1και από τον n2 στον n3 μέσω της ακμής l2.

WebLog

Στην γλώσσα επερωτήσεων WebLog, η σύνταξη και η σημασιολογία στηρίζεται στην λογική. Σε αυτές τις γλώσσες, υπάρχει μια απλή αντίληψη του σχήματος και οι σελίδες θεωρείται ότι είναι μέσα σε ένα τύπο, για παράδειγμα, σαν κόμβοι σε ένα γράφο όπου οι περισσότεροι έχουν συγκεκριμένο σύνολο γνωρισμάτων. Διαφοροποιείται από τις παραπάνω γλώσσες, στο ότι χρησιμοποιεί συμπερασματικούς κανόνες και όχι SQL.
Θα μπορούσαμε να πούμε ότι η WebLog είναι σαν την Datalog γλώσσα για την επερώτηση και την αναδιάρθρωση του WWW. Η WebLog είναι κατάλληλη για

α) επερωτήσεις και αναδόμηση των πληροφοριών του Web
b) εξαγωγή μερικών γνώσεων, τις οποίες έχουν οι χ΄ρηστες στις επερωτήσεις τους.
c) επερωτήσεις σε δυναμικές πληροφορίες στο web.

Οι σημερινές μηχανές αναζήτησης, έχουν τα εξής προβλήματα:

- Μερικές γνώσεις που ο χρήστης μπορεί να αναζητάει, δεν εξάγονται από την σελίδα.
- Η ικανότητα της αναδόμησης είναι ελάχιστη ή μηδαμινή.
- Δεν υπάρχει δυνατότητα για αξιοποίηση εξωτερικών διαθέσιμων βιβλιοθηκών συναρτήσεων επεξεργασίας συμβολοσειρών και εγγράφων.
- Η ποιότητα των αποτελεσμάτων είναι συμβιβαστική σε σχέση με την δυναμική κατάσταση των εγγράφων στο Web.

Για αυτό αναζητήθηκε η λύση μιας γλώσσας επερώτησης, όπως η WebLog.
Μερικά από τα σημαντικά σημεία της είναι:

- Παρέχει ένα δηλωτικό interface τόσο για επερωτήσεις όσο και για αναδόμηση.
- Διανέμει μερική γνώση, που ο χρήστης μπορεί να αναζητάει.
- Αναγνωρίζει την δυναμική φύση των πληροφοριών στο Web.

Σύνταξη

Χρησιμοποιούμε συμβολοσειρές με μικρό το πρώτο γράμμα για να δηλώσουμε σταθερές και κεφαλαίο για μεταβλητή. Το λεξικό της WebLog αποτελείται από ζευγάρια αριθμήσιμων συνόλων G (σύμβολα συναρτήσεων), S (σύμβολα μη-συναρτήσεων), V (μεταβλητές) και οι συνήθης λογικοί σύνδεσμοι $\neg, \vee, \wedge, \exists, \forall$. Κάθε

σύμβολο στο $S \cup V$ είναι όρος της γλώσσας. Αν $f \in G$ τότε είναι μια n-συνάρτηση και $t_1, \ldots, t_n$ είναι όροι, τότε $f(t_1, \ldots, t_n)$ είναι όρος.

Ένας ατομικός τύπος στην WebLog είναι η έκφραση ενός από τους ακόλουθους όρους:
<url>[<rid>:<attr>->><val>]
<url>[<rid>:<attr>-><val>]
<url>[<attr>]
<url>

Όπου <url>,<rid>,<attr> και <val> είναι όροι της WebLog.

Παράδειγμα

http://www.com[X:title->'Web΄,hlink->>L, occurs ->>'example΄]

Σε αυτό το παράδειγμα, η url διεύθυνση είναι ο όρος url, οι όροι title, hlink και occurs είναι attr όροι, X είναι ο όρος rid και οι υπόλοιποι όροι είναι οι val όροι.

Σημασιολογία

Hyperlink Navigation and Searching Ttitles

Ένα από τα πρωτότυπα χαρακτηριστικά της WebLog είναι ότι συμπεριφέρεται στα hyperlinks σαν "πρώτης τάξης πολίτες". Αυτό επιτρέπει την πλοήγηση σε html έγγραφα χρησιμοποιώντας ένα WebLog πρόγραμμα.

Παράδειγμα

Θέλουμε να επιλέξουμε όλες τις δημοσιεύσεις που αναφέρονται σε html έγγραφα και εμφανίζονται στην σελίδα Database Systems & Llogical Programming. Επίσης θέλουμε αυτή η συλλογή, να περιέχει τον τίτλο του αρχείου και τον συγγραφέα. Η εντολή είναι η ακόλουθη:

Ans.html[title-> 'all citations',hlinks->>L, occurs->>T]
<- leyurl[hlink->>L],hreff(L,U),U[title->T].

Leyurl είναι η διεύθυνση της σελίδας Database Systems & Llogical Programming.
Η μεταβλητή L στην πρώτη επίτευξη μεταβάλετε σε όλα τα hyperlinks στο leyurl. Το ενσωματωμένο κατηγόρημα href χρησιμοποιείται για πλοήγηση στις δημοσιεύσεις στην σελίδα στο leyurl. Ο κανόνας δημιουργεί μια νέα σελίδα html ans.html που είναι συλλογή όλων των δημοσιεύσεων στο leyurl.

Querying Keywords in Documents

Ley server είναι η συλλογή των εγγράφων από το leyurl. Αυτά τα έγγραφα έχουν την ιδιότητα να:

- Μπορούν να τα φτάσουν από links πλοήγησης.
- Τα url τους θα έχουν το πρόθεμα το leyurl.

Lorel

Η γλώσσα Lorel σχεδιάστηκε για επερωτήσεις σε ημιδομημένα δεδομένα. Τα ημιδομημένα δεδομένα γίνονται όλο και πιο διαδεδομένα, για παράδειγμα σε δομημένα έγγραφα όπως html, όταν προκαλούμε ένταξη δεδομένων από πολλαπλές πηγές. Τα παραδοσιακά μοντέλα δεδομένων και γλώσσες επερωτήσεων, δεν είναι κατάλληλα αφού τα ημι-δομημένα δεδομένα δεν έχουν κανονική μορφή, μερικά δεδομένα λείπουν, παρόμοιες έννοιες παρουσιάζονται με διαφορετικούς τύπους, υπάρχουν ετερογενή σύνολα. Η Lorel είναι μια φιλική γλώσσα στο στιλ της SQL/OQL για επερωτήσεις τέτοιων δεδομένων. Για λόγους ευρεία συμβατότητας το απλό μοντέλο αντικειμένου μπορούμε να το βλέπουμε σαν μια επέκταση του μοντέλου δεδομένων ODMG και την Lorel γλώσσα σαν επέκταση της OQL

Οι κύριες καινοτομίες της Lorel είναι:

1. Η απαλλαγή του χρήστη από την αυτηστή χρήση εντολών της OQL που είναι άστοχη για ημι-δομημένα δεδομένα.
2. Ισχυρές εκφράσεις μονοπατιού που επιτρέπουν προσαρμόσιμο σχήμα δηλωτικής πλοήγησης πρόσβασης και είναι μερικώς διαθέσιμα όταν οι πληροφορίες της δομής δεν είναι γνωστές στον χρήστη.

Lorel είναι η υλοποίηση της γλώσσας επερώτησης συστήματος διαχείρισης βάσεων Lore. Το Web δεν περιορίζει την δομή των html αρχείων. Όταν κάνουμε επερωτήσεις σε ημι-δομημένα δεδομένα, δεν μπορούμε να ξέρουμε την πλήρη δομή, ειδικά αν έχουμε δυναμική δομή. Για αυτό είναι απαραίτητο, να μην απαιτείται πλήρη γνώση της δομής για να κάνουμε επερωτήσεις.

Παράδειγμα

Να βρεθούν τα ονόματα και οι ταχυδρομικοί κώδικες, από όλα τα φθηνά εστιατόρια.

Δεν γνωρίζουμε αν ο ταχυδρομικώς κώδικας είναι μέρος της διεύθυνσης αλλά μπορεί από μόνο του να είναι ένα αντικείμενο του εστιατορίου. Επίσης δεν ξέρουμε αν η συμβολοσειρά φθηνά θα είναι μέρος της κατηγορίας, τιμής, περιγραφής ή άλλου αντικειμένου του εστιατορίου. Μπορούμε να κάνουμε το επερώτημα στην Lorel ως εξής:

```
select Guide.restaurant.name,
Guide.restaurant(.address)?.zipcode
where Guide.restaurant.% grep "φθηνό"
```

Το ? μετά την .address, σημαίνει ότι η διεύθυνση είναι προαιρετική στο στην έκφραση του μονοπατιού. Ο χαρακτήρας % θα ταιριάζει οποιαδήποτε αντικείμενο του εστιατορίου και ο χειριστής grep θα επιστρέψει true αν υπάρχει η συμβολοσειρά φθηνά σε κάποιο αντικείμενο. Το μοντέλο δεδομένων που υπάρχει στην Lorel λέγεται OEM (Object Exchange Model). Μια βάση που προσαρμόζεται σε OEM μπορούμε να την φανταστούμε ως έναν γράφο με πολύπλοκες τιμές στους εσωτερικούς κόμβους, ατομικές τιμές στους κόμβους-φύλλα και ακμές με ετικέτες.

Για να ορίσουμε την σημασιολογία της Lorel πάνω σε μια OEM βάση με όρους OQL και ODMG, προσθέτουμε έναν νέο τύπο στο ODMG μοντέλο για την αναπαράσταση OEM αντικειμένων. Τότε ένα σημαντικό θέμα στην Lorel είναι η επέκταση της ισότητας στην OQL για να διαχειρίζεται OEM αντικείμενα.

Object Exchange Model

Το Object Exchange Model είναι ένα μοντέλο δεδομένων για την αναπαράσταση των ημι-δομημένων δεδομένων. Τα δεδομένα που αναπαριστώνται στο OEM θεωρούνται γράφος. Στο μοντέλο δεδομένων OEM όλες οι οντότητες είναι αντικείμενα. Κάθε αντικείμενο έχει μοναδικό αναγνωριστικό αριθμό. Μερικά αντικείμενα είναι ατομικά και περιέχουν και περιέχουν μια τιμή από βασικούς τύπου δεδομένων π.χ. ακέραιος, πραγματικός και λοιπά. Όλα τα άλλα αντικείμενα είναι πολύπλοκα, η τιμή τους είναι ένα σύνολο από αναφορές σε αντικείμενα, δηλωμένα σαν σύνολο από ζευγάρια (label, object identifier).

Απλές εκφράσεις μονοπατιών

Όταν γίνονται επερωτήσεις σε ημι-δομημένα δεδομένα, ειδικά όταν δεν είναι γνωστή η δομή, είναι συνετό να χρησιμοποιείται σχήμα από επερώτηση "πλοήγησης" που στηρίζεται στις εκφράσεις των μονοπατιών. Η ιδέα είναι να οριστούν μονοπάτια στο OEM γράφο που στηρίζονται στην σειρά των ετικετών στις ακμές. Οι απλές εκφράσεις μονοπατιών, επιτρέπουν στους χρήστες να βρουν το σύνολο των αντικειμένων που είναι προσβάσιμα, ακολουθώντας μια σειρά από ετικέτες ξεκινώντας από ένα αντικείμενο στο γράφο του OEM. Μια απλή έκφραση μονοπατιού, είναι η σειρά Z.l1…l2 όπου l1…l2 είναι ετικέτες και το Z είναι το όνομα ενός αντικειμένου ή μια μεταβλητή που δηλώνει ένα αντικείμενο. Ένα μονοπάτι δεδομένων είναι μια σειρά $o_0, l_1, o_1, \ldots, l_n, o_n$ όπου o_i είναι αντικείμενα και για κάθε i υπάρχει μία ακμή με το όνομα l_i μεταξύ του o_{i-1} και o_i. Ξεκινώντας από ένα αντικείμενο $Z = o_0$ μπορούν να υπάρχουν αρκετά μονοπάτια δεδομένων που ταιριάζουν στην απλή έκφραση μονοπατιού $Z.l_1 \ldots l_2$.

Για παράδειγμα υποθέτουμε ένα αντικείμενο με το όνομα Guide και την απλή έκφραση του μονοπατιού Guide.A.B.C. Αυτό το μονοπάτι μπορεί να ερμηνευτεί με πλοήγηση ως εξής: Εκκίνηση από το αντικείμενο Guide, μετά την ακμή A, μετά την ακμή B, και τελικά την ακμή C. Αφού υπάρχουν πολλές πιθανές ακμές με το όνομα A,B,C η έκφραση του μονοπατιού μπορεί να ταυτιστεί με έναν αριθμό μονοπατιών στο OEM γράφο. Εναλλακτικά μπορούμε να ερμηνεύσουμε αυτή την έκφραση μονοπατιού χρησιμοποιώντας OQL-στιλ object-component αναφορά: Guide:A ορίζει ένα σύνολο από αντικείμενα R με μια ακμή A από το Guide στο R, Guide.A.B ορίζει τα αντικείμενα Z τα οποία για μερικά R στο Guide:A υπάρχει ακμή B από το R στο Z, παρόμοια για το Guide.A.B.C.

Lorel ερώτημα

```
select Z
from Guide.restaurant.zipcode Z
```

OQL ερώτημα

select Z
from Guide.restaurant R, R.zipcode Z

Τα ερωτήματα στην Lorel δεν χρειάζεται να έχουν from, δηλαδή το ερώτημα

select Guide.restaurant.name
where Gguide.restaurant.category=”gourmet”

γίνεται σε

select Guide.restaurant.name
from Guide.restaurant
where Gguide.restaurant.category=”gourmet”

Γενικές εκφράσεις μονοπατιών

Οι συνήθης εκφράσεις μονοπατιών είναι ένας ισχυρός μηχανισμός για την εύρεση αντικειμένων στην βάση.

Παραδείγματα:
Guide.restaurant(.address)?.zipcode

Αυτή η έκφραση δηλώνει ότι ξεκινάει από το Guide, συνεχίζει μια ακμή του restaurant, μετά το zipcode και ενδιάμεσα προαιρετικά η διεύθυνση.

Guide.restaurant.#@P.comp%.name

Αυτή η έκφραση, αγνοώντας το @P ξεκινάει από το Guide, συνεχίζει με ακμή του restaurant, ακολουθεί ένας αυθαίρετος αριθμός ακμών με μη προσδιορισμένο όνομα (σύμβολο #), ακολουθεί ακμή που η ετικέτα της ξεκινάει με “comp” (comp%) και τελικά τερματίζει με μια ακμή name. Η μεταβλητή του μονοπατιού P έχει ως όριο το @P σε κάθε μονοπάτι δεδομένων.

Guide.restaurant(.nearby)*{R}.name

Αυτή η έκφραση, αγνοώντας τον όρο {R}, ξεκινάει από το Guide, συνεχίζει με ακμή του restaurant, ακολουθεί ένας αυθαίρετος αριθμός κοντινών ακμών και τελικά τερματίζει με μια ακμή name. Ο όρος {R} είναι πολύ χρήσιμος γιατί είναι ένας συνεκτικός τρόπος για την προσθήκη μεταβλητών σε αντικείμενα στην μέση μεγάλων μονοπατιών.

Παράδειγμα

Αν θέλουμε να βρούμε τα ονόματα όλων των εστιατορίων με ταχυδρομικό κωδικό 11573 είτε στην διεύθυνσή του είτε ως πεδίο στο εστιατόριο, τότε έχουμε:

```
select Guide.restaurant.name
where
      Guide.restaurant(.address)?.zipcode=11573
```

Είναι σημαντικό να αναφερθεί ότι ενώ απλές εκφράσεις μονοπατιών μπορούν πάντα να μεταφραστούν σε OQL, δεν μπορεί να γίνει το ίδιο και με τις γενικές εκφράσεις μονοπατιών.

Κατασκευή αποτελεσμάτων

Ένα ερώτημα select-from-where στην Lorel έχει την ίδια σημασιολογία με ένα ερώτημα select-from-where της SQL ή OQL επιστρέφει ένα σύνολο αποτελεσμάτων. Στην Lorel το αποτέλεσμα είναι πάντα μια συλλογή OEM αντικειμένων και υπάρχει περιορισμός της ύπαρξης δύο ίδιων αντικειμένων από το object identifier. Για ένα "top-level" ερώτημα (για παράδειγμα ερώτημα που δεν έχει ενσωματωμένο υπο-ερώτημα), η τελική συλλογή είναι πακέτο σε ένα μόνο OEM αντικείμενο.

Παράδειγμα

```
select X
from Guide.restaurant X
```

Απάντηση:

```
Answer &155
   restaurant &19
   restaurant &35
   restaurant &77
```

Παρατηρούμε ότι μόνο το 155 είναι νέο αντικείμενο. Το αποτέλεσμα του ερωτήματος μπορεί να χρησιμοποιηθεί αργότερα, σε άλλα ερωτήματα .

WebOQL

Το σύστημα WebOQL συνδυάζει αρχιτεκτονική, μοντέλο δεδομένων, και γλώσσα επερώτησης ώστε να παίρνουμε πληροφορίες από δομημένα έγγραφα χωρίς να χρειαζόμαστε την βοήθεια εξωτερικών προγραμμάτων. Το μοντέλο δεδομένων WebOQL υποστηρίζει τις απαραίτητες ενέργειες για εύκολη μοντελοποίηση record-based δεδομένων, δομημένων εγγράφων και hypertexts. Η γλώσσα επερώτησης μας επιτρέπει να αναδομήσουμε τις 3 περιπτώσεις που αναφέραμε. Η WebOQL συνθέτει ιδέες από γλώσσες επερωτήσεων για το web, για ημιδομημένα δεδομένα, για αναδόμηση website και κάνει σημαντικές συνεισφορές, ειδικά η ιδέα επερώτησης εγγράφων με την διαχείριση συντακτικών δέντρων και την υποστήριξη του web ως τύπο δεδομένων.

Στην WebOQL ένα συντακτικό δέντρο για κάθε έγγραφο ίδιας οικογένειας (π.χ. html) δημιουργείται από τον ίδιο wrapper, οποιαδήποτε και αν είναι η δομή του εγγράφου. Η γλώσσα είναι αρκετά ισχυρή για να κάνει επερώτηση ή να αλλάξει την δομή σε αυτά τα δέντρα με αρκετούς τρόπους.

Web Queries

Με μια web γλώσσα επερωτήσεων, όπως η WebSQL, W3QS, WebLog, μοιραζόμαστε την ιδέα της εμφάνισης του web σαν μια βάση που μπορούμε να κάνουμε επερωτήσεις. Αυτές οι γλώσσες όμως έχουν ένα μεγάλο περιορισμό, έλλειψη αξιοποίησης της δομής των εγγράφων. Της WebOQL το υπόδειγμα πλοήγησης είναι η γενίκευση της WebSQL γενικές εκφράσεις μονοπατιών.

Semistructured Data

Το μόνο εμπόδιο στην αξιοποίηση της εσωτερικής δομής των Web εγγράφων είναι η έλλειψη σχήματος ή τύπου και η πιθανή έλλειψη κανονικής μορφής που μπορεί να προκύψει. Το πρόβλημα της επερώτησης δεδομένων που η δομή τους είναι άγνωστη ή μη κανονική έχει αναφερθεί ξανά, και καλείται γλώσσα επερώτησης ημιδομημένων δεδομένων. Αυτά τα συστήματα χρησιμοποιούν πολύ χαμηλού επιπέδου αναπαράσταση δεδομένων, που στηρίζεται σε γράφους.

Ένα πρόβλημα με μοντέλα ημιδομημένων δεδομένων είναι ότι παρέχουν κυρίως labeled graphs. Δεν υποστηρίζουν ordered collections. Η υποστήριξη της order είναι ένα στοιχείο-κλειδί για την μοντελοποίηση δομημένων εγγράφων, αναφορικά, μας επιτρέπει να μοντελοποιούμε hyperlinks μεταξύ εγγράφων χρησιμοποιώντας εγγραφές που μπορούμε εύκολα να αναπαραστήσουμε σχεσιακούς πίνακες χωρίς να επινοήσουμε encodings για να τα προσομοιώσουμε.

Hypertrees

Η κύρια δομή δεδομένων που παρέχεται από την WebOQL είναι η hypertree. Hypertrees είναι κατευθυνόμενα δένδρα με labels στα βέλη. Υπάρχουν δύο ειδών βέλη, τα εσωτερικά και τα εξωτερικά. Τα εσωτερικά βέλη χρησιμοποιούνται για την αναπαράσταση δομημένων αντικειμένων και τα εξωτερικά για την αναπαράσταση hyperlinks μεταξύ των αντικειμένων. Στα διαγράμματα, χρησιμοποιούμε πλήρης γραμμές για τα εσωτερικά βέλη και διακεκομμένες γραμμές για τα εξωτερικά βέλη. Τα εξωτερικά βέλη, δεν μπορούν να έχουν απογόνους και η εγγραφή που έχουν ως label πρέπει να είναι ένα Url. Urls είναι συμβολοσειρές. Τα Hypertrees είναι πολύ χρήσιμες δομές δεδομένων επειδή ταξινομούν τις τρεις ιδέες που θέλουμε να υποστηρίξουμε: collections, nesting και ordering. Με την διαφοροποίηση μεταξύ εσωτερικών και εξωτερικών βελών, η αντίληψη της αναφοράς φαίνεται από τα δέντρα και το γεγονός ότι οι labels είναι εγγραφές, μας επιτρέπουν, να αναπαραστήσουμε τις συλλογές των εγγραφών. Όταν μοντελοποιούμε πληροφορίες που βρίσκονται στο Web, ένα hypertree θα αντιστοιχήσει σε ένα έγγραφο. Όμως ένα hypertree μπορεί να αναπαραστήσει και ένα σχεσιακό πίνακα, μια ιεραρχία φακέλων και λοιπά.

Webs

Παρόλο που τα hypertrees είναι το κλειδί στο WebOQL, υποστηρίζεται και ένα υψηλότερο επίπεδο αφαίρεσης που μας επιτρέπει να μοντελοποιήσουμε σύνολα συσχετισμένων hypertrees, το web. Ένα web είναι ένα ζευγάρι (t,F) που αποτελείται από ένα hypertree t και μια συνάρτηση F που αντιστοιχεί Urls σε hypertrees. Αναφερόμαστε σε αυτά τα δύο components σαν σχήμα και browsing fuction του web, αντίστοιχα. Επίσης το ζευγάρι που αποτελείται από ένα Url u και το hypertree F(u) είναι σελίδα στο web και επίσης λέμε F(u) είναι το περιεχόμενο της σελίδας.
Η συνάρτηση αναζήτησης (browsing fuction) ορίζει ένα γράφο όπου οι κόμβοι είναι σελίδες και υπάρχει βέλος μεταξύ του κόμβου a και του κόμβου b αν το περιεχόμενο της σελίδας στον κόμβο a περιέχει ένα εξωτερικό βέλος του οποίου το Url, είναι το Url της σελίδας στον κόμβο b. Ένα web μπορεί να χρησιμοποιηθεί για να μοντελοποιήσουμε ένα σύνολο από συσχετιζόμενες σελίδες. Τόσο τα hypertrees ¨οσο και το web μπορούν να τα διαχειριστούμε χρησιμοποιώντας WebOQL.

Η γλώσσα

Η κύρια κατασκευή που παρέχετε από την WebOQL είναι η οικειότητα με **select-from-where**. Ας δούμε ένα παράδειγμα από την χρήση του. Αν υποθέσουμε ότι το όνομα *csPapers* ορίζει την βάση των papers και θέλουμε να εξάγουμε τον τίτλο και το Url της πλήρης έκδοσης των papers με συγγραφέα τον "Smith". Το ακόλουθο είναι το ερώτημα:

***Q1*: select** [*y.Title*, *y'.Url*]
from *x* **in** *csPapers*, *y* **in** *x'*
where *y.Authors* ~ "Smith"

Στο ερώτημα, το x επαναλαμβάνει στα απλά δέντρα της *csPapers* και δίνοντας μια τιμή στο x, το y επαναλαμβάνει στα απλά δέντρα του μοναδικού υποδέντρου του x. Το quote είναι το σύμβολο για το prime operator, το οποίο επιστρέφει το πρώτο υποδέντρο από το όρισμά του. Η τελεία είναι το σύμβολο για το peek operator, το οποίο εξάγει ένα πεδίο από την εγγραφή με label το πρώτο απερχόμενο βέλος από το όρισμά του. Οι αγκύλες δηλώνουν το *Hang* operator, που δημιουργεί ένα arc labeled με μια εγγραφή με τα ορίσματα. Τέλος η περισπωμένη αναπαριστά το ταίριασμα του πρότυπου συμβολοσειράς.

Η απάντηση σε ένα select**-from-where** ερώτημα αποκτάτε ως εξής: για κάθε μεταβλητή στην ενότητα **from**, γίνεται έλεγχος της συνθήκης στην ενότητα **where**, αν είναι αλήθεια, αξιολογείτε το ερώτημα στο **select** και προστίθεται το αποτέλεσμα στην απάντηση.

Η WebOQL παρόλο που διατηρεί όλες τις εξερευνητικές δυνατότητες για εσωτερικά έγγραφα, έχει πολύ ισχυρά χαρατηριστικά εξαγωγής δεδομένων από εσωτερικά έγγραφα. Αυτό το κάνει πολύ καλό για δημιουργία εφαρμογών που παραλαμβάνουν και ενώνουν πληροφορίες από υπάρχον web sites.

Strudel

Η γλώσσα αναζήτησης για το σύστημα διαχείρισης ιστοσελίδας STRUDEL είναι η STRUQL. Παρόλα αυτά η STRUQL αναπτύχθηκε στο περιβάλλον μιας συγκεκριμένης διαδικτυακής εφαρμογής. Είναι μια γενικού σκοπού γλώσσα αναζήτησης βασισμένη σε ένα μοντέλο δεδομένων από μαρκαρισμένους κατευθυντικούς γράφους. Επιπρόσθετα το μοντέλο δεδομένων STRUDEL περιέχει ονομασμένες συλλογές και υποστηρίζει διάφορους ατομικούς τύπους, όπου συχνά εμφανίζονται σε ιστοσελίδες, όπως URLs, Postscript, κείμενα, εικόνες και HTML αρχεία. Το αποτέλεσμα από μια αναζήτηση STRUQL είναι ένας γράφος με ίδιο μοντέλο δεδομένων όπως και οι γράφοι εισόδου. Στο σύστημα STRUDEL η STRUQL χρησιμοποιήθηκε για δύο εφαρμογές:

1) Αναζήτηση ετερογενών πηγών για να τις μετατρέψει σε μια ιστοσελίδα που είναι γράφος δεδομένων
2) Αναζήτηση σε αυτό το γράφο για να παράξει μια ιστοσελίδα γράφο.

Μια STRUQL αναζήτηση είναι ένα σύνολο από δυνατά εμφωλιασμένα μπλοκς, το καθένα της μορφής:

[Where C1,…..,Ck]
[Create N1,…..,Nn]
[Link L1,.....,Lp]
[Collect G1,…..,Gq]

Η where πρόταση μπορεί να περιέχει είτε συνθήκες μελών είτε συνθήκες ζευγαριών από κόμβους που εκφράζονται με κανονικές εκφράσεις. Η where έκφραση παράγει όλους τους σύνδεσμους ενός κόμβου και και το arc ποικίλει ανάλογα με τις τιμές στο γράφο εισόδου. Οι υπόλοιπες προτάσεις χρησιμοποιούν Skolem συναρτήσεις για να κατασκευάσουν ένα νέο γράφο από αυτούς τους συνδέσμους.
Παρακάτω αναφέρουμε STRUQL με μια αναζήτηση καθορίζοντας μια ιστοσελίδα ξεκινώντας από ένα αρχείο βιβλιογραφίας που είναι καθορισμένος γράφος. Η ιστοσελίδα θα αποτελείται από τρία είδη σελίδων: μια παρουσίαση δημοσίευσης για κάθε είσοδο βιβλιογραφίας, μια σελίδα χρόνου έκδοσης και μια κεντρική σελίδα, η οποία θα δείχνει σε όλες τις δημοσιεύσεις, όπου έχουν δημοσιευθεί σε όλα αυτά τα χρόνια.

// Create Root
create RootPage()
// Create a presentation for every publication x
where Publications(x), x->l->v
create PaperPresentation(x)
link PaperPresentation(x) -> l -> v
{ // Create a page for every year
where l = "year"
create YearPage(v)
link
YearPage(v) -> "Year" -> v

```
YearPage(v)->"Paper"->PaperPresentation(x),
// Link root page to each year page
RootPage() -> "YearPage" -> YearPage(v)
}
```

Στην where πρόταση, το Publications(x) σημαίνει ότι το χ ανήκει σε μια συλλογή από δημοσιεύσεις και το atomx ! l ! v δηλώνει ότι υπάρχει ένας σύνδεσμος μέσα στο γράφο από x to v και η ετικέτα στο τόξο είναι l. Η ίδια παρατήρηση συμβαίνει και στη πρόταση του link όπου αναφέρει πρόσφατες δημιουργημένα κορυφές στο γράφο αποτελέσματος. Αφού δημιουργήσει τη κεντρική σελίδα, το CREATE παράγει μια σελίδα για κάθε δημοσίευση. Το δεύτερο CREATE, εμφωλιασμένο μέσα στην εξωτερική αναζήτηση παράγει τη σελίδα χρόνου για κάθε χρόνο και τη συνδέει με τι κεντρική σελίδα και τις σελίδες που παρουσιάζουν τις δημοσιεύσεις, που έγιναν εκείνο το χρόνο. Να επισημάνουμε εδώ πως η σελίδα χρόνου εγγυάται ότι κάθε σελίδα χρόνου για συγκεκριμένο χρόνο δημιουργείται μόνο μια φορά, άσχετα πόσες δημοσιεύσεις είχαν δημοσιευτεί σε αυτό το χρόνο.

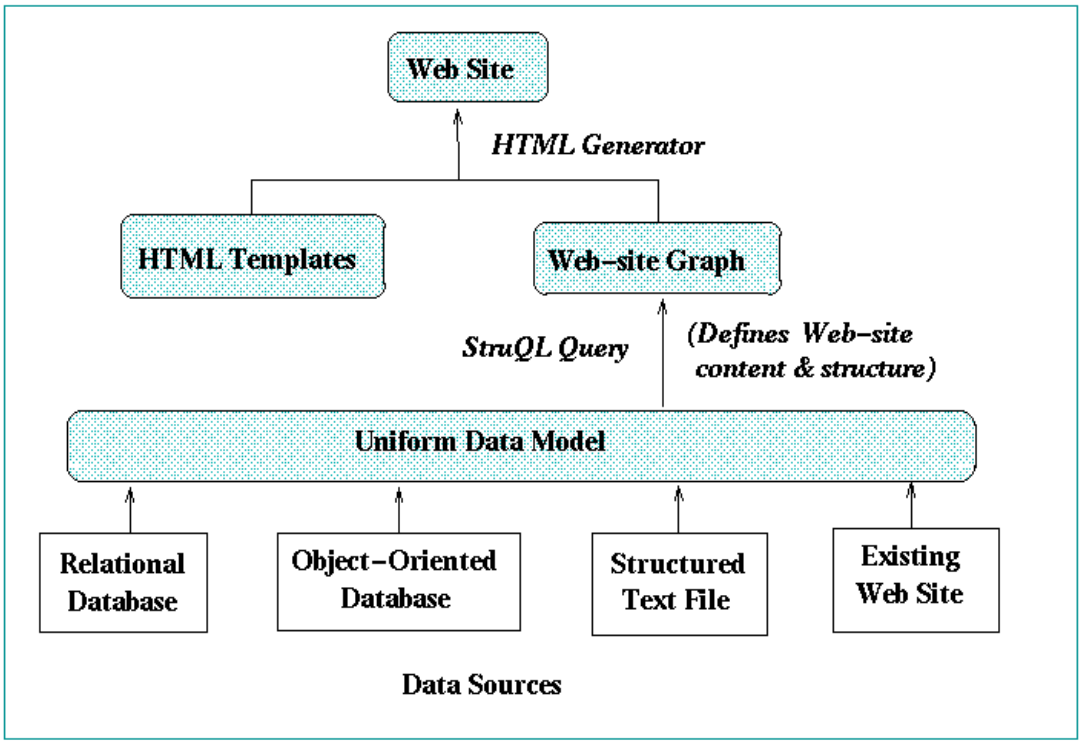
Χαρακτηριστικά

- Ενοποιεί δεδομένα από διαφορετικές πηγές
- Υψηλού επιπέδου δηλωτική γλώσσα για διαχείριση δομής σελίδας (Struql)

Πλεονεκτήματα

- Παράγει πολλαπλές σελίδες από τα ίδια δεδομένα
- Υποστηρίζει την έυκολη αναδόμηση και αναδιαμόρφωση
- Παρέχει πλατφόρμα για:
  1) ενδυνάμωση της ακεραιότητας των περιορισμών
  2) σχεδίαση πολιτικής για αποτελεσματικό χρόνο εκτέλεσης διαχείρισης των σελίδων

Η αρχιτεκτονική αυτού του συστήματος διαχείρισης σελίδας STRUDEL φαίνεται στο παρακάτω διάγραμμα:



Η Strudel είναι βασισμένη πάνω σε ένα μοντέλο ημιδομημένων δεδομένων: Καθορισμένων κατευθυνόμενω γράφων

- οι κόμβοι στους γράφους παρουσιάζουν τα αντικείμενα
- Οι ετικέτες πάνω στα τόξα αναπαριστούν τα ονόματα των γνωρισμάτων
- Ονοματιζόμενες συλλογές.

Ποιος ο λόγος για ημιδομημένα δεδομένα;

- Τα ανεπεξέργαστα δεδομένα είναι συνήθως ημιδομημένα
- Εύκολο για ενοποίηση δεδομένων
- web-sites είναι τελικά γράφοι.

Florid

Η Florid είναι μια πρωτότυπη υλοποίηση της F-logic. Για να χρησιμοποιηθεί ως μια μηχανή αναζήτησης του web θα πρέπει ένα web κείμενο να μοντελοποιηθεί με βάση τις δύο παρακάτω κλάσεις:

url::string [get => webdoc]

webdoc::string[url => url; author => string;
modif =>string;
type => string; hrefs@(string) =>> url;
error =>> string]

Η πρώτη δήλωση εισάγει μια κλάση url, υποκλάση ενός string με τη μέθοδο get. Η εντολή get =>webdoc σημαίνει ότι το get είναι μια μέθοδος που επιστρέφει ένα αντικείμενο τύπου webdoc. Η μέθοδος get καθορίζεται από το σύστημα. Το αποτέλεσμα της επίδρασης της εντολής get για ένα url u είναι να ανακτήσει από το web το κείμενο με εκείνο το URL και να το αποθηκεύσει στη τοπική FLORID βάση δεδομένων ως ένα αντικείμενο webdoc, με αντικείμενο αναγνώρισης τη u.get.

Η κλάση webdoc με μεθόδους self, author, modif, type, hrefs kai error μοντελοποιεί τη βασική πληροφορία που είναι κοινή με όλα τα web documents. Η πρόταση hrefs@(string) =>> url σημαίνει ότι η μέθοδος hregs παίρνει ένα string ως είσοδο και επιστρέφει ένα σύνολο αντικειμένων τύπου url. Η ιδέα είανι ότι αν το d είναι ένα webdoc, τότε το d.hreds@(aLabel) επιστρέφει όλα τα urls των κειμένων που δείχνουν σε μέσα στο κείμενο d, από όλα τους συνδέσμους με το όνομα αLavel.

Οι υποκλάσεις κειμένων μπορούν να δηλωθούν ως απαραιτήτες χρησιμοποιώντας την F-logic κληρονομικότητα.

Htmldoc::webdoc[title => string; text =>string]

Για παράδειγμα το παρακάτω πρόγραμμα τραβά από το web το σύνολο όλων των κειμένω που μπορούν να ανακτηθούν άμεσα ή έμμεσα από το url www.cs.toronto.edu από συνδέσμους που περιέχουν το string “database”.

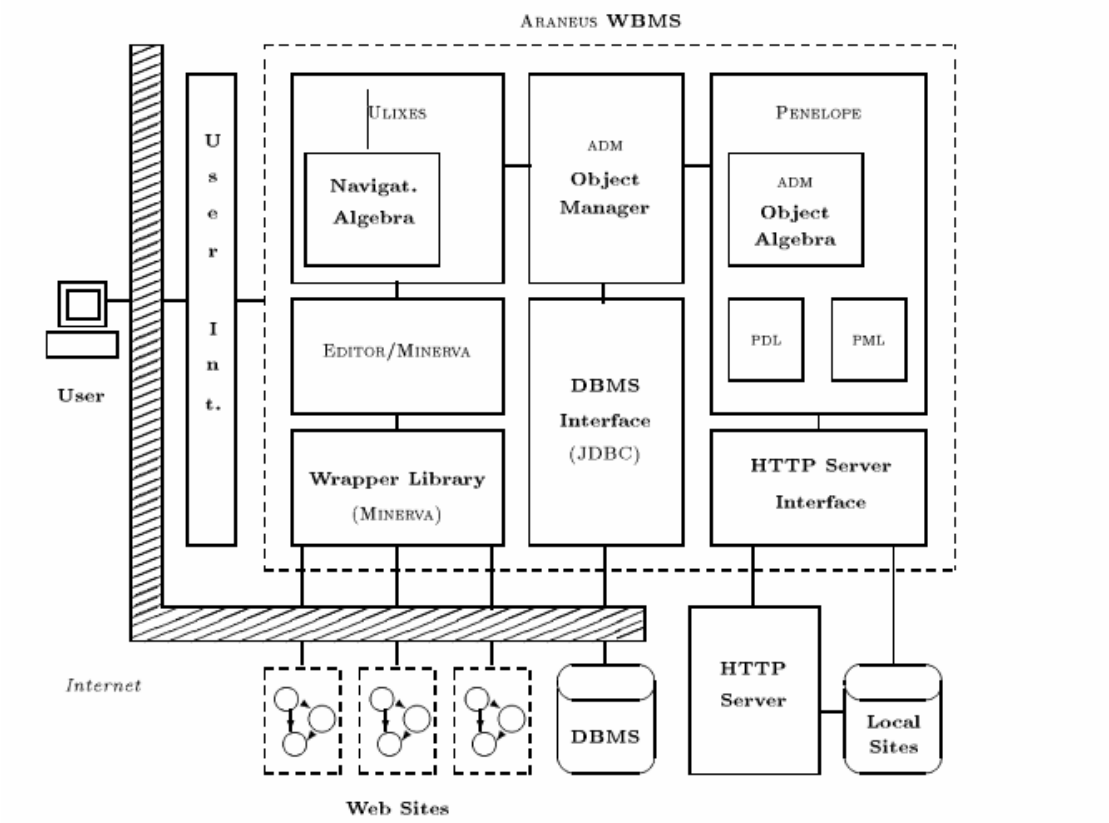
(“www.cs.toronto.edu”:url).get
(Y:url).get <-
(X:url).get [hreds@(L) =>> {Y}]
Substt (“database”,L)

Επιπλέον η FLORID παρέχει μια δυνατή μέθοδο για διαχείριση ημιδομημένων δεδομένων μέσα σε ένα περιβάλλον web. Δεν υποστηρίζει παραυτά τη δομή νέων webs ως αποτέλεσμα μιας αναζήτησης. Το αποτέλεσμα πάντα είναι ένα σύνολο F-logic αντικειμένων αποθηκευμένο στην τοπική βάση δεδομένων.

Araneus Project

Στο project Araneus, η αναζήτηση και η διαδικασία αλλαγής δομής διακρίνεται σε δύο φάσεις. Στη πρώτη φάση η Ulixes γλώσσα χρησιμοποιείται για να κατασκευάσει σχεσιακές όψεις πάνω στο web. Αυτές οι όψεις μπορεί να αναλυθούν και να μετατραπούν χρησιμοποιώντας συγκεκριμένες τεχνικές βάσεων δεδομένων.
Οι Ulixes αναζητήσεις εξάγουν σχεσιακά δεδομένα από στιγμιότυπα σχημάτων σελίδων καθορισμένα στο ADM (Araneus Data Model) model, κάνοντας βαριά χρήση από εκφράσεις μονοπατιών.
Η δεύτερη φάση αποτελείται από παραγόμενες όψεις δεδομένων χρησιμοποιώντας τη Penelope γλώσσα.

Η αρχικτεκτονική αυτού του συστήματος διαχείρισης σελίδας φαίνεται στο παρακάτω διάγραμμα:



Το Araneus εισάγει νέα εργαλεία και τεχνικές για διαχείριση ιστοσελίδας και βάσης δεδομένων. Το σύστημα, που φαίνεται παραπάνω, είναι κατασκευασμένο με java και τρέχει πάνω σε πλατφόρμα java. Το interface του χρήστη είναι ολοκληρωτικά γραμμένα σε γλώσσα HTML, έτσι ώστε οι τελικοί αποδέκτες και διαχειριστές να μπορούν να έχουν στο σύστημα από οποιοδήποτε client στο δίκτυο. Λόγω της ετερογενής φύσης των βάσεων δεδομένων διαδικτύου πολλά μοντέλα δεδομένων αλλά και γλώσσες έχουν χρησιμοποιηθεί.

Από τη δική μας πλευρά, τα δομημένα δεδομένα είναι κυρίως πίνακες βάσεων δεδομένων. Ως εκ τούτου το μοντέλο δεδομένων που έχει χρησιμοποιηθεί για να περιγράψει τα δεδομένα είναι το σχεσιακό μοντέλο και η αντίστοιχη γλώσσα περιγραφής SQL. Η υιοθεσία του σχεσιακού μοντέλου παρέχει τεράστιες δυνατότητες.

Λέμε ένα μοντέλο ότι είναι page-oriented, με την έννοια ότι οι σελίδες έχουν ενα κεντρικό ρόλο. Κάθε σελίδα θεωρείται ως ένα αντικείμενο με αναγνωριστικό (identifier), URL, και ένανα αριθμό γνωρισμάτων. Τα γνωρίσματα μπορεί να είναι είτε ατομικά, όπως string, εικόνες και συνδέσμους σε άλλες σελίδες είτε ομαδικά. Τα ομαδικά γνωρίσματα είναι κυρίως λίστες από μια ακολουθία γνωρισμάτων ίσως και εμφωλιασμένων μεταξύ τους.
Οι σελίδες είναι ομαδοποιημένες σε page-schemes, το οποίο μοιάζει κατά πολύ με το σχεσιακό σχήμα ή τις κλάσεις στις βάσεις δεδομένων, όπως οι συλλογές αντικειμένων από ετερογενείς δομές. Με άλλα λόγια μια ιστοσελίδα είναι μια συλλογή από page-schemes συνδεδεμένα μεταξύ τους με συνδέσμους (links).

Το σύστημα ARANEUS επιτρέπει την αναζήτηση σε ιστοσελίδες που χρησιμοποιούν μια γλώσσα που ονομάζεται ULIXES (αναφέρθηκε παραπάνω). Όμως η αναζήτηση απομακρυσμένων σελίδων τα οποία δεν είναι υπό τη άμεση διαχείριση του συστήματος απαιτεί μια πιο σύνθετη διαδικασία.
Αυτή έχει ως εξής. Μια περιγραφή ADM (Araneus Data Model) της σελίδας θα πρέπει να παραχθεί αναλύοντας το περιεχόμενο της. Στη συνέχεια από το site θα πρέπει να εξάγουμε δεδομένα από σελίδς και να τα αναγνωρίσουμε ως στιγμιότυπα page-schemes. Αυτοί που αναλαμβάνουν τη παραπάνω δουλειά ονομάζονται **wrappers** και είναι αυτοί που επιτρέπουν την αναζήτηση και την αναδόμηση των ημιδομημένων κειμένων.

Unql

Ένα ακρώνυμο το οποίο βγαίνει από τα αρχικά των λέξεων Unstructured Data Query Language, που σημαίνει γλώσσα αναζήτησης μη δομημένων δεδομένων. Η ιδέα σε αυτή τη γλώσσα αναζήτησης είναι να περιορίσει τη φόρμα ενός αναδρομικού προγράμματος με το να το δένει αυστηρώς στενά με την αναδρομική δομή των δεδομένων. Η Unql ήταν μια από τις πρώτες γλώσσες αναζήτησης για ημιδομημένα δεδομένα και πληρεί πολλά από τα κριτήρια για μια γλώσσα αναζήτησης δεδομένων (βελτιστοποιημένη άλγεβρα, απλή γλώσσα αναζήτησης, σύνθεση κ.α).

Επιπρόσθετα με τη δομημένη αναδρομή, η Unql εισήγαγε την ιδέα της χρησιμοποίησης pattern (μορφής) και template (φόρμας) σε μια απλή επιφανειακή σύνταξη. Μιλώντας όμως πιο ουσιαστικά η Unql είναι ένα μοντέλο σα το σχεσιακό μοντέλο βασισμένο σε τιμές και αυτό επιτρέπει την ανάλογη βελτιστοποίηση με αυτές των σχεσιακών βάσεων δεδομένων.

Το κύριο χαρακτηριστικό αυτής της γλώσσας αναζήτησης είναι η δομημένη αναδρομή (structural recursion). Η ιδέα αυτού του χαρακτηριστικού είναι ότι η φόρμα του προγράμματος ακολουθεί τη δομή δεδομένων. Οι περιορισμοί, που είναι συντακτικού εγγυώνται ότι η αναδρομή πάντα τερματίζει. Η δομημένη αναδρομή είναι δηλωτική και επιτρέπει βελτιστοποιήσεις αναμενόμενες απο μια γλώσσα αναζήτησης βάσης δεδομένων.

Συγκριτικός πίνακας

Μετά την περιγραφή διάφορων γλωσσών αναζήτησης, καταλήξαμε στο παρακάτω πίνακα όπου φαίνονται συγκεντρωτικά οι γλώσσες.

System	Data Model	Language Style	Path Expression	Graph
Websql	Relational	SQL	YES	NO
W3QS	LMG	SQL	YES	NO
WebLOG	Relational	Datalog	NO	NO
Lorel	LG	OQL	YES	NO
Weboql	hypertrees	OQL	YES	YES
UnQL	LG	Recursion	YES	YES
FLORID	F-logic	Datalog	YES	NO
Strudel	LG	Datalog	YES	YES
Araneus	Page Schemes	SQL	YES	YES

Βιβλιογραφία

1. UNQL: A Query language and algebra for semistructured data based on structural recursion.
   (Peter Buneman, Mary Fernandez, Dan Suciu)

2. Catching the boat with Strudel:
   Experiences with a web-site management system.
   (Mary Fernandez, Daniela Florescu, Jaewoo Kang, Alon Levy)

3. Querying the world wide web
   (Alberto O.Mendelzon, George A. Mihaila, and Tova Milo)

4. The Araneus web-base management system
   (G. Mecca, P.Atzeni, A.Masci, P.Merial, G.Sindoni)

5. Database Techniques for the world wide web: A survey
   (Daniela Florescu, Alon Levy, Alberto Mendelzon)

6. Formulating disjunctive coupling queries in a web warehouse
   (Sourav S Bhowmick, Ang Kho Kiong and Sanjay Madria)

7. A Declarative Language for Querying and Restructing the Web
   (Laks V.S. Lakshmanan, Fereidoon Sadri, Iyer N. Subramanian)

8. WebOQL: Restructuring Documents, Databases and Webs
   (Gustavo O. Arocena, Alberto O. Mendelzon)

9. Information Gathering in the World-Wide Web: The W3QL Query Language and the W3QS System.
   (David Konopnicki, Oded Shmueli)

10. The Lorel Query Language for Semistructured Data
    (Serge Abiteboul, Dallan Quass, Jason McHugh, Jennifer Widom, Janet L. Wiener)

11. Lore: A Database Management System for Semistructured Data
    (Jason McHugh, Serge Abiteboul, Roy Goldman, Dallan Quass, JenniferWidom)

12. Design and Development of Data-Intensive Web Sites: The ARANEUS Approach
    (Paolo Merialdo, Paolo Atzeni)

13. Applications of a Web Query Language
    (Gustavo O. Arocena, Alberto O. Mendelzon, George A. Mihaila)