Cabinet and Card

Federated Access to Chemical and Biological Data

What is Cabinet

* Chemical And Biological Informatics NETwork (aka Fedora).
* A federation of servers which each serve a particular data set.
* No unified data model - each server uses the best model for the particular information.
* The servers are useful in isolation or in any combination.

Data Models and Languages

* Cabinet servers can provide different types of searches depending on their particular data model (small molecule similarity, protein sequences similarity, etc).
* Servers speak HTTP to clients (browsers and other Cabinet servers).
* Each Cabinet servers can send queries to other Cabinet servers via languages (e.g. what do you know about molecules like this SMILES?).

Data Models and Languages

* Cabinet servers don't contain much metadata. Searches are carried out in memory.
* Queries can be posed without needing an understanding of the data organization.

Why Not Unification?

* An alternative approach would be to use a unified data model.
* This potentially suffers from the least common denominator problem.
* Creating a unified model of diverse subject matter is hard and difficult to expand.

Why Not Unification?

* Imposing elaborate structure on data can limit the types of questions that can be asked.
* However, attempts to unify knowledge are certainly worthwhile and unified approaches to integration are complementary to federated approaches.

Federation Advantages

* No need for a global data model.
* Addition of new subject areas is thus easier since there are fewer global dependencies.
* Each subject is represented in the most appropriate data model.

Federation Disadvantages

* Loose coupling and anarchy.
* Results are sometimes unexpected (just like searching the WWW).
* If a query can be posed to a more highly structured informatics system, the results are likely to be more precise.

Demo

* A web accessible demo version of Cabinet is available by signing up at http://cabinet.metaphorics.com.

cabinet: **home page**
chemical and biological informatics **network**

CABINET status help

Welcome to cabinet, a federation of high-performance scientific databases

This a collection of research databases which collaborate via weblike interfaces. These servers have been developed by Metaphorics LLC, Daylight CIS and their partners.

Most of these servers require you to log in as a user. You are not currently logged in. If you are a registered user, you may **login here** for access to cabinet services. If not (or if you've forgotten your password), you may **register here** for a demo account.

Note: your e-mail address is now your cabinet **User Name**.

Note: Netscape (4.7 or later) is the preferred browser; current versions of Mozilla and IE also work.

cabinet chemical and biological informatics network		
bbc biobyte calculations	**morpho** morphochem 15840x5	**sandman** server and daemon manager
cabinet chemical and biological informatics network	**orange** FDA Orange Book	**tcm** traditional chinese medicines
dcm dictionary of chinese medicine	**park** photo arkive	**wdi** world drug index
download server/helper downloads	**planet** protein-ligand association network	**wombatdemo** world of molecular bioactivity
ecbook enzyme commission codebook	**postage** postage -- PDF/postscript depictions with data	**zi4** chinese character service
empath metabolic pathway chart	**qsar** quantitative structure-activity relationships	

CARD

Cabinet Access to Relational Data

Motivations

* Primary motivation:
  provide users with a means to incorporate their own data within servers in the Cabinet federation.
  * Probably already stored in RDBMS or other structured format

Motivations

* Provide Metaphorics with a tool set to generate Cabinet servers for data which require an RDBMS environment.
* Rapidly changing/updated
* Large datasets

Motivations

* Provide a tool for generating DayCart based interfaces and applications, perhaps outside of the Cabinet federation.

Incorporating User Data

To incorporate user data from an RDBMS in Cabinet servers, we need to provide three things:

1. User control of look and feel. This is accomplished by providing a template system in CARD
2. User control of data access. Provided via a small amount of scripting plus SQL access to the back end RDBMS. An embedded SQL engine is also available (based on SQLite).
3. Access to Cabinet functionality. Provided by the underlying CARD application.

Look and Feel

Web content generation is widely done via template systems. The CARD template system, ANTES (ANother TEmplate System) is based on the StringTemplate language which enforces a strict separation of presentational and procedural components. The author of StringTemplate (Terence Parr) has written an excellent paper on the benefits of such a design.

ANTES

* is a Metaphorics implementation of the StringTemplate language based on the language grammar from the StringTemplate public documentation.
* is written in C and uses the Daylight toolkit object system.
* provides API binding for C and the Lua scripting language. Other bindings are possible.

ANTES

* ANTES templates are simply text (HTML) with embedded named slots (called attributes) where values can be substituted.
* Simplest example:

  <h1>Hello, $planet$</h1>
  evaluates to (when the value of attribute planet is "World"):
  <h1>Hello, World</h1>

* There are more advanced features in ANTES, including multi-valued attributes and templates called from other templates.

Scripting

* Small scripts are used to generate SQL queries and map returned data to template attributes.
* The scripts are written in the Lua language:
  * Small, elegant language
  * Written in portable, standard C
  * Easy to embed in applications
  * Easy to call C routines

CARD Application

* Generic application which is specialized to a particular database and application via a configuration file:
  * templates
  * scripts
  * help
  * preferences

CARD Application

* Application provides:
  * HTTP server (HTTP toolkit)
  * RDBMS backend access
    * internal RDBMS access
    * Daylight and Metaphorics tools
    * page header and footer, navigation menu, preferences and help
    * Cabinet services: server to server queries

Functional Overview

* Receive HTTP request.
* Pass arguments from URL and/or POST to appropriate page script.
* Select template.
* Generate SQL and submit to database.
* Assign returned data to template attributes.
* Evaluate template and return generated HTML string.

Demo

Cabinet

* Cabinet provides an easy to use method for exploring chemical and biological relationships.

CARD

* a database driven, scriptable, template based web content generation system.
* provides chemical and biological information processing facilities via:
  * internal facilities
  * DayCart and RDBMS backends
* generates servers which are full members of a Cabinet federation.

Acknowledgments

* Metaphorics
  * Dave Weininger
  * Vera Povolna
* Daylight
  * Jack Delany
  * Mick Kappler

Status Example

status

http://obelisk.daylight.com:26599/card/status.ht

card: **status**

CARD status help preferences examples

card server status summary:

This **card** service is hosted by **obelisk :26599** at **10.44.1.94** .
This server has been up for **15:33:59** and has serviced **91** HTTP requests.
239492 data objects are current.

Metaphorics	*Metaphorics LLC*	*card 4.83*

Done

Disabled

Status Example

```
<h3><font color='#FF0000'>card</font> server
status summary:</h3>
<blockquote>
This <b>card</b> service is hosted by
<b>obelisk :26599</b> at <b>10.44.1.94</b>.
<br>This server has been up for <b>15:31:59</b>
and has serviced <b>91</b> HTTP requests.
<p><b>239492</b> data objects are current.</p>
</blockquote>
```

card server status summary:

This **card** service is hosted by **obelisk :26599** at **10.44.1.94** .
This server has been up for **15:31:35** and has serviced **87** HTTP requests.

239492 data objects are current.

Status Example

```
<h3><font color='#FF0000'>card</font> server
status summary:</h3>
<blockquote>
This <b>card</b> service is hosted by
<b>$card_host$:$card_port$</b> at <b>$card_lips$</b>.
<br>This server has been up for <b>$card_uptime$</b>
and has serviced <b>$card_nreqs$</b> HTTP requests.
$if (ndata)$<p><b>$ndata$</b> data objects are current.</p>$endif$
</blockquote>
```

Status Example

```
path="/card/status",
type = ".html",
script = [[
local g = antes.createGroup("status")
g:setSuperGroup(card.master)
local t = g:getInstanceOf("card/status")
local db=oracle.connect(card.db)
card.set_title("status")
...
```

Status Example

```
...
local e,s = db:SQL("select count(*) from acd_main",
 1,0,function(ncols,cols,names) t:setAttribute("ndata",cols[1]) end)
t:setAttribute("card_uptime",card.card_uptime)
t:setAttribute("card_host", card.get_prop(card.htob, "_host"))
t:setAttribute("card_port", card.get_prop(card.htob, "_port"))
t:setAttribute("card_lips", card.get_prop(card.htob, "_lips"))
t:setAttribute("card_nreqs", card.get_prop(card.htob, "_nreqs"))
return t:toString()
]]
```