Discovering your Discovery System in Real-Time

Annette Bailey

Godmar Back
(Academic) Libraries in 2014: some observations

- Majority of library resources used are electronic
- Physical space is being dedicated to student work areas and digital displays
- Librarians are becoming active partners in all stages of research
- Libraries are replacing their OPACs with discovery systems
Risks associated with these changes

- Use of library collections becomes invisible
- Search engines compete with discovery systems for users requiring substantial marketing effort for the discovery system

Challenge:

Make the discovery process & use of resources visible!
Can We See What Our Users are Doing?

- There’s the COUNTER standard which gives us usage data
  - Usually, after the fact, and (somewhat) hard to process
  - Topic of a different talk/project ...
- Summon provides analytics data on search terms & click data
  - Aggregate, not real-time
- Google Analytics can record data
  - But no real-time access

- But what if we could see what our users are doing in real time (while still affording them some privacy)?
- And what if we could share that with other members of the community?
Discovering Discovery

- Visualize, in real-time, the items users discover as they click on results in Summon
- It’s as if we could see what object the user is plucking off the shelf. We can’t see if they actually use the object, but we know that they are interested enough to go further than the search results.
Live Demos

- Let’s see what users are clicking on in the VT instance of Summon right now!
- Twitter-Style Widget: [Link]
- Summon Cube [Link]
- Word Cloud (Subject/Last50) (Title/Last50)
- Gauge [Link]
- Discipline Ticker: [Link]
- Publication Year Chart [Link]
Summon Cube

Generating Domain-Specific Visual Language Tools from Abstract Visual Specifications
by Ali Norhayati Mohd; Li, Karen Na; Grundy, John C; Hosking, John; Huh, Jun; Li, Richard Lei
IEEE Transactions on Software Engineering. ISSN 0098-5589, Volume 39, Issue 4, pp. 487–515
Content Type: Journal Article
Publisher: Institute of Electrical and Electronics Engineers, Inc
Date: 2013
Abstract: Domain-specific visual languages support high-level modeling for a wide range of application domains. However, building tools to support such languages is very challenging. We describe a set of key conceptual requirements for such tools and our approach to addressing these requirements, a set of visual language-based metatools. These support definition of metamodels, visual notations, views, modeling behavors, design critics, and model transformations and provide a platform to realize target visual modeling tools. Extensions support collaborative work, human-centric tool interaction, and multiplatform deployment. We illustrate application of the metatoolset on tools developed with our approach. We describe tool developer and cognitive evaluations of our platform and our exemplar tools, and summarize key future research directions.
Ticker & Gauge

Live ticker of Summon Disciplines

eutics, & Pharmacology: 1 (▲1)  Business: 1 (▼-1)  History & Archaeology: 1 (▲0)  Anatomy & Physiology: 1 (▲)

This ticker shows a ticker-style histograms of the disciplines accessed in the last 50 Summon hits. The deltas shown are between the current histogram and the histogram 10 clicks ago.
Impact

- Our research community is not just using Google, they are using Summon and it is clear from the searches and results that they are using Summon for research.
- This project allows not just librarians, but the community (and potential members of the community – potential students) to see what research is happening at Virginia Tech in real time.
- Intended both as a webservice/widget, or as a public display in the library.
Technical Details: Click Recording

Summon embeds the Summon ID of each result in the page.
Technical Details: Click Recording

```javascript
$(function () {
    $('#resultPage .document').each(function (idx, doc) {
        var $doc = $(doc);
        var id = $doc.attr('id');  // extract id
        function recordClick() {
            var cImg = new Image(1, 1); // create unique URL to avoid browser cache
            cImg.src = "http://libx.lib.vt.edu/services/summonvis/click.gif?"
                + "id=" + encodeURIComponent(id)
                + "&_ts=" + Math.floor(Math.random() * 10000000);
        }
        // register click handler for the multiple links a user may use
        $doc.find("a.documentLink").click(recordClick);
        $doc.find("div.previewDocumentTitle a").click(recordClick);
        $doc.find("div.thumbnail a").click(recordClick);
        $doc.find("div.Availability div.summary a").click(recordClick);
    });
});
```
Technical Details: Logging Clicks

- Use Apache Common Log Format (no recording of referrer URL)

```text
xxx.82.xxx.75 - - [07/Nov/2013:16:20:28 -0500] "GET /services/summonvis/click.gif?id=FETCH-webofscience_primary_0002758417006911&_ts=390057 HTTP/1.1" 200 333
xxx.82.xxx.92 - - [07/Nov/2013:16:21:27 -0500] "GET /services/summonvis/click.gif?id=FETCH-vt_catalog_b223625751&_ts=6148119 HTTP/1.1" 200 333
xxx.82.xxx.11 - - [07/Nov/2013:16:21:27 -0500] "GET /services/summonvis/click.gif?id=FETCH-proquest_abstracts_14361167591&_ts=1597476 HTTP/1.1" 200 333
```

# Excerpt from Apache config file: /etc/apache2/sites-enabled/000-default
# do not log user's query terms

```
SetEnvIf Request_URI "^/services/summonvis" dontlog
```

# log resource ids users actually click on

```
SetEnvIf Request_URI "/services/summonvis/click.gif" clicklog
```

```
CustomLog /var/log/apache2/access.log combined env=!dontlog
CustomLog /var/log/apache2/click.log common env=clicklog
```
Summon Clicks
1/13/2013 to 3/24/2014
767,997 clicks
491,204 unique items discovered
Log Tracing

- When a click entry is added, we must contact Summon API right away
  - Summon IDs are very short-lived due to constant record update & merging

- Summon API
  - Supports querying in the same manner as Summon web front-end
  - Supports additional query-styles, such as retrieve by ID
  - Requires API-Key (which must be safe-guarded)
  - Returns JSON or XML

- SQLite
  - We store all records in SQLite DB
  - See http://www.sqlite.org/

```python
import sqlite3
conn = sqlite3.connect(dbfilename)
conn.execute('''CREATE TABLE IF NOT EXISTS summonrecords
(id TEXT PRIMARY KEY, recordjson TEXT)''')
```
Sample Record

- Selected fields from JSON response

http://en.wikipedia.org/wiki/Tanager

```json
{
    "Author": [
        "Zdziarski, JM",
        "Little, SE",
        "Adkesson, MJ"
    ],
    "ContentType": [
        "Journal Article"
    ],
    "DOI": [
        "10.1605/01.301-0000384790.2006"
    ],
    "DatabaseTitle": [
        "ProQuest Illustrata: Natural Sciences",
        "ProQuest Deep Indexing: Biological Science"
    ],
    "Discipline": [
        "Veterinary Medicine"
    ],
    "EISSN": [
        "1937-2825"
    ],
    "SubjectTerms": [
        "Cyanerpes caeruleus",
        "Cyanerpes cyaneus",
        "Tanager whole blood",
        "Polymerase chain reaction results",
        "Dacnis cayana",
        "Tissue samples"
    ],
    "Title": [
        "Atoxoplasmosis In Tanagers"
    ],
    "PublicationYear": [
        "2005"
    ],
    "SourceType": [
        "Index Database"
    ],
    "StartPage": [
        "265"
    ],
    "EndPage": [
        "272"
    ],
    "ISSN": [
        "1042-7260"
    ],
    "IsPeerReviewed": [
        "true"
    ],
    "IsScholarly": [
        "true"
    ],
    "Issue": [
        "2"
    ],
    "Language": [
        "English"
    ],
    ...
}
```

http://en.wikipedia.org/wiki/Tanager
{  
"Abstract": [  
"Introduction. To report a live birth following egg retrieval after only 12 hours from hCG priming. Patients. A childless couple with five-years-lasting secondary infertility. Methods. IVF was performed according to the long protocol. Two immature oocytes were retrieved following only 12 hours after hCG priming due to the patient misunderstanding. The eggs were cultured in vitro and ICSI was performed following polar body extruded after 24 hours in culture. After additional 24 hours a 4-cell embryo was developed and ET was performed. Results. A viable pregnancy was achieved and a healthy baby girl was delivered at 38 weeks of gestation. Conclusion. In a rare and unexpected situation when immature oocytes are retrieved following a short hCG priming, the eggs should be cultured in vitro, late ICSI should be performed, and a pregnancy may be expected."
],  
"Snippet": [  
"Introduction. To report a live birth following egg retrieval after only 12 hours from hCG priming. Patients. A childless couple with five-years-lasting...",  
"Introduction. To report a live birth following egg retrieval after only 12 hours from hCG priming. Patients. A childless couple with five-years-lasting..."
],  
"Title": [  
"A Live Birth Subsequent to IVF following Egg Retrieval Only 12 Hours after hCG Priming"
],  
"URI": [  
"http://dx.doi.org/10.1155/2013/634385",
"http://search.proquest.com/docview/1428017900",
]}

 Sample Record (2)

- Selected fields from JSON response
Current Log Analysis

- Extracts & tabulates easy-to-analyze variables
  - over time periods: 1 min, 5 min, 1 hours, 1 day, 1 week
  - over counts: last 50, last 100, last 200, last 1000, last 10000

- Tabulates frequency by categories such as
  - Discipline
  - ContentType
  - SourceType (of record)
  - PublicationYear

- Tabulates word frequencies for
  - Abstract, Title, Abstract & Title together (individual words)
  - Keywords, Subject Terms, Keywords & Subject Terms together (preserving compounds)
Example 1:
Tabulating Disciplines

- Last 1 hour

```
{ Discipline:
  { Psychology: 4,
    'Applied Sciences': 1,
    'Public Health': 1,
    Forestry: 1,
    Religion: 6,
    'Environmental Sciences': 4,
    Music: 2,
    Medicine: 7,
    Mathematics: 2,
    'Sciences (General)': 2,
    Biology: 5,
    Business: 1,
    Government: 2,
    Philosophy: 6,
    Anthropology: 1,
    'Computer Science': 3,
  },
'Sociology & Social History': 2,
  Law: 5,
  'Languages & Literatures': 6,
  'International Relations': 1,
  'Library & Information Science': 1,
  Economics: 4,
  Physics: 4,
  'Visual Arts': 1,
  Ecology: 3,
  'History & Archaeology': 1,
  Geology: 2,
  'Anatomy & Physiology': 1,
  Engineering: 7,
  Zoology: 1,
  Education: 5,
  Agriculture: 6,
  Film: 1,
  Geography: 1 },
timestamp: '2013-11-07T19:07:00-05:00' }
```
Example 2: Tabulating PubYear

- Last 1 Day

<table>
<thead>
<tr>
<th>PublicationYear</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>'1765'</td>
<td>1</td>
</tr>
<tr>
<td>'1775'</td>
<td>1</td>
</tr>
<tr>
<td>'1790'</td>
<td>1</td>
</tr>
<tr>
<td>'1791'</td>
<td>1</td>
</tr>
<tr>
<td>'1793'</td>
<td>2</td>
</tr>
<tr>
<td>'1850'</td>
<td>1</td>
</tr>
<tr>
<td>'1855'</td>
<td>1</td>
</tr>
<tr>
<td>'1863'</td>
<td>1</td>
</tr>
<tr>
<td>'1871'</td>
<td>1</td>
</tr>
<tr>
<td>'1876'</td>
<td>1</td>
</tr>
<tr>
<td>'1879'</td>
<td>1</td>
</tr>
<tr>
<td>'1880'</td>
<td>1</td>
</tr>
<tr>
<td>'1881'</td>
<td>1</td>
</tr>
<tr>
<td>'1882'</td>
<td>1</td>
</tr>
<tr>
<td>'1889'</td>
<td>1</td>
</tr>
<tr>
<td>'1895'</td>
<td>2</td>
</tr>
<tr>
<td>'1899'</td>
<td>3</td>
</tr>
<tr>
<td>'1900'</td>
<td>1</td>
</tr>
<tr>
<td>'1901'</td>
<td>4</td>
</tr>
<tr>
<td>'1902'</td>
<td>1</td>
</tr>
<tr>
<td>'1903'</td>
<td>1</td>
</tr>
<tr>
<td>'1905'</td>
<td>1</td>
</tr>
<tr>
<td>'1913'</td>
<td>1</td>
</tr>
<tr>
<td>'1915'</td>
<td>1</td>
</tr>
<tr>
<td>'1916'</td>
<td>2</td>
</tr>
<tr>
<td>'1920'</td>
<td>3</td>
</tr>
<tr>
<td>'1923'</td>
<td>1</td>
</tr>
<tr>
<td>'1924'</td>
<td>1</td>
</tr>
<tr>
<td>'1925'</td>
<td>1</td>
</tr>
<tr>
<td>'1926'</td>
<td>2</td>
</tr>
<tr>
<td>'1930'</td>
<td>2</td>
</tr>
<tr>
<td>'1931'</td>
<td>2</td>
</tr>
<tr>
<td>'1932'</td>
<td>2</td>
</tr>
<tr>
<td>'1933'</td>
<td>2</td>
</tr>
<tr>
<td>'1934'</td>
<td>1</td>
</tr>
<tr>
<td>'1936'</td>
<td>3</td>
</tr>
<tr>
<td>'1938'</td>
<td>1</td>
</tr>
<tr>
<td>'1941'</td>
<td>2</td>
</tr>
<tr>
<td>'1942'</td>
<td>2</td>
</tr>
<tr>
<td>'1946'</td>
<td>1</td>
</tr>
<tr>
<td>'1947'</td>
<td>1</td>
</tr>
<tr>
<td>'1948'</td>
<td>2</td>
</tr>
<tr>
<td>'1951'</td>
<td>1</td>
</tr>
<tr>
<td>'1952'</td>
<td>1</td>
</tr>
<tr>
<td>'1954'</td>
<td>3</td>
</tr>
<tr>
<td>'1955'</td>
<td>3</td>
</tr>
<tr>
<td>'1956'</td>
<td>2</td>
</tr>
<tr>
<td>'1957'</td>
<td>4</td>
</tr>
<tr>
<td>'1958'</td>
<td>7</td>
</tr>
<tr>
<td>'1959'</td>
<td>4</td>
</tr>
<tr>
<td>'1960'</td>
<td>3</td>
</tr>
<tr>
<td>'1961'</td>
<td>9</td>
</tr>
<tr>
<td>'1962'</td>
<td>5</td>
</tr>
<tr>
<td>'1963'</td>
<td>1</td>
</tr>
<tr>
<td>'1964'</td>
<td>5</td>
</tr>
<tr>
<td>'1965'</td>
<td>9</td>
</tr>
<tr>
<td>'1966'</td>
<td>9</td>
</tr>
<tr>
<td>'1967'</td>
<td>8</td>
</tr>
<tr>
<td>'1968'</td>
<td>3</td>
</tr>
<tr>
<td>'1969'</td>
<td>13</td>
</tr>
<tr>
<td>'1970'</td>
<td>10</td>
</tr>
<tr>
<td>'1971'</td>
<td>5</td>
</tr>
<tr>
<td>'1972'</td>
<td>8</td>
</tr>
<tr>
<td>'1973'</td>
<td>2</td>
</tr>
<tr>
<td>'1974'</td>
<td>16</td>
</tr>
<tr>
<td>'1975'</td>
<td>6</td>
</tr>
<tr>
<td>'1976'</td>
<td>5</td>
</tr>
<tr>
<td>'1977'</td>
<td>8</td>
</tr>
<tr>
<td>'1978'</td>
<td>4</td>
</tr>
<tr>
<td>'1979'</td>
<td>8</td>
</tr>
<tr>
<td>'1980'</td>
<td>9</td>
</tr>
<tr>
<td>'1981'</td>
<td>7</td>
</tr>
<tr>
<td>'1982'</td>
<td>10</td>
</tr>
<tr>
<td>'1983'</td>
<td>12</td>
</tr>
<tr>
<td>'1984'</td>
<td>10</td>
</tr>
<tr>
<td>'1985'</td>
<td>16</td>
</tr>
<tr>
<td>'1986'</td>
<td>16</td>
</tr>
<tr>
<td>'1987'</td>
<td>16</td>
</tr>
<tr>
<td>'1988'</td>
<td>12</td>
</tr>
<tr>
<td>'1989'</td>
<td>21</td>
</tr>
<tr>
<td>'1990'</td>
<td>29</td>
</tr>
<tr>
<td>'1991'</td>
<td>16</td>
</tr>
<tr>
<td>'1992'</td>
<td>32</td>
</tr>
<tr>
<td>'1993'</td>
<td>29</td>
</tr>
<tr>
<td>'1994'</td>
<td>29</td>
</tr>
<tr>
<td>'1995'</td>
<td>34</td>
</tr>
<tr>
<td>'1996'</td>
<td>43</td>
</tr>
<tr>
<td>'1997'</td>
<td>33</td>
</tr>
<tr>
<td>'1998'</td>
<td>62</td>
</tr>
<tr>
<td>'1999'</td>
<td>58</td>
</tr>
<tr>
<td>'2000'</td>
<td>56</td>
</tr>
<tr>
<td>'2001'</td>
<td>85</td>
</tr>
<tr>
<td>'2002'</td>
<td>72</td>
</tr>
<tr>
<td>'2003'</td>
<td>70</td>
</tr>
<tr>
<td>'2004'</td>
<td>106</td>
</tr>
<tr>
<td>'2005'</td>
<td>100</td>
</tr>
<tr>
<td>'2006'</td>
<td>104</td>
</tr>
<tr>
<td>'2007'</td>
<td>103</td>
</tr>
<tr>
<td>'2008'</td>
<td>149</td>
</tr>
<tr>
<td>'2009'</td>
<td>177</td>
</tr>
<tr>
<td>'2010'</td>
<td>222</td>
</tr>
<tr>
<td>'2011'</td>
<td>251</td>
</tr>
<tr>
<td>'2012'</td>
<td>323</td>
</tr>
<tr>
<td>'2013'</td>
<td>281</td>
</tr>
<tr>
<td>'2014'</td>
<td>1</td>
</tr>
</tbody>
</table>

Discovered Items During Last Day, By Publication Year

last updated a few seconds ago

Generated using Google Charts [Link to live version]
Example 3: Tabulating Title

- Last 1 Hour
- Single words
- Lowercase

```
{ timestamp: '2013-11-07T20:47:26-05:00',
  Title:
  [ ['special', 14],
    ['olympics', 13],
    ['fundraising', 10],
    ['cannabis', 10],
    ['food', 9],
    ['united', 7],
    ['history', 7],
    ['intake', 6],
    ['sweeteners', 6],
    ['genetically', 6],
    ['sex', 6],
    ['energy', 6],
    ['social', 6],
    ['transsexuality', 6],
    ['children', 6],
    ['changed', 6],
    ['modified', 6],
    ['effects', 5],
    ['states', 5],
    ['discourse', 5],
    ['meyerowitz', 5],
    ['trafficking', 5],
    ...
  }
```

[ 'north', 4 ],
[ 'mentoring', 4 ],
[ 'medical', 4 ],
[ 'narrative', 4 ],
[ 'cbd', 4 ],
[ 'joanne', 4 ],
[ 'journal', 4 ],
[ 'state', 4 ],
[ 'mexico', 4 ],
[ 'era', 3 ],
[ 'america', 3 ],
[ 'economic', 3 ],
[ 'education', 3 ],
[ 'artificial', 3 ],
[ 'peace', 3 ],
[ 'high', 3 ],
[ 'security', 3 ],
[ 'treatment', 3 ],
[ 'weight', 3 ],
[ 'bugs', 3 ],
[ 'stink', 3 ],
[ 'cancer', 3 ],
[ 'students', 3 ],
[ 'theory', 3 ],
Visualization Implementation

Source: http://jsfiddle.net/danielfilho/GdCcA/
Plotting Charts with **Google Charts**

- Developed & Hosted by Google
- Closed-source, but free to use
- Uses SVG
- Very customizable
- Supports animations
- Example: [https://developers.google.com/chart/interactive/docs/gallery/columnchart](https://developers.google.com/chart/interactive/docs/gallery/columnchart) #Example
- Alternatives: flot, jqplot, HighCharts, etc. etc.
d3.js

- **Source:** [http://d3js.org/](http://d3js.org/)

- **D3: Data-Driven Documents**
  Michael Bostock, Vadim Ogievetsky, Jeffrey Heer
d3.js - Description

**D3** allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, you can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive SVG bar chart with smooth transitions and interaction.

D3 is not a monolithic framework that seeks to provide every conceivable feature. Instead, D3 solves the crux of the problem: efficient manipulation of documents based on data. This avoids proprietary representation and affords extraordinary flexibility, exposing the full capabilities of web standards such as CSS3, HTML5 and SVG. With minimal overhead, D3 is extremely fast, supporting large datasets and dynamic behaviors for interaction and animation. D3’s functional style allows code reuse through a diverse collection of **components** and **plugins**.

Source: [http://d3js.org/](http://d3js.org/)
Using d3.js

- Though well-designed, has a steep learning curve
- Follows jQuery in style – provides operations on selections (e.g. everything can operate on multiple elements at once)
- Supports transitions
  - E.g., change this circle’s size + position from 100 at (0, 0) to 150 at (45, 70) and make it move from the old to the new position in 1,000 ms.
- Requires OO JavaScript skills
- Fortunately: many components out there, many examples, very active user community.
  - E.g. WordCloud by Jason Davies [Click for demo]
Server Side Technologies Used

- As data changes, need to update client display
- Implemented in CoffeeScript using node.js [Link]
- Node.js: JavaScript platform for running server-side code
  - Based on Google’s V8 JavaScript VM (same as used in Chrome browser)
- CoffeeScript: is a language that compiles to JavaScript, but uses a cleaner and richer syntax [Link]
- Socket.IO [Link]: is a library that supports client/server communication
  - Includes support for websockets
  - Kind of a successor to AJAX, makes “pushes” from server to client easier
- Code is here
Introducing LibFX

- Right now, a bunch of scripts JavaScript, Python, CoffeeScript
  - Source code of course available
  - Contributions/collaborations are invited

- Idea: Build a cloud platform that allows other libraries to adopt and extend it
  - Particularly for non-programmers/non-sys admins
  - Not reliant on cloud, but ready to run “in the cloud” – on a IaaS or PaaS platform

- LibX Edition Builder experience: adoption really took off once we created a management interface that allowed independent setup & exploration
Commercial or open source discovery system, e.g. Summon, EBSCO, Blacklight

Users

Public Display & Interaction

Web-based visualizations, Incl. social media

libFX

Real-time Data Collection

Data Processing & Storage

Data Visualization

Augmentation

Internal Services, e.g. Circulation Statistics, COUNTER Reports

External Services, e.g. Mendeley API, WorldCat Identities, CiteULike, altmetrics.com

Web-Based Control & Configuration Interface

Offline Analysis

Connector
Federation of libraries employing libFX
Challenges & Opportunities

- What other information can be visualized?
  - Search terms? Search successes?
- How to create an attention-catchng, yet unattended visualization
  - Digital Signage
- How to make the visualization interactive, allowing users to participate
  - E.g. QR codes, user feedback

- Last demo: [Listen To Summon]
- Website:
  - http://libfx.lib.vt.edu

Credits

- Jason Davies for d3 wordcloud
- Bojan Đuričić for justGage
- Stephen LaPorte and Mahmoud Hashemi for Listen To Wikipedia
  - Based on Maximillian Laumeister Listen To BitCoin
- The Art Of Web.com for 3D Cube example