Towards Engaging Big Data for CS 1/2

Steven Benzel
Georgia Highlands College, Cartersville, GA

Nadeem Abdul Hamid
Berry College, Mount Berry, GA

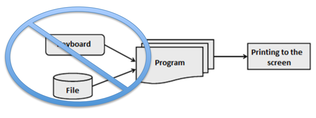
Experience it Firsthand

Abstract

We present progress on a code framework and methodology to facilitate the incorporation of large, online data sets into traditional CS1 and CS2 courses. The goal of our project is to develop a way to provide students a library that relieves them from low-level issues of reading and parsing raw data from web-based data sources and that interfaces with data structures and representations defined by students themselves. In addition, the library requires minimal syntactic overhead to use its functionality and allows students and instructors to focus on algorithmic exercises involving processing live and large data obtained from the Internet. At a minimum, the library should serve to create drop-in replacements for traditional programming exercises in introductory courses - raising the engagement level by having students deal with "real" data rather than artificial data provided through standard input.

Motivation

- Contextualized approaches to introductory Computer Science cater to students' interests
- We live in an age of "big data" – massive, open-accessible online datasets
- Instead of text-based "standard input" and artificial student-generated data, let's engage students with "real" data



Goals

- ✓ Seamless direct connection to web-based APIs
- ✓ **Relieve students from low-level I/O and parsing** raw data
- ✓ **Minimal syntactic overhead** (e.g. on par with setting up a Scanner in Java to read from System.in)
- ✓ **Interface with student-defined data structures**
- ✓ Provide ability to obtain lists/arrays of parsed data when appropriate
- ✓ Built-in help/usage functionality; robust error-handling and reporting
- ✓ Support data source "plug-ins" and contributions

❖ Some Challenges / ✓ Tentative Approach

- ❖ Lots of different APIs
  - ✓ Target common data formats – XML, JSON, CSV – provide robust parsing functionality
- ❖ Registration/developer key requirements for some APIs
  - ✓ Make it as seamless as possible to supply data source options and parameters
- ❖ Network load / large data sets
  - ✓ Support intelligent caching of data sets / pagination
- ❖ Data sets provide much more information than might be of interest
  - ✓ Provide convenient methods to extract data fields of interest; map to student-defined data structures

Technical Details

Java/Processing implementation
- ➢ Utilizes Reflection API to dynamically construct objects at runtime
- ➢ Primarily supports XML, CSV data formats so far

Racket implementation
- ➢ Currently only Kiva API implementation (see example below)
- ➢ Macros provide compile-time syntax checking of extracted fields

Future Work

- ✓ Assignment/curricular development
- ✓ Field-testing and surveying actual use of the library

Examples

Try it out: http://cs.berry.edu/big-data

```
DataSource ds;

/* online API */
ds = DataSource.connect("http://services.faa.gov/airport/status/JFK?format=application/xml").load();
double temp = ds.fetchDouble("Weather/Temp");  // in Celsius

/* with options */
ds = DataSource.connectXML("http://api.worldweatheronline.com/free/v1/weather.ashx");
ds.set("q", "30149").set("key", "...<apikey>...").set("format", "xml").load();
ds.printUsageString();

/* local file */
ds = DataSource.connect("vehicles.xml").load();
ArrayList<Car> cs = ds.fetchList("Car", "make", "model", "city08");

/* data source specification file */
ds = DataSource.connect("FAAAirportStatus.xml").set("airportCode", "LAX").load();
DataSourceIterator iter = ds.iterator();
while (iter.hasData()) {
      String name = iter.fetchString("Name");
      String status = iter.fetchString("Status Comment");
      System.out.println(name + ": " + status);
      iter.loadNext();
}
```

```
class Car {
   String company;
   String model;
   int mpgCity;
   …
}
```

```
(require "kiva-teachpack.rkt")

(define-struct borrower(name size-of-team country type-of-bus req-amt raised-amt))
...
(define (convert-kiva-data kivadata)
  (kiva-data->structs kivadata make-borrower
                      (name size country activity loan_amt funded_amt)))
...
(show-html
   (stat->cloud
      (tally-countries (convert-kiva-data (get-kiva-pages 1 20)))))
```

Uganda Philippines Tajikistan Pakistan Cambodia El Salvador
Bolivia Kenya Timor-Leste Mali Colombia