A Database Abstraction for Data-intensive Social Applications

Atsuyuki Morishima
University of Tsukuba

Joint work with Noriaki Anzai

Background

• Applications of Social/Human Computation are emerging
  – ESP Games
  – Q&A Sites
  – Wikipedia
  – Price Information Sites
  – Twitter
  – reCAPCHA
• Most of such applications are *data-intensive*. 
A Natural (and Important) Question

What is a good abstraction to describe (and program) such data-intensive applications of social/human computation?

A possibility: Since they are data-intensive, database languages may work well.

However ...

Current DBMSs are used merely as a storage facility.
Example: StatusNet
  – A Twitter Clone
  – Implemented with PHP and MySQL
  – Only 218 out of more than 20K lines are written in SQL

This suggests the limitation of the current database abstraction when used for such applications.
Our Approach

• We extend the notion of information spaces to *cybernetic information spaces* where people are considered to be information sources.

• We introduce *open* predicates into Datalog and let people evaluate the open predicates.

InfoSpaces to *Cybernetic InfoSpaces*
CyLog: A Datalog-like Language Executed in Cybernetic InfoSpaces

- Open Predicates/Attribute Values
- Game Aggregation
- Data-Move-Payoff Abstraction

Introducing Open Predicates/Attribute Values

```
closed(Member);
AddressList(ID, Name, Address, Tel) <- Member(ID, Name)
PaperThisYear(pid, title) <- Paper(pid, title, year)
```

- Predicate $p$ is “open” if $p$ is not “closed”.
- Predicate $p$ is “closed” if:
  - $p$ is declared as “closed”, or
  - $p$ is derived only from closed predicates.
- Value $v$ of attribute $a$ is open if:
  - $a$ is not bound to any value, or
  - $v$ is NULL.
- Open predicates/attribute values are evaluated by people.
Evaluation of Open Predicates/Values

AddressList(ID, Name, Address, Tel) ← Member(ID, Name);
PaperThisYear(pid, title) ← Paper(pid, title, year);

Evaluating Open Predicates/Attribute Values

People

CyLog Interpreter

Web Server

Mail Server/Twitter

Generated Forms

Generated Relations Web Pages

Queries

Results

Payoff

Programs

Data

Queries

Server

Relations

Generated Web Pages
Challenges

- Who should we ask for giving the answer?
  → Find people based on the stored data

- How can we expect that we will get what you want?
  → Mechanisms based on the game theory

Finding People based on the Stored Data:
Our first attempt

Currently, a simple approach is taken.

1. Use the player name embedded in the rule
   \[
   \text{AnzaiPapers}(ID, \text{Author}, \text{Title}) \leftarrow \text{PaperAuthor}(ID, \text{Author}), \text{Author} = \text{“anzai”};
   \]

2. Use the player names embedded in the relation appearing in the rule body
   \[
   \text{AddressList}(ID, \text{Name}, \text{Address}, \text{Tel}) \leftarrow \text{Member}(ID, \text{Name});
   \]

3. Find the player name by joining relations using referential keys
4. Optional: Broadcast the query to all members
   (In addition, players can suggest the names of other players in the entry form)
Obtaining What You Want

• CyLog introduces *Game Aggregations* to make people evaluate the open predicate/attributes appropriately if the people behave rationally.

• CyLog supports the *DMP (Data-Move-Payoff) abstraction* as a top-level abstraction of Cybernetic Datalog Programs.

---

A Game

• A game can be described with players, their options, and payoffs
• Example: A payoff table of the Prisoner’s Dilemma

<table>
<thead>
<tr>
<th>Player A</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>(-1, -1)</td>
<td>(-10,0)</td>
</tr>
<tr>
<td>Defect</td>
<td>(0,-10)</td>
<td>(-5,-5)</td>
</tr>
</tbody>
</table>

This suggests the importance of game design.
### Payoff Table of the ESP Game

<table>
<thead>
<tr>
<th>Player A \ Player B</th>
<th>Term A</th>
<th>Term B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term A</td>
<td>(1, 1)</td>
<td>(0, 0)</td>
</tr>
<tr>
<td>Term B</td>
<td>(0, 0)</td>
<td>(1, 1)</td>
</tr>
</tbody>
</table>

Solution:

### Payoff Table of the Q&A Game

<table>
<thead>
<tr>
<th>Player A \ Player B</th>
<th>Best Answer</th>
<th>Third best Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Second Best Answer</td>
<td>(10, 20)</td>
<td>(20, 10)</td>
</tr>
<tr>
<td>Worst Answer</td>
<td>(10, 20)</td>
<td>(10, 20)</td>
</tr>
</tbody>
</table>

Solution:
Payoff Table of the Janken Game
(가위바위보)

<table>
<thead>
<tr>
<th>Player A \ B</th>
<th>Goo</th>
<th>Choki</th>
<th>Pa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goo</td>
<td>(0, 0)</td>
<td>(1,-1)</td>
<td>(-1,1)</td>
</tr>
<tr>
<td>Choki</td>
<td>(-1,1)</td>
<td>(0,0)</td>
<td>(1,-1)</td>
</tr>
<tr>
<td>Pa</td>
<td>(1,-1)</td>
<td>(-1,1)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

No Solution

Sequential Games
(vs. Simultaneous-Move Games)
Data-oriented Games: A Classification

- Contribution Game:
  - A player is rewarded if his/her input is adopted as a value for the database.
- Evaluation Game
  - A player is rewarded according to the “value” of his/her input.

Social Applications as Games: Examples

<table>
<thead>
<tr>
<th></th>
<th>Contribution Game</th>
<th>Evaluation Game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simultaneous-Move Game</td>
<td>ESP</td>
<td>(Pure) Q&amp;A evaluated in terms of given points</td>
</tr>
<tr>
<td>Sequential Game</td>
<td>Wikipedia</td>
<td>Twitter evaluated in terms of the number of followers</td>
</tr>
</tbody>
</table>
Game Aggregation

Game Aggregation
– represents a game with an output, and
– modeled as a mapping from the Input relation
  \( R(\text{order}, \text{player}, \text{move}) \) to the payoff relation \( S(\text{player}, \text{payoff}, \text{value}) \)

Ex)

<table>
<thead>
<tr>
<th>Player A \ B</th>
<th>Goo</th>
<th>Choki</th>
<th>pa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goo</td>
<td>(0, 0) Tie</td>
<td>(1,-1) A Wins</td>
<td>(-1,1) B Wins</td>
</tr>
<tr>
<td>Choki</td>
<td>(-1,1) B Wins</td>
<td>(0,0) Tie</td>
<td>(1,-1) B Wins</td>
</tr>
<tr>
<td>pa</td>
<td>(1,-1) A Wins</td>
<td>(-1,1) B Wins</td>
<td>(0,0) Tie</td>
</tr>
</tbody>
</table>

Example: Game Aggregation using the “Janken” game

<table>
<thead>
<tr>
<th>Player A \ B</th>
<th>Goo</th>
<th>Choki</th>
<th>Pa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goo</td>
<td>(0, 0) Tie</td>
<td>(1,-1) A Wins</td>
<td>(-1,1) B Wins</td>
</tr>
<tr>
<td>Choki</td>
<td>(-1,1) B Wins</td>
<td>(0,0) Tie</td>
<td>(1,-1) B Wins</td>
</tr>
<tr>
<td>Pa</td>
<td>(1,-1) A Wins</td>
<td>(-1,1) B Wins</td>
<td>(0,0) Tie</td>
</tr>
</tbody>
</table>

Input Relation: Data_input

<table>
<thead>
<tr>
<th>Order</th>
<th>Player</th>
<th>Move</th>
<th>to</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Goo</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>Choki</td>
<td>B</td>
</tr>
</tbody>
</table>

Data_input@Janken

<table>
<thead>
<tr>
<th>Player</th>
<th>Payoff</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>A Wins</td>
</tr>
<tr>
<td>B</td>
<td>-1</td>
<td>A Wins</td>
</tr>
</tbody>
</table>
Built-in Game Aggregations (1/2)

Duplicates

<table>
<thead>
<tr>
<th>Player A \ B</th>
<th>Term A</th>
<th>Term B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term A</td>
<td>(1,1)</td>
<td>(0,0)</td>
</tr>
<tr>
<td>Term B</td>
<td>(0,0)</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>

Data_input

<table>
<thead>
<tr>
<th>Order</th>
<th>Player</th>
<th>Move</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>apple</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>banana</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>apple</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>orange</td>
</tr>
</tbody>
</table>

Data_input@Duplicate

<table>
<thead>
<tr>
<th>Player</th>
<th>Payoff</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>apple</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>apple</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>apple</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>apple</td>
</tr>
</tbody>
</table>

Built-in Game Aggregations (2/2)

The following game aggregations are different to each other in what are chosen for the output values. The players who gave the value are rewarded by payoff points.

- Intersection (Values given by everyone)
- Union (All values given by any player)
- Duplicates (Values given by more than one player)
- Majority (Values given by the largest number of people)
- Unique (Values given by only one person)
- First (The value given first)
DMP Abstraction

A top-level abstraction to integrate components of CyLog programs

Data:
(Datalog rules with game aggregations)

Move:
(Constraints on the timing of inputs by players)

Payoff:
(Payoff expressions with aggregation functions, including game aggregations)

Example: A Simple ESP Game

Data:
Metadata(x, metadata) /duplicate <- File(x);

Move:
Active;

Payoff:
Metadata_input@duplicate;
InfoSpace Builder

- A RAD (Rapid Application Development) tool to design and implement Cybernetic InfoSpace Programs
- Language: CyLog
- Evaluation Engine: Chimera Interpreter
- Template Generator: Chimera Compiler

Executed by the machines and people in the Cybernetic InfoSpace

A Simple Version of Gas Price Site

Data:

\[
\text{GSList}(id: \text{auto}(), \text{shopname}, \text{address}) \leftarrow \text{Member}(id: \text{user}); \\
\text{PriceList}(uid: \text{user}, sid: \text{shop}, \text{price}, date: \text{now}()) \leftarrow \\
\quad \text{Member}(id: \text{user}), \text{GSList}(id: \text{shop});
\]

Move: Passive;
Payoff: \text{GSList\_input}\text{@contribution};

http://gogo.gs/
A Simple Version of Twitter

Data:

- foreign-key(Follow.dest, Member.id);
- Tweet(uid:user, date:now(), message) <- Member(id:user);
- Follow(source:user, dest) <- Member(id:user);

Move: Passive;
Payoff: Follow@Count;

http://twitter.com/

Other Applications for which Rapid Development with InfoSpace Builder is Applicable:

There are many applications where:

- it is the key function to generate, collect, maintain, and retrieve data, and
- it is important to exploit the power of people and machines for the application to work well

Ex) Maintenance of publication lists, metadata generation and maintenance, Q&A knowledge base, opinion collection tools, dataflow management, reCapcha-like applications, Communication tools, Maintenance of member lists, Integrated query to people and databases, etc.
Related Work (1/4)

Open-world DATALOG [1]
- A Datalog that does not adopt the closed world assumption.
- When it finds no answer in the database, it returns “unknown” instead of “no.” It does not consider to search other places for the answer.

CyLog asks people to evaluate open predicates/values and tries to get appropriate answers.


Related Work (2/4)

- Amazon Mechanical Turk [2]
  - A marketplace for work that requires human intelligence, providing APIs.
- Google Docs [3]
  - Has “Google form” function to easily make Web forms to collect data.

CyLog is not merely providing APIs or particular functions, but provides an principled abstraction used as a language for design, discussion, optimization, and implementation.

Related Work (3/4)

Human Computation, GWAP [4]
   – Seeks to exploit human abilities to solve computationally difficult problems

• We show that a database abstraction can be a basis of a declarative language to discuss, design, and implement data-intensive applications of Human/Social Computation
• Provide Logic-based, fine-grained integration of computations by machines and people


Related Work (4/4)

• Hybrid Computers [5]
   – Combining analogue and digital computers
• Heterogeneous Computing [6]
   – Combining general-purpose processors and special-purpose processors such as GPUs

We integrate digital and human/social computers

Summary

• InfoSpace Builder: A RAD tool for applications of data-intensive cybernetic computation, based on declarative descriptions.
• CyLog: A declarative language for data-intensive applications, featuring:
  – Execution in Cybernetic InfoSpaces
  – Open predicates/attribute values
  – Game aggregations, and
  – DMP abstraction
• Chimera: A Prototype Interpreter/Code Generator based on the CyLog Programs

Open Problems

• Optimization Issues
• Advanced mechanism for player selection
• Development of various types of data-oriented game elements
• Design theory
• Definitive rationality
• Non-atomic values