Constraint-based Information Integration

Craig Knoblock
University of Southern California

Thanks to Jose Luis for some of these slides
Constraint Satisfaction and Propagation for Integration

- Integrating data from multiple sources often involves reasoning about the information
- Constraints provide a approach to expressing relationships and filtering data
Outline

Part I

- Constraint satisfaction in SmartClients
- Constraint propagation in Heracles

Part II (Dr. Jose Luis Ambite)

- Building your own applications in Heracles
SmartClients [Torrens et al, 2002]

- Cast an integration problem as a Constraint Satisfaction Problem (CSP)
- Given a request, the server retrieves the required data and sends the data and the CSP to the client
- Client solves the CSP locally
  - Large complex problem transmitted in small amount of space
  - Provides fine-grained user interaction with the data
Example Problem

I live in Bern, Switzerland, and would like to visit colleagues in Princeton (New Jersey), and London. I would like to spend at least two days in each place, and will need to travel in the first two weeks of February.

1. 1\textsuperscript{st} leg from Bern to Princeton: flights from ZRH/BSL/CVA to JFK/EWR/PHL on the dates from 1\textsuperscript{st} to 10\textsuperscript{th} February,

2. 2\textsuperscript{nd} leg from Princeton to London: flights from JFK/EWR/PHL to LGW/LHR/LCY on the dates from 4\textsuperscript{th} to 12\textsuperscript{th} February, and

3. 3\textsuperscript{rd} leg from London to Bern: flights from LGW/LHR/LCY to ZRH/BSL/CVA on the dates from 6\textsuperscript{st} to 14\textsuperscript{th} February.
Constraint Satisfaction Problem

- CSP
  - Variables
  - Domains for each variable
  - Constraints on variables
    - Supports both hard and soft constraints

- Algorithms and techniques for solving CSPs have been widely studied
Example CSP Graph

Variables $Arr_{T_1}$, $Dep_{T_1}$, Start-Time, and End-Time contain information about Dates and Times
Formalization of Example: Variables

\[ X = \{DT_0, ..., DT_{n-1}, AT_0, ..., AT_{n-1}, Airports_0, ..., Airports_n, Flights_0, ..., Flights_{n-1}, AirCrafts, Fares, Airlines, ... \} \]

is a set of variables. There are several kinds of variables:

- \( DT_i \) and \( AT_i \) represent the dates and times on which the traveler could depart and arrive respectively.
- \( Airports_i \) represents the possible airports near the departure for leg \( i \) of the itinerary.
- \( Flights_i \) stands for the possible flights in between the airports of \( Airports_i \) and \( Airports_{i+1} \).
Formalization of Example:
Domains

\[ D = \{D_1, \ldots, D_n\} \] is the set of domains. There are several kinds of domains depending on the type of the associated variable:

- For variables \( DT_i \) or \( AT_i \): the domain contains all possible departure and arrival times for the \( leg_i \).
- For variables \( Airports_i \): the domain is a set of airports for the departure of the \( leg_i \).
- For variables \( Flights_i \): the domain is the set of possible flights from \( Airports_i \) to \( Airports_{i+1} \).
- For variables \( AirCrafts, Fares \) and \( Airlines \): the domain is the set of different aircrafts, the set of available fares or the set of airline companies respectively.
Formalization of Example: Constraints

\( C = \{C_1, \ldots, C_k\} \) is the set of constraints. Basically, there are two kinds of constraints: those imposed by the user’s preferences and those imposed by flight schedules. There are constraints on the variables \( \text{Flights}_i, \text{Airports}_i, DT_i \) and \( AT_i \) that guarantee that the flight is compatible with the airports, departures times and arrival times. A binary constraint in between \( AT_i \) and \( DT_{i+1} \) takes into consideration that the flight for \( \text{leg}_{i+1} \) departs after the flight for \( \text{leg}_i \) arrives. Then most of the user’s preferences are expressed by means of constraints between \( \text{Flight}_i \) variables and \( \text{Aircrafts}, \text{Airlines}, \text{Fares} \) and other variables.
Constraint Graph for Example
Architecture for SmartClients

- Database with information for the CSP
- SERVER: Generation of CSP
- CLIENT: Resolution of CSP

Initial specification
smart agent
Pros and Cons

- **Pros**
  - Elegant approach that exploits past work on CSPs
  - Minimizes the data retrieval and supports complex reasoning and integration of the data

- **Cons**
  - Assumes that all data can be retrieved before any reasoning about the data
  - In the travel planning, assumes that prices are the same on any date and there are no issues with flight availability
Heracles Constraint Integration Framework

- Framework for building integrated applications
- Interleaves planning and information gathering
- Uses a constraint reasoner to decide what sources to query and to integrate the results
The Travel Assistant

Meeting With: Jim Hendler
Company Name: DARPA

Subject: CoABS PI Meeting
Location: Washington, DC

Starting At: Feb 16, 2001, 01:00 PM
Ending At: Feb 18, 2001, 03:00 PM

Leaving From: 2700 University Park, Los Angeles, CA
Traveling To: 1120 19th ST NW, Washington, DC

Destination weather: Partly Cloudy

Distance (miles): 2294

Mode to Destination: Fly

Click to Expand
Dynamically Updates Slots as Information Becomes Available
Supports Informed Choices
Changes Propagate Throughout
User Can Specify High-Level Preferences
Constraint Networks for Integrating Information

- Components:
  - Representation of the variables
  - Representation of constraints
  - Hierarchical template representation
  - Constraint propagation and cycle detection
Constraint Networks for Integrating Information

- Components:
  - Representation of the variables
  - Representation of constraints
  - Hierarchical template representation
  - Constraint propagation and cycle detection
Constraint Variables

- Constraint network consists of a set of variables such as:
  - MeetingStartTime
  - MeetingLocation

- Each variable depends on a set of ancestors.

- Variables are related by constraints that determine the possible values of a solution.
Constraint Networks for Integrating Information

Components:
- Representation of the variables
- Representation of constraints
- Hierarchical template representation
- Constraint propagation and cycle detection
Constraint Representation

- Constraints are computable components:
  - Local calculations based on Xquery
    - MeetingStartTime + MeetingDuration --> MeetingEndTime
  - Web and Database Wrappers
    - ITN: DepartureAirport, ArrivalAirport, Date --> Flights
    - Yahoo Weather: City, Date --> Weather predication
  - External Programs (Outlook, Planners, etc)
    - Outlook Calendar: Date --> Meetings
Constraint Structure

Constraint

- Arguments: input and output variables
- Call:
  - Construct table with inputs and corresponding calls (http requests, SQL queries, etc) to sources (wrappers, DBs, etc) [using XML Query]
  - Calls are executed and results stored in an table
- Output
  - Restructure source results into desired output [using XML Query]
Drive or Take a Taxi?

- Departure Date: Sep 30, 2000
- Return Date: Oct 2, 2000
- Departure Airport: LAX
- Distance: 15.1 miles
- Parking Rate: $7.00/day
- Parking Total: $21.00
- Taxi Fare: $23.00
- Select Mode to Airport: Drive
Constraint Networks for Integrating Information

Components:
- Representation of the variables
- Representation of constraints
- Hierarchical template representation
- Constraint propagation and cycle detection
Hierarchically-Partitioned Constraint Networks

- Template:
  - Groups related variables and constraints
  - Organizes information for computation and presentation to user

- Templates organized hierarchically
  - Template decomposed into subtemplates
  - Choose among alternative subtemplates
Template Structure

Template

- Arguments: input and output variables
- Variables: name, type, default values
- Constraints
- Expansions: alternative subtemplate calls
- GUI specification
Partitioned Constraint Network

Who
Subject
Starting Time
Ending Time
Company
Dest. Addr.
Origin Addr.
Dest Weather
Origin Weather
Distance
Travel Mode
Depart Time
Depart Airport
Arrival Time
Dist. toAirport
Flight Num
Parking Lot
Parking Rate
Arrival Airport
Taxi Fare
Mode toAirport
Template Hierarchy for the Travel Assistant

```
Trip
  1       AND
  |       |
  |       |
ModeToDestination
  |       OR
  |       |
  Drive  Fly  Taxi

  2
  |
  ModeHotel
  | OR |
  | Hotel  NoOvernight |

  3
  | OR |
  | ModeNext |

  EndTrip
  Trip (Return Home)
  Trip (Return Office)
  Trip (New Leg)

Trip (Return Home)
Trip (Return Office)
Trip (New Leg)

ModeToAirport
  | OR |
  | Drive  Taxi |

FlightDetail
ModeFromAirport
  | OR |
  | Drive  Taxi |
```
Dynamic Constraint Networks

Generalization of Constraint Networks

- Variables can be active or inactive
- Normal Constraints
  \[ x_1 = k_1 \land \ldots \land x_m = k_m \rightarrow x_n = k_n \]
- Activity constraints:
  \[ x_1 = k_1 \land \ldots \land x_m = k_m \rightarrow \text{active}(x_n) \]
- Inactive variables do not participate in the network, i.e., do not propagate constraints
Heracles: Template Selection

- Core network
  - Computes values of template selection vars
  - Always active

- Template selection variables
  - Inputs to activity constraints: determine the choice of subtemplates, i.e., which additional variables are active
Constraint Networks for Integrating Information

- Components:
  - Representation of the variables
  - Representation of constraints
  - Hierarchical template representation
  - Constraint propagation and cycle detection
Constraint Propagation

- **Approach**
  - When a variable is assigned a value, re-compute the value sets and assigned values of all dependent variables
  - Proceeds recursively until no values are changed or a cycle is detected

- **Core network**
  - Propagates all variables through the core network
  - Remaining variables are computing when a template is opened

- **Does not perform full CSP**
  - Less costly
  - Does not require all information in advance
  - Makes choices locally, so may fail to find optimal assignment
Cycle Detection

- Address cyclic *interactive* networks
  - Multiple input paths:
    - region/country/city vs. lat/long
  - Conversion rounding errors:
    - lat/long, temperature, …

=> Cycle detection in constraint propagation
Interactive Cyclic Network

C3: Closest City

C1: Find City Center

C2: Copy

City (v1)

City center (v2)

Lat/Long (v3)
Interactive Cyclic Network

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>Ø</td>
<td>LA</td>
</tr>
</tbody>
</table>

- **C1: Find City Center**
- **C2: Copy**
- **C3: Closest City**
Interactive Cyclic Network

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>Ø</td>
<td>LA</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>v1</td>
<td>34N118W</td>
</tr>
</tbody>
</table>
Interactive Cyclic Network

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>∅</td>
<td>LA</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>v1</td>
<td>34N118W</td>
</tr>
<tr>
<td>3)</td>
<td>1</td>
<td>v1 v2</td>
<td>34N118W</td>
</tr>
</tbody>
</table>
Interactive Cyclic Network

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>Ø</td>
<td>LA</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>v1</td>
<td>34N118W</td>
</tr>
<tr>
<td>3)</td>
<td>1</td>
<td>v1 v2</td>
<td>34N118W</td>
</tr>
<tr>
<td>4) Blocked!</td>
<td></td>
<td>v1 ∈ vis(v3)</td>
<td></td>
</tr>
</tbody>
</table>
Interactive Cyclic Network

C1: Find City Center
C2: Copy
C3: Closest City

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>Ø</td>
<td>LA</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>v1</td>
<td>34N118W</td>
</tr>
<tr>
<td>3)</td>
<td>1</td>
<td>v1 v2</td>
<td>34N118W</td>
</tr>
<tr>
<td>4)</td>
<td>Blocked!</td>
<td>t(v3) = t(v1) ^ v1 ∈ vis(v3)</td>
<td></td>
</tr>
<tr>
<td>5)</td>
<td>2</td>
<td>Ø</td>
<td>40N70W</td>
</tr>
</tbody>
</table>
Interactive Cyclic Network

**Sequence of events** | **User time** | **Visited Variables** | **Value**
--- | --- | --- | ---
1) | 1 | ∅ | LA
4) Blocked! | t(v3) = t(v1) \( \land v1 \in vis(v3) \)
6) | 2 | v3 | NY

**City (v1)**
- C1: Find City Center
- C2: Copy
- C3: Closest City

**City center (v2)**
- Lat/Long (v3)

**Lat/Long (v3)**
- 3) 1 v1 v2 34N118W
- 5) 2 ∅ 40N70W
**Interactive Cyclic Network**

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>∅</td>
<td>LA</td>
</tr>
<tr>
<td>4)</td>
<td>Blocked!</td>
<td>$t(v_3) = t(v_1) \land v_1 \in vis(v_3)$</td>
<td></td>
</tr>
<tr>
<td>6)</td>
<td>2</td>
<td>$v_3$</td>
<td>NY</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>$v_1$</td>
<td>34N118W</td>
</tr>
<tr>
<td>7)</td>
<td>2</td>
<td>$v_3 \land v_1$</td>
<td>40N73W</td>
</tr>
<tr>
<td>3)</td>
<td>1</td>
<td>$v_1 \land v_2$</td>
<td>34N118W</td>
</tr>
<tr>
<td>5)</td>
<td>2</td>
<td>∅</td>
<td>40N70W</td>
</tr>
</tbody>
</table>

- **C1: Find City Center**
- **C2: Copy**
- **C3: Closest City**
### Interactive Cyclic Network

<table>
<thead>
<tr>
<th>Sequence of events</th>
<th>User time</th>
<th>Visited Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>1</td>
<td>∅</td>
<td>LA</td>
</tr>
<tr>
<td>4)</td>
<td>Blocked!</td>
<td>$t(v3) = t(v1)^v1 \in \text{vis}(v3)$</td>
<td></td>
</tr>
<tr>
<td>6)</td>
<td>2</td>
<td>v3</td>
<td>NY</td>
</tr>
<tr>
<td>2)</td>
<td>1</td>
<td>v1</td>
<td>34N118W</td>
</tr>
<tr>
<td>7)</td>
<td>2</td>
<td>v3 v1</td>
<td>40N73W</td>
</tr>
<tr>
<td>3)</td>
<td>1</td>
<td>v1 v2</td>
<td>34N118W</td>
</tr>
<tr>
<td>5)</td>
<td>2</td>
<td>∅</td>
<td>40N70W</td>
</tr>
<tr>
<td>8)</td>
<td>Blocked!</td>
<td>$t(v3) = t(v2)^v3 \in \text{vis}(v2)$</td>
<td></td>
</tr>
</tbody>
</table>
Discussion

- General framework for interleaving planning and information gathering
  - Retrieves information as needed
  - Gathers and integrates data in a uniform framework
  - Evaluates tradeoffs and selects among alternatives
  - Allows the users to explore alternatives
  - Supports a wide variety of information types: databases, web pages, images, video, etc.