Logic Programming

Worksheets

Michael Genesereth
Computer Science Department
Stanford University
Meaningful Data Display
   All data readily accessible
   Tables, Charts, Graphs

Modifiability
   What-you-see-is-what-you-get
   Random access - data can be changed in any order

Constraint Checking
   Completeness and Consistency
   Problem alerting and Guidance in solving

Automatic Computation of Results
   Consequences computed
   Presentation automatically updated
**DEPARTMENT OF COMPUTER SCIENCE**

**MSCS Program Sheet (2010-11)**

**Primary Specialization**

- Artificial Intelligence

---

**Name:** Charles Pamell Naut  
**Advisor:**  
**Student ID #:**  
**Email:** cnaut@stanford.edu  
**Proposed date for degree conferral:**  
**Date:** 10/8/2010

---

**GENERAL INSTRUCTIONS**

Before the end of your first quarter, you should complete the following steps. Detailed instructions are included in the Guide to the MSCS Program Sheet in your orientation packet (an online version is available at es.stanford.edu/degrees/mscs/programsheets/):

- Complete this program sheet by filling in the number, name and units of each course you intend to use for your degree.
- Create a course schedule showing the year and quarter in which you intend to take each course in your program sheet.
- Meet with your adviser and secure the necessary signatures on the program sheet.

---

**FOUNDATIONS REQUIREMENT**

You must satisfy the requirements listed in each of the following areas; all courses taken elsewhere must be approved by your adviser on a foundation course waiver form. Required documents for waiving a course include course descriptions, syllabi, and textbook lists. These documents can be organized here: es.stanford.edu/degrees/mscs/waivers/. Do not enter anything in the “Units” column for courses taken elsewhere.

**Note:** If you are amending an old program sheet, enter "on file" in the approval column for courses that have already been approved.

<table>
<thead>
<tr>
<th>Required</th>
<th>Equivalent elsewhere (course number/title/institution)</th>
<th>Approval</th>
<th>Grade</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic, Automata and Complexity (CS 103)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probability (CS 109, STATS 116, CME 106, or MS&amp;E 220)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithmic Analysis (CS 161)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Organization and Systems (CS 107)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Principles of Computer Systems (CS 110)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT:** 10

**Note:** This total may not exceed 10 units.

- **7 Requirements Left**  
- **Total Units:** 10  
- **Status:** Draft
Current Approach

The Big 5
Do you master them all?
Do It Yourself!

Worksheets :: Spreadsheets
Page Data
Document Object Model (DOM)
<center>
<input id='mynode'
type='text'
value='hello'
size='30'
style='color:black'/>
</center>
value(widget, value) - true whenever the value associated with widget is value. The widget here may be a text field, selector, radio button field, slider, and so forth.

holds(widget, value) - true whenever one of the values associated with the multi-valued node widget. The widget in this case is a multi-valued selector or a checkbox field.

attribute(widget, property, value) - true whenever the property attribute of widget is value.

style(widget, property, value) - true whenever the property style of widget is value.

innerhtml(widget, content) - true whenever the innerHTML of widget is content. The content is typically a string of characters.
DOM:

```html
<center>
  <input id='mynode'
         type='text'
         value='hello'
         size='30'
         style='color:black'/>
</center>
```

Dataset:

```plaintext
value(mynode,hello)
attribute(mynode,size,30)
style(mynode,color,black)
```
DOM:

```html
<center>
<input id='mynode'
    type='text'
    value='hello'
    size='30'
    style='color: black'/>
</center>
```

Dataset:

- `value(mynode, hello)`
- `attribute(mynode, size, 30)`
- `style(mynode, color, black)`
Gestures
Gestures performed by the user:
  Making a selection from drop-down list
  Changing value of text field
  Clicking a button

Automatic Actions:
  Loading a page
  Clock tick
**Operations**

`select(widget, value)`: This action occurs when the user enters or selects `value` as the value of `widget`.

`deselect(widget, value)`: This action occurs when the user erases or deselects `value` as the value of `widget`.

`click(widget)`: This action occurs when the user clicks on `widget`.

`tick`: This action occurs periodically. By default, it happens once per second.

`load`: This occurs when a page is first loaded.

`unload`: This action occurs when a user leaves a page.
Example

DOM:
<center>
  <input id='mynode' type='text' value='hello' -> goodbye
  size='30'
  style='color:black'
  onchange='modtext(this)'/>
</center>

Resulting Action:
select(mynode, goodbye)
Operation Definitions
Buttons

- click(orange) :: style(page, color, orange)
- click(blue) :: style(page, color, blue)
- click(purple) :: style(page, color, purple)
- click(black) :: style(page, color, black)

click(X) :: style(page, color, X)

click(X) ::
    style(page, color, Y) & distinct(X, Y)
=> ~style(page, color, Y)
select(pagecolor, X) :: style(page, color, X)
select(pagecolor, X) ::
    style(page, color, Y) ==> ~style(page, color, Y)

select(pagecolor, X) :: value(pagecolor, X)
select(pagecolor, X) ::
    value(pagecolor, Y) ==> ~value(pagecolor, Y)
click(X) :: value(pagecolor,X)
click(X) ::
    value(pagecolor,Y) ==> ~value(pagecolor,Y)
View Definitions
Alternative Approaches

Direct Approach:
Every gesture directly changes visible features.

View Definitions:
Every gesture changes dataset.
Visible features defined as views of the dataset
and are changed implicitly in Worksheets code.
Changing color:

\[ \text{click}(X) :: \text{style}(\text{page}, \text{color}, X) \]
\[ \text{select}(\text{pagecolor}, X) :: \text{style}(\text{page}, \text{color}, X) \]
\[ \text{select}(\text{pagecolor}, X) :: \]
\[ \text{style}(\text{page}, \text{color}, Y) \implies \neg \text{style}(\text{page}, \text{color}, Y) \]

Changing selector value:

\[ \text{click}(X) :: \text{value}(\text{pagecolor}, X) \]
\[ \text{click}(X) :: \]
\[ \text{value}(\text{pagecolor}, Y) \implies \neg \text{value}(\text{pagecolor}, Y) \]
\[ \text{select}(\text{pagecolor}, X) :: \text{value}(\text{pagecolor}, X) \]
\[ \text{select}(\text{pagecolor}, X) :: \]
\[ \text{value}(\text{pagecolor}, Y) \implies \neg \text{value}(\text{pagecolor}, Y) \]
Changing selector value:

\[
\text{click}(X) :: \text{value}(\text{pagecolor}, X)
\]
\[
\text{click}(X) :: \text{value}(\text{pagecolor}, Y) \implies \neg \text{value}(\text{pagecolor}, Y)
\]
\[
\text{select}(\text{pagecolor}, X) :: \text{value}(\text{pagecolor}, X)
\]
\[
\text{select}(\text{pagecolor}, X) :: \text{value}(\text{pagecolor}, Y) \implies \neg \text{value}(\text{pagecolor}, Y)
\]

View Definition:

\[
\text{style}(\text{page}, \text{color}, X) :: \text{value}(\text{pagecolor}, X)
\]
Putting It All Together
Converting Web Pages to Worksheets

Start with an HTML page.

(1) Add worksheets code.
(2) Initialize.
(3) Add identifiers and event handlers.
(4) Add Data and Rules.

Done.
<html>
  <head>
  </head>
  <body>
    <input type='button' value='orange'/>
    <input type='button' value='purple'/>
    <input type='button' value='black'/>
    <br/>Some text.<br/>
    <select>
      <option>orange</option>
      <option>purple</option>
      <option>black</option>
    </select>
  </body>
</html>
<html>
  <head>
    <script type='text/javascript'
      src='http://epilog.stanford.edu/javascript/epilog.js'/>
    <script type='text/javascript'
      src='http://worksheets.stanford.edu/javascript/worksheets.js'/>
  </head>
  <body>
    <input type='button' value='orange'/>
    <input type='button' value='purple'/>
    <input type='button' value='black'/>
    <br/>Some text.<br/>
    <select>
      <option>orange</option>
      <option>purple</option>
      <option>black</option>
    </select>
  </body>
</html>
<html>
  <head>
    <script type='text/javascript'
      src='http://epilog.stanford.edu/javascript/epilog.js'/>
    <script type='text/javascript'
      src='http://worksheets.stanford.edu/javascript/worksheets.js'/>
  </head>
  <body onload='initialize()'>
    <input type='button' value='orange'/>
    <input type='button' value='purple'/>
    <input type='button' value='black'/>
    <br/>Some text.<br/>
    <select>
      <option>orange</option>
      <option>purple</option>
      <option>black</option>
    </select>
  </body>
</html>
Add Identifiers and Event Handlers

<html>
  <head>
    <script type='text/javascript'
      src='http://epilog.stanford.edu/javascript/epilog.js'/>
    <script type='text/javascript'
      src='http://worksheets.stanford.edu/javascript/worksheets.js'/>
  </head>
  <body id='page' onload='initialize()'>
    <input type='button' value='orange' id='orange' onclick='modbutton(this)'/>
    <input type='button' value='purple' id='purple' onclick='modbutton(this)'/>
    <input type='button' value='black' id='black' onclick='modbutton(this)'/>
    <br/>Some text.<br/>
    <select id='pagecolor' onchange='modselector(this)'>
      <option>orange</option>
      <option>purple</option>
      <option>black</option>
    </select>
  </body>
</html>
<html>
<head>
    <script type='text/javascript'
        src='http://epilog.stanford.edu/javascript/epilog.js'/>
    <script type='text/javascript'
        src='http://worksheets.stanford.edu/javascript/worksheets.js'/>
</head>
<body id='page' onload='initialize()'>
    <input type='button' value='orange' id='orange' onclick='modbutton(this)'/>
    <input type='button' value='purple' id='purple' onclick='modbutton(this)'/>
    <input type='button' value='black' id='black' onclick='modbutton(this)'/>
    <br/>Some text.<br/>
    <select id='pagecolor' onchange='modselector(this)'>
        <option>orange</option>
        <option>purple</option>
        <option>black</option>
    </select>
    <textarea id='lambda' style='display:none'></textarea>
    <textarea id='library' style='display:none'>...</textarea>
</body>
</html>
Some text
Remarks
DEPARTMENT OF COMPUTER SCIENCE
MSCS Program Sheet (2010-11)

Primary Specialization

Name: Charles Pamell Naut  Advisor:  Proposed date for degree conferral:  Date: 10/8/2010
Student ID #:  Email: cnauj@stanford.edu  HCP?  Coterm?

GENERAL INSTRUCTIONS
Before the end of your first quarter, you should complete the following steps. Detailed instructions are included in the Guide to the MSCS Program Sheet in your orientation packet (an online version is available at es.stanford.edu/degrees/mscs/programsheets/):

- Complete this program sheet by filling in the number, name and units of each course you intend to use for your degree.
- Create a course schedule showing the year and quarter in which you intend to take each course in your program sheet.
- Meet with your adviser and secure the necessary signatures on the program sheet.

FOUNDATIONS REQUIREMENT
You must satisfy the requirements listed in each of the following areas; all courses taken elsewhere must be approved by your adviser on a foundation course waiver form. Required documents for waiving a course include course descriptions, syllabi, and textbook lists. These document can be organized here: es.stanford.edu/degrees/mscs/waivers/. Do not enter anything in the “Units” column for courses taken elsewhere.

Note: If you are amending an old program sheet, enter "on file" in the approval column for courses that have already been approved.

Required:
Logic, Automata and Complexity (CS 103)

Probability (CS 109, STATS 116, CMB 106, or MS&E 220)

Algorithmic Analysis (CS 161)

Computer Organization and Systems (CS 107)

Principles of Computer Systems (CS 110)

Equivalent elsewhere (course number/title/institution)  Approval  Grade  Units

TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT: 10

Note: This total may not exceed 10 units.
Worksheets = Interactive Webpages

changes checked for compliance with constraints
consequences computed and display updated
APPLICATION FOR DISABILITY COMPENSATION
AND RELATED COMPENSATION BENEFITS

VA DATE STAMP
(Do not write in this space)

Important: Please read the Privacy Act and Respondent Burden on page 10 before completing the form.

SECTION I: IDENTIFICATION AND CLAIM INFORMATION

1. VETERAN/SERVICE MEMBER NAME (First, Middle Initial, Last)

2. VETERAN’S SOCIAL SECURITY NUMBER

3. HAVE YOU EVER FILED A CLAIM WITH VA?
   - YES
   - NO

4. VA FILE NUMBER

5. DATE OF BIRTH (MM/DD/YYYY)

6. SEX
   - MALE
   - FEMALE

7. VETERAN’S SERVICE NUMBER (If applicable)

8A. ARE YOU CURRENTLY HOMELESS OR AT RISK OF BECOMING HOMELESS?
   - YES
   - NO

8B. POINT OF CONTACT (Name of person that VA can contact in order to get in touch with you)

8C. POINT OF CONTACT TELEPHONE NUMBER (Include Area Code)

9A. SERVICE (Check all that apply):
   - ARMY
   - NAVY
   - MARINE CORPS
   - AIR FORCE
   - COAST GUARD

9B. COMPONENT (Check all that apply):
   - ACTIVE
   - RESERVES
   - NATIONAL GUARD

10A. CURRENT MAILING ADDRESS (Number and street or rural route, P.O. Box, City, State, ZIP Code and Country)
   - No. & Street
   - Apt./Unit Number
   - City
   - State/Province
   - ZIP Code/Postal Code

10B. FORWARDING ADDRESS AND EFFECTIVE DATE (Provide the date you will be living at this address)
   - No. & Street
   - Apt./Unit Number
   - City
   - State/Province
   - ZIP Code/Postal Code
   - EFFECTIVE DATE:
     - Month
     - Day
     - Year

11. PREFERRED TELEPHONE NUMBER

12A. PREFERRED E-MAIL ADDRESS (If applicable)

12B. ALTERNATE E-MAIL ADDRESS (If applicable)
Games
Collaborative Worksheets
Worksheets

Create dynamic, interactive web pages. Publish online for personal or public use. Interlink to support collaborative work.

Public
Click here to access public worksheets.

Private
Click here to manage your own worksheets.

Cloud-based. No need to install hardware or software. Do It Yourself. No traditional programming required. Easy to manage. Dashboards and drop down lists.

Learn More