DEPARTMENT OF COMPUTER SCIENCE
MSCS Program Sheet (2010-11)

Name: Charles Parrell Naut  Advisor:  Proposed date for degree conferred: Date: 10/8/2010
Student ID #: Email: cparnell@stanford.edu

Primary Specialization

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 cs.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: cs.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)

TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT: 10

Note: This total may not exceed 10 units.

7 Requirements Left  Total Units: 10  Status: Draft
Heterogeneous Worksheets
Collaborative Heterogeneous Worksheets
Architectural Alternatives
Syntactic Worksheets

User gestures (e.g. clicking buttons) change widget state

Widget state (e.g. values of selectors) stored in lambda

Page state (e.g. colors of text) affects the display

Semantic Worksheets

User gestures translated to application operations

Application operations view and change application state

Application state (e.g. courses student has taken) stored

Page state defined as views of application state

Page state (e.g. colors of text) affects the display
Multiple Perspectives Example
Course Scheduling Perspectives

When

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
<th>Course 3</th>
<th>Course 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autumn</td>
<td>Autumn</td>
<td>Autumn</td>
<td>Autumn</td>
</tr>
<tr>
<td>Winter</td>
<td>Winter</td>
<td>Winter</td>
<td>Winter</td>
</tr>
<tr>
<td><strong>Spring</strong></td>
<td><strong>Spring</strong></td>
<td><strong>Spring</strong></td>
<td><strong>Spring</strong></td>
</tr>
<tr>
<td>Summer</td>
<td>Summer</td>
<td>Summer</td>
<td>Summer</td>
</tr>
</tbody>
</table>

What

<table>
<thead>
<tr>
<th>Autumn</th>
<th>Winter</th>
<th>Spring</th>
<th>Summer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course 1</td>
<td>Course 1</td>
<td>Course 1</td>
<td>Course 1</td>
</tr>
<tr>
<td>Course 2</td>
<td>Course 2</td>
<td>Course 2</td>
<td>Course 2</td>
</tr>
<tr>
<td>Course 3</td>
<td>Course 3</td>
<td>Course 3</td>
<td>Course 3</td>
</tr>
<tr>
<td>Course 4</td>
<td>Course 4</td>
<td>Course 4</td>
<td>Course 4</td>
</tr>
</tbody>
</table>

http://logicprogramming.stanford.edu/examples/courses/index.html
### Schedule 1

<table>
<thead>
<tr>
<th>Course 1</th>
<th>Course 2</th>
<th>Course 3</th>
<th>Course 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autumn</td>
<td>Autumn</td>
<td>Autumn</td>
<td>Autumn</td>
</tr>
<tr>
<td>Winter</td>
<td>Winter</td>
<td>Winter</td>
<td>Winter</td>
</tr>
<tr>
<td>Spring</td>
<td>Spring</td>
<td>Spring</td>
<td>Spring</td>
</tr>
<tr>
<td>Summer</td>
<td>Summer</td>
<td>Summer</td>
<td>Summer</td>
</tr>
</tbody>
</table>

click(when(C,Q)) :: style(when(C,Q), "background-color", white)  
===> ~style(when(C,Q), "background-color", white) &  
    style(when(C,Q), "background-color", grey)

click(when(C,Q)) :: style(when(C,Q), "background-color", grey)  
===> ~style(when(C,Q), "background-color", grey) &  
    style(when(C,Q), "background-color", white)
### Schedule 2

<table>
<thead>
<tr>
<th>Autumn</th>
<th>Winter</th>
<th>Spring</th>
<th>Summer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course 1</td>
<td>Course 1</td>
<td>Course 1</td>
<td>Course 1</td>
</tr>
<tr>
<td>Course 2</td>
<td>Course 2</td>
<td>Course 2</td>
<td>Course 2</td>
</tr>
<tr>
<td>Course 3</td>
<td>Course 3</td>
<td>Course 3</td>
<td>Course 3</td>
</tr>
<tr>
<td>Course 4</td>
<td>Course 4</td>
<td>Course 4</td>
<td>Course 4</td>
</tr>
</tbody>
</table>

**What**

\[
\text{click(what(Q,C)) :: style(what(Q,C),"background-color",grey)} \\
\quad \implies \neg \text{style(what(Q,C),"background-color",grey)} \land \text{style(what(Q,C),"background-color",white)}
\]

\[
\text{click(what(Q,C)) :: style(what(Q,C),"background-color",white)} \\
\quad \implies \neg \text{style(what(Q,C),"background-color",white)} \land \text{style(what(Q,C),"background-color",grey)}
\]
Syntactic Mapping Rules

\[
click(\text{when}(C,Q)) :: \text{style}(\text{when}(C,Q),"background-color",\text{white}) \\
\quad \Longrightarrow \sim\text{style}(\text{when}(C,Q),"background-color",\text{white}) \land \\
\quad \text{style}(\text{when}(C,Q),"background-color",\text{grey})
\]

\[
click(\text{when}(C,Q)) :: \text{style}(\text{when}(C,Q),"background-color",\text{grey}) \\
\quad \Longrightarrow \sim\text{style}(\text{when}(C,Q),"background-color",\text{grey}) \land \\
\quad \text{style}(\text{when}(C,Q),"background-color",\text{white})
\]

\[
click(\text{when}(C,Q)) :: \text{style}(\text{what}(Q,C),"background-color",\text{grey}) \\
\quad \Longrightarrow \sim\text{style}(\text{what}(Q,C),"background-color",\text{grey}) \land \\
\quad \text{style}(\text{what}(Q,C),"background-color",\text{white})
\]

\[
click(\text{when}(C,Q)) :: \text{style}(\text{what}(Q,C),"background-color",\text{white}) \\
\quad \Longrightarrow \sim\text{style}(\text{what}(Q,C),"background-color",\text{white}) \land \\
\quad \text{style}(\text{what}(Q,C),"background-color",\text{grey})
\]

+ 4 analogous rules for \text{what}(Q,C)
Operations (similar to previous operation definitions):
click(when(C,Q)) :: offered(C,Q) ==> ~offered(C,Q)
click(when(C,Q)) :: ~offered(C,Q) ==> offered(C,Q)

click(what(Q,C)) :: offered(C,Q) ==> ~offered(C,Q)
click(what(Q,C)) :: ~offered(C,Q) ==> offered(C,Q)

Lambda:
offered(course1,autumn)
offered(course2,autumn)

Views (in place of mapping rules):
style(when(C,Q),"background-color",grey) :- offered(C,Q)
style(when(C,Q),"background-color",white) :- ~offered(C,Q)

style(what(Q,C),"background-color",grey) :- offered(C,Q)
style(what(Q,C),"background-color",white) :- ~offered(C,Q)
Schedule

<table>
<thead>
<tr>
<th>Course</th>
<th>Room</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cs157</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cs161</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Schedule</th>
<th>g100</th>
<th>g200</th>
<th>g300</th>
</tr>
</thead>
<tbody>
<tr>
<td>morning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>afternoon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>evening</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Schedule Problem
## Schedule

### Course Table

<table>
<thead>
<tr>
<th>Course</th>
<th>Room</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cs157</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cs161</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Schedule Table

<table>
<thead>
<tr>
<th>Schedule</th>
<th>g100</th>
<th>g200</th>
<th>g300</th>
</tr>
</thead>
<tbody>
<tr>
<td>morning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>afternoon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>evening</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Schedule Problem
Collaborative Worksheets
Skirmish

http://worksheets.stanford.edu/examples/skirmish/peertopeer.html?room=skirmish
Collaborative Skirmish

http://worksheets.stanford.edu/examples/skirmish/peertopeer.html?room=skirmish
Tic Tac Toe

http://worksheets.stanford.edu/examples/tictactoe/peertopeer.html?room=cs151
Trifecta

Deck

1,2
1,3
2,1
2,2
2,3
3,3

White

1,1
3,1

Black

3,2

http://worksheets.stanford.edu/examples/trifecta/peertopeer.html?room=cs151
Tic Tac Toe - Trifecta

X O X

Deck
1,2
1,3
2,1
2,2
2,3
3,3

White
1,1
3,1

Black
3,2
Remote Collaboration

**Dataset Sharing**
- Easy to implement and debug
- May move lots of data
- Allows all users to see and modify all data

**Message Passing** (Communication Channels)
- Difficult to implement and debug
- Moves minimal data
- Privacy and security assured

**Backend Server** (MySQL, PHP, etc.)
- Moderate effort to implement and debug
- Development and maintenance of backend infrastructure
- Moves minimal data
- Privacy and security assured
Collaboration Code

<script src='http://epilog.stanford.edu/javascript/epilog.js'></script>

<script src='http://worksheets.stanford.edu/javascript/worksheets.js'></script>

<script src='http://worksheets.stanford.edu/javascript/warehouse.js'></script>
Collaboration Control

<location(cell(a,1),piece(white,rook,1))>
<location(cell(b,1),piece(white,knight,1))>
...  
<location(cell(g,8),piece(black,knight,2))>
<location(cell(h,8),piece(black,rook,2))>
white(50)
black(50)
control(white)
step(1)

</textarea>
Sierra is a browser-based interactive development environment (IDE) for Etplog. It allows users to view and edit datasets and rulesets. It provides a variety of tools for querying and modifying datasets and rulesets. As changes are made, it automatically updates visible datasets in spreadsheet-like fashion in accordance with the user’s rules. It also provides tools for analyzing datasets and rulesets, tools for tracing program execution, and tools for saving and loading files.

The command bar across the top provides access to menus concerning files, tools, and worksheets. Click here for an introduction to the system and its capabilities.

This version of Sierra utilizes different windows for different tools. It replaces a previous version in which all tools are accessed in the same window. Click here to switch to the old version.
Editing Code

<script src='http://epilog.stanford.edu/javascript/epilog.js'></script>

<script src='http://worksheets.stanford.edu/javascript/worksheets.js'></script>

<script src='http://worksheets.stanford.edu/javascript/debugger.js'></script>↑↑↑