### DEPARTMENT OF COMPUTER SCIENCE

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

<table>
<thead>
<tr>
<th>Name:</th>
<th>Charles Parrell Naught</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advisor:</td>
<td></td>
</tr>
<tr>
<td>Proposed date for degree conferred:</td>
<td></td>
</tr>
<tr>
<td>Date:</td>
<td>10/8/2010</td>
</tr>
<tr>
<td>□ HCP?</td>
<td>□ Coterm?</td>
</tr>
</tbody>
</table>

### 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 advisor 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 advisor 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: 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.

<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>4</td>
</tr>
<tr>
<td>Probability (CS 109, STAT 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>5</td>
<td></td>
</tr>
<tr>
<td>Computer Organization and Systems (CS 107)</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Principles of Computer Systems (CS 110)</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

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

Note: This total may not exceed 10 units.
Architectural Alternatives
Syntactic vs Semantic Worksheets

**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

<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>

<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>
<tr>
<td></td>
<td></td>
<td></td>
<td></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

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

click(what(Q,C)) :: style(what(Q,C),"background-color",white)
  ==> ~style(what(Q,C),"background-color",white) &
      style(what(Q,C),"background-color",grey)
Syntactic Mapping Rules

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)

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

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

+ 4 analogous rules for what(Q,C)
Data:

offered(course1, autumn)
offered(course2, autumn)

Operations (similar to previous operation definitions):

\[
\text{click(when}(C, Q)\text{)} :: \text{offered}(C, Q) \implies \neg \text{offered}(C, Q)
\]
\[
\text{click(when}(C, Q)\text{)} :: \neg \text{offered}(C, Q) \implies \text{offered}(C, Q)
\]
\[
\text{click(what}(Q, C)\text{)} :: \text{offered}(C, Q) \implies \neg \text{offered}(C, Q)
\]
\[
\text{style(when}(Q, C)\text{)} :: \neg \text{offered}(C, Q) \implies \text{offered}(C, Q)
\]

Views (in place of mapping rules):

\[
\text{style(when}(C, Q), \text{"background-color"}, \text{grey}) :: \text{offered}(C, Q)
\]
\[
\text{style(when}(C, Q), \text{"background-color"}, \text{white}) :: \neg \text{offered}(C, Q)
\]
\[
\text{style(what}(Q, C), \text{"background-color"}, \text{grey}) :: \text{offered}(C, Q)
\]
\[
\text{style(what}(Q, C), \text{"background-color"}, \text{white}) :: \neg \text{offered}(C, Q)
\]
### 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

<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
Collaborative Worksheets
http://worksheets.stanford.edu/examples/skirmish/peertopeer.html?room=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
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
<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>
<textarea id='lambda' type='text/hrf'
broadcast='true' reception='true'
style='display:none'>
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>
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>

↑↑↑