Principles of Software Engineering: System Deployment

Ethan Jackson And Wolfram Schulte,
Research in Software Engineering (RiSE)
Microsoft Research
Grading

- Project 1 graded, working on Project 2.

<table>
<thead>
<tr>
<th>Correctness of Solution (C)</th>
<th>Application of Formalism (F)</th>
<th>To Boldly Go (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2  Perfect solution</td>
<td>2    Deep and appropriate</td>
<td>2    Major extensions</td>
</tr>
<tr>
<td>1  Minor glitches</td>
<td>1    Solid application</td>
<td>1    Special insights / features</td>
</tr>
<tr>
<td>0  One significant problem</td>
<td>0    Basic usage</td>
<td>0    Solid approach</td>
</tr>
<tr>
<td>-1 Several significant problems</td>
<td>-1   Insufficient application</td>
<td></td>
</tr>
<tr>
<td>-2 Doesn’t work</td>
<td>-2   Little or no use</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{grade} = 3.4 + 0.1 \times (C + F + B).
\]
Goals

- Based on the FORMULA cloud deployment example (already on website) develop your own model of software components and computing nodes.

- Software components require memory, CPU time, etc…

- Computing nodes provide memory, CPUs, other resources. There may be heterogeneous kinds of CPUs.
Goals

► Build a “Software Component” domain in FORMULA where you can describe systems of software components.

► Build a “Computing” domain in FORMULA where you can describe available computing resources.

► Build a “Mapping” domain which explains how software can be mapped to hardware. Should include constraints, e.g. code must fit into memory.

► Synthesize a valid architecture by constructing a partial model and using the FORMULA model finder.
Thanks And Questions!

http://www.cs.washington.edu/csep503