Hard versus Soft Systems Methodology

David Cairns
CSC9T4

Systems approaches

• So we know what the systems approach is now.

• But, there’s different ways of viewing a system:
  – Soft systems approach
  – Hard systems approach

• Two ends of a scale, a bit grey in the middle
Soft Systems Approach

• Engineering approach can be inappropriate for ‘soft problems’ (with fuzzy requirements).

• Soft systems approaches (Soft Systems Methodology, Soft OR) assume:
  - organisational problems are ‘messy’ or poorly defined
  - stakeholders interpret problems differently (no objective reality)
  - human factors important
  - creative, intuitive approach to problem-solving
  - outcomes are learning, better understanding, rather than a ‘solution’

Hard Systems Approach

• Hard systems approaches (systems analysis (structured methods), systems engineering, operations research) assume:
  - Objective reality of systems in the world
  - Well-defined problem to be solved
  - Technical factors foremost
  - Scientific approach to problem-solving
  - An ideal solution

• More traditional way of viewing systems in Computing Science
What's the difference in hard and soft systems methodologies?

- HARD systems - rigid techniques and procedures to provide unambiguous solutions to well-defined data and processing problems, focused on computer implementations

- SOFT systems - a loose framework of tools to be used at the discretion of the analyst, focused on improvements to organisational problems

Hard Systems Approach - Example

• “Design a virtual resource management system where resources are held in staff offices but are made available via a web based interface. Resources will be booked out via the on-line management system which will also send reminders when items are due for return.”

• Identify any obvious requirements via nouns & verbs
  - Web Based
  - Booking System
  - Users
  - Resources
Example

Expand Requirements
Web Based
- Web Server - Platform?
- Web Client - Browser / Applet
Booking System
- Flow control
- Event manager
- Data Base
Users
- Staff
- Students
- Administrators
Resources
- Books
- Journals
- Rooms?
- CDs?

Analysis

- Once we have a proper set of requirements, the next step is the analysis of the problem.
  - This involves understanding the scope of the problem
  - Identifying the parts in the requirements that will be part of the eventual solution
    - For example, is the web server component part of our solution?
  - Connect them together in an appropriate manner
Design

- Analysis of the problem indicates what the major components in the system are, it will not tell us how these components work.

- Design involves
  - Identification of major component boundaries
  - Decomposition of the major components into smaller semi-independent sub-systems
  - Design of the interfaces between these major components & sub-systems

Design

- Design involves (continued)
  - Identification of new components necessary to bridge the gap between objects in the problem domain and the solution domain.
  - Flow of control within the system
  - Flow of data within the system
Implementation & Testing

- **Implementation** - The translation of the Design into Source Code
  - For each identified component and interface in the design phase, create the source code that will implement it
  - Integration of code components such that they perform as one system

- **Testing**
  - Check that each element / sub-system / component does what it is required to do by the design
  - Check system meets the requirements specification
  - Check system meets the clients expectations
  - Check system meets the users expectations

Design - Example

```
User
  Staff
  Students
  Administrator

Resource
  Book
  Journal
  CD

Booking System
  Controller
  Event Manager
  Database
```
### Hard vs. Soft Systems: Review

<table>
<thead>
<tr>
<th>Hard Systems</th>
<th>Soft Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem has a definite solution</td>
<td>There are many ‘problems’ to be solved</td>
</tr>
<tr>
<td>Problem has a number of achievable goals</td>
<td>Goals cannot be measured</td>
</tr>
<tr>
<td>They answer the ‘how’ questions</td>
<td>Emphasis is placed on ‘what’ as well as ‘how’</td>
</tr>
<tr>
<td>Has a deterministic complexity</td>
<td>Has a unpredictable, non-deterministic, non-definable complexity</td>
</tr>
<tr>
<td>Likely to have defined parameters for failure</td>
<td>Less easily dealt with</td>
</tr>
</tbody>
</table>

- Hard Systems vs. Soft Systems
  - **Hard Systems** focus on **what** and **how** questions, which are typically associated with achievable goals and solutions that can be measured. They are characterized by deterministic complexity and likely have defined parameters for failure, making them easier to address.
  - **Soft Systems**, on the other hand, deal with less easily defined problems that are likely to have defined parameters for failure. They are characterized by unpredictable, non-deterministic, and non-definable complexity, which makes them less easily dealt with. Goals are often less measurable, and solutions are not as definite.