TORTOISE
Task Orientation and Tailoring Of Interactive Software Explanations

Ali Erdem
Information Sciences Institute/CS Dept
University of Southern California

Outline

• Introduction and Background
• Question Study
• Task Study
• Implementation
• Status and Future Work
Introduction and background

• Motivation
  – Software Maintenance is costly
  – A large portion of maintenance activity is spent on Software Understanding

• Software Understanding is the reconstruction of logic, structure and goals in the program

• It consists of inquiry episodes and described as a discovery process

Inquiry Episodes

• Studies of verbal comprehension protocols
  – Read some code
  – Ask a question about the code
  – Conjecture an answer
  – Search the code and the documentation to confirm

• Sample Protocol (from Letovsky)
  – READ: DBASE is an array: 200,7
  – ASK: Why seven?
  – CONJECTURE: Seven fields, I’l bet.
  – SEARCH: I looked in the documentation and there are 7 fields.
Attempted Solutions

- Improve Search
  - SODOS: Linked software life cycle documents
  - Rajlich: Layered documentation
  - LaSSIE: Query language
  - Soloway: Delocalized plans

- Improve Recognition
  - Programmer’s Apprentice: Cliches
  - LaSSIE: Domain Model

Factors that affect Software Understanding

- Complexity of the problem solved
- User’s ability and experience
- The task being performed
- The structure of the program
- Inherent properties of software
  - Complexity, Conformity, Changeability, Invisibility
- Documentation structure
Our Approach

- Motivation:
  - User and task are important factors in understanding
  - Previous studies focused on the program and ignored the user and his task

- Goal
  - Tailor the explanation to the user's expertise
  - Help user in performing the task
  - Support inquiry episodes

Question Study

- Motivation
  - What do programmers ask?
  - What kind of answers do the experts provide?

- Study
  - 1250 messages posted to USENET comp.lang.tcl newsgroup between 2/17/95-4/22/95
  - studied questions (type, subject) and answers
Types of user questions on USENET

- **Goal oriented**
  - Plan request (45%) \( \text{How do I do x?} \)
  - Goal satisfiability (22%) \( \text{Can I do x?} \)
- **Symptom oriented** (24%)
  - \( \text{What is wrong?} \)
- **System oriented**
  - Motivational (2%) \( \text{Why is x necessary?} \)
  - Conceptual (2%) \( \text{What is x?} \)
  - Explanatory (5%) \( \text{How does x work?} \)

**Question Model**

- **Simple Questions:** What are the inputs to X?
  - Question topic: X
  - Question type: What
  - Relation type: input
- **Complex Questions:** How many inputs are there to X?
  - Attribute: What is the type of variable X?
  - Count: How many inputs are there to X?
  - Significance: What is the slowest function in project X?
  - Comparison: What is the difference between X and Y?
### Simple questions

<table>
<thead>
<tr>
<th>Model</th>
<th>What If?</th>
<th>What’s Wrong?</th>
<th>Have I Found the Problem?</th>
<th>What Do I Do Now?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
</tr>
<tr>
<td>Verification</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
</tr>
<tr>
<td>Proactive</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
</tr>
<tr>
<td>Reactive</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
</tr>
<tr>
<td>Temporal</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
</tr>
<tr>
<td>Location</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
<td>suppose?</td>
</tr>
</tbody>
</table>

### Problems of the question model

- Unsupported question types
  - Hypothetical questions (what-if)
  - Symptom oriented questions (what’s wrong?)
  - Decision questions (have I found the problem?)
  - Task oriented questions (what do I do now?)
- Question Topic
  - Undefined topics are not supported
  - Topic structure can become quite complex
Task Study

- Motivation
  - Investigate the dynamics of inquiry episodes
  - Investigate the task structure
  - Investigate user goals during task performance
  - Test sufficiency of the question model

- Study
  - Impact Analysis task on ModSAF
  - Northrop-Grumman integration bug fix task study

Task study: observations

- Existence of similar subgoals among the tasks
  - gather, comprehend, test, find-method
- Different types of information needed in task steps
- Layers of abstractions exist
- Relations between user goals and questions
Implementation

• Tailoring by planning
  – User tailoring
    • Don’t tell the user what he already knows
    • Tailor to user’s level of expertise
    • Tailor to the user’s role
  – Task tailoring
    • Tailor to user goal

• Task support
  • Explicit task support
  • Record topics of interest and questions for recall
Status and future work

- Status
  - User tailoring achieved by planning
  - Explicit task support
  - Project info at http://www.isi.edu/isd/media-doc/media-doc-body.html

- Future work
  - Structuring of question topics
  - Building a taxonomy of user goals and tasks
  - Improved plan operators for tailoring to task
  - Evaluation