MediaDoc

W. Lewis Johnson

USC Information Sciences Institute
Marina del Rey, CA
http://www.isi.edu/isd/I-DOC/media-doc.html
Introduction

Software understanding involves inquiry episodes:
- read code, other artifacts
- formulate hypotheses
- search to test hypothesis

Understanding can be facilitated by:
- responding directly to inquiries
- making search more efficient by:
  - suppressing irrelevant detail
  - presenting in a form that facilitates search
Objectives of Effort

- Generate user-tailored software explanations
  > focus on generation from rich rationale representations
- Combine with graphical presentations
- Support knowledge acquisition
- Development objectives:
  > Create a usable integrated presentation system
  > Create usable system components
  > Interface to non-EDCS and other components
User Tailoring Factors

- Knowledge/Expertise (vocabulary, abstractions)
- Task
- Amount of detail
- Exposure to previous presentations
- Presentation preferences
Example
Another Example

Codes dependent on Tscript Protocol
Approach

- Identify user questions
- Develop a user model
- Identify tailoring strategies
- Capture and represent necessary knowledge
- Generate user tailored explanations
- Evaluate
Types of user questions

- Goal oriented
  - Plan request
  - Goal satisfiability

- Symptom oriented

- System oriented
  - Motivational
  - Conceptual
  - Explanatory

- How do I do x?
- Can I do x?
- What is wrong?
- Why is x necessary?
- What is x?
- How does x work?
Structured Classification of Questions

**Question Type**
- Conceptual (What/where/when)
- Motivational (Why)
- Explanatory (How)
- Problem (What is wrong?)
- Goal satisfiability (Can I?)
- Plan request (How can I?)

**Further processing of answers**
- Count (How many?)
- Significance (What is the most ...?)
- Comparison (What is the difference?)

**Question topic**
- **Topic dimensions**
  - structure vs. process
  - individual vs. aggregate
  - qualitative vs. quantitative
  - external vs. internal
  - physical vs. logical

- **Topic complexity/types**
  - attribute of object
  - relations between objects
  - preconditions/postconditions
  - states
  - events
  - plans/methods
  - goals
  - constraints

**USC INFORMATION SCIENCES INSTITUTE**
Current I-Doc System

Advanced Multi-Purpose Support Environment

Application Programmer Interface

An application programmer may invoke the following functions:

- REO Model Control Functions

Programmers may also interface to Advanced Multi-Purpose Support Environment by accessing designated global data elements:

- REO Model data

Further Information

- What does Advanced Multi-Purpose Support Environment do?
- How is Advanced Multi-Purpose Support Environment structured?
- What are the platform dependencies?
- Full description of system
Architecture

Documentation Repository

Methods | Annotations

SW Repository Interface

CGI Scripts

Web Server

Web Client

Software Repository

Code | Design Info

User DB

Web Client

USC INFORMATION SCIENCES INSTITUTE
Presentation

```perl
if ($task eq "Port to new platform") {
  print "<H3> Platform Information </H3>",
  print $CONF_SCRIPTS("MPSE,-!USER!PLATFORM");
  print "<H3> Further Information </H3>",
  print <$\nEnd$;
  print '<li><a href="$CONF_SCRIPTS(task-oriented/username-$username?t
  print What does Advanced Multi-Purpose Support Environment do?\n  print '<li><a href="$CONF_SCRIPTS(task-oriented/username-$username?t
</ul>
```

Others

```
WHAT

The AMPSE system is a support environment satisfying the Mission or
(MOCR) requirements for testing embedded software as specified in the
Command (AFCL) Long-Range Plan for Embedded Computer System (ECS) &
The AMPSE provides an alternative to configuration, weapon system
environments. The objective of the AMPSE design is to improve logistics
system software by reducing cost and increasing both capability and
environment. The REO model is one of several software models that pre
environment for the ECS software under test. The purpose of the REO
```

W. Lewis Johnson
Mar. 10, 1997
Knowledge representation

- Finer grain representation is necessary
  - The annotation content is not understood by the tool
  - Information overlap between annotations
  - Annotation content was not used in the search
  - Large number of annotations required for tailoring
- Analysis of USENET messages for shared knowledge
- Survey of literature
  - Syntactic, semantic, schematic, strategic knowledge
User Types

- Analyst: interested in domain actions, requirements
- Designer: interested in realization of requirements
- Installer: interested in system components, procedures
- Programmer: focus on structure, function, behavior
- Manager: interested in resource usage, risks
- User: interested in behavior, usage
Node and relation types

- Instance: Instances are instantiations of concepts
- Concept: Objects are words
- Object: Objects have attribute-value pairs in state
- State: Processes capture input-output relations. Manner is necessary to qualify the act.
- Process: Event puts process in context and binds an actor. Actor, process can be multi-valued.
User saves his understanding of a concept as text in order not to forget it.
R1: User should be able to edit the text

R2: The text should have flexible format

R3: The text should be persistent

R4: User should be able to link the explanations
D3: A text database will realize Req-3
D1: Annotation edit function will realize Req-2
D2: Text database will be in SGML format
D4: Annotations are in HTML format
Proposed Architecture

Documentation
Parser
Assertions

Loom classifier

Loom Knowledge Base

Dynamic User model

Web Client
Queries

Web Server

CGL Scripts

Text generator

Software Refinery

Source code
Q/A Architecture/flow chart

Question

User & task model

Parse & Disambiguate

Search answer(s)

Select content (best answer)

Abstraction & Detail selection

Language generation

Presentation

Domain knowledge

Software Repository

FUF & SURGE

USC INFORMATION SCIENCES INSTITUTE

W. Lewis Johnson
Mar. 10, 1997
Graphical Presentation

Principles

- Organize around dimensions present in data
- Retain visual organization as data changes
- Avoid juxtaposing high-saturation colors
- Present according to how data will be used

Sources:
  - Tufte, Ellis, Hovy and Arens, etc.
Graphical Presentation
Approach

- Create presentation plan for background data
- Update with plan for foreground annotations
Markup for Authoring Presentations (MAP)

- SGML-based
- Will define:
  - Layout regions
  - Organization for region
  - Data elements
  - Sentence plans for textual annotations
  - Animation procedures
Presentation Process

- Map content model on existing databases
- Select relevant content
- Generate presentation plan
- Enact plan
- Map edits onto presentation plan, content model
MAP Specification

```
<node name=n1 label="Enabled" url=null>
<node name=n2 label="Travel/in/Formation" url=null>
<node name=n3 label="Action/on/Contact" url=null>
<node name=n4 label="Goal/Location" url=null>
<node name=n5 label="Disabled" url=null>
<edge name=e1 from=n1 to=n2>
<edge name=e2 from=n2 to=n4>
<edge name=e3 from=n2 to=n3>
<edge name=e4 from=n3 to=n5>
```
<loop name="loop1">
  <step name="step1">
    <actor layer="bg">
      <method type="color" rgb="00FF00" object="node name=nl1">
      </method>
    </actor>
    <actor layer="fg" type="string" name="Initiating" rgb="05A705" fontstyle="BOLD" fontsize="16">
      <method type="move" object="node name=n1" object2="node name2=n2" speed="10">
      </method>
    </actor>
  </step>
  <step name="step2">
    <actor layer="bg">
      <method type="blink" object="edge name=el">
      </method>
    </actor>
    <actor layer="fg" type="string" name="Contact sighted" rgb="fff900" fontstyle="BOLD" fontsize="16">
      <method type="move" object="node name=n2" object2="node name2=n3" speed="50">
      </method>
    </actor>
  </step>
</loop>