Using Dynamic Analysis for Generating End User Documentation for Web 2.0 Applications

Domenico Amalfitano
Anna Rita Fasolino
Porfirio Tramontana

Dipartimento di Informatica e Sistemistica
University of Naples Federico II, Italy
Motivation

- Web 2.0 Applications have emerged as a new generation of Web applications offering greater usability and interactivity than traditional ones.
  - Web pages have been transformed in complex GUIs with synchronous and asynchronous interactions with the user and with the resources

- End User Documentation needs
  - The complexity and the variety of interactivity features provided by a modern Web 2.0 Applications may often confuse the user, so end user documentation is needed to improve the usability of the applications
User documentation definitions

According to the IEEE 1063 standard:

- Instructional mode
  - “Usage mode that is intended to teach the use of software in performing tasks”
- Reference mode
  - “Usage mode that is intended to provide quick access to specific information for software users who are generally familiar with the software functions”
- Procedure:
  - “Ordered series of steps that a user follows to do one or more tasks”
- Step:
  - “One element of a procedure. A step contains one or more actions”
- Action:
  - Element of a step that a user performs to complete a procedure.
- Tutorial:
  - “Instructional procedure in which the user exercises software functions using sample data supplied with the software or documentation”
Some documentation examples

- **Tudu list user documentation** (our case study)
- Other examples
  - ASSP Web Application
  - TravWeb
Web 2.0 applications

- Web 2.0 applications have complex User Interfaces, with both client side and server side interactions
  - Business criticality is related to their learnability, operability and usability
- The client side code is usually dynamically generated
  - Static analysis techniques can extract only a limited set of information
Benefits and effort of a good documentation

- Learnability
- Operability
- Usability

- Effort needed to write documentation
- Effort needed to maintain the documentation up-to-dated
User documentation generation

- Many tools available for the generation of internal documentation, usually based on static analysis of the source code:
  - Javadoc
  - Doxygen
- Some tools can generate tutorials from dynamic analysis:
  - Smart Tutor
  - DocWizards
  - JTutor
  - EpiDocX
- Is it possible to support and partially automate the process of generation of end user documentation for a Web 2.0 application?
Our documentation generation approach

- Supports the automatic generation of some end user documentation items
  - Overall navigation Graph
  - Scenario execution tutorials
  - End user reference guide

- Three steps:
  - Web Application Dynamic Analysis
  - Generation of the Navigation Graph
  - End User Documentation Generation
Web Application Dynamic Analysis

- Use case scenarios of the Web application are navigated and described by the documenter
  - With the support of the CReRIA tool
- He has to give a name to any Interface Instance and to any Transition Instance
  - The tool suggests the reuse of identical names when it judges that the interface/transitions are “similar” to previously encountered and described ones
    - Different clustering heuristics are available
Web Application Dynamic Analysis

Add a new list

Description
Allow RSS publication?
[Submit] [Cancel]
Navigation Graph Generation

- Overall and scenario specific navigation graphs can be obtained
  - Automatically with the support of the CRerIA tool for generation and of the GVEdit tool for visualization
- Graphs and documentation information can be furtherly edited
  - With the support of CRerIA
A case study

- Tudu Lists
  - 10 KLOC in Java/JSP
  - Rich Client Interfaces
- Existing user documentation ([http://www.julien-dubois.com/tudu-lists/user-documentation](http://www.julien-dubois.com/tudu-lists/user-documentation)) is limited to a list of the 8 main use cases, with 4 screen shots
- Often used as a case study for RIAs reverse engineering and testing
Process execution metrics

- 3 actors
  - Generic User, Logged User, Administrator
- 23 use cases
- 120 scenarios
  - 120 collected execution traces
- 42 Interfaces
  - By clustering together 425 Interface Instances
  - Clustering suggestions accepted 420 out of 425 times
- 138 Transitions
  - By clustering together 306 Transition Instances
  - Transition clustering accepted 278 out of 306 times
- 120 generated graphs
- 120 generated tutorials
End User Documentation Generation

- On the basis of the information extracted and abstracted in the previous phases, have been automatically generated:
  - Overall navigation Graph
  - Scenario execution tutorials and graphs
  - End user reference guide
Overall Navigation Graph
An example of scenario navigation graph

Classic Graph View

Graph view with Screenshots

Clickable nodes
### An example of scenario description

**Quick Add**

In this scenario a logged user can insert a todo in a selected list of todos.

**Event Sequences**

**Related Regular Scenarios**
- Advanced Add

**Related Exception Scenarios**
- Insertion of Wrong Data
Scenario Tutorials

- Automatically generated as HTML hypertexts
  - They can be included in the Web application itself

- The tutorial template includes:
  - Scenario name and description
  - Involved Actors
  - Use case name
  - For each interface/transition composing the scenario
    - Interface name and description
    - Transition name and description
      - Input names and values
      - Triggering Event
Scenario Tutorial example
Scenario Tutorial example
Scenario Tutorial example

Advanced Add

- Description
  - new task
- Priority
  - 1
- Assigned to
  - example of advanced add
- Notes

Transition: add task
Input:
- insert 'new task' in description Text field
- insert '1' in priority Text field
- insert '09/26/2011' in dueDate Text field
- select 'mmmm' in assignedUser Select field
- insert 'example of advanced add in notes' in notes Text area
- Event: click on link 'Submit'
Scenario Tutorial example
Possible extensions

- Customization of presentation styles
  - Generation in textual formats
  - Choose between different styles

- More detailed documentation

- Use of collected traces in testing and reverse engineering processes

- Use of collected traces for on-line replaying of tutorials

- Assessment of documentation regression
Conclusions and Future Works

- This paper introduced a process and a tool for the semi-automatic generation of end user documentation for Web 2.0 applications via dynamic analysis.

- In future works we plan:
  - To increase the features of the tool.
  - To carry out experiments aiming at the comparison between documentation manually generated and semi-automatically generated with the support of the tool.
    - Quality of the documentation
    - Needed effort
Time is over ... Is there any (clarification) question?