Specifying user interfaces with extended use cases

Demosthenes Akoumianakis PhD

Department of Applied Information Technology & Multimedia
Technological Education Institution of Crete
and
Center of Technological Research of Crete (CTR-C)
da@epp.teiher.gr

Plan

- Rationale and context of the work
- Method base line
  - Objective
  - Instruments used
  - Process for using the instruments
  - Outcomes
- Tool description using a case study
  - Scalable ftp application

Objective & research question

- Scenario-based representation for expressing / specifying adaptable & adaptive interactive behaviors
- Rationale
  - Problems with model-based tools; they cannot model underlying goal requiring adaptable and adaptive behavior
  - Problems with visual languages (such as UML)
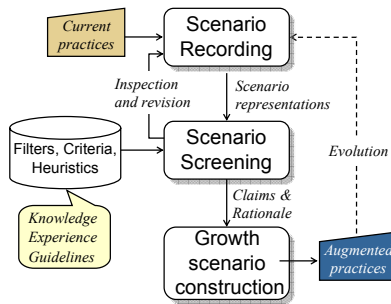   - UML*i*
- Relevant methods
  - Scenario networks

Proposed approach

- Evolutionary scenario management
  - Reference scenarios as explicit (accountable) resources
    - Computer-mediated representation
  - Scenario critiquing to identify design breakdowns
    - Using heuristics
    - Using non-functional requirements
    - Using (user-, task-, context-) specific filters
  - Scenario augmentation
    - Scenario relationships
    - Growth scenarios
  - Global execution context graph
    - Integration of reference and growth scenarios into a single representation

Process for using the instruments
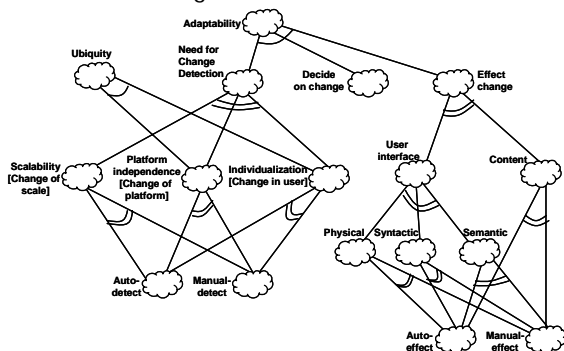
- Three stages for scenario management

Current practices → Scenario Recording

Inspection and revision | Scenario representations

Filters, Criteria, Heuristics

Knowledge Experience Guidelines

Scenario Screening

Evolution

Claims & Rationale

Growth scenario construction → Augmented practices

Scenario recording

- Techniques for representing scenarios
  - Free narratives
  - Partitioned narratives
  - Task hierarchies
  - Use case diagrams
  - Pseudocode
  - Video
- Why so many alternative forms?
  - Scenarios may change form
    - ... as our understanding is improved
    - ... depending on the use it is put
    - ... as a result of arbitrary factors (designer's choice)
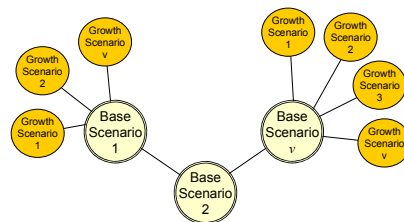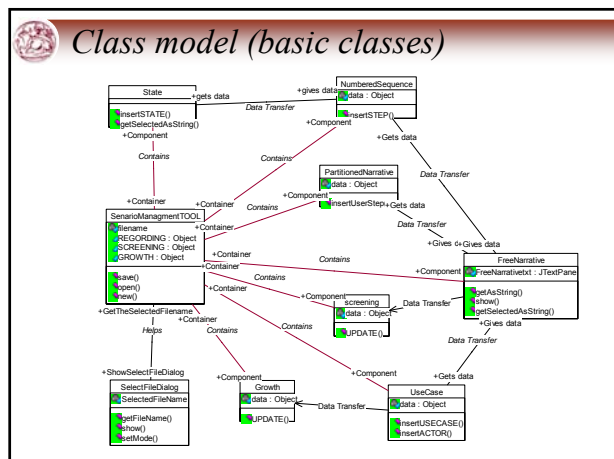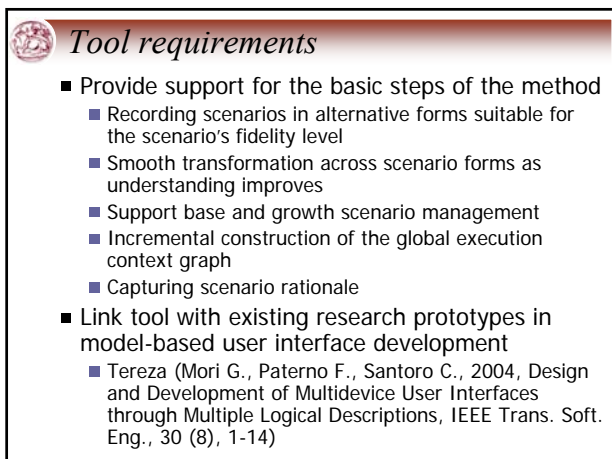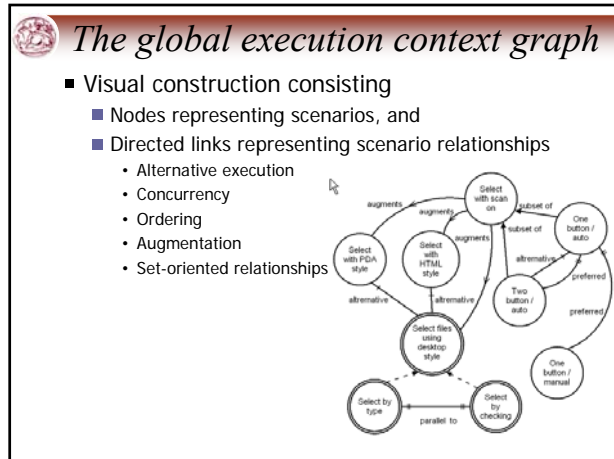 - Each form offers alternative insight

Scenario filtering

- Goal modeling

Adaptability
Ubiquity
Need for Change Detection
Decide on change
Effect change
Scalability [Change of scale]
Platform independence [Change of platform]
Individualization [Change in user]
User interface
Content
Auto-detect
Manual-detect
Physical
Syntactic
Semantic
Auto-effect
Manual-effect
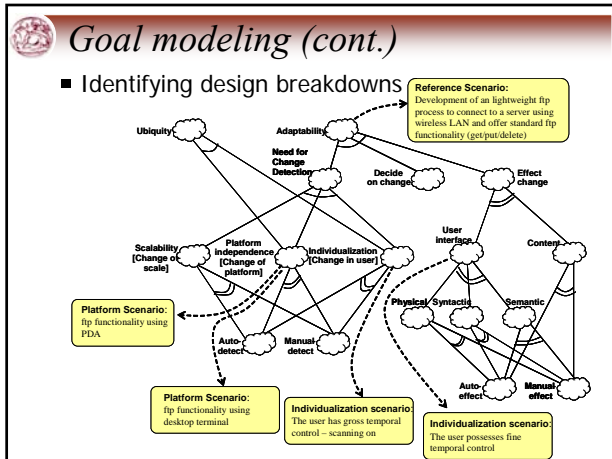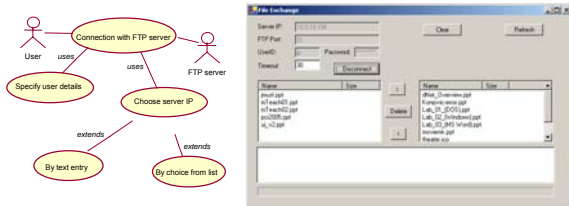
Growth scenarios

- Scenarios which 'remove' design breakdowns
- Expressed in the same form as base scenarios
- Growth scenario intertwining

Growth Scenario v
Growth Scenario 2
Growth Scenario 1
Base Scenario 1
Base Scenario 2
Growth Scenario 1
Growth Scenario 2
Growth Scenario 3
Base Scenario v
Growth Scenario v

Goal modeling (cont.)

- Identifying design breakdowns

Reference Scenario:
Development of an lightweight ftp process to connect to a server using wireless LAN and offer standard ftp functionality (get/put/delete)

Platform Scenario:
ftp functionality using PDA

Platform Scenario:
ftp functionality using desktop terminal

Individualization scenario:
The user has gross temporal control – scanning on

Individualization scenario:
The user possesses fine temporal control

The global execution context graph

- Visual construction consisting
  - Nodes representing scenarios, and
  - Directed links representing scenario relationships
    - Alternative execution
    - Concurrency
    - Ordering
    - Augmentation
    - Set-oriented relationships

Tool requirements

- Provide support for the basic steps of the method
  - Recording scenarios in alternative forms suitable for the scenario's fidelity level
  - Smooth transformation across scenario forms as understanding improves
  - Support base and growth scenario management
  - Incremental construction of the global execution context graph
  - Capturing scenario rationale
- Link tool with existing research prototypes in model-based user interface development
  - Tereza (Mori G., Paterno F., Santoro C., 2004, Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions, IEEE Trans. Soft. Eng., 30 (8), 1-14)
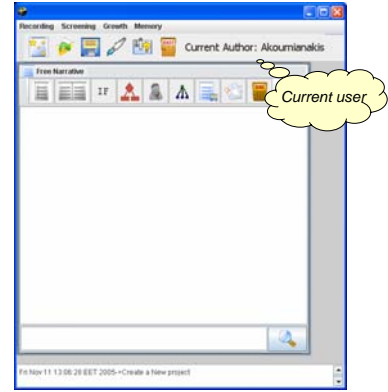
Class model (basic classes)
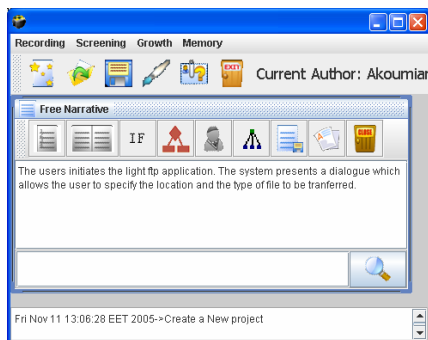
Design case

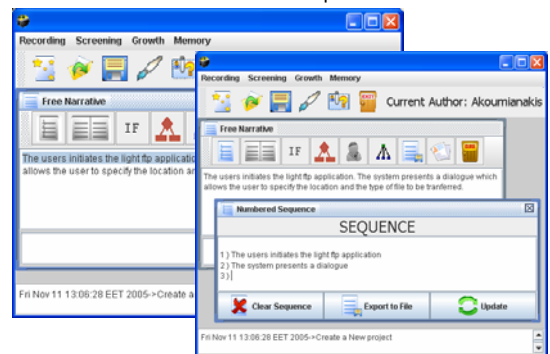- Initial use case model

A typical session

Scenario narratives
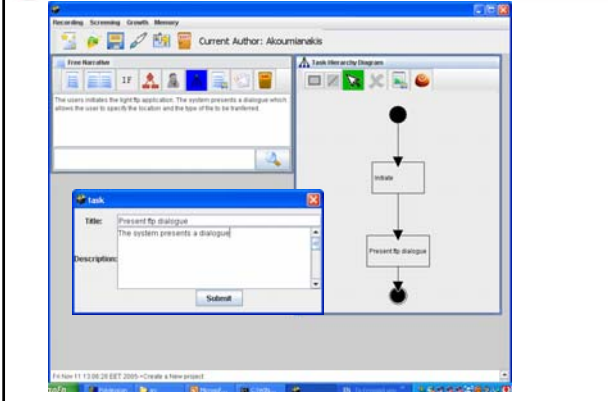
- Narrative description of a scenario

Scenario numbered sequence

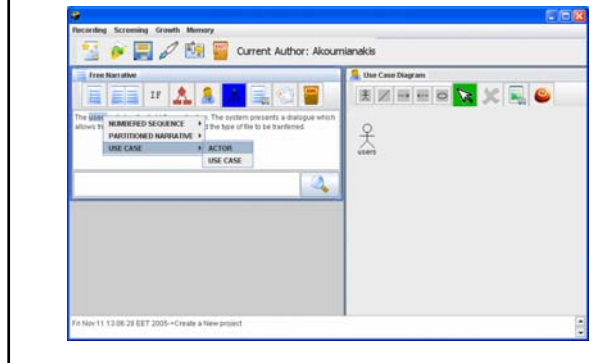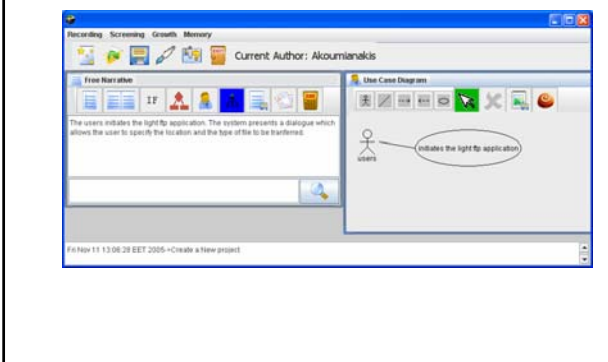- From narratives to numbered sequences

State diagram

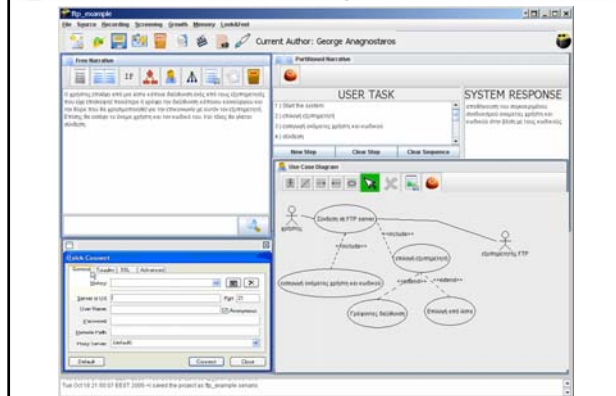## Use case representation

- Defining an actor
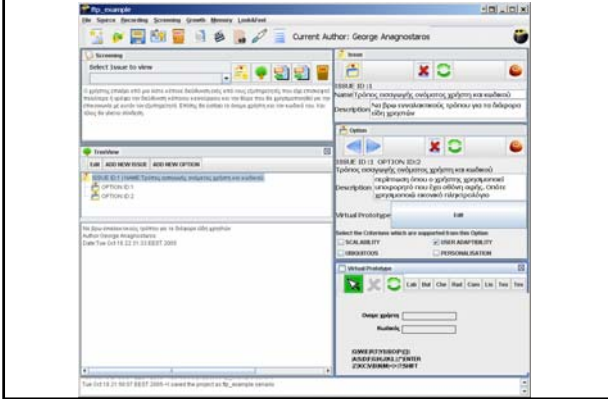

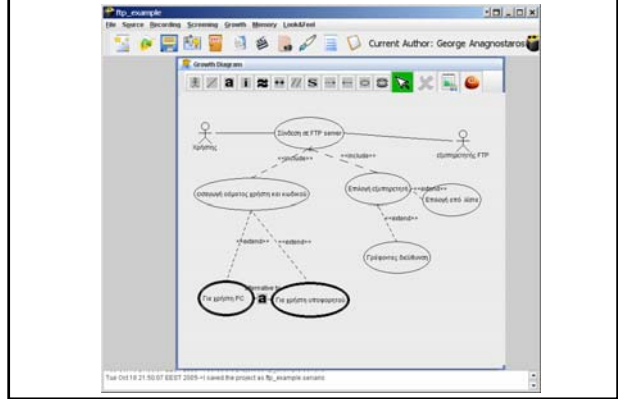## Use case representation (συν.)

- Defining a use case

Scenario recording
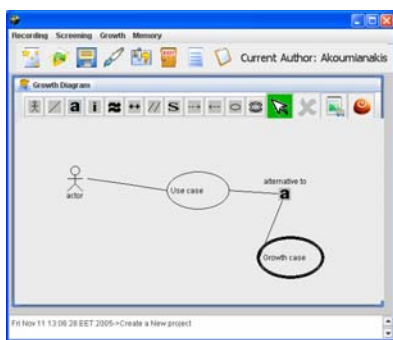
5

Scenario filtering
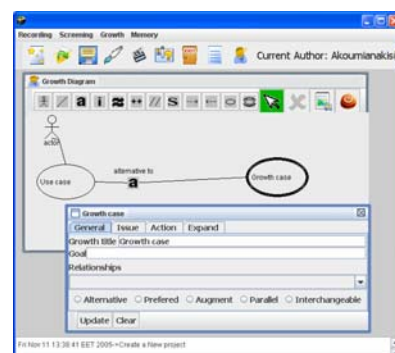

Growth scenarios

Growth scenarios

- Adding a growth scenario

Growth scenario

- Reviewing a growth scenario

Memory component

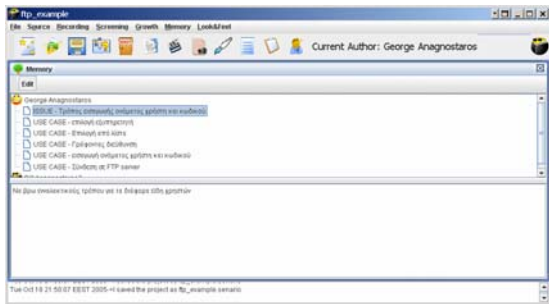

Output – XML issue database

```
<?xml version="1.0" encoding="UTF 8" ?>
  <java version="1.5.0_05" class="java.beans.XMLDecoder">
  <object class="screening.Issue">
  <string>Demos Akoumianakis</string>
  <object class="java.util.Date">
  <long>1129540312279</long>
  </object>
  <int>1</int>
  <string>How to log in</string>
  <string>ways that user will use to log in the server</string>
  <int>2</int>
  <int>1</int>
  <object class="java.util.Vector">
  <void method="add">
  <object class="screening.Option">
  <array class="boolean" length="4">
  <void index="1">
  <boolean>true</boolean>
  </void>
  </array>
  <string>By typing a user name and a password using a
keyboard</string>
  <int>1</int>
  <int>1</int>
```
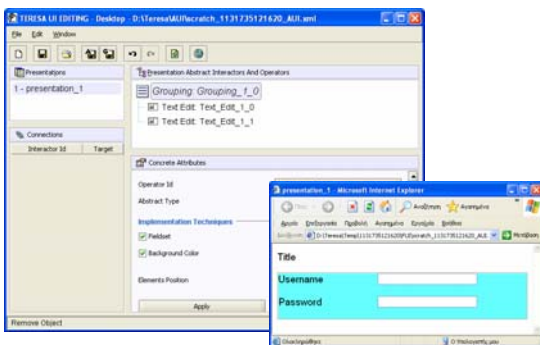
Interoperation with Tereza (on-going)

- Example of UI generation

Summary & contributions

- Synergy between
  - Scenario-based requirements engineering
  - Goal modeling in requirements engineering
  - Non-functional requirements
- Global execution context graph
  - Unified representation of existing & foreseen practices
  - Scenario relationships
- Growth scenarios explicitly linked to base scenarios
  - Managing evolutionary requirements
- Representation of adaptable & adaptive behavior
  - Tracing what is to change, why and how