Monitoring and Control in Scenario-Based Requirements Analysis

Emmanuel Letier, Jeff Kramer, Jeff Magee and Sebastian Uchitel

*International Conference on Software Engineering (ICSE ‘2005)*

2006. 2. 8

Han, Ah-Rim
Contents

- Introduction
- Overall approach
- Background
  - Implied Scenarios
- Input–Output(IO) Implied Scenarios
- Model Elaboration Using IO–Implied Scenarios
- Conclusion and Contribution
- Discussion
- Ongoing Work
Introduction (1/3)

- **Scenarios**
  - Sequences of interactions between agents of a system
    - Each interaction corresponds occurrences of events
      - Controlled events – performed or initiated by agents
      - Monitored events – observed by agents
  - Effective means for eliciting, validating, and documenting requirements
Scenario analysis

- Provide early feedback about unforeseen and potentially harmful consequences
  - Highly cost-effective

Techniques

- Implied scenarios
- Model checking
- Race condition
- Non-local choices
- Timing conflicts
Introduction (3/3)

- Motivation of this paper
  - **Unsuitable** for analyzing requirement-level scenarios using standard implied scenarios
    - Based on concurrent state machines that do not distinguish between the monitored and controlled events of agents

- Newly provided idea of this paper
  - Define **input–output implied scenarios**
    - Taking into consideration of distinction between monitored and controlled events
    - Ensuring an agent cannot prevent the occurrence of events it monitors
Overall Approach

Add/Modify Scenario
- Avoid unacceptable implied scenarios
- Remove some scenarios
- Change the sequential order of some events
- Introduce new events or agents into the system

Validation
- Positive Scenario
- Negative Scenario

Automated Analysis

Scenario spec. (MSC)

Synthesis

Input-Output
Implied scenario

Input-Output
Behavior model (LTS)
Implied Scenarios (1/3)

- Definition
  - Not part of scenario-based model but present in every concurrent state machine model consistent with the scenarios
    - Scenario-based model (MSC)
      - System behaviors are viewed globally
    - Concurrent state machine model (LTS)
      - Each agent acts locally

- Detection
  - Check whether $\text{Bh}(M_{\text{min}}) \subseteq \text{Bh}(T)$
    - $M_{\text{min}}$ – minimal concurrent state machine model
    - $T$ – monolithic LTS model capturing global behaviors
  - Generate a trace $\text{tr}$ of $M_{\text{min}}$ to the point where it first deviates from $T$
Implied Scenarios (2/3)

- Model elaboration with implied scenarios
  - Make a decision whether implied scenarios are allowed to happen
    - Positive scenario
      - Allow to happen
      - Specify how to continue
      - Add in the hMSC graph
    - Negative scenario
      - Not allowed to happen or impossible to occur
      - Specify unwanted behavior
      - Modify initial set of scenarios
  - Iterate a process until no more implied scenarios are detected
Implied Scenario of Boiler Control System

- **Sensor** sends data to **Control**.
- **Control** processes data and sends commands to **Actuator**.
- **Actuator** receives commands and performs actions.

**Diagram:**

- **Boiler Control System**
  - **Sensor** sends data to **Control**.
  - **Control** processes data and sends commands to **Actuator**.
  - **Actuator** receives commands and performs actions.

- **Pressure** and **Query** flows are indicated in the diagram.

**Legend:**

- **Sensor**
- **Database**
- **Control**
- **Actuator**

**Flow:**

- **Pressure** flow from Sensor to Control
- **Query** flow from Sensor to Control
- **Data** flow from Control to Actuator
- **Command** flow from Actuator to Control

**Nodes:**

- Node 0: **Control**
- Node 1: **Sensor**
- Node 2: **Data**
- Node 3: **Actuator**
Ability to declare which events are monitored and controlled by agents is essential in requirements modeling language.

Description of the behaviors of an agent must
- Be defined in terms of agent’s monitored and controlled events
- Not constrain the occurrence of monitored events
- Not refer to the future occurrences of monitored events
Definition

Definition of implied scenario with constraint that state machine model (LTS) of each component cannot constrain monitored events.

Contradiction!

Scenarios for delivery system

LTS models for delivery system
Input-Output Implied Scenarios (3/4)

- Detection
  - Construct minimal concurrent state machine $M_{IO-min}$
  - Check whether $Bh(M_{IO-min}) \subseteq Bh(T)$

**LTS for Client**

**Input-output consistent LTS for Client**

**Additional sink state**
Model elaboration with IO-implied scenario

- Avoid undesirable system behavior by changing scenario
Model Elaboration Using IO-Implied Scenarios (1/3)

- Example 1. ATM model
  - Change the sequential order of events
Example 2. Web interface application
- Make some undistinguishable events distinguishable

Scenario of the ERP System

IO-Implied Scenario of the ERP System
Example 3. Toaster

- Identify missing events in the scenarios
Conclusion and Contribution

- Demonstrate monitoring and control are important issues when analyzing requirement-level scenarios
  - Define input–output implied scenario
    - Ensure an agent cannot inhibit the occurrence of events it monitors
  - Show how to detect input–output implied scenario and provide tool support
  - Demonstrate the relevance of flaws through a number of examples
Discussion

- Difficulties of elaboration process using input-output implied scenarios
  - Simple processing of validating implied scenarios by adding scenarios makes model complicated
    - Need refactoring
  - Correcting problems requires design decisions
    - Need support for such as adding new events, components and changing events order
  - Classification of implied scenarios as positive/negative is difficult
    - Need domain knowledge
    - Conflict views about required system properties among stakeholders
Define a formal framework integrating scenario-based and goal oriented requirements elaboration techniques within the LTSA toolset