Eliciting and modeling dependability requirements: a control case based approach

Introduction

- **Functional requirements (FR)** of a software address what services the to-be-built software is desired to deliver in terms of the business logics.
- **Dependability requirements (DR)** of a software address how the software will ensure the dependability of the delivered services when facing at various threats and changeful environment.
- Current RE approaches mainly focus on eliciting FRs. The control case based approach aims to systematically elicit and model the DRs by modeling a dependable software system as a feedforward-feedback control system.

Background

- Software situates in an open and dynamic environment.
- Some environmental entities are safety-critical, some are error-prone, some are malicious, etc.
- Software needs to be trustworthy.
- How to elicit and model these dependability related requirements is a challenge.

Feedforward-feedback control model of a dependable software

- **Core system**: delivers the desired services to the users according to the users’ desired behaviors.
- **Feedforward controller**: monitors the threats, and responds to them by imposing some controls on the core system.
- **Feedback controller**: monitors the behavior deviations of the software and responds to them by imposing some controls on the core system.

Feedforward-stimulation response cases

- **Software System**: Desired Environment behaviors
- **Core system**: System behaviors
- **Feedback controller**: System behavior deviations
- **Feedforward stimulus-response cases**: Threats (Attacks, Malicious Usage, Wrong Operations, etc)
- **Stimulation-response cases**: Uncertain Setting Changes

A HAZOP based process to guide the DRs elicitation

**Step 1**: take each use case as background, and identify the threats and system behavior deviations by following HAZOP.

**Step 2**: assess the risks of the identified threats and system behavior deviations.

**Step 3**: determine the cost-effective controls for the threats and system behavior deviations.

Future work

1. Enrich the knowledge base by including more instances of the concept classes.
2. Develop a systematical process with helpful guidance to elicit and model the dependability requirements.

This material is supported by Key Project of National Natural Science Foundation of China and National 973 Fundamental Research and Development Program of China.

Jin Zhi (zhijin@sei.pku.edu.cn)
Professor,
Key Lab of High Confidence Software Technologies (PKU), MOE
School of Electronics Engineering and Computer Science
Peking University, China