

Methoden des Software-Engineering

Dr. Philip Mayer, Dipl. Medien-Inf. Lenz Belzner

Aufgabe 1 Implementieren Sie eine Simulation des Heartbleed-Bugs in Java. Implementieren Sie die Klassen *Client* und *Server*. Der Server bietet eine Methode *heartbeat(String, Int) : String* an, die den Heartbeat des SSL-Protokolls implementiert. Verwenden Sie einen String zur Simulation des Server-Speichers. Schreiben Sie die Parameter eines heartbeat-Aufrufs durch den Client in den String (z.B. an einer zufälligen Stelle).

Wie können auf diese Weise sensible Serverdaten an Clients gelangen? Welches Prinzip des Dependable Programming wurde hier verletzt? Wie könnte das Problem in Ihrem Programm behoben werden?

Siehe auch <http://de.wikipedia.org/wiki/Heartbleed>.

Aufgabe 2 Erstellen Sie mit Xtext (<https://eclipse.org/Xtext/>) eine DSL zur Repräsentation von Aktionen eines Agenten (siehe Abb. 1). Aktionen haben einen Namen. Vor- und Nachbedingung einer Aktion sind jeweils Mengen von benutzerdefinierbaren Properties; Negation soll möglich sein. Ausserdem kann zu jeder Aktion eine Wahrscheinlichkeitsangabe getroffen werden. Wahrscheinlichkeiten sind diskret, z.B. 'low', 'medium' und 'high'.

```
prop open
prop closed

- action doOpen
  pre open not(closed)
  =>
  post closed not(open)
  with high probability

- action doClose
  pre closed not(open)
  =>
  post open not(closed)
  with medium probability
```

Abbildung 1: DSL Beispiel.

Erzeugen Sie eine Warnung, falls eine Property mehrfach in Vor- oder Nachbedingung vorkommt.

Zu einer benutzerdefinierten DSL-Datei soll eine Textdatei erzeugt werden, die die Namen aller definierten Aktionen enthält.

Wie würden Sie eine entsprechende DSL als interne DSL implementieren, z.B. in Java?