Problem 7 - The telephone problem

/~gibson/Teaching/CSC4504/Problem7-Telephone.pdf

You are to work in teams of 3 or 4
We are going to use Event-B contexts to formally model the behaviour of a telephone and a network of telephones.

How many states do we need?

- on hook
- ringing
- dialling
- talking
We are going to use Event-B contexts to formally model the behaviour of a telephone and a network of telephones.

Which type of network will we model?
We are going to use Event-B contexts to formally model the behaviour of a telephone and a network of telephones.

An abstraction
We are going to use Event-B contexts to formally model the behaviour of a telephone and a network of telephones

Step 1: implement a telephone in Java, write the unit tests and link each test with a formal requirement written in Event-B

Step 2: implement a telephone switch network in Java, write the unit tests and link each test with a formal requirement written in Event-B

Step 3: specify the invariants of the telephone and telephone system

Step 4: write a Java simulation of a telephone (network) and show that the system stays in a safe state.

Step 5: try to prove the safety of your design using RODIN
A quick revision of context language - the agatha problem

2.6 Introducing Contexts

Goals: In this section we introduce contexts that apply to the theoretical concepts that were introduced in the previous section (2.5). We will create a very simple model consisting of just one context file.

In this tutorial, we will create a model of the well known Agatha puzzle. We use this instead of the already introduced traffic light example because it provides us with more possibilities to apply Event-B’s logic and to use operations on relations. Here is a brief description of the puzzle:

Someone in Dreadsbury Mansion killed Aunt Agatha. Agatha, the butler, and Charles live in Dreadsbury Mansion and are the only ones to live there. A killer always hates, and is no richer than his victim. Charles hates no one that Agatha hates. Agatha hates everybody except the butler. The butler hates everyone not richer than Aunt Agatha. The butler hates everyone whom Agatha hates. No one hates everyone. Who killed Agatha?

Contexts are used to model static properties of a model, things that do not change over time. Whereas with machines we model the dynamic properties like the traffic light above. The objective of this section is to get familiar with contexts by modelling the Agatha puzzle.

2.6.1 Create a Context
2.6.2 Populate the Context
2.6.3 The Final Context

A quick revision of context language

2.6.3 The Final Context

CONTEXT
  agatha

SETS
  persons

CONSTANTS
  Agatha
  butler
  Charles
  hates
  richer
  killer
A quick revision of context language

axioms

person_partition:
  partition(persons, \{Agatha\}, \{butler\}, \{Charles\})

hate_relation:
  hates \in persons \leftrightarrow persons

richer_relation:
  richer \in persons \leftrightarrow persons \land
  richer \land id = \emptyset \land
  (\forall x, y, z \cdot (x \rightarrow y \in richer \land y \rightarrow z \in richer) \land
  \Rightarrow x \leftrightarrow z \in richer) \land
  (\forall x, y \cdot x \in persons \land y \in persons \land x \neq y
  \Rightarrow (x \rightarrow y \in richer \leftrightarrow y \rightarrow x \notin richer))

killer_type:
  killer \in persons

killer_hates:
  killer \rightarrow Agatha \in hates

killer_not_richer:
  killer \rightarrow Agatha \notin richer

charles_hates:
  hates[\{Agatha\}] \cap hates[\{Charles\}] = \emptyset

agatha_hates:
  hates[\{Agatha\}] = persons \setminus \{butler\}

butler_hates_1:
  \forall x \cdot (x \rightarrow Agatha \notin richer
  \Rightarrow butler \rightarrow x \in hates)

butler_hates_2:
  hates[\{Agatha\}] \subseteq hates[\{butler\}] \land
  (\forall x \cdot x \in persons \Rightarrow hates[\{x\}] \neq persons)

noone_hates_everyone:
  \forall x \cdot x \in persons \land hates[\{x\}] \neq persons

solution:
  \boxed{\text{theorem}} killer = Agatha