Workshop on Formal Methods in Human Computer Interaction (FoMHCI)

Search this site

FOMHCI WORKSHOP SUBMISSIONS USE CASES

IMPORTANT DATES PROGRAM COMMITTEES

FOMHCI WORKSHOP

SUBMISSIONS

USE CASES

IMPORTANT DATES

PROGRAM

COMMITTEES

Use Cases

On this page, two use cases are described based on which the submissions to the workshop should discuss the presented formal methods. For more information on the submission specification, go to the submission website.

Please consider the following points if preparing your submission to the workshop:

- The use cases should **guide the presentation of the formal method** as far as this enables the participants to understand the differences between the methods and discuss this in the workshop.
- The use cases **do not have to be considered in every detail**. The detailed descriptions are given to provide a full image of the use

case. Nevertheless, how detailed the chosen use case is considered in the presentation of the formal methods is left to the authors.

- Besides the discussion of the whole use case along the presented formal methods, also **parts of the use cases can be focused on** in the descriptions.

In the following, the two uses cases will be described in detail.

Use Case 1 - Control of a Nuclear Power Plant

Informal Description

The control of a Nuclear Power Plant is an activity which involves a high degree of automation in the control tasks. The operation of a nuclear power plant includes the full manual or partially manual starting and shut down of the reactor, adjusting the produced amount of electrical energy, changing the degree of automation by activating or deactivating the automated steering of certain elements of the plant, and the handling of exceptional circumstances. In case of the latter, the reactor operator primarily observes the process because the safety system of today's reactors suspends the operator step by step from the control of the reactor to return the system back to a safe state.

Figure 1 shows a simplified Boiling-Water Reactor (BWR) design [1]. It is comprised of three main components: the reactor core (lower left) containing the fuel elements and control rods, the turbine, which is connected to a generator, and the condenser, which condenses the steam generated by the core back to fluid water. The whole process is driven by water pumps: two pumps pumping feedwater from the condenser into the reactor core (WP1 and WP2) and one pump transporting water through the cooling pipes in the condenser (CP). Thus, the latter controls the amount of water returned from steam, which is available for being transported back into the reactor core.

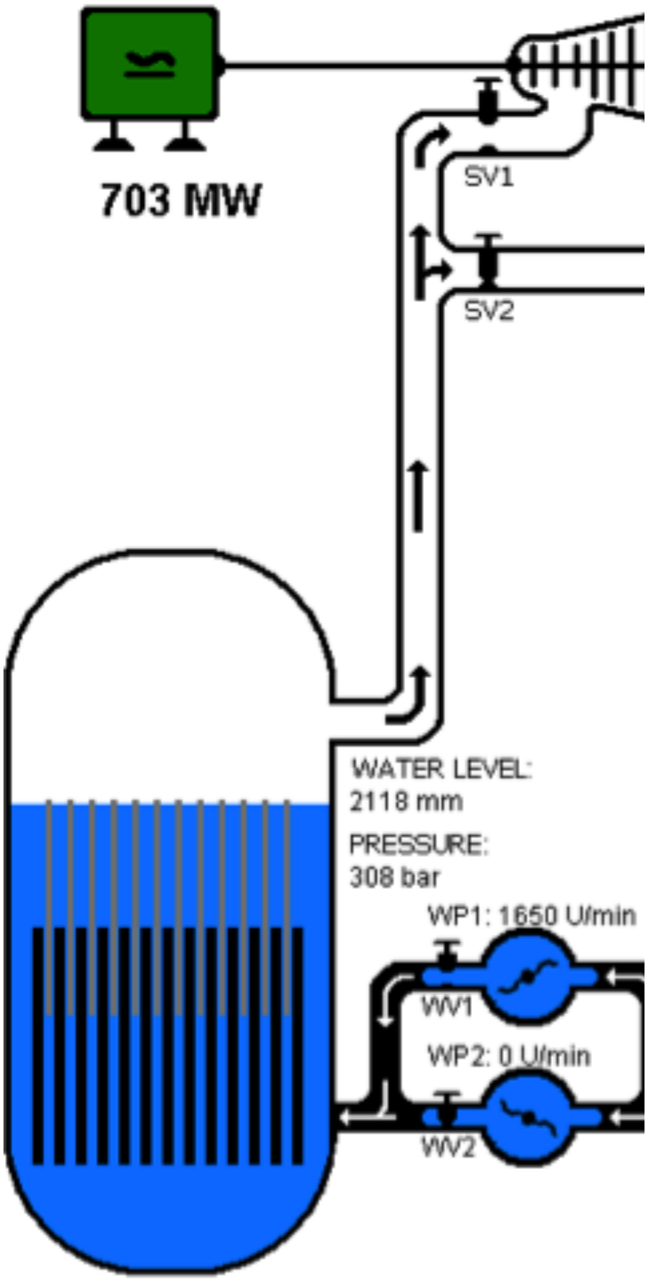

Figure 1: Sketch of a simplified

circuit of a Boiled-Water Reactor

The reactor is responsible for the production of heat in the system[1]. The amount of produced heat in the core and thereby the amount of steam producing electrical energy in the turbine is controlled by two parameters: (a) the amount of water pumped into the core and (b) the position of the control rods. Control rods are equipped with material, which is able to absorb neutrons and thereby reduce the chain reaction. Because the feedwater acts as moderator, increasing the amount of water pumped into the core enhances the amount of fissionable neutrons in the core, which thereby increases the amount of heat. A safe state of the reactor is given in the range of up to 70 bar pressure and up to 286 °C in the core [3]. Further control parameter are the water level in the reactor and the condenser as well as the output power at the generator. Finally, valves can be used to control the flow of water and steam in the system as can be seen in Figure 1.

Automation

Automation is an essential part in controlling a nuclear power plant. In the context of this use case, the simplified BWR as shown in Figure 1 will be considered in the context of the following description of automation. Therefore, two main aspects of automation can be separated: (a) automation of control of certain components in the BWR and (b) the safety system, which overtakes control regarding detected problems in the system to bring the BWR back into a safe state.

(a) Automation in Control

The following components can be set by automatic mode:

1. *Feedwater Pump:* The amount of feed water pumped into the reactor core is controlled along with the water level, the (pre-selected) output power, and the pressure in the reactor tank. Therefore, the automation controls the pump's speed.
2. *Control Rods:* The position of the control rods is controlled automatically along with the water level in the reactor, the (pre-selected) output power, and the pressure in the reactor tank.
3. *SCRAM:* The system offers an emergency reactor shutdown that shuts down the reactor automatically and immediately.

(b) Automation in Error Cases

In case of a system failure, the safety system kicks in excluding the user from the manual control of the system. This exclusion differentiates between 3 stages:

1. *Abnormal Operation:* This category specifies failures, which can be handled while the reactor is running without risks to the environment and the structure and function of the reactor.

2. *Design Basis Accident:* This category describes failures, which endanger the structure and function of the reactor. The system has to shut down immediately.
3. *Nuclear Accident:* This category describes failures which endanger the environment.

In case 1, the system partially excludes the user from the control. In this use case if certain system values are exceeded or underun certain boundaries, the system regulates the system back into a safe state. In case 2 and 3, the system excludes the operator completely from the system and shuts down the reactor by executing the SCRAM procedure.

User Interface of a Simplified Steam Water Reactor

The presented simplified BWR has been implemented as JAVA implementation, which includes the described automation features. It further offers a very simple graphical user interface. This is meant to enable the authors to first play around with the implementation, develop a certain view to it and finally use the simulator for their own submission, if relevant. This is NOT a requirement but an option for a possible submission. Figure 2 shows the simple GUI, which is only intended to be used for a first play around with the simulator and is NOT intended to be any kind of specification, etc. for a submission.

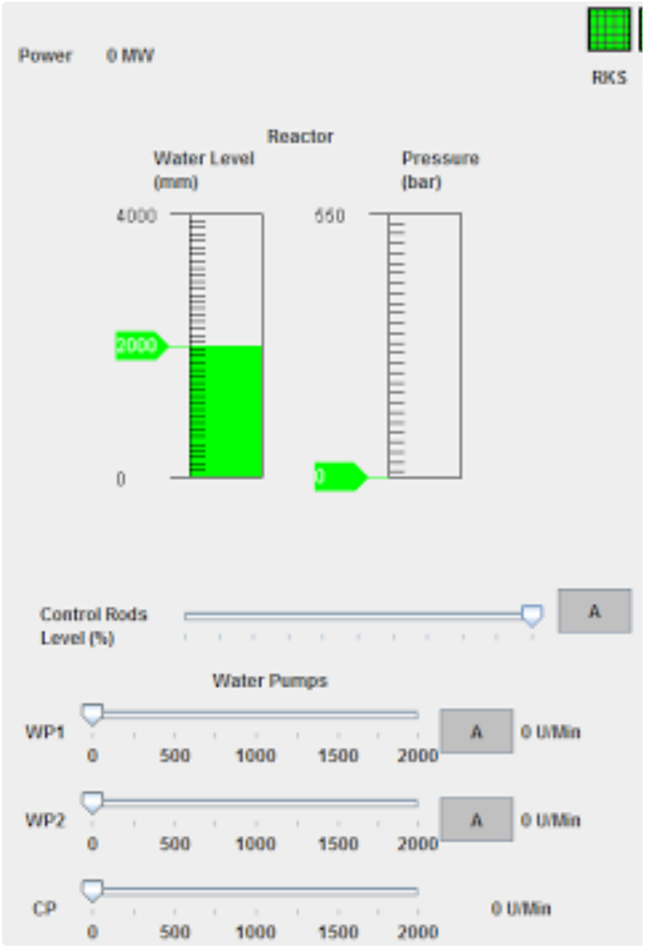

Figure 2: Graphical user interface for controlling the simplified BWR simulation.

Simulator

The simulator can be downloaded in two versions:

1. The "sandbox" version as executable .jar file starting the reactor with the simple user interface ---> Download | Description <---
2. The source code to include it into a demo implementation for the presentation of the formal method ---> Download | Documentation <---

For the sandbox demo, a description of possible control procedures are attached to this page.

[1] It is based on a nuclear chain reaction [2], which emits radiation and heat through nuclear fission of, normally, Uranium 235. Responsible for the fission are neutrons, which are themselves

emitted through previous fission. The precondition for fission is that emitted neutrons are slow enough, such that they are able to hit further fissionable atoms. Therefore, a moderator is needed in the reactor, which reduces the speed of neutrons emitted from previous fission. In case of a BWR, light water is normally used for moderation. Therefore, the feed water fulfills two functions: (a) being boiled to produce steam and (b) to moderate the reactor to produce slow neutrons used for the chain reaction.

References

[1] Nuclear Regulation Commission, "Boiling Water Reactors", http://www.nrc.gov/reactors/bwrs.htm last visited Dec 08, 2014.

[2] http://www.nrc.gov/reading-rm/basic-ref/glossary/chain-reaction.html

[3] Gesellschaft für Reaktorsicherheit (Society for Reactor Safety), "Siedewasserreaktor (SWR)", http://www.grs.de/aktuelles/begriff-des-monats-siedewasserreaktor-swr , last visited Dec 08, 2014, german.

Use Case 2 - Arrival Manager and Air Traffic Controller Radar Screen

Informal description

The Air Traffic Control activity in the TMA (Terminal Manoeuvring Area) is an intense collaborative activity involving at minimum two air traffic controllers (see image below) communicating with a set of aircrafts. The TMA is the area where controlled flights approach and depart in the airspace close to the airport.

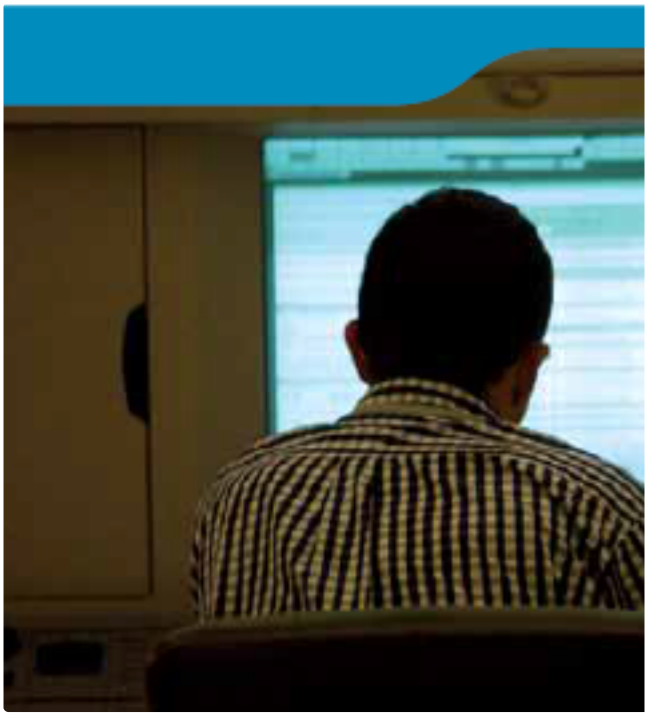

Figure 3: 2 TMA controllers working collaboratively

The planner controller (left-hand side) is in charge of planning clearances (orders) to be sent to pilots by the executive controller (right-hand side of figure) who uses a radar screen.

The AMAN (Arrival MANager) tool is a software planning tool suggesting to the air traffic controller an arrival sequence of aircraft and providing support in establishing the optimal aircraft approach routes. Its main aims are to assist the controller to optimize the runway capacity (sequence) and/or to regulate/manage (meter) the flow of aircraft entering the airspace, such as a TMA [1]. It helps to achieve more precisely defined flight profile and to manage traffic flows, in order to minimize the airborne delay, leading to better efficiency in terms of flights management, fuel consumption, time, and runway capacity utilization. The AMAN tool uses the flight plan data, the radar data, an aircraft performance model, known airspace/flight constraints and weather information to provide to the traffic controllers, via electronic display, two kind of information:

- A Sequence List (SEQ_LIST), an arrival sequence that optimizes the efficiency of trajectories and runway throughput (see Figure below)
- Delay management Advisories, for each aircraft in the ATCO's airspace of competence.

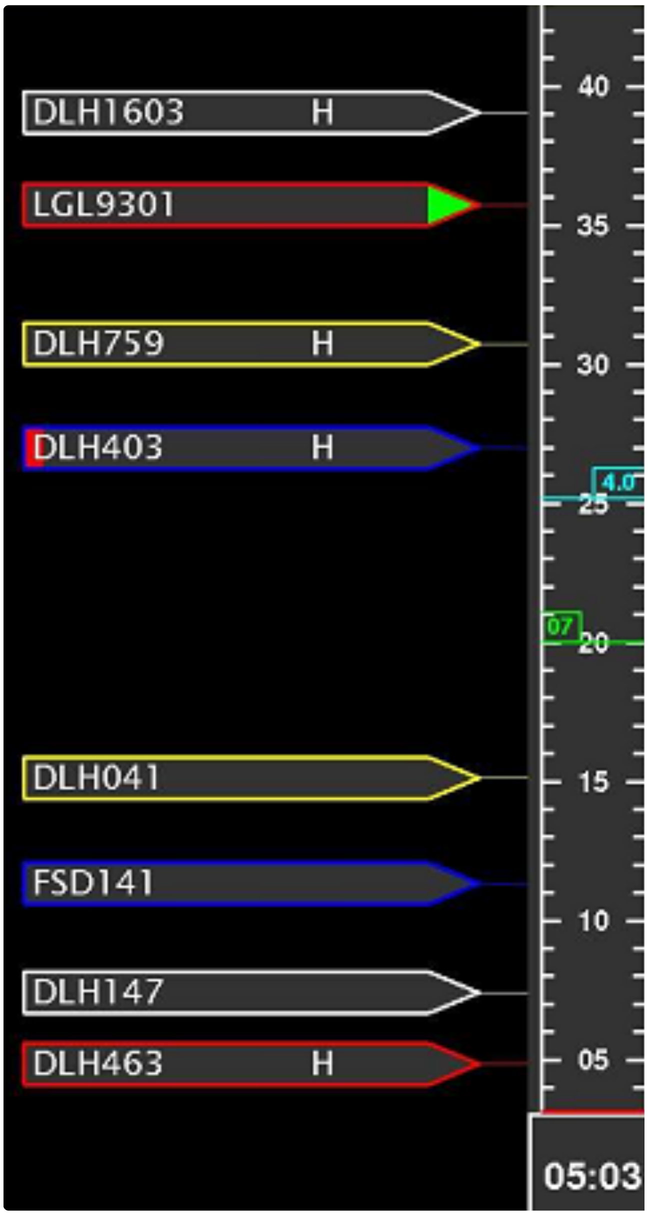

Figure 4: Screenshot of a subpart of an AMAN User Interface (arrival sequence)

The EXC_TMA is the controller deputed to handle the communications ground/air/ground, communicating to the pilots and releasing clearances to aircrafts. He/she has the tactical responsibility of the operations and he/she executes the AMAN advisories to sequence aircraft according to the sequence list.

For the case study scenario, we propose that the pilots assume a passive role, limited to the reception and execution of the clearances. Other more active roles (such as requesting an emergency landing) can be considered but are likely to make thing significantly more complex.

Air-Traffic Controller Tasks

Tasks of the EXEC_TMA air traffic controller is described below using the HAMSTERS notation [2] and [3]. The notation presented in [4] explicitly supports collaborative activities among users.

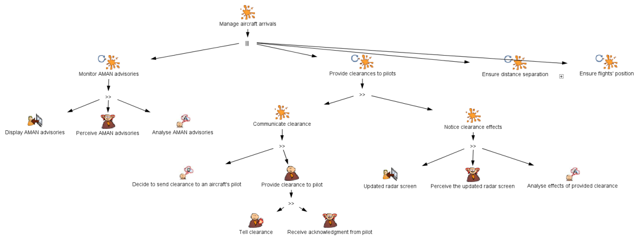

Figure 5: A task model of the management of arrivals in the TMA area

User Interface of AMAN

The Figure below show the Maestro user interface of AMAN (excerpt from document at the bottom of the page). We provide here several examples of UIs for AMAN.

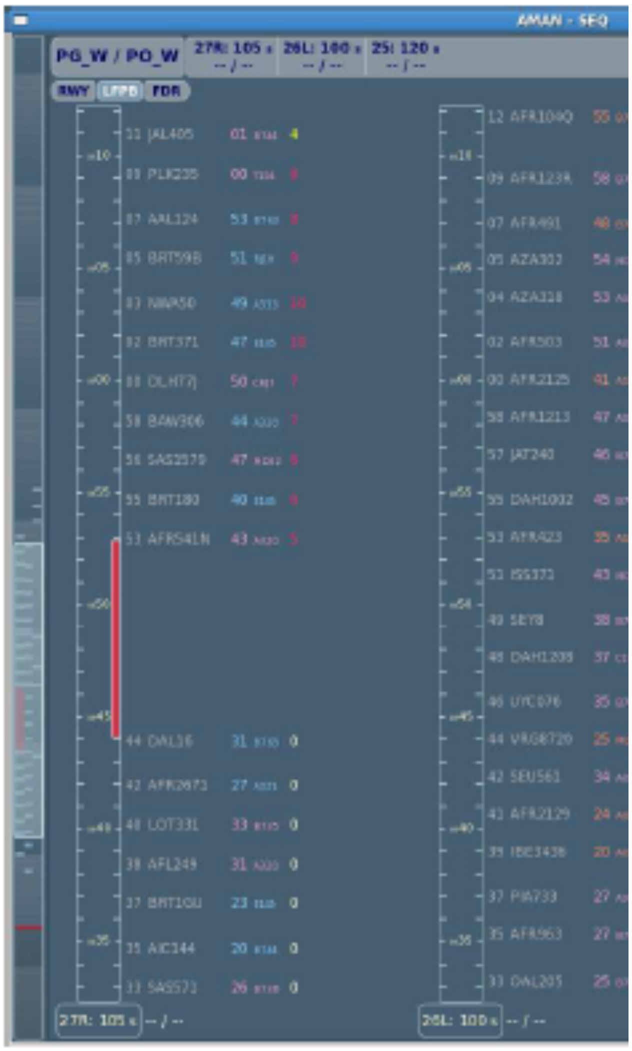

Figure 6: MAESTRO HMI - Runway view

User Interface of Radar Screen

An example of an ATC radar screen is presented in the figure below. On that figure one can see the labels associated with each aircraft including information such as aircraft callsign, cleared flight level, ... The line ahead of the aircraft spot is called the speed vector and describes the position of the aircraft in 3 minutes time. The longer the line the faster the aircraft. That line does not take into account the change in heading if any i.e. if the aircraft is changing heading then it will not be in 3 minutes where the speed vector is indicating. Behind the spot of the aircraft, the set of dots identify the previous positions of the aircraft (usually 5 of them).

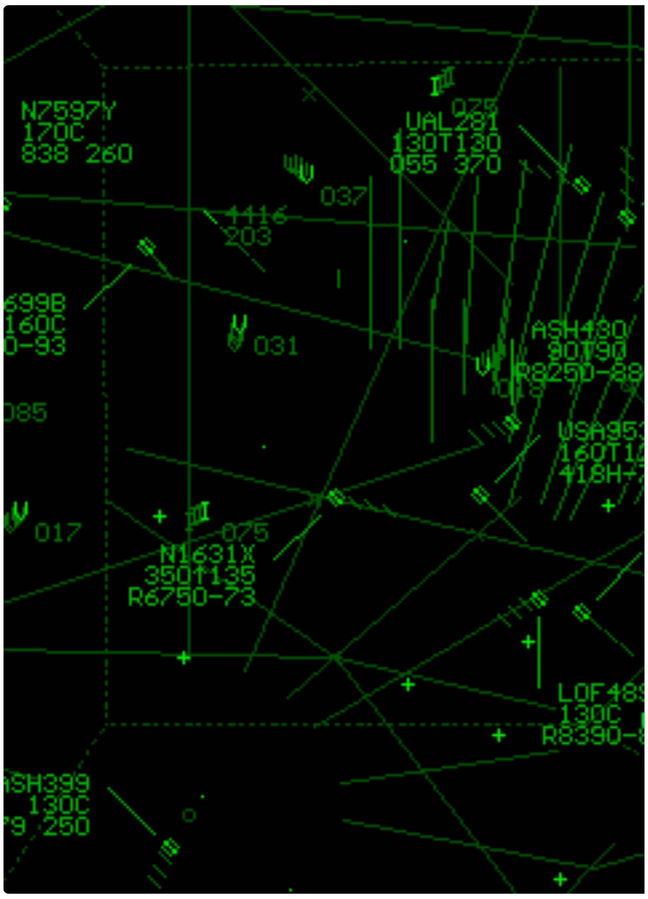

Figure 7: The ATC radar screen (each label representing an aircraft)

Documentation related to AMAN

More information about the AMAN application (as specified by Eurocontrol can be found here)

[1] EUROCONTROL, Arrival Manager. Implementation GUIDELINES and Lessons Learned. Edition 0.1, 2010

[2] Célia Martinie, Philippe A. Palanque, Marco Winckler: Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. INTERACT (3) 2011: 589-609

[3] Peter Forbrig, Célia Martinie, Philippe A. Palanque, Marco Winckler, Racim Fahssi: Rapid Task-Models Development Using Sub-models, Sub-routines and Generic Components. HCSE 2014: 144-163

[4] Célia Martinie, Eric Barboni,

David Navarre, Philippe A. Palanque, Racim Fahssi, Erwann Poupart, Eliane Cubero-Castan: Multi-models-based engineering of collaborative systems: application to collision avoidance operations for spacecraft. EICS 2014: 85-94

B...	BE...	V.1	⬇
FA...	PH...	V.1	⬇

Commentaires