Case Studies

Benjamin Weyers, Michael Harrison, Judy Bowen, Alan Dix, Philippe Palanque

Abstract This chapter introduces a set of case studies that are used in the rest of the book. They encompass well known problem domains in Human-Computer Interaction research and provide a practical focus for the approaches presented in this book. The set of case studies includes case studies concerned with: the controller interface to a (semi-automated) nuclear power plant; a partly-autonomous arrival management interactive system in the domain of air-traffic control; a user interface for new interactive cockpits and an interactive system used in rural and urban areas to maintain wind turbines. The final case study brings an interesting perspective for formal techniques, namely interactive public displays.

Introduction

This chapter introduces a set of case studies that encompass well known problem domains in Human-Computer Interaction research. They offer a basic set of case studies for the various approaches to formal methods presented in this book. The set includes case studies concerned with: the controller interface to a (semi-automated) nuclear power plant; a partly-autonomous arrival management interactive system in the domain of air-traffic control; a user interface for new interactive cockpits and an interactive system used in rural and urban areas to maintain wind turbines. The final case study brings an interesting perspective for formal techniques, namely interactive public displays. The first three case studies are presented in detail while the other two are presented more briefly. We categorize the case studies according to how many users are involved, whether there is a formal system definition, whether it is safety critical and what kind of interaction technique it offers (WIMP, post-wimp, etc.). Additionally, Table 1 specifies in which chapters of the book the presented case studies have been used. Beside these categories, some specific characteristics will be given under the category "others" that are specific to the individual case study. Besides aspects such as the number of persons involved or WIMP versus post-wimp, the case studies were selected according to the challenge they pose to formal methods in Human-Computer Interaction (these are indicated in the rightmost column in Table 1).

Case study 1 "nuclear power plant" offers a formal specification of a technical system and defines a set of standard operating procedures, which should be mapped

	#people involved	safety critical	formal specification	post-wimp	used in chapter	challenge to formal methods
1 – nuclear power plant	1	✓	✓	-[(1)]	5, 6, 10, 11, 14, 16, 19	- semi-automated system
2 – air traffic control	2-3	✓	-	✓	13	- automation - description of collaborative activities
3 – interactive aircraft cockpit	1-2	✓	-	-[(2)]	15, 17, 20	- need for reconfiguration mechanisms - conformity with certification requirement/need to deal with faults
4 – wind turbine maintenance	1	-	-	-[(3)]	9, 19	- distributed system
5 – public display	n≥1	-	-	✓	7, 12	- context information

Tab. 1 Classification of all presented case studies according to four characteristics and a set of specific attributes only addressing each individual case study.

[(1)] WIMP

[(2)] WIMP including multi-modal graphical input

[(3)] WIMP as physical implementation

to a user interface design and implementation. It further raises the challenge of implementing interactive systems for semi-automated systems and to answer the question as to how far formal methods can tackle the challenge of automation in the control of safety critical systems.

Case study 2 “air traffic control” picks up the combination of an automated and safety critical system and maps this scenario to a team of 2 to 3 controllers. Thus, it poses the challenge to formal methods to address multi-user interaction scenarios in a safety-critical context, which includes a high degree of automation and also post-wimp interaction techniques and concepts.

Errors and faults are a major issue in Human-Computer Interaction research, which is especially relevant for users who control safety critical systems but also for scenarios which address non-safety critical systems but for which performance

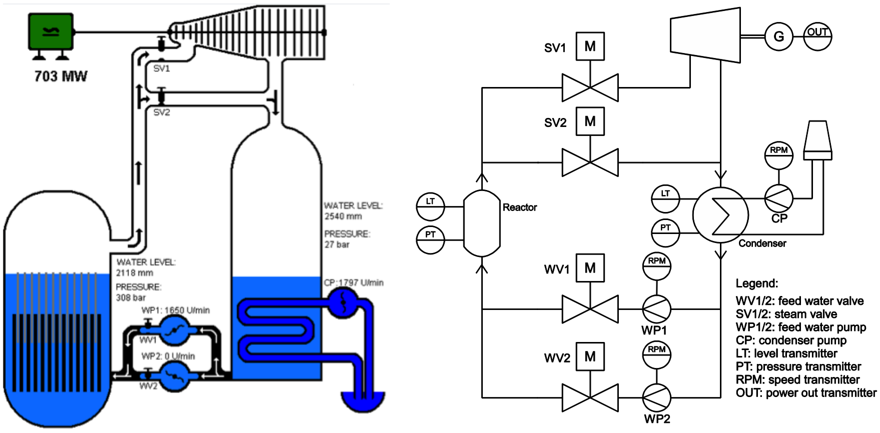


Fig. 1 Sketch of a simplified circuit of a Boiled-Water Reactor. On the left: the process is presented as a sketch; on the right: the process is presented as process-flow diagram.

is a concern. Case study 3 “interactive aircraft cockpit” presents the role of hardware and software failures in the context of system certification. The challenge posed here is to which extend formal methods can support the design and development of interactive systems and user interfaces for certified technical systems. For this case study software and hardware failures are addressed explicitly.

Case study 4 “wind turbine maintenance” and case study 5 “public display” leave the context of safety critical systems and pose other challenges to formal methods. Case study 4 describes a distributed and multi-device scenario which raises the problem domain of coordination between devices, the distribution of interaction and information as well as synchronous versus asynchronous communication. Case study 5 instead addresses post-wimp interaction in a complete different spatial context. It further poses the challenge of addressing a completely unknown user base which is heterogeneous regarding pre-knowledge, interaction behavior and training status. The question here is how formal methods support the development of such open-space interactive systems.

Case Study 1 - Control of a Nuclear Power Plant

The control of a Nuclear Power Plant involves a high degree of automation required to support the human controllers’ tasks. These tasks include the full manual or partial 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 management of certain elements of the plant, and the handling of exceptional circumstances. In the case of the latter, the reactor operator’s primary role is to observe the process as the reactor's safety system gradually suspends the operator from control until the system is returned to its safe state.

Figure 1 shows a simplified Boiling-Water Reactor (BWR) design (United States Nuclear Regulation Commission 2016) as catch on the left and as process flow diagram on the right. It comprises 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 reactor core back to fluid water. The whole process is driven by water pumps: two pumps pumping feedwater (purified water in contact with the reactor core) 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 then transported back into the reactor core where WP1 and WP2 control the amount of water pumped into the reactor core for cooling and steam production.

The reactor (core) is responsible for the production of heat in the system. 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 (WP1 and WP2) and (b) the position of the control rods. Control rods are equipped with material that is able to absorb neutrons and thereby reduce the chain reaction responsible for the emerging heat. Because the feedwater acts as moderator, increasing the amount of water pumped into the core increases the quantity of fissionable neutrons in the core, thereby increasing the heat. A safe state of the reactor core is specified as being in the range of up to 70 bar pressure and up to 286 °C (Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH 2016). Further control parameters and observables are the water level in the reactor and the condenser as well as the output power of the generator. Finally, valves can be used to control the flow of water and steam in the system, as seen in Figure 1. WV1 and WV2 are able to cut off the feedwater; SV1 controls the steam for the turbine and SV2 the by-pass of steam which is then sent directly to the condenser.

Formalization of the Simplified BWR Design

The previously introduced design for a BWR can be formalized in various ways and with different types of formalizations as is done in Chapter 6 using Z (O'Regan 2014) or in Chapter 16 using a domain specific and visual formal language. In this section, a formalization is given as a PVS language-based listing (Owre et al. 1992) originating from the work presented in Chapter 14, which has been slightly changed to the specification given in (Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH 2016).

Listing 1 specifies the boundary conditions in which the system can be considered as safe. If the system leaves one of these ranges, it has to be considered as unstable or critical.

Listing 1 Maximum values

```
max_vp                 : nat = 2
reactor_max_level      : nat = 4000
condensor_max_level    : nat = 8000
reactor_max_pressure   : nat = 550
condensor_max_pressure : nat = 180
max_pressure           : posnat = 900
control_rods_max_level : nat = 100
max_flow               : posnat = 2000
```

Listing 2 lists the various types used for the specification of the Nuclear Power Plant system. Some types are equipped with restrictions based on the boundaries defined in Listing 1, some are composed values, e.g. `valve_type`.

Listing 2 Type definitions

```
itimes          : TYPE = posnat
temperature     : TYPE = nonneg_real
rods_level_type : TYPE = {x: nonneg_real |
                          x <= control_rods_max_level}
pos_type        : TYPE = nonneg_real
volume_type     : TYPE = nonneg_real
press_type      : TYPE = {x: nonneg_real | x <= max_pressure}
speed_type      : TYPE = nonneg_real
flow_val        : TYPE = real
vp_number       : TYPE = {n: upto(max_vp) | n > 0}

valve_type      : TYPE = [# flow: flow_val, on: Boolean #]
valves_type     : TYPE = [ vp_number -> valve_type]
pump_type       : TYPE = [# speed: speed_type, on: Boolean #]
pumps_type      : TYPE = [vp_number -> pump_type]

process_type    : TYPE = [# level : volume_type,
                            pressure: press_type #]
```

The various types define the various components relevant in the process through their characteristic parameters, such as `temperature` or `rods_level_type` as specific characteristics for the reactor. The `process` which represents the system state of the modeled system has been defined as complex type (see Listing 3).

Listing 3 System definition as a tuple of system values.

```
IMPORTING definitions_th[]

npp: TYPE = [#
  time                 : itimes,
  sv                   : valves_type,
  wv                   : valves_type,
  wp                   : pumps_type,
  cp                   : pump_type,
  reactor              : process_type,
  pos_rods_reactor     : rods_level_type,
  old_pos_rods_reactor : rods_level_type,
  time_delta_pos       : itimes,
  condensor            : process_type,
  bw                   : temperature,
  poi_reactor          : pos_type,
  rest_heat            : temperature,
  cooled_water         : temperature,
  boiled_water        : temperature
#]
```

The nuclear power plant and the reactor as central part are defined as a set of values or parameters, which will be related to each other in the following specifications and descriptions and equipped with dynamic behavior in Listing 5. The first definition is the water flow (`flow_val`, Listing 4 line 2) for each water valve `wv`. In case if a water valve is opened and its associated water pump is running (e.g., WV1 with WP1, cf. Figure 1), the water flow is set to the current speed of the pump. Otherwise, the flow is set to 2 or -2 to simulate the effect of communicating vessels if the difference of the water level in the condenser and the reactor tank differs more than 470 mm (see Listing 4, ll. 7-11 and ll. 13-17).

Listing 4 Definition of the flow value for each water valve in the system.

```
1  flow_update(n: vp_number, st: npp):
2   flow_val =
3   COND (wv(st)(n)`on AND wp(st)(n)`speed > 0 AND
4          condensor(st)`level > 0)
5   -> wp(st)(n)`speed,
6
7   (wv(st)(n)`on AND wp(st)(n)`speed = 0 AND
8   condensor(st)`level > 0 AND
9   ((condensor(st)`level - reactor(st)`level) > 470) AND
10  (sv(st)(n)`on OR sv(st)(n)`on))
11  -> 2,
12
13  (wv(st)(n)`on AND wp(st)(n)`speed = 0 AND
14  condensor(st)`level > 0 AND
15  ((condensor(st)`level - reactor(st)`level) < 470) AND
16  (sv(st)(n)`on OR sv(st)(n)`on))
17  -> -2,
18
19  ELSE -> 0
20  ENDCOND
```

The dynamic change of the system is defined as function `tick`, which describes the time-dependent evolution of the individual system values specified in Listing 3. Therefore, Listing 5 specifies the update of the various system values for each time step for an instance `st` of the nuclear power plant process type `npp`. First, the timer is increased (l. 3). In the next step, the steam flow through the two steam valves is updated (ll. 6-13 for SV1 and ll. 15-21 for SV2). Additionally, for both valves, the current on-status is propagated (l. 13 and l. 21).

During operation, the reactor gets effected by chemical reactions that change the behavior of the reactor, reflected in a poisoning factor (`poi_reactor`, ll. 25-34). The poisoning of the reactor is positively correlated with the change in the control rod position over time (l. 26,27). It influences the produced amount of steam. If the poisoning is high, the reactor generates more heat because the poisoning chemical acts as moderator. As a stopped reactor still produces heat because of fuel decay, a factor of rest heat (`rest_heat`) is calculated if the reactor is shut off (`pos_rods_reactor(st) = 100`, l. 37). The amount of boiled water (`boiled_water`, l. 46-58) is calculated most simply using the amount of water boiled by the chain reaction (factor `bw`, ll. 43,44), the rest heat and the poisoning factor as multiplier. The produced cooled water (`cooled_water`, ll. 60, 61) is calculated depending on the speed of the condenser pump CP and the pressure in the condenser vessel. The water pumped from the condenser into the reactor vessel is determined using the `flow_update` function given in Listing 4 (ll. 63-71). Finally, the reactor (ll. 73-78) and condenser (ll. 80-85) pressure and water level parameter are updated according to the boiled water in case of the reactor and cooled water in case of the condenser vessel.

Listing 5 Dynamic behaviour of the nuclear power plant as discrete simulation.

```
1  tick(st: npp):
2    npp = st WITH [
3      time := time(st) +1,
4
5      sv := LAMBDA (n: vp_number):
6        COND n=1
7        -> (# flow :=
8           COND sv(st)(1)`on
9           -> (reactor(st)`pressure - condensor(st)`pressure)/10,
10
11          ELSE -> 0
12          ENDCOND,
13          on := sv(st)(1)`on #),
14
15       n=2
16       -> (# flow :=
17          COND sv(st)(2)`on
18          -> (reactor(st)`pressure - condensor(st)`pressure)/2.5,
19
20          ELSE -> 0
21          ENDCOND,
22          on := sv(st)(2)`on #)
23       ENDCOND,
24
25     poi_reactor :=
26       LET num_reactor =
27           (old_pos_rods_reactor(st) - pos_rods_reactor(st))
28
29       IN (
30         COND num_reactor >= 0
31         -> num_reactor / (time(st) - time_delta_pos(st)),
        ELSE -> -num_reactor / (time(st) - time_delta_pos(st))
        ENDCOND),

    rest_heat :=
      COND (pos_rods_reactor(st) = 100) AND (poi_reactor(st)=0)
      -> rest_heat(st)/1.05,

      ELSE -> 0
      ENDCOND,

    bw := (2*(100 - pos_rods_reactor(st))
          *(900-reactor(st)`pressure))/620,

    boiled_water :=
      COND (pos_rods_reactor(st) = 100 AND poi_reactor(st) = 0)
      -> bw(st) + rest_heat(st),

       (pos_rods_reactor(st) = 100 AND poi_reactor(st) > 0)
      -> (bw(st) + rest_heat(st))*poi_reactor(st),

       (pos_rods_reactor(st) < 100 AND poi_reactor(st) = 0)
      -> bw(st),

       (pos_rods_reactor(st) < 100 AND poi_reactor(st) > 0)
      -> bw(st)*poi_reactor(st)
      ENDCOND,

    cooled_water := 0.003 * cp(st)`speed *
                    SQRT(condensor(st)`pressure),

    wv := LAMBDA (n: vp_number):
      COND n=1
      -> (# flow := flow_update(1, st),
            on   := wv(st)(1)`on #),

      n=2
      -> (# flow := flow_update(2, st),
            on   := wv(st)(2)`on #)
      ENDCOND,

    reactor :=
     (#pressure := 0.25 * (reactor(st)`pressure - sv(st)(1)`flow -
                   sv(st)(2)`flow + boiled_water(st)),

       level    := reactor(st)`level + wv(st)(1)`flow +
                   wv(st)(2)`flow - boiled_water(st)#),

    condensor :=
     (#pressure := condensor(st)`pressure + sv(st)(1)`flow +
                   sv(st)(2)`flow - cooled_water(st),

       level    := condensor(st)`level - wv(st)(1)`flow -
                   wv(st)(2)`flow + 4 * cooled_water(st)#)
```

Standard Operating Procedures

For the operation of the presented system, three illustrative standard operating procedures (SOP) are presented below. These procedures explain how the reactor controller is to define (e.g. through a user interface) the various parameters of the system. The *SOP Start-up* defines how to start the reactor to produce electrical energy where the *SOP Shut-down* defines the stopping of the reactor. The *SOP System Failure WP1* is an example of how the reactor controller should react if WP1 has a failure and stops working.

SOP Start-up

Aim

1. Bring output power to 700 MW, i.e. 100% of possible output power.
2. Hold water level in the reactor tank stable at 2100 mm.

SOP

1. Open FV2.
2. Set KP to 1600 U/min.
3. Open WV1.
4. Set WP1 to 200 U/min.
5. Stabilize water level in the reactor tank at 2100 mm by pulling out the control rods.
6. Open FV1.
7. Close FV2.
8. Increase U/min of WP1 and, in parallel, pull out control rods so that the water level in the reactor tank is stable at 2100 mm.
9. At 700 MW power output, stop pulling out the control rods. Water level of the reactor tank has to be stable at 2100 mm.

SOP Shut-Down

Aim

1. Reduce output power to 0 MW.
2. Hold water level in the reactor tank stable at 2100 mm.

SOP

1. Reduce output power to 200 MW by reducing the speed (U/min) of WP1 and, in parallel, push the control rods into the core. Hold water level in the reactor tank stable at 2100 mm.
2. Open FV2.
3. Close FV1.
4. Reduce output power to 0 MW so that:
    a. Control rods are completely pushed into the core,
    b. WP1 is stopped and
    c. the water level in the reactor tank is stable at 2100 mm.
5. Close WV1.
6. Set KP to 0 U/min.
7. Close FV2.

SOP System Failure of WP1

Aim

1. Prevent a reactor meltdown.
2. Reduce power output to 0 MW in case of breakdown of WP1.
3. Hold water level in the reactor tank stable at 2100 mm.

SOP

1. Discharge control rods immediately into the core.
2. Open WV2.
3. Set WP2 to 800 U/min.
4. Control WP2 in such a way that the water level in the reactor tank stays stable at 2100mm.
5. After residual heat is completely dissipated and pressure in the reactor tank and the condenser is equal to 0 bar do:
    a. close FV1, FV2, WV1, WV2 and
    b. set all pumps to 0 U/min.

Automation

Automation is an essential part in controlling a nuclear power plant. In the context of this case study, 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) automation of the safety system, which takes control of the reactor when problems are detected in the system to bring the BWR back to a safe state.

(a) Automation in Control

The control of the BWR can be automated as follows:

1. *Feed Water Pumps:* The amount of feed water pumped into the reactor core can be controlled by means of the water level (to keep it at, e.g., 2200mm in the reactor tank), the (pre-selected) output power, and the pressure in the reactor tank. Therefore, the automation controls the pump's speed as variable. The used strategy can be manifold, e.g., using a linear transfer function.
2. *Control Rods:* The position of the control rods can be controlled automatically by keeping the water level and pressure constant in the reactor vessel, to reach the (pre-selected) output power.
3. *SCRAM:* The system could additionally offer an operation implementing an emergency reactor shutdown that pushes the control rods completely into the reactor core to stop the chain reaction, opens both feed water valves, sets all pumps to a predefined speed as well as shuts done the steam flow to the turbine and opens the pass-by channel for the keeping the circulation of steam and water open. The constant water flow ensures the ongoing cooling of the reactor core after the shutdown.

(b) Automation in Error Cases

In a real nuclear power plant, the safety system excludes the user from the manual control of the system when there is a recognized system failure. This exclusion differentiates between 3 stages:

1. *Abnormal Operation:* This category specifies failures that 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 that endanger the structure and function of the reactor. It is necessary for the system to shut down immediately.
3. *Nuclear Accident:* This category describes failures which endanger the environment. The system is shut down automatically and the controller is not able to restart it.

In case 1, the system partially excludes the user from the control. If certain system values are exceeded or do not reach certain boundaries, the system regulates itself 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.

Connection with Formal Methods

The main challenge for an interactive system in scenario a) (automation in control) is to offer a relevant interaction mechanism to monitor the automated systems as well as to offer relevant interaction operations, which enable the controller of the system to react if necessary. For instance, this is the case if failures occur which cannot be handled by the system but needs intervention by the system controller. An additional research question is in how far the mental model of the controller is influenced by this partial automation of the system control. Both, the design of the interactive system as well as the reaction of the user on the partial automation offers a variety of challenges to be addressed in HCI research from which the use of formal methods can benefit.

As in scenario a), the perspective on the system's design in scenario b) (automation in error cases) as well as on the user raises various research questions. Compared to scenario a), scenario b) considers full automation as well as the exclusion of the user in some circumstances. Based on this, the presented system allows the investigation of system designs as well as the user's behavior in cases of error-driven automation of the system, which is a specific type of research on the interaction with automated systems.

Case Study 2 - Arrival Manager within an Air Traffic Control Workstation

The Air Traffic Control activity in the TMA (Terminal Maneuvering Area) is an intense collaborative activity involving at minimum two air traffic controllers (see Figure 2) communicating with more than one aircraft. The TMA is the area where controlled flights approach and depart in the airspace close to the airport. The planner controller (left-hand side of Figure 2) is in charge of planning clearances (orders) to be sent to pilots by the executive controller (right-hand side of Figure 2) who uses a radar screen.

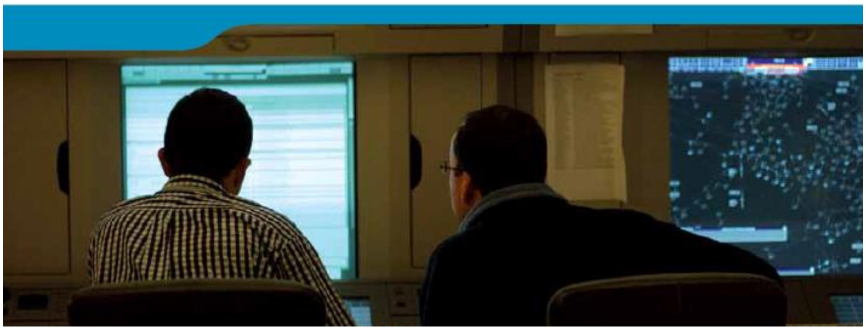


Fig. 2 Two TMA controllers working collaboratively

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 (by determining the sequence) and/or to regulate/manage (meter) the flow of aircraft entering the airspace, such as a TMA (EUROCONTROL 2010). It helps to achieve more precisely defined flight profiles and to manage traffic flows, to minimize the airborne delay, and to lead to improved efficiency in terms of flight 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 kinds of information:

- A Sequence List (SEQ_LIST), an arrival sequence that optimizes the efficiency of trajectories and runway throughput (see Figure 3)
- Delay management Advisories, for each aircraft in the ATCO's airspace of competence.

The EXC_TMA is the controller delegated to handle the ground/air/ground communications, communication with pilots and releasing clearances to aircraft. 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 things significantly more complicated.

Air-Traffic Controller Tasks

Tasks of the EXEC_TMA air traffic controller is described in Figure 4 using the HAMSTERS notation (Martinie et al. 2011, Forbrig et al. 2014). The notation, presented in (Martinie 2014), explicitly supports collaborative activities among users. Figure 4 details the "Manage aircraft arrivals" task (first row of Figure 4). This task consists of performing concurrently four different tasks (second row of Figure 4): monitoring AMAN advisories, providing clearances to pilots, ensuring distance separation between planes and ensuring flights' positions. The "monitor AMAN advisories" abstract task is refined as follows (see row 4 of Figure 4). First the AMAN system displays the advisories, then the air traffic controller perceives these advisories, finally, the air traffic controller analyzes these advisories. This task is followed by the "Provide clearance to pilots" task that is further refined in Figure 4. Due to lack of space, "Ensure distance separation" and "Ensure flights' position" tasks are ellipsised in Figure 4.

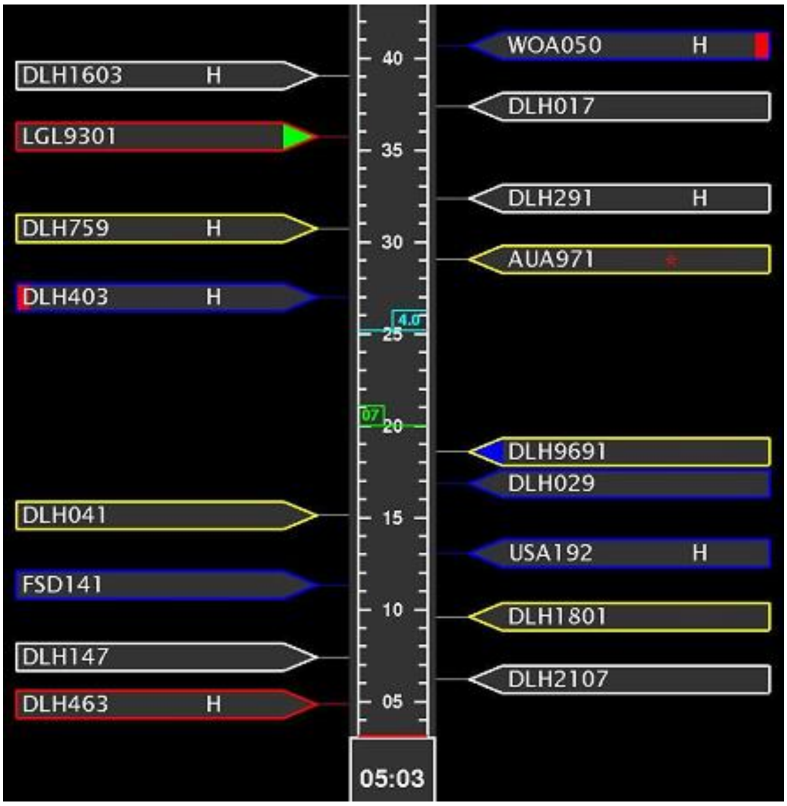


Fig. 3 Screenshot of a subpart of an AMAN User Interface (arrival sequence)

User Interface of the Air Traffic Control Radar Screen

An example of an ATC radar screen is presented in Figure 5. In this figure one can see the labels associated with each aircraft including information such as aircraft callsign, cleared flight level, etc. 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 where the speed vector indicates in 3 minutes. Behind the spot that indicates the position of the aircraft, the set of dots identifies the previous positions of the aircraft (usually 5 of them).

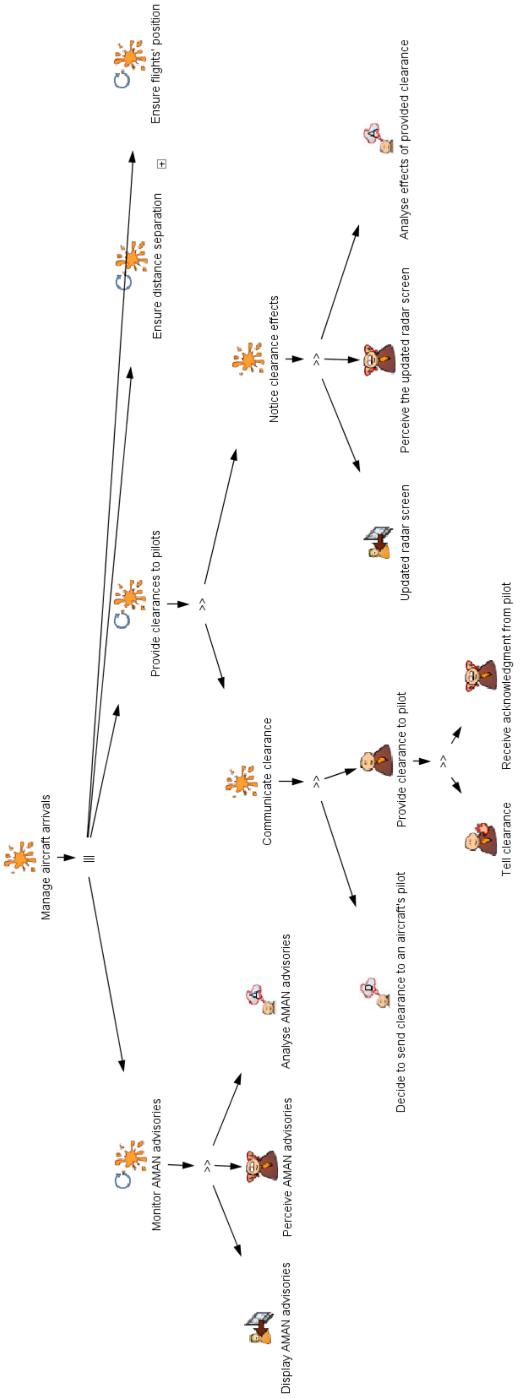


Fig. 4 A task model of the management of arrivals in the TMA area

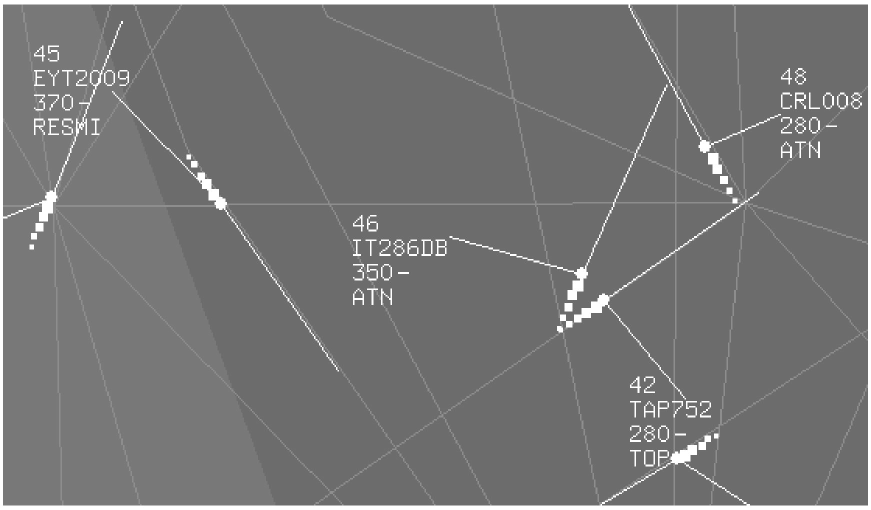


Fig. 5 The ATC radar screen (each label representing an aircraft)

This case study addresses various challenges for formal methods in HCI. First, the tasks that have to be performed by the air traffic controllers involve several operators using a complex workstation with multiple screens and multiple input devices. Second, the operators' tasks are highly collaborative, including collocated and remote collaborations (e.g., communications between the two air traffic controllers and between the air traffic controllers and the aircraft pilots within the sector). It also highlights the integration of partly-autonomous support to operator's tasks (via the AMAN software) that raise the issue of complacency, situation awareness, control and behavior forecasting of the autonomous systems.

For many years interaction with the ATC workstations was limited to zooming on a CRT (Cathodic Ray Tube) display. Recently a lot of effort has been deployed in integrating new technologies such as tactile interactions which raises more constraints in terms of specification of interactive systems and their related interaction techniques.

Case Study 3 – Interactive Aircraft Cockpits

With the introduction of the ARINC 661 specification (Airlines Electronic Engineering Committee 2006) in the early 2000's, the new generation of aircraft (e.g., Airbus A380, A350 WXB, Boeing 787 …) include graphical interfaces in their cockpits. These graphical interfaces are applications that feature graphical input and output devices and interaction techniques such as you would find in any other digital interactive systems (office and home computers, web applications …). The set of

Graphical User Interfaces (GUIs) in the cockpit is called the Control and Display System.

As illustration, the example of the Airbus A380 cockpit is presented in Figure 6. The Control and Display System is composed of eight output devices called Display Units (composed of a LCD screen, a graphics processing unit and a central processing unit) and two input devices called Keyboard and Cursor Control Units (KCCUs). The pilots can interact with some of the applications displayed on the Display Unit by using the keyboard and track ball of the KCCU.

This section first presents an example of an interactive application within interactive cockpits as defined by the Flight Control Unit Software (FCUS). The interactive cockpits architecture (instantiated within the FCUS application) is then detailed, followed by a description of the tasks that have to be performed by the pilots when using this application. Finally, the connection between this case study and formal approaches is discussed.

The FCUS Application

This case study focuses on a single application: the FCUS application (see Figure 8). It is inspired by the FCU Backup application that is designed to allow the crew members to interact with the Auto-Pilot and to configure flying and navigation displays. It is composed of two interactive pages Electronic Flight Information System Control Panel (EFIS CP) (left-hand side of Figure 7) and Auto Flight System Control Panel (AFS CP) (right-hand side of Figure 7) and is displayed on two (one for each flying crew member) of the eight Display Units. The crew members can interact with the application via the Keyboard and Cursor Control Units.

Figure 8 details the interactive system architecture in the cockpit. In this figure, we present the display of the FCUS on a single Display Unit. The interactive system architecture in the cockpit and the interactive applications are based on ARINC 661 specification (Airlines Electronic Engineering Committee 2006) as this is the required standard in the area of interactive cockpits for large civil aircraft. More precisely, the ARINC 661 standard specifies firstly the communication protocol between the Control and Display System and the aircraft system and secondly the software interface of interactive objects (namely, the widgets).

The Control and Display System is composed of the following components (as depicted in Figure 8):

- ***FCUS widget set:*** is composed of a set of all the interactive graphical elements called widgets composing the FCUS application. The widgets are organized in a hierarchical way and correspond to the interactive facilities for the FCUS User Application. The hierarchical widgets organization is managed by the server.

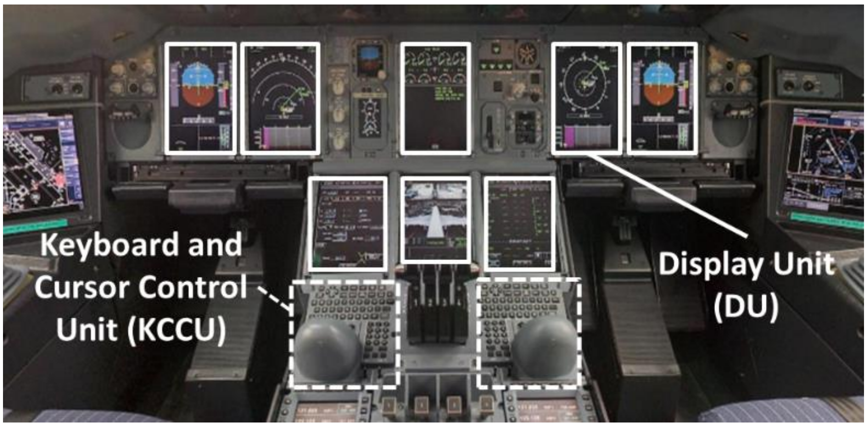


Fig. 6 Airbus A380 interactive cockpit

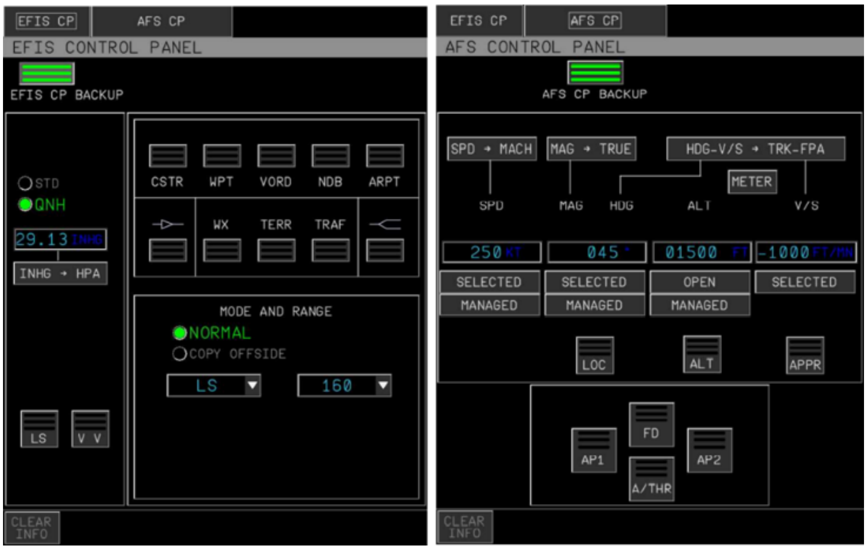


Fig. 7 The two windows of the FCUS application, inspired by the A380 FCU Backup

- ***Server:*** is responsible for i) the management of the Keyboard and Cursor Control Unit graphical cursor, ii) the rendering of graphical information on the DU, iii) the management of the widget hierarchy and iv) the dispatching of Keyboard and Cursor Control Units events to the targeted widgets (usually called picking in HCI). The server is also in charge of the management of several widget sets, each one corresponding to one User Application.

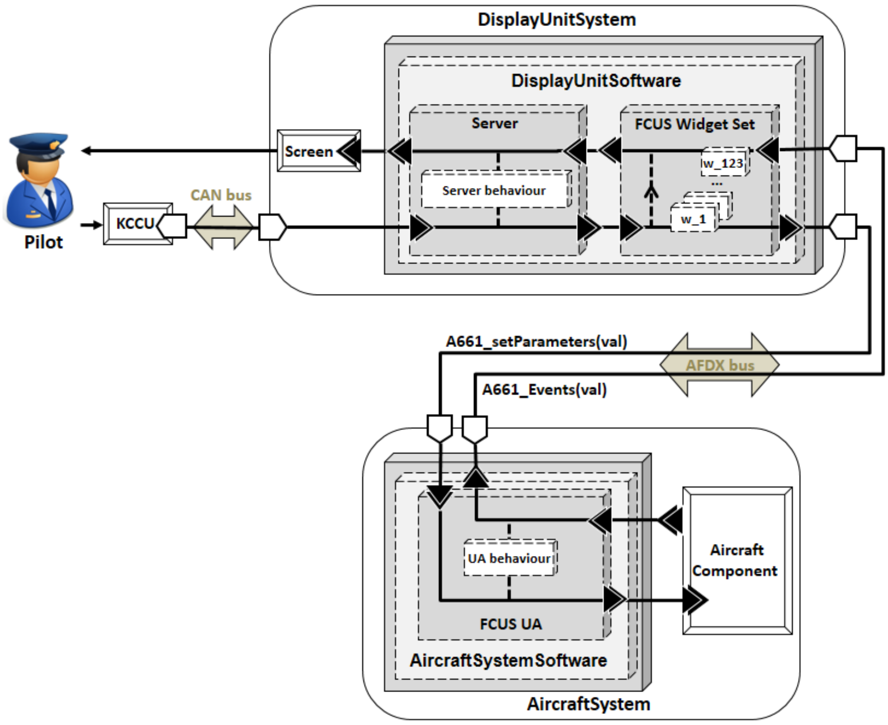


Fig. 8 Interactive cockpits architecture exemplified with the FCUS application

- ***Input and output devices:*** Keyboard and Cursor Control Unit and LCD screen (Display Unit). They allow the interaction between crew members and the interactive system. They are related to software components (device drivers) that are not pictured here and that will not be considered in this paper.

As presented in Figure 8, following the ARINC661 protocol, the Control and Display System provides information for:

- ***Crew members (captain and first officer):*** their role is to fly the aircraft by i) monitoring all aircraft systems through their associated displays (LCD screens) and ii) controlling these systems through the associated input devices (Keyboard and Cursor Control Units). Their goal is to maintain the aircraft and to complete the mission i.e. take the aircraft from the departing airport to its destination.

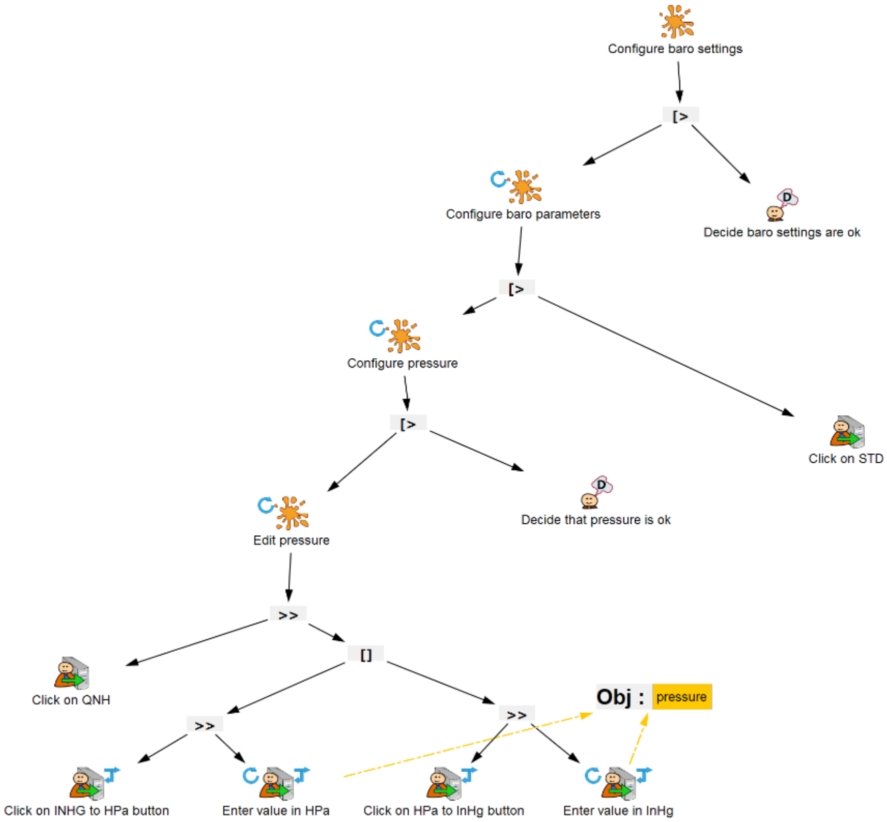


Fig. 9 A subset of the tasks of the flying crew interacting with the FCUS

Aircraft systems: they are composed of two components: the first one called User Application (in this example the FCUS) is the software interface to the physical aircraft component. The User Application is responsible for managing information to and from the Control and Display System: it processes the *A661_Events(val)* from the Control and Display System and triggers commands related to the physical aircraft components. The User Application can update the application display through the *A661_setParameters(val)* methods (applicable to the graphical widgets) to provide feedback to the crew about the actual state of the aircraft component. Graphically speaking, the User Application consists of one graphical user interface composed of a set of widgets stored in the Control and Display System. Its behavior defines the availability of widgets (enabled, visible, ...) that are not presented here for sake of brevity (more details can be found in Barboni et al. (2006)). The flying crew interacts with this user interface to perform their operations by triggering commands and perceiving the status of the aircraft system.

Pilots Tasks

The tasks of the flying crew are rather complex covering both aircraft and mission management as explained above. Providing a full description of these tasks goes beyond the level of details of the description of these case studies. However, Figure 9 provides a description of a subset of these tasks dedicated to the configuration of the barosettings (upper left part of Figure 7). Before landing, crew members may be asked to configure the barometric pressure as relevant to the local airport context. The barometric pressure is used by the altimeter as an atmospheric pressure reference in order to process correctly the plane altitude. When the pilot is asked to enter a new value for the pressure reference, he/she first chooses the QNH mode (interactive input task "Click on QNH"). Then he/she configures the pressure unit by choosing hPa (interactive input task "Click on InHg to hPa button") or InHg (interactive input task "Click on hPa to InHg button"). He/she can then choose to edit the value in hPa (interactive input task "Enter value in hPa") or in InHg (interactive input task "Enter value in InHg"). The STD mode (interactive input task "Click on STD") is used during the cruise.

Connection with Formal Methods

This case study addresses various challenges for formal methods in HCI. First, it corresponds to an embedded interactive system used for controlling a safety critical system. As this presents a public risk, this system must be conformant with certification requirements as expressed in DO-178C (RTCA and EUROCAE 2012) and CS 25 (EASA 2014). While addressing these certification needs is common in airborne software, HCI has so far remained out of the loop and certification of interactive systems remains an agenda for the research community.

On the contrary, the interaction techniques involved in the interactive cockpit context remain rather simple and follow the transposed IBM CUA 89 (IBM 1989) WIMP standard to interactive cockpit (called ARINC 661 specification (Airlines Electronic Engineering Committee 2006)). However, it is important to note that the presence of two graphical input devices (the KCCU of the captain and the one of the first officer) brings the issue of multimodal interaction (similar to bimanual interactions for a single user (Bier et al. 1993)).

Lastly, the requirements for dependability of the interactive cockpits require taking into account possible failures of hardware and software calling for reconfiguration mechanisms (of the interactive system – moving required information from a faulty display to functioning one). The same holds for interaction technique reconfiguration in case of input devices failures (Navarre et al. 2008).

Case Study 4 - Interactive Systems in Rural Areas – Maintenance of Wind Turbines

Wind turbines as a technical facility to produce electrical energy from wind get more and more important in times of depleting fossil sources and reduced acceptability of atomic energy within the community. Wind turbines are often co-located in huge groups, so called wind parks, or can be single installations. They are almost exclusively found in rural areas. To keep the reliability of these installations high and reduce technical failures, regular maintenance is necessary. This is applied by trained technicians who follow specified processes offered by the producing company[1]. These involve access to technical information. This information includes details of the inspected installation (such as previous maintenance and repairs), as well as the maintenance process. The latter specifies which regular repairs have to be applied to the installation, which inspections have to be done (such as visual inspections of technical parts or temperature inspection using infrared cameras) and what the documentation looks like.

Interactive systems are mainly used in this scenario for: leading the maintenance process; providing general information for the maintenance as well as process specific details of the procedure; enable documentation; and finally to plan and apply the maintenance process. This last use comprises the plan of which wind turbines have to be inspected, where to find them, and how different technicians are to coordinate. The information collected during maintenance is further post-processed to report the costs, trigger further repair requests if the inspection reveals problems that could not be fixed during the maintenance or to schedule the next regular maintenance.

Tilley - a community wind turbine

Tilley is a community owned 900kw Enercon E44 wind turbine installed in 2009 (TREL, 2016). Power from Tilley feeds into the grid and income from this is used to fund various island community projects.

The island's electricity supply is connected to its neighbouring island Coll via an undersea cable, and this in turn is connected with the larger island of Mull and from there to the mainland and National Electricity Grid (see Fig. 10). The island typically consumes between 1Mw and 2Mw, including the power for the 'golf ball', the radar that serves civilian North Atlantic air traffic control.

As Tilley can provide such a large proportion of the island's electricity its correct functioning is particularly important as a malfunction could distort the island electricity supply. This is a particular issue when there are problems in the undersea

[1] http://archive.northsearegion.eu/files/repository/20120320111424_PC_Skills-Compendiuminmaintenance.pdf

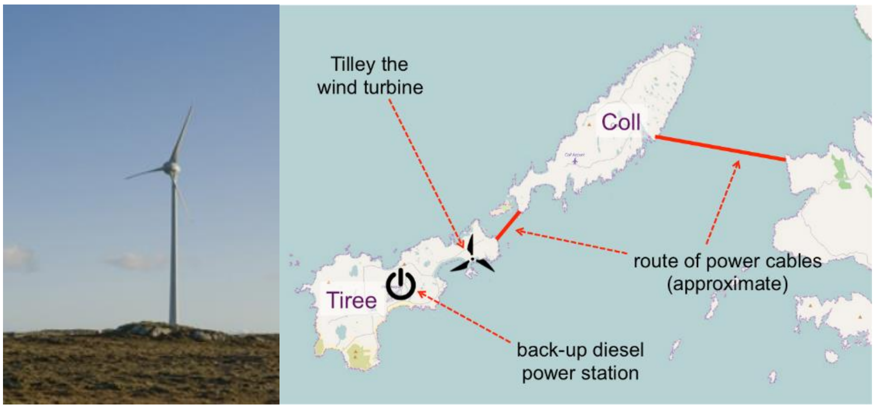

Tilley the wind turbine
Coll
route of power cables (approximate)
Tiree
back-up diesel power station

Fig. 10. (i) Tilley the Tiree wind turbine (ii) Power generation and distribution on Coll and Tiree (map OpenStreetMap, CC BY-SA)

cables and the island relies on the 3.5Mw diesel back-up power station. At these times Tilley has to be braked to 'ticking over speeds' as the largely manually controlled power station cannot cope with the potential rapid changes in supply as the wind speed changes.

The large proportion of Tilley's operation is operated remotely from Germany using an industry standard SCADA (Supervisory Control And Data Acquisition) interface. The turbine includes an SMS telemetry device from which operational data can be downloaded and to which commands can be sent. However, there is also a small internal control area within the wind turbine tower for use during on-site maintenance or when the mobile signal fails (a common hazard at a remote location).

The two main control panels are shown in Figures 11 and 12. It should be noted that this is by definition in an exposed location on the island, that the control area has no heating, and that the engineer may well have been climbing up further inside, or have come from the outside, potentially in a winter storm. All the controls must therefore be capable of being used with wet, cold and potentially gloved hands.

Fig. 11 is the 'digital' display and control panel. On the right it has status indicators (small LEDs) and on the left are numeric outputs for wind speed, rotation speed, power output, etc. Between these is an area with a numeric keypad and screen for entering more detailed parameters. Fig.12 is the 'physical' panel, with large buttons and knobs including an emergency stop button, which is particularly important to be able to operate with gloved hands. Although these are described as 'digital' and 'physical', of course the 'digital' panel has physical buttons albeit of the flat membrane type.

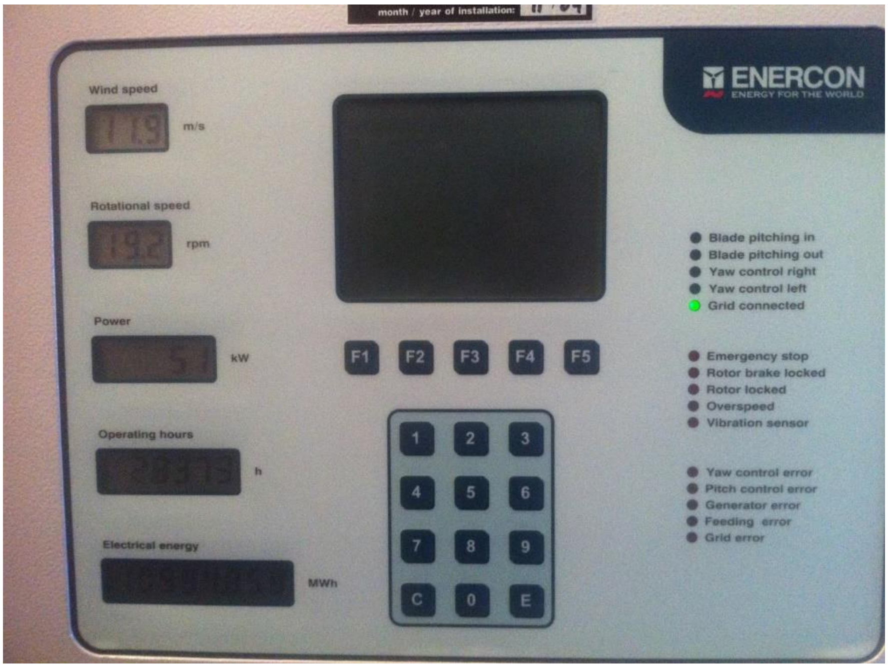


Fig. 11. Digital display and control panel in Tilley (photo © William Simm)

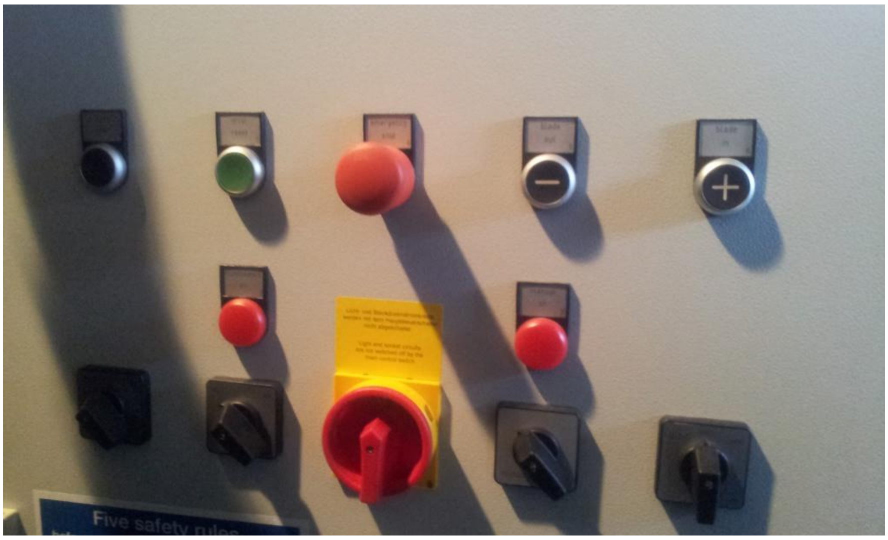


Fig. 12. Physical control panel in Tilley (photo © Maria Angela Ferrario)

Connection with Formal Methods

Formal methods can help to describe and analysis certain aspects of the user interfaces presented in Fig. 11 and 12. The buttons and display in the digital panel (Fig. 11) can be modelled using a range of dialogue-level techniques and standard properties verified such as the predictability of actions or reachability of the state (Dix et al., 2004). However, one of the key features of the turbine controls is their physical nature, in two senses. First, as with the previous two case studies, the majority of controls directly affect some physical process, such as the wind turbine angle of attack, electrical power. The second is that the physical properties of the buttons on the control panel in Fig. 12 is crucial to their operation, for example, the emergency stop button is large enough to be easily pressed with gloved hands, but has enough resistance so that it is not activated accidentally. Chapter 9 of this book gives one example of analyzing the latter using physiograms, another approach would be to use detailed force-displacement graphs (Zhao et al., 2014). Ideally, one would 'unplug' the Tilley control panel to experiment with the feel of physical buttons; or experiment with the live panel to get the 'feel' of the buttons, but Tilley is in constant use, so this was not possible for this case study. However, during the design of an interface such as this, detailed physical examination and specification would be possible.

Further work in formal description and analysis could focus on various other types of interactive systems and devices supporting maintenance work of wind turbines. For instance, mobile devices could be used to provide and to collect information in the field during the maintenance procedure, e.g. using the SMS-based communication interface of Tilley, or to implement a cooperative communication infrastructure between various technicians. Alternatively, augmented reality devices could be facilitated to offer installation and maintenance information while looking at the machines and user interfaces to be worked on. Thus, this case study addresses various challenges for formal methods in HCI. First, more than one mobile device can be involved as well as more than one user. Second, it has to be considered that asynchronous as well as synchronous communication patterns exist. Third, various aspects of information processing and presentation have to be considered: location, workflow, as well as unknown situations. Finally, the case studies address physical, technical interfaces which have to work in rough environment raising unusual requirements.

Case Study 5 - Interactive Systems in Public Areas – Interactive Public Displays

Nowadays, large interactive displays can be found in various types of public areas either in urban or rural spaces. Their main purpose is to provide information or to entertain people. More and more relevant are displays that are interactive or even persuasive. They react to a person's input, whether to provide personalized information or to change its content according to the context or the intended information provided. For these interactive displays, various interaction methods and concepts have been developed, such as gesture-based interaction, face detection or other possible input devices. This interaction with the public display could not only involve a single person but also multiple persons, such as shown in the example of the "Domain Mall Interactive Display"[2] or the "The Magic Carpet" (Paradiso et al. 1997). Another class of public displays is presented by community information displays focusing on the information rather than on the entertainment aspect of public displays.

Community information displays - The Internet enabled shop-open sign

A community public display system on the Isle of Tiree has been produced as part of the biannual Tiree Tech Wave, a series of technology/maker meetings. The system is a long-term 24/7 deployment. At several Tiree Tech Wave events, islanders have talked with participants about different aspects of island life. Communication has often emerged as an issue; while the archetypal view of rural life supposes everyone knows everyone else's business, in fact with a widely distributed population, it can be harder to tell people about events and news than in a city.

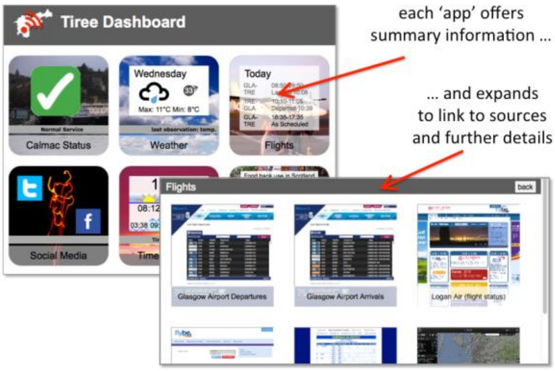
This has led to a number of community communication interventions including an SMS broadcast system for youth work, a web-based 'Dashboard' (Fig. 13) and a public tickertape display in an island café (Fig 14.).

The Dashboard and public display share a common data-oriented architecture (see Fig. 15). Raw data sources are gathered using a variety of techniques:

1. APIs and RSS feeds of public services (e.g. BBC local news)
2. scraping of web-based information
3. dedicated information entry interfaces (e.g. island events calendar)
4. sensor data

This raw data is then sampled in different ways to create parts of the data suitable for the different displays. This data is then passed to the final display device. In the case of the public display this is converted into plain text for the tickertape, in the case of the Dashboard, JSON formatted data to be passed into dedicated Dashboard 'apps'.

[2] https://www.youtube.com/watch?v=NRhDpDxTsLA



each 'app' offers summary information ...

... and expands to link to sources and further details

Fig. 13. TireeDashboard (http://tireetechwave.org/TireeDashboard)

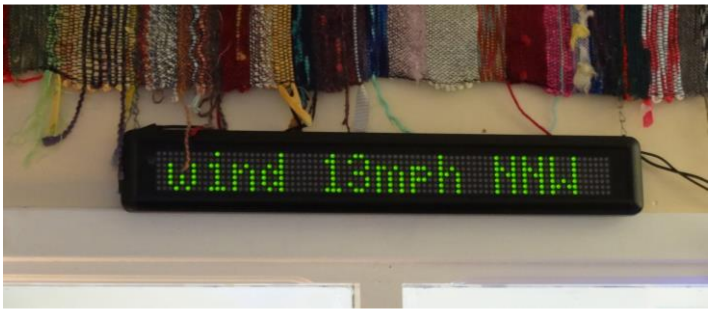


Fig. 14. LED public tickertape display.

Much of the information is from existing public sources (1 and 2), and some created explicitly (3), and the sensor data (4). Data extracted from web APIs and web page scrapping includes weather, tide times, BBC News headlines, local Twitter feeds and up-to-date travel status for plane and ferry. The last of these is particularly critical as the extreme island weather can often cause transport disruption. In addition, there are dedicated web forms for the café and island community trust to add messages about upcoming events to compliment web scrapes of the island 'what's on' page.

Many public display projects are in large cities or institutions such as university campuses or airports. In these contexts, it is possible to have staff whose job includes monitoring and keeping information current. In a small community public display it is crucial that there is a baseline of information that is current and valuable (hence regional news, weather, transport, etc.) so that displays are useful without additional effort. If displays, both web based and physical are actively used, then this makes it more worthwhile for community members to add useful information, and hence create a virtuous spiral.

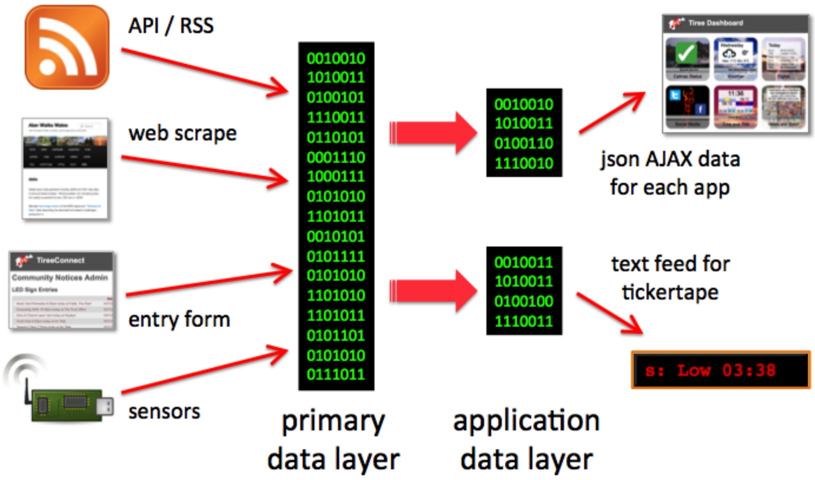


Fig. 15. Tiree data architecture

Live sensor data is used to adapt the information displayed. For example, the island data infrastructure includes an internet-enabled 'Open' sign (see Fig. 16), which is installed in the Cobbled Cow café' on the island. The sign constantly broadcasts its status to the web data infrastructure, and this is used to display whether the shop is open or closed.

However, in general, it cannot be assumed that all sensor data is always reliable. Fig. 17 shows an analysis of the tasks performed by the café owner during a typical day. Crucially tasks 4.1 and 6.2, when the sign is turned on and off, are performed as a matter of routine, but could be forgotten, it is physically possible to open the shop without turning on the sign. Similarly, it is possible to close the shop without turning off the sign, although this is likely to be noticed as the glow of the sign would be visible except in high summer. These issues are described in greater detail in Chapter 7.

The final part of the island public display infrastructure is a projected table-top display (Dix et al. 2016). So far this has not been included as part of the data architecture and instead used for stand-alone projects including the launch of Frasan (Fig. 18), the island mobile heritage app (Dix, 2013). The display consists of an LCD projector and Kinect-style depth camera; these are mounted in a custom-designed enclosure fixed to the ceiling (Fig 19). The enclosure rotates so that the display can be used on either the table top (also custom designed for the space) or the wall. The system was designed to enable multi-user, multi-touch interactions (Bellucci et al., 2011), but could also be used for other depth-based installations, such as shifting sand piles to drive geographic simulations (Kirn, 2011), and has already been used for BYOD (bring your own device) tangible interactions (Turchi et al., 2015).

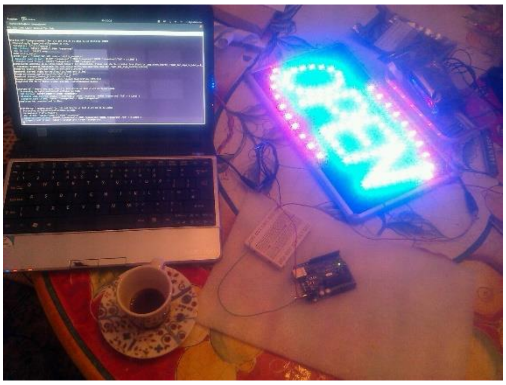


Fig. 16. (left) Internet-enabled open sign under development (photo Rory Gianni); (right) Electric Imp module (photo www.electricimp.com media resources)

0. Running cafe

1. drive to cafe
2. enter cafe (through side door)
3. prepare for opening:

   turns on power, lights, etc.
4. open up cafe (at opening time)

   4.1 turn on open sign

   ** sensed by system

   4.2 open café doors
5. serving customers

   take order, cook and serve food, wrap, take money
6. close up cafe (at closing time)

   6.1 close café doors

   6.2 turn off open sign

   ** sensed by system
7. tidy up
8. leave cafe (side door)
9. go home

Fig. 17. Task analysis Cobbled Cow café

Connection with Formal Methods

The community information display case study is unusual in that insights from formal analysis directly influenced the design of some elements; this is described in

Fig. 18. Table-top display in use at launch of Frasan mobile heritage app

detail in Chapter 7. There are also particular issues for formal modeling and analysis that are highlighted by this case study.

The island environment has many technical and social constraints that are different from those in city-based displays, formal descriptions of these could help designers who are not familiar with the context of use. The shop sign showed the importance of understanding these rich contexts and the pitfalls that may arise.

The complex environment also means that a broad range of types of display are included in the island deployment: the simple on-off open sign, the non interactive LED ticker tape with short text messages, the interactive touch table and web-based dashboard. This range is wider than typically considered by standard tools such as responsive designs, ranging from tangible to non-interactive, graphical to textual displays. Model-based formal methods could help understand the way information is mapped in different ways to these radical different devices. The range of data sources as well as display modalities suggests the potential for rich information models that take into account different timeliness, locality and dependability of information flows.

A variety of further challenges and research questions can be identified for the use of formal methods in context of public displays in general. First, the design and description of interaction methods for public displays raises various challenges. As the needed interaction methods are not WIMP based as classic input devices are not available, e.g. mouse and keyboard, methods are needed that range from gesture-based interaction to 3D user interfaces or speech-based interaction to only mention a few. Second, both the presented content as well as the used interaction methods could or should be adaptable to the individual user and the context of use. In this case, the user has to be assumed as completely unknown and the context as completely uncontrolled. Here, the question is how far formal methods could support the gathering process of information to generate user models, to describe such models as well as to support the validation of such systems in use. The validation could be of value during runtime as well as beforehand or after the usage phase. The same is true for the context, thus how far formal methods enable the description and handling of the dynamic context and the emerging conditions. As public displays influence public areas, it is relevant to investigate the newly created socio-technical system. This finally is an aspect less discussed in the context of formal methods research and offers a new perspective in research on formal methods in HCI. The main challenge and opportunity in this regard is the modeling and analysis of these systems using formal methods.

References

Airlines Electronic Engineering Committee (2013) ARINC 661 - Cockpit Display System Interfaces to User Systems. ARINC Specification 661-5

Barboni E, Conversy S, Navarre D, Palanque PA (2006) Model-Based Engineering of Widgets, User Applications and Servers Compliant with ARINC 661 Specification. In: Design Specification and Verification of Interactive Systems, Springer, pp 25-38

Bellucci, A., Malizia, A. and Aedo, I. (2011). TESIS: Turn Every Surface into an Interactive Surface. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces.

Bier EA, Stone MC, Pier K, Buxton W, DeRose TD (1993) Toolglass and magic lenses: the see-through interface. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, ACM, pp 73-80

Dix A, Finlay J, Abowd G, Beale R (2004) Dialogue Notations and Design. Chapter 16 in Human–Computer Interaction, 3rd ed. Prentice Hall, Englewood Cliffs. http://www.hcibook.com/e3/chapters/ch16

Dix A (2013). Mental Geography, Wonky Maps and a Long Way Ahead. GeoHCI, Work-shop on Geography and HCI, CHI 2013. http://alandix.com/academic/papers/GeoHCI2013/

Dix A, Malizia A, Turchi T, Gill S, Loudon G, Morris R, Chamberlain A, Bellucci A (2016) Rich Digital Collaborations in a Small Rural Community. Chapter 20 in Collaboration Meets Interactive Spaces, C. Anslow et al. (eds.), Springer. doi: 10.1007/978-3-319-45853-3_20

EASA (2014) CS-25. Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes

EUROCONTROL (2010) Arrival Manager. Implementation GUIDELINES and Lessons Learned. Edition 0.1

Forbrig P, Martinie C, Palanque PA, Winckler M, Fahssi R (2014) Rapid Task-Models Development Using Sub-models, Sub-routines and Generic Components. In Proceedings of 5th International Conference on Human-Centered Software Engineering, Springer, pp 144-163

Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH (2016) Siedewasserreaktor (SWR) (in german). http://www.grs.de/aktuelles/begriff-des-monats-siedewasserreaktor-swr. Accessed 27 Sep 2016

IBM (1989) Common User Access: Advanced Interface Design Guide. Document SC26-4582-0

Peter Kirn (2011). In Sand and Pixels, Playing with Worlds Virtual and Tangible; Built with Kinect. Create Digital Media, July 7, 2011. http://cdm.link/2011/07/in-sand-and-pixels-playing-with-worlds-virtual-and-tangible-built-with-kinect/

Martinie C, Barboni E, Navarre D, Palanque PA, Fahssi R, Poupart E, Cubero-Castan E (2014) Multi-models-based engineering of collaborative systems: application to collision avoidance operations for spacecraft. In proceedings of the 6th ACM SIGCHI Symposium on Engineering Interactive Systems, ACM, pp 85-94

Martinie C, Palanque PA, Winckler M (2011) Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. In: Proceedings of INTERACT, Springer, pp 589-609

Navarre D, Palanque PA, Basnyat S (2008) A Formal Approach for User Interaction Reconfiguration of Safety Critical Interactive Systems. In: Proceedings of 27th International Conference on Computer Safety, Reliability and Security, Springer, pp 373-386

O'Regan G (2014) Z formal specification language. In: Introduction to Software Quality, Springer, pp 311-325

Owre S, Rushby JM, Shankar N (1992) PVS: A prototype verification system. In: International Conference on Automated Deduction, Springer, pp 748-752

Paradiso J, Abler C, Hsiao KY, Reynolds M (1997) The magic carpet: physical sensing for immersive environments. In: CHI'97 Extended Abstracts on Human Factors in Computing Systems, ACM, pp 277-278

RTCA and EUROCAE (2012) DO-178C / ED-12C, Software Considerations in Airborne Systems and Equipment Certification

TREL (2016) Tilley, Our Turbine. Tiree Renewable Energy Limited. (accessed 24/4/2016) http://www.tireerenewableenergy.co.uk/index.php/tilley-our-turbine/

United States Nuclear Regulation Commission (2016) Boiling Water Reactors. http://www.nrc.gov/reactors/bwrs.html. Accessed 27 Sep 2016

Vogel D, Balakrishnan R (2004) Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In: Proceedings of the 17th annual ACM symposium on User interface software and technology, ACM, pp 137-146

Zhou W, Reisinger J, Peer A, Hirche S (2014) Interaction-Based Dynamic Measurement of Haptic Characteristics of Control Elements. In: Auvray M, Duriez C (eds) Haptics: Neuroscience, Devices, Modeling, and Applications: 9th International Conference, Eu-roHaptics 2014, Versailles, France, June 24-26, 2014, Proceedings, Part I, pp 177–184. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. http://dx.doi.org/10.1007/978-3-662-44193-0_23.