



HAL
open science

A Generic Software and Hardware Architecture for Hybrid Interactive Systems

Elodie Bouzekri, Alexandre Canny, Célia Martinie, Philippe Palanque

► **To cite this version:**

Elodie Bouzekri, Alexandre Canny, Célia Martinie, Philippe Palanque. A Generic Software and Hardware Architecture for Hybrid Interactive Systems. Workshop on Heterogeneous Models and Modeling Approaches for Engineering of Interactive Systems @ EICS 2018, Jun 2018, Paris, France. hal-02983159

HAL Id: hal-02983159

<https://hal.science/hal-02983159>

Submitted on 6 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Generic Software and Hardware Architecture for Hybrid Interactive Systems

Elodie Bouzekri, Alexandre Canny, Célia Martinie, Philippe Palanque

ICS-IRIT, University Paul Sabatier – Toulouse III

Toulouse, France

elodie.bouzekri@irit.fr, alexandre.canny@irit.fr, martinie@irit.fr, palanque@irit.fr

INTRODUCTION

Hybrid Interactive Systems integrate both hardware and software elements so their engineering requires bridging the continuous analog real world and the discrete digital world. Cyber Physical Systems (CPS) are a specific kind of Hybrid Interactive Systems found in domains such as smart housing, transportation or healthcare.

For example, in smart-houses, users may use a remote control or a smartphone to operate systems such as light bulbs, window shutters or air-conditioning. However, the user can directly perceive cooling by getting closer to the air-conditioning device itself. From the HCI point of view, this means that the design and development of such hybrid systems require blending information from the two elements:

- The physical part of the CPS, (e.g. light bulbs or window shutters) that is (possibly) perceivable and actionable by the users and belong to the analog real world.
- The command and control part of the CPS (e.g. a remote control or a smartphone), that allows the user to use the CPS, might integrate a hardware and software part and may have (part of) their behavior described in the discrete digital world.

A classical interactive application (e.g. a desktop application such as Microsoft Word) belongs to the digital world and is most of the time designed without careful consideration of the hardware part of the interactive system that will host and allow users to manipulate it. Indeed, usually the assumption designers and developers make is that standard input and output devices will be used, and thus do not address them while designing the interactive software. This is particularly visible on interactive applications architectures such as Seeheim [3] or Arch [4] where input and output devices are not even mentioned. However, other work, such as the ones on multi-device applications make explicit (in a platform model) the execution platforms and their input/output devices [6]. This is also required when interaction must be reconfigured according to input devices failures as in [7]. Due to the recent evolution of interaction techniques and technologies (e.g. the large distribution of touch devices) more recent interactive systems architectures such as MIODMIT [5]

make explicit the hardware parts (belonging to the input and output devices).

An architectural model similar to Seeheim but adapted to CPS would require making explicit all the components, their interrelations and what can be perceived and manipulated by the users. Indeed, in [1], Broy et al. describe challenges in engineering CPS and state that regarding to their implementation, “new ways must be found to manufacture products that are software driven but have physical components”.

The position paper presents a generic architecture for CPS that enables the separation into several components of both elements of CPSs. Such architecture supports the design and development of CPSs by decomposing a CPS into several, connected but loosely coupled components. To demonstrate its use and applicability, we describe two existing CPS with this architecture: the Philips Hue lighting systems [2] and the Auxiliary Power Unit of a large civil aircraft.

A GENERIC ARCHITECTURE FOR CPS

Figure 1 presents a generic architecture, mixing both physical systems and software components, illustrating how a CPS can be broken down into generic components in order to support their engineering. In this Figure, we highlighted three parts: i) the “Monitored and Controlled CPS”, ii) the “Transducing and Control” and iii) the “Command and Control System of the CPS”. Each of them is divided into one or more hardware or software components.

The following paragraphs describe each one of these components from left to right.

The Monitored and Controlled CPS component

The CPS Physical Device

The CPS physical device is the concrete physical system to be operated using a command and control user interface. A detailed description of its inner behavior is not always known and not always mandatory to allow operations. The inner state of such device is usually accessible by means of sensors while it can be manipulated using actuators.

The Physical User Interface

The physical user interface of a CPS represents the physical phenomenon that can be perceived by the operator. It is for instance smoke, vibrations, explosion... The operators may

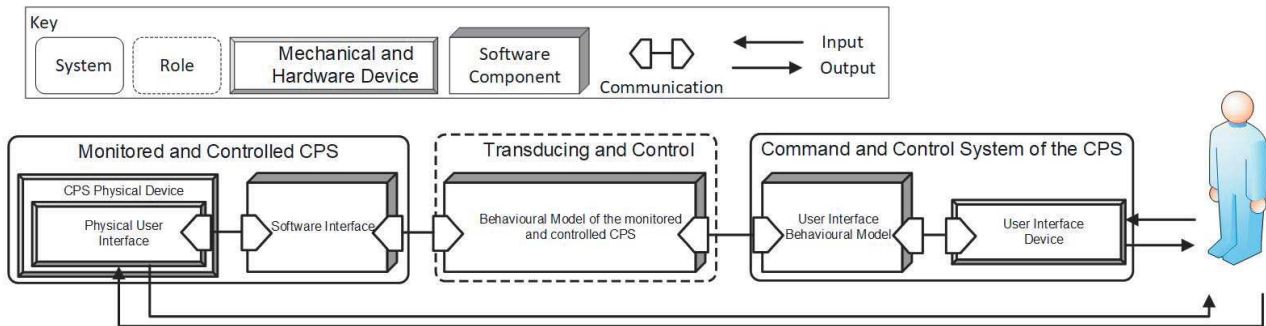


Figure 1. The generic architecture for supporting the engineering of cyber physical systems.

perceive this information (e.g. by touching the device) and might be able to interact with the device directly (e.g. pushing or moving it).

The Software Interface

The software interface component of the “Monitored and Controlled CPS” part provides a view on how to handle the CPS physical device and how to assess its inner state. This component provides a set of observable parameters that are the result of the discretization of potentially continuous physical phenomenon. For instance, it could provide an integer value (produced by sensors) corresponding to an abstraction of a real engine speed. The Software Interface thus provides both a set of observation mechanisms and a set of commands that may be performed on the CPS physical device (for instance starting or stopping the device).

The Transducing and Control component

The transducing and control component is composed of a unique component: the behavioral model of the monitored and controlled CPS. This component is responsible for translating both low-level information from the Software Interface into higher-level data (e.g. the availability of an electric service only available if the engine speed is above a given threshold) and higher level commands from the operators into low-level commands acceptable by the device. In other words, this component is responsible for digitizing the behavior of the monitored and controlled CPS (which is, per definition, analog).

It is important to note that this component has been defined, within our architecture, as a “role” and not as a system. Indeed, we have come across different types of CPSs where the functions and data of that component has been located within the monitored and controlled CPS system itself, within its command and control system or within a third “intermediate” system.

The Command and Control System of the CPS

The command and control system of the CPS corresponds to the elements of the system designed in order to allow users to interact with the CPS. It is important to note that for one CPS there may be several command and control systems (e.g. a TV controlled with a dedicated remote control and a cellphone).

The Behavioral Model of the User Interface

The behavioral model of the user interface deals with both the rendering (towards the user) of parameters provided by the behavioral model of the CPS triggering of functions within the CPS following user’s actions. This component may have some local interaction behavior that does not directly affect the device (for instance, some temporary states during operations before acting on the CPS). As this behavior may be complex, this component is usually broken down into smaller communicating sub-components.

The User Interface Device

The User Interface Device is a set of input and output devices dedicated to allow users to interact with the monitored and controlled CPS (for instance, in an aircraft it can be dedicated screens, keyboards, and physical buttons).

The proposed architecture enables the separation of a complex system into several components. These components are then easier to apprehend. Besides, these components are strongly consistent and loosely coupled. This allows the separation of concerns (for instance, the components related to the UIs are situated within the UI side) and the locality of modification (a modification within one of the components will not necessarily affect the other components).

ILLUSTRATIVE EXAMPLE WITH A CONSUMER GRADE PRODUCT

In this first illustrative example, we focus on a consumer grade-product, the Philips Hue smart light system illustrated in Figure 2. In this system, the user pairs a set of light bulbs to the Hue Bridge to be able to control their On/Off state as well as on their light color and brightness. The user operates these controls via a smartphone or a remote control. In this example, we consider that a light bulb is already paired with the Hue Bridge so the user can control it. Figure 3 presents the instantiation of the proposed architecture for the Philips Hue lighting system when a lightbulb is paired to the Hue Bridge.

The Monitored and Controlled CPS

Since this example focuses on controlling a light bulb in the smart light system, the monitored and controlled system here is the light bulb itself.

The CPS Physical Device

In the smart light system, the light bulb correspond to the CPS physical device (“Hue light bulb” in Figure 3).

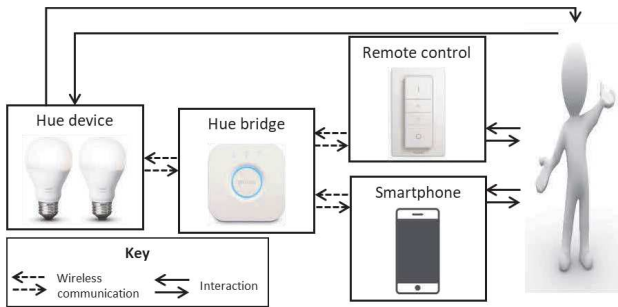


Figure 2. Overview of the Philips Hue system and its command and control systems.

The User Interface Device

In the smart light system, the physical user interface (“Hue PUI” in Figure 3) corresponds to both the light and heat produced by the light bulb that are directly perceivable by the user. Another possible interaction that may be described within this component is the fact that the user may decide to unplug the light bulb.

The Software Interface

This Software Interface (named “Hue SI” in Figure 3) describes the set of observation mechanisms allowing knowing if the light bulb is ON or OFF, its color, its intensity, It also describes the fact that the light bulb provides possibilities to manipulate its state: the possibility to turn it on, to turn it off, to change color or intensity.

The Transducing and Control

This component (“Hue BM” –for Behavioral Model– in Figure 3) represents the discrete behavior of the Hue light bulb. Concretely, the behavioral model both describes the different discrete states of the light bulb (such as "ON", "OFF", "Turning ON", "Turning OFF"....) and the different possibilities for state changes (e.g. how is it possible to switch from "ON" state to "OFF" state, going through "Turning ON" state). The Hue bridge integrates this behavioral model and is in charge of the role “transducing and control”.

The Command and Control System of the CPS

As the Philips Hue CPS provides two command and control systems (a remote control and a smartphone), Figure 3 presents those two elements respectively named “Remote control” and “Smartphone”. The following paragraphs describe the UIBM and UID for each of these command and control systems.

The Behavioral Model of the User Interface

Each command and control systems in the Philips Hue CPS, exhibits its own UIBM:

- **The remote control UIBM** (“Hue UIBM - Remote Control” –for Behavioral Model of the User Interface– in Figure 3) describes the behavior of the remote control. For instance, it describes the fact that a key of the remote control can have two states ("Pressed" or "Released").
- **The smartphone UIBM** (“Hue UIBM - Smartphone” –for Behavioral Model of the User Interface– in Figure 3) describes the behavior of the application that is running on the smartphone. For instance, this component describes the different modes of the application.

The User Interface Device

In the architecture for the Philips Hue CPS, the user interface devices are:

- **For the remote control** the concrete remote control device composed of the different keys of the remote control.
- **For the smartphone** the concrete device is the smartphone touchscreen.

This section has demonstrated that the various components of a CPS can be dealt with using the generic architecture. The user interface makes it explicit that the user can perceive directly the state of the device (lamp on or off when he/she is collocated with the device) represented by the direct arrow between the CPS box and the user icon. The user can also manipulate the device states using two different devices.

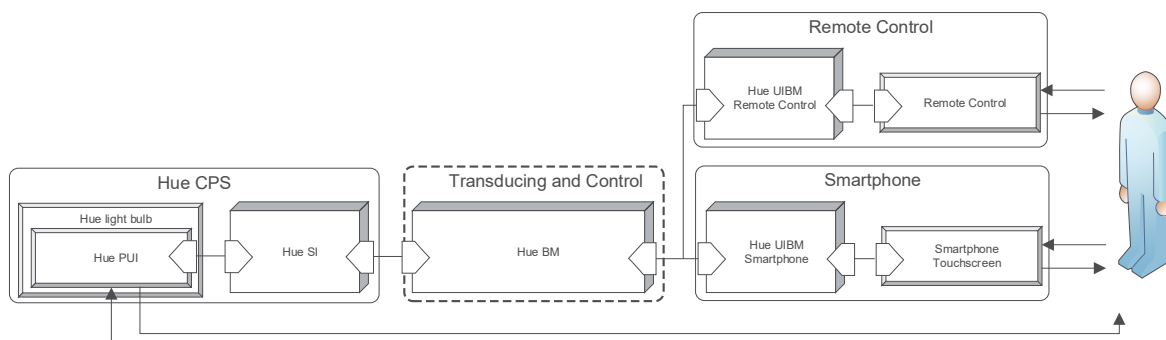


Figure 3. Application of the generic architecture to the command and control of the Philips Hue CPS (key is in Figure 1).

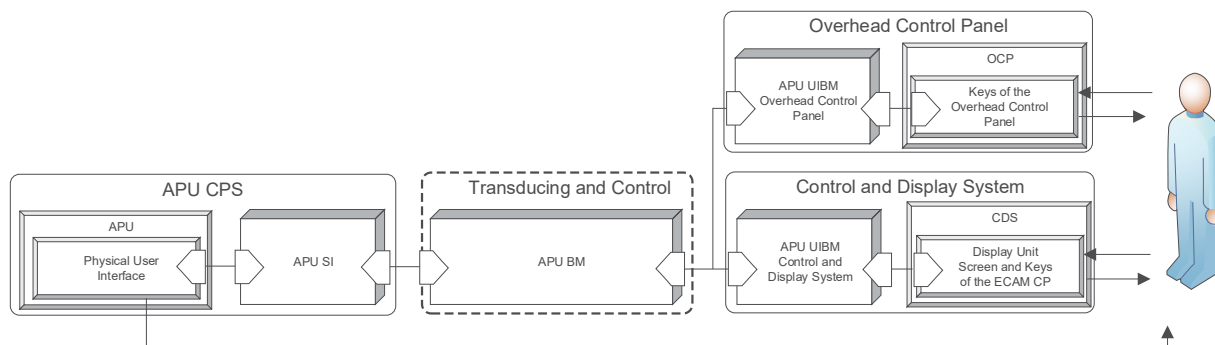


Figure 4. Application of the generic architecture to the command and control of the APU (key is in Figure 1).

ILLUSTRATIVE EXAMPLE WITH THE APU OF AN AIRCRAFT

The Auxiliary Power Unit (APU) of an aircraft is a small engine designed to provide electricity and bleed air to the aircraft on ground or mid-air. An overview of the APU command and control system is presented in Figure 5. To operate the APU from the cockpit, pilots may use controls from the Overhead Control Panel (containing several buttons, see Figure 5) or the Control and Display System (made of the ECAM Control Panel (for input and a dedicated Display Unit for output). Pilots may also perceive the behavior of the APU through the noise and vibrations the turbine produces (especially on ground).

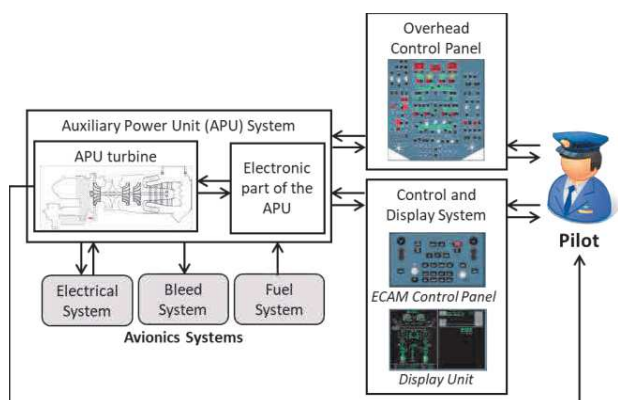


Figure 5. Overview of the cockpit command and control systems of the Auxiliary Power Unit

Figure 4 presents the instantiation of the proposed architecture for the command and control system of the APU. In the same way as the Philips Hue, the generic architecture can be tuned to represent the components of this complex CPS.

CONCLUSION

Manufacturing products that are software driven but have physical components is one of the challenges in engineering Cyber-Physical System. To help overcome this challenge, we proposed an architecture in which we highlighted the components that seems relevant when designing both the controlled part and the command and control part of the CPS. We showed that a transducing component, located within or outside the monitored CPS, allows the discretization of the monitored system behavior through a

Behavioral Model (BM). This BM is use to feed data to the command and control systems of the CPS. Each of these command and control systems use this BM to build its own behavioral model of the monitored system so user can easily take advantage of the available Input/Output. To illustrate the applicability of the proposed architecture, an instantiation on it on the Philips Hue lighting system and on the APU of an aircraft is provided.

Due to space constraints, we have not presented how the behavior of each component can be described and how to ensure that this behavior exhibits the expected behavior in terms of usability and user experience.

REFERENCES

1. Broy M. and Schmidt A. 2014. Challenges in Engineering Cyber-Physical Systems. *Computer* 47, 2: 70–72.
2. Philips, Hue lamp. <http://www2.meethue.com/en-us/products/>. (????). Last accessed in May 2018.
3. Pfaff G.E. (ed.): Proceedings of IFIP/EG Workshop on User Interface Management Systems (November 1983, Seeheim, FRG). Springer-Verlag, Berlin.,1985
4. Bass, L., Little, R., Pellegrino, R., Reed, S., Seacord, R., Sheppard, S., & Szezur, M. R. (1991). The arch model: Seeheim revisited. In *User Interface Developers' Workshop*.
5. Cronel M. Une approche pour l'ingénierie des systèmes interactifs critiques multimodaux et multi-utilisateurs: Application à la prochaine génération de cockpit d'aéronefs. Thèse de doctorat. Université de Toulouse, Université Toulouse III-Paul Sabatier. 2017
6. Lo K. et al. i* Chameleon: a platform for developing multimodal application with comprehensive development cycle. Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM. 2013.
7. Navarre, D., Palanque, P., Basnyat, S., Usability Service Continuation through Reconfiguration of Input and Output Devices in Safety Critical Interactive Systems, Proc. SAFECOMP 2008, , LNCS 5219, 373-386