



HAL
open science

Specific Task Modeling for Cyber Physical Systems

Elodie Bouzekri, Alexandre Canny, Célia Martinie, Philippe Palanque

► **To cite this version:**

Elodie Bouzekri, Alexandre Canny, Célia Martinie, Philippe Palanque. Specific Task Modeling for Cyber Physical Systems. Workshop on Heterogeneous Models and Modeling Approaches for Engineering of Interactive Systems, in conjunction with the 10th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2018), Jun 2018, Paris, France. hal-02983160

HAL Id: hal-02983160

<https://hal.science/hal-02983160>

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.

Specific Task Modeling for Cyber Physical 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

ABSTRACT

User interfaces for the command and control of systems usually integrate several types of interaction elements: physical space, hardware and software. Within such kind of environment, operators have to complete their tasks manipulating hardware and software elements that are distributed in the physical space. However, current tasks description notations do not take into account physical and hardware aspects beyond manipulation of input devices such as mouse and keyboard. This paper proposes a set of extensions to task modeling techniques in order to represent user goals and activities when interacting with distributed hardware and software user interfaces. These extensions are demonstrated with the HAMSTERS-XL notation and tool. These models can be analyzed and simulated to support identification of cumbersome tasks, motoric and perceptive efforts and sub-optimal UI distributions. We present how such information can be used for improving the user interface of the hardware and software distributed user interfaces. We exemplify the approach on a simple but realistic example: The Philips Hue light bulb system.

Author Keywords

Interactive Systems; Cyber Physical Systems; Model-based Design; Task Modeling.

INTRODUCTION

User interfaces for the command and control of interactive systems, such as power plant workstations or aircraft cockpits, are made up of hybrid and distributed interaction elements including hardware and software components. Within this kind of environments, operations take place in the physical workplace and operators have to complete their tasks manipulating distributed hardware and software elements. During these activities, the performance of the users is impacted by the positions (within the physical working environment) of the hardware and software elements composing the command and control interactive system. We argue that representing hardware and software

elements explicitly and systematically in task models contribute to the design, the assessment and the development of usable, reliable and safe distributed interactive systems. We propose a set of extensions to task modeling techniques in order to represent user goals and activities when interacting with distributed hardware and software user interfaces. We present the benefits of using such extensions as, for example the automation of analysis of the task models, and the predictive assessment of the duration and complexity of a task.

MODELING USERS TASKS WITH CPS

CPS are intended to be used by humans [3], via direct interaction [6] and/or using a monitoring and control user interface [9]. This section presents user tasks when using a CPS and a way to describe these tasks by modeling them with HAMSTERS notation and tool.

Taking into account user activities for the command and control of a CPS as well as the way they may interact with the CPS (directly or indirectly) requires precise and complete description of their activities. Furthermore, in order to be able to analyze potential user errors that may happen at runtime, the following set of requirements has to be match:

- Identify and describe, in a complete and unambiguous way, the user tasks with the CPS, whether they interact in a direct way or with the Command and Control of the CPS.
- Identify and describe the Data (information, objects, knowledge) manipulated by a user for accomplishing her/his tasks.

Task modeling permits to describe user tasks with interactive systems. HAMSTERS-XL is an extension of HAMSTERS [5] notation that proposes to go beyond the description of simple WIMP interaction. HAMSTERS-XL enables to add customized tasks. This extension permits to adapt the notation to tasks specific to a device, a context or a domain.

In term of input and output devices, interacting with WIMP interactive systems usually requires no more than perceiving what is rendered on screen and to be aware of the presence of a keyboard and a pointing device. For this reason, the standard HAMSTERS notation proposes a generic “Perceptive Task” that mainly corresponds to the visual perception of information on a screen. However, while describing perception when using a CPS, precise description of the sight sense is required. Thus, we propose to extend

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:


- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.

Each submission will be assigned a DOI string to be included here.

perceptive task and to provide support to refine it to the sight task (see Table 1).

Table 1. Perceptive task integrated in HAMSTERS-XL

Perceptive task		Example of use
Icon	Name	
	Sight	Gathering data from outside of the aircraft (location in the airport, clouds, etc.), locating controls, assessing aircraft status from the screens and lamp in the cockpit, etc.

In the same way, while describing motoric actions when using a CPS, precise description of the several actions is required. We propose to extend motoric tasks and to provide support to refine it to the tasks described in Table 2 and Table 3.

Table 2. Motoric limb tasks integrated in HAMSTERS-XL





Motoric task			Usage
Icon (left/right)	Limb	Action	
	Finger	Press	Press button
	Hand	Grasp	Grasp joystick
	Arm	Fold	Reach interaction devices
	Arm	Stretch	Reach interaction devices

Table 3. Motoric head tasks introduced in HAMSTERS-XL

Head		
	Turn Left	Turn Right

ILLUSTRATIVE EXAMPLE OF THE HUE CPS

The Philips Hue smart lamp [8], which is considered one of the typical IoT devices, and the most popular smart light system (see Figure 1). While the command and control is located on the remote control and/or on the mobile App, the physical device (the light bulb) is producing both light and heat that are both perceivable by the users. When using the remote control (which is only an input device and has no display on it), the user has to target the connection bridge (not the light bulb) and the only way to know the status of the device is to look at its hardware (physical) part.

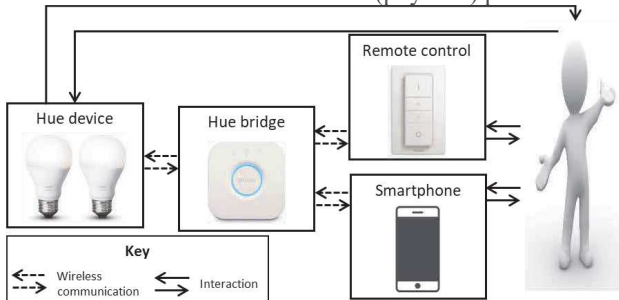


Figure 1. Overview of the Philips Hue system and its command and control systems.

This section presents the tasks that have been identified to be performed by the user with the physical UI and with the command and control UI. In this example, we focus on the user tasks to switch off the light bulb. These tasks may be performed in several contexts and with different devices. The following sections presents these user tasks required to reach the goal: “Switch off light in a room”. To reach this goal, several alternatives of tasks are described (under the choice “[]” operator). Figure 2 presents these alternatives.

Figure 3, Figure 4 and Figure 5 present the descriptions of the tasks that the user has to carry on to switch off light in a room. Figure 3 and Figure 4 respectively described this goal with HAMSTERS notation and HAMSTERS-XL notation. Figure 5 presents “Switch off light in a room with the Philips Hue smartphone app” goal with HAMSTERS-XL notation.

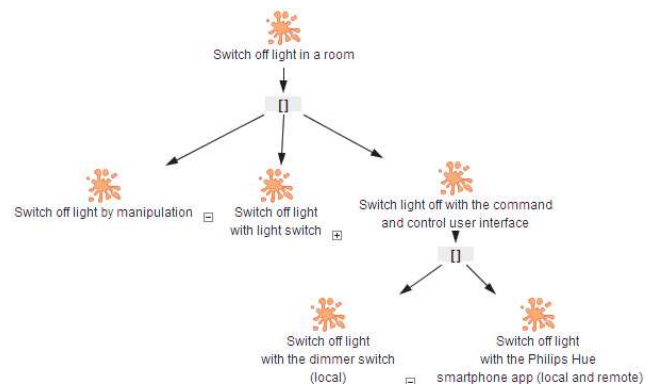


Figure 2. Description of user alternatives of tasks to "Switch off light in a room" with HAMSTERS notation.

Modeling User Tasks with the Physical UI with HAMSTERS.

Although the main aim of the Philips Hue light bulb is to enable end user programming of the home lighting, it is still possible to control it as a legacy home lighting system. A manual switch installed on the wall aims at providing the user a way to close or open the electrical circuit composed of the switch, the wires and the light bulb. In the task model depicted in Figure 3, these tasks are described in the branch “Switch off light with light switch”. The user has to perform the following sequence of tasks (tasks described under the sequence “>>” operator):

- “Perceive the light switch” (perceive task) that requires to access to “Light switch device” (input/output device),
- “Recognize light switch” (cognitive task),
- “Walk to light the switch” (user task),
- “Push off the light switch” (motoric task) that requires to access to “Light switch device” (input/output device),
- “Switch off light bulb” (interactive input task) that requires to access to “Light switch device” (input/output device),

- “Light off” (interactive output task) that requires to store OFF state in “Light bulb lamp” (output device),
- “Perceive light is off” (perceptive task) that requires to access to “Light bulb lamp” (output device), and then
- “Deduce light is off” (cognitive task).

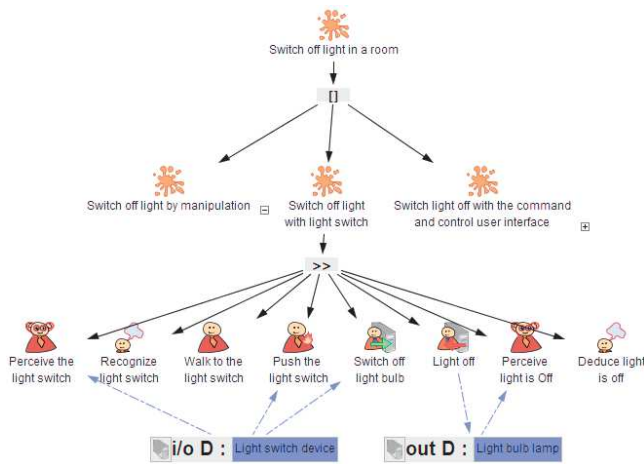


Figure 3. Description of the user tasks to “Switch light off in a room” with HAMSTERS notation.

Modeling User Tasks with the Physical UI with HAMSTERS-XL.

In the task model depicted in Figure 4, “Perceive the light switch”, “Switch off the light bulb” and “Perceive light is off” tasks of Figure 3 are refined to describe better user activity to reach “Switch off light of the room” goal:

- “Perceive the light switch” is refined with “Locate light switch” (iterative user task) disabled by “Recognize light switch” (cognitive task). “Locate light switch” is refined in a sequence (tasks described under the sequence “>>>” operator) of three tasks:
 - “Turn head right” (optional head right task)
 - “Turn head left” (optional head left), and
 - “Search light switch” (sight task) that requires to access to “Light switch device” (input/output device).
- “Push off the light switch” is refined into a sequence of two tasks:
 - “Stretch arm to the light switch” (Arm right stretch task)
 - “Press light switch” (Hand right finger press task) that requires to access to “Light switch device” (input/output device)

- “Perceive light is off” (perceptive task) is detailed in “See light” (sight task) that requires to access to “Light bulb lamp” (output device).

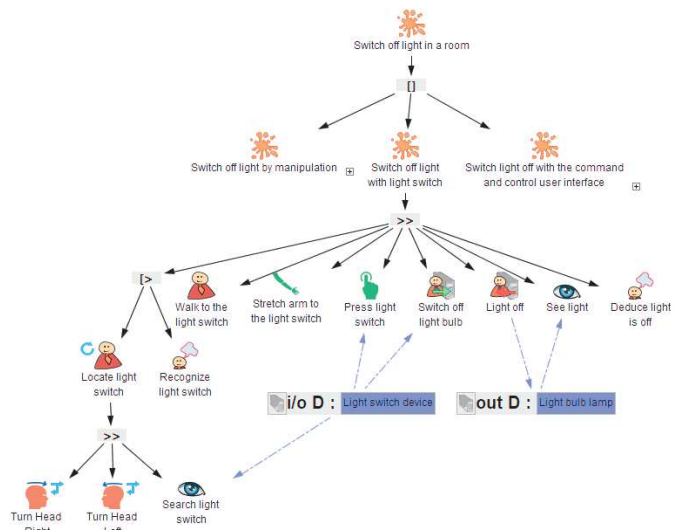


Figure 4. Description of the user tasks to “Switch light off in a room” with HAMSTERS-XL notation.

Modeling User Tasks with Command and Control UI with HAMSTERS-XL

To switch off the light bulb, the user has the possibility to use one of two available command and control interfaces: a dimmer switch device or the Philips hue smartphone application. The Figure 5 describes the user tasks to switch off the light bulb with the Philips hue smartphone app. The user has to perform the following sequence of tasks:

- The user has to “Remember smartphone location” (cognitive task) or (“[]” operator) to “Locate the smartphone” (user task) until (“[>” operator) “Recognize the smartphone” (cognitive task)
- Next, to “Stretch arm to the smartphone” (Stretch arm task),
- To “Grasp the smartphone” (Grasp task),
- To “Interact with the smartphone to open light switch window” (interactive abstract task) using “Philips Hue app” (software application) and
- To “Switch off light bulb” (interactive input task) using “Philips Hue app” (software application).
- Then, then “Smartphone” (input/output device) “Send event to the bulb” (system task) to switch off the light thanks to the “Hue Bridge” (output device).
- The “Light bulb lamp” (output device) switches off: “Switch off light” (interactive output task).
- The user can “See light is off” (sight task) only if the “User location” (information) is equal to the “Room” (“Test” arc between the sight task and the information);
- Then the user has to “Deduce light is off” (cognitive task) to reach the goal “Switch off the light in a room”.

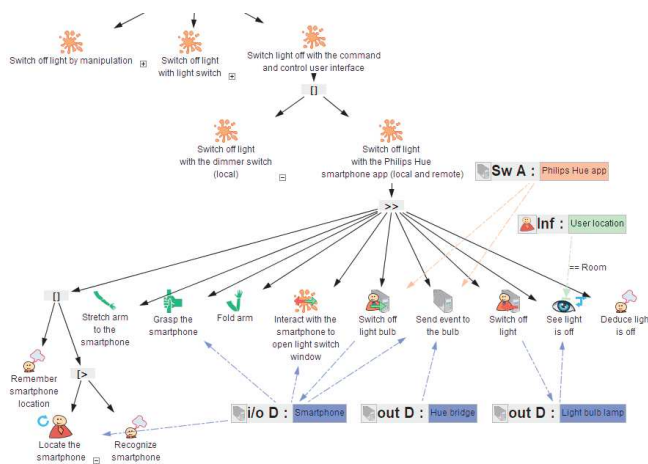


Figure 5. Description of the user tasks to “Switch light off with the Philips Hue smartphone app” with HAMSTERS-XL notation

MODELS ANALYSIS

This section analyses the tasks models of the previous section and presents the benefits of a specific task modeling notation like HAMSTERS-XL compared to a standard task modeling notation like HAMSTERS [5].

Most of the task modeling notations, such as HTA [1] and CTT [7], enables to describe user tasks and goals, as well as manipulated objects. Task models built with such notations can embed additional textual information to refine the information about the types of tasks and the types of data required to accomplish the tasks. Although this is sufficient to understand and identify main user tasks and goals, a specific task modeling notation like HAMSTERS-XL improves the description power for the motoric and perceptive tasks of these standard task modelling notations. For example, move a cursor thanks to a mouse task and press a button task are described with the standard task modeling notation with the same task: a motoric task. The specific task modeling notation permits to categorize these motoric and perceptive tasks. In consequence, the analysis of the task model is no longer text parsing and can be more easily automatize. Indeed, it is possible to measure the number of occurrence and performance time of a type of motoric or perceptive task (grasp task for example) or of a particular couple of motoric or perceptive task with a particular object (grasp the smartphone task for example) necessary to reach a goal directly from the task model. For example, contrary to a standard task modeling notation, it is possible to measure the performance time of “stretch arm to the smartphone” task independently of the “Grasp the smartphone” task of the Figure 5. The relative execution time of the stretch task is after the user recognizes the smartphone or remembers the smartphone location to when s/he begins to fold his arm to bring the smartphone closer to her/him. It is also possible to associate a minimum and a maximum execution time to “Stretch arm to the smartphone” task to estimate an absolute execution time. In addition, the specific task modeling

notation permits to detect automatically impossible tasks like two concurrent left hands tasks: grasp and release left hand tasks for example. Moreover, it is also possible to detect impossible tasks due to human limitation [1]. For example, to perform two non-automatic tasks concurrently.

Furthermore, the specific task modeling notation permits other task model analysis when coupled with a 3D space model of the user context. Indeed, it is possible to determine occlusion with objects position. For example, visual occlusion due to the user position can be determined. In addition, the specific task modeling notation permits to measure the quantity of movement and the effort cost of a motoric or perceptive task associated with a particular object thanks to the source point position and the destination point position. In addition, specific task modeling notation is useful for training. Training program for complex systems must cover all the tasks that the user has to carry out [4]. Determining complete and unambiguous user tasks to reach a goal, occlusion, workload and effort cost is necessary to establish training program of complex systems.

PERSPECTIVES

Due to the chosen domain and context, the example explores a limited set of perception and motoric tasks (sight, stretch arm, press with a finger, grab, and turn head left and right). For example, speech, touch, smell, be turned away from the light bulb or from the switch are not represented in this paper. Future work must address other domains, contexts and devices to analyze deeper specific task modeling notation for CPS.

REFERENCES

1. John Annett, K. D. Duncan. 1967. Task analysis and training design. *Occupational psychology*, 41, 211-221.
2. Sylvain Charron and Etienne Koechlin. 2010. Divided Representation of Concurrent Goals in the Human Frontal Lobes. *Science* 328, 5976: 360–363. <https://doi.org/10.1126/science.1183614>
3. Cyber-Physical Systems Virtual Organization Home page. <http://cps-vo.org/>. Last accessed in May 2018.
4. Célia Martinie, Philippe Palanque, David Navarre, Marco Winckler, and Erwann Poupart. 2011. Model-based training: an approach supporting operability of critical interactive systems. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11)*. ACM, New York, NY, USA, 53-62.
5. C. Martinie, P. Palanque, M. Winckler. “Structuring and Composition Mechanisms to Address Scalability Issues in Task Models”. *IFIP TC13 Human Computer Interaction 2011 (INTERACT)*, pp. 134-152. Springer-Verlag.
6. David Navarre, Philippe Palanque, Jean-Francois Ladry, and Eric Barboni. 2009. ICOs: A Model-based User Interface Description Technique Dedicated to Interactive Systems Addressing Usability, Reliability and Scalability. *ACM Trans. Comput.-Hum. Interact.* 16, 4, Article 18 (Nov. 2009).
7. Fabio Paterno, Carmen Santoro, Lucio Davide Spano. 2012. Concur Tast Trees (CTT), W3C Working Group Submission, www.w3.org/2012/02/ctt/, last accessed March 2018.
8. Philips, Hue lamp. <http://www2.meethue.com/en-us/products/>.
9. Ronny Seiger, Christine Keller, Florian Niebling, and Thomas Schlegel. 2015. Modelling complex and flexible processes for smart cyber-physical environments. *Journal of Computational Science* 10 (2015), 137 – 148.