Engineering Human-in-the-Loop Interactions in Cyber-Physical Systems

Miriam Gil[a,*], Manoli Albert[a], Joan Fons[a], Vicente Pelechano[a]

[a] *Centro de Investigación en Métodos de Producción de Software, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain*

Abstract

Context: Cyber-Physical Systems (CPSs) are gradually and widely introducing autonomous capabilities into everything. However, human participation is required to accomplish tasks that are better performed with humans (often called human-in-the-loop). In this way, human-in-the-loop solutions have the potential to handle complex tasks in unstructured environments, by combining the cognitive skills of humans with autonomous systems behaviors.

Objective: The objective of this paper is to provide appropriate techniques and methods to help designers analyze and design human-in-the-loop solutions. These solutions require interactions that engage the human, provide natural and understandable collaboration, and avoid disturbing the human in order to improve human experience.

Method: We have analyzed several works that identified different requirements and critical factors that are relevant to the design of human-in-the-loop solutions. Based on these works, we have defined a set of design principles that are used to build our proposal. Fast-prototyping techniques have been applied to simulate the designed human-in-the-loop solutions and validate them.

Results: We have identified the technological challenges of designing human-in-the-loop CPSs and have provided a method that helps designers to identify and specify how the human and the system should work together, focusing on the control strategies and interactions required.

Conclusions: The use of our approach facilitates the design of human-in-the-loop solutions. Our method is practical at earlier stages of the software life cycle since it allows domain experts to focus on the problem and not on the solution.

Keywords: Human-in-the-Loop design, Human-System interactions, Autonomous Cyber-Physical Systems, User attention

1. Introduction

Cyber-Physical Systems (CPSs) can be defined as the integration of computation, networking, and physical processes. A CPS is a kind of feedback system that integrates the dynamics of physical processes with software and networking, providing abstractions and modelling, design, and analysis

[*] Corresponding author

Email addresses: `mgil@pros.upv.es` (Miriam Gil), `malbert@pros.upv.es` (Manoli Albert), `jjfons@pros.upv.es` (Joan Fons), `pele@pros.upv.es` (Vicente Pelechano)

techniques for the integrated whole [1]. Applications of CPSs include automotive systems, manufacturing, medical devices, power generation, robots, and many other systems. Today's CPSs should be capable of autonomously adapting themselves at run time to new environmental conditions, unpredictable situations, changing user needs, new types of physical devices, new technologies and new services.

To this end, CPSs need to introduce autonomous capabilities [2, 3]. Different application domains and industrial sectors (such as autonomous cars, robots, drones, Industry 4.0) face the challenge of achieving full autonomy. Nevertheless, the diversity of systems, domains, environments, context situations, and social and legal constraints all point to a world where this full autonomy is a utopia within the short or medium term [4, 5]. This scenario has led software engineers to face the challenge of developing CPSs that support the concept of autonomy levels [2]. Each autonomy level involves different actions that should be allocated to humans to assist the CPS (while low autonomy levels imply that the human has more control over the CPS, high levels imply that the human only performs supervision tasks and a few actions). Due to this context, systems will require human support to guarantee their complete and correct behavior in all situations (human-in-the-loop) [1].

CPSs are typically designed as a network of interacting elements with physical input and output instead of as standalone devices and are to some extent autonomous systems, usually having feedback loops. In these systems, in most cases, the human plays a very important role as either an operator or supervisor, depending on the specific application. The human-in-the-loop CPS (HiLCPS) introduces the human into the feedback loop so that s/he can participate in the different phases: monitoring (helping the system to identify situations in the context), analyzing (collaborating with the system in the taking of decisions), planning (helping the system to plan a set of changing actions), and executing (handling, directing, or executing tasks). These solutions are complex to develop because integrating systems with humans is not a trivial task [1]; humans must be able to cooperate with systems efficiently and intuitively. At all times, the human-system integration must guarantee correct autonomous operation (from an operational point of view, quality of service, reliability, efficiency, etc.) [6], and in turn must yield, under certain conditions, full or partial control to the human to carry out some tasks [7]. The correct system operation depends on two design aspects [3]: 1) the *interface* between the human and the autonomous components, and 2) the *control* strategies for these human-in-the-loop systems. In this paper, we focus on both aspects.

This work aims at providing an approach to analyze and design HiLCPS solutions. The approach contributes with a method for defining human collaboration in the early stages of the development life cycle by both domain experts and interaction designers. To do so, the method provides the abstractions for describing how the human is integrated into a CPS in order to collaborate with it to achieve the system's goals. The proposal considers the role and the interface with human(s) in the loop together with the control strategies to be central aspects. In order to build the proposal, we first identify the human-related challenges in designing HiLCPS solutions, and we build our proposal based on them. To determine these challenges, we review the literature and extract the most relevant quality properties that have been identified to provide a good user experience. From this review, we focus on designing interactions that:

1. ensure human participation even in those situations where their attentional resources are limited,
2. provide a natural and understandable collaboration, and
3. avoid disturbing the human.

Therefore, our goal is to design robust, intuitive, and unobtrusive interactions to increase the usability of CPSs. Taking this into account, an important element of our approach is the consideration of human attention as a critical factor for user participation in order to acquire human attention to involve the human in the system and, at the same time, avoid overwhelming users and allow "natural" involvement. Thus, we propose a solution to manage human attention based on the attentional resources needed to perform the interaction. Starting from an abstract specification of the required attentional resources, we drive the selection of the different interacting elements (with physical input/output) that are available in the CPS to perform the human involvement. Moreover, human factors such as attention, stress, or fatigue, which might affect interactions and thus successful task performance, are considered to be key elements of our approach. The design of the human integration uses a model that captures human factors to improve the likelihood of achieving the task's goal and also to provide a better user experience.

The method also proposes applying fast-prototyping techniques to simulate the HiLCPS solutions by means of rapid prototypes. This allows testing human-in-the-loop solutions without actually implementing the supporting system. This method is fast to apply and reproduces a level of user experience that is considered to be very close to what users expect from a final system. The use of a rapid prototype also allows designers to gather relevant feedback in terms of user performance and experience, which is essential when reconsidering the HiLCPS designs prior to the development of the final system.

The paper is structured as follows. Section 2 reports on background related to human-computer interaction in CPSs. Section 3 introduces a running example. Section 4 identifies the research questions and goals of the approach and presents the proposal overview. Section 5 presents the design principles on which our proposal is based. Section 6 defines the approach for designing the human collaboration within CPSs. Section 7 presents the evaluation method, which uses fast-prototyping techniques to validate that the obtained prototypes fulfill the design principles. Section 8 presents an experiment performed to validate the usefulness of the design method proposed in this paper. Finally, Section 9 presents the conclusions and future work.

2. Related work

This section first reviews some works from the Human-Automation Interaction field that are related to the human-in-the-loop problem since our approach has an important component of human-computer interaction. Then, it focuses on previous works on how to integrate humans into CPSs. Finally, it provides a synthesis of the reviewed works in comparison with our work.

2.1. Human-in-the-loop in Human-Computer Interaction

In a recent review of the history of human-automation interaction research [8], the authors highlight three research themes that have been covered in the past but that continue to be important areas for research. These themes are: 1) function and task allocation between humans and machines; 2) trust, incorrect use, and confusion; and 3) attention management. These themes are aligned with the design challenges faced by our approach. We make use of these challenges to introduce some related works in the area. Regarding the function and task allocation between humans and machines, several works, such as [9, 10, 11, 12, 13], have discussed how to overcome the problem of suitably distributing functions or tasks between humans and automated systems. These works make use of functional models to describe the tasks that must be provided by the system or by the human, and they constitute a solid basis to be used when deciding how tasks should be allocated to humans or system in a way that maximizes the operator's ability to maintain control and handle unexpected events. In addition to these works, other ones that overcome task allocation are those works related to the HCI field that propose how to describe the logical activities that an interactive system should support to reach users' goals. Specifically, task models are widely used to define interaction with humans [14]. Some of these models are GOMS [15], Hierarchical Task Analysis (HTA) [16], and ConcurTaskTrees (CTT)[17]. These works show the growing usage of task modeling and its remarkable results and possibilities for modeling user interaction with the system. Our approach is based on the foundations of functional models and task models for describing user and system tasks, but it also introduces new aspects to be modeled when designing the shared control flow of collaborative tasks. These aspects focused on engaging the human, providing natural and understandable collaboration, and avoiding disturbing the human in order to improve the human experience. These additional aspects are: 1) the attentional resources required to perform a collaborative task, and 2) the human factors that can impact the success of human participation. With regard to the first aspect, the attentional resources specify the attentional demand imposed on the human by a task. They are used to drive the selection of the interaction mechanisms. For example, high attentional demand requires interaction mechanisms that are more obtrusive or that draw more attention. With regard to the second aspect, humans are influenced by critical factors, e.g. fatigue, stress, motivation, etc. These factors are very important for human performance. Thus, our approach uses a model that captures these human factors to improve the likelihood of achieving the task's goal while providing a better user experience.

With regard to trust in automation, according to Stumpf et al. [18], making the system's decisions and behavior understandable by end users is a first step in successful interactions. This theme has been investigated over the years [19, 20]. To this end, Lim and Dey [21] evaluated intelligibility features of context-aware applications by investigating how many explanations the users viewed and how that affected their understanding. They provided recommendations for designing intelligibility to promote the effective use of systems. In our work, we do not focus on system explanations, instead we incorporate these recommendations to provide interactions with feedback to users. In order to offer intelligibility, several works [22, 23, 18] identify quality properties, such as *observability*, *obtrusiveness*, or *feedback*, which are relevant to design user interactions for intelligent and autonomous systems. In [24], quality properties that are scattered throughout the literature are integrated into a design space for engineering adaptive user interfaces. Using the knowledge provided in these works, we have considered these quality properties as the basis to design our approach and achieve unobtrusive and understandable HiLCPS.

Finally, creating optimal workload levels for the human interacting with automation to avoid underload and overload is the third theme that has garnered persistent attention (we refer to this theme as attention management). Miller and Parasuraman [25] propose a method to support human-machine delegation interactions taking into account the human workload of some automation levels. However, the delegation is performed manually by operators once the system is deployed, enabling operators to behave more like a supervisor. Conversely, in our approach, the delegation is made by designers and tested with operators before development in order not to overwhelm the human at execution time. Our approach also takes into account other human factors to involve the human. In [26], the authors introduce a framework for designing mobile services that can be personalized in terms of obtrusiveness. This framework is adapted in our approach to achieve unobtrusive HiLCPSs. In the domain of autonomous cars, several works have performed different experiments trying to understand how humans respond to car signals and automotive user interfaces.

In [27] Rajaonah et al. examine the relationship between driver reliance on Adaptive Cruise Control (ACC) and variables such as trust, perceived workload, and perceived risk. The work concludes that drivers' behavior depends on their propensity to feeling overloaded and their perception of the risk taken by using ACC. Recently, in [28], the authors have investigated how susceptible human drivers are to auditory signals in different situations (stationary, driving, or autonomous driving), concluding that it is important to consider the limitations of the human brain in terms of susceptibility to auditory alerts. Looming sounds are analyzed in [29], where authors investigate and demonstrate that looming sounds can similarly benefit emergency braking in managing a vehicle with ACC. The authors suggest that it is a viable design principle to design auditory warnings to be consistent with the visual event that the warnings are intended to elicit a fast response to. All of these outcomes are complementary to our research and can be used in more advanced phases of the development life cycle where the system is deployed in a specific environment. In the case of the autonomous car domain, designers can use these outcomes to provide a suitable obtrusiveness and interaction design.

In addition to the above-mentioned themes, another key issue in the involvement of humans in CPSs is the consideration of the interaction context in the design of user interfaces. Since Thevenin et al. introduced the concept of plasticity in [30], several efforts have been made to build user interfaces with the capability of being aware of the context (context awareness) and to react to changes in this context [31, 32]. Similarly to these proposals, we use the user context to select the appropriate interaction mechanisms to allow human-system interactions. Specifically, we consider user attention as a context factor to be taken into account in order to achieve human engagement in the system.

A more philosophical approach is presented in [6], where Farooq et al. discuss the end of the era of human-computer interaction leading to the era of human-computer integration. To achieve this vision, advanced technologies for human-machine communication are required. Schirner et al. [33] propose augmenting human interaction in the physical world with transparent interfaces that use existing electrophysiological signals such as electroencephalography (EEG), electrocardiography (ECG), and electromyography (EMG). Moreover, they propose interface algorithms for human intent inference. In [34], Huang et al. presented the preliminary results of creating a brain mouse to command actions of a software system based on human intention. The goal was to incorporate the mental state of humans in systems so that they can "feel" and "anticipate" user intentions and put the human in the loop. Furthermore, Lloyd et al. [35] introduced the concept of a co-adaptive, human-computer interaction system where the system benefits from the human by recognizing his/her mental states. All interfaces of this kind can be used in our proposal as concrete interaction mechanisms, depending on the application domain.

2.2. Human-in-the-loop approaches in Cyber-Physical Systems

The problem of how to integrate humans into CPSs has been addressed from several perspectives in the literature of the last decade. In this section, we analyze some of the works that tackle this problem and that are close to our approach in someway. The goal of this section is to contextualize our approach in the literature of the Human-in-the-loop Cyber-Physical Systems of the Software Engineering domain.

In [36], Munir et al. discuss three major research challenges for involving humans in the control loop of CPSs. The authors highlighted the importance of determining how to incorporate human behavior models into the formal methodology of feedback control loops. This challenge is aligned with the focus of our work, since we propose a design technique that includes human factors as first-class elements to take into account when specifying human-system collaboration. More fine-grained challenges are highlighted in a recent survey on Human-in-the-Loop Applications in [37]. In the context of Internet of Things and CPSs, the survey defines taxonomic overviews of Human-in-the-loop CPSs and the human roles within them. Our proposal fits the "Direct Control" and "Closed loop" taxonomy views of current Human-in-the-loop Applications and agrees with the three taxonomic views of Human Roles. In [38], the authors propose the opportunity-willingness-capability (OWC) ontology for defining model elements that impact task performance. We use this ontology to identify the human factors on which collaborative work is conditioned.

The implications of integrating people in the loop of Cyber-Physical Systems are discussed in [7]. Sowe et al. focus on identifying what aspects of humans have to be modeled as new components of the CPS. They then propose the human service capability description (HSCD) model to represent them. In the same line, our work uses a model to represent the characteristics of humans that we consider to be relevant for integrating the human in the loop; however, we go beyond this issue and represent other aspects of HiLCPSs, such as the shared control flow between the human and the system. Also, related to this last aspect is a specific concern that arises when humans cooperate with autonomous systems, i.e., choosing between performing a manual or automated task. Camara et al. study this concern in [39]. The study is useful for identifying the various roles that operators can perform when cooperating with self-adaptive systems. The work extends the Rainbow framework, which is used to design self-adaptive systems, to include humans. This extension focuses on analyzing the trade-offs of involving humans in adaptation. This framework could be used to complement our approach.

A framework for synthesizing a semi-autonomous controller from high-level temporal specifications that expect occasional human intervention for correct operation is proposed in [40]. This proposal focuses on the problem of control switching in controllers. This suitably fits with the use of formal languages such as linear temporal logic (LTL), which is used in the analyzed work. Similar in concept to [40] is the work presented in [41], though it synthesizes automated controllers that perform joint tasks with human operators rather than controllers that switch between purely human and purely autonomous control. Also, the work in [42] focuses on the control transfer problem in the field of robots. This work introduces a framework that describes different ways that a human operator and a robot take actions together and ways that the robotic control is shared. In addition to this particular problem of shared control, our approach deals with the human-system interactions that are required to perform human-system cooperation. Therefore, we include interaction aspects when addressing the building of HiLCPSs. Cranor proposes a framework for reasoning about human-in-the-loop in secure systems in [43]. This work provides a systematic approach to identify potential causes for human failure. This framework can be used by system designers to identify problem areas before a system is built and to proactively address deficiencies. Since our approach deals with the specification of human-system collaboration, Cranor's work may be complementarily used to identify potential problems in this specification. Also note that the framework proposed in [43] is domain dependent and it is only applicable on secure systems.

In [44] the authors introduced a framework to reason about the effects of changes at the software level on the interactions of users. This approach focuses on a collaborative topology that is capable of offering a collaborative adaptation process to allow more sophisticated adaptations. However, it does not explicitly introduce the user into the adaptation process as our work does. In [1], Jirgl et al. provide an approach to predict human reactions under different conditions and influencing factors from the information gathered about human interaction with a device or a process. The authors claim that this knowledge allows the human environment to be optimized, ensuring higher effectivity, performance, and elimination or reduction of the human error. Our approach is also concerned with this issue even though we use a different perspective to tackle it. We use fast-prototyping as a tool for assessing human-system collaboration designs in the first stages of the development life cycle. Validation by means of fast-prototyping allows us to build more suitable designs of human-system collaboration that lead to higher effectivity and performance in human cooperation.

Finally, it is important to highlight that one of the goals of Industry 4.0 is to include human workers in Cyber-Physical Systems to provide manual, value-added content. In [45], a CPS architecture is augmented to include consideration of how the human element must be integrated. However, while this architecture proposes a general structure that is composed of five levels to guide the implementation of what is called Cyber-Human Systems, our work is located at a low abstraction level and deals with techniques to address the different levels introduced by the architecture. Also, the work presented in [46] takes into consideration the demand of integrating cognitive capabilities in the loop of production-related processes in the context of industrial Cyber-Physical Systems. The work introduces some key concepts of human cognitive capabilities in socio-technical systems and then implements them in different application cases.

2.3. Summary of the literature review

Our review of the literature shows the need for proposals that provide designers with assistance to design human-in-the-loop solutions in the early stages of the CPS development life cycle. Our proposal tries to bridge this gap since there is a lack of design methods that cover this need [1]. However, we have found some works that deal with aspects that are directly related to our approach, which have inspired us as a basis for building our proposal. Following, we identify which works are baselines for our approach and summarize them in Table 1, where the main characteristics of our approach are identified and the works of the literature review related to these characteristics are highlighted.

We have reviewed works that deal with task allocation and task analysis issues, which are related to the problem of sharing the control flow between the human and the system in HiLCPSs. As mentioned, our approach is based on the foundations of functional models ([9, 10, 11, 12, 13]) and task models ([14, 15, 16, 17]) for describing user and system tasks. However, it introduces new aspects to be modeled when designing the shared control flow of collaborative tasks (the attentional resources required by collaborative tasks and the human factors that can impact the success of human participation).

Moreover, our approach includes the issue of the interaction in order to provide a holistic approach for facing the challenge of building HiLCPSs. Therefore, it also defines the interaction mechanisms required to establish the communication between humans and systems. In addition, as we seek to develop HiLCPSs that perform human involvement in a robust, intuitive, and non-intrusive way, we have based our proposal on the quality properties for interactions identified in the literature ([18, 24, 22, 23]). These quality properties are relevant in designing user interactions for intelligent and autonomous systems.

Finally, as is well-known, prototyping is a fundamental aspect of User-Centered Design, and it is a key activity when checking the design before system development. Fast-prototyping is used in several approaches such as in [47, 48], but we have not found this technique applied in the design of collaborative work for HiLCPSs.

Relevant dimensions that our method deals with		**Related works dealing with these dimensions**									
		[9, 10] [11, 12] [13, 14] [15, 16]	[25]	[18]	[21]	[22] [23] [24]	[7]	[31] [32]	[1] [44]	[40] [41] [43]	[47] [48]
Shared Control (function/task allocation)		++	+	+	−	−	−	−	+	−	−
Quality Properties for HCI	Trust	−	−	+	+	+	−	−	−	−	−
	Intelligibility	−	−	−	++	+	−	−	−	−	−
	Understandability	−	−	++	+	+	−	−	−	−	−
	Obtrusiveness	−	+	−	−	+	−	−	−	−	−
Attention Management		−	+	−	−	−	+	−	−	−	−
Interaction Context		−	−	−	−	−	−	++	−	−	−
Modeling human factors		−	−	−	−	−	+	+	+	++	−
Prototyping		−	−	−	−	−	−	−	−	−	++

Table 1: Summary of the main characteristics and relationships with other works

3. Running example: the autonomous car

The best way to intuitively understand the proposal is through a real example. One of the most well-known and popular HiLCPSs today is the case of autonomous cars. Autonomous car technology is ready-to-market and many car units are already driving autonomously on the roads with minimal driver intervention under controlled situations. One representative example of these autonomous cars is the Google car, which is ranked at level 3 (L3) of autonomy [49]. This car is capable of traveling thousands of kilometers without human intervention under restricted conditions (specially marked roads and good weather). Nevertheless, the driver is required to be prepared to intervene at any moment in the case of conflictive situations [50].

The case of the autonomous car at L3 is used as a running example in this work. In order to illustrate our proposal, six tasks that require human integration have been selected from the tasks that are usually performed in an autonomous car. These tasks are the following:

- T1 *Supervised Autonomous Driving*: This represents the collaborative work that the system and the human perform in AutoPilot mode. The AutoPilot (an autonomous system process) is responsible for performing the automated driving task (lane keeping with adaptive cruise control) in an operational situation. However, the L3 AutoPilot mode requires a driver to always be in attentive mode and ready to take over (if required).
- T2 *Supervised Manual Driving*: This represents the collaborative work that the system and the human do in Manual mode. The human is responsible for performing the driving task. However, the L3 Manual mode requires the system to supervise the state of the human (and take the appropriate actions when necessary).

These two tasks are the basic ones that are performed in an autonomous car to accomplish the main goal of the car, which is safely traveling between destinations. Each task represents a different driving mode. In order to perform the control transition between autonomous and manual mode, additional tasks are required. Based on [51], the transition of control from the autonomous system to the driver is called *takeover*, while the transition from the driver to the system is called *handover*. Since both takeover and handover can be initiated by the system or by the human, transitions of this type require different tasks to perform the transition of control, which are the following:

- T3 *Handover*: The driver gives the order to the system for it to take over control of the car.
- T4 *Emergency Handover*: The system realizes that the human cannot continue driving (e.g., it realizes the driver has fallen asleep) and takes control of the car in a rush.
- T5 *Takeover*: The system transfers control of the car to the human in a situation that does not require hurry (e.g., the car is approaching a city and the car cannot continue driving autonomously).
- T6 *Emergency Takeover*: The system must leave the car in a safety situation while it transfers control of the car to the human. This task takes place in emergency situations (e.g., a sensor fails and the system cannot continue driving autonomously).

Note that, based on [52], we distinguish between two types of takeover initiated by the system: one corresponds to a situation that does not require a rushed transition and the other represents an emergency situation. This distinction is important since it impacts the design of the transition tasks.

4. Overview of the proposal

The human-system collaboration is crucially important in autonomous CPSs for achieving the system's goals. However, the design of this human collaboration is a complex task that should be carried out in the early stages of the development life cycle by both domain experts and interaction designers. Several aspects of the system, its environment, and, especially, the human must be taken into account to appropriately define how the system control is shared between the human and the system. It is also necessary to determine which interactions are required so that the human and the system can effectively achieve a specific goal. Interactions are crucial in achieving the goal of the collaboration. It is important for humans to trust the system and perceive a good experience when interacting with it, which requires not overwhelming them.

The previous analysis on related work indicates the lack of design methods to define the human-system collaboration in autonomous CPSs. Specifically, the challenge that this work addresses can be stated by the following research questions:

- **RQ1**. How to design the human-system collaboration to achieve interactions that engage the human, provide natural and understandable collaboration, and avoid disturbing the human?
- **RQ2**. How can designers validate human-system collaboration designs at early stages of the development to check if the previous requirements satisfy the users' preferences and needs?

4.1. Goals

The main goals of this work have been developed to answer the research questions presented above. Next, we summarize the main contributions of our work.

Regarding RQ1, the first goal of this paper is to define a **design method** for helping domain experts and interaction designers to analyze and design HiLCPSs solutions where human-system

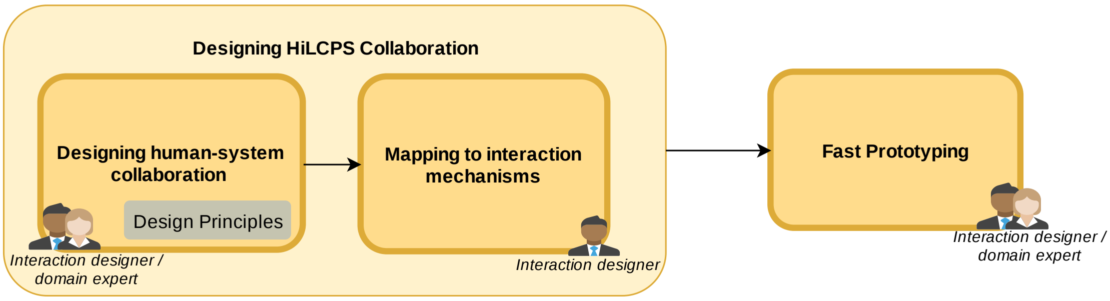

Figure 1: Big picture of the proposal

collaboration is crucially important for achieving the system's goals in the early stages of the development life cycle. With this design method, domain experts and interaction designers should be able to describe the tasks that the human and the system must perform, specify the necessary interactions to involve the human when necessary, and, thus, efficiently achieve the human's engagement to the allocated task. In order to achieve robust, unobtrusive, and intuitive human-system collaboration, some quality factors must be taken into account. Therefore, we propose a set of **design principles** that pursue the satisfaction of quality factors for interactions. Then, based on these principles, we define the concepts to be used in the design stage.

Regarding RQ2, the second goal of this work is to propose a **technique based on fast-prototyping** to gather user feedback that can be used by designers to validate and improve the human-system collaboration designs. The validation focuses on checking whether the human-in-the-loop solutions engage the human, provide natural and understandable collaboration, and avoid disturbing the human in a way that fits the users' preferences and needs.

4.2. Outline of the approach

Fig. 1 summarizes the main steps of the proposal. The first composed phase is the design of HiLCPS collaboration, which in turn is composed of two steps. The first step specifies human participation and describes interactions logically in terms of attentional resources, without links to concrete interaction mechanisms. In other words, interactions are defined independently of the concrete interaction mechanisms of the HiLCPS under development. Then, the second step maps these abstract attentional resources to concrete interaction mechanisms. This allows collaboration at the abstract level to be specified first, and then allows the concrete interaction mechanisms that are available in the HiLCPS domain to be selected.

Finally, an important concern of our proposal is to gather fast feedback from users to learn faster and improve the design. As any system that seeks to satisfy the needs of the user, getting human perceptions in the early stages of the development life cycle is necessary. This is performed in the second phase of the proposal. This feedback helps the domain expert to determine whether the design achieves the goal of the collaboration and helps the interaction designer to check whether the interaction mechanisms are appropriate and whether the system manages to provide the human with feedback without unnecessarily disturbing him/her. In order to gather this feedback from users, the proposal suggests the use of rapid prototypes generated from the design of the collaboration to validate and refine it. It is important to note that this proposal is aligned with the principles of human-centered design of the ISO standard [53] on the ergonomics of human-system interaction.

5. Design principles to define the HiLCPS collaboration

This section identifies a set of design principles that aim at achieving a human-system collaboration design that engages the human, provides a natural and understandable collaboration, and avoids disturbing the human. We first introduce two basic aspects that must be considered in the design of such systems, and then we derive four design principles that must be taken into account to achieve human-system collaborations that fulfill most of the quality factors present in the literature.

The two design aspects for achieving a correct operation of the system and what we propose in order to achieve these aspects are [3]:

- The **control strategy**, which determines how the system control is shared between the human and the system. Therefore, the proposed method allows domain experts to describe the tasks that the human and the system must perform. Moreover, since human factors and their context greatly impact collaboration performance, this method provides mechanisms to get the human's attention when necessary, and thus it efficiently achieves the human's engagement in the allocated task.
- The **interactions**, which are responsible for establishing the appropriate communication so that humans and systems can work in a collaborative way and can understand each other. Interactions are crucial in achieving the goal of the collaboration. It is important for humans to be able to trust the system and have a good experience when interacting with it, which requires not overwhelming them. Our method allows the design of interactions in a way that provides the user with feedback to benefit the human's understanding and trust but without overwhelming him/her.

Taking these two design aspects and the related work review (especially the human-automation interaction properties) as the basis, we identify four design principles:

1. **Share control**: A shared-control relationship that provides a space for human-system integration is required to define how the human involvement is carried out in the HiLCPS [3]. This is addressed in the literature of the human-automation field as task or function allocation. Therefore, this design principle establishes that it is necessary to specify how the control is shifted between the human and the system and which actions are better performed by each one. To do this, it should be noted that there are certain actions in which humans are superior to systems, such as perception, intuitive control, and high-level decision making [54]. For example, in the case of the autonomous car, the T1 *Supervised Autonomous Driving* task shares the system control by allocating the driving function to the system and the supervising function to the human.
2. **Get user attention**: Several works in the reviewed literature ([26], [28] and [29]) have reported that human involvement requires human participation even when human attentional, cognitive, and physical resources are limited. The human can be distracted or focused on other tasks at the moment the CPS could require his/her participation. The human can even lose attention to the task while collaborating with the CPS. The CPS must perform effective actions to obtain the human's involvement and to help him/her to maintain a suitable attention level. Thus, this design principle is necessary to capture the human's attention, and thereby maximize the probability of success when performing the collaborative task. Thus, the HiLCPS must perform effective actions to get the human involved and to help him/her

to maintain a suitable level of attention. In the case of the autonomous car, when the human participation is required by the system, the system should perform actions that are oriented towards engaging the human in certain (collaborative) driving tasks.

3. **Avoid obtrusiveness**: In the context of CPSs, humans are surrounded by a lot of services and devices, such as mobile devices, smart watches, smart cars, etc., which continuously make demands on human attention. One challenge of the HiLCPSs is to regulate the requests for the user's attention, as pointed out in the literature ([22], [23] and [26]). Therefore, the system must avoid disturbing or overwhelming the user with unnecessary actions that can require too much attention or cause undesired results or bad user experiences. Thus, this design principe states that the system should regulate the extent to which it places demands on the user's attention to avoid disturbing or overwhelming him/her with unnecessary information. In the example of the autonomous car, the system should adapt the interaction mechanisms used to notify the driver according to the driver's situation.
4. **Achieve understandability**: Several works, such as [18], [21], [22] and [23], state that human-computer interaction must be present in autonomous system design in a way that makes the decisions of the systems and their behavior understandable by the users. This will allow users to understand what the system is doing with a measure of trust and confidence. Some information should be given to the human to obtain his/her confidence. Thus, this design principle establishes that it is necessary to make use of mechanisms that provide relevant feedback and feedforward to the human. For example, the task T1 *Supervised Autonomous Driving* informs the user about the actions and decisions related to autonomous driving.

Thus, our method proposes defining the human integration into the CPS by fulfilling these design principles.

6. Design of HiLCPS collaboration

Human-system collaboration is performed within what we call *HiL tasks*, which are (autonomous) system tasks that are extended to require (to some degree) human participation to complete the task goal. In order to integrate humans into some of the tasks of CPSs, a redesign of their behavior is necessary. Therefore, we assume the way of working for HiLCPSs presented in [55]. A simplification of this HiLCPS behaviour is the following:

1. The HiLCPS is continuously monitoring its environment to identify situations in which human integration is required. For example, in the case of the autonomous car, the system monitors events or conditions that require human participation, such as adverse weather conditions or the proximity to a city.
2. If the HiLCPS identifies a situation that requires human participation, it raises an event that triggers the need for a human to provide help or to carry out a HiL task. For example, the autonomous car identifies that the car is approaching a city and requires transferring control to the human.
3. The system requests a human with specific capabilities to perform the task. The system requires a human with driver capabilities to transfer control.
4. If the human is available, the system runs the HiL task. If a human with driver capabilities is available in the car, the system transfers control to him/her.

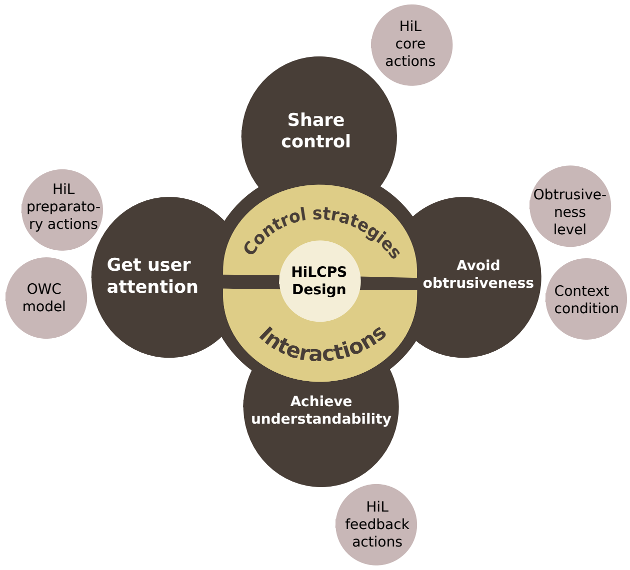

Figure 2: Relationships of HiLCPS design aspects, design principles, and concepts of the proposal

During the execution of the task, errors can occur. In those cases, the system should execute alternative actions (e.g., a fallback plan) to solve an inappropriate situation in order to complete the task successfully and to leave the system in a consistent and safe state. For example, in the case of the autonomous car, the fallback plan may be "*Drive the car to a minimal risk condition*".

Subsection 6.1 introduces the concepts that we propose for specifying HiL tasks. In the first step, the concrete interaction mechanisms are not considered. Then, Subsection 6.2 presents the next phase of the development process where the concrete interaction mechanisms are specified.

6.1. Defining HiLCPS Collaboration

This section identifies the concepts for designing the human-system collaboration based on a HiL task. The concepts are introduced in clusters according to the design principle they support. Fig. 2 shows a graphical metaphor that displays the different concepts that are involved in the HiLCPS design. The figure highlights the relationships of the design principles and the concepts introduced to address them.

The concepts introduced in this section constitute the vocabulary of a language for the design of HiLCPS collaboration. Fig. 3 shows the metamodel with these concepts and their relationships. The metamodel is organized around the *HiL Task* class. Each of its instances represents a system task where a human participates. As we have stated, a HiL task can have a *Fallback Plan* associated to it.

6.1.1. Share control

Human-system collaboration can be decomposed into *actions* that are shared by the human and by the system. As the metamodel of Fig. 3 shows, *actions* compose a *HiL task*. These actions are simpler subtasks whose aim is to achieve a subgoal of the HiL task. An action represents a perceivable interaction that is performed either by the system or by the human, or an internal function that is carried out by the system. The basic actions that allow the human and the CPS

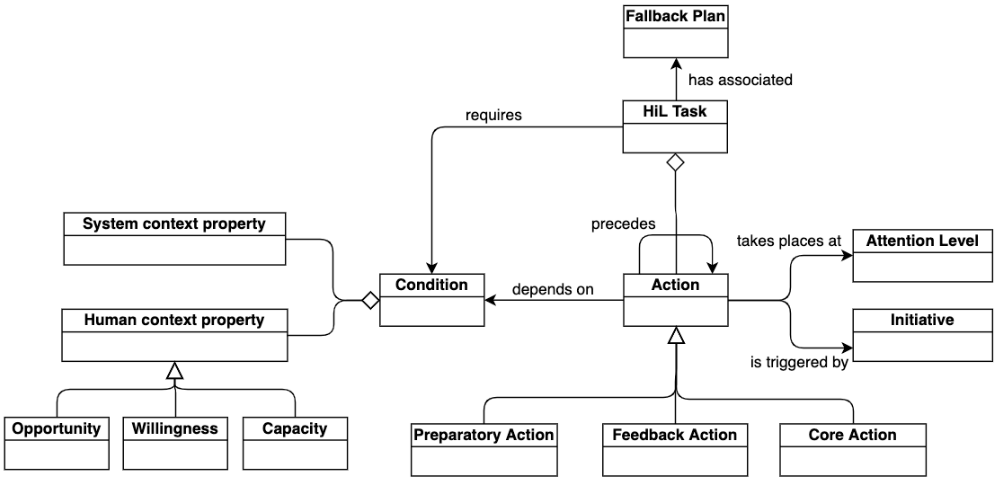

Figure 3: Human-system collaboration design metamodel

to collaborate and to perform the HiL task are called **Core Actions**. The metamodel shows the different types of actions, which include the Core Actions. For example, the T5 *Takeover* task can be decomposed into three core actions:

1. *The system notifies the human that s/he is required to take over control of the car.*
2. *The human confirms the takeover.*
3. *The system transfers control to the human.*

These three actions are necessary to perform the basic functionality of the T5 *Takeover* task.

Human-system collaboration follows a process that determines the sequence of actions that allows the goal to be achieved. This means that actions are inter-related to determine a partial *order* in which they must be processed. This partial order determines a precedence relationship between a pair of actions. The metamodel represents it with the *precedes* relationship.

Fig. 4 shows the proposed notation for the T5 *Takeover* task; the diagram on the left is called the *Core and Feedback Action Diagram*. This diagram shows the core actions in which the T5 *Takeover* task is decomposed and their order relationship. Core actions are represented by a green circle. The temporal order between the actions is represented by an arrow. The initial and final actions are marked with initial and final points.

6.1.2. Achieve understandability

HiLCPSs require a high level of trust to be accepted in our daily lives. Trust is defined as *the level of confidence a human has in an autonomous system* [56]. In order to build trust in HiLCPSs, the systems must include a number of essential actions that are oriented towards enhancing the human's confidence. This can be done by including explanations about what and why the system operates as it does. Therefore, we propose HiL tasks that include actions that are oriented to providing explanations to the human so that s/he properly receives knowledge to collaborate. These actions are called **Feedback Actions** (note that they are included as a subtype of the *Action* class in the metamodel). For example, for the T5 *Takeover* task, we include the feedback action *the system informs the driver about the new driving mode*, which must be executed once the human takes

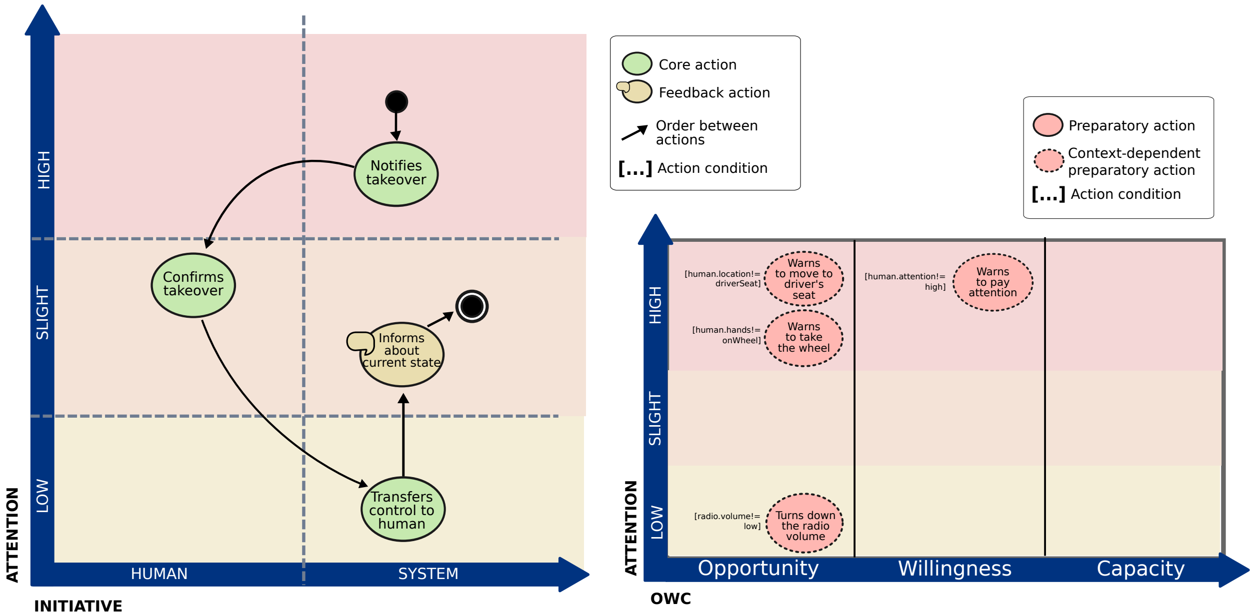

Figure 4: Core and Feedback Action Diagram and Preparatory Action Diagram: T5 *Takeover* task

control of the car. Feedback actions help us satisfy the second design challenge, which is to *achieve understandability*.

Feedback actions are in the same control flow of the HiL task as core actions. Therefore, an order relationship must be defined between these types of actions. Fig. 4 shows the feedback actions together with the core actions for the T5 *Takeover* task. In this case, the feedback actions are represented by a yellow circle.

6.1.3. Get user attention

The correct operation of HiLCPSs crucially depends on human performance. Since humans are alive and their behaviors are influenced by critical factors (e.g. fatigue, stress, motivation, etc.), the use of context is necessary in HiLCPSs. By improving the system's access to context, we increase the richness of communication in human-computer interaction and make it possible to produce more useful interactions. The *context* definition that Dey introduced in [57] is used to specify the semantics of the context concept: *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is relevant to the interaction between a user and an application, including the user and applications themselves"*. From this definition, it is possible to derive the relevant properties of the context as the state of the physical environment (e.g., temperature, noise, etc.), the state of the resources regarding the system surroundings, the state of the system itself, and the user's situation (e.g., its location, activity, etc.). The user's situation acquires special importance in HiLCPSs when humans are an active part of the system. To better represent their context, we use the **opportunity-willingness-capability (OWC) model** described in the context of cyber-human systems [38]. This model classifies a set of HiLCPS elements (especially human factors) on which task performance is conditioned. With this model, to achieve proper task performance, the model state must be the intersection of possible model states for which OWC (as described below) can be evaluated as true. Specifically, the different elements can be categorized into:

- *Opportunity:* This captures the prerequisites for task performance in terms of a set of variables that are required to attempt a task. These variables are related to task-specific opportunity elements. This identifies the set of variables that humans need to fulfill in order to attempt a task, for example, the human's situation (i.e., the user's location, the activity that is being carried out, etc). In the autonomous car, possible variables and states for this element are *human.location = inDriverSeat*, *human.hands = onTheWheel*, *radio.volume = low*.
- *Willingness:* This indicates the human's predisposition to perform the task. This factor is related to human attention, stress level, load, motivation, or how busy the human is at the time. It is important to note that the user will not be able to perform the task if s/he does not have the necessary attentional resources available that the task requires. Possible variables and states for this element are *human.attention = high*, *human.stressLevel = low*.
- *Capacity:* This defines the human's skills and abilities that are necessary to successfully execute the task. It is related to the human's knowledge, level of experience or training, cognitive or physical skills, required devices, emotional states that may reduce the human's ability, etc. Possible variables and values for this element are *human.experience = high*, *human.levelOfTraining = high*.

Under this definition of human context, specific human context conditions can be required to execute HiL tasks. These conditions determine the appropriate human situation for participating in the system and are required to maximize the chance of successfully completing the task. The metamodel represents this kind of condition by means of the *Context condition* class that *is required* by a *HiL Task*. A *Context condition* instance is composed of *System context properties* and *Human context properties*. The latter can be of different types according to the OWC model. For example, in order to perform the T5 *Takeover* task, the human must be attentive to the system, s/he must be in the driver's seat, and his/her hands must take the wheel. These conditions must be satisfied to guarantee proper task performance, and, consequently, a safe takeover (the human may not likely be aware of the notification about taking control of the car). Therefore, we include specific actions to the T5 *Takeover* task to prepare and maintain a proper human context. We call these actions **Preparatory Actions** (see them in the metamodel as a subclass of *Action*). Therefore, the task should include the following preparatory actions:

1. *The system warns the driver to take the wheel.*
2. *The system warns the driver to move to the driver' seat.*
3. *The system warns the driver to pay attention.*

HiL preparatory actions are not related to the core and feedback actions with a precedence order; the HiL preparatory actions could be activated at any moment throughout the execution of the task in an order that is independent of the core and feedback actions. This is due to the fact that preparatory actions are in a different flow than the core and feedback actions. Both flows are executed in the HiL task. Preparatory actions are represented in the *Preparatory Action Diagram* (Fig. 4, right). This diagram shows the definition of the proper human context for performing the task and the preparatory actions for achieving that context. The proper context is divided into the three dimensions of the OWC model. For each dimension, the human condition is specified and the preparatory actions (represented by a red circle) are defined and associated to each human condition.

6.1.4. Avoid obtrusiveness

In CPSs, humans are surrounded by a lot of services and devices, such as mobile devices, smart watches, smart cars, etc. Since these can continuously make demands on one of the most valuable resources of users, human attention, a challenge of the HiLCPSs is to regulate the requests for users' attention. In other words, interactions should behave in a considerate manner by taking into account the degree to which each interaction intrudes on the user's mind (i.e., the degree of obtrusiveness).

Therefore, we must manage human attentional resources to get the user's attention without overwhelming the human. The HCI has proposed solutions for managing human attention based on levels of automation and workload [58, 59, 60]. We analyzed and designed the actions from two points of view: 1) who *initiates* the interaction (the system or the human); and 2) the *attentional resources* needed to perform the interaction (the attentional demand imposed on the human). The metamodel shows the *takes place at* and *is triggered by* relationships from the *Action* class to the *Attention Level* class and the *Initiative* class, respectively. We consider this specification of *initiative* and *attention* (attentional demand) to be appropriate since these are factors that vary independently. For example, an interaction of feedback (initiated by the system) may require more or less attention depending on the importance/criticality of the information to be offered. In the same way, a keeper action could be initiated by the system (if it tries to capture the user's attention) or by the human (if the system waits for an answer from the human to confirm that s/he is prepared). The intersection of the level of attention and initiative is called the **obtrusiveness level**. Each HiL action is associated to an obtrusiveness level, for example:

$NotifiesTakeover = \{(Initiative, human), (Attention, high)\}$

$TransfersControlToHuman = \{(Initiative, system), (Attention, low)\}$

The *Core and Feedback Action Diagram* and the *Preparatory Action Diagram* of Fig. 4 have a vertical axis, which displays the attention dimension, and a horizontal axis, which displays the initiative dimension. In this example, the initiative dimension is divided into two segments according to who initiates the action: the *human* or the *system*. The attention dimension is divided into three segments: *low attention* –if the user is required to make a small effort to perceive the interaction–; *slight attention* –if the action implies medium demand for the human attentional resources–; and *high attention* –if the user is required to be fully conscious of the interaction–. In each quadrant, the designers should locate the different actions for performing the task. For example, in the *Core and Feedback Action Diagram* (Fig. 4, left), the first action in the figure is *Notifies takeover*, which requires high attention and is performed by the system. The second action, *Confirms the takeover* is performed by the human and requires slight attention. Then, there are two actions performed by the system: *Transfers control to the human*, which only requires low attention (from the human) and *Informs about the current state of the car*, which requires a slight attention level.

The obtrusiveness level will later help designers in selecting the most suitable interaction mechanisms for each action. It determines the type of interactions that can be used to interact with the human in terms of his/her attentional resources. It is worth noting that actions that are system functions are associated to an obtrusiveness level as well. This is because these actions can entail an interaction that makes them perceivable (in a more or less noticeable way) to the user. For example, the action *the system transfers control to the human* can be tied to an interaction so that the human becomes aware of the transfer.

In addition, when designing HiL tasks, we use the contextual information of the user (e.g., what the user knows or where the user is) or the system to achieve a better adjustment of human-system interactions without overwhelming the user with unnecessary information. For example, if the system is aware that the human is paying attention, the preparatory action of warning the human to pay attention would be avoided. Therefore, actions can be designed to be *context-dependent*, and, depending on the user or system context, they may or may not be executed within the task. A condition is attached to a *context-dependent action* to determine whether the action must (or must not) be executed based on the situational context. This is represented in the metamodel by the *depends on* relationship between the *Action* class and the *Condition* class.

This condition acquires special importance in determining whether or not an action is appropriate for a specific situation of the user. For instance, the condition that determines the execution of the preparatory action *the system warns the driver to take the wheel* of the T5 *Takeover* task would be:

Opportunity: [*human.hands != onTheWheel*].

As shown in Fig. 4, the context-dependent actions are represented in the *Preparatory Action Diagram* by a dashed circle. These actions are labeled with their corresponding conditions.

6.2. Mapping to Interaction Mechanisms

The second step in the design of HiLCPS collaboration is to determine *which* concrete interaction mechanisms are used to interact with the human. According to our proposal, every HiL action is defined at a specific obtrusiveness level (intersection of attention and initiative). This division in obtrusiveness levels helps designers to classify the types of interaction (at a high level of abstraction) and then, select the most appropriate interaction mechanism (at a low level of abstraction) to support the HiL action. The following subsections explain these two steps.

6.2.1. Define abstract interaction modalities

In the first sub-step, the interaction designer must associate each obtrusiveness level with a set of **interaction modalities** that support its initiative and attention requirements. An interaction modality is a type of input or output that is associated to a concrete interaction with a system. Obtrusiveness levels that require more attention must be associated with interaction modalities that are more obtrusive or that draw more attention. Conversely, obtrusiveness levels that require less attention must be associated with discreet interaction modalities.

Figure 5 (left) shows an example of interaction modalities that support each obtrusiveness level in the autonomous car domain. Each interaction modality determines the modality type and a manifestation of this modality (represented as *modality.manifestation*). In the autonomous car domain, we specify three modality types: visual, auditory, and haptic. Each modality includes a set of manifestations of input/output interactions. For example, manifestations of the auditory output include beeps, synthetic speech, music, and an acoustic alarm. Each manifestation has its own features, based on which it can be identified and selected for use. It is important to note that actions initiated by the system (system initiative) will be supported by output interaction modalities and that actions initiated by the human (human initiative) will use input interaction modalities.

The selection of interaction modalities for the obtrusiveness level of Fig. 5 *(system, high)* is assigned as follows:

$Obtrusiveness_{(system,high)} = \{auditory.synthetic\_speech, visual.textual\}$

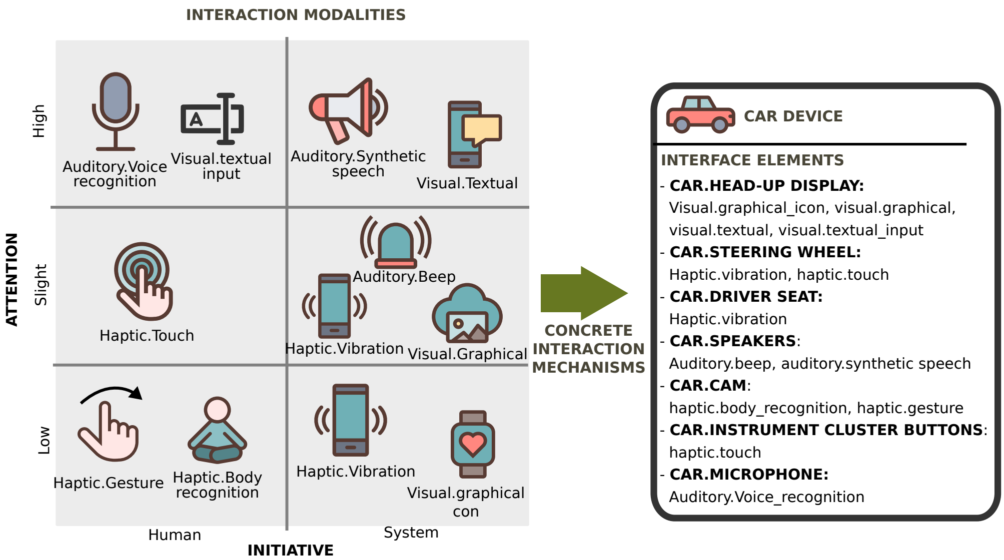

Figure 5: Selection of interaction modalities of each obtrusiveness level and concrete interaction mechanisms of the car device

The selection of the interaction modalities that best suit each obtrusiveness level can be based on existing multimodal design taxonomies and frameworks that depend on the cognitive characteristics of interaction modalities, i.e., how information carried by different modalities is perceived by the human perceptual-sensory system [61, 62, 63, 64]. For example, an image uses less mental workload than a text, and visual-auditory combinations impose less cognitive load than visual-visual combinations [64]. Auditory modalities are useful for attention alerting [65] and vibration or lighting feedback do no interrupt other activities [66].

6.2.2. Selection of concrete interaction mechanisms

In the second step, the interaction designer should choose the concrete interaction mechanisms for each HiL action based on the classification of interaction modalities. To do this, the interaction designer needs to map each interaction modality identified in the previous step with the concrete interaction mechanisms that are available in the system domain.

Due to the nature of CPSs, interaction is not confined to one device; it should encompass multiple physical devices [67, 3]. Therefore, interaction mechanisms that are domain-dependent are described by means of: 1) the interface element that supports the interaction modality (e.g., a button, a vibration, a screen, etc.); and 2) the computing device that offers the interaction mechanism (e.g., a mobile, a smart car, etc.). For example, concrete interaction mechanisms can be *a visual message on the screen of the mobile phone*, or *a visual message on the screen of the car*.

In the autonomous car domain, several systems under research have been described in the literature identifying methods that warn drivers to improve situational awareness. These methods include vibration on the steering wheel [68] or the driver's seat [69], audio alerts and visuals on the head-up display [70, 71]. From these studies, Fig. 5 (right) defines the possible interaction mechanisms that can be used for each interaction modality shown in the left part of the figure. The figure shows the supported interaction modalities for each interface element of the car device. It is worth noting that we include the computing/physical device that offers the interaction mechanisms

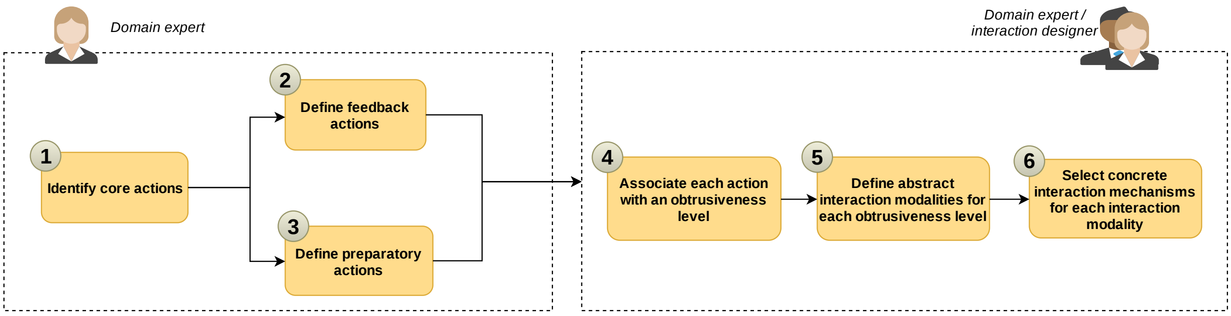

Figure 6: Graphical representation of the design process

in the specification of the interaction mechanisms. This is useful when the interaction mechanisms are in different computing devices. In our example, we only use the interaction mechanisms of the car, but we could use interaction mechanisms of the mobile device such as mobile speakers or notification messages.

In order to specify which interaction mechanisms support a certain HiL action for a given obtrusiveness level, we use the *Superimposition operator* ($\bigodot$). This operator takes a HiL action and an obtrusiveness level and returns the set of concrete interaction mechanisms required for that action. Some examples of the relationship between the obtrusiveness levels and the actions for the T5 *Takeover* task are as follows:

$\bigodot Notifies\_takeover_{(system,high-attention)} = \{Car.voice\_feedback\_speakers\}$

$\bigodot Confirms\_takeover_{(human,slight-attention)} = \{Car.steering\_wheel\_pressure\}$

$\bigodot Inform\_current\_state_{(system,slight-attention)} = \{Car.image\_head\text{-}up\_display, Car.beep\_speakers\}$

6.3. The design process

We propose a process for defining HiL task specifications using the identified concepts. The process is designed to be carried out by domain experts and interaction designers at early stages of the software development life cycle. Fig. 6 shows the process graphically. It consists of the following steps for the description of each HiL task:

1. **Identify core actions** to achieve the assigned goal of the task. The work to be performed by the system and by the human must be specified throughout HiL core actions. The domain expert decides which core actions are allocated to the human and which are allocated to the system. These actions are the minimum set of actions necessary for achieving the goal of the task. The actions must be ordered in a workflow that determines the shared control flow between the human and the system.
2. **Define feedback actions** required for providing feedback or feedforward to the human. Once the core actions have been specified, the domain expert studies what information may be provided to the human to give him/her a high level of confidence in the system. This information is provided by means of feedback actions. It allows humans to comprehend the state of the system under control and efficiently perform their work or supervise the system's activities.
3. **Define preparatory actions**. The domain expert establishes which context is the proper one for maximizing the task performance. Then, s/he defines the preparatory actions that

help the system to achieve this context. These preparatory actions have an associated context condition that triggers them.

4. **Associate each action with an obtrusiveness level**. The interaction designer and the domain expert decide what is the most appropriate obtrusiveness level for each action.
5. **Define abstract interaction modalities for each obtrusiveness level**. The interaction designer must associate each obtrusiveness level with a set of abstract interaction modalities according to the requirements of that obtrusiveness level (initiative and attention level).
6. **Select concrete interaction mechanisms for each interaction modality**. The interaction designer associates each interaction modality to the concrete interaction mechanisms that are available in the system domain.

As Fig. 6 shows, Step 2 and Step 3 can be performed in parallel. Step 4 can be performed once both Step 2 and Step 3 have been completed.

In order to support Steps 1-5, we propose the graphical notation for specifying HiL tasks shown in Fig. 4. To support Steps 6-7 we propose the textual notation shown in Subsection 6.2.

7. Fast-prototyping for validation and refinement

The previous sections introduced a design method for the definition of HiLCPS collaborations. However, when a HiL task is designed, there is no guarantee that the resulting interactions meet the four design challenges: *share control, get human attention, achieve understandability, and avoid obtrusiveness*. In order to gather fast feedback from users and check the fulfillment of the design principles, we suggest applying the fast-prototyping technique from the User-Centered Design (UCD) discipline. It allows designers to learn faster and then change the prototype models to improve the design. De Sá and Carriço [48] showed that prototyping techniques can be determinant during the consequent evaluation stages, allowing users to freely interact with the system, improve it, and use it in realistic settings without being misled.

In this section, we introduce a technique for the early-stage evaluation of HiL tasks by means of fast-prototyping. Even though the proposed prototypes can be built quickly, they are capable of reproducing a level of user experience that is considered to be very close to what users expect from a final system. Thus, flaws in the HiL actions defined for HiL tasks and the interaction design can be detected before efforts are made in the development of the final system. Fast-prototyping involves the following steps: 1) build an early-stage prototype, 2) perform the test, and 3) analyze the results. The following subsections apply these steps for the example of the autonomous car.

7.1. Build an early-stage prototype of the autonomous car

The goal of fast-prototyping is to immerse the user in an environment that makes the user feels as if s/he is using the final system despite the fact that a non-functional prototype is being used. This prototype simulates the flow of the HiL actions within a HiL task using Wizard-of-Oz techniques [72] and uses the defined interaction mechanisms for each HiL action to provide the user with the expected interaction given a set of context conditions and a HiL action. Simulation is performed by an operator that provides the current HiL action based on the action diagram model and the user context situation. This user context situation is also taken into account by the operator to provide an appropriate HiLCPS interaction.

The architecture of the autonomous car prototype is made up of two components: 1) a *Physical Platform*, which is made up of the physical devices used to simulate the interaction; and 2) a *Control*

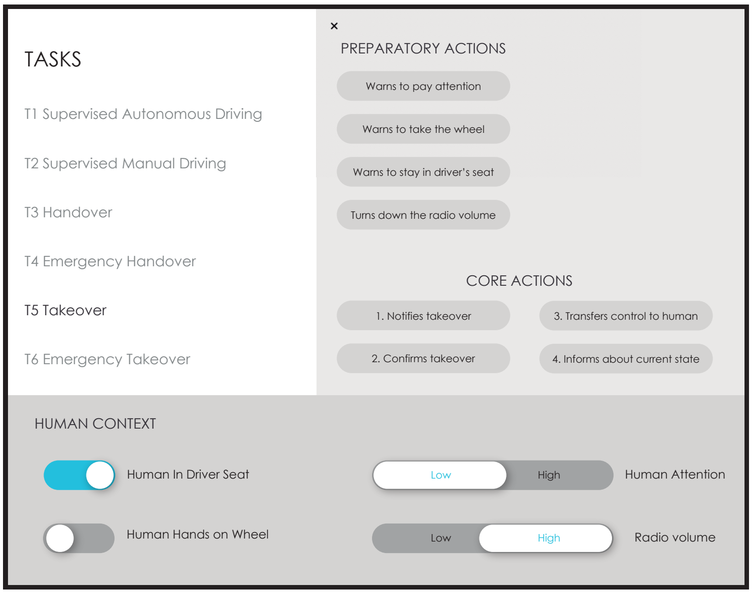

Figure 7: Web interface for the operator to simulate context and trigger the HiL actions

Unit, which is responsible for controlling the execution of the HiL tasks triggered by the operator. These components perform the following functions:

1. The ***Physical Platform***. This platform provides a set of devices that allow the HiLCPS interactions to be simulated. The car simulator uses the following devices: a brake pedal, a steering wheel, a speaker to provide voice feedback, a car console to show visual feedback, a small pilot bulb, and a button to confirm actions.
2. The ***Control Unit***. This component is responsible for simulating the HiL tasks. Using the Wizard of Oz, an operator simulates the flow of the HiL tasks according to their specification [1]. The operator has a web interface available to trigger the actions and to simulate the human context conditions as shown in Fig. 7. The interface shows the list of tasks (on the left) and the HiL actions for each task (on the right). Moreover, it shows a panel (at the bottom) for changing the human context. The interaction mechanisms between the human and the car are also simulated using the physical devices from the *Physical Platform*. The Control Unit also implements a small web application (built using HTML, JQuery, and Bootstrap) that simulates a car dashboard like the one used in Tesla Model 3 (see Fig. 8). This dashboard provides mechanisms to interact with the user as if a real car were being used. We have used the concrete interaction mechanisms that are shown in Fig. 5 (right) for each obtrusiveness level.

7.2. Perform the test

We tested the prototype by means of an experiment where several participants "drove" the prototype of an autonomous car. The following subsections present the details of the experiment.

[1] The specification of all the tasks can be found in: http://hil.tatami.webs.upv.es/docs/AutonomousCarTaskSpecification.pdf

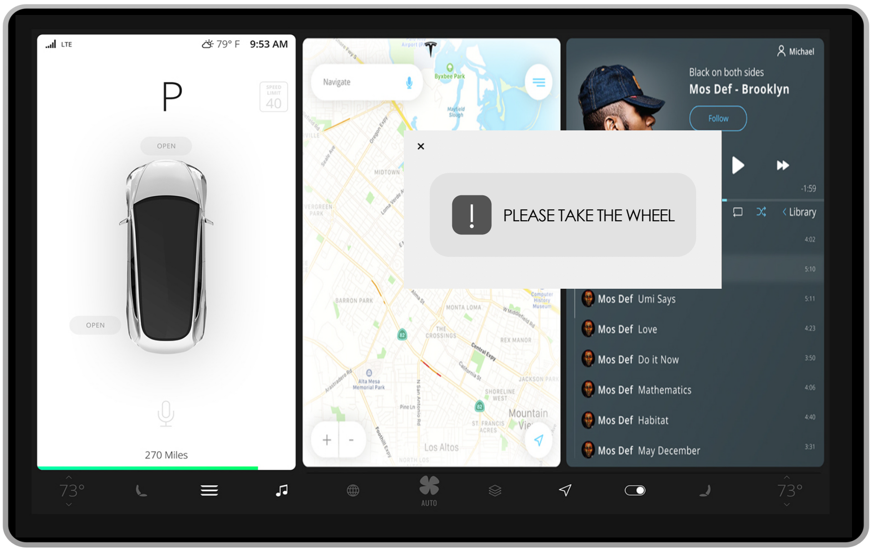

Figure 8: Web interface dashboard of the car prototype

7.2.1. Goal

According to the Goal/Question/Metric template [73], the research goal is:

Analyze *the prototype of the autonomous car*

For the purpose of *evaluating the HiLCPS collaborations*

With respect to *the fulfillment of the design principles*

From the viewpoint of *users*

In the context of *end-users using the prototype of the autonomous car*

Considering this research goal, the research questions and the null hypothesis proposed for the experiment are:

RQ1: *Do the HiLCPS collaborations of the prototype of the autonomous car fulfill the "share control" design principle?* The null hypothesis tested to address this research questions is: H_{01} - *The HiLCPS collaborations implemented in the prototype do not fulfill the "share control" design principle.*

RQ2: *Do the HiLCPS collaborations of the prototype of the autonomous car fulfill the "get user attention" design principle.* The null hypothesis tested to address this research question is: H_{02} - *The HiLCPS collaborations implemented in the prototype do not fulfill the "get user attention" design principle.*

RQ3: *Do the HiLCPS collaborations of the prototype of the autonomous car fulfill the "avoid obtrusiveness" design principle.* The null hypothesis tested to address this research question is: H_{03} - *The HiLCPS collaborations implemented in the prototype do not fulfill the "avoid obtrusiveness" design principle.*

RQ4: *Do the HiLCPS collaborations of the prototype of the autonomous car fulfill the "achieve understandability" design principle.* The null hypothesis tested to address this research question is: H_{04} - *The HiLCPS collaborations implemented in the prototype do not fulfill the "achieve understandability" design principle.*

7.2.2. Identification of variables

We identified two types of variables:

Independent variables: The human-system collaboration implemented in the prototype was identified as a factor that affects the dependent variables.

Dependent variables: In order to evaluate the prototype, four dependent variables were used:

- *Get human attention.* This was measured by the average human response time. For each task, we timed how long the human took to perform the requested actions (or if the task did not require human actions, how long the human took to be aware of a system action), and we calculated the average. This measurement was used to assess how the HiLCPS collaboration fulfills the *get human attention* design principle.
- *Share control.* This is defined as the user perceptions of performing each task with respect to the design principles *share control.* This is measured by means of a Likert scale from 1 (the lowest score) to 5 (the highest score) points to evaluate each question.
- *Understandability.* This is defined as the user perceptions of performing each task with respect to the design principles *achieve understandability.* This is measured by means of a Likert scale from 1 (the lowest score) to 5 (the highest score) points to evaluate each question.
- *Obtrusiveness.* This variable assesses the user's subjective experience of the overall workload and is used to evaluate the design principle *avoid obtrusiveness.* This is measured by using the NASA Task Load Index (TLX), which evaluates the workload by means of six different sub-scales: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. Moreover, this is also measured by means of a Likert scale from 1 (the lowest score) to 5 (the highest score) points to evaluate each question.

7.2.3. Experimental subjects

A total of six subjects participated in the experiment. We used this number of participants since, according to [74], the best results when testing usability come from testing no more than five users. Note that it is better to distribute the resources for user testing across many small tests instead of using all of them in a single, elaborate study. Therefore, in this paper, we perform a small test that is just the first step of many.

The participants had to be subjects with different profiles (different ages, gender, and expertise) since we considered it necessary to conduct the experiment with a heterogeneous group to avoid any bias in the sample. Therefore, we recruited three females and three males; two were between 19 and 30 years old, two were between 31 and 45 years old, and two were between 46 and 55 years old. The recruitment of the subjects was made by personal invitation (close relatives and personal friends) since we wanted subjects with occupations that were not related to the academic environment or computer science. Therefore, none the participants had any experience in computer science technology or the usage of autonomous cars. Since the selection of the subjects was not random, there is a threat to the validity of the experiment.

7.2.4. Instrumentation

The instruments that were used to carry out the experiment were:

- A script to recreate the user context situation: the description of the user context situation to start each task.
- A Post-Task Questionnaire: A questionnaire of eight questions containing Likert-scale values ranging from 1 (strongly disagree) to 5 (strongly agree) to evaluate satisfaction with the human-system collaboration designs when using the prototype. This questionnaire was created based on similar instruments used[26]. Figure 10 shows the post-task questionnaire that was used to assess the fulfillment of the aforementioned design principles.
- The NASA Task Load Index (TLX) app [75]: A questionnaire of six questions containing Likert-scale values ranging from 1 to 10 to assess the user's subjective experience of the overall workload when using the prototype.

7.2.5. Experimental design, context, and procedure

We followed a simple design (one factor with one treatment), where all subjects were exposed to the treatment. We needed to immerse the user in an environment that made the user feel as if s/he were using the final system despite the fact that a non-functional prototype was being used. The study was conducted in our laboratory to simulate the different scenarios on which the experiment was based. In-situ evaluation was possible since the technique did not require a complex infrastructure. Driving sounds were played as background during the testing.

Before starting the test, the participants were briefly introduced to the main features of the autonomous car. The participants were alone in the laboratory with one member of the testing team, who was the test operator. The test operator observed the process and guided the participant when necessary. To start the experiment, the participants were given a script to recreate the initial user context situation (e.g., the user in the driver seat, attentive to the road, etc.) for each task. The experiment started in the autonomous driving mode. From this mode, the set of tasks being tested were performed. The tasks were spaced with gaps of 30 seconds where the participants were encouraged to use their mobile phones. Every task required human attention and/or participation. The orders for the tasks were the following:

1. T5 *Takeover*. The car prototype is in the autonomous driving mode and transfers control to the user. When this task ends, the car is in the manual driving mode.
2. T3 *Handover*. The car prototype is in the manual driving mode. The test operator asks the driver to transfer control to the car. When this task ends, the car is in the autonomous driving mode.
3. T6 *Emergency Takeover*. The car prototype is in the autonomous driving mode and transfers control to the driver because of an emergency situation. When this task ends, the car is in the manual driving mode.
4. T4 *Emergency Handover*. The car prototype is in the manual driving mode. The test operator asks the driver to focus his/her attention on their mobile phone so that the system takes control. When this task ends, the car is in the autonomous driving mode.
5. T1 *Supervised Autonomous Driving*. The car is driving autonomously for four minutes.
6. T2 *Supervised Manual Driving*. The participant drives the car manually for four minutes.

In the experiment, our six drivers dealt with these seven tasks. Figure 9 shows a scenario of a user doing the T1 *Supervised Autonomous Driving* task.

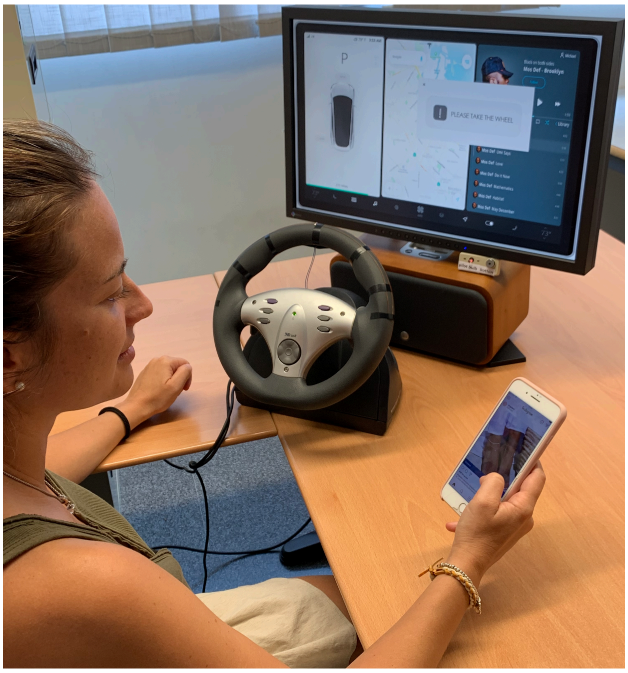

Figure 9: Scenario of the Supervised Autonomous Driving task with an inattentive driver

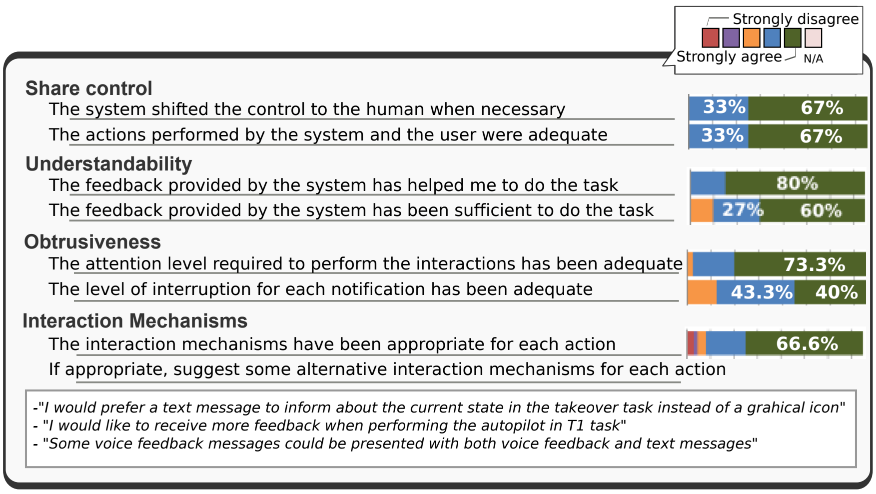

Figure 10: Questionnaire to test the share control, understandability, obtrusiveness, and interaction mechanisms

Table 2: Human Response Time Results

HiL Tasks	*T1*	*T2*	*T3*	*T4*	*T5*	*T6*
Human Response Time	1.35 sec.	1.05 sec.	0.95 sec.	1.25 sec.	1.3 sec.	1.15 sec.

Figure 11: NASA TLX Results

7.3. Analysis and interpretation of results

The results for share control, understandability, obtrusiveness, and interaction mechanisms are shown in Fig. 10. Table 2 shows the results of the measurements of the human response time. The figure and the table show the average measurements for all of the participants. For reasons of simplification, we do not show the results for each participant separately. Figure 11 shows the results of the measurements on obtrusiveness. We show each sub-scale in a different diagram.

7.3.1. Analysis of the fulfillment of the design principles

Fig. 12 shows a summary of the results of human response time, share control, understandability, and obtrusiveness, which are quite positive. To classify the human response time measurements, we took the data reported in [76] as reference. We focused on the relationship between the reaction time (RT) and accidents (which ranges from 0.47 sec. to 2.2 sec., with a mean value of 0.71 sec. and a standard deviation of 0.16 sec.). We considered the following classification: Excellent RT $<=$ 0.5 sec.; Very Good RT $<=$ 0.71 sec.; Acceptable RT $<=$ 1.42 sec. (double of the mean value); Non Acceptable RT $>$ 1.42 sec. Every task is within a minimum on acceptable range for the four design challenges: Share Control has an average of Excellent, Obtrusiveness has an average of Very Good, and Understandability has an average of Excellent.

According to the data in Fig. 11, the mental demand diagram shows that not all of the tasks were simple and easy. Mental demand was mostly low, but some tasks in the experiment required more attention, increasing the mental demand. Some users would prefer more automation in the

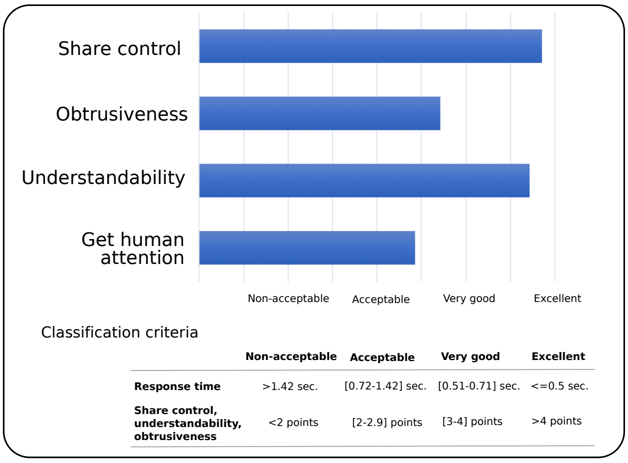

Figure 12: Assessment of fulfillment of design principles

tasks. Physical demand was low except for the tasks that required more attention. Moreover, the users were not familiar with this kind of physical platform. For these users, the physical demand was higher. The low workload was accompanied by good performance. The majority of users could accomplish the goals of the tasks proposed (see the performance diagram) without much effort (see the effort diagram) and with a low degree of frustration (see the frustration diagram). Temporal demand did not provide any significant results since the results are very scattered. These results show that the users did not understand the question very well.

The information obtained related to interaction mechanisms (Fig. 10) shows that the chosen interaction mechanisms were quite suitable. Just two participants made recommendations with regard to alternative interaction mechanisms; in both cases, the recommendation was to use text messages instead of icons or voice.

7.3.2. Interpretation of the analysis

From the analysis of the obtained results, we can state the following lessons:

- The design challenge for *share control* had the best results. The users were satisfied regarding the control that the system shared with them and the actions that the users had to perform. This means that the users agreed with the design of human-collaboration regarding the work that was allocated to the human and to the system.

- The design challenge *get human attention* was assessed as the worst design challenge. Nevertheless, this result could be due to the characteristics of the technique itself that was used for the evaluation. We recreated a driving environment, but the participants were aware that they were performing an experiment. Therefore, response times were more relaxed than in actual driving environments.

- The design challenges *avoid obtrusiveness* and *achieve understandability* both had medium results. However *avoid obtrusiveness* had worse results than *achieve understandability*. In the case of the autonomous car, where secure task performance is a priority, achieving understandability is a goal that is more critical than avoiding obtrusiveness. Therefore, these results are justified.

- The results obtained have allowed us to improve the specification of the tasks. In a first iteration, the results for the understandability variable in most of the tasks were not good. After analyzing the model, we realized that it was necessary to introduce feedback actions in order to let the user know about the driving mode achieved when a task ends.

8. Evaluating the usefulness of the design method

This section presents an experiment that was performed to validate the usefulness of the design method proposed in this paper. The aim of the experiment was to compare the usefulness measure obtained by our proposed method with hand-coding development. The experiment followed the guidelines presented by Kitchenham et al. in [77] and Wohlin et al. in [78]. In the following subsections, we present each experimental element.

8.1. Goal

According to the Goal/Question/Metric template [73] the objective of the experiment was to:
Analyze *our design method to design HiL interactions*
For the purpose of *evaluating its usefulness*
With respect to *a hand-coding development of HiL interactions*
From the viewpoint of *an interaction designer*
In the context of *researchers and practitioners interested in HiL interactions*

8.2. Research question and hypothesis formulation

Since usability is an abstract concept, we need to operationalize it through more measurable concepts. According to ISO 25062-2006 [79], usability can be measured through effectiveness, efficiency, and satisfaction. Following the work of Moody [80], satisfaction can be measured using perceived usefulness (PU), perceived ease of use (PEOU), and intention to use (ITU).

With this perspective, the research questions and the null hypothesis (named H_{0i}, with i = [1..5] corresponding to each research question) proposed for the experiment are:

RQ1: *Is there any difference between the effectiveness of the code-centric method and our method?* The null hypothesis tested to address this research questions is: H_{01} - *There is no difference between the effectiveness of our method and the code-centric method for the development of HiL solutions.*

RQ2: *Is there any significant difference between the efficiency of the code-centric method and our method?* The null hypothesis tested to address this research question is: H_{02} - *There is no difference between the efficiency of our method and the code-centric method for the development of HiL solutions.*

RQ3: *How do subjects perceive the ease of use of our method in relation to the code-centric method?* The null hypothesis tested to address this research question is: H_{03} - *Our method is perceived as easier to use as the code-centric method for the development of HiL solutions.*

RQ4: *How do subjects perceive the usefulness of our method in relation to the code-centric method?* The null hypothesis tested to address this research question is: H_{04} - *Our method is perceived as equally useful as the code-centric method for the development of HiL solutions.*

Table 3: Summary of response variables, metrics, and research questions

Response variable	Metrics	Definition	RQ	Hypotheses
Effectiveness	Percentage of correct tasks carried out (PCT)	This is the relationship between: the number of tasks correctly completed and the total number of tasks	RQ1	H_{01}
Efficiency	Time to finish the task (TFT)	This is the number of minutes spent on the experiment tasks	RQ2	H_{02}
Satisfaction	Perceived ease of use (PEOU)	This is the arithmetic mean of the Likert scale values of MEM questionnaire items related to perceived ease of use	RQ3	H_{03}
	Perceived usefulness (PU)	This is the arithmetic mean of the Likert scale values of MEM questionnaire items related to perceived usefulness	RQ4	H_{04}
	Intention of use (ITU)	This is the arithmetic mean of the Likert scale values of MEM questionnaire items related to intention of use	RQ5	H_{05}

RQ5: *What is the intention to use of our method compared to the code-centric method?* The null hypothesis tested to address this research question is: H_{05} - *Our method has the same intention to use as the code-centric method for the development of HiL solutions.*

8.3. Identification of variables and metrics

The development method was identified as a factor (aka "independent variable") that affects the response variable. This variable had two treatments: 1) our method and 2) the code-centric method.

Response variables are the effects studied in the experiment that are caused by the manipulation of factors. In this experiment, we evaluated our method with regard to: effectiveness, efficiency, and satisfaction.

The metrics used to answer the research questions RQ1 and RQ2 (effectiveness and efficiency) were the percentage of correct tasks carried out in the development of HiL solutions (PCT) and the time required to finish the task (TFT). In addition, to answer research questions RQ3, RQ4, and RQ5, we defined a metric for each one with the aim of measuring satisfaction through PEOU, PU, and ITU. We used a 5-point Likert scale to measure PEOU, PU, and ITU[2]. Table 3 describes response variables, their metrics, definition, and the research questions that we aim to answer.

[2]We are aware that Likert scales are qualitative data, but some studies propose converting them to quantitative data to work with statistical tests [81]

8.4. Experimental context

8.4.1. Experimental subjects

The experiment was conducted in the context of the Universitat Politècnica de València (Spain). We had 22 subjects (16 males and 6 females) who were students in the Master's Degree program in Software Engineering, Formal Methods, and Information Systems. The experiment was part of the "Design of Ubiquitous and Adaptive Systems" course. The background and experience of the subjects were found through a demographic questionnaire handed out at the first session of the experiment. This instrument consists of 10 questions on a 5-point Likert scale. According to the questions included in the demographic questionnaire, we concluded:

- The subjects were between 22 and 40 years old.
- All of the subjects had an extensive background in Java programming and 72.73% had some experience in modeling tools.
- All of the subjects had knowledge about OSGi technology since it is taught in the "Design of Ubiquitous and Adaptive Systems" course.
- For the experience using Eclipse IDE, 36.36% of the subjects reported that they were ranked as "Expert", and 63.64% considered that they were "Intermediate".
- Few subjects had experience in development of Internet of Things systems (18.18%) and no one had experience in the development of HiL solutions, although all of the subjects had taken a human-computer interaction (HCI) course as part of the Computer Science degree program.

8.4.2. Experiment design

In this experiment, we used a one-factor design with two treatments. This is a type of design where each subject is randomly assigned to one method (treatment). Since we had the same number of subjects per treatment, the design was balanced.

The expected time to fulfill the tasks defined in each treatment was around one hour, and we established this a maximum time for the experiment. This value was estimated based on a previous pilot test and the KLM method (Keystroke Level Method) [82, 83] for predicting the time that an expert user needs to perform a given task on a given computer system.

8.4.3. Experimental objects

The experiment was conducted using the running example used throughout the paper, i.e., the autonomous car. The object used in the experimental investigation is a requirements specification created for this purpose. It contained the description of the T5 *Takeover* task[3].

In order to shorten the evaluation process for both development methods and to achieve similar implementations from user to user, we provided the subjects with an example of a HiL task to guide the development of the autonomous car tasks. Specifically, we provided them with a modeling example of the T3 *Handover* task as well as its implementation.

[3] The requirement specification document can be downloaded from: http://hil.tatami.webs.upv.es/docs/RequirementsSpecification.pdf

8.4.4. Instrumentation

The instruments that were used to carry out the experiment were:

- **A demographic questionnaire:** A set of questions to know the level of the users' experience in Java/OSGi programming, modeling tools, and task modeling. This document included questions containing Likert-scale values ranging from 1 (strongly disagree) to 5 (strongly agree).
- **Task description document for the code-centric method[4]:** A document that describes the work/activities to be performed in the experiment using the code-centric method and containing empty spaces to be filled in by the subjects with the start and end times of each step of the experiment. This document contained guidelines to guide the subject throughout the experiment and the source code of the autonomous car prototype to simulate the implemented tasks.
- **Task description document for our method[5]:** A document that describes the work/activities to be performed in the experiment using our method and containing empty spaces to be filled in by the subjects with the start and end times of each step of the experiment. This document contains guidelines to guide the subject throughout the experiment and the source code of the autonomous car prototype to simulate the implemented tasks.
- **Post-test questionnaire:** A questionnaire with 16 questions containing Likert-scale values ranging from 1 (strongly disagree) to 5 (strongly agree) to evaluate satisfaction with the entire process.

8.5. Experiment procedure

The study was initiated with a short presentation in which general information and instructions were given. Then, the experiment started with a Demographic Questionnaire to capture the users' backgrounds. The results of this questionnaire are described in Subsection 8.4.1. Afterwards, the task description document was given to the subjects and they started to develop the HiL tasks following the assigned method (the code-centric method or our method). For each activity of the development, they filled in the task description document to capture the development times. After the implementation of the HiL tasks, subjects filled in the post-test questionnaire.

Specifically, the activities carried out in each method were the following:

The code-centric method. First of all, we provided the subjects with a basic guide of the OSGi technology needed to develop the case study following the code-centric development. Then, from the case study requirements specification, they started the implementation of the HiL tasks. Generally, they implemented the classes to support the actions of the HiL task at the different attention levels. Finally, the subjects executed the task and checked its functionality to validate it.

[4] The task description document for the code-centric method can be downloaded from: http://hil.tatami.webs.upv.es/docs/TaskDescriptionCC.pdf

[5] The task description document for our method can be downloaded from: http://hil.tatami.webs.upv.es/docs/TaskDescriptionDM.pdf

Our method. We provided the subjects with a tutorial where the design language and the provided tools were explained. Following our design method (see Subsection 6.3), the subjects first designed the HiL task according to the case study requirements specification. Then, they specified the interaction mechanisms for each attention level. After this step, they completed the implementation of the designed task using our implementation templates, executed them in the simulator, and checked its functionality to validate it.

8.6. Threats to validity

The various threats that could affect the results of this experiment and the measures that we took were the following:

- **Conclusion validity:** This validity is concerned with the relationship between the treatment and the outcome. Our experiment was threatened by the *random heterogeneity of subjects*. This threat appears when some users within a user group have more experience than others. This threat was minimized with a demographic questionnaire that allowed us to evaluate the knowledge and experience of each participant beforehand. This questionnaire revealed that most users had experience in OSGi/Java programming and modeling techniques. This threat was also minimized by providing the subjects with a HiL task example, which helped and guided them in the development of the T5 *Takeover* task. Our experiment was also threatened by the *reliability of measures* threat: objective measures, which can be repeated with the same outcome, are more reliable than subjective measures. In this experiment, the metrics for effectiveness and efficiency are objective, so they not are affected by this threat. However, the metrics for satisfaction are subjective, so they are subject to this threat. Finally, another conclusion validity threat that our evaluation suffered was *validity of the statistical tests applied*. This was resolved by applying the Wilcoxon Signed-rank test, which is one of the most common tests used in the empirical software engineering field. According to Wohlin et al. [78], if we have a sample whose size is less than 30 and we have a factor with two treatments, we can use non-parametric statistical tests such as the Wilcoxon Signed-rank test. The non-parametric tests used in this experiment are detailed in Subsection 8.7.

- **Internal validity:** This validity concern is related to the influences that can affect the factors with respect to causality, without the researcher's knowledge. Our evaluation had the *maturation* threat: the effect that users react differently as time passes (because of boredom or tiredness). We solved this threat by dividing the experiment into different activities and limiting the evaluation to one hour. Another internal validity threat that our evaluation had was *instrumentation*: even though tasks and questionnaires were the same for all subjects, an incorrect interpretation of the task may affect the results. This threat was minimized by the researcher, who helped the subjects to understand the tasks.

- **Construct validity:** Threats to construct validity refer to the extent to which the experiment setting actually reflects the construct under study. Our experiment was threatened by the *hypothesis guessing* threat: when people might try to figure out what the purpose and intended result of the experiment is and they are likely to base their behaviour on their guesses. We minimized this threat by hiding the goal of the experiment. Another threat that appears in our evaluation was *experiment expectations*: people who have done the evaluation can talk to future subjects about their experience. This can bias the results based on what the future subjects expect from the evaluation. This threat was resolved by warning subjects against talking to future subjects.

- **External validity:** This validity concern is related to conditions that limit our ability to generalize the results of the experiment to industrial practice. Our experiment might be affected by *interaction of selection and treatment*: the subject population might not be representative of the population we want to generalize. We used a confidence interval where conclusions were 95% representative. This means that if the conclusions followed a normal distribution, the results would be true 95% of the times the evaluation was repeated. With respect to the use of students as experimental subjects, several authors suggest that the results can be generalized to industrial practitioners [84].

8.7. Data analysis and interpretation of results

The calculated values were checked to determine the p-value (significance level). An important issue is the choice of significance level, which specifies the probability of the result being representative. Generally speaking, the practice dictates rejecting the null hypothesis when the significance level is less than or equal to 0.05 [85]. Thus, 0.05 was established to statistically test the results obtained by the subjects in the experiment. The analysis has performed using the SPSS v.26 statistical tool.

The first step was to analyze the reliability of the data obtained in the experiment: we started by calculating the Cronbach coefficient (alpha). In this case, the result obtained was 0.899. According to [86], if the Cronbach coefficient is greater than or equal to 0.7, then the reliability of the data is assumed. A normality test using the Shapiro–Wilk test was required in order to verify whether or not the data was normally distributed. We used this test as our numerical means of assessing normality because it is more appropriate for small sample sizes (<50 samples). Then, using the Shapiro–Wilk test, we obtained the result that the data is normally distributed. Therefore, we can apply the t-test.

Table 4 shows the descriptive statistics for each variable (PCT, TFT, PEOU, PU, and ITU). The variables are measured for the two treatments: the code-centric method (M1) and our method (M2)[6]. Below, we analyze the results for each research question.

Analyzing effectiveness. According to Table 4, the mean of PCT_M2 (93.63%) is greater than the mean of PCT_M1 (84.09%), that is, the subjects achieved a greater percentage of correctly carried out tasks using our method than when they employed the code-centric method.

Fig. 13 presents the box-and-whisker plot containing the distribution of the PCT variable per method. The median of M2 is slightly greater than the median of M1 since the percentage of correctly carried out tasks achieved by the subjects using our method is greater than the percentage achieved when the subjects use the code-centric method. This means that our method is slightly more effective than the code-centric method when the subjects develop HiL solutions.

In order to check whether the observed differences were significant, we ran the t-test. The result obtained with this test was: 2-tailed p-value = $0.075 > 0.05$. Therefore, we cannot reject the null hypothesis H_{01} and can conclude that there is no difference between the effectiveness of our method and the code-centric method for the development of HiL solutions.

Analyzing efficiency. According to Table 4, the mean of TFT_M1 (51.45) is greater than that of TFT_M2 (36.63), that is, the time required to develop the HiL tasks in the experiment

[6]The raw data can be downloaded from: http://hil.tatami.webs.upv.es/docs/rawdata03022020.sav

Table 4: Descriptive statistics for metrics

Variable	N	Min.	Max.	Mean	Std. Dev.
PCT_M1	11	50	100	84.09	15.30
PCT_M2	11	80	100	93.63	7.10
TFT_M1	11	38	60	51.45	7.62
TFT_M2	11	25	56	36.63	8.41
PEOU_M1	11	2.17	2.83	2.56	0.21
PEOU_M2	11	2.67	4.17	3.63	0.45
PU_M1	11	1.50	2.63	2.14	0.36
PU_M2	11	3.13	4.25	3.95	0.34
ITU_M1	11	1.5	4	2.31	0.68
ITU_M1	11	3	5	4.31	0.64

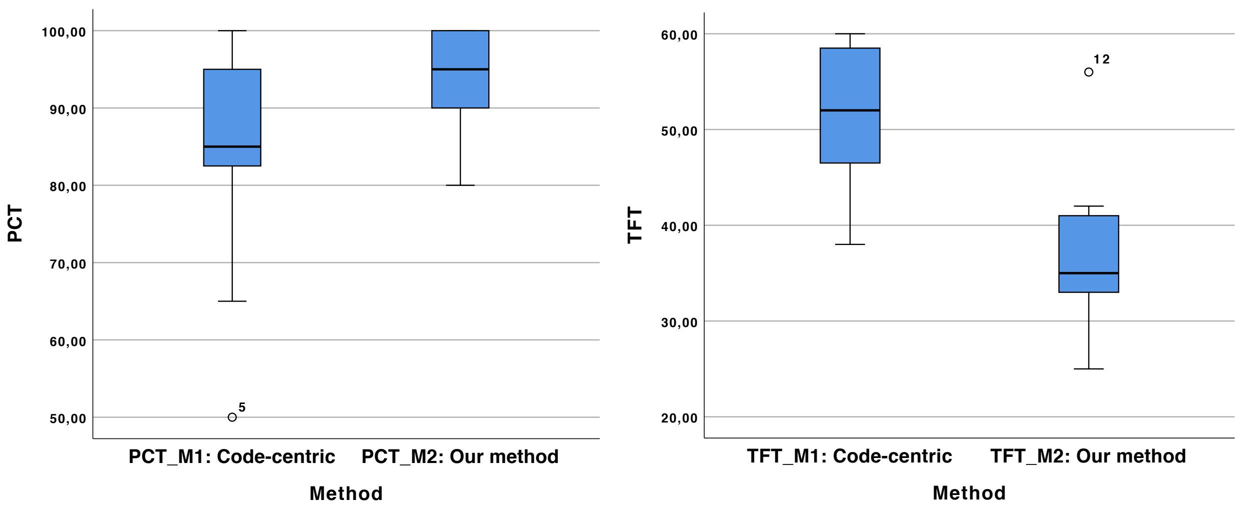

Figure 13: Box-plot of PCT and TFT

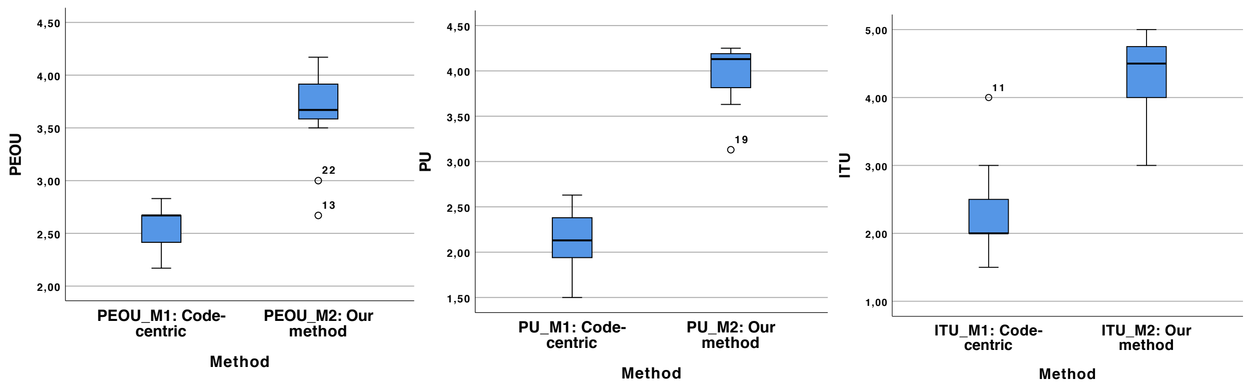

Figure 14: Box-plot of PEOU, PU, and ITU

using the code-centric method was greater than the time needed to perform this task using our method.

Fig. 13 presents the box-and-whisker plot containing the distribution of the TFT variable per method. The medians, first quartile, and third quartile, are better for TFT_M2 since the time needed to conduct the experiment was less when the subjects used our method than when the subjects used the code-centric method. This means that the time to finish the task with our method was better than with the code-centric method.

In order to check whether the observed differences were significant, we ran the t-test. The values obtained with this test are: 2-tailed p-value $= 0.000 < 0.05$. Therefore, we reject the null hypothesis H_{02} and we can conclude that our method is more efficient than the code-centric method for the development of HiL solutions.

Analyzing satisfaction. Satisfaction was defined by measuring the scores given by the subjects according to a Likert scale dealing with three metrics: perceived usefulness (PU), perceived ease of use (PEOU), and intention to use (ITU).

According to Table 4, for all of the satisfaction metrics evaluated, the averages of the results obtained are higher for M2 than for method M1, which indicates better results for satisfaction when using the proposed design method.

Fig. 14 presents the box-and-whisker plot containing the distribution of the PU, PEOU, and ITU variables for each method. It can be observed how, in all cases, the median, the first quartile, and the third quartile for the satisfaction metrics using our method are greater than when using the code-centric method. This means that the subjects expressed greater satisfaction for the three criteria when evaluating our method compared to the code-centric method.

In order to check whether the observed differences were significant, we ran the t-test. The value obtained with this test for the PU variable was: 2-tailed p-value $= 0.000 < 0.05$. Therefore, we reject the null hypothesis H_{03}, and we can conclude that our method is perceived to be easier to use than the code-centric method for the development of HiL solutions.

The value obtained with this test for the PEOU variable was: 2-tailed p-value $= 0.000 < 0.05$. Therefore, we reject the null hypothesis H_{04} and we can conclude that our method is perceived to be more useful than the code-centric method for the development of HiL solutions.

The value obtained with this test for the ITU variable was: 2-tailed p-value $= 0.00 < 0.05$. Therefore, we reject the null hypothesis H_{05} and we can conclude that our method achieves greater intention to be used than the code-centric method for the development of HiL solutions.

8.8. Overall discussion of the results

In summary, the null hypotheses, H_{02} to H_{05}, have been rejected for the variables efficiency and satisfaction, and H_{01} has been accepted for the effectiveness variable. Thus, our method is considered to be more efficient, easier to use, more useful, and to have a greater intention of use than the code-centric method. With regard to the effectiveness, the results show that there is no significant difference between the results obtained when the subjects applied our method and when the subjects applied the code-centric method. We consider that the small difference obtained is because: 1) the subjects used the existing source code (included in the task description document) instead of writing the source code from scratch, as is done in a typical development process; 2) the subjects were not familiar with the design process defined by our approach; 3) the subjects did no have experience in the development of systems of this kind and the requirements imposed by them. Therefore, although the users were not familiar with the design process, this helped them to develop the tasks with fewer problems than with the code-centric method.

It is important to highlight that the differences between our method and a code-centric method arise even when working with simple experimental problems, such as those in our experiment.

9. Conclusions

The research work presented in this paper identifies four design principles that are critical for achieving seamless and solid HiLCPS collaborations. Taking into account these principles, the work defines an approach to specify the co-work between humans and CPSs by means of HiL tasks. The approach focuses mainly on the control strategies and interaction aspects of the HiLCPS collaboration. It provides a set of concepts and a graphic representation for them that allow designers to describe the co-work of humans and systems. Key aspects of the specification are the characteristics related to the attentional resources required to obtain an interaction that *shares the control* between the human and the system, *gets human attention*, *achieves understandability*, and *avoids obtrusiveness*. These attentional resources are used to determine the type of interaction modality that should be used for the HiLCPS collaboration. Then, based on the type of modality, designers select the appropriate concrete interaction mechanisms. This way, our proposal allows designers to describe *what* and *how* HiLCPS interactions must be performed to support HiLCPS collaboration.

The HiLCPS designs that are obtained by applying our approach can easily be put into practice. Fast-prototyping techniques can be used to validate the interaction of the HiL collaboration according to the design principles. The feedback obtained from evaluations of this type can be used to better adjust the models defined at design time. In this work, we apply this technique to validate a prototype of an autonomous car with human collaboration. The validation allows us to check that the design principles are satisfied in the prototype. Once the design has been validated with users, a more detailed design of the human collaboration (or even its implementation) can be carried out. Further validation of the collaboration in real environments would be necessary to verify the complete fulfillment of the design principles and other application domains.

Ongoing work is devoted to providing methodological guidance and tools to apply our approach. We are working on the construction of a set of tools that automate the generation of HiL interaction simulators. These tools provide the designers with prototypes that are generated from their specifications and that allow them to be validated. We also plan to extend the *Core and Feedback Action*

Diagram to include different humans in the horizontal axis to allow several humans to interact with the CPS at the same time.

Acknowledgements

This work has been developed with the financial support of the Spanish State Research Agency and the Generalitat Valenciana under the projects TIN2017-84094-R and AICO/2019/009, and co-financed with ERDF.

References

[1] M. Jirgl, Z. Bradac, P. Fiedler, Human-in-the-loop issue in context of the cyber-physical systems, IFAC-PapersOnLine 51 (6) (2018) 225 – 230.
[2] A. Moore, T. O'Reilly, P. D. Nielsen, K. Fall, Four thought leaders on where the industry is headed, IEEE Software 33 (1) (2016) 36–39. `doi:10.1109/MS.2016.1`.
[3] S. A. Seshia, S. Hu, W. Li, Q. Zhu, Design automation of cyber-physical systems: Challenges, advances, and opportunities, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 36 (9) (2017) 1421–1434.
[4] N. Jaynes, Timeline: The future of driverless cars, from audi to volvo (Aug 2016).
[5] Digital transformation of industries. automotive industry (Jan 2016).
[6] U. Farooq, J. Grudin, Human computer integration, ACM Interactions 23 (6) (2016) 26–32.
[7] S. K. Sowe, E. Simmon, K. Zettsu, F. de Vaulx, I. Bojanova, Cyber-physical-human systems: Putting people in the loop, IT Professional 18 (1) (2016) 10–13. `doi:10.1109/MITP.2016.14`.
[8] C. P. Janssen, S. F. Donkera, D. P. Brumbyb, A. L. Kunc, History and future of human-automation interaction, International Journal of Human-Computer Studies 131 (2019) 99–107.
[9] T. Sheridan, Function allocation: algorithm, alchemy or apostasy?, International Journal of Human-Computer Studies 52 (2000) 203–2016.
[10] E. Hollnagel, A. Bye, Principles for modelling function allocation, International Journal of Human-Computer Studies 52 (2000) 253–265.
[11] J. M. Bindewald, M. E. Miller, G. L. Peterson, A function-to-task process model for adaptive automation system design, International Journal of Human-Computer Studies 72 (2014) 822–834.
[12] P. Wright, A. Dearden, B. Fields, Function allocation: A perspective from studies of work practice, International Journal of Human Computer Studies 52 (2) (2000) 335–355.
[13] M. L. Bolton, E. J. Bass, R. I. Siminiceanu, Using formal verification to evaluate human-automation interaction: A review, IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans 43 (3) (2013) 488–503.
[14] P. Johnson, Tasks and situations: considerations for models and design principles in human computer interaction, HCI International (1999) 1199–1204.
[15] B. E. John, D. E. Kieras, The goms family of user interface analysis techniques: Comparison and contrast, ACM Trans. Comput.-Hum. Interact. 3 (4) (1996) 320–351.
[16] A. Shepherd, Hierarchical task analysis, Taylor and Francis, 2001.
[17] F. Paternò, Concurtasktrees: An engineered approach to model-based design of interactive systems (2002).
[18] S. Stumpf, W.-K. Wong, M. Burnett, V. Pipek, End-user interactions with intelligent and autonomous systems, in: CHI'12, 2012.
[19] J. D. Lee, K. A. See, Trust in automation: Designing for appropriate reliance, Human Factors 46 (50-80).
[20] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, H. P. Beck, The role of trust in automation reliance, International Journal of Human Computer Studies 58 (6) (2003) 697–718.
[21] B. Y. Lim, A. K. Dey, Evaluating intelligibility usage and usefulness in a context-aware application, in: HCI 2013: Human-Computer Interaction. Towards Intelligent and Implicit Interaction, 2013, pp. 92–101.
[22] A. Jameson, User modeling meets usability goals, in: User Modeling 2005, 2005.
[23] D. M. Russell, P. Maglio, R. Dordick, C. Neti, Dealing with ghosts: Managing the user experience of autonomic computing, IBM Sys. Journal 42 (1) (2003) 177–188.
[24] S. Bouzit, G. Calvary, J. Coutaz, D. Chêne, E. Petit, J. Vanderdonckt, Design space exploration of adaptive user interfaces, in: IEEE 11th Int. Conference on Research Challenges in Information Science RCIS, 2017.

[25] C. Miller, R. Parasuraman, Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control, Human factors 49 (2007) 57–75.
[26] P. G. Miriam Gil, V. Pelechano, Personalization for unobtrusive service interaction., Personal Ubiquitous Comput. 16 (2012) 543–561.
[27] B. Rajaonah, N. Tricot, F. oise Anceaux, P. Millot, The role of intervening variables in driver–acc cooperation, International Journal of Human-Computer Studies 66 (2008) 185–197.
[28] R. M. A. van der Heiden, C. P. Janssen, S. F. Donker, L. E. S. Hardeman, K. Mans, J. L. Kenemans, Susceptibility to audio signals during autonomous driving, PLoS ONE 13 (8).
[29] M. Lahmer, C. Glatz, V. C. Seibold, L. L. Chuang, Looming auditory collision warnings for semi-automated driving: An erp study, in: ACM (Ed.), AutomotiveUI, 2018.
[30] D. Thevenin, J. Coutaz, Plasticity of user interfaces: Framework and research agenda, in: INTERACT 1999, 1999.
[31] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, J. Vanderdonckt, A unifying reference framework for multi-target user interfaces, Interacting with Computers 15.
[32] M. Jovanovic, D. Starcevic, Z. Jovanovic, Bridging user context and design models to build adaptive user interfaces, in: International Conference on Human-Centred Software Engineering, 2014, pp. 36–56.
[33] G. Schirner, D. Erdogmus, K. Chowdhury, T. Padir, The future of human-in-the-loop cyber-physical systems, Computer 46 (1) (2013) 36–45.
[34] S. Huang, P. Miranda, Incorporating human intention into self-adaptive systems, in: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Vol. 2, 2015, pp. 571–574.
[35] E. Lloyd, S. Huang, E. Tognoli, Improving human-in-the-loop adaptive systems using brain-computer interaction, in: 2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2017, pp. 163–174.
[36] S. Munir, J. A. Stankovic, C.-J. M. Liang, S. Lin, Cyber physical system challenges for human-in-the-loop control, in: Presented as part of the 8th International Workshop on Feedback Computing, USENIX, San Jose, CA, 2013.
[37] D. S. Nunes, P. Zhang, J. S. Silva, A survey on human-in-the-loop applications towards an internet of all, IEEE Communications Surveys and Tutorials 17 (2) (2015) 944–965.
[38] D. Eskins, W. H. Sanders, The Multiple-Asymmetric-Utility System Model: A Framework for Modeling Cyber-Human Systems, in: Proceedings of the 2011 Eighth International Conference on Quantitative Evaluation of SysTems, IEEE Computer Society, Washington, DC, USA, 2011, pp. 233–242.
[39] J. Cámara, D. Garlan, G. Moreno, B. Schmerl, Evaluating Trade-Offs of Human Involvement in Self-Adaptive Systems, 2017, Ch. 7, pp. 155–180.
[40] W. Li, D. Sadigh, S. S. Sastry, S. A. Seshia, Synthesis for human-in-the-loop control systems, in: International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2014), LNCS, volume 8413, Springer, Berlin, Heidelberg, 2014, pp. 470–484.
[41] C. Wiltsche, L. Humphrey, U. Topcu, Synthesis of human-in-the-loop control protocols for autonomous systems, IEEE Transactions on Automation Science and Engineering 13 (2) (2016) 450–462.
[42] S. won Leigh, H. Agrawal, P. Maes, Robotic symbionts. interweaving human and machine actions, IEEE Pervasive Computing 17 (2).
[43] L. Cranor, A framework for reasoning about the human in the loop., in: UPSEC'08 Conference on Usability, Psychology, and Security, 2008.
[44] C. Dorn, R. N. Taylor, Coupling software architecture and human architecture for collaboration-aware system adaptation, in: Proceedings of the 2013 International Conference on Software Engineering, ICSE '13, IEEE Press, Piscataway, NJ, USA, 2013, pp. 53–62.
[45] M. Krugh, L. Mears, A complementary cyber-human systems framework for industry 4.0 cyber-physical systems, Manufacturing Letters 15 (2018) 89–92.
[46] C. Emmanouilidis, P. Pistofidis, L. Bertoncelj, V. Katsouros, A. Fournaris, C. Koulamas, C. Ruiz-Carcel, Enabling the human in the loop: Linked data and knowledge in industrial cyber-physical systems, Annual Reviews in Control 47 (2019) 249–265.
[47] M. Kuniavsky, Smart Things: Ubiquitous Computing User Experience Design, Morgan Kaufmann, Burlington, MA, 2010.
[48] M. de Sá, L. Carriço, Low-fi prototyping for mobile devices, in: CHI '06: CHI '06 extended abstracts on Human factors in computing systems, ACM, New York, NY, USA, 2006, pp. 694–699.
[49] T. Litman, Autonomous Vehicle Implementation Predictions: Implications for Transport Planning (2013).
[50] J. Muller, No hands, no feet: My unnerving ride in google's driverless car, Forbes.

[51] A. G. Mirnig, M. Gärtner, A. Laminger, A. Meschtscherjakov, S. Trösterer, M. Tscheligi, R. McCall, F. McGee, Control transition interfaces in semiautonomous vehicles: A categorization framework and literature analysis, in: International Conference on Automotive User Interfaces and Interactive Vehicular Applications, 2016.
[52] SAE International, Surface Vehicle Recommended Practice. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, Tech. rep., SAE International (2016).
[53] ISO 9241-210:2019(en), Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems.
URL https://www.iso.org/obp/ui/{#}iso:std:iso:9241:-210:ed-2:v1:en
[54] D. Miller, W. Ju, Joint cognition in automated driving: Combining human and machine intelligence to address novel problems (2015).
[55] M. Gil, M. Albert, J. Fons, V. Pelechano, Designing human-in-the-loop autonomous cyber-physical systems, International Journal of Human-Computer Studies 130 (2019) 21–39.
[56] S. Nahavandi, Trusted autonomy between humans and robots:toward human-on-the-loop in robotics and autonomous systems, IEEE Systems, Man, and Cybernetics Magazine 3 (1) (2017) 10–17.
[57] A. K. Dey, Understanding and using context, Personal Ubiquitous Comput. 5 (1) (2001) 4–7.
[58] E. Horvitz, C. Kadie, T. Paek, D. Hovel, Models of attention in computing and communication: from principles to applications, Commun. ACM 46 (3) (2003) 52–59.
[59] W. Ju, L. Leifer, The design of implicit interactions: Making interactive systems less obnoxious, Design Issues 24 (3) (2008) 72–84.
[60] B. Buxton, Integrating the periphery and context: A new model of telematics, in: Proceedings of Graphics Interface, 1995, pp. 239–246.
[61] N. O. Bernsen, Foundations of multimodal representations: a taxonomy of representational modalities, Interacting with Computers 6 (4) (1994) 347 – 371.
[62] Z. Obrenovic, J. Abascal, D. Starcevic, Universal accessibility as a multimodal design issue, Commun. ACM 50 (5) (2007) 83–88.
[63] S. Lemmelä, A. Vetek, K. Mäkelä, D. Trendafilov, Designing and evaluating multimodal interaction for mobile contexts, in: Proceedings of the 10th international conference on Multimodal interfaces, ICMI '08, ACM, New York, NY, USA, 2008, pp. 265–272.
[64] Y. Cao, M. Theune, A. Nijholt, Modality effects on cognitive load and performance in high-load information presentation, in: Proceedings of the 14th international conference on Intelligent user interfaces, IUI '09, ACM, New York, NY, USA, 2009, pp. 335–344.
[65] L. M. Reeves, J. Lai, J. A. Larson, S. Oviatt, T. S. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin, M. McTear, T. Raman, K. M. Stanney, H. Su, Q. Y. Wang, Guidelines for multimodal user interface design, Commun. ACM 47 (2004) 57–59.
[66] N. Savio, J. Braiterman, Design sketch: The context of mobile interaction, in: Proceedings of MobileHCI 2007, 2007, pp. 248–286.
[67] J. E. Bardram, Distributed interaction, in: Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems, EICS '12, ACM, 2012, pp. 3,4.
[68] F. Beruscha, K. Augsburg, D. Manstetten, R. B. G. Schwieberdingen, Haptic warning signals at the steering wheel: A literature survey regarding lane departure warning systems (short paper) (2011).
[69] Efficacy of haptic blind spot warnings applied through a steering wheel or a seatbelt, Transportation Research Part F: Traffic Psychology and Behaviour 21 (2013) 231 – 241.
[70] M. M. Trivedi, S. Y. Cheng, Holistic sensing and active displays for intelligent driver support systems, Computer 40 (5) (2007) 60–68.
[71] S. Winkler, J. Kazazi, M. Vollrath, Distractive or supportive – how warnings in the head-up display affect drivers' gaze and driving behavior, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015, pp. 1035–1040.
[72] N. Dahlbäck, A. Jönsson, L. Ahrenberg, Wizard of oz studies: Why and how., in: Proceedings of International Conference on Intelligent User Interfaces 93, 1993, pp. 193–200.
[73] V. Basili, G. Caldiera, H. Rombach, Goal Question Metrics paradigm, in: J. Marciniak (Ed.), Encyclopedia of Software Engineering, Wiley, 1994, pp. 528–532.
[74] J. Nielsen, T. K. Landauer, A mathematical model of the finding of usability problems, in: Proceedings of ACM INTERCHI'93 Conference, 1993, pp. 206–213.
[75] N. T. T. Program, https://software.nasa.gov/software/arc-15150-1a.
[76] T. J. Triggs, W. G. Harris, Reaction time of drivers to road stimuli, Human Factors Report No. HFR-12.
[77] B. Kitchenham, L. Pickard, S. L. Pfleeger, Case Studies for Method and Tool Evaluation, IEEE Software 12 (4)

(1995) 52–62.
[78] C. Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, A. Wesslen, Experimentation in Software Engineering : an Introduction, Springer US, 2000.
[79] ISO - ISO/IEC 25062:2006 - Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports.
URL `https://www.iso.org/standard/43046.html`
[80] D. Moody, The method evaluation model: A theoretical model for validating information systems design methods, 2003, pp. 1327–1336.
[81] S. Jamieson, Likert scales: How to (ab) use them, Medical education 38 (2005) 1217–8.
[82] D. Kieras, Using the keystroke-level model to estimate execution times.
[83] S. K. Card, A. Newell, T. P. Moran, The Psychology of Human-Computer Interaction, L. Erlbaum Associates Inc., USA, 1983.
[84] P. Runeson, Using students as experiment subjects – an analysis on graduate and freshmen student data, in: Proceedings 7th International Conference on Empirical Assessment & Evaluation in Software Engineering, 2003, pp. 95–102.
[85] N. Juristo, A. Moreno, Basics of Software Engineering Experimentation, 2001.
[86] M. KD, Applied statistics for software managers.