



**HAL**  
open science

# Teaching Human-Computer Interaction in the Software Engineering Master's Degree Program of the University Grenoble Alpes

Sybille Caffiau, Laurence Nigay

## ► To cite this version:

Sybille Caffiau, Laurence Nigay. Teaching Human-Computer Interaction in the Software Engineering Master's Degree Program of the University Grenoble Alpes. Sense, Feel, Design, 13198, Springer International Publishing, pp.270-278, 2022, Lecture Notes in Computer Science, <10.1007/978-3-030-98388-8\_24>. <hal-03685075>

**HAL Id: hal-03685075**

**<https://hal.science/hal-03685075v1>**

Submitted on 1 Jun 2022

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

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



HAL Authorization

# Teaching Human-Computer Interaction in the Software Engineering Master’s Degree Program of the University Grenoble Alpes

Sybille Caffiau<sup>1</sup>[0000–0003–4222–2018] and Laurence Nigay<sup>1</sup>[0000–0002–4854–626X]

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France  
`FirstName.LastName@univ-grenoble-alpes.fr`

**Abstract.** The training of the Master’s degree in software engineering of the University Grenoble Alpes covers foundational courseware in computer science (programming, complexity, database, networks, interactive systems) during the first year and more advanced engineering courses (in terms of cloud computing, large-scale data management, architecture, program testing and verification) during the second year. This paper focuses on two HCI courses as part of this curriculum in software engineering, and describes the content and the pedagogical approach we implemented for teaching HCI to computer science students. The paper explains why the authors adopt a tool-based approach for the first-year course on engineering HCI and a project-based approach with experimental evaluation for the second-year course on advanced interaction including multimodality.

**Keywords:** Software Engineering · Human-Computer Interaction · User-centered Design · Multimodality · Augmented Reality · Education.

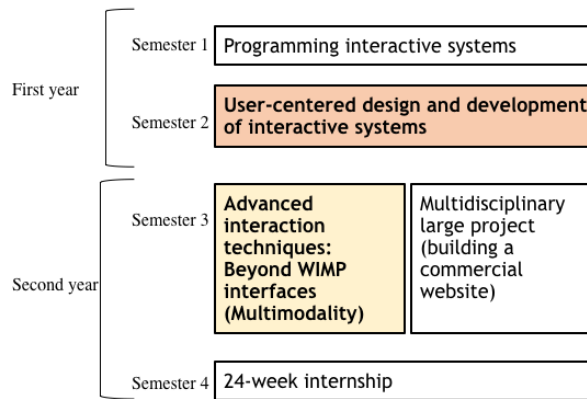
## 1 Introduction

The Master’s degree in software engineering at the University of Grenoble Alpes welcomes students with a Bachelor’s Degree (three years) in either Computer Science or Computer Engineering with courses in computing and practice in programming. This Master’s degree responds to a very strong demand in the job market (including the numerous software firms in the Grenoble region – ATOS, CAPGEMINI, KELKOO, etc.) on software development and management. Nearly all the students (> 95%) obtained permanent employment as developers, architects, analysts and project managers (after few years) before the final defense of their 24-week internship of the second year. Very few students (less than one per year) are starting a PhD after receiving this Master’s diploma.

In this context, as teachers in Human-Computer Interaction (HCI) within a software engineering program, our goal is to train the students to have specific skills related to the design and development of interactive software systems. During the last year (before the 24-week internship), such HCI skills are put into practice as part of a unifying multidisciplinary large project managed by

a group of teachers including an HCI teacher. The goal of the project made by groups of students is to design, develop and evaluate a commercial website. The pedagogical goal is to apply taught concepts, methods and tools of three domains: Agile Software Engineering, Distributed Systems and Human-Computer Interaction (HCI).

The HCI skills are taught through three dedicated courses. One course is on programming interactive systems, including event-based programming, automata, design patterns and toolkits. Because this course focuses on software design and development, it is more easily well perceived by the future software engineers than the two other courses. The paper focuses on these two other HCI courses of the Master’s program that are given after the course on programming interactive systems (Figure 1). One course is given during the first year and focuses on user-centered design and development of interactive systems. The second one is given after the previous one during the second year of the Master’s program: the course focuses on advanced interaction techniques beyond standard graphical user interfaces (WIMP). The identified difficulty for these two courses is to motivate the students to understand the users, their tasks and contexts when using the system that they designed and developed. Based on the teaching experience of the authors, the paper outlines the approaches to overcome this difficulty.



**Fig. 1.** The HCI courses within the Master’s program in software engineering of the University of Grenoble Alpes.

The paper is structured as follows: Section 2 describes the approach for the first year Master’s course on user-centered design and development of interac-

tive systems and Section 3 presents the approach for the course during the second year on advanced interaction techniques beyond WIMP interfaces. Finally, we present concluding remarks on the smooth articulation of these two courses within the Master's program on software engineering.

## 2 First year of the master's program: User-centered design and development

The course is organized along the steps of a user-centered design approach. The goal of this course is to understand the conceptual foundations, models and notations in HCI that are necessary to any practitioner involved in the design, implementation and evaluation of useful and usable interactive systems (Table 1).

For each design step, exercises are performed using software tools when available. For instance, for task modeling, the tool K-MADE is used to produce task models [2], for evaluation, the students use usability testing tools (such as <https://www.optimalworkshop.com> or <https://interfacemetrics.aalto.fi>).

In addition to the exercises to illustrate each part of the course, a project is conducted. 5-student groups have to design a controller to drive a kart in a kart racing game. To design this controller, teachers provide:

- the functional core of a game: a kart racing game. The students must design and develop the controller of SuperTuxKart kart racing game. The set of commands and the output device (a screen) are fixed. The game and its commands are described at <https://ihm2019.afihm.org/challenge.html>;
- Three answers of three potential users (a grandfather, a 7-year-old child and a roommate student) to the question "Do you ever play video games such as kart racing?" to start the user-centered design process. These answers are fictive ones but allow students to define persona and realistic contexts of use. Figure 2 provides an example of these answers.

At the end of the semester the controller may be in the form of a mock-up or a (functional) prototype. Students must explain how they designed it according to the UCD principles and how it is usable for the target user and the target context of use. Figure 3 presents an example of a designed controller.

## 3 Second year of the master's program: Advanced interaction techniques – Multimodality

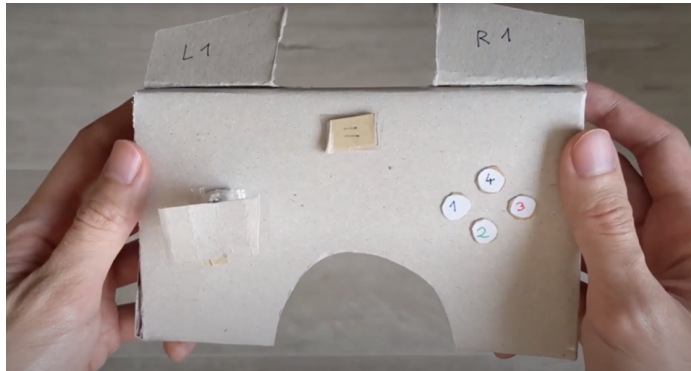
Building on the solid background on user-centered design processes and user interfaces development, this course focuses on engineering advanced interaction techniques, beyond WIMP interfaces. The goal of this course is that the students acquire insight into the ergonomic and software design of multimodal user interfaces. The objectives of the course include:

**Table 1.** Topics of the first-year course of the master's program.

1 Introduction	Domain - History - Definitions - Challenges
2 User-centered Design and the user needs	User-centered design Activity Analysis Persona Design scenarios Collection methods (interviews and observation)
3 Design models and their integration within software engineering life cycles	Task modeling: formalisms and notations Low and high-fidelity mock-ups Dialog models
4 UI Presentation	Exploration and creativity tools Flat design, skeuomorphism design, responsive design Psychology and ergonomics guidelines (Gestalt's laws, Nielsen's heuristics)
5 Evaluation methods of interactive systems	Evaluation dimensions Evaluation by experts (Hick's and Fitt's laws, KLM-GOMS methods) User tests

Yes, but not often. I started because one of my granddaughters, Laura, came to our house during a recent holiday with her video game console. Usually when she comes, I went with her to the park but this time she was a little sick so we opted to stay inside most of the time. So, she spent a lot of time playing with her console. Of course, when I was a kid, I didn't have that, so I wanted to see how it works and what she is doing with it. I am curious. My granddaughter wanted to try but I didn't understand the game she was playing. They're heroes now, you see ... I don't know them. But Laura had a kart racing game and she said to me "OK grandpa, you know how to drive, so you have no excuse" and I couldn't say no: my granddaughter challenged me. I played and ...I lost. I lost because this thing is not the same as driving. But now when Laura comes, she always takes her console to play with me. She is happy because for once she is teaching me something so she is proud.

**Fig. 2.** Example of an answer of a fictive user, used as the starting point for designing the controller of the kart racing game.



**Fig. 3.** Example of a designed controller of the kart racing game.

- Introduction of the key principles and examples of multimodal interaction (including multimodal interaction techniques on mobile devices and augmented reality)
- Presentation of the ergonomic design space of multimodal interaction
- Introduction of the main issues in software design for multimodal interaction.

The course consists of lectures and a group-work project on designing and developing a multimodal interactive system (Table 2).

The topics of Table 2 are only introduced during lectures with supporting documents. Concepts and methods are then applied during a group-work project. To guarantee a user-centered design approach instead of a technology-driven approach, the project is made of two parts. The first part is to design a multimodal interactive system with no development. The second part is to develop and experimentally test a sub-part of the designed system.

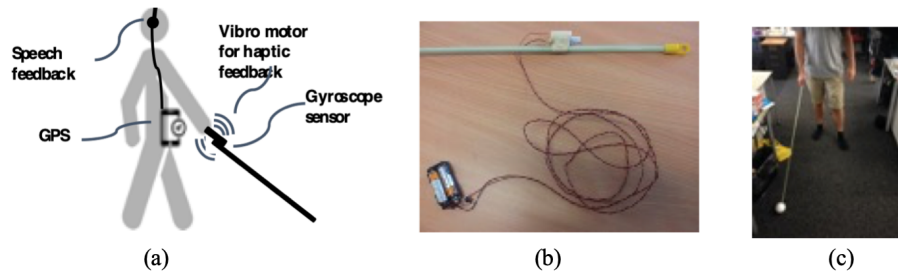
The project starts after the first lecture on definitions and on multimodal application examples. For the project, the students identify an application domain and a potential interaction problem that could require advanced interaction techniques. The identified problem is the motivation of the project and must be based on activity analysis including interviews with users as well as observations. The identified problem is presented by means of activity scenarios from which requirements are identified. By applying a user-centered design approach taught the year before that is enriched by the new design elements on interaction modalities and multimodality, the students design an “ideal project” with projected scenarios as a representation of future activities (Figure 4-a). In addition to scenarios, the steps include a task model, as well as the rationale for the choice of interaction modalities and of their combinations based on CARE diagrams. The “ideal project” is designed independently of the difficulty of developing it.

The second step is to select a specific part of the designed “ideal project” to be implemented and experimentally evaluated with users. Since some modalities may be difficult to implement (hardware and software) and fusion mechanisms difficult to implement, only a subpart of the “ideal project” is implemented within

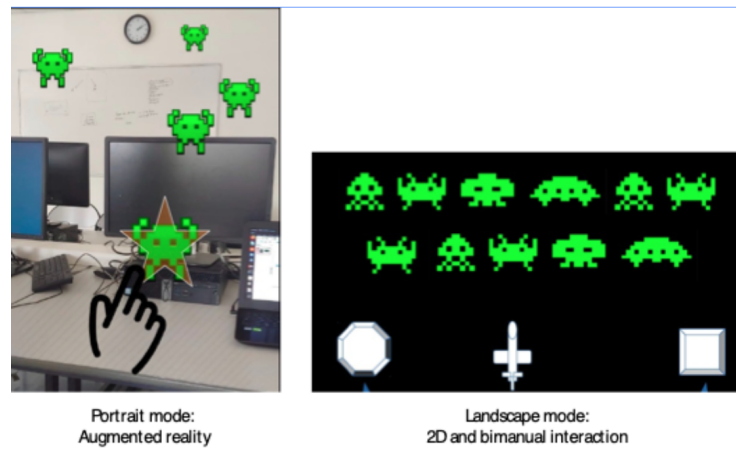
**Table 2.** Topics of the second-year course of the master's program.

1 Introduction	Domain - Definitions - Challenges
2 Multimodal applications	Examples - Application domains illustrated by videos
3 Elements on human perception and action	Why study human perception/action Human modalities Multisensory perception ICS: Interactive Cognitive Sub-systems
4 Ergonomic Design	Underlying concepts MSM Pipe-Line [6] Definition of a modality Design Space Selection of one or several modalities: Actors & Criteria Characterization of a modality Composition of modalities - CARE properties [3] [6]
5 Usage	Empirical results Ten myths of multimodal interaction [7]
6 Software Design	Fusion mechanism Several approach Fusion criteria [4] Software architecture [5] [6] Tools Prototyping tools - Wizard of Oz [9] Component-based approach (ICARE [1], Open-Interface [8])

the time limit of the course (14 weeks). Figures 4 and 5 present two examples of developed projects. Moreover, the developed project can include simulated modalities applying the taught wizard of Oz approach. This wizard of Oz approach has been often used to simulate speech recognition.



**Fig. 4.** Example of a project: (a) “Ideal project” (b) Developed prototype with haptic feedback (c) tested by users.



**Fig. 5.** Example of a developed project on mobile phone: the game has been tested with 11 players on a mobile phone.

## 4 Conclusion

In this paper we described our approach for raising students’ awareness of the importance of the users, the contexts and tasks in the design, development and

evaluation of interactive systems. Two courses are proposed to the Master's degree students.

The first one aims at acquiring fundamental HCI methods and concepts to follow the user-centered design principle. Students have to apply them in a project to design an input interface adapted to specific users and contexts of use to control a kart in a racing game. The specificities of the project are 1) that the functional core is provided and the focus is only on the input user interface 2) that the provided elements to start the user-centered design process are realistic representations of starting points of industrial projects.

Based on this first course, the second course on advanced interaction techniques adopt a complementary approach that is based on the experience gained during the first course. We adopt a project-based learning strategy:

- The starting point of the project is a problem chosen by the students including users' capabilities, context of use for which advanced interaction techniques may be required. So, the students start from the requirement analysis that motivates the project.
- The project follows a user-centered design approach as taught during the first course and enriched with new design elements for multimodality, augmented reality and mobile interaction. The project is fully designed without considering the technical issues of implementing it. It is the first part of the project entitled the "ideal project".
- Because the project involves advanced interaction techniques that are complex to develop and tune, the students develop only a selected part of the "ideal project" that must be experimentally evaluated with representative users. Wizard of Oz techniques are used to enable the experimental study with users when an interaction modality is not available.

The topics of these two HCI courses are not specific to software engineering students. Persona, scenarios, design models, testing methods are taught in other Master's degrees but we have adapted the manner to teach them:

- We use many different application examples to facilitate the generalization/specification of concepts (well-understood process for computer scientists).
- We systematically use software tools to apply the HCI concepts.
- We drive students to question the usability of the interactive systems that they designed during group-work project by adopting the user's point of view.

This overall approach relying on two courses (for a total of 120 hours including 60 hours with the students and 60 hours of personal work) enables us to motivate the students who are future software engineers in industry to learn HCI. One comment from a second-year student "After this course, the way I perceive user interfaces and systems in general is changed".

As HCI teachers, our goal is to train the students to obtain specific skills related to the design and development of interactive software systems as part of an

HCI teaching curriculum for software engineers. We believe that it is important to acquire such HCI skills since these software engineers will develop the future interactive systems of our everyday life.

## References

1. Bouchet, J., Nigay, L., Ganille, T.: Icare software components for rapidly developing multimodal interfaces. In: Proceedings of the 6th international conference on Multimodal interfaces. pp. 251–258 (2004)
2. Caffiau, S., Scapin, D., Girard, P., Baron, M., Jambon, F.: Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interacting with Computers* **22**(6), 569–593 (2010)
3. Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., Young, R.M.: Four easy pieces for assessing the usability of multimodal interaction: the care properties. In: *Human—Computer Interaction*, pp. 115–120. Springer (1995)
4. Lalanne, D., Nigay, L., Palanque, P., Robinson, P., Vanderdonckt, J., Ladry, J.F.: Fusion engines for multimodal input: a survey. In: Proceedings of the 2009 international conference on Multimodal interfaces. pp. 153–160 (2009)
5. Nigay, L., Coutaz, J.: A generic platform for addressing the multimodal challenge. In: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 98–105 (1995)
6. Nigay, L., Coutaz, J.: Multifeature systems: The care properties and their impact on software design. *Intelligence and multimodality in multimedia interfaces* (1997)
7. Oviatt, S.: Ten myths of multimodal interaction. *Communications of the ACM* **42**(11), 74–81 (1999)
8. Serrano, M., Juras, D., Nigay, L.: A three-dimensional characterization space of software components for rapidly developing multimodal interfaces. In: Proceedings of the 10th international conference on Multimodal interfaces. pp. 149–156 (2008)
9. Serrano, M., Nigay, L.: Temporal aspects of care-based multimodal fusion: from a fusion mechanism to composition components and woz components. In: Proceedings of the 2009 international conference on Multimodal interfaces. pp. 177–184 (2009)