

 Open access • Proceedings Article • DOI:10.1145/2820619.2825011

Teaching task analysis for user interface design: lessons learned from three pilot studies — [Source link](#)

[Káthia Marçal de Oliveira](#), [Patrick Girard](#), [Taisa Guidini Gonçalves](#), [Sophie Lepreux](#) ...+1 more authors

Institutions: [Centre national de la recherche scientifique](#)

Published on: 27 Oct 2015 - [l'Interaction Homme-Machine](#)

Topics: [Task analysis](#) and [User interface design](#)

Related papers:

- [Designing User Interfaces from Analyses of Users' Tasks](#)
- [Designing effective tasks for digital library user tests: lessons learned](#)
- [An interactive multimedia tutorial for user interface design](#)
- [Drawing on the Right Side of the Brain](#)
- [Variations of a theme: card-based techniques for participatory analysis and design](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/teaching-task-analysis-for-user-interface-design-lessons-4hggcju6en>



HAL
open science

Teaching Task Analysis for User Interface Design: Lessons Learned from Three Pilot Studies

Káthia Marçal de Oliveira, Patrick Girard, Taisa Guidini Gonçalves, Sophie
Lepreux, Christophe Kolski

► **To cite this version:**

Káthia Marçal de Oliveira, Patrick Girard, Taisa Guidini Gonçalves, Sophie Lepreux, Christophe Kolski. Teaching Task Analysis for User Interface Design: Lessons Learned from Three Pilot Studies. 27ème conférence francophone sur l'Interaction Homme-Machine., Oct 2015, Toulouse, France. pp.w10, 10.1145/2820619.2825011 . hal-01219093

HAL Id: hal-01219093

<https://hal.archives-ouvertes.fr/hal-01219093>

Submitted on 26 Oct 2015

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.

Teaching Task Analysis for User Interface Design: Lessons Learned from Three Pilot Studies

Káthia Marçal de Oliveira

UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
kathia.oliveira@univ-
valenciennes.fr

Patrick Girard

LIAS / ISAE-ESMA
1 rue Clément Ader, BP 40109,
86961 Futuroscope – France
patrick.girard@univ-poitiers.fr

Taísa Guidini Gonçalves

UVHC, LAMIH CNRS UMR 8201
59313 Valenciennes, France
taisa.guidinigoncalves@etu.univ-
valenciennes.fr

Sophie Lepreux

UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
sophie.lepreux@univ-
valenciennes.fr

Christophe Kolski

UVHC, LAMIH CNRS UMR 8201
Le Mont Houy cedex 9
59313 Valenciennes, France
christophe.kolski@univ-
valenciennes.fr

Abstract

Task analysis is recognized by the Human-Computer Interaction community as good practice to improve the understanding of how a user may interact with software interfaces to reach a given goal. During more than one decade, we have taught task analysis in undergraduate and graduate HCI programs for the design of better interactive systems. In this paper, we describe three ways of teaching task analysis and the lessons learned from those practices. We consider this the first step of a larger study looking to improve task analysis education effectiveness.

Keywords

Task analysis, HCI Education, user interface design.

Keywords - ACM classification

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

Introduction

The Human-Computer Interaction (HCI) community has recently defended the importance of task analysis as one of the most important activities to be performed in order to ensure user-centered design [4]. Task analysis has been therefore used in various phases of design and evaluation of interactive systems [12].

Convinced of the importance of task analysis, we have been teaching this practice in undergraduate and graduate courses focusing on its integration within

In general, what is the difficulty for users to understand the tasks to be performed in this model?

Very difficult

1	2	3	4	5
---	---	---	---	---

 Very easy

What is your opinion about the grouping of tasks proposed in the model?

Very bad grouping

1	2	3	4	5
---	---	---	---	---

 Excellent grouping

Are the terms used all the time with the same meaning?

Never

1	2	3	4	5
---	---	---	---	---

 Always

What is the maximum number of nodes from the high abstract task to the primitive task?

--

What is the minimum number of nodes from the high abstract task to the primitive task?

--

Figure 1. Extract of evaluation questions for the task models

software engineering activities for years. This paper reports the difficulties we found and the lessons we learned in three studies performed by different HCI teachers, who worked independently.

In the next section we describe task analysis in the context of UI design. Then, we present the three studies, and we conclude by a discussion.

Task Analysis for user interface design

According to [5], task models are “logical descriptions of activities that are designed to be carried out in reaching user’s goals in an interactive system.” In general, task analysis methods involve three main elements [10]: (1) a stepwise approach, in which a sequence of steps is used to work on models; (2) models that capture some facets of the problem and translate them into specifications; and (3) software tools to support the approach.

Several formalisms are available to implement task analysis [11]. We can quote for instance: Hierarchical Task Analysis (HTA) [1], ConcurTaskTrees (CTT) [12], Model for Activity Description (MAD) [13]. Task models may be represented at different levels of abstraction [17]. Typically, task models are detailed in small level of granularity for the design of final user interface. However, it can also be used to specify the requirements of how activities should be performed, considering only high-level tasks.

First study: Teaching the basics of task analysis in undergraduate courses

We have been teaching task analysis in the computer science undergraduate course at University of Valenciennes (UVHC) since 2005. Teaching task analysis is not isolated from other courses, but integrated in the “system design” course along with the

UML course (and specifically use case diagrams approach).

One of our teaching experiences lied in the design and evaluation of UI. We divided the group of 43 students in two subgroups, here after named Group A and B. Each group was given a separate system description. At first we asked all students to perform the task analysis and design using CTTE. Two teachers reviewed the task models using an inspection form based on quality criteria (such as functional understandability [6], minimal action feedback grouping/distinction of items [14] and so on). Figure 1 presents a partial example of this form. The students had to correct their task models based on the reviews.

Afterwards, they were asked to produce mockups using their task models. To that end, we set as one of the criteria for the final note that the mockups corresponded to what was modeled. Finally, we defined an evaluation session where Group B evaluated the mockups produced by Group A and vice versa. This evaluation was also guided by a questionnaire based on the same quality criteria (see some examples in Figure 2). We had two goals with this teaching application: (1) to verify whether the students produced their mockups using the previous task models, and (2) to verify whether we found some correlations between the early evaluation performed in the task models and the final evaluation (using the mockups considering the same criteria).

Concerning our first goal, we verified manually the mockups based on the task models. We concluded that the students, in general, followed the same organization of tasks but not in all details (particularly for the definition of system tasks and sequence of

In general, what is the difficulty for users to understand the tasks to be performed in this application?

Very difficult

1	2	3	4	5
---	---	---	---	---

 Very easy

How would you rate the effort to access a functionality that you want to execute?

Very difficult

1	2	3	4	5
---	---	---	---	---

 Very easy

Are the terms used all the time with the same meaning?

Never

1	2	3	4	5
---	---	---	---	---

 Always

In general, what do you think about the navigability (possibility to go forward and backward in the system)?

Very difficult

1	2	3	4	5
---	---	---	---	---

 Very easy

Figure 2. Example of evaluation questions for the UI

abstract tasks). For the second goal, we did not find significant correlations between the evaluation criteria from task models and final mockups, unlike expected [10]. We supposed that the reason for these results is the fact that the students were still learning task analysis and also that several modifications were performed while producing the mockup considering the facilities of the used language. Nevertheless, we could attest the importance of performing evaluation throughout the task analysis and design using the inspection forms. We therefore defined a checklist to be used by the students after task modeling and before prototyping. The students have been authorized to correct the design models while designing the mockups.

From that and other experiences of teaching task analysis and *mockup* design, we conclude that the importance of task modeling is questioned by the students, since the mockups are more direct, easy to produce with the new languages and explain better the UI requirements.

Second study: Teaching task analysis for requirement specification in graduate level

Our second study was performed in the context of master program in Automation & Human-Machine Systems at the University of Valenciennes. A course of HCI has been offered in this program (1994 to 2003; 2010 to today). The main objective is to sensitize students about the importance of specification, design and evaluation issues related to HCI. After 12 hours of classes (9 hours of plenary classes and 3 hours of supervised exercise classes including task modeling using CTT and HTA), students received a description of a real problem involving an industrial process with interconnected mixing stations that need a supervisory

HCI (6 different profiles of users potentially concerned, human factors, security and production aspects to consider). They were asked to perform the requirement specification considering HCI issues (during 4 sessions of 1h 30 in presence of the teacher, completed by personal work). The students were organized in groups of 3-4 members in a competitive context (project-based learning approach). They had no or little prior knowledge of the domain application. As a result of the course they should provide a detailed report (20 to 30 pages) with the specification of the proposed system.

A first analysis considered the reports of the students from nine cohorts within this course (1994/1995 – 2002/2003). In this analysis, some criteria directly related to the HCI specification of the specific problem were discussed, such as issues relating to the room layout, graphics and representation of the information on the screen. Sixty-three project reports were analyzed [8]. In general, we concluded that the HCI specifications were very rich, allowing the identification of design errors, the presence of design creativity by the students and the great diversity of the solutions.

Since 2010, we included a specific criteria related to task analysis, focused on task modeling in the report evaluation. We analyzed 21 project reports from 2010/2011 – 2014/2015. Our first goal was to answer the following question: *Did the students perform task modeling for the specification of final user tasks?* To answer this question we considered a 7-point Likert scale from (-1) task modeling “not considered” to (5) task modeling very well defined (i.e. *very good*). Figure 3 presents the results of our findings. We note that 11 of the reports did not present any description generated from a task analysis.

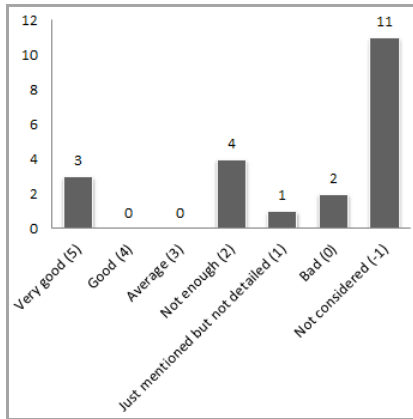


Figure 3. Results about task modeling

We considered as task modeling any specification that describe the final user tasks. In this context we found different approaches used by students, from informal specifications (not using a method) to the use of task analysis methods or modeling languages (e.g., UML). As informal specification we can quote the use of simple list of task description about final user tasks, flow of tasks execution, global tree of tasks and informal students also used some classical methods, such as: MCT (Modèle Conceptuel des Traitements, from Merise) [15], HTA [1], and CTT [12]. Figure 4 presents the findings about the use of some approach for documentation of task analysis activity.

We note that from the 10/21 reports that presented some task modeling specification, 6/10 used an informal approach and 4/10 used some classical approach for task modeling. After this general analysis, we performed a detailed evaluation of the 10/21 reports that used some classical method for task modeling or informal description of interactive tasks, looking to define the level of detail of the task model used.

In this evaluation, we classified the modeling in global modeling or detailed modeling. For instance, when using CTT we considered global modeling when the report presented only a high level of task tree (abstract tasks) without defining the primitive tasks. Detailed modeling, on the other hand, considers several levels of abstraction in the task tree going up to the definition of primitive tasks. From the 10 reports, 4 presented a detailed task model and 6 presented a global task model. Finally, we note that about half of the reports

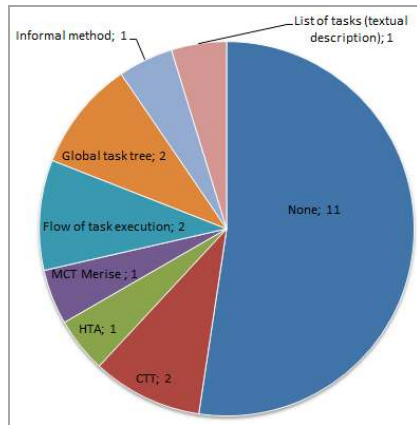


Figure 4. Results about the use of task modeling approaches

11 did not present any documentation of the task analysis activity (model or informal description).

In general, from this experience we noted that although the students are trained in the use of task analysis and design for the system specification, they do not use the methods studied during the course: no method used by some groups, informal approaches by others, and only few groups used "classical" methods.

Third study: Teaching task analysis for requirement validation in post-graduate courses

We report in this section an experience of teaching task modeling at the University of Poitiers. For more than 12 years, this section has been included in the HCI course for graduates, with two kinds of students (20-30 computer scientists versus 30-40 double competency students), in a software engineering perspective. Three main objectives were given to the course: discovering the major topics of Human-Computer Interaction, discovering and programming enhanced interaction, and understanding the major role of users in human-centered design. This was, and still is a tool-based course, firstly with CTTE, and now with K-MADe.

During many years, teaching task analysis was very frustrating: even if task notations were correctly mastered (CTT, K-MAD), the global feedback of students focused mainly on the usability problems of the tools: they did not understand how they could use them in their projects... A good example can be seen in the exploration of how advanced topics, such as object domains and expressions in models, were introduced as reported in [3].

A few years ago, the course goals were revised, in order to reach a simpler objective: can we convince students to use task modeling for the better design of interactive systems? Using task modeling in this perspective appears in three course sections: (1) understanding the basics of task analysis, with examples of activities such as booking a ticket, or travelling by train (2) introducing task analysis in the software engineering cycle for specifying systems (3) reusing task analysis in a project-based sequence (agile method). In this paper, we focus on points 2 and 3.

Task analysis for specifying

Once students are able to efficiently use the task model notation, our goal is to teach them how to use it in a concrete design phase. We work on two case studies, with a same objective: designing a new post-WIMP version of an existing system. Task modeling is used to describe the activity on the reference system, and the expected activity on the future system. Students are asked to explain how their proposal fits better the user needs, leaning on task analysis, eventually using the task model simulator *Prototask*¹ to validate their hypotheses.

Using task analysis in projects.

During the next phase, students are supposed to chose what methods and tools they could use to design and develop a prototype, discovering the SCRUM agile method at the same time.

Feedback

This concrete approach proved useful: by the use of *Prototask* most students found the task model notation useful to focus on some aspects of interactive systems.

¹ <http://www.lias-lab.fr/forge/projects/prototask>

Nevertheless, this result is quite weak: one group (40 students) completely missed the objective. While they were able to build correct task models, and they seemed to understand the role of task models in the global process, only 2 of 40 used task analysis during the third phase.

Discussion and future work

In the previous section we described different and independent experiences of teaching task analysis for different professors (see Table 1). These studies presented analysis from specific projects and general observation from years of teaching. From these preliminary results, we can observe that:

- Most students are not convinced by the utility of using task analysis in software design. Usually, they produce models only when forced to do so.
- They do not find a real added value in using task models, even if their benefits are highlighted.
- Even if they know how to build syntactically-correct task models, they do not really exploit their semantics, or how to use it in new design.

We believe that the main reason for these results is that the students see task analysis as an additional charge in their project without estimating the positive impacts. These findings differ from the only study [2] we found about task modeling education (our previous study was about HCI specification in general [8]). In [2] they identified that bioinformatics students found useful the use of task models. Nevertheless, the authors recognize that for better conclusions their study should be performed with other student's profiles, such as computer scientists and human engineers. In this context, some major questions emerge: (1) are students really able to integrate models from different origins, such as software

Table 1. Summary of the studies

Item	Study		
	1 st	2 nd	3 rd
University	UVHC	UVHC	Poitiers
University program	Under-grad.	Grad.	Grad.
Number of students	43	63 groups of 3-4 students	50-70
Task modeling formalism	CTT	CTT, HTA	K-MAD
Presented analysis	specific	specific	general

engineering models (UML) and HCI models (task models)? (2) How can we convince them to use task analysis in their professional activity?

To answer these questions, we propose to plan a controlled experiment to identify the actual reasons of this behavior since we believe the main limitation of the

References

- [1] Annett, J. Hierarchical Task Analysis. In Hollnagel (Ed.), *Handbook of Cognitive Task Design* (pp. 17-35), Lawrence Erlbaum Associates, New Jersey. (2003).
- [2] Caffiau S., Scapin D., Sanou L., Guidet L., Using K-MADe for learning task modeling : Interest and Difficulties. *Journal d'Interaction Personne-Système*, vol 1., n1., 4:1-4:28. (2010)
- [3] Caffiau, S., Scapin, D., Girard, P., Baron, M., and Jambon, F. Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interacting with Computers*, 22(6), (2010), 569-593.
- [4] Diaper, D., Stanton, N. A. (Eds). *The Handbook of Task Analysis for Human-Computer Interaction*. CRC Press, (2004).
- [5] Guerrero, J., Lemaigre, C., Gonzalez, J.M. and Vanderdonckt, J. Model-Driven Approach to Design User Interfaces for Workflow IS. *Journal of Universal Computer Science*, 14 (19) (2008), 3160-3173.
- [6] ISO/IEC 25023. *Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality* (2011).
- [7] Kolski C. (Ed.), *Analyse et Conception de l'IHM. Interaction Homme-machine pour les SI*, Vol. 1, Hermès, Paris, (2001).
- [8] Kolski, C., Loslever, P., and Sagar, M. The performance of future designers on the specification of supervisory HCI: case study of a simulated work situation. *Cogn Tech Work*, (2012), 14, 107-128.
- studies is that they were not planned to answer specific stated hypothesis to be investigated. Moreover, we are inviting other colleagues to report their experience in task model education, in the hope to propose accurate teaching methods to enhance the effectiveness of task model teaching.
- [9] Limbourg, Q., and Vanderdonckt, J. Comparing Task Model for User Interface Design. In [4], 135-154.
- [10] Oliveira, K., Lepreux, S., Kolski, C., and Seffah, A. Predictive usability evaluation: aligning HCI and software engineering practices. *IHM'14*, (2014), 177-182.
- [11] Palanque, P., Bastide, R. Synergistic Modelling of Tasks, Users and Systems Using Formal Specification Techniques. *Interacting With Computers* 9, no. 2 (1997), 129-53.
- [12] Paternò, F. *Model-based design and evaluation of interactive applications*. Springer, (1999).
- [13] Scapin, D. L., and Bastien, J. M. C. Analyse des tâches et aide ergonomique à la conception : l'approche MAD*. In [7], 85-116.
- [14] Scapin, D.L., and Bastien, J.M.C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information Technology*, v. 16, n.4 (1997), 220-231.
- [15] Tardieu, H. Rochfeld, A., Colletti, R. *La méthode Merise : Principes et outils*, Paris, Éditions d'organisation, 1983.
- [16] Theofanos, M., Stanton, B., and Bevan, N. A. Practical Guide to the CIF: Usability Measurements, *ACM Interactions*, Vol. 8, No. 6, (2006), 34-37.
- [17] Vanderdonckt, J., Tarby, J.C., and Derycke, A. Using Data Flow Diagrams for Supporting Task Models. In *Proc. of 5th Eurographics Workshop on DSVIS* (1998).