HAL
open science

Using K-MADe for learning task modeling: interests and difficulties

Sybille Caffiau, Dominique L. Scapin, Loé Sanou, Laurent Guittet

To cite this version:

Sybille Caffiau, Dominique L. Scapin, Loé Sanou, Laurent Guittet. Using K-MADe for learning task modeling: interests and difficulties. Journal d'Interaction Personne-Système, 2014, Volume 1 (1), pp.1-28. 10.46298/jips.60 . hal-01058930

HAL Id: hal-01058930

https://inria.hal.science/hal-01058930v1

Submitted on 28 Aug 2014

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.

Using K-MADe for Learning Task Modeling: Interests and Difficulties

SYBILLE CAFFIAU
LISI, ENSMA Futuroscope Chasseneuil
DOMINIQUE L. SCAPIN
INRIA Domaine de Voluceau
LOE SANOU
LAURENT GUITTET
LISI, ENSMA Futuroscope Chasseneuil

Résumé : L'enseignement de la conception de logiciel centrée utilisateur concerne de nombreux aspects. Un de ces aspects les plus importants est la modélisation des tâches. La modélisation des tâches contribuant fortement à la qualité ergonomique et à l'acceptation du logiciel résultant, il est indispensable qu'elle soit la mieux comprise possible par les futurs concepteurs de logiciels. Dans ce but, cette étude cherche à évaluer un modèle de tâche particulier (K-MAD) et son outil associé (K-MADe) dans une démarche d'enseignement de la modélisation des tâches. Cet article décrit la problématique de l'étude, le modèle et le logiciel enseigné, les différentes phases de l'enseignement, et le retour d'expérience d'utilisation concrète du logiciel. Ce retour d'expérience identifie les apports et les inconvénients liés à cette utilisation, lors de son enseignement, et propose des pistes d'évolution des modèles obtenus et d'édition de ces modèles.

Mots clés : Interaction Homme-Machine, Conception centrée-utilisateur, Modèles de tâches, Analyse hiérarchique des tâches, Outils de modélisation, Enseignement, Évaluation.

Abstract: Teaching user-centred software design covers many aspects. One of the most important ones is task modeling. Since task modeling contributes largely to the ergonomic quality and acceptance of the resulting software, it is essential for task modeling concepts to be well understood by future software designers. To this end, this study aims at evaluating a task model (K-MAD) and its associated tool (K-MADe) as regards the task modeling training. This article describes the rationale of the study, the model and the software tool used, the teaching steps, and the experience feedback on the practical use of the software. The latter identifies the benefits and disadvantages of this practical use from the teaching point of view, but also from a practical standpoint, including in terms of evolution of the models obtained and of their edition.

Key words: Human-Computer Interaction, User-centered design, Task model, Hierarchical task analysis, Modelled tools, Teaching, Evaluation.

Les articles de JIPS sont publiés sous licence Creative Commons Paternité 2.0 Générique.

1. INTRODUCTION

Nowadays, taking the user into account is an essential need for interactive application design [Diaper 2004]. Satisfying this need can be achieved partly by offering appropriate methods for training students (future HCI designers) particularly in the area of task modeling. Students have been taught how to model the human activities for several years. However, the design of task models that express activities within software application is a complex task. Moreover, the duration usually assigned to this specific training is limited. These two reasons require looking for techniques that are both non time-consuming and pedagogically beneficial for students.

In order to conduct such teaching activities, dedicated software has been used for several years in order to support task model training in our university. Among the user-oriented task models that are supported by freely available tools, two models have been used: CTT [Paternò, et al. 1997] (with CTTE as the tool); and since 2007, K-MAD [Lucquiaud 2005] (with K-MADe as the tool [Baron, et al. 2006]). The goal of this paper is neither to explain this choice, nor to compare the use of these two formalisms but to present an assessment of the use of one particular task model tool to support the learning of task modeling. This study was conducted using the K-MADe software. The main reason for that choice is that K-MAD is a task model *kernel* composed of all main concepts essential to design a complete task model (i.e., common to most existing task models). The other reason is that current research on K-MADe is the assessing of the model (e.g. according to new application domains) and the developing of the tool. Thus, this study, through the assessment of the task modeling learning process, aims at producing data useful for improving the current support software.

In this paper, the first section describes the model being taught as well as its associated tool. The following sections present the goals of the study, the participants, the experimental phases, and the evaluation method. Then, the results are presented with data collection and analysis. Finally, the conclusion summarizes the evaluation and identifies future work for the evolution of K-MADe.

2. THE MODEL AND THE TOOL

This first section describes the K-MAD model and the K-MADe tool, which were used during the learning sessions.

2.1 The K-MAD model

The K-MAD task model is hierarchical. It represents the user's activity as a tree of tasks, from the most general level (root) to the most detailed level (elementary actions). In order to express activities, the K-MAD model consists of components and scheduling characteristics.

In the K-MAD model, a task is identified with a name, a goal and an identification number. In addition, a task is characterized by several elements: duration, feedback, executant (user, system, interactive or abstract), modality (when it is user-performed), frequency, importance level, triggered events and additional comments, when applicable.

The task scheduling characteristics are: interruptability, mandatory (or not), iteration (number or expression that sets the number of times the task is performed), pre condition, event (which constrains the task execution) and object modifications, named post conditions (which express the object dynamics).

The objects that are defined are used to characterize the pre conditions, the iterations and the object manipulation expressions.

The scheduling decomposition of parent tasks is also specified. The task schedules its subtasks either sequentially, concurrently, alternatively or without any order. When a task is not split further, it is defined as an elementary task.

2.2 The K-MADe Tool

K-MADe [Baron, et al. 2006] allows to describe, modify and query the task models (queries are under development).

K-MADe implements the above characteristics of the model as well as additional characteristics such as the *unknown* executant type for more tool flexibility.

Although all kinds of profiles can use this tool, it is particularly intended for ergonomists and specialists in HCI. As there are many skill variations in such populations, the tool allows different levels of description, from simple graphical drawings to detailed mathematical expressions using several available tools:

- A graphic editor of the K-MAD task model, which uses direct manipulation techniques for constructing, handling and canceling tasks (label 1 in Figure 1).
- A task characteristic editor. The editor has three forms. A reduced form with only the main task characteristics; (Figure 2 a) a form which lists all task characteristics in tables (label 2 in Figure 1); and finally a printable form with all task characteristics (Figure 2 b).
- An editor of abstract objects, users, events and concrete objects. Objects can be added, modified and removed. The modification and the removal of an object imply modifications of all associated objects. *Sheets* (label 3 in Figure 1) allow to access these object definition spaces.
- A tool for expressing the pre conditions, post conditions and iterations. The tool checks whether the grammar of an expression is correct or not (Figure 3; and later on Figure 11).
- A simulator for running and animating models of tasks (Figure 8).
- Various tools for analyzing task models (statistical (Figure 7a), coherence (Figure 7b), queries,…).
- A tool for printing task trees and detailed task characteristics.

Tools to check the task model grammar (Figure 7b), to simulate (Figure 8) and to analyze the task models are triggered from the *tool* menu (encircled in Figure 1). More information about the K-MADe tool may be found in the user manual [K-MADe].

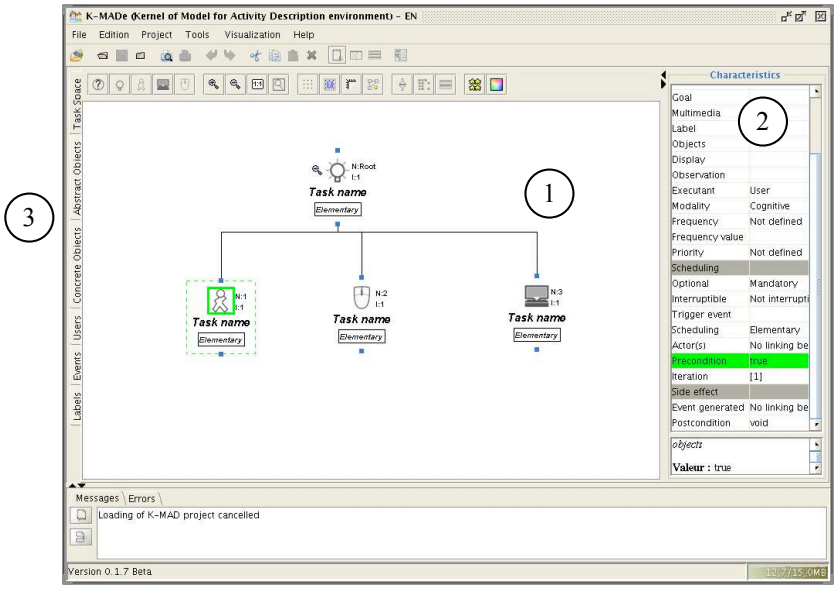


Fig. 1. The main window of K-MADe tool

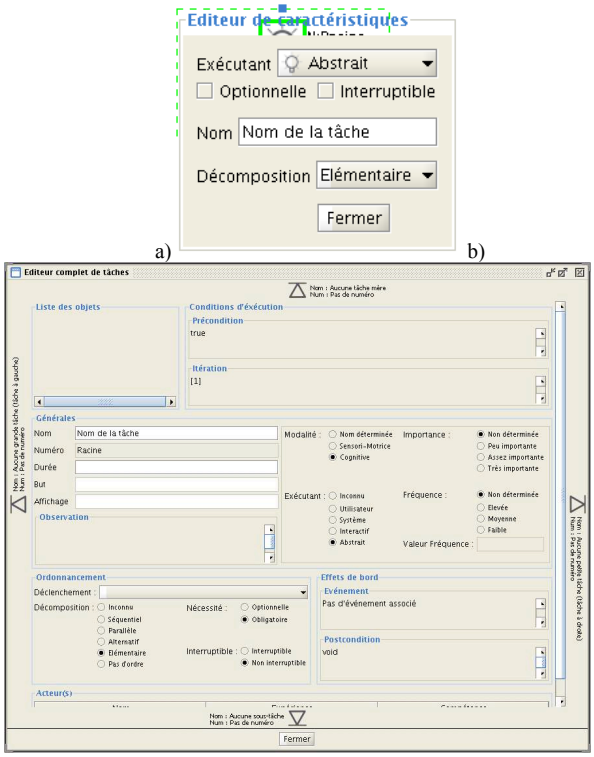


Fig. 2. Two other task editors

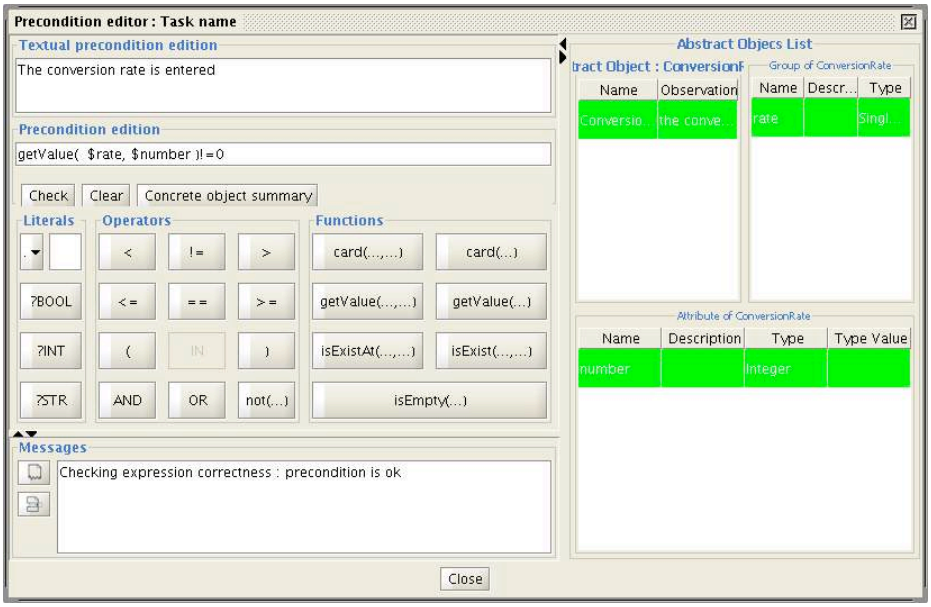


Fig. 3. The window to edit a task precondition

3. OBJECTIVES

Data gathered in this evaluation aims at analyzing the role played by the K-MADe tool during the K-MAD model learning process. We identified three steps that characterize the learning process: the *modeling discovery*, the *model appropriation* and, once the model is learned, the *model use* via K-MADe (the tool).

For the *modeling discovery* phase, we wished to analyze the students' motivation and willingness to use the software when performing task modeling. We then wished to study the consequences of using K-MADe on the model appropriation in terms of concepts. In particular, the issue was whether the software adds complexity to the task analysis, or helps with the learning.

Lastly, once students learned the model, we wanted to evaluate how they use K-MADe in order to design their own task models; which K-MADe functions are used; how they define concepts (when, what and why), which ones are assimilated and to what extent K-MADe does help in that regard.

4. PARTICIPANTS

Two groups of students took part in the study. All subjects were students in their fourth year at a French university. They were all bio-informatics students (Master Génie Physiologique et Informatique: http://www-gphy.univ-poitiers.fr). The first group was composed of the 48 students (31 males and 17 females) from the 2007/2008 school-year, and the second one of the 50 students of the 2008/2009-school-year (28 males and 22 females). Participants were aged from 21 to 25 years old. Both years were taught the same HCI program. Participant characteristics are presented in Table I.

Table I. Participant characteristics

	2007/2008 school-year group	2008/2009 schoole-year group
Male	31	28
Female	17	22
Number	48	50
Formation	bio-informatics students in 4th year at a French University	
Aged	from 21 to 25 years old	

5. STUDY SCHEDULE

The study of the K-MADe role in the learning of task modeling took place in two years. Each year, the HCI course was composed of two phases: the lecture and the pratical sessions.

The lecture part presented the basic principles of user-centered design and task modeling. During this lecture, the K-MAD model was detailed and illustrated with K-MADe tool captures. Then, during the practice courses, the students used the K-MADe tool to express several activities (i.e. sets of tasks). Each year, students followed two practice training sessions. In addition, in the first year, a session was added in order to perform a specific evaluation step. During this session, students were observed when they practiced task modeling without any help.

Figure 4 summarizes the different sessions for each group according to the task modeling steps and the evaluation year: the initial lecture (Figure 4), the practicing following the HCI course composed of training course sessions (Figure 4) and the *usage* evaluation session (Figure 4).

Evaluation data was only collected in practice course sessions. In terms of timing, the evaluations of both student groups' studies were one year distant: the contents of the second-year learning phase was modified in order to complete the first year data.

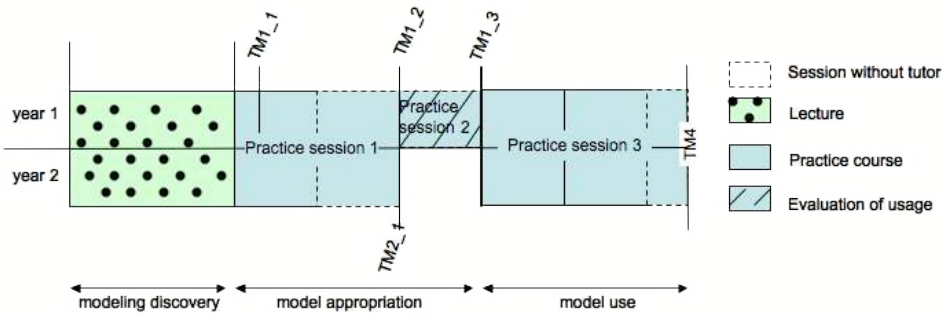


Fig. 4. Schedule of task model learning steps

5.1 The Lecture

The HCI course focused on user-centered design, and task modeling was a part of it. Task models [Balbo, et al. 2004, Limbourg and Vanderdonckt 2004] may be used to analyse systems, and express the different activities that the user wishes to perform when using the software [Diaper and Stanton 2004]. These models are designed in order to reach several goals [Balbo, et al. 2004], such as to help during the design step, to help

during the validation step [Mori, et al. 2004], or to generate an application component. The lecture on the task modeling process did not aim at teaching all task models; it only focused on the K-MAD model. This model was initially explained in details during the lecture (nearly 4 hours), then the students practiced task modeling using K-MADe during the training courses that followed (§ 5.2). Even if they were not modeling experts, they were more extensively trained on how to use a task model notation than on actual ergonomics expertise [Couix 2007].

The second part of the lecture was focused on various evaluation concepts and on the main methods used [Nielsen 1993]. This survey of concepts and methods served also as background for the students of the first school-year, who played the role of observers in the *usage* evaluation session. Even though their participation was limited to observing and taking notes (the notes were completed by other data sources), the presentation of evaluation concepts aimed at facilitating the student evaluation work (this study was their first practical evaluation).

Both year courses concerned the same concepts and followed the same schedule.

5.2 Practical Training Step

The practical training step was composed of two training courses sessions (without any pattern in Figure 4) and one session of usage evaluation (with the hachured pattern in Figure 4). The first and third sessions were similar for both groups, except for the number of task models to design. However, the second practice session (the *usage* evaluation) was only performed by the first school-year group. In this section, we describe the exercises performed during these three sessions as well as the corresponding instructions.

5.2.1 First practice session, first school-year group. During the first practice training session, the first year group had to design two different task models. The first one described their activity before arriving at the university each morning (TM1_1). Few concepts were used to design this task model. They had to decompose this activity without defining objects or conditions. The training target of this exercise was to show students the capacity of K-MAD to model the different scenarios that were anticipated by each student. In order to design this first task model, students could use either the K-MADe tool or simply paper and pencil. After this first modeling, students used K-MADe as a task model editor.

The second task model described the activity of a clerk in a car rental office. In the design of that second task model (TM1_2), all K-MADe concepts were used: tasks, objects, events and users.

Tasks. In a car rental office, the clerk welcomes visitors, establishes contracts, receives cars that come back from rental or from repairing and last does what is needed for the damaged cars. Concurrently to these activities, the clerk answers the phone when it rings. Tasks performed by a car rental clerk may be expressed in the form of task trees such as in Figure 5.

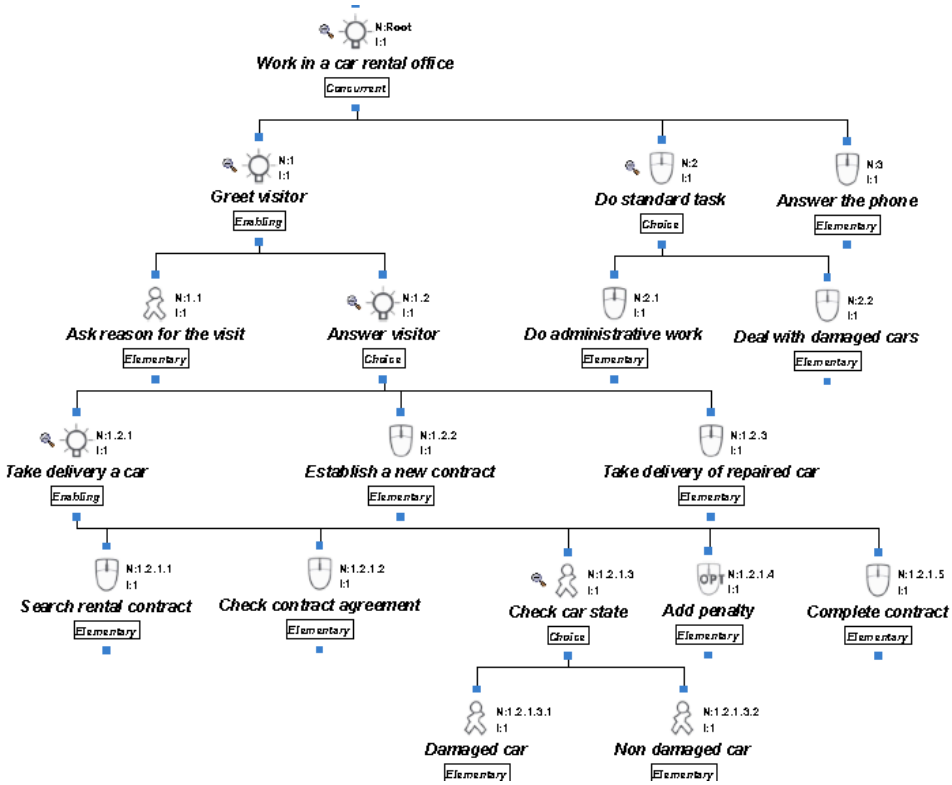


Fig. 5. K-MAD task tree of car rental clerk work

Objects. Two objects were handled in this activity: *car* and *contract*. A car is defined by a color (String) and its production year (Integer) and a contract concerns a client (String) and a car (String), and contains a price (Integer) and a rental time (Integer). K-MAD objects are stored into groups. A car and a contract may be member of different groups according to their state: avalaible, rent, damaged, repaired for a car and new, current or past for a contract. Table II summarizes the characteristics of objects and of groups that store them.

Table II. Car rental Objects

Object Name	**Object attributes**		**Object groups**		
	Name	**Type**	**Name**	**Description**	**Type**
Car	color	String	AvailableCar	Cars that can be rented	Set
			RentCar	Cars not available	Set
			CarInReception	The car currently observed	Singleton
	year	Integer	DamagedCarTo	Damaged cars	List

				that need to be repaired	
			DamagedCarAtGarage	Cars at garage	Set
Contract	client	String	CurrentContract	The contract currently in edition	Singleton
	car	String	NewContract	Contracts for a new client	Singleton
	time	Integer	Store	Contracts stored in the car rental office	Set
	price	Integer			

The defined objects are used to express computable expressions. For example, the post-condition of the *Search rent contract* task is that a stored contract comes back to be edited (*replace($Store,$CurrentContract)*) and its computing allows the *Complete contract* task execution whose pre-condition is "a contract is in edition" (*Card($CurrentContract)*).

Events. Only one event is modeled for this activity: the phone ringing. This event triggers the *Answer the phone* task.

Users. Three different users are associated to the task tree presented in Figure 5: the clerk woman, the garage and a client.

To design that task model, the students were paired and they did not have any time limit, after the end of the practice step (the duration averaged about two hours). During the first hour and half, a tutor was present and answered the student's questions. When designing this task model, the students were able to explore the K-MADe tool.

5.2.2 First practice session, second school-year group. During this first task modeling session, the second group of students had to perform only one task model (TM2_1). This task model expresses the activities they might perform with a calendar (such as editing an event, synchronizing a calendar on a computer and on a phone, for example). All K-MAD concepts were mandatory. In order to limit the initial difficulties of the K-MADe use for the students, they began their task model without K-MADe (by using paper-pencil) during two hours, in order to focus on the concepts rather than on the tool. After that, they completed their task model using K-MADe (nearly two hours with the K-MADe tool). As for the first year group session (described in 5.2.1), students were only helped at the beginning of the task model design process and they completed their models alone.

5.2.3 Second practice session (first school-year group). During the *usage* evaluation session (the second practice session), students had to model the activity of completing a volley-ball game marking sheet (TM1_3) alone, without any tutor help. Instructions for this activity were given at the beginning of the session. The instructions were extracted from the official instructions from the French Federation of Volley Ball (FFVB), and two examples of marking sheets (completed and non-completed ones) were provided. This task model design was limited in time: one hour and thirty minutes.

5.2.4 Third practice session (both school-year groups). Finally, to design the last task model (TM4), students could ask questions to tutors, they modeled activities of members of a genetic analysis laboratory (technicians, secretary, manager…). From this task model, they had to design an application for the laboratory

(http://www.genindexe.com/). They worked by groups of three or four students to perform this last modeling exercise.

6. EVALUATION METHOD

This evaluation aimed to analyze the use of K-MADe during the whole task model learning process. In order to perform this assessment, we used two classical evaluation techniques [Nielsen 1993]: observation of subjects when using the tool and consultation of users by analyzing questionnaires.

A two level observation was conducted: an observation of a whole group (global observation) and an observation of the behavior of each student (individual observation). The individual observation of all students focused on evaluating the usage of the K-MADe formal components.

In the next section, we describe the experimental procedure.

6.1 Global Observation

In the first year, after each session, the tutors noted their observations: the difficulties in understanding (wordings and/or concepts) were precisely recorded. The tutors also noted the student behavior (reactions/ comments/ questions). These *global notes* aimed at helping the interpretation of other gathered data (such as the task models produced). They contained two types of information: the exercise (or the question) and the comments about it.

During the whole learning process, the tutors noted also the individual queries. Those concerned the individual activities such as checking the models or difficulties in using the tool. They were either orally asked or sent by email. As during the training, each individual query and correction was explained to the whole group, *individual note* is considered as global data.

In addition to the common note data (observation reference and comment), this type of notes (*individual notes*) included the reference of the observation author (student name or group). Due to individual information, the *personal notes* were added to the *global observation* and to the *individual observation*. The tutors' notes played an important information role and completed the understanding of other gathered data. Moreover, during the sessions, tutors collected specific data on evaluation topics.

6.2 Individual Observations

The individual observations of student behavior when using K-MADe to design their task models required a particular method. These observations were set up during the second practice session (Figure 4). All students of the first school year were paired. During the first half part of the session, one student acted as a task model designer (using K-MADe, named *user*), while the second acted as the evaluation expert (named *observer*). They reversed roles during the second part. The tutors assigned roles played by each student in order to be sure that all the student levels were represented. Each session lasted one hour and a half with a fifteen minute break between sessions. The activity that was to be modeled was the same for all students and it was described at the beginning of the sessions.

6.2.1 Observers' notes. During the modeling, the *observer* ensured that the user verbally described his/her modeling process, and noted what s/he observed concerning the use of the tool by his/her *user* (hesitations, exploration of several parts of the

software without actions, etc.). In order to help observers in their evaluation, they were given observation sheets (illustrated in Table III). These sheets consisted mainly in a three-column table corresponding to the three types of information recorded for each observation:

- The type of the observation among a set of defined categories (user goal (G), tool functionalities (F), functionality utilization (FU) and information (I)).
- The observation in textual form.
- The time indication about the observation.

In order to complete the observers' notes and the designed task model, another type of data was used: the user-logs.

Table III. Observation sheet example

Type	**Observation**	**Time**
FU	The main window is not accessible ("simulation" is noted but the simulation window is not accessible either). => launch K-MADe again	14h32
G	looking for the object definition	14h34 14h37
F	user does not understand the meaning of the button with shell-hole	14h40

6.2.2 User-Logs. To complete the observations carried out during the *usage* evaluation (practice session 2 of the first school year), the students used a version of K-MADe that produced user-logs. This version allowed keeping track of the user's actions using timestamps and produced a user-log as a text-file. Specifically, the log indicated when the user entered and exited each K-MADe space: task space, abstract objects, condition edition windows (pre, post and iteration), etc. Figure 6 shows an example of such information. User-logs were not generated in order to be used to replay user actions. Evaluators used them in order to deduce a mental representation of student task modeling (in order to complete observation notes).

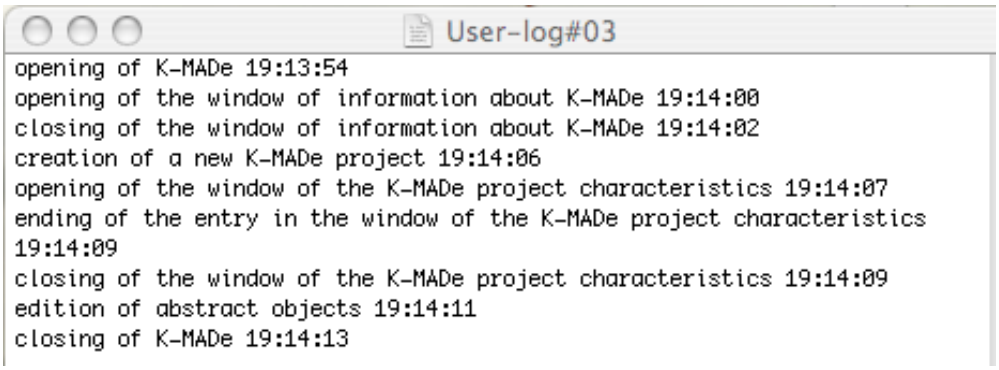


Fig. 6. A user-log produced using K-MADe

6.3 Questionnaires

Results of the first school-year evaluation allow us to point out several questions on the task modeling learning process. In order to answer these specific questions, we defined two questionnaires. The students of the second school-year were asked to complete them as regards the use of the tool. They completed the first one at the end of the *model appropriation* step (once TM2_1 was designed) and the second one at the end of the *model use* step (once TM4 was designed). These questionnaires (presented in the Annex part) were composed of questions on the use of the K-MADe tool as a way to learn task modeling. Three questions were common to both questionnaires in order to analyze the progression of students' answers.

7. EVALUATION DATA

Since each session tried to reach different goals, we did not collect the same data for each experiment. In this section, we present the results for each session. Table IV summarizes the data gathered according to the session. In the second part of this section, the selection of data included in the analysis (Table V) is explained.

7.1 Data collection

What are the benefits of using a tool at the beginning of the model appropriation? During the first practice session of the first year, the goal was to understand the benefits of the tool at the beginning of the *model appropriation*. In order to analyze this first point, we counted the cases where the students chose to use the K-MADe tool to design their first task model (TM1_1). Then the tutors noted the distribution between students.

How did students design their task models? What K-MADe functionalities were used to perform task modeling? The knowledge of the task modeling process requires to collect data for the design of the three task models (TM1_2, TM2_1 and TM1_3) as well as the answers to the questionnaires (by focusing on specific items). The TM1_2 task model allowed the students of the first year to become familiar with the K-MADe tool in order to guarantee that the design of the TM1_3 task model is independent from the tool discovery step.

The second practice session (designing TM1_3) aimed at analyzing the task modeling process, particularly when the K-MADe entities were defined and when they were manipulated. In order to obtain this information, we used user-logs and notes from the observers. These two types of information allowed the collection of two complementary sets of data. While the observer was focused on the user usage, on her/his goals and on his/her conceptual models, the user-logs gave information on how the K-MADe components were used. By using timestamps on both data, we can establish how users used the tool components.

Moreover, we requested each student to exploit their notes and the user-log in order to produce an evaluation report. This report included the modeling process of the observed user, his/her use of the tool, and an analysis of the resulting model. Even though the documents produced were readable and quite organized, models, observer notes and user-logs were also collected for expert analysis.

How did the students' learning progress? This *usage evaluation* session helped to understand the learning of the task model. However, as the first year evaluation needed to be completed, additional data was collected on the learning process of the second

year students. The main point that needed additional information was the learning progression. The collected task models were designed without any time limitation and thus provide a first indication about the learning progression. Two other task models performed in the second year (TM2_1 and TM4) completed the data.

Finally, in order to complete the data about the evolution of K-MADe use, questionnaires were submitted to the second school year students. These questionnaires were composed of questions about specific points raised from the exploitation of the first-year evaluation data.

Lastly, during all the sessions, when the tutors were present, they took notes (including for the last session). There notes were used to complete the answers of all questions.

Table IV. Data collected

Learning step	**Practice session**	**Data**	**Year 1**	**Year 2**
Model appropriation	session 1	tutor notes	X	X
		designed task model TM1_1	X	
		designed task model TM1_2	X	
		designed task model TM2_1		X
		questionnaires		X
	session 2	user-logs	X	
		observer-student notes	X	
		student exploitation document	X	
		designed task models TM1_3	X	
Model use	session 3	designed task models TM4	X	X
		tutor notes	X	X
		questionnaires		X

7.2 Selection of data

Eleven models were assigned to the first group as students were paired to perform the second task model (TM1_2). However, one task model was not completed due to a virus and another one was not returned at all. Then 9 TM1_2 were analyzed.

During the *usage evaluation* session (practice session 2) we collected one folder (set of information) per student (48 folders). It included: observer notes, observer exploitation document, task model, user-log. The second practice session aimed at gaining some understanding of the modeling process. The data used to infer the user modeling process was mainly taken from user-logs. This file was automatically generated. However, we did not wish to use these user-logs without taking the context into account (described in the observer-student notes and the exploitation documents). Four of the folders were not complete and therefore were not included in the analysis. We ended up considering 44 out of the 48 folders in our analysis. However, the TM1_3 folders were only used when we considered that they had a strong impact: for example, when we analyzed the impact of K-MADe on the training progression, we only considered the 18 completed folders corresponding to the folders of the students that designed the collected TM1_2 task models.

Three specific data sources were gathered from the second year. Firstly, we gathered their TM2_1 task models. To design these task models, students worked in groups of three or four people (16 groups). All 16 groups designed task models that were returned and used.

Secondly, after the first use of K-MAD and K-MADe to design the TM2_1 task models, we collected the first questionnaires. The answers to the questionnaire (43 questionaires were returned while 7 were not) were analyzed according to their different items.

Thirdly when the students completed their last task models (TM4), we collected the second questionnaires. 37 questionnaires were returned. As for the first questionnaire, we selected questionnaires according to the potential impact of the answers. To study the learning process, we analyzed the students who answered both questionnaires (questionnaire 1 and questionnaire 2). From all questionnaires, we collected 35 pairs of questionnaires.

Finally we collected the designed TM4 of both evaluation groups. To design the TM4 task models, students were put into groups. Each group designed one or several task models. All the groups returned all their task models (regrouped into folders).

Table V. Selection of the data

year/group	data	collected	used
1	TM1_2	11	9
1	first group TM1_3 folders	48	44
2	second group TM2_1	16	16
2	first questionnaire	43	according to their relevance
2	second questionnaire	37	
2	both questionnaires	35	

8. RESULTS

The data collected during all the learning sessions allowed us to illustrate the task modeling learning process. We classified our results according to three lines: the training point of view; the autonomy of the students to correct their models; and the evaluation of the completed models.

8.1 The use of the tool from a training point of view

During the first session of the first year, 22 students were present; among these students, 16 chose to design their first task models (TM1_1) with K-MADe. This choice can be explained by two reasons: the major part of students chose to use K-MADe before they were asked to use it to design the other models (in order for them to factorize their learning, i.e., learn at the same time both the model and the tool); others explained that they thought the use of K-MADe would help them to learn a new formalism. The students had a bioinformatics background so they had no apprehension about using the software.

The four students that chose paper and pencil at the beginning of the TM1_1 design, ended up using K-MADe, even though the tutors imposed the use of K-MADe only for the second task model (TM1_2). They changed their choice (to use the K-MADe tool) after a discussion with the tutors about the validation of their models. After this discussion, as they had the feeling that their models were less readable (deletions,

adding of hierarchical level) than the K-MADe user models, they preferred re-editing their task models and completing them using K-MADe.

In the second year, in order to complete this first analysis, we modified the instructions to design the task models. The design of their first task models (TM2_1) was divided into two steps. Firstly, they expressed the task decomposition and informal objects without K-MADe (using paper and pencil). Secondly, they used K-MADe to complete the design of their TM2_1 task models. In the questionnaires, we asked students what learning process they preferred: the use of K-MADe only (from the start of task modeling) - a two step process with a first shorter paper-pencil step - or the first step for as long as it took in the second year (nearly 2 hours for the paper-pencil step) - or a longer paper-pencil step. The results are presented in Table VI.

Table VI. The students' preferences regarding the task modeling learning step

Task modeling learning step	**Number**
K-MADe use only	5
Shorter paper-pencil step	10
Steps as in the second school year training schedule	26
Longer paper-pencil step	2

Students most predominantly (84% or 38 out of 43 expressed) indicated that they found the splitting of the model learning into two steps more adapted to the task. The duration of the paper-pencil step was not precisely defined. While 60% judged two hours as an appropriate duration, 23% considered it too long.

Three task modeling processes may be defined: (1) K-MADe not used; (2) modeling by using paper and pencil prior to using K-MADe; (3) the use of K-MADe only.

At the end of the task modeling learning, 17% of students indicate that they began the task modeling on paper before completing it by using K-MADe (second task model process). All these students specified that the paper step focused more on task decomposition. All other students (29/35) designed their task models only by using K-MADe (third task model process). However, some students changed their task modeling processes. These processes were evaluated from the beginning to the end of the task modeling learning. Table VII presents the migration of the students' processes.

The migration of students from the process without K-MADe (first task model process) or mixed (second task model process) to the design using only K-MADe (third task model process) may mean that all students would have designed their task models by following the third process if they had practiced long enough.

Table VII. The migration of student processes

Initial task model process	**Final task model process**	**Number of students**
first task model process	third task model process	4
second task model process	third task model process	8
third task model process	second task model process	3
		Total: 15/35 expressed

Due to the small size of the first task model (TM1_1), the benefits of K-MADe for navigating task models is not highligted by this first modeling exercise. In order to facilitate such navigation, K-MADe provides two functionalities: the zoom on the whole

task model and the capacity to display or not the subtasks of a task. The data gathered during practice session 2 shows that 93.75% of students used at least one of these functionalities in order to navigate within their task models.

While the K-MADe software provides several functionalities to facilitate the edition of task models, the paper-pencil method offers a greater freedom in personalization. An assessment of five experts in the ergonomics domain [Couix 2007] shows that they mainly use a personal formalism (4/5). The use of software tools cannot be adapted to this type of users. However, K-MADe allows users to personalize the task model presentation (without any formalism implication). Even though tutors did not present this personalization functionality, 37.6% of the first school-year students modified their preferences during the design of their TM1_3 model. Thus, the personalization of their design tool seems to be a key functionality for the students. Moreover, to be appropriate for task modeling learning, the use of a model that they have never used before needs strict application of the grammar rules in order to allow tutors to identify the student misunderstandings. The use of software tools (such as K-MADe or CTTE) that support training, increases the need to comply with formalism requirements.

K-MADe offers two tools that help to observe modeling rules: the grammar-checking tool (Figure 7a) and the simulation tool (Figure 8). The grammar-checking tool detects mistakes such as the fact that an elementary task cannot be composed of subtasks. Since the beginning of K-MADe use, the tutors have indicated that the presence of both tools in K-MADe has helped to teach, but that it has done so only as regards the task modeling grammar difficulties.

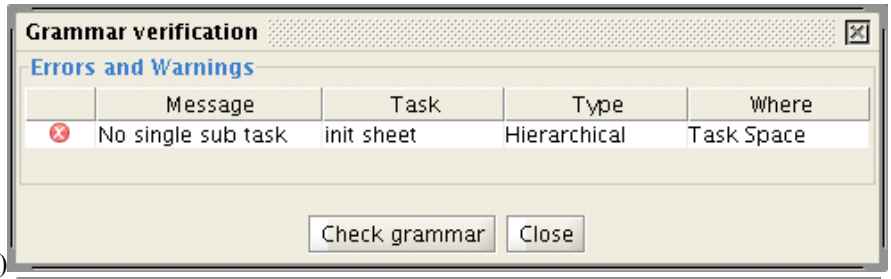
The students detected and corrected their grammar mistakes alone. In addition to this planned use, the grammar-checking tool highlighted the students' misunderstanding of the task composition concept (nearly 1/3 of the first school-year students). The students mixed up the notion of decomposition and inheritance and thus defined tasks composed of only one subtask. This misunderstanding was explained and corrected by the tutors as soon as it was detected (during the first practice session).

8.2 Correction of task models by the students

The formal aspects of task modeling allow the detection of discrepancies (by using the grammar checking tool for example, Figure 7a) prior to allowing the access to the simulation tool (Figure 8). The simulation tool presents a dynamic simulation of the task description (label 1 in Figure 8). Once the available task set (label 2 in Figure 8) is calculated, the designer selects a task that will be added to the current scenario (label 3 in Figure 8). According to the described context (label 5 in Figure 8), the selected task is performed and expressions are computed (label 4 in Figure 8). In addition to the tools that may be used to ensure task model consistency with formalism requirements, another tool indicates information characteristics on the designed task models (Figure 7b). This informational tool was never used by students which shows that during the learning process, students only used tools that helped them to ensure consistency with formalism requirements.

Some observations in this evaluation concerned the formal verification tools for the model design. The reasons why students used K-MADe correction tools and their timing (grammar-checking tool and simulation tool) in the task model processes can be found below.

a)



b)

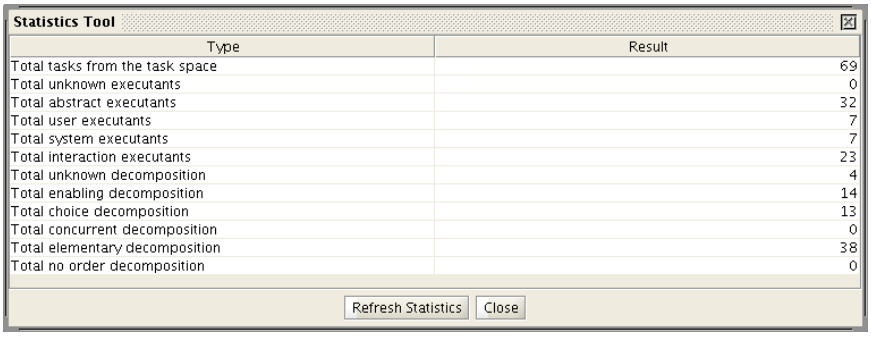


Fig. 7. Views of analysis of task models: a) the grammar-checking tool and b) the statistics tool

When students designed task models without any help (TM1_3), 62.5% of them used at least one correction tool while 12.5% indicated that they did not because of lack of time.

The study of user-logs (from practice session 2) shows that users triggered the correction tools, especially after they defined their task trees (37.5%) and when they considered the task modeling as complete (75%). These uses correspond to the errors detected by the grammar-checking tool. This tool detects at least one error in 90% of all cases indicating that users need this type of tools to design their models. Moreover this grammar-checking tool is the most used correction tool (87.5% of students closed the simulation tool as soon as any coherence error is detected). This last observation introduces the idea that the students triggered the simulation tool in order to grammatically check their task models. Moreover, the formal conditions were systematically checked (sometimes many times) when they were written in order to verify their syntax.

While data from the *usage evaluation* (practice session 2) indicates that users take advantage of the grammar-checking tool, it does not allow to evaluate the full benefits of the simulation tool. During the first year, students asked a lot of questions about the simulation tool functioning and use. To design their TM1_3 task models, only half of students successfully opened the simulation tool and only 6.25% created at least one scenario. During the next sessions (to design the TM4 task model), tutors explained the use of the K-MADe simulation tool again. From tutors' notes, two main reasons

explained that students did not understand the simulation tools: first, when the tool was presented (during the lecture), the students did not understand the relevance of this tool and thus they did not memorize the simulation tool functioning; secondly, when students were ready to verify their models, the tutors showed where the simulation tool could be triggered but did not show how to create scenarios.

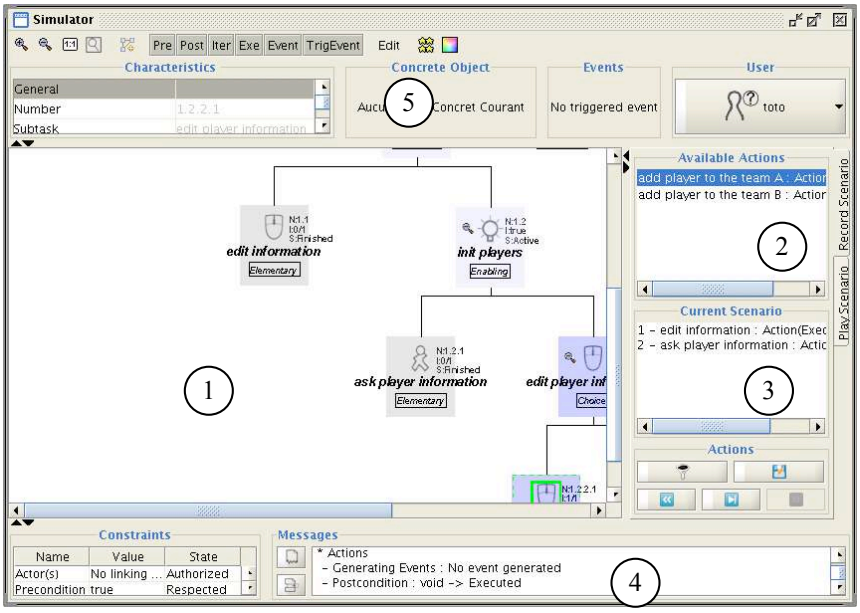


Fig. 8. View of the simulation tool

In order to make sure that the misunderstandings on the simulation tool were due to a teaching problem, during the second year, a part of the lecture focused on the task model checking by using tools. The independent presentation of the simulation and grammar-checking tools showed that both tools could be used in order to validate task models. In the questionnaires, the students were asked to explain in their own words what these tools actually were. All 35 pairs of returned questionnaires were considered. For the students that defined tools (*define* in Table VIII), the main part provided a correct definition (*correct* in Table VIII).

Table VIII. Number of students that defined verification tool

	From the first questionnaire		**From the second questionnaire**	
	Define	Correct	Define	Correct
grammar-checking	21	18	18	18
simulation	19	19	20	19

The definition of the verification tools and their use allows a good understanding of these two tools and their benefits (at the end of the learning process 100% of students

correctly explained the grammar-checking tool and 95% of students correctly defined the simulation tool).

In order to analyze this observation more precisely, in the second year, the tutors presented the triggering of the grammar-checking tool independently from the simulation tool. Table IX presents the results of the questionnaire answers about the use of the grammar-checking and simulation tools by the second year students. In the middle of the learning process (once TM2_1 task models were designed), 3 questionnaires (from the 35 pairs of returned questionnaires) were not used for the analysis of this point as they contained inconsistencies or as student answers were unclear. For the 32 other questionnaires, 23 separately used both tools. The others (9) used only one of the verification tools (4 the grammar-checking tool and 5 the simulation tool).

At the end of the learning, from the 32 selected questionnaires, 30 separately used both tools and 2 used only one of the tools (1 the grammar-checking tool and 1 the simulation tool). No student that used both tools launched the grammar-checking tool via the simulation one. Thus, the first school-year students performed the grammar-checking via the simulation tool because it was presented according to this process. Both tools were used in 93.75% of the student designs.

Table IX. Results of questionnaire answers about tool uses

	Questionnaire 1	**Questionnaire 2**
Separate use of grammar-checking and simulation tools	23	30
Use of grammar-checking tool only	4	1
Use of simulation tool only	5	1
Use of grammar-checking tool via simulation tool	0	0
No significant answers		3

Finally, 77.8% of the students in the first year who produced task models with grammar-errors modified their task models to correct them. All of these students corrected their task model until there were no grammar errors. The user-logs showed that the students who did not correct their TM1_3 task models understood the functioning of the grammar-checking tool later (five minutes prior to the end of the session). We may make the hypothesis that they did not correct their task model grammar because of lack of time. The other reason may be the misunderstanding of the grammar-checking tool messages (Figure 9). From the questionnaires, we analyzed the understanding level of the grammar-checking messages. This analysis highlights that, at the end of the task modeling training, 77% of students claimed that they understood the grammar-checking messages. However, the message understanding requires time (50% of students declare they did not understand the messages after their first modeling task (TM2_1) and this number decreases to 17% after their TM4 modeling task). The TM4 task models supported this idea. All the collected task models were grammatically correct.

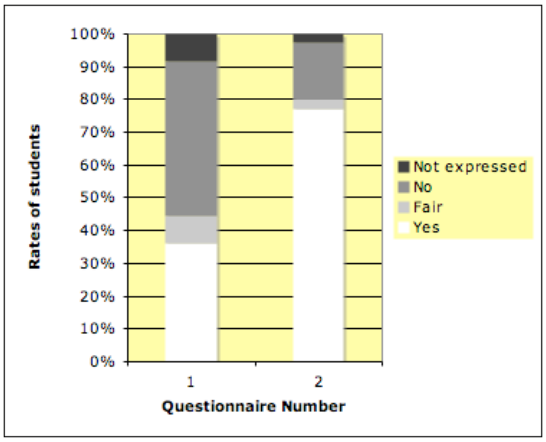


Fig. 9. Understanding of the grammar-checking messages

8.3 Evaluation of models

In this section, we exploit task models that were designed without any time limit in order to analyze time-independent data.

During the first practice session, in order to design their second task model (TM1_2), the first school-year students had to define abstract objects, concrete objects, events, users and to use them to write pre, post and iteration conditions. All these concepts were presented during the lecture and the tutors illustrated the editing of a pre condition. They also corrected the students' mistakes. We did not evaluate whether students correctly used the entities, but only whether they used them or not. Figure 10 presents the percentage of first school-year students who defined formal entities to design the second task model (TM1_2). In addition to the entity definition shown in Figure 10, all TM1_2 task models were composed of at least two abstraction levels.

As shown in Figure 10, the abstract and concrete objects and their use to define *post conditions* seem to be well understood. The definition and use of objects require the manipulation of strings (to name), set types (to store objects) and basic computer types (to define attributes). All the students have a computer science background and therefore no difficult technical issue was raised. Whilst the need for defining entities is well understood by students, the roles and links do not seem to be as well assimilated. Concrete objects are typically used to define conditions. However, 12.5% of students did not instantiate them when defining abstract objects, and 12.5% did not use objects at all to express conditions. 25% of students thus defined abstract objects without any link to any other concept.

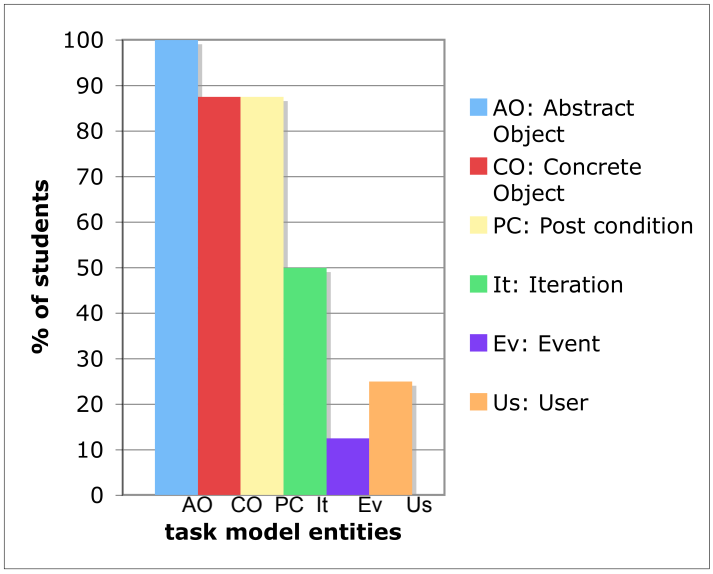


Fig. 10. Percentage of students that defined formal entities during the first session

Concerning the formal **conditions**, 87.5% of students defined at least one post condition from the first session onwards. We did not take into account the definition of pre conditions because tutors used them to illustrate the use of the K-MADe calculator (see Figure 3). When students did not succeed to formally define their conditions (based on the B language [Lano 1996]) using the K-MADe calculator, they did it literally by using textual definition (Figure 11). Even though the definition of the K-MAD formal entities and conditions when using K-MADe caused some difficulties, the necessity of their use in the task modeling process is natural for students in order to complete the task scheduling.

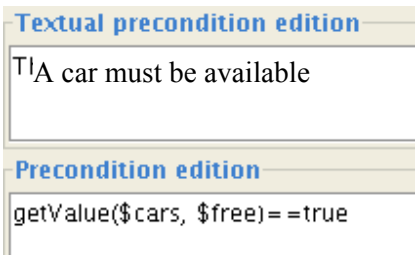


Fig. 11. Edition of a pre condition

The post condition concept is known by the students as Boolean conditions. However, K-MADe defines post conditions as the action(s) performed by task execution. This definition may imply that students misunderstand and that tutors ought to clarify the post condition definition in order for the students to correctly use K-MADe (as they did for the following year students).

Figure 10 shows that few students used the *iteration* conditions, *events* and *users*. Iterations are defined as pre and post conditions (using the calculators); events as well as users are defined as strings. The lack of use of these concepts is not due to the teaching. Whilst objects, pre and post conditions were presented (and illustrated) by tutors, iteration, user and event were only presented in the lecture (without any example) during the first year. Conversely, in the second school year, tutors explicitly presented all concepts. There were no significant changes in the rates of iteration, user and event definitions for the same activities (TM4).

In the first year, we studied the student task modeling process during practice session 2 and we especially observed the impact of formal entities in these processes. Prior to identifying the impact of formalising in the task modeling process, we observed that some students did not define objects. Indeed, 26% of first school-year students (12/46) in the second practice session did not try to define (or use) any K-MADe formal entities. However, we cannot precisely identify why. Two reasons may explain the absence of these elements in the task model process: the limited duration of the experiment, or the non-assimilation of object concepts. Student notes and reports did not allow us to identify the main reason. Six participants indicated that the sessions were not long enough but others (6 out of 12) did not give any relevant information on the subject. The task models of the second school-year students (that are designed without any time limitation) support this idea (only 2 of the 43 TM2_1 are not composed of objects).

From the 34 remaining TM1_3 folders (folders with task model integrating formal entities), we identified two main processes followed by students to define objects and conditions. Table X presents the repartition of students for each process. The most used one (71.5% of processes integrating formal objects and conditions) separately defined objects and conditions. For this student group, 7 out of 10 did not define their objects again once the conditions were defined. These observations give us some understanding about the role objects have in the task modeling process. As an example, the concurrent definition of objects and task tree composition indicates that the user associates objects and tasks. In contrast, when the definition and the use of objects are separated from the task tree composition, we can deduce that the user defines objects only to use them for condition expressions. Therefore, objects associate properties and tasks for some students.

Table X. Repartition of definition processes

Did not use any formal entities	**Separately**	**Conjointly**
12	25	9

9. CONCLUSION AND FUTURE WORK

In this paper, we presented an assessment based on the exploitation of data gathered during the use of K-MADe to support the learning of task modeling by bioinformatics students. This evaluation concerned the use of K-MAD by students in a design phase. The data gathered highlights that students find the functionalities of task model navigation (93.75% of students use at least one of the navigation functionalities) useful. Moreover, they indicate that, after being introduced to task modeling (60% of students think that the first two hours of task modeling without K-MADe are mandatory), they prefer using the K-MADe software because the designed task models are more readable

(all students that chose to use paper-pencil method to design their first task model changed their choice after the first modifications). From a training point of view, the use of K-MADe provides a syntactic help (90% of students that launched the simulation tool had at least one syntactic error). Moreover, this analysis shows that students use computing K-MAD concepts only (*user*, *event* and *iteration* concepts are less used than others even though they are presented) because they have some feedback (by using the simulation tool for example). So, this evaluation highlights that the presence of task model tools is essential for learning modeling as it is for editing task model [Giese, et al. 2008].

Overall, this study shows that the use of K-MADe is attractive for task modeling learning. It is particularly beneficial when students are trained to use the conditions in K-MAD. In order to increase these learning benefits, the tool must be modified. These modifications aim at better integrating the grammar-checking tool and facilitating the simulation tool use. For example, Figure 12 shows a grammar error detection for a task (this task is the first task of the marking sheet activity), which is modified in order to integrate the grammar-checking tool. Students can thus see the grammar-checking of their task models as soon as they edit them.

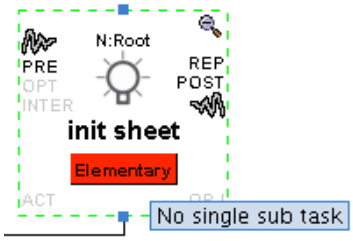


Fig. 12. Example of grammar-checking for a task

During this training evaluation, we also observed that the definitions of objects and conditions are difficult for students. However, both concepts are mandatory to completely express activities [Dix 2008] and to make verifications to validate the models. Another assessment was performed in order to specifically study object and condition definitions [Caffiau, et al. 2008].

Finally, this study focuses on a specific use of the task modeling process (to design interactive applications) for specific user profiles (bioinformatics students). In order to complete this study the same assessment ought to be performed with different user profiles (e.g. computer scientists or ergonomics experts on HCI design). However, this first study allows us to establish some guidelines for task modeling teaching. These guidelines are not general (as they result from a study in which participants have specific knowledge and backgrounds, and in which only the K-MADe tool is studied), but we use them in order to improve our task modeling teaching for students with computer science knowledge and to propose the same teaching for students with different skills (I.T. students, bio-informatics and biology students).

The first identified guideline is issued from the observation of the benefit of K-MADe use during the learning process. The tool helps students to design task models that are consistent with model formalism requirements and therefore facilitates tutor work and students' learning (by increasing student autonomy) during task modeling training.

It must be noted that the use of the tool for task modeling learning is beneficial only if tutors are expert users. As they can anticipate usage difficulties and implementation bugs (for our training, we used a K-MADe version that is currently being developed).

The second identified guideline concerns the activities that are modeled during the training. If the students are familiar with the first modeled activity, the activity discovery step does not take a long time and the students can thus focus only on the modeling process. However, the use of an activity for which the number of available scenarios may be important (such as going to university) is prohibited.

Moreover, to provide exercices for teaching task modeling to students with different skills, activities that include applications (such as playing an on-line game or sending emails) must be limited because computer scientists will describe an activity from a system viewpoint. However, in order to illustrate the use of task modeling in user-centered design, we propose to use the task modeling of an activity for which students will design an interactive application (such as marking a volley-ball marking sheet).

To conclude, even if we define three task-modeling processes for task modeling training, we recommend enforcing the concept definition order. All concepts will thus be used. From our teaching experience, before the learning step, some students modify the concept definition order and adapt their modeling process.

RÉFÉRENCES

BALBO, S., OZKAN, N. AND PARIS, C. 2004. Choosing the Right Task-modeling Notation: A Taxonomy. In *The Handbook of Task Analysis for Human Computer Interaction.* Lawrence Erlbaum Associates. 445-466.

BARON, M., LUCQUIAUD, V., AUTARD, D. AND SCAPIN, D. 2006. K-MADe : un environnement pour le noyau du modèle de description de l'activité. In *Proceedings of IHM'06.* ACM. 287-288.

CAFFIAU, S., GIRARD, P., SCAPIN, D. L., GUITTET, L. AND SANOU, L. 2008. Assessment of Object use in Task Modeling. In *Proceedings of Engineering Interactive Systems (HCSE 2008 and TAMODIA 2008).* Springer (LNCS 5247). 14-28.

COUIX, S. 2007. Usages et construction des modèles de tâches dans la pratique de l'ergonomie : une étude exploratoire.Conservatoire National des Arts et Métiers, Université Paris Descartes, Université Paris 8, Université Victor Segalen Bordeaux 2. Available at: http://www.biomedicale.univ-paris5.fr/taskmodelsurvey/accueil/.

DIAPER, D. 2004. Understanding Task Analysis for Human-Computer Interaction. In *The Handbook of Task Analysis for Human Computer Interaction*. Lawrence Erlbaum Associates, Inc. 5-48.

DIAPER, D. AND STANTON, N. A. 2004. The Handbook of Task Analysis for Human Computer Interaction. *Lawrence Erlbaum Associates*.

DIX, A. 2008. Tasks = Data + Action + Context: Automated Task Assistance through Data-Oriented Analysis (invited paper). In *Proceedings of Engineering Interactive Systems (HCSE 2008 and TAMODIA 2008).* Springer (LNCS 5247). 1-13.

GIESE, M., MISTRZYK, T., PFAU, A., SZWILLUS, G. AND DETTEN, M. V. 2008. AMBOSS: A Task Modeling Approach for Safety-Critical Systems. In *Proceedings of Engineering Interactive Systems (HCSE 2008 and TAMODIA 2008).* Springer (LNCS 5247). 98-109.

K-MADe electronic reference with downloap of the software and the manuel (french and english). http://kmade.sourceforge.net/.

LANO, K. 1996. The B Language Method: A guide to practical Formal Development. In *Formal Approaches to Computing and Information Technology (FACIT)*. Springer.

LIMBOURG, Q. AND VANDERDONCKT, J. 2004. Comparing Task Models for User Interface Design. In *The Handbook of Task Analysis for Human Computer Interaction*. Lawrence Erlbaum Associates, Inc. 135-154.

LUCQUIAUD, V. 2005. Sémantique et Outil pour la Modélisation des Tâches Utilisateur: N-MDA. Thesis. Poitiers. 285.

MORI, G., PATERNÒ, F. AND SANTORO, C. 2004. Design and Development of Multidevice User Interfaces through MultipleLogical Descriptions. *IEEE Transactions on Software Engineering.* 507-520.

NIELSEN. 1993. Usability Engineering. *Academic Press*.

PATERNÒ, F., MANCINI, C. AND MENICONI, S. 1997. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. In *Proceedings of IFIP TC13 human-computer interaction conference (INTERACT'97).* 362-369.

BIOGRAPHIES

Sybille Caffiau a obtenu son doctorat en 2009 sur le thème de la prise en compte de l'humain en conception d'applications interactives. Ces recherches couvrent les disciplines de l'ergonomie, l'Ingénierie Dirigée par les Modèles (IDM) et l'Interaction Homme Machine (IHM). Elle est actuellement Attaché Temporaire d'Enseignement et de Recherche (ATER) à l'Ecole Nationale Supérieure de Mécanique et d'Aérotechnique (ENSMA) de Poitiers et membre du Laboratoire d'Informatique Scientifique et Industrielle (LISI). Elle est co-auteur de plusieurs articles dans des conférences nationales et internationales et d'articles de revue.

Dominique L. Scapin est Directeur de Recherche à l'I.N.R.I.A., Rocquencourt, France où il a été Responsable Scientifique des projets MErLIn *(Méthodes pour l'Ergonomie des Logiciels Interactifs)* et PsychoErgo. Il mène depuis 1977 des recherches en ergonomie des logiciels, notamment sur les méthodes d'évaluation et de conception des IHM : par ex., sur les langages de commande ; les critères ergonomiques ; les méthodes et outils de description des tâches ; sur le multimedia et la multimodalité. Auteur de nombreuses publications scientifiques en ergonomie des IHM, il est membre du comité de rédaction des revues BIT (et éditeur associé), IwC, UAIS, IJHCI, RIHM, JIPS ; lecteur pour la plupart des journaux scientifiques du domaine ; membre du comité scientifique de la plupart des conférences du domaine. Il a été président des démos pour INTERCHI'93 ; président d'IHM'94 ; co-président des articles de recherche pour INTERACT'95 ; vice-président d'IHM'99 ; co-président d'ERGO-IHM 2000 ; co-président du comité de programme d'IHM'02. Il a géré une vingtaine contrats de recherche nationaux, internationaux, ou industriels ; encadré une quinzaine de thèses de 3ème cycle en ergonomie et en informatique. Il est expert pour diverses organisations nationales et internationales (par exemple : MESR, ANR, EU, ISO) ; consultant auprès de diverses entreprises et organisations. Il est un ancien président de l'Human Factors Society Europe; et de l'AFIHM.

Loé Sanou est actuellement Attaché Temporaire d'Enseignement et de Recherche (ATER) à l'Université de Poitiers et rattaché au Laboratoire d'Informatique Scientifique et Industrielle (LISI) de l'Ecole Nationale Supérieure de Mécanique et d'Aérotechnique (ENSMA). Il a obtenu son doctorat en 2008 sur le thème de la Programmation sur Exemple (PsE) en définissant une boîte à outils d'implémentation d'application intégrant la PsE, avec application dans le domaine des tests d'IHM (conformité de l'IHM par rapport au modèle de tâches prescrit). Il est auteur et co-auteur de plusieurs articles de conférences nationales et internationales. Il est membre de l'AFIHM depuis 2002.

Laurent Guittet est Maître de Conférences au Laboratoire d'Informatique Scientifique et Industrielle (LISI). Il enseigne à l'école nationale supérieure de mécanique et d' aérotechnique (ENSMA). Après un diplôme d'ingénieur de cette école en 1981, et un doctorat d'informatique en 1992 sur l'architecture des systèmes interactifs, ses activités de recherche s'articulent autour de la conception et la validation des systèmes interactifs. Il est co-auteur de nombreux articles dans des revues et conférences nationales et internationales.

10. ANNEX: THE QUESTIONNAIRES

Nom : Prénom : Spécialité :

Point 1 : Approche de la modélisation des tâches

1.1 – Les premiers modèles de tâches (pour le calendrier) ont été réalisés sur papier. Aurais-tu préféré les réaliser directement avec l'outil K-MADe ?

- ☐ oui, dès le premier modèle à réaliser
- ☐ oui, à la première séance mais pas dés le début
- ☐ non, la première séance sur papier était une étape nécessaire
- ☐ non, et j'aurai souhaiter rester sur papier plus longtemps

Pourquoi ? _____

1.2 – À présent, lorsque tu réalises un modèle de tâches, comment procèdes-tu ?

- ☐ je le fais sur papier.
- ☐ je le commence sur papier puis je le termine en utilisant K-MADe.
  Précises, ce que vous faites sur papier et ce qui est fait sur K-MADe : _____
- ☐ je n'utilise que K-MADe.

Point 2 : Outil d'aide à la conception

2.1 – K-MADe dispose d'outils pour aider à la conception des modèles de tâches. Parmi ceux-ci, tu utilises :

☐ l'outil de vérification de la cohérence

Indiques avec tes mots en quoi consiste cet outil :
Comment le lances-tu ?
Quand l'utilises-tu ?

- ☐ avant le début de la conception
- ☐ pendant le processus, très régulièrement
- ☐ après certaines étapes du processus. Lesquelles ? _____
- ☐ une fois que je pense le modèle de tâche terminé
- ☐ autre. Précises : _____

☐ l'outil de simulation

Indiques avec tes mots en quoi consiste cet outil :
Comment le lances-tu ?
Quand l'utilises-tu ?

- ☐ avant le début de la conception
- ☐ pendant le processus, très régulièrement
- ☐ après certaines étapes du processus. Lesquelles ? _____
- ☐ une fois que je pense le modèle de tâche terminé
- ☐ autre. Précises : _____

2.2 – Comprends-tu toujours les erreurs soulevées par l'outil de vérification de cohérence ?

☐ oui ☐ non

Comment fais-tu pour les rectifier ?

- ☐ je me réfère au(x) message(s)
- ☐ je cherche dans tout l'arbre (sans lire le message, ce qui m'importe c'est qu'il y ait des erreurs de détectées)
- ☐ j'accède directement à la tâche concernée en double-clickant sur le message
- ☐ autre. Précises : _____

Merci.

Nom : Prénom : Spécialité :

Point 1 : Approche de la modélisation des tâches

1.1 – Lorsque tu réalises un modèle de tâches, comment procèdes-tu ?

- ☐ je le fais sur papier.
- ☐ je le commence sur papier puis je le termine en utilisant K-MADe.
  Précises, ce que vous faites sur papier et ce qui est fait sur K-MADe : _____
- ☐ je n'utilise que K-MADe.

Point 2 : Outil d'aide à la conception

2.1 – K-MADe dispose d'outils pour aider à la conception des modèles de tâches. Parmi ceux-ci, tu utilises :

☐ l'outil de vérification de la cohérence

Indiques avec tes mots en quoi consiste cet outil :
Comment le lances-tu ?
Quand l'utilises-tu ?

- ☐ avant le début de la conception
- ☐ pendant le processus, très régulièrement
- ☐ après certaines étapes du processus. Lesquelles ? _____
- ☐ une fois que je pense le modèle de tâche terminé
- ☐ autre. Précises : _____

☐ l'outil de simulation

Indiques avec tes mots en quoi consiste cet outil :
Comment le lances-tu ?
Quand l'utilises-tu ?

- ☐ avant le début de la conception
- ☐ pendant le processus, très régulièrement
- ☐ après certaines étapes du processus. Lesquelles ? _____
- ☐ une fois que je pense le modèle de tâche terminé
- ☐ autre. Précises : _____

2.2 – Comprends-tu toujours les erreurs soulevées par l'outil de vérification de cohérence ?

☐ oui ☐ non

Comment fais-tu pour les rectifier ?

- ☐ je me réfère au(x) message(s)
- ☐ je cherche dans tout l'arbre (sans lire le message, ce qui m'importe c'est qu'il y ait des erreurs de détectées)
- ☐ j'accède directement à la tâche concernée en double-clickant sur le message
- ☐ autre. Précises : _____

Si tu as des remarques sur K-MADe (problèmes récurrents, propositions d'amélioration…), n'hésites pas à les ajouter au dos de la feuille. Merci.