

# Responsive Task Modelling

Davide Anzalone, Marco Manca, Fabio Paternò, Carmen Santoro

CNR-ISTI, HIIS Laboratory

Via Moruzzi 1, 56124 Pisa, Italy

{davide.anzalone, marco.manca, fabio.paterno, carmen.santoro}

## ABSTRACT

In this paper we present a new tool for specifying task models (Responsive CTT), which can be accessed through touch-based mobile devices such as smartphones and tablets as well. The tool is Web-based and responsive in order to provide adapted user interfaces to better support the most common activities in task modelling through various types of devices. We describe the relevant aspects to take into account for this purpose and how we have addressed them in designing the tool. We also report on first user tests.

## Author Keywords

Task modelling; Touch-based devices; Responsive Design;

## ACM Classification Keywords

H.5 Information Interfaces and Presentation; H.5.2 User Interfaces.

## INTRODUCTION

Task models indicate the logical activities that an interactive application should support to reach users' goals. In engineering approaches to HCI, task modelling is useful for various activities e.g. better understand the application domain, record the results of interdisciplinary discussions, support user interface design, usability evaluation, and documentation. Various task model notations are available (e.g. UsiXML [4], ConcurTaskTrees [8], Hamsters [7]), which differ on aspects such as the syntax, the level of formality and/or the operators. An important factor for their adoption is the availability of automatic environments that support model editing, analysis and transformation. Indeed, not all task model notations are supported by (publicly available) tools, and the vast majority of such tools are limited to desktop-based environments in spite of the fact that mobile technologies have radically changed the way we use and develop applications (see e.g. the current *mobile*

*first*<sup>1</sup> approach in Web design). In general, tasks that in the past were suitable only for desktop systems (e.g. accessing videos or long texts) now can be easily accomplished on smartphones. As mobile devices are now part of everyday life and widely applied in various domains, we judged it interesting to investigate the opportunities offered by them for task modelling. For example, if a designer has a new idea when on the move and wants to discuss it with colleagues or stakeholders, mobile devices would enable modelling/visualising tasks anytime/anywhere, which can also shorten the overall design time. We focused on truly mobile devices, i.e. those that can be fully and comfortably used even by users on the go. To do this, we designed and developed a new tool for task modelling implemented following the general responsive design approach[6]. The tool supports task models specified according to the ConcurTaskTrees (CTT) [8] notation.

When building task modelling tools a fundamental issue is how to provide suitable representations and editing features, to effectively visualise and modify the models while addressing scalability. To this aim some approaches have been put forward, such as the CTT Environment (CTTE) [5], HAMSTERS [7] and K-MADe [1]), although they all focused on desktop platforms. Attempts to consider modelling tasks on a different platform were carried out in [3], with a tool supporting collaborative editing of CTT task models on a multi-touch large table; and [9], with a prototype supporting visualisation of task models based on icicle plot rendering with a vertical layout adaptive to screen width.

To summarize, we can say that apart very few attempts, task model tools have been mainly confined to desktop. Our goal is to investigate the opportunities of touch-based modelling on mobile devices, and for this purpose we have designed and developed a new responsive tool for the CTT notation to use on a range of devices, including touch-base mobile ones.

In the paper, after introducing the initial requirements for this work, we describe and discuss how we have designed support for typical task modelling activities during the editing and visualization of such models, the corresponding implementation, and report on first user tests. Lastly, indications for further work are provided.

---

<sup>1</sup> <http://designshack.net/articles/css/mobilefirst/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

EICS'15, June 23 - 26, 2015, Duisburg, Germany

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3646-8/15/06...\$15.00

DOI: <http://dx.doi.org/10.1145/2774225.2775079>

## THE RESPONSIVE DESIGN FOR TASK MODELLING

### Initial Requirements

The development of the tool was driven by a number of requirements, which were identified in our experience of several projects, collaborations and work in which task modelling has been applied. First, in order to widen the impact, interoperability, and possible adoption of the modelling tool, it was developed as a **Web application** exploiting HTML5, CSS3 (for the presentation) and JavaScript (for the dynamic aspects) accessible by any browser-enabled device. Moreover, in order to provide an optimal experience across a wide range of devices the application was also conceived to support **responsive design** to be able to effectively adapt the model representations to the screen area of the available device. In addition, since on mobile devices the screen size is key for an effective analysis and editing of task models, it was judged relevant to think about exploiting **information visualization** techniques for dynamically representing the task models also in mobile devices to harness the power of visualization anytime, anywhere, while relying on a more limited cognitive effort than the one requested in stationary settings. Indeed, mobile contexts make it difficult for users to focus their attention on the device due to external events or activities that have to be considered in parallel, therefore in such contexts users have less cognitive resources to dedicate to task modelling, then the user interface has to allow for a correct interpretation of the information with minimal user effort. Finally, to store and share task models remotely, also because in various mobile devices it is not possible to save files locally from the browser, the application is **cloud-based**, which also facilitates potential sharing and collaboration among users.

### Supporting Typical Task Modelling Activities on Mobile

When modelling tasks, users generally need to: i) interactively modify/rearrange it so that it describes the expected behaviour; ii) visualise/explore/navigate the model, in order to be able to analyse it and take appropriate decisions. Below we describe and discuss the design choices/rationale we identified in order to support such typical activities in our tool.

#### *Editing the Task Model on Touch Devices*

The mobile version has been designed to support flexible ways to edit the task model through touch. In the mobile version users are allowed to: i) touch an empty screen area to create a new task; ii) perform a long press on a task to edit it; iii) drag-and-drop a task on the bottom of the screen to delete it. Users can also navigate the task model by using “swipe up” and “swipe down” gestures to move up/down through the task model levels.

By selecting a specific task, the application sets the focus to that task, and changes the fisheye visualisation of the model accordingly. When users tap on the task that currently has the focus, a semi-transparent circular menu appears,

showing the actions that can be executed on that task: change task properties, add new tasks, add new objects and pre-post-conditions associated with tasks. Figure 1 shows an example of the contextual, circular menu. Also, when users select a task different from the one currently having the focus, the visualisation of the task model is dynamically recalculated to show the new task having the focus in a prominent position of the window and rearrange the model visualisation accordingly. To help users to follow such dynamic rearrangement of the model, a transition effect was implemented to show the change of focus from one task to another: the task that was having the focus before the selection of the new task changes its icon colour from green to blue, while the other changes its icon colour to green. In addition, it is possible to select multiple tasks and then apply an editing operation on all of them in just one gesture (e.g. add an operator, change category, remove a task, add a new level having the selected tasks as task children).

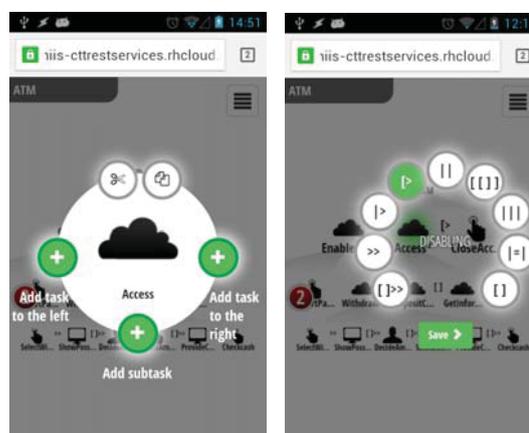


Figure 1: The UI for editing task (left) and operators (right)

Moreover, by selecting the icon of a temporal operator a contextual menu appears, visualising the possible operators (see Figure 1, right part), presented in a ordered way according to their priority. Furthermore, a number of gestures are supported in Responsive CTT: “pinch to zoom” to support zooming in the task model, “swipe down/up” to move up/down in the task model. When task editing is active it is possible to change various task attributes (e.g. name, category), add the specification of objects manipulated by the task and pre-/post- conditions associated to it. Finally, in Responsive CTT the users can also save models in a dedicated cloud-based service.

#### *The Smart Interactive Task Model Visualisation*

In mobile settings the user interface should support a correct interpretation of the information with minimal user effort since users have less cognitive resources to dedicate due to external events/activities that have to be considered concurrently. On the one hand, burdening users with unnecessary data can make it more difficult to reason about the problem; on the other hand, visualizing insufficient data would lead users to take incorrect decisions. Thus, a

convenient presentation is needed to provide effective mobile at-a-glance visualisations. In Responsive CTT, a smart, focus+context, fisheye-based visualisation technique [2] has been designed and implemented as an interactive visual aid to support the visualization, exploration and navigation of models.

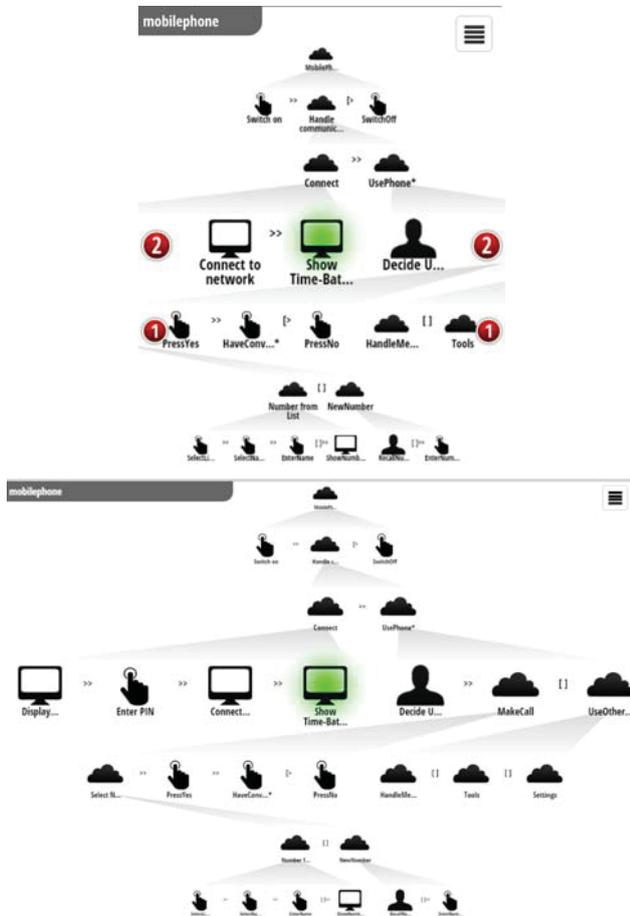


Figure 2: (top) Task model shown on a mobile device (some tasks are hidden); (bottom) Complete task model (on desktop)

The visualisation of a task model is arranged in such a way that the tasks closest to the one that currently has the focus are more emphasized in terms of larger screen space of the associated icons. This is particularly useful on touch-based mobile devices, where precise task selection is difficult due to small screens and limited accuracy of touch commands. The visualisation modality that we propose can be considered as a particular case of “focus+context” representation, where the “focus” area is determined by the selected task, and includes its nearest siblings and children tasks, while the “context” is composed of the remaining tasks of the model. The neighbouring tasks gradually decrease in size, when progressively moving further away from the task that currently has the focus (see task *Show Time Battery* in Figure 2). The task that has the focus is supposed to be the most “important” one, thus it is always

placed in a central position within the working window, and highlighted by a specific colour. More generally, every task has a *degree of interest* dynamically calculated, which is inversely proportional to its *distance* from the currently selected task: the further the focused task is, the smaller the icon of the considered task is. In our case, the “distance” between two tasks is the number of levels to be traversed in the tree to reach one task from the other. Also the height of each level varies depending on the distance of the level of the considered task from that of the task with focus (*focusNode* in the below formula), as defined by the following function:

$$Height(L) = \text{MAX}((\text{maxHeight}/(1 + |L - \text{focusNode.level}| * p)), \text{minHeight})$$

*minHeight* (*maxHeight*) is the minimum (the maximum) possible height, and *p* (which is equal to 0.75) is aimed to smooth the curving of the function to avoid that the function can reach quickly too small values. In the above formula, the denominator of the fraction increases depending on the increased distance of the concerned level with the level of the task with the focus. The minimum value of the denominator is one, held when the concerned level *L* is the level of the task currently having the focus.

When it is difficult to graphically represent tasks in a way sufficiently perceivable by the user because of the limited space, they are automatically replaced with a cue-based technique that shows numbers representing how many tasks are not presented. Such numbers are visualised at the same task level and side of the hidden tasks, with the size of the icon visualising the number proportional to the number itself. Figure 2 top shows an example with hidden tasks both on the left and right parts of the model and at various levels. By interactively selecting a numbered icon, the tasks at the considered level that were formerly hidden, are shown. In addition, we realised that, whatever the task currently having the user’s focus of attention, it is needed to clearly visualise at least: i) the left and right sibling tasks, as they are involved in a temporal relationship with the current task; ii) the path from the root to the task currently with the focus, since it gives the context of the highest level task which the current task belongs to; iii) some of its subtasks, to give an idea of the granularity of the decomposition of the current task. Thus, there is an adaptation of the model layout (see Figure 3) so that such elements semantically connected are lined up in the central part of the model.



Figure 3: Adaptation to move the selected tasks and those more semantically connected in the central area

Furthermore, since large CTT task models tend to grow horizontally, the limited size of the mobile device viewport sometimes does not allow to visualise the model in its

entirety (especially on the horizontal axis). To decrease the width of the model structure first we superimposed a grid on it, so that every leaf can be mapped to a column (see Figure 4).

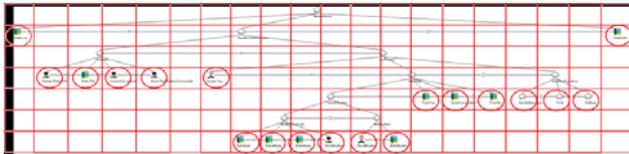


Figure 4: Grid drawn superimposed on the task model

Then, the tasks in the first and last column, which fall in the most outlying positions, are moved toward the centre to save space. This is repeated recursively starting from the most external tasks and going toward the centre, until no further change is possible. Our solution makes such changes to the task model presentation whenever a task model is loaded or any editing operation modifies its structure, before actually visualising it.

Tasks can also have some preconditions, visualised in the task model through a small coloured rounded icon close to the task icon, and whose colour changes according to the precondition state: if it is true the colour is green, otherwise it is red. Figure 5 shows an example while the task model simulator (a module of the tool supporting an interactive simulation of the task model) is running on the cloud-based server. During the interactive simulation the tasks currently enabled are shown in green.

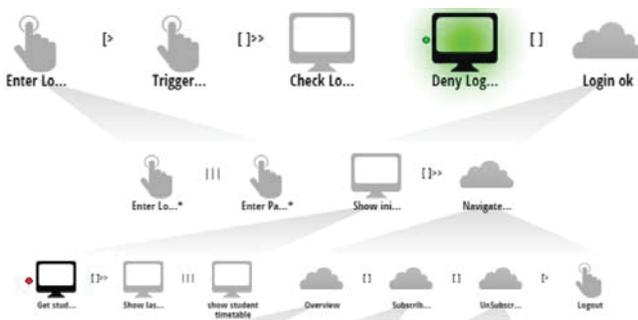


Figure 5: Presenting tasks with preconditions within the Responsive CTT simulator

In the example in Figure 5, after the execution of the *Check Login* task there is a choice between *Deny Login* and *Get Student Data* tasks. Both have a precondition associated with the login state. Since the login is false, the precondition associated with the *Deny login* task is true (small round icon beside the task graphical representation is green) and the task is enabled (shown in green) while the *Get Student Data* task has the small round icon beside the task graphical representation in red since its associated precondition is false, and thus it is disabled.

**Implementation**

The tool implementation is based on the Bootstrap framework (<http://getbootstrap.com/>), which supports the development of responsive applications through these breakpoints: i)Extra small devices (smartphones, up to 768px); ii)Small devices (tablets, 768px and up); iii)Medium devices (desktops, between 992px and 1200); iiiii)Large devices (large desktops, 1200px and up). We exploited Bootstrap in designing various additional responsive features of the task modelling tool. For example, Figure 6 shows how the list of task models is presented on the mobile device (left part), and on desktop (right part): some of the extra space available is used to introduce additional controls (such as that for uploading task models in the cloud from the local repository).

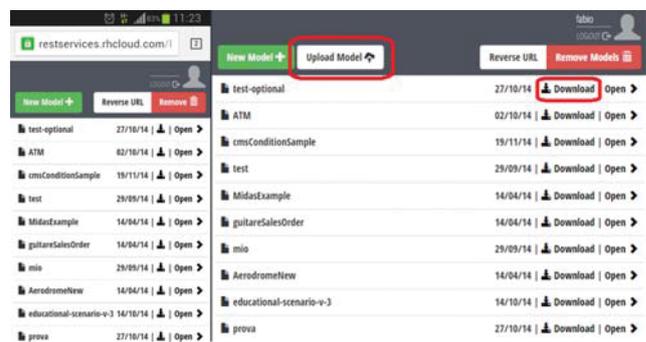


Figure 6: The desktop version of the Responsive CTT visualized on mobile (left) and desktop (right) platforms

We also exploited the open source Hammer JS library that recognizes gestures made by touch (<http://hammerjs.github.io/>).

On the server side, Responsive CTT uses RESTful Web Services to handle requests for log in to the system, open/save/synchronise the files for the clients. The server side is hosted in the cloud through OpenShift, a cloud computing platform as a service (PAAS) given by Red Hat.

The client-server communication supports the following operations: login; logout; open model; save model; get next tasks enabled and save object value (these are used in interactive simulation). The communication is codified using JSON and the (client-side) requests are sent via asynchronous Ajax requests.

**EVALUATION**

Two evaluations were conducted to collect user feedback on the usability of Responsive CTT. We aimed at considering two complementary groups of users: one test involved university students, the other one had professional users as target (recruited through the CTT and the Institute mailing lists). The procedure was similar in the two tests: the users developed a task model both with CTTE on desktop systems and Responsive CTT on mobile devices. The only differences were that students used only smartphone and desktop because it was difficult for them to

have three devices for the test and they started the test during a lesson with a moderator available and completed it later on, while the other evaluation was remotely conducted. The goal was the same in both cases: to assess the usability of some typical task modelling activities using the CTTE on the desktop and Responsive CTT on the mobile device.

### Evaluation with Professional Users

#### Participants and Tasks

Nine people (3 females) aged 26 to 46 ( $M = 35.8$ ,  $SD = 7.8$ ) participated. All had good experience in using mobile devices. They were recruited using both CTT and Institute mailing list. All were novices with Responsive CTT, 6 were familiar with CTTE desktop, 3 had used other tools (e.g. EMODE, Hamsters, HTA, UsiXML). Users were asked to: i) create 3 task models (described in natural language) on 3 different platforms (build a different task model on each platform); ii) send the task models created to evaluators; iii) fill in a questionnaire. The task models considered were of comparable difficulty: handle a conference registration, organize a holiday, handle loans in a library.

#### Procedure, Design and Method

Users remotely carried out the test in their own environments. We sent an email describing the goal, requirements and timeline of the study. We provided users with and instructions about how to access both tools. The study was a within-subject: each user used and analysed both CTTE (for desktop) and Responsive CTT (for smartphone and tablet), and performed the tasks each time using a different device to build a different task model. We counterbalanced the presentation order of the task to model (manage a library, organize a conference, arrange a holiday) and of the platform to use (smartphone/tablet/desktop). After the test, users filled in a questionnaire having a demographic section (about e.g. education, device usage experience/habits), and then questions related to the tools.

#### Results

In the questionnaire, a 1 to 5 Likert scale (i.e. 1 = very unusable, 2 = unusable, 3 = neutral, 4 = usable, 5 = very usable) was used to provide ratings on the tool features.

*Creating a task.* (Desktop mean  $\pm$  SD =  $3.9 \pm 0.8$ ; Tablet mean  $\pm$  SD =  $3.9 \pm 1.1$ ; Phone mean  $\pm$  SD =  $3.6 \pm 1.4$ ). A user suggested contextual menus also in desktop CTTE.

*Editing a task.* (Desktop mean  $\pm$  SD =  $4 \pm 0.5$ ; Tablet mean  $\pm$  SD =  $3.9 \pm 1.1$ ; Phone mean  $\pm$  SD =  $3.3 \pm 1.4$ ). A user said that by selecting the browser's "Back" button, work not already saved could be lost: he did not notice the confirmation dialogue introduced for this goal.

*Control of the graphical editing.* (Desktop mean  $\pm$  SD =  $3.2 \pm 1.1$ ; Tablet mean  $\pm$  SD =  $3.7 \pm 0.9$ ; Phone mean  $\pm$  SD =  $3.3 \pm 1$ ). A user found adding subtasks a bit complicated on

mobile devices, as it required many steps. He suggested supporting swiping gesture in the desired direction.

*Specifying objects.* (Desktop mean  $\pm$  SD =  $4 \pm 0.5$ ; Tablet mean  $\pm$  SD =  $3.7 \pm 1.2$ ; Phone mean  $\pm$  SD =  $3.2 \pm 1.6$ ). A user noticed that, on mobile devices, to specify task objects he needed to navigate through several windows. Another one suggested allowing reusing objects across tasks.

*Specifying preconditions.* (Desktop mean  $\pm$  SD =  $3.7 \pm 0.7$ ; Tablet mean  $\pm$  SD =  $3.6 \pm 1.2$ ; Phone mean  $\pm$  SD =  $3.2 \pm 1.5$ ).

*The least usable functionality.* On desktop, it was rearranging task models, e.g. inserting a child or a sibling task, which on desktop is done from the top menu.

*Overall rating.* (Desktop mean  $\pm$  SD =  $3.7 \pm 0.5$ ; Tablet mean  $\pm$  SD =  $3.8 \pm 1$ ; Phone mean  $\pm$  SD =  $3.1 \pm 1.5$ ). Many users declared to prefer the mobile version due to its better intuitiveness, in spite of well-known limitations in screen size. Ratings for the various aspects on desktop and tablet were similar, while for the smartphone version were lower.

### Evaluation with Students

Students received 1,5 hour lesson on CTT and related concepts. The test was partially done during a lesson and partially at home. They used both CTTE (on the desktop) and Responsive CTT (on the smartphone).

#### Participants and Tasks

20 students (16 females) aged 22 to 30 ( $M = 24.4$ ,  $SD = 2$ ) participated. The vast majority of users use the smartphone daily, 2 use it rarely, all use desktop every day. 4 users have prior experience with CTTE, one used Enterprise Architect. Users were asked to: i) create 2 task models (manage a library; arrange a holiday); ii) send the files created back to evaluators; iii) fill in the questionnaire.

#### Results

*Creating a task.* (Desktop mean  $\pm$  SD =  $3.6 \pm 0.8$ ; Phone mean  $\pm$  SD =  $3.3 \pm 1$ ). On phone, about half of them (9 out of 20) found usable the support for creating a task.

*Editing a task.* (Desktop mean  $\pm$  SD =  $3.6 \pm 0.8$ ; Phone mean  $\pm$  SD =  $3.2 \pm 0.8$ ). The most recurring issues were cutting/pasting subtrees, and also the zooming (on phone).

*Specifying objects.* (Desktop mean  $\pm$  SD =  $3.5 \pm 0.8$ ; Phone mean  $\pm$  SD =  $3.2 \pm 0.9$ ). On smartphone one user reported difficulties in selecting and deleting objects.

*Specifying preconditions.* (Desktop mean  $\pm$  SD =  $3.3 \pm 0.8$ ; Phone mean  $\pm$  SD =  $3.1 \pm 0.9$ ). A user had difficulties with preconditions associated to objects used by different tasks.

*Control of the graphical editing.* (Desktop mean  $\pm$  SD =  $2.9 \pm 1$ ; Phone mean  $\pm$  SD =  $2.9 \pm 1$ ). On smartphone the visualisations were sometimes perceived cluttered.

*The least usable functionality.* On smartphone, for 7 users it was specifying pre-/post- conditions, especially when

associated to objects manipulated by different tasks. 4 users reported editing tasks, especially on smartphones.

*Overall rating.* (Desktop mean  $\pm$  SD =  $3.5 \pm 0.7$ ; Phone mean  $\pm$  SD =  $3.1 \pm 1$ ). One user said that CTTE requires high screen size and then the smartphone is not suitable for task modelling. Another user reported problems in “separating subtrees” and assigning values to objects.

*Device where users encountered the greatest difficulties.* The smartphone was the most problematic for 13 users. For desktop, one user had problems in managing the task model because of a lack of a complete view of all tasks/relations.

*Final remarks/suggestions.* On smartphones one user suggested improving the graphics to better separate the tasks, 2 users questioned the opportunity of using a smartphone for creating CTT models due to screen size.

### Discussion

Overall, the most problematic aspects were the degree of control of the graphical editing and the specification of preconditions. As for the first, it was judged similarly both on desktop and on mobile devices, which was a bit surprising since desktops have larger screen sizes that should facilitate users to get an overall view of the model. Some users questioned about the suitability of using smartphones for supporting a tool for task modelling. Improvements on the graphical aspects of the tool were suggested to allow for better control of the whole model. Additional problems were found in (un-)selecting the intended tasks (due to limited size of task icons). Regarding preconditions, there were two issues: a conceptual one, since specifying preconditions requires a deeper mastering of the concepts associated to task models; another one connected with the tool support associated with specifying preconditions, which was found not very intuitive. To sum up, the results of the test show that the Responsive CTT was appreciated. In the first experiment the tablet version seemed competitive with the desktop CTTE, while using the smartphone platform for modelling tasks remains debatable for various users.

### CONCLUSIONS AND FUTURE WORK

We have presented the design choices and the associated implementation of a new tool for specifying CTT task models in a responsive manner so as to adapt to interaction through touch mobile devices as well, also reporting on initial user testing. As reported in the paper, during the development of the tool a number of relevant design issues have been addressed.

To the best of our knowledge the tool is currently one of the first cross-device tools for task modelling ever developed with advanced features for smartphones as well. Initial user testing of the tool, while providing promising feedback, highlighted that desktops and tablets are more suitable for supporting task modelling than smartphones since modelling tasks is a medium/long-term, cognitively

demanding activity which is better performed when the supporting devices allows for performing it in a comfortable manner. Thus, among mobile devices, the smartphone has clear limitations (e.g. screen space), and task modelling on the go seems still premature, even if the increasing adoption of mobile technologies can make it more realistic in the near future. Nowadays, tablets appear better candidates, and we plan to further develop support for cross-device task modelling activities, and carry out further empirical evaluations to identify the most efficient solutions to support them.

The Responsive CTT tool can be accessed at <http://ctt.isti.cnr.it>.

Its cloud-based architecture will also be investigated for supporting the creation of communities interested in task modelling, and in particular in activities such as sharing task models and discussing their design.

### REFERENCES

1. Caffiau, S., Scapin, D., Girard, P., Baron, M., Jambon, F., Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interacting with Computers* 22, 6 (2010) 569-593
2. A. Cockburn, A. Karlson, B. Bederson, A Review of Overview+Detail, Zooming, and Focus+Context Interfaces, *ACM Computing Surveys*, Volume 41 Issue 1, December 2008, Article No. 2
3. Eggers, J., Hülsmann, A. & Szwillus, G., (2013). Aufgabenmodellierung am Multi-Touch-Tisch. In: Boll, S., Maaß, S. & Malaka, R. (Hrsg.), *Mensch & Computer 2013: Interaktive Vielfalt*. München: Oldenbourg Verlag. (S. 325-328).
4. Limbourg, Q., Vanderdonck, J., Michotte, B., Bouillon, L., and López-Jaquero, V. *UsiXML: A language supporting multi-path development of user interfaces*. *Engineering Human Computer Interaction and Interactive Systems*, (2005), 200–220.
5. Mori, G., Paternò, F., and Santoro, C., 2002. CTTE: support for developing and analyzing task models for interactive system design. *IEEE Trans. Softw. Eng.* 28, 8 (August 2002), 797-813.
6. Marcotte, E.: *Responsive Web Design, A Book Apart* (2011), <http://www.abookapart.com/products/responsive-web-design>
7. Martinie C., Palanque P., Winckler M.: Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. *INTERACT* (3) 2011: 589-609
8. Paternò, F., 2000. *Model-Based Design and Evaluation of Interactive Applications*. Springer Verlag.
9. Spano, L.D., and Fenu, G., 2014. IceTT: a responsive visualization for task models. In *Proceedings of EICS 2014*. ACM, New York, NY, USA, 197-200.