

Of Models, Rationales and Prototypes: Studying Designer Needs in an Airborne Maritime Surveillance Drawing Tool to Support Audio Communication

Catherine Letondal, Pierre-Yves Pillain, Emile Verdurand, Daniel Prun, Olivier Grisvard

▶ **To cite this version:**

Catherine Letondal, Pierre-Yves Pillain, Emile Verdurand, Daniel Prun, Olivier Grisvard. Of Models, Rationales and Prototypes: Studying Designer Needs in an Airborne Maritime Surveillance Drawing Tool to Support Audio Communication. HCI 2014, 28th BCS International Conference on Human-Computer Interaction, Sep 2014, Southport, United Kingdom. pp 8, 10.14236/ewic/hci2014.8 . hal-01205503

HAL Id: hal-01205503
https://enac.hal.science/hal-01205503

Submitted on 28 Apr 2020

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.

Of Models, Rationales and Prototypes: Studying Designer Needs in an Airborne Maritime Surveillance Drawing Tool to Support Audio Communication

Catherine Letondal
Univ. Toulouse, ENAC

31055 Toulouse, France
catherine.letondal@enac.fr

Pierre-Yves Pillain Emile Verdurand
Télécom Bretagne
Lab-STICC, UMR CNRS 3192
29238 Brest, France
{py.pillain, emile.verdurand}@ telecom-bretagne.eu

Daniel Prun
Univ. Toulouse, ENAC

31055 Toulouse, France
daniel.prun@enac.fr

Olivier Grisvard
Télécom Bretagne / THALES
29238 Brest, France
olivier.grisvard@ fr.thalesgroup.com

In this work, we seek to understand the needs of interaction designers involved in industrial system engineering processes. While current research offers a set of methods and tools for them, we believe that more empirical user studies focusing on designers are needed, in particular to support how model-based activity analysis may inform their decisions. Our designers' need analysis is conducted through participatory design and contextual inquiry, and applied through a real use-case project: a distributed tactile tool for airborne maritime surveillance. Thanks to this study, we report on our insights on the usability problems and needs related in particular to scenario-based modeling, model-based design rationales and design-based model refinement.

Interactive System Design; System Engineering Empirical Studies; Model-based Engineering; Participatory Design; Scenario-based Design; Design Rationales.

1. INTRODUCTION

Improving interactive system design processes is the goal of a large part of the HCI community. This can be done either with tools or models that are meant to support this process, but also through the adaptation of system and software engineering processes to interactive systems. Generally, these approaches are rather prescriptive: they tend to propose solutions through these models and methods. In contrast, another field of research, human factors in software engineering, seeks to understand and specify the needs of the people involved in these processes, by conducting empirical studies. These studies however, are less often focused on *interactive* system design. Moreover, less research work reports on longitudinal studies of real processes, where concrete problems raised by interactive system design can be described and analyzed, and from which specific needs may emerge.

In this paper, we report on the analysis of designers' needs through a user-centered contextual study, that is based on an actual project. The domain of this project is an industrial application for maritime surveillance, where both system engineering and HCI methods or tools had to be used. Our intention was not to define a new engineering process, but, in a context that included model-based and model-driven approaches, to observe and analyze how the designers would use and appropriate these models. Based on the modeling and design outcomes of the project, we reflected with them on the expressiveness of task models as compared to concrete work scenarios, on their use as design rationales and on how design work might help refine models back. So, our analysis includes two levels: the maritime surveillance study, and a meta-level, the engineering study, where we also gathered explicit data from designers about their needs through contextual interviews and participatory workshops.

After describing relevant related work, we provide an overview of the context (section 3), and describe the two study settings (section 4). The core of the paper is in section 5 that describes the case study: the maritime surveillance application and the related modeling and design activities, and in section 6, that reports on the engineering study based on this case study, where we gathered information and experimented on methods and tools to better support designers' needs.

© The Authors. Published by BCS L&D Ltd.
Proceedings of the 28th BCS Human Computer Interaction Conference

2. RELATED WORK

This research draws on two main fields of research: the first one addresses the adaptation of system engineering to complex interactive systems, while the second investigates empirical studies and human factors in system engineering. Much research has been conducted on how to adapt software engineering methods to interactive software. Approaches encompass designing adapted formalisms such as CTT (ConcurTaskTrees) [21], UsiXML (USer Interface eXtended Markup Language) [16] or Petri nets, that can be used to model in details interactive systems including task but also AUI (the Abstract User Interface with states and transitions), as well as CUI (the Concrete UI, i.e presentation) elements [19]. Acknowledging the limits of MDE (Model-Driven engineering) for interactive systems [5], other approaches combine contextual studies and software engineering models such as [2], or transform informal scenarios into models such as [20] or [15] who rely on an ontology of model elements to guide this process and defines three method levels: manual classification, dictionary based and semantically based. Somewhat fewer approaches address system engineering adaptation to interactive systems, in particular [12] who integrates participatory design [3] and system engineering processes or [28] that analyzes the different backgrounds underlying both the system engineering and creative design processes on which HCI design rests. In contrast to these prescriptive approaches, empirical studies in software engineering seek to understand underlying problems, and analyze the activities and needs of process stakeholders [9,23]. These studies either focus on human factors [27] or on social and collaborative issues in development projects, often aiming at understanding, more than at improving processes through novel approaches [8]. Additionally, they generally do not address interactive system engineering, but rather general software development. So, in comparison to existing research, our work 1) seeks to improve processes in interactive system engineering, by 2) conducting empirical research about practitioners and experimenting methods.

3. CONTEXT

This work was undertaken within a research project, MEDUSA (MÉthode de conception de systèmes Dirigée par les modèles et orientée par les USAges), involving 3 academic teams, 1 large company involved in airborne systems, and 1 middle-size company that builds system engineering software tools. The project use case was a maritime surveillance application. In this section, we first briefly describe the application context, then we provide some background on the research project and its objectives.

3.1 Maritime Surveillance

In military maritime surveillance, tactical operators are aboard planes flying over strategic maritime zones (fig 1.a). Missions to be realized can either be civil (e.g. exclusive economic zone control, search and rescue, traffic, fisheries or pollution control) or military (surface warfare, anti-submarine warfare, targeting for cruise missiles, joint littoral warfare). These missions are operated through the intensive use of sensors providing information on the environment, and communications with command centers on ground, such as CECLANT (French Atlantic Marine operational center) or CROSS (French Surveillance & Rescue Regional Operational Center). Each operator is responsible for their own on-board equipment, i.e. the configuration and operation of their allocated sensors. As a team, their task is to establish and maintain the tactical situation, i.e. ideally a complete description of the present surface and submarine ships (identifications, positions, routes, behaviors, etc.). This tactical situation can then be used for carrying out other missions. A standard team in a Falcon 50 plane is composed of five operators: a mission commander (captain), a co-pilot, a radar operator (tactical coordinator), and two observers (INMARSAT observer, dispatcher-observer) (fig 1.b). One main human factor lies in the constraints the operators are subjected to. Indeed, missions usually involve safety risks, have to be performed in all weather conditions, are often time-constrained (through mission objectives, plane autonomy and especially flight speed) and are highly cooperative to share their representation of the current goal and situation.

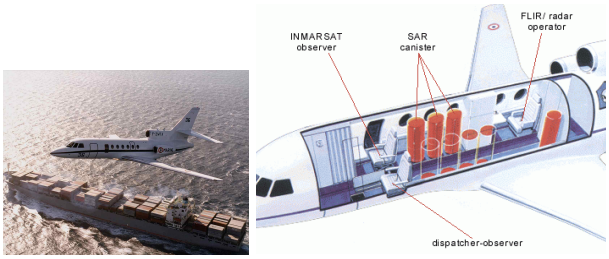


Figure 1: *Maritime surveillance. On the right: placement of crew members in the plane.*

3.2 Interactive System Engineering

This work took place within a project aiming at improving tools and methods for UI design in industrial projects, where system engineering methods are widely used. It is generally agreed that the design of complex interactive systems requires an interdisciplinary approach that mixes system engineering (SE) formal methods with those of HCI, in order to improve system adequacy to user

expectations. SE uses various formalisms to model domain, functions, tasks, logical and physical architectures. HCI favors human involvement (user, UI designer) and uses contextual observations, interviews, and participatory design workshops. Greater importance is given to less formal representations such as text, graphics, videos or storyboards, for instance by building scenarios [24,1] and low-fi prototypes [10]. Some issues remain open when trying to combine both approaches: how to ensure consistency between formal and informal representations from SE and HCI? how valuable are representations for the end-users (to enable early validation) and for the UI designers? are representations able to express useful information related to task, functions, AUI and CUI descriptions?

While these issues are central to the design of industrial engineering processes, our approach aims to first understand stakeholder – and mostly designer - needs related to the associated methods and tools. For this purpose, rather than directly proposing a new process model, we first wanted to analyze the existing activity during a real process.

4. STUDY SETTING

As explained in the introduction, the "engineering study" consisted in experimenting and applying an engineering process on a real case study, a maritime surveillance application, that we call in this paper the "application study". In the engineering study, our aim was to refine our needs and requirements for such a process. Some members of the team were both designers and HCI or SE researchers.

4.1 Maritime Surveillance Study

The study involved 4 UI designers and 4 analysts and was run over a period of approximately 1 year. Activities related to the application study took place either at the large company, or at one of the academic sites. We were unable to board such a plane due to military restrictions, and only one meeting was organized at the air and sea base with the maritime crew. For this study (fig. 2), we ran several contextual interviews [2], two 2-day participatory workshops first involving radar operators then the complete crew (fig 3), 1 design scenario walkthrough, 1 meeting to establish scenario data, 1 video-prototyping session with a domain expert (a former radar operator recruited at the large company), and a final 1-day evaluation and re-design workshop, again involving the whole crew. The interviews and workshops were filmed and transcribed. Section 5 "Application study" reports on this study and associated results in more detail.

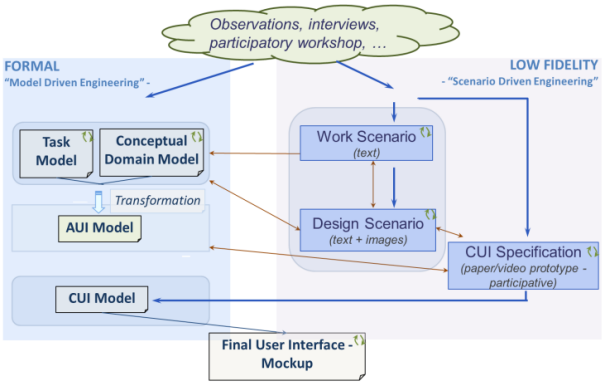


Figure 2: *The UI design activities.*

4.2 Interactive System Engineering Study

Based on this application study, and the empirical data it provided (actual models and UI design questions), we ran a study focusing on methods for requirement capture and design work. We carried out 3 interviews with 2 designers who did not belong to the project, and 3 interviews of designers from the large company who were directly involved in the project. Some results from these interviews were captured through work scenarios. 2 participatory workshops (brainstorming, storyboarding and video-prototyping) were also organized (7-8 participants from the project team). The team also ran a 2-day collaborative workshop involving from 5 to 7 participants (HCI designers and researchers, system engineers), where modeling and design tools were discussed and investigated in the context of the maritime case study. Section 6 "Engineering study" reports on this part of the work.

5. APPLICATION STUDY

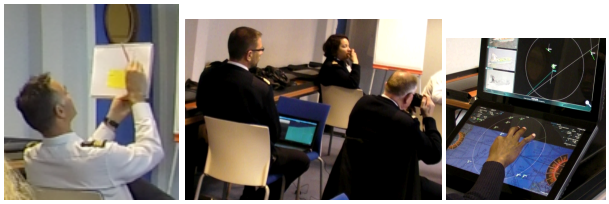


Figure 3: *Participatory Workshop: the crew plays a recent rescue scenario in the meeting room; as in a real mission, the pilot (b, middle) and captain (a) sit in front, the 2 observers (taking photos) in the middle (b), the radar operator sits in the back (c).*

Observations, together with interviews and their transcripts, enabled work and design scenarios to be built (see a work scenario in Table 1). A walkthrough of the scenarios was performed with an expert radar operator over the phone. Task and data models were built partly from work scenarios and directly from transcriptions, videos, direct observations and previous operational knowledge. The concrete UI (CUI) was specified extensively in

a document. It was built from the analysis of transcribed interviews and video-prototypes. The design scenarios were built from both this CUI and the narratives provided in the work scenarios. A final video-prototype was made by the designers and given to the interaction designers and programmers.

The study had two main phases: the first one focused on the radar operator activity and the associated touch-based application designed and developed by the large company; the second one started after having acknowledged during the 2-day workshop with the complete crew that the activity is highly collaborative. In order to be able to run the engineering study about this use-case project within a few months, we chose a reduced set of needs that were identified during this 2-day workshop: in a specific situation that proved particularly critical and stressful for the crew, when notified of a sailor over board, the crew has to interrupt their current mission and decide on a search strategy, a route pattern, in order to maximize the chances of finding the sailor. This decision is especially stressful when the crew has to accelerate the decision process, for instance when the event occurred in a close area and the transit time is short. During the interviews, the staff acted in the actual way they would collaboratively decide on a strategy – balancing between following the drift or in the opposite crossing it – by "picturing in the air" (cf. Table 1). After having the pilot confirm that they would actually do "just like that", and the radar operator testifying that he had indeed often used a pen and paper to represent the situation and show it to the captain, we proposed to switch to a prototyping mode using paper notebooks as if they had graphical tablets (fig 3.a). We then decided to design a collaborative drawing tool to help them build a common representation on the route pattern to follow.

5.1 Analyzing and Modeling the activity

Activity analyses included both formal modeling and ethnographic or scenario-based work.

Table 1: *Work scenario excerpt: "decide a route pattern".*

The captain consults his information on the remaining fuel then contacts the French CROSS: "We will finish the *bicycle wheel* in 1 hour and I have 2 hours and half fuel endurance before leaving the area." CROSS : "Ok, but in fact we have received some additional information: in fact the sailor did not fall into the sea in 1/2 hour ago but 12 hours ago." Radar operator: "If he had fell in 12 hours ago, he would have been subjected to the current and wind drifts, so we will have to request CROSS that they send us the drift types ..." Captain asks CROSS: "What was the boat's route?" CROSS answers: "090/ 5 knots". The radar operator notes the information in his notebook. Captain asks CROSS: "Do you have an idea of the type of drift over the area ...?" CROSS has no immediate information (and they also have a high workload – so he will probably have to wait). The radar operator thinks out aloud on the on-board phone: " ... if they saw him 12 hours ago and since they don't know anything, we must go back up the transit boat [draws the axis of the boat on his notebook] ... here we will imagine that he fell between these two branches, there is no drift , or that there is ..., and well it joined, … so in this case ... then I 'd rather go up the axis ... [hesitates a bit and draw an arrow in his notebook] ... the opposite axis of the boat ... and get down on the drift ... crenels , but parallel to the axis of the boat ..." Captain: "... and if he fell just 12 hours ago he could have drifted this far, whereas if he entered the water 5 minutes ago, he would have drift a lot less. It will be rather a search in cone pattern ... " The radar operator, after having completed a sketch of the pattern, went forward to show it to the commandant and said: "Well I suggest we go up the wake and then follow an *expanding circle* with spacing of 0.2 knots in the drift direction, to the south (facing the wind)." Captain: "Ok ... but wait ... then we may have ... depending on the length of the branches ... Look, we, by the time we have done this [takes the notebook and draws a line along one of the branches on the radar operator's sketch] ... and have done that [line in the opposite direction] our progression will actually be slower [draws an arrow on the research axis up to the end of the zone] than the man overboard who drifts [draws an arrow representing the displacement of the man over board]… we may be behind him. Dispatcher-observer: " Yeah, so, in fact you can do the opposite: at the most - if he has fallen 12 hours ago he will have drifted at ... 5, ... 8 knots." Captain: "... yes: let's just start at the maximum drift ... and we will go in the opposite direction - [taking back the drawing and tracing the pattern starting from the maximum drift] ... and then ... either we do a rectangle ... a *rectangle* search and start from the west and back to the east ... okay, making crenels like this [draws crenels] ... or we search ... we take an axis … we center the search like this ... [puts the pencil on the paper to show an axis] ... and we do something like that ... so always regularly spaced ..." Radar operator: "I'm ok for the second solution ..."

5.2.1 Scenarios

From the transcribed contextual interviews, we built work scenarios. Table 1 is an excerpt of a work scenario. The purpose of this scenario was to illustrate a stressful situation, i.e. to identify critical problems and to situate potential solutions. We use this scenario in this paper to illustrate our process.

5.2.2 Formal models

Formal models are essential in C2 complex critical systems such as those for airborne maritime surveillance [13]. The eFFBD (Enhanced Functional Flow Block Diagram) notation [17] that we used to specify the task model is widely used in system engineering. Tasks may be defined through functions, assembled with logical control operators (sequence, parallel …) and optional information flows. In a first phase, which was focused on the radar operator activity, we built the main task models of the radar operator work. After having shifted to the collaborative aspects, we decided, as described above, to focus on the situation where the crew has to decide on a pattern. The modeling was performed collaboratively by the designers and HCI researchers, while reading the scenario aloud (Table 1). The resulting task model describes how the crew, before taking the actual decision, first achieves an understanding of the problem through successive representations, built on estimations of the various parameters of the problem (drift, limits of the rescue zone, wake of the boat). As can be

seen in the scenario, this building of the problem space occurs through representation sharing and mutual understanding supported through the on board phone and through spatial elements conveyed through verbal or pictorial descriptions. We describe a part of the model below.

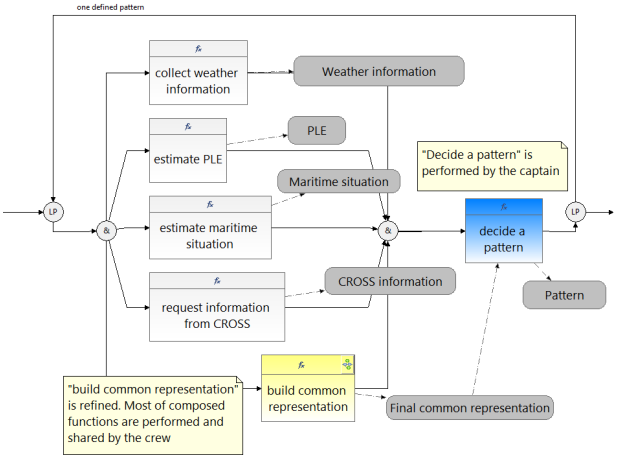


Figure 4. *Modeling: decide for a route pattern.*

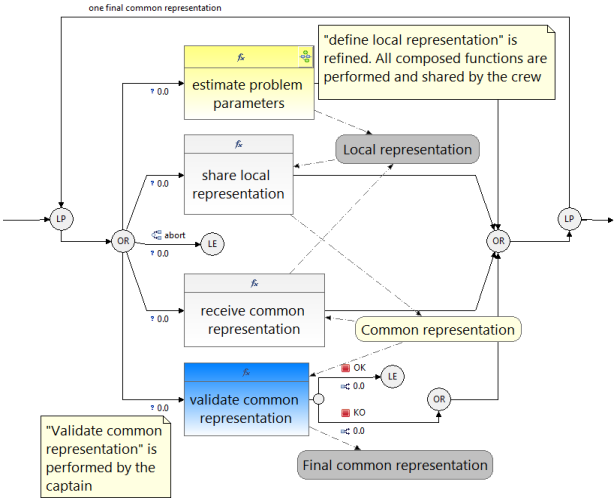


Figure 5. *Modeling: build common representation.*

Fig 4 represents the top level task: “decide for a route pattern”. While the final function “decide a pattern” is performed by the captain only, the “build common representation” task is fully collaborative (fig 5). This common representation helps manage the decision, together with other information such as the “PLE” (deadline for reaching the airport according to fuel and distance), weather information or maritime situation, which can be estimated by any crew member.

Fig 5 depicts how a “Local Representation” may be shared between crew members. The concept of a local representation attempts to capture the fact that some crew members – such as the radar operator in the scenario – actually use paper and pencil when there is a need to visualize the problem (the crew members can hardly see each other, see fig 1); it also represents the process of first internally reflecting on a given aspect and then speaking out to share it. The “estimate problem parameters” function provides the main data for this common representation of the problem (fig 6), with local representations such as area boundaries, boat trajectory or drift.

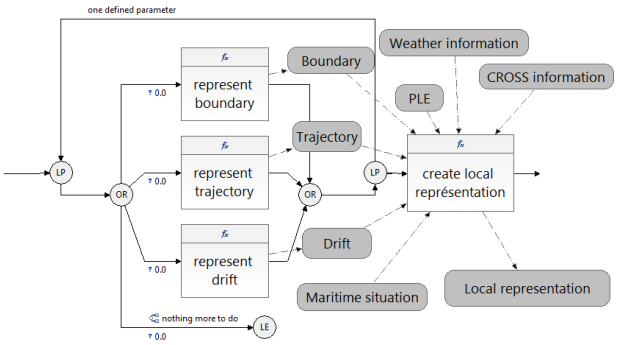


Figure 6. *Modeling: estimate problem parameters*

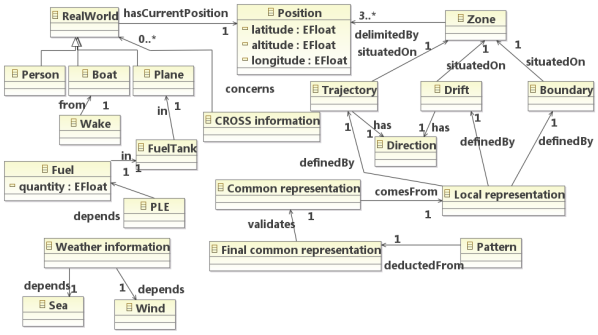


Figure 7. *Modeling: domain conceptual model*

In the eFFBD notation, information is represented as a flow. UML models were used for a more suitable notation (fig 7) and to generate the abstract user interface (AUI). Our specification of an AUI targets the notion of service provided to a user, and aims to provide an abstract description of service grouping in the UI, which may be deduced from the constraints specified in task model flows. For instance, tasks may be grouped either in sequences, such as for the "represent trajectory" and "create local representation" tasks, because there is a "trajectory" concept that is the output of the former and the input of the latter. They can also be grouped in parallel whenever they share the same source, such as the "represent boundary”, “represent trajectory” and “represent drift” tasks that share the “receive common representation” (see fig 5).

5.2 UI design

Regarding the user interface design, we ran a serie of specification and prototyping activities. Based on the complete CUI description that specifies the tools for each position in the plane and the design scenario, a video-prototype was built at the end of a participatory session with the expert radar operator associated with the project (fig 8.a): its

purpose was to provide a complete specification of the drawing tool, in the form of a list of features rather than in the form of a scenario.

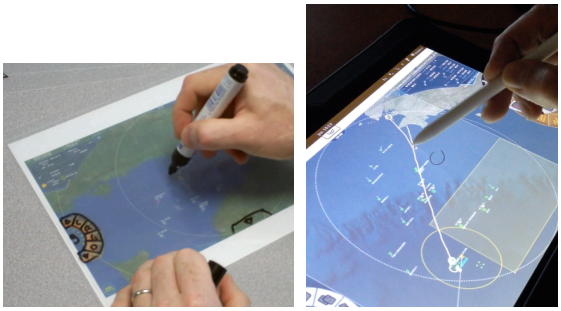


Figure 8. *a) Paper and video prototype; b) distributed drawing tool: during a workshop, the captain remotely explains to the radar operator where he suggests sending a rescue boat to repatriate injured people.*

Several issues were discussed regarding the design of this tool (Table 2). Some questions were shared editing issues [25], although this tool rather supports a distributed visual conversation, based on spatial elements, than aims to build a document.

Table 2: *Design questions.*

1. Colors
   a. Background
   b. Color range (wrp to luminosity, roles, sharing, spectrum, …)
   c. Color semantics *vs*. free choice through colored pens
2. Sharing representations
   a. Shared mode or shared drawings?
   b. Need of "private" strokes (as draft)
   c. Action(s) to trigger sharing
   d. Granularity: dots, strokes, objects, drawings?
   e. Synchronous or asynchronous sharing
3. Crew roles
   a. As filters
   b. Representation (colors, names, …)
4. Editing
   a. Selecting (or not)
   b. Erasing
   c. Free *vs*. object-based drawing
5. Managing drawings
   a. Single or several drawings?
   b. Saving or not? And inclusion in official mission reports?
   c. Deleting
   d. Accessing

This resulting prototype runs a on Linux tablet and enables the members of the crew to share their understanding of the situation and discuss the problem's parameters. They were able to run a 2-hour mission based on realistic data, and to use the drawing tool in the context of a real event (fig 8.b).

6. ENGINEERING STUDY

The objective of the engineering "meta" study is to further understand the needs of UI designers in a more explicit way, taking the maritime application as our use case. It was organized in two types of activities: 1) through a reflective study where we investigated how to foster the use of models to help designers and 2) through complementary interviews of designers, and participatory workshops to explore tools and methods to better support the system engineering process. This study was also reflexive because 5 persons among UI designers and analysts of the maritime study being also HCI researchers, we were our own users. The question whether this setting is valid and enables proper analysis or design is addressed by [8], who refers to the idea of "reflective software development" where researchers become stakeholders involved in software projects, and where software practitioners also reflect on their activity and how to enhance it – a setting through which empirical research gets an actual opportunity to improve processes.

6.1 Reflexive analysis on our system engineering process

During the project, we reflected on the design of the undergoing process. In particular, we discussed supporting our two main requirements: early validation from users, and maintaining consistency throughout the process, as explained in section 3.2. The first requirement is fulfilled through contextual interviews and participatory design that we do not discuss here because this approach is widely acknowledged [3]. The second requirement amounts to discuss how to relate requirement and design artefacts, in particular models and design elements, and also to iterate and refine each step. In this subsection, we reflect on three candidate methods to support linking and consistency: scenario-based modeling, model-based design rationales, and design-based model refinement.

6.2.1 Scenario-based task modeling

Task and data models such as the ones described in the previous section were built during a collaborative 1-day session, based on the walkthrough of the work scenario of Table 1. The scenario was read aloud and the modeling was performed verbally while drawing on a whiteboard and entering the sketched models in the MDWorkbench tool [31]. The discussions were recorded, transcribed and collaboratively analyzed. While the idea of building a task model from a scenario is not new, the scenario we chose was particularly specific and concrete – i.e. built from filmed and transcribed observations, as opposed to typical use-cases that depict a nominal and sometimes abstract procedure. Since our scenario was not abstract this approach varies from text-based modeling methods such as [21], where tasks are directly drawn from the words given in the text: we actually had to abstract them into higher level functions (e.g when the text "he draws an arrow" turns into the "represent drift" function).

The limits of formal models, for instance to communicate with users, have been widely discussed (e.g. [2]). What we sought to identify is also the expression power of both types of representations, and the loss or gain of information that can occurs when transforming one – a concrete work scenario (Table 1) – into another (fig 4-6). The modeling of the corresponding task diagram raised several issues.

Representing collaboration. To represent collaboration at a proper level, we had 2 solutions. The first was to define a collaborative task as an abstract task: for instance "build common representation" is specified as an abstract task – no roles have been allocated, but we still had to informally specify that some representations of the problem had to change from "local" to "shared" state. An alternative was to define a collaborative task with multiple allocations, which is not supported by the eFFBD notation; this enforces to specify as many "allocated tasks" as there are different roles, e.g. a "build radar operator representation", a "build captain representation", etc. This constitutes an over-specification, since each member of the crew is flexibly contributing to build this common representation. This might be better addressed by role or agent "lanes" (columns) as in GTA (Groupware Task Analysis) [30].

Representing communication. The notation enforces to represent a communication task as at least two sub-tasks: emission, reception, so that one often has to specify parallel (i.e. linked by '&') communication tasks – since users often exchange information orally in addition to performing other tasks. This may overload the diagrams, in particular in highly collaborative and multi-modal contexts. In contrast, there is no need to specify communication sub-tasks in a textual scenario: there is a tacit understanding that when something is said, it is heard – and that it should be specified only when it is not. Notations such as CUA (Collaboration Usability Analysis) [22] or coding schemes for interaction pattern analysis, as used in [6], integrate the communication types and mechanisms, and through this, both reduce the diagram complexity and enable a more informative specification.

Representing information scope. There was no support to represent the scope of a piece of information. One has to either depict information status informally (e.g. "local" in the name of the flow), or represent explicit dataflows among roles, which enforces allocation (see above item 2.b). Notably, neither GTA nor CUA provide any notation for information scope among team members.

Cognitive and emotional states. Scenarios are free to report about the cognitive or emotional state of the persons – in our case about the stress brought by their deadline for taking the least poor decision. Scenarios are also known for providing a rich context that lets the readers, including designers, understand what actually happened [1]: we observed such phases of sense making happening during our modeling session. It occurred when reading aloud the part of the scenario saying that they were in an anxious state when missing critical information from CROSS (fig 4). Several participants of the modeling sessions actually questioned including cognitive tasks, let alone cognitive states in the formal models.

Informal actions. Other issues were related to the type of specification afforded by the notation [1]. Notably, informal actions in the scenario, such as drawing a stroke to externalize a concern, turned, while modeling, into a task of "representing" or even "defining". This shows a level of consciousness or commitment that is higher than just externalizing. Targeting abstraction and formal expression may lead to this kind of normative specification level. This also raises the question about when to stop modeling.

Explicit vs. implicit control. Our work scenario started with the alert received by the radio-observer, that there was a shipwrecked sailor (part not shown in Table 1). Our scenario did not explicitly specify, when the alert arises, whether the task of "suspending current mission" should wait for the task "reception of the alert" to be completed ; this is probably most often decided based on an acceptable common sense but also sometimes based on the designer's domain knowledge. Abstract modeling enforces explicit statements and questioning, even though some collaborative task modeling frameworks [30,22] do not enforce to specify task order when there is none.

6.2.2 Model-based design rationales

A second type of reflection was focused on the use of task models as design rationales, in order to articulate and decide on design questions (see Table 2). Such method relates to claim analysis [4], except that we based our rationales on models. It also stands for the design rationality that is described by [26]. The model-based design rationales were discussed with three designers of the maritime application, and the discussion recorded. According to them, using models for discussing and articulating design questions makes sense and they found the notation easy to understand. In particular, they appreciated that the models provide them with a clear understanding of data flows. We provide an excerpt of some rationales below (indices refer to items in Table 2).

Colors (1.b, 1.c). Available colors were first decided according to environmental constraints (luminosity, etc.) (1.a, 1.b). We then chose to let the users allocate freely available colors rather than associating a color to a role, or a sharing type (1.d). As for these semantics, the task model does not

directly provide any support, but it both a) discards explicit roles, b) acknowledges for 2 sharing modes - which may advocate some flexibility and not enforcing color attribution for one of each.

Sharing representations (2.d, 2.e). The task model supports synchronous sharing (2.e) at the stroke level, rather than at the drawing level (2.d): as specified in fig 6, estimating problem constraints - drift or boundaries - which are mere strokes, and sharing each estimation requires enabling this type of display time congruence [25], rather than waiting for a whole drawing to be completed.

Managing drawings (5.a). The possibility of creating and accessing more than one drawing (5.a) supports several aspects, such as enabling comparisons between hypotheses, having alternative ideas or even disagreements, or when the context radically changes because a new piece of information arrives from CROSS and a brand new strategy has to be found. The task model supports the latter (with a "modify pattern" task after validation that is not shown in figures 4-6). Regarding the rise of disagreements within the crew, the transcribed interviews provide quite helpful insights that can be directly conveyed within a textual scenario: the radar operator explained when asked about potential disagreements: "Well … we do not disagree … of course we are different… it's more that … while we reflect by discussing ... we reach an agreement…". So the idea is rather to reach an agreement from various complementary standpoints than to negotiate a decision starting from conflicting views. This quite subtle difference can hardly be conveyed by a task model, and yet provides an understanding or even a rationale not only for the type of sharing, but also for the type of drawing management that is needed, i.e. something quite light, not needing sophisticated version or navigation mechanisms.

We do not seek here to provide a detailed analysis of design rationale cognitive processes as in [32]: we rather provide a case showing how formal models may contribute to the decision process.

6.2.3 Design-based model refinement

Design discussions, such as the ones depicted above, enable to iterate on modeling and progressively consolidate them [4]. For instance, the discussion on the sharing level led to a reflection on the representation tasks (fig 6), currently limited to 3 definite objects. Some flexibility was missing in our model to describe the fact that the crew should be able to represent and estimate unanticipated problem parameters or just take notes. Another refinement was to add a signature attribute to the "local representation" concept, when we discussed having colors representing roles (1.c). A final example is when we discussed selecting elements of the drawing (4.a): designating something in the representation while building a common representation was not taken into account. A variant of this feature was discussed during the first scenario walkthrough, in the form of non-persistent strokes. While selection is a current feature in general-purpose drawing tools, this was not considered as a priority, probably first for cost reasons, but perhaps also because there is no strong need for it. The designers of the large company know that the crew has an on board collective phone, which is the current medium that they use for this type of decision, and through which they designate problem parameters with quite precise names, so that the designation of drawing elements could be made verbally. In fact this on board phone, apart from being non visual, proved quite a good candidate metaphor for the collaborative drawing tool, as a cumulative non persistent flexible medium for externalizing and sharing everyone's representations. The use of design questions to refine models was acknowledged as valuable by the designers who suggested that the approach could even be used to check models coverage.

6.2 Interviews and participatory workshops involving designers

We ran complementary activities, interviews and participatory design workshops, focused on process stakeholders needs in the context of the maritime surveillance project. During 4 interviews with designers, several themes emerged.

Status and properties of representations. We observed a need to better manage the properties of the representations: their status, such as "decided", "to be investigated", "question", "proposal", "archive", "new", "alternative", "rejected", and their context of apparition, such as "who", "when", and their achievement level such as "draft", or "done". Keeping track of the fact that a representation is validated proved particularly important to them.

The artefact transformation process. There was a concern about where an artefact comes from, how far it corresponds to another artefact, and also the idea of an animated sequence (a "movie") of succeeding transformations.

Containers. Which structure or media could gather and integrate design artefacts? There is a need for a place, to reduce artefact dispersal, to "synchronize" them, or to link them consistently, to be able to retrieve them, and not to forget them when needed (see Table 3 for an example).

Action triggers. Which events trigger an activity (e.g a bug report, an email, a meeting, a deadline)?

Design rationales: Designers need libraries of design ideas, and their rationales.

During 2 participatory workshops involving designers, software architects and HCI researchers, we ran brainstorming and video-prototyping sessions (see Table 3). Several ideas, in particular linking various artefacts, advocate something similar to the tool described in [18], which does not however link to paper-based artefacts nor integrate within an IDE, let alone within the prototype.

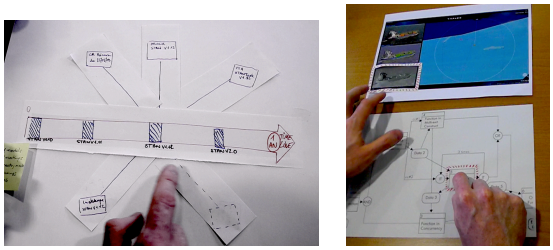


Figure 9. *a) timeline-based tool to support process; b) linking UI to models.*

Table 3: *Some of the low-fi prototypes and ideas.*

Using the prototype under development as a container to gather design data: rationales, status (see table 2), written or oral annotations during experimentations or meetings, etc. and most importantly to link them to UI elements.
Tools to support the process, such as a tool based on a timeline metaphor enabling navigation through requirement and design data (fig 9.a).
Support for integrating task models, scenarios into the IDE: linking design elements back to models (fig 9.b) or scenarios, running scenarios as scripts, and support to integrate pen & paper drawings into the design tools or IDE.
Meeting notes tool, such as a digital-pen based solutions (fig 9.c), to support the process (link to rationales, design decisions, UI components, persons in charge of a component).
Software development tools: introspective tools, tools for managing changes in the prototype, integrated graphic tools.

7. DISCUSSION

Running empirical and qualitative studies on system and software engineering processes is challenging [8], and using participatory design methods to study designers needs is not common [14]. We had the opportunity to profit from a real and complete case study involving a C2 maritime surveillance interactive application, where the process proved to be rather successful: initial design ideas converged rapidly, as illustrated through the enthusiastic reaction of the crew to the use of the tool during the final evaluation workshops. During this case study we both used formal models and applied HCI approaches, which gave us the possibility to experiment how task models may help designers in their real work, involving prototyping and interaction design issues (section 5.2). The automatic generation of full complex interfaces being still unrealistic, formal models were not used to automatically generate the CUI. However, they helped the designers in their specification of the CUI: the model-driven generated AUI identified valid interaction logics and helped the designers in their specification of the CUI, as reported in section 5.1; they also supported the long-term preservation of information obtained during the workshops, and to structure concrete exchanges among analysts and designers as design rationales (section 6.2.2). While links between analysis and design representations throughout the process can be made more explicit, through software support of integration and traceability as in [18] and as proposed during designers' workshops (section 6.1), this study advocates for sharing representations in a more cooperative process, where designers participate in observations, contextual interviews and collaborative modeling sessions (section 6.2.3), as suggested by [29] and [11]. In this view, models could be better appropriated as a shared view among stakeholders, as the "key representation" identified in [7] or as the "external record", "freeing the designer from having to remember everything about the work situation" advocated by [22].

8. CONCLUSION

This paper seeks to contribute to interactive system engineering by grounding the study of designers' needs in system engineering processes in a concrete use case. Current work includes running other processes to collect new data and confirm the first insights gained in our experimentation, in order to progress toward a better design environment complying with the needs that we have analyzed.

3. REFERENCES

1. Benyon, D. and Macaulay, C. 2004. A Scenario-based Design Method for Human-Centred Interaction Design. In *Scenarios and Use Cases. Stories through the System Life-Cycle*, Alexander I. and Maiden, N. (eds). Wiley. pp 211-234.
2. Beyer, H. and Holtzblatt, K. 1998. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Inc., San Francisco, CA, USA.
3. Bødker, S. and Iversen, O.S. Staging a professional participatory design practice: moving PD beyond the initial fascination of user involvement. In *Proc. NordiCHI '02*. ACM Press. pp 11-18.
4. Carroll, J.M., and Rosson, M.B. Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Tr. Inf. Syst.* 10, 2 (Apr 92), 181-212.
5. Coutaz, J. User interface plasticity: model driven engineering to the limit! In *Proc. EICS '10*. ACM Press (2010), 1-8.

6. Chauvin, C., Coppin, G. and Chéné, H. Analysis of the dynamics of common ground: a methodological proposal. In *Proc. ECCE '10*. ACM, 209-212.
7. Christensen M., Crabtree A., Heide Damm C., Hansen K. M., Lehrmann Madsen O., Marqvardsen P., Mogensen P., Sandvad E., Sloth L. and Thomsen M. The M.A.D Experience: Multiperspective Application Development in evolutionary prototyping. In *Proc. ECCOP'98*. Springer-Verlag, 13-40.
8. Dittrich, Y. 2002. Doing empirical research on software development: finding a path between understanding, intervention, and method development. In *Social thinking*, *Software Practice*, Dittrich, Y., Floyd, C. and Klischewski, R (Eds.). MIT Press, Cambridge, MA, USA 243-262.
9. Dittrich, Y., John, M., Singer, J. and Tessem, B. Editorial: For the Special issue on Qualitative Software Engineering Research. *Inf. Softw. Technol.* 49, 6 (June 2007), 531-539.
10. Ehn, P. and Kyng, M. 1992. Cardboard computers: mocking-it-up or hands-on the future. In *Design at work*, Joan Greenbaum and Morten Kyng (Eds.). L. Erlbaum Assoc. Inc., Hillsdale, NJ, USA 169-196.
11. Fitzpatrick, G. 2002. The Locales Framework: making social thinking accessible for software practitioners?, *Social Thinking, Software Practice*, Dittrich, Y., Floyd, C. & Klischewski, R. (Eds), MIT Press, 141-160.
12. Gaspard-Boulinc, H. and Wippler, JL. Integrating the human factor. *Complex systems and systems of systems engineering*, chap. 7, 380-392, Wiley, 2011
13. Jacobs, R. W. Model-Driven Development of Command and Control Capabilities For Joint and Coalition Warfare. *Command and Control Research and Technology Symposium*. June 15-17, 2004.
14. Janecek, P., Ratzer, A. & Mackay, W (1999). Petri Nets in Use: Redesigning Design CPN. In *Proc. of the Second Workshop on Practical Use of Coloured Petri Nets and Design/CPN*.
15. Lemaigre, C., Garcia, J. G., Vanderdonckt, J. Interface Model Elicitation from Textual Scenarios, *Human-Computer Interaction Symposium, IFIP International Federation for Information Processing*, Volume 272, 2008, pp 53-66.
16. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López-Jaquero, V. USIXML: A Language Supporting Multi-path Development of User Interfaces. In *Proc. DS-VIS 2004*: 200-220.
17. Long J. Relationships between common graphical representations in Systems Engineering. In *Proc. INCOSE 1995*, updated July 2002.
18. Memmel, T., Vanderdonckt, J. and Reiterer, H. Multi-fidelity User Interface Specifications. In *Proc. DSV-IS 2008*, 43-57.
19. Navarre, D., Palanque, P., Paternò, F., Santoro, C., and Bastide, R. A Tool Suite for Integrating Task and System Models through Scenarios. Interactive Systems: Design, Specification, and Verification, *LNCS* Vol. 2220, 2001, pp 88-113.
20. Paternò, F., Santoro, C. and Spano, L.D. 2012. Improving support for visual task modelling. In *Proc. HCSE'12*. Springer-Verlag, 299-306.
21. Paternò, F. and Mancini, C. Developing task models from informal scenarios. In *Ext. Abstracts CHI 1999*. ACM. 228-229.
22. Pinelle, D., Gutwin, C., and Greenber, S. Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ACM Trans. Comput.-Hum. Interact.* 10, 4 (December 2003), 281-311.
23. Robinson, H.M., J. Segal, and H. Sharp. Ethnographically-informed Empirical Studies of Software Practice. *Information and Software Technology, 2007. 49(6)*: pp. 540-551.
24. Rosson, M.B. and Carroll, J.M. 2002. Scenario-based design. In *The human-computer interaction handbook*, J.A. Jacko & A. Sears (Eds.). L. Erlbaum Associates Inc., Hillsdale, NJ, USA 1032-1050.
25. M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, and D. Tatar. 1987. WYSIWIS revised: early experiences with multiuser interfaces. *ACM Trans. Inf. Syst.* 5, 2 (April 1987), 147-167.
26. Stolterman, E. 2008. The Nature of Design Practice and Implications for Interaction Design Research. *International Journal of Design* 2:1.
27. Tomayko, J. E., & Hazzan, O. (2004). *Human aspects of software engineering* (electrical and computer engineering series), Charles River Media. Inc., Rockland, MA.
28. Vetting Wolf, T., Rode, JA., Sussman, J. and Wendy A. Kellogg. Dispelling "design" as the black art of CHI. In *Proc. CHI 2006*, ACM, 521-530.
29. Viller S. and Sommerville, I. Coherence: An Approach to Representing Ethnographic Analyses in Systems Design. *HUMAN-COMPUTER INTERACTION*, 1999, vol 14, pp. 9-41.
30. Van Welie M. & van der Veer, G.C. 2003. Chapter 19 - Groupware Task Analysis. In: *Handbook of Cognitive Task Design*. Erik Hollnagel (Ed). Erlbaum, Inc. pp 447-476.
31. http://sodius.com/products-overview/mdworkbench (visited on Dec 17th 2013)
32. Zannier, C. and Maurer, F. A qualitative empirical evaluation of design decisions. In Proceedings of HSSE '05, ACM.