DR 3.1: Trustworthy tactical SA support

Norbert Pahlke[*], Robert Maul[*], Tina Mioch[+], Joachim de Greeff[#], Miroslav Janicek[‰] and the TRADR consortium

*Fire Department Dortmund, Institute of Fire Service and Rescue
  Technology
+TNO, Soesterberg, The Netherlands
#Interactive Intelligence Group, Delft University of Technology,
  The Netherlands
‰DFKI, Saarbrucken, Germany

‹npahlke@stadtdo.de›

This report represents the results of WP 3 for the first year of TRADR. The overall objective of WP3 is to create situation awareness between the humans and robots, while jointly exploring a disaster area. Humans and robots shall learn from each other and share information to all team members. Grounded on common, persistent information actors shall learn to transfer knowledge from previous sorties to the next sortie. In Year 1 the work in WP3 focused on the creation of a situation awareness tool that makes the physical environment understandable for all human-robot team members.

Executive Summary

One of the objectives of TRADR is to create persistent situation awareness to human-robot teams. Situation awareness is a crucial element for actors in emergency management. It is necessary for an adequate response and tactical as well as strategic decision making process. In order to overcome this challenge, we investigate novel methods for situation-adaptive human-robot information exchange. That includes, among others, graphical user interfaces (GUIs).

This report presents our work in WP3 in Year 1. We worked on the TRADR display system (TDS) which constitutes the base for situation awareness. The TDS grounds on the experience with the NIFTI system, but is a follow-up instead of an only improved structure. The findings are presented in chapter 1.2 In addition to that, we examined the opportunity to integrate a speech interface. In the next step, this report presents our efforts to approach the TDS development from an end-users (tactical) point of view. Lastly, we report in chapter 1.3 how our findings relate to the existing state of the art.

Role of trustworthy tactical SA support in TRADR

The goal of trustworthy tactical SA support is to create a tool for information exchange between all team members over multiple sorties. This allows all team members of being aware of the situation at all time. For TRADR, this is crucial: Without having a platform to collect and share information, persistence cannot be achieved.

Secondly, this task contributes to collaborative mapping. To process data, the sourcing of information must have a common ground that will be developed further in the next periods.

Contribution to the TRADR scenarios and prototypes

The contributions of WP3 on adaptive multi-modal interaction in a human- robot team have been fully integrated into the system developed in WP7. WP3 contributes directly to the overall project vision of natural interactive human-

robot cooperation, and particularly to Objective 3, user-adaptive human-robot communication. In Year 1 we have covered different aspects of understanding, grounding, and producing communication (GUI with spoken dialogue via radiotelephone), with a focus on communication needs, arising in human-guided exploration of an industrial accident disaster area.

The prototype developed in NIFTi (TREX) was used as a starting point for exercise with end-users in the TRADR Joint Exercise (T-JEx, see Fig. 1). The usage of this prototype in a real training site of the Italian Fire Brigade lead to the decision to discontinue the use of TREX. Subsequently we started working on a TRADR Display System (TDS) and its architecture that adheres to the Situation Awareness Requirements by the end-users.

Fig. 1, TREx

1. Tasks, objectives, results

Work Package 3 contributes situation awareness of human-robot teams to TRADR. This is meant as the perception of the environment in time and space. Moreover, the recognition of environmental elements is another main aspect in general but also regarding the dynamic changes e.g. by acting. A human-robot

team shall be aware of these elements regarding their meaning in actual or future sorties and take particular action if necessary. The super ordinate objectives are:

- Promote trustworthy and relevant information about the physical environment for operational, tactical and strategic decisions corresponding to the three command levels (A)

- Provide a hierarchical representation of experiences which supports tactical decision making (B)

- Support awareness and decision management tool by managing known and unknown tactical information (C)

- Provide dialog-based support of communication and multi-modal interaction (D)

[13]

With these objectives, the WP contributes to the project-wide objective 2 (Persistent environment model) and objective 3 (Persistent model for human-robot teaming).

In this chapter, we describe the work that was executed in Year 1. Firstly, we describe the work plan for the first Year. In chapter 1.2, we explain our actual Work in WP3 with an excursion to planned work in Year 2. Afterwards, we give a description of how our work relates to the state of the art in chapter 1.3.

1.1.   Planned work

During the first year, WP3 had to design "Trustworthy tactical SA support" (T3.1). The concrete objectives were to develop a situation awareness tool that enables a general understanding of the physical environment by all team-members and meets the requirements of expectation management. It targets the exchange of knowledge between different actors and the transmission of knowledge between multiple sorties in a large-scale static disaster area, which means that the scenario changes will result from the progress of actions by the

operational units not by the incident itself. [13]. The goal is to make the physical environment understandable as far as this is necessary to achieve mission objectives. It was planned to include references to landmarks, static threats, spatial structure and tactical information. For example, occurrences like fires or places of victims shall be entered into the tool. Additionally, the system shall enable a routing of Infield Rescuer inside the area. This data should be collected in a persistent database. (WP1). Based on that, T3.1 should implement the collaborative mapping of a disaster scene. The main scope was to develop the prototype of a graphical user interface, which displays the collected data. This task is closely connected to tasks T2.1 and T5.1.

1.2.   Actual work performed

1.2.1. Overview of the development towards the TRADR Tactical Display System

To support Situation awareness (SA), for TRADR we are in the process of developing a user interface through a user-centered design perspective that allows for multi-modal interaction and can run on various displays. The aim of the TDS is to provide trustworthy and relevant tactical information about the physical environment and give access to a hierarchical representation of experiences to support tactical decision making, such as dynamic task allocation, (re)planning and coordination The TDS makes the handling of the robots comfortable concerning their processing status like an incident log. It will be designed to support guided (a)synchronous information exchange between distributed or co-located actors through multi-modal interaction (graphical UI and spoken dialogue). This guidance needs to be personalized and context-tailored, which we aim to achieve through the utilization of an agent-based framework developed in WP5 [1].

Because TRADR is a follow-up of the NIFTi project there was an existing UI which (partially) addressed the requirements of SA; this NIFTi UI is dubbed TREX. However, based on experiences from NIFTi, as well as end-user feedback and observations from the TRADR team, it was decided to discontinue the development of TREX and instead focus efforts on retaining those features that were good, while in addition going beyond what was possible in TREX. Towards this end, requirements were identified, and a first design of the TDS architecture was created. More details about this process are described in Appendix 1 [2].

1.2.2. Interaction Design Patterns for Coherent and Re-usable Shape Specifications of Human-Robot Collaboration.

We extended the methodology for situated Cognitive Engineering (sCE) by incorporating interaction design patterns into the methodology. So far, the sCE methodology provided a specification of the (functional) user requirements with the related scenarios, use cases and claims (i.e., a specification at the task level). Interaction design patterns provide a structured format to capture and share design knowledge at the communication level (i.e., the shape of the interaction).

The extended methodology has been applied to the development of human-robot cooperation in the urban search and rescue domain, or more specifically to team-awareness display functionalities, see Appendix 2 [3]. We found that a design specification can be valid on a task level, while the evaluation shows sub-optimal results because of a moderate communication level. Based on this evaluation result a design improvement on the communication level has been proposed without the need to adjust the task level design solution.

1.2.3. TDS speech interface

In Year 1, we performed the foundational work for integrating spoken language processing capabilities into the TRADR Display System. We formulated four design goals for building spoken language interfaces that are useful: flexibility, non-intrusiveness, responsiveness, and transparency. Since the standard paradigm for building spoken dialogue systems does not address these concerns well, we formulated an approach based on an object-oriented (conceptual) decomposition of the spoken language interface. This decomposition is similar to, and inspired by, the model-view-controller design pattern for building graphical user interfaces, but that apart from the basic objects it also reifies interactions between these objects. These aspects are then weaved together at runtime by a behavioral programming framework, utilizing a software transactional memory for orchestration. This allows us to re-use the basic building blocks and design their combinations in an incremental fashion (design-evaluate-revise), in line with the user-centric design methodology adopted by the TRADR project. For more details, see Annex 3 [4].

1.2.4. Outlook for Year 2: TDS development with tactical criteria

We end users analyzed our hierarchical structures and discussed our tactical approach in different scenarios to establish a harmonized operational process for disaster response (see Fig. 4). We also determined a ranking between the different tasks during an operation and differentiated between time critical,

critical and static operational phases. They are flatly characterized by saving life, saving environment and finishing the operation.

In DR7.1 we described our end user needs with view to the robots. How could we involve the information in our classical procedure, which is established over years? We have to analyse our processes again to adapt the new technology in a well-coordinated way.

Based on that we are preparing a field test at the Phoenix furnace where the Year 2 evaluation will take place. It is an additional test to the yearly exercise to get the chance for experiments independent of the predefined evaluation settings. It seems to be necessary because of the collected experience by using an UAV. We will share the live video from the UAV with the different command levels and analyse the reaction compared with the classical procedure. A discussion about the pros and cons and about requirements will complete the test series. The complex 3D structure of the old Phoenix furnace will be a special challenge with requirements related to the information collection (agent-based framework) as also information and expectation management (interaction design pattern). The TJEx GUI from NIFTi offered a GoogleMaps photo from the incident area. It doesn't inform the user about the current situation. Another suboptimal point was the nearly vertical perspective of the photo down to the ground. Both conditions wouldn't satisfy the end users requirements for operations at the furnace or other complex 3D surroundings. Figure 2 and 3 show pictures from the furnace under different angles. It is obviously that the vertical photo, which is necessary for the 3D model processing doesn't offer information in an adequate quality for the operation compared with the lateral photo, because of the missing information along the y axis. But for a complete overview from lateral perspective we would need a huge amount of pictures. So we have two concurrent interests. On the one hand the interest for a quick and adequate information to start the operation on a certain information level and on the other hand the interest for a 3D model which presents the incident area in one picture but with a delay of fifteen minutes and more, depending on the incident area size.

Fig. 2 vertical photo, no information along y axis

Fig. 3 lateral photo, only a small insight of the area

So we want to explore how and which information delivered from an UAV could support the situation assessment and improve the response along the command chain and what have to be done that the particular command level has a high efficient gain (see Fig.4). E.g. an IR who shall be guided throughout a complex industrial plant needs information about the destination point in relation to his own current position over all three x, y, and z axis especially when his freedom of movement is limited by breath protection. We want to extend our experience of what information at a disaster scene are needed by fire fighters, where do they

look at, how fast do information have to be available, and what is the best way to present the information for particular command levels. The results will flow into the design of the SA tool – TDS.
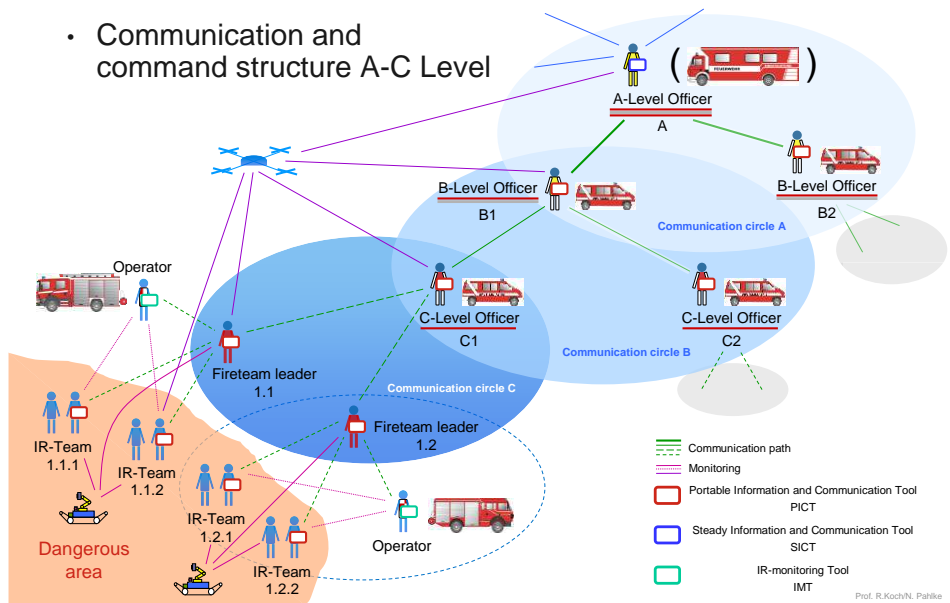

Fig.4, communication and command structure

The command chain bottom up:

Infield Rescuer-Team → Fire team Leader → C-Level Officer → B-Level Officer → A-Level Officer (crisis team)

1.3.   Relation to the state-of-the-art

1.3.1.   Interaction Design Patterns for Coherent and Re-usable Shape Specifications of Human-Robot Collaboration

Our proposal to use so-called *interaction design patterns*, complementary to the requirements specification of the functions and the information presentation is new. These patterns provide the designer a structured format to capture and share design knowledge for a recurring problem in a specific context with a common language for multidisciplinary teams. Similar to the justification of requirements via (task-level) claims, the interaction design patterns are justified via so-called (communication-level) premises.

Interaction design patterns have recently received considerable attention in the field of human computer interaction as a means for developing and communicating design knowledge to support good design pattern. Design patterns describe the core (key invariants) of a good design solution to a recurring problem in a specific context [5]. In general, patterns provide practice-based solutions accompanied by a theoretical account. Alexander's [6] philosophy of constructive, coherent and meaningful design in architecture, inspired the development of pattern languages in many other domains and application fields. These include, for example, Software Design Patterns [7] in the field of software engineering, Activity Patterns in the field of activity and ethnographic research, User Interface Design Patterns [8][9], Interaction Design Patterns [10] and Design Patterns for sociality in human-robot interaction [11] in the field of Human-Computer Interaction.

Dearden and Finlay [5] argue in their review that the discussion about a universal pattern format is still ongoing. We propose to integrate interaction design patterns into current functional specifications of use cases, user requirements and claims. As a complement, they should cover the embodiment (shape) of the requirements within the given context of the use cases. Therefore, we propose it should (at least) consist of title and ranking, the design problem (what, interaction intention), the context (related use case), the design rationale (why does it work, trade-offs, premises), the design solution (how), the related patterns

(at the same/different level of scale and abstraction, other context), and examples.

1.3.2. TDS speech interface

Our extension of the state-of-the-art with respect to spoken dialogue systems is threefold. First, by — pragmatically — seeing the problem as primarily that of software design (rather than that of artificial intelligence), we lower the threshold for applying software engineering techniques necessary for managing complexity in software systems. Second, de-composition of the system functionality along conceptual and processing lines increases the modularity of the design, enabling incremental design, development, and deployment. Third, by breaking the functionality down to elementary blocks, we are in a good position to experiment with processing the sensory input incrementally, including acting on such partial input.

With regard to the underlying framework for orchestrating the building blocks of our system, we extend the behavioral programming of Harel et al in two main areas. First, our system supports concurrency (where Harel et al's system assumes a single processing thread), thanks to the use of a software transactional memory. Second, our elementary units retain their original semantics (where Harel et al's are arbitrary labels), and this allows us to phrase conditions and constraints on the original data as provided by external processing modules. Both aspects are essential for a successful use of a behavioral programming framework in a real-world application. [12]

2. Annexes

2.1.   De Greeff (2015), "Overview of the development towards the TRADR Tactical Display System"

Bibliography Joachim de Greeff. ``Overview of the development towards the TRADR Tactical Display System'' Unpublished technical report. Delft University of Technology, the Netherlands, 2015

Abstract

In this report we describe the development of the TRADR Tactical Display System. Starting from an inhered system from the NIFTi project (TREX), we describe how this was used during the first joint evaluation session, what end-user feedback was, how requirements were identified. Then we describe the process of working towards a new TDS.

Relation to WP   Documents the development of the situation awareness interface so far, and thus directly contributes to T3.1.

Availability   Restricted. Not included in the public version of this deliverable.

2.2. Mioch, et al. (2014), "Interaction Design Patterns for Coherent and Re-usable Shape Specifications of Human-Robot Collaboration"

Bibliography Tina Mioch, Wietse Ledegang, Rosie Paulussen, Mark A. Neerincx, Jurriaan van Diggelen. Interaction Design Patterns for Coherent and Re-usable Shape Specifications of Human-Robot Collaboration. Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 75--83. Rome, Italy, 2014.

Abstract

Sharing and re-using design knowledge is a challenge for the diverse multi-disciplinary research and development teams that work on complex and highly automated systems. For this purpose, a situated Cognitive Engineering (sCE) methodology was proposed that specifies and assesses the functional user requirements with their design rationale in a coherent and concise way. This paper presents this approach for the development of human-robot collaboration, focusing on a recently added component: the application of interaction design patterns to capture and share design knowledge on the shape of the human-robot interaction (i.e., the communication level). The sCE case study in the urban search and rescue domain provided the specification and assessment of functions and shape of a team-awareness display. Twenty fire fighters participated as operator of a ground or aerial robot, in several realistic earth quake scenarios to assess the functions and shapes of this display in different settings. It showed that the functions (i.e., the task level requirements and rationale) were valid, while the shape (communication level) was (yet) suboptimal. Based on this evaluation result, a design improvement on the communication level has been proposed without the need to adjust the task-level design solution.

Relation to WP   Describes the methodology used to design the situation awareness interface, and thus directly contributes to T3.1.

Availability   Public. Included in the public version of this deliverable.

2.3   Janicek (2014), "An incremental design methodology for building speech-enabled multi modal systems"

Bibliography Miroslav Janicek. "An incremental design methodology for building speech-enabled multi modal systems". Unpublished technical report. Saarbrucken, Germany, 2014.

Abstract

In this paper we examine the problem of integrating spoken language processing into the TRADR tactical display system. In order to be useful, a speech interface must be responsive and flexible, and not get in the way of the user's needs. However, the traditional paradigm for building dialogue systems does not promise to deliver a good solution. As an alternative, we propose to build the interface using an incremental bottom-up design methodology based on an object-oriented decomposition of the problem, and the com- position of more complex behaviours out of smaller re-usable behavioural building blocks.

Relation to WP   Described the framework proposed for building the multimodal speech-enabled situation awareness interface, and thus directly contributes to T3.1.

Availability   Restricted. Not included in the public version of this deliverable.

References

[1]   K. Hindriks. Programming rational agents in GOAL. In A. El Fal-lah Seghrouchni, J. Dix, M. Dastani, and R. H. Bordini, editors, Multi-Agent Programming: Languages, Tools and Applications, pages 119-157.Springer, 2009.

[2]   J. de Greeff. Overview of the development towards the TRADR Tactical Display System. Unpublished technical report. Delft University of Technology, the Netherlands, 2015

[3]   T. Mioch, W. Ledegang, R. Paulussen, M. A. Neerincx, J. van Diggelen. Interaction Design Patterns for Coherent and Re-usable Shape Specifications of Human-Robot Collaboration. Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems, pp. 75--83. Rome, Italy, 2014.

[4]   M. Janicek. An incremental design methodology for building speech-enabled multi modal systems. Unpublished technical report. Saarbrucken, Germany, 2014

[5]   A. Dearden, J. Finlay. Pattern languages in hci: A critical review. Human–computer interaction 21, 1(2006), 49–102.

[6]   C. Alexander, S. Ishikawa, M. Silverstein. A Pattern Language: Towns, Buildings, Construction. Center for Environmental Structure Berkeley, Calif: Center for Environmental Structure series. Oxford University Press, USA, 1977.

[7]   E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design patterns: Elements of reusable object-oriented software. Addinson-Wesley Longman Inc. (1995).

[8]   J. Tidwell. Designing interfaces. O'Reilly, 2010.

[9]   M. van Welie, G.C. van Der Veer, and A. Eliens. Patterns as tools for user interface design. In Tools for Working with Guidelines, C. Farenc and J. Vanderdonckt, Eds. Springer, 2001, 313–324.

[10]   J. Borchers. O. A pattern approach to interaction design. AI &

SOCIETY 15, 4 (2001), 359–376.

[11]   P. H. Kahn, N. G. Freier, T. Kanda, H. Ishiguro, J. H. Ruckert, R. L. Severson, and S. K. Kane. Design patterns for sociality in human-robot interaction. In Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, ACM (2008), 97–104.

[12]   D. Harel, A. Marron, and G. Weiss. Behavioral programming. Communications of the ACM (2012), 55(7):90-100.

[13]   G. J. Kruijff. Long-Term Human-Robot Teaming for Robot-Assisted Disaster Response. EU FP7 ICT-609763, 2013, p.46.

Interaction Design Patterns for Coherent and Re-usable Shape Specifications of Human–Robot Collaboration

Tina Mioch, Wietse Ledegang, Rosie Paulissen, Mark A. Neerincx, Jurriaan van Diggelen
TNO
Kampweg 5, 3769 DE Soesterberg, The Netherlands
{tina.mioch, wietse.ledegang, rosie.paulissen, mark.neerincx, jurriaan.vandiggelen}@tno.nl

ABSTRACT

Sharing and re-using design knowledge is a challenge for the diverse multi-disciplinary research and development teams that work on complex and highly automated systems. For this purpose, a situated Cognitive Engineering (sCE) methodology was proposed that specifies and assesses the functional user requirements with their design rationale in a coherent and concise way. This paper presents this approach for the development of human-robot collaboration, focusing on a recently added component: the application of interaction design patterns to capture and share design knowledge on the shape of the human-robot interaction (i.e., the communication level). The sCE case study in the urban search and rescue domain provided the specification and assessment of functions and shape of a team-awareness display. Twenty fire fighters participated as operator of a ground or aerial robot, in several realistic earth quake scenarios to assess the functions and shapes of this display in different settings. It showed that the functions (i.e., the task level requirements and rationale) were valid, while the shape (communication level) was (yet) suboptimal. Based on this evaluation result, a design improvement on the communication level has been proposed without the need to adjust the task-level design solution.

Author Keywords

Cognitive engineering; Interaction design patterns;
Human-robot collaboration.

ACM Classification Keywords

H.5.2 User Interfaces: Evaluation/Methodology

INTRODUCTION

Urban search and rescue (USAR) missions are very stressful and consist of high-demand tasks, as the layout of the situation is often uncertain and dangerous situations can easily arise. Requirements may change during the development- and the application phase. This makes an iterative design process necessary, with continuous enhancements and evaluations, involving end-users and human factors experts. When developing personalized support systems for the USAR team, the social, cognitive, and affective state of the team members need to be taken into account. A common way to address the complexity of developing such systems is to use a cognitive engineering approach (We refer to Norman [16] and Vicente [23] for an overview).

In the NIFTi and TRADR project[1], we develop systems to improve human-robot cooperation in the USAR domain. Robots are part of the USAR team, share a common goal with their human team members, and have their own capabilities and responsibilities. This means they function as full-fledged team members. Yearly evaluations in a realistic environment with end-users are executed to be able to make sure the robots are built to be used in the context of the USAR domain by the eventual end-users.

When developing these kinds of systems, an iterative development process is necessary. After each cycle, the user requirements the system needs to fulfill are revisited, leading to validations and refinements. So far, the acquisition, specification and validation of requirements have been mainly focused on the functional level. However, the actual shape (i.e., the "look, feel and hearing" in the human-technology communication) of these functions affects their effectiveness substantially. Design specifications and assessments should therefore explicitly address both the functional and communication level [15]. In general, the cognitive engineering methodology should clearly distinguish two levels of specification, the function (or task) and shape (or communication) level, and explicating the design rationale on both levels. So far, a re-usable specification of the shape, with explicit and coherent relations to functional requirements and their design rationale, is lacking. This leads to the following research question:

- How to specify and evaluate the communication level of human-robot collaboration in a situated cognitive engineering methodology that already provides a sound task level specification and evaluation?

In this paper, we present a methodology for situated cognitive engineering, focusing on the application of reusable interaction design patterns for specifying and assessing the shape of human-robot interaction (i.e., the communication level). We

---

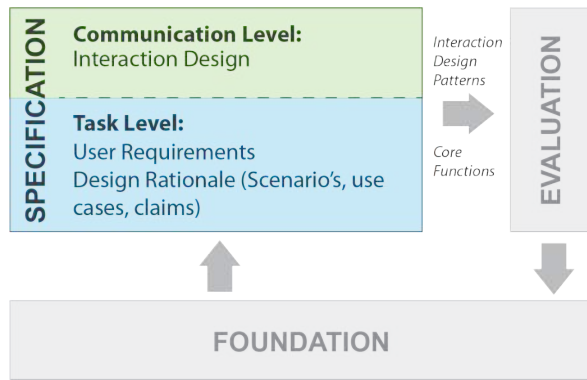[1]www.nifti.eu, www.tradr-project.eu

Figure 1. sCE design process.

demonstrate how the methodology is applied for the development of human-robot cooperation in the urban search and rescue domain, including the evaluation of the human-robot collaboration at the task and the communication-level.

SITUATED COGNITIVE ENGINEERING METHODOLOGY

The situated Cognitive Engineering (sCE) methodology is an iterative human-centered development process, aiming at an incremental development of advanced technology [14]. It consists of an iterative process of generation, evaluation, and refinement, and is based on earlier views on Cognitive Engineering (e.g., [11]). Figure 1 shows the general structure of the sCE methodology, consisting of three components: the foundation entails operational, human factors, and technological analyses to derive a sound and practical design rationale, the specification and maintenance of the requirements baseline, and the evaluation by means of simulation or a prototype, to validate and refine the requirements baseline. In the first component, foundational knowledge is described to identify actors, objectives, and contexts of the system and the (task) environment. This paper focuses on the second component, distinguishing two levels of specification: the task and communication level [15]. We choose for a minimal approach to keep the specification concise, whereas others propose more extensive abstraction levels (e.g., the four levels of the Unifying Reference Framework: task and concepts, abstract user interface, concrete user interface and final user interface, [4]).

Task level

The task level specification consists of the construction and maintenance of the requirements baseline, and the general design rationale that consists of the core functions, claims, and scenarios & use cases. The *core functions* are derived from the analyses of the first component. For each core function, one or more testable *claims* on its operational effects have to be specified; such a claim can be assessed unambiguously in the evaluation process. Both positive and negative claims can be specified. Furthermore, for each core function, one or more *requirements* have to be specified for the future system (i.e., what the system must do). *Use cases* describe the general behavioural requirements for software systems, and have

a specific specification format. According to the methodology, each use case should explicitly refer to one or more requirements and each requirement to one or more claims.

The requirements baseline can subsequently be justified according to its associated (task-level) claims. So far, the sCE methodology focused on the design and evaluation of the functional or task-level aspects of human-technology collaboration. This includes the mapping of situated user goals, information needs, and support needs to (adaptive) technology functions, information provisions and dialogue acts [15]. In this way, the system's functions and information provision are specified or assessed (i.e., the task level, such as user fit, work context and information needs conformance). However, human-robot collaboration should also be established well at the communication level, such as consistency, feedback and mode awareness, interaction load and user control [15].

Communication level

For the control of the functions and the presentation of the information, we propose to use so-called *interaction design patterns*, providing the designer a structured format to capture and share design knowledge for a recurring problem in a specific context with a common language for multidisciplinary teams. Similar to the justification of requirements via (task-level) claims, the interaction design patterns are justified via so-called (communication-level) premises.

Interaction design patterns have recently received considerable attention in the field of human computer interaction as a means for developing and communicating design knowledge to support good design [5]. Design patterns can be defined as follows: a pattern describes a problem that occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice [1]. Or in other words, a pattern provides a structured format to capture and share design knowledge. It describes the core (key invariants) of a good design solution to a recurring problem in a specific context [5]. A pattern language includes a collection of such patterns organized in a meaningful way.

Alexander's [1] philosophy of constructive, coherent and meaningful design in architecture, inspired the development of pattern languages in many other domains and application fields. These include, for example, Software Design Patterns [7] in the field of software engineering, Activity Patterns in the field of activity and ethnographic research, User Interface Design Patterns [17, 22], Interaction Design Patterns [3] and Design Patterns for sociality in human-robot interaction [12] in the field of Human-Computer Interaction.

In general, patterns provide practice-based solutions accompanied by a theoretical account. Design patterns emerge from 'best practices', specifying solutions of how a problem can be solved. Theory is necessary for the justification of a chosen design patterns. Theory can provide the rationale (e.g. a theoretical account) of why the solution works in a specific context, and what trade-offs are involved [5].
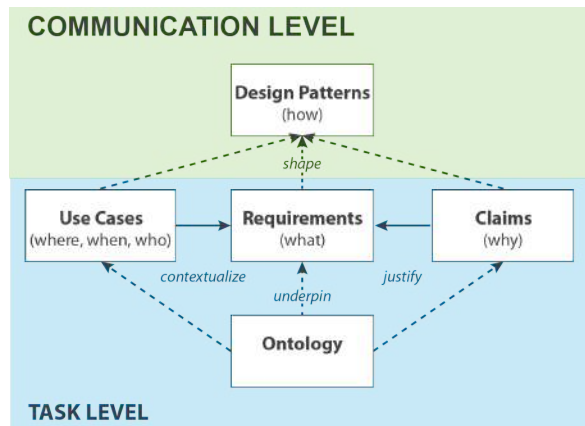
Figure 2. The sCE specification model.

Furthermore, a design pattern should show examples of successful implementation in applications. These examples play an important role in the validation of patterns. Successful implementation in applications in practice provides empirical evidence for the pattern's validity. Design Patterns have to deal with the dynamic aspects of an interaction, which could be represented, for example, by exemplifying storyboards [3].

Patterns can include different levels of scale and abstraction, varying from a general overview of the problem to specific characteristics for the solution.

Alexander's [1] patterns contain a unique name, the patterns context (including relation to other patterns), description of the design problem, design solution, how to implement the solution, rationale of why the design solution is good and in what context the pattern can be applied. Dearden and Finlay [5] argue in their review that the discussion about a universal pattern format is still ongoing. We propose to integrate interaction design patterns into current functional specifications of use cases, user requirements and claims. As a complement, they should cover the embodiment (shape) of the requirements within the given context of the use cases. Therefore, we propose it should (at least) consist of title and ranking, the design problem (what, interaction intention), the context (related use case), the design rationale (why does it work, trade-offs, premises), the design solution (how), the related patterns (at the same/different level of scale and abstraction, other context), and examples.

For an overview on how design patterns fit into the specification phase of the sCE methodology, please see Figure 2.

EXAMPLE DESIGN FOR HUMAN-ROBOT COOPERATION IN USAR MISSIONS

In the following, we will describe how behaviours and functions for a human-robot cooperation system are translated into reusable interaction design patterns at the communication level. To do this, we give an example specification of each step in the sCE methodology. We first shortly outline the theoretical foundation to establish the design rationale, followed by the specification of requirements, claims, and a suggestion for a concrete human-robot design example in the urban search and rescue domain. This design example is then improved as a first step towards a generic design pattern.

Foundation

Operational demands
After a disaster, robots can be needed for a safe and conscientious reconnaissance of the area. The scenario that has been specified with the end-users in the NIFTi project is an earthquake site, in which the rescue team's goal is to safely extract victims. It is suspected that several people are still alive in the area. The team needs to perform reconnaissance of the area, and identify and triage victims.

Human factors knowledge
Developing adequate situation awareness proves to be difficult during the reconnaissance, recover, and rescue operations of human-robot teams in disaster areas. Robot operators spend up to 60% of their time attempting to establish and maintain situation awareness, leaving little time for the operation of the robot or the visual search for victims. In addition to the need for cognitive load reduction, there is a clear need of theoretically and empirically founded, design proposals for integrated, context-sensitive situation maps [9, 10]. In addition, team awareness needs to be supported. In the NIFTi project, the team consists of at least two robotic team members (UGV, Unmanned Ground Vehicle, and UAV, Unmanned Aerial Vehicle), and 4 human team members (UGV-operator, UAV-operator, in-field rescuer, and mission commander). All these actors can be sources of valuable information to the professionals that work in this environment, but forwarding all information to everybody will cause unmanageable overload and interruption.

Technological innovation
All team members have access to the team awareness display. The team awareness system is tailored to support teams of professionals in their complex task environments [6]. Such environments are characterized by a large number of humans, networked computing devices, sensors, and possibly robots working together. All these actors can be sources of valuable information to the professionals that work in this environment, but forwarding all information to everybody will cause unmanageable overload and interruptions. To make optimal use of the vast amounts of digitally stored information we need a system that delivers the Right Message at the Right Moment in the Right Modality, or $(RM)^3$ [20].

Such a system could be based on a set of extendible OWL ontologies which specify the formats in which information can be represented and shared.

In addition, the envisioned system can pro-actively send information to a user, but a user can also request information. By allowing a user to express his or her information needs, the system can return appropriate pieces of existing information, or even actively create new relevant information (e.g. by asking other users or sensors to share information).

The system adapts its information supply to its user. For example, people in different roles receive information that fits
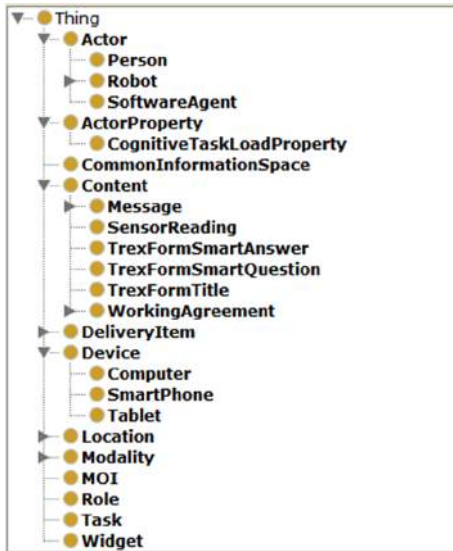
Figure 3. Part of the NIFTi Ontology.

Step nr.	Description	Requ.nr.
1	In-field rescuer hears victim scream, it appears to come from no-go area.	
2	In-field rescuer adds SUSPECTED victim to team awareness display and adds 'smart question'.	R048, R097
3	The Mission commander sees the SUSPECTED victim on the team awareness display, and gives the UGV operator command to investigate the victim.	
4	UGV-operator has a look at the current information and the 'smart question' concerning victim in no-go area.	R098
5	UGV-operator indicates waypoint and UGV proceeds to area.	R085

Table 1. Example use case for in-field rescuer and UGV cooperation.

their task description, people at a certain location receive information which is relevant for that location, and people with a high cognitive task load receive less information because they are assumed to have less information processing capacity.

The system supports multiple devices, such as visual displays (e.g. stationary computers, tablet computers, surface tables, smart phones), auditory devices (e.g. headphones), or tactile devices (e.g. a tactile vest). This means that the system must be capable of choosing the most appropriate interface, depending on type of information and the user.

The information that is shared between users can be facts, i.e. information describing the current situation. Another type of information is policies (information that describes the conditions under which collaboration takes place). An example of a policy is a no-go area. Such a policy may have its origin in a mutual working agreement, an order from the commanding officer, or a law. Policy-monitoring agents ensure that appropriate action is taken when the user breaks a policy, e.g. by notifying the user, or by preventing access.

Specification
Ontology
In information science, an ontology is defined as a specification of a conceptualization [8]. It describes the terms and

Requ.nr.	Name	Description	Step nr.
R097	Identifying and adding information needs.	The NIFTi system shall present up-to-date and combined overviews of available and missing information that infield rescuers and robot operators can adjust and complement (e.g., frames for victim's status that can be edited in a structured way).	Step 2
R098	Communicating information needs.	When a need for a specific information object has been identified (i.e., a "known unknown"), the NIFTi system shall notify the appropriate team member who can acquire this information (e.g., based on location and capabilities).	Step 4

Table 2. Example of a specification of requirements.

Claim nr.	C006
Name	Improvement of shared situation awareness
Description	Each team member has insight in the tasks, plans, and capabilities of the other team members. This also includes that 'unknowns' are made explicit, and that the 'right' team member knows (according to his capabilities) that there is still some information missing.
Positive	The team members can see the plans and tasks of the robot, which increases the trust in the system [trust questionnaire]. More optimal task distribution possible, as team members can see which 'unknowns' fit their capabilities [performance: time it takes for a user to act on new 'unknowns'?].
Negative	Having access to a team display in which a lot of team information is presented might lead to too much information, which leads to a higher workload to process the information. [workload questionnaire; heart rate and GSR measurement when processing the information]; [usability questionnaire on system]
Relations	R097, R098

Table 3. Example of a specification of a claim.

concepts and relations that are used in a certain domain. As argued in [19], ontologies are used for a variety of purposes:

- Communication between people with different needs and viewpoints arising from their different context. Domain knowledge of the fire fighters can be shared with a multi-discplinary team, supporting common and consistent understanding.

- Inter-operability between systems (e.g. databases using the same database schema's). Human-robot systems consist of many components which can have dedicated (sometimes temporarily) databases.

- System engineering benefits:
    - Re-usability. A shared understanding is the basis of the formal encoding of the important entities in a computer.
    - Automatic verification. A formally specified ontology can be used to automatically detect inconsistencies, serving as a verification of the domain analysis (e.g., for the identification of a fire or triage of a victim)
    - Specification. A shared understanding can assist the process of identifying requirements and defining a specification in IT system design.

Based on literature research, the scenario analyses and discussions with NIFTi end-users, we formulated an ontology

for the human-robot collaboration in the NIFTi-project. A part of the NIFTi ontology is presented in Figure 3.

Use Case

The use cases provide a (formal) contextualization (conditions, scope) in which the requirements are applicable (*when* the requirements apply). For a simplified example of the specification of a use case, see Table 1.

Requirements

The requirements describe *what* the system shall do. For a simplified example, see Table 2.

Claims

The claims provide the justification behind the requirements (i.e., *why* the requirement is important). Claims should always refer to evaluation methods or tools (such as performance time measurements and user questionnaires). For a brief example, see Table 3.

Interaction Design Patterns

The interaction design patterns provide a (formal) description of the shape of the requirements, as shown in Figure 2. The specification of the design patterns is explained below.

Name The name of the design pattern should provide a meaningful description that indicates the essence of the pattern.

Ranking The ranking should indicate the validity of the patterns premise. It can help the reader to distinguish early pattern ideas from patterns confirmed in practice [3].

Design problem The design problem describes the design problem in terms of the interaction intention (the effect on the user and/or user interaction with the system and/or other parties). The intention of an interaction can be extracted from the user requirements.

Use case, requirement, claim Here, the corresponding use case, requirements, and claim should be specified.

Context The context describes the characteristics of the tasks, the users, and the environment for which the pattern can be applied. This should provide the designer insight in when the design pattern can be used, and when the design pattern is less suitable. The use cases already provide the situational factors (e.g., dialogue partner(s), physical and social context, interaction platform, and dialogue context) that influence the design solution (specific embodiment of the dialogue). The design pattern should only list the contextual characteristics that determine in what situation the design solution can be applied.

Design solution The design solution provides a concrete description of the solution for the design problem. This encompasses the specific shape of the dialogue by describing what characteristics express the intended interaction within the given context, e.g., which verbal and non-verbal communication should be used, which dialogue rules should be followed. Only the core of the solution should be described, references to other relevant patterns can be used.

Design pattern level According to Woods [24], interface design can be assessed at different levels: workspace, views, forms, fragments, atoms and pixels, where each of these levels builds on the design decisions of the level below it. Accordingly, we describe Interaction Design Patterns at these levels of abstraction, although especially the high-levels are more useful to describe generally applicable and reusable design patterns. Furthermore, patterns at different levels of abstraction can be related in a hierarchical way, e.g., a general high-level pattern describes the sharing and handling of (un)knowns, while a lower level child-pattern covers the specifics for interactions on a mobile phone, tablet or touch table.

Design rationale The rationale provides insight in how the design pattern works, why it works and how it is based on underlying principles and mechanisms. It provides a convincing argumentation on the effects of the chosen design solution, including trade-offs. It includes premises that may need empirical validation.

Examples The examples should show successful uses of the pattern (e.g., best practices). It shows how the pattern can manifest itself differently in various 'real-life' applications.

Related patterns Links to any related patterns should be mentioned here. For example, a parent pattern (similar interaction intention, higher in the abstraction hierarchy), sister pattern (similar interaction intention, same abstraction level) and/or other relating patterns (different interaction intention, but in another way related to context and/or product characteristics of the design solution).

For the NIFTi team awareness system, the following two example Interaction Design Patterns are worked out: Explicit unknowns and Area policies. It must be noted that the two examples both consist of a set of individual interaction design patterns that together form the total design solution.

1. Explicit unknowns (for a screenshot of the design, see Figure 4). In the NIFTi team awareness system, users have the possibility to identify information that is unknown or indirectly submit a request for information to others.

    (a) Raise an explicit unknown
    (b) Notification that an Explicit Unknown is raised, based on relevant stakeholder and priority (specified in Table 4)
    (c) Answering an Explicit Unknown
    (d) Notification that the Explicit Unknown is answered

2. Area policies (for a screenshot of the design, see Figure 5). In the NIFTi team awareness system, an area with a specific policy (e.g. no-go area) can be defined and visualized on the digital map, while its policy will be enforced to in-field users of the system.

    (a) Creation of an Area Policy
    (b) Visualization of an Area Policy
    (c) Enforcement of an Area Policy

In Table 4, the interaction design pattern is worked out for 'Notification that an Explicit Unknown is raised, based on relevant stakeholder and priority'. Note that the interaction

design pattern worked out in Table 4 does not concern the total Explicit Unknown design solution, but only a subset.
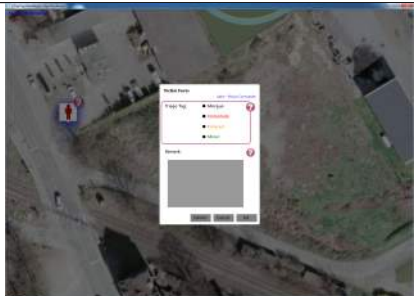
Title	Notification that an Explicit Unknown is raised, based on relevant stakeholder and priority
Ranking	1b
Design problem (what)	Make other relevant stakeholder aware of required information that is missing, its location and priority
Use case (context, use when)	Person1 finds a victim and creates a new victim icon at the specific location on the map display. But, because he cannot (completely) fill the required information fields, he raises an Explicit Unknown to make other people aware of the information need. Since the information request concerns an important triage, a medic is notified with high priority.
Requirement	- Identifying and adding information needs - Communicating information needs
Claim	- Improvement of shared situation awareness
Design solution (how)	- An explicit unknown is visualized with a question-mark icon, which is presented both at the relevant information field and at the specific location on the map. - If an explicit unknown is relevant to the user the question-mark icon on the map display is colored red. If the explicit unknown is relevant to other stakeholders, the icon is colored grey. - If an explicit unknown is relevant to the user and has high priority the icon is enlarged, compared to the normal icon size.
Design rationale (why, premises)	By showing the explicit unknown with a recognizable and dedicated element of information at a specific location on the map, it becomes clear where there is a request for information, to whom it applies, and what is the level of priority.
Design pattern level (workspace, views, forms, fragments, atoms, pixels)	Forms
Example, files, selected	
Related patterns	1a - Raise an explicit unknown 1c - Answering an Explicit Unknown 1d - Notification that the Explicit Unknown is answered

Table 4. Interaction Design Pattern for '1b - Notification that an Explicit Unknown is raised, based on relevant user and priority'

EVALUATION

Setup of evaluation

An evaluation has been conducted, in which a realistic team task in the urban search and rescue domain was executed by fire fighters in cooperation with robots. Each participant performed the experiment once, either in the role of the robot (UGV or UAV) operator, mission commander, or as an infield rescuer. There were 5 runs, with 20 participants. The participants were mostly male professional fire fighters (with the exception of one female fire fighter).

Task

The participants were asked to execute the following scenario: An earthquake occurred. The buildings in the area have
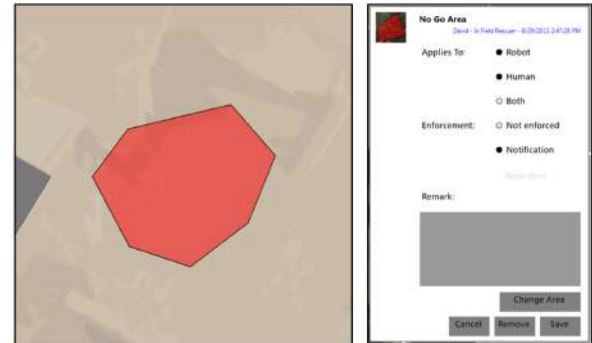

Figure 4. Explicit unknowns.

Figure 5. Area policies.

collapsed. One of the building has been a hospital, with suspected radioactive material present. Also, it is suspected that several people are still in the area, probably alive. Human victims need to be found as fast as possible, and the situation needs to be evaluated.

The scenario was a team reconnaissance task; the team consisted of an unmanned ground vehicle (UGV), the operator of the UGV, an infield rescuer, a mission commander, an unmanned aerial vehicle (UAV), and the operator of the UAV.

For the validation of the design patterns, eighteen of the twenty expert users (UGV and UAV operators, mission commanders, and in-field rescuers) answered a questionnaire on the interaction design patterns 'Explicit Unkowns' and 'Area Policy' of the team awareness system. In the questionnaire, statements were rated with 5-point rating scales (fully disagree to fully agree) on both task and communication levels.

Materials

In the evaluation, two robots were used: an unmanned ground vehicle (UGV) and an unmanned aerial vehicle (UAV), see Figure 6. For more details on the specification, see [18]. The UGV, see Figure 6, is a custom-made robot for situational assessment during the early phases of a disaster response [2]. The operator of the robot has access to a variety of information sources, in a multi-screen multimodal user interface set-up. The views include UGV Operator Control Unit (OCU [13]), and tactical views for team-level situation awareness (the team awareness display [21]). The infield rescuer also had the possibility to add information to the team-level situation awareness tool, by means of a tablet, see Figure 7. In addition, the robot operator was asked to fill in the experienced workload at that moment into a PDA.

RESULTS

Figure 6. The UGV and UAV in the scenario.

Figure 7. Screenshot of team awareness display.

The subjective evaluation aimed to assess the validity of the team awareness design specifications on 'Explicit Unknown' and 'Area Policy'. In the evaluation, 18 subjects rated statements on the task- and communication level design specifications with an ordinal 5-point Likert-scale (1 - fully disagree to 5 - fully agree). A design specification is assumed to be acceptable for ratings above neutral (rating above 3). Whereas all average ratings were between 2.8 and 3.8, here only the most noticeable deviations from average will be highlighted.

Explicit Unknown
Task level The results on 'Explicit Unknowns' indicated that 11 out of 18 subjects rated the task level being acceptable. More specifically, 11 out of 18 subjects rated the task level acceptable on the fact that 'unknowns' can be made explicit by raising an information request. For only 5 out of 18 subjects it was clear enough where attention was drawn when help was needed, enhancing their mental picture of the situation (task level).

Communication level The communication level of the design solution was rated acceptable by 12 out of 18 subjects. More specifically, 11 out of 18 subjects rated the communication level above 3 regarding the fact that an 'explicit unknown' can intuitively be raised with the question-mark button next to the relevant information field. For only 5 out of 18 subjects

it was clear enough that the color of the question-mark symbol indicated who is responsible to answer on the information request.

Area Policy
Task level The results on 'Area Policy' indicated that 10 out of 18 subjects rated the task level being acceptable. More specifically, 11 out of 18 subjects rated the task level acceptable on the fact that a working agreement can be indicated explicitly on the map to share with team members. Only 7 out of 18 subjects rated above 3 on both the understanding how working agreements can be overruled whenever necessary, and on the type of enforcement and to whom it applies.

Communication level The communication level of the design solution was rated acceptable by 12 out of 18 subjects. More specifically, 11 out of 18 subjects appreciated that a pop-up notification indicates that a working agreement area has been reached.

It must be noted that the three UAV operators, who had less time to familiarize with the system for their highly dynamic task, rated both design solutions overall lower than the other users. For both design solutions, only 1 out of 3 subjects rated the task- and communication level being acceptable.

DISCUSSION AND CONCLUSION
This paper presented a study on cognitive engineering and the integration of communication level specifications into a "functional" requirements baseline. In general, this approach provided concise and delimited specifications, which are traceable in the prototype and, consequently, distinctive in the evaluation. Componentsto maintain or improve could be well-identified in a user evaluations. More specifically, design specifications at a task level could be validated, while the evaluation showed sub-optimal results because of a moderate communication level.

So-called Interaction Design Patterns proved to provide a useful language to specify the communication level, complementary to the task level design specification. This can be illustrated with an example from the team awareness design specification validation on Explicit Unknowns. It is observed that on the one hand users appreciate that an 'unknown' can be made explicit by raising an information request to a responsible team member (task-level). On the other hand, the color use of the question-mark symbol, indicating who is responsible to answer an information request (communication), is not clear enough. In this example it is clear that the design specification on functionality level satisfies, while an iterative improvement on the communication level may be required. A suggested design improvement is to extend the visualization of the Explicit Unknown icon on the map with source- and destination information (see Figure 8), which obviously requires validation.

Interaction Design Patterns are complementary to the traditional task level design specification, and therefore facilitate a more specific validation of design specifications by splitting task level and communication level. It facilitates the validation of interaction design solutions, because clear components are constructed for evaluation, based on 'premises', dis-

Figure 8. Suggested Design improvement for the 'explicit unknown' icon. A source- and destination information is added to make it clearer who is responsible for answering an information request.

tinguishing the communication-level effectiveness from the task-level effectiveness (i.e., 'claims'). It benefits from the structured format to capture and share design knowledge. In this respect, the gap between validation and design specification with a differentiation of functional and communication level can be narrowed. Furthermore, in this way, we are developing a library of (validated) interaction design patterns for human-robot collaboration that will be shared, re-used and further developed in other projects (e.g., from the NIFTi-project to the TRADR-project that focuses on persistent human-robot team performance during all disaster response phases).

During the NIFTI project, both the method of situated Cognitive Engineering (sCE) and the sCE tool (www.scetool.nl), developed within TNO, have been improved and extended. It allows designers to generate and test interaction shapes in an structured way, aiming at concise and coherent specifications that can be easily shared, refined and re-used. The Specification phase of the sCE tool has been extended to allow documentation of the Interaction Design Patterns such that incrementally a validated Design Pattern Library can be built for future projects in which interaction aspects play an important role.

REFERENCES

1. Alexander, C., Ishikawa, S., and Silverstein, M. *A Pattern Language: Towns, Buildings, Construction.* Center for Environmental Structure Berkeley, Calif: Center for Environmental Structure series. Oxford University Press, USA, 1977.

2. Balmer, P., Lê, H., and Terrien, G.and Tomatis, N. Platform specification and design. Tech. Rep. Public Deliverable DR 6.1.1, EU FP7 NIFTi / ICT-247870, 2010.

3. Borchers, J. O. A pattern approach to interaction design. *AI & SOCIETY 15*, 4 (2001), 359–376.

4. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A unifying reference framework for multi-target user interfaces. *Interacting with Computers 15*, 3 (jun 2003), 289–308.

5. Dearden, A., and Finlay, J. Pattern languages in hci: A critical review. *Human–computer interaction 21*, 1 (2006), 49–102.

6. Diggelen, J. v., and Neerincx, M. A. Electronic partners that diagnose, guide and mediate space crew's social, cognitive and affective processes. In *Proceedings of Measuring Behaviour* (2010), 73–76.

7. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design patterns: Elements of reusable object-oriented software. *Addinson-Wesley Longman Inc* (1995).

8. Gruber, T. R., et al. A translation approach to portable ontology specifications. *Knowledge acquisition 5*, 2 (1993), 199–220.

9. Gunawan, L. T., Alers, H., Brinkman, W., and Neerincx, M. A. Distributed collaborative situation-map making for disaster response. *Interacting with Computers 23*, 4 (2011), 308–316.

10. Gunawan, L. T., Fitrianie, S., Yang, Z., Brinkman, W., and Neerincx, M. A. Travelthrough: A participatory-based guidance system for traveling through disaster areas. In *Proceeding of Computer Human Interaction, CHI 2012* (Austin, Texas, USA, May 2012), 241–250.

11. Hollnagel, E., and Woods, D. D. Cognitive systems engineering: new wine in new bottles. *Int. J. Man-Mach. Stud. 18*, 6 (June 1983), 583–600.

12. Kahn, P. H., Freier, N. G., Kanda, T., Ishiguro, H., Ruckert, J. H., Severson, R. L., and Kane, S. K. Design patterns for sociality in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, ACM (2008), 97–104.

13. Larochelle, B., Kruijff, G., Smets, N., Mioch, T., and Groenewegen, P. Establishing human situation awareness using a multi-modal operator control unit in an urban search & rescue human-robot team. In *RO-MAN, 2011 IEEE*, IEEE (2011), 229–234.

14. Neerincx, M., and Lindenberg, J. Situated cognitive engineering for complex task environments. In *Naturalistic Decision Making and Macrocognition*, J. M. C. Schraagen, L. Militello, T. Ormerod, , and R. Lipshitz, Eds. Aldershot, UK: Ashgate, 2008.

15. Neerincx, M. A. Situated cognitive engineering for crew support in space. *Personal Ubiquitous Computing 15*, 5 (June 2011), 445–456.

16. Norman, D. A. Cognitive engineering. In *User centered system design; new perspectives on human-computer interaction*, D. A. Norman and S. W. Draper, Eds. L. Erlbaum Associates Inc., 1986, 31–61.

17. Tidwell, J. *Designing interfaces*. O'Reilly, 2010.

18. Tretyakov, V., Winzer, J., and Worst, R. Platform manufacturing and sensor integration (uav). Tech. Rep. Public Deliverable DR 6.1.2b, EU FP7 NIFTi / ICT-247870, 2011.

19. Uschold, M., Gruninger, M., et al. Ontologies: Principles, methods and applications. *Knowledge engineering review 11*, 2 (1996), 93–136.

20. van Diggelen, J., Grootjen, M., Ubink, E. M., van Zomeren, M., and Smets, N. J. Content-based design and implementation of ambient intelligence applications. In *Ambient Intelligence-Software and Applications*. Springer International Publishing, 2013, 1–8.

21. van Diggelen, J., van Drimmelen, K., Heuvelink, A., Kerbusch, P. J., Neerincx, M. A., van Trijp, S., Ubink, E. M., and van der Vecht, B. Mutual empowerment in mobile soldier support. *Journal of Battlefield Technology 15*, 1 (2012), 11.

22. Van Welie, M., Van Der Veer, G. C., and Eliëns, A. Patterns as tools for user interface design. In *Tools for Working with Guidelines*, C. Farenc and J. Vanderdonckt, Eds. Springer, 2001, 313–324.

23. Vicente, K. J. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum Associates, New Jersey, April 1999.

24. Woods, D. D. The theory and practice of representation design in the computer medium (ver 4.2). Tech. rep., Cognitive Systems Engineering Laboratory, The Ohio State University, 1997.