Cogn Tech Work (2017) 19:329–343
DOI 10.1007/s10111-017-0408-4

ORIGINAL ARTICLE

Value sensitive design of a virtual assistant for workload harmonization in teams

Maaike Harbers[1,2] · **Mark A. Neerincx**[1,3]

Received: 18 July 2016 / Accepted: 17 March 2017 / Published online: 25 March 2017

Abstract Uneven workload distributions in teams can lead to suboptimal team performance. This paper therefore describes the design of a virtual assistant that supports workload harmonization in teams by measuring workload, informing team members about their own and other team members' workload, and supporting team members in the redistribution of workload. The virtual assistant was developed in the context of train traffic control according to the situated Cognitive Engineering (sCE) methodology, which was extended to allow for a value sensitive design process. More specifically, the values 'insight,' 'helping others' and 'privacy' were explicitly accounted for throughout the design of the virtual assistant. A prototype of the virtual assistant was evaluated positively in a focus group. Thereby, the contribution of the paper is twofold. First, an improvement in a human-centered development methodology—sCE with values—is described and its use is demonstrated in an actual design case. Second, a novel, positively evaluated solution for workload harmonization in teams is presented.

Keywords Virtual assistant · Teamwork · Traffic control · Value sensitive design · Situated Cognitive Engineering · Human–computer interaction · Workload

✉ Maaike Harbers
M.Harbers@hr.nl

1 Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

2 Rotterdam University of Applied Sciences, Wijnhaven 103, 3011 WN Rotterdam, The Netherlands

3 TNO Human Factors, Kampweg 5, 3769 DE Soesterberg, The Netherlands

1 Introduction

Effective teams are essential for the functioning of many organizations. One of the challenges faced by teams in complex, dynamic work environments is to maintain a proper distribution of (mental) workload over the members in a team (Porter et al. 2003). This is important because high workload levels can lead to overload and extended periods of low levels of workload can lead to underload, which may both result in a decrease in performance (Meshkati and Hancock 2011). To maintain an even workload distribution, team members need to be aware of their own and each other's activities and status, including workload levels (Mesmer-Magnus and DeChurch 2009; Salas et al. 2008). This awareness allows team members to ask for support and provide support to others when needed, in a timely manner. Research has shown that such backup behavior increases team performance (Porter et al. 2003).

Various technological solutions have been proposed to support teamwork. First, technology can be used to *measure* workload of individual team members (Levin et al. 2007), where workload measurement may include physiological (Hankins and Wilson 1998), subjective (Reid and Nygren 1988) or task-related (Neerincx 2003) measures. Second, technology can be used to *inform* workers about the status of other team members, e.g., by displaying information on so-called observability or situational awareness panels (Carroll et al. 2006). Third, technology can provide *assistance* to teams, for instance, by supporting individual team members in the completion of their own tasks or by supporting the team as a whole (Sycara and Lewis 2004). We believe that a combination of these different ways to support teamwork can improve workload harmonization in teams. In this paper, we describe the design and evaluation of a virtual assistant that measures workload, informs team members about their own and others workload and provides assistance in managing workload.

A design approach was needed to develop the virtual assistant. Along with multiple scholars, we believe that it is important to address humans values such as trust, security and privacy during the design process of information systems in general (e.g., Flanagan et al. 2008; Friedman et al. 2013; Van den Hoven 2007), and human–machine systems specifically (Flemisch et al. 2012). For example, a virtual assistant that measures a worker's workload and shares that information with team members—without asking—may violate the worker's privacy. Or workers, whose tasks are—automatically—reallocated to other team members, may feel threatened in their autonomy. Over the past two decades, several methodologies that account for values in design have been developed (Flanagan et al. 2008; Friedman et al. 2013). However, though these approaches offer several methods and tools for eliciting and analyzing values, they have been criticized for failing to provide a systematic method to use elicited values at later stages in the design process (Manders-Huits 2011; De Greef et al. 2013). Therefore, to design the virtual assistant, we used the situated Cognitive Engineering (sCE) methodology (Neerincx and Lindenberg 2008) and integrated values into it. sCE is a design methodology that is specifically geared for the development of human–machine collaboration. It combines a human-centered approach with an emphasis on technology, and it stresses the importance of theory creation based on practical experiences. sCE consists of a foundation, specification and evaluation component.

The sCE methodology assumes a *situated* design and evaluation process, i.e., a specific domain, stakeholder group and context are being studied and involved during design and evaluation. We applied sCE with values to design a virtual assistant for workload harmonization in the context of train traffic control in the Netherlands. Train traffic in the Netherlands is controlled by teams of operators who, most of the time, merely monitor train traffic that is automatically controlled, but in case of a disruption, they take over from the system and redirect train traffic by manually controlling traffic lights and rail switches. Disruptions can hugely increase the workload of one team member in a matter of seconds. Immediately backing up that team member is crucial in such situations, as train delays quickly spread through the railway network. However, in practice, operators do not always instantly notice that their help is needed, and operators in need of help do not always warn others in a timely manner (Siegel and Schraagen 2014). The virtual assistant was designed to provide support in such situations. Much of its properties are not specific to the domain of train traffic control, and thus, the virtual assistant forms a novel solution for workload harmonization in teams in general.

The outline of this paper is as follows. We start with a discussion of how values can be addressed in design, in which we discuss the shortcomings of Value Sensitive Design, provide an overview of sCE and describe how values can be embedded into the sCE design process (Sect. 2). Then, we describe how we applied sCE with values to design a virtual assistant for workload harmonization in train traffic control teams, involving an analysis of the work domain and stakeholder values, a specification of scenarios, objectives, functions and effects, and a prototype of the virtual assistant (Sect. 3). We evaluated the prototype in a focus group session with domain experts (Sect. 4). We end the paper with a discussion and a conclusion (Sect. 5). Thereby, the contribution of this paper is twofold. First, an improvement in a human-centered development methodology—sCE with values—is described and its use is demonstrated in an actual design case. Second, a novel, positively evaluated solution for workload harmonization in teams is presented.

2 Values in design

In this section, we will first provide some background on Value Sensitive Design, a design methodology that explicitly accounts for values in design, but not throughout the complete development process. Subsequently, we will give an overview of the sCE methodology, which supports the full design and development process. Then, we will describe how values and value-based methods can be embedded in sCE.

2.1 Value sensitive design

Over the last 20 years, a considerable body of work has focused on developing theoretical and methodological frameworks to deal with values in designing information systems. Much of this work has emerged in the field of human–computer interaction (e.g., Flanagan et al. 2008; Friedman et al. 2013; Van den Hoven 2007). Value Sensitive Design (VSD) is the most elaborate of these frameworks. Key concepts in most work on VSD are values, stakeholders and value tensions. Values are defined as 'what a person or group of people considers important in life' (Friedman et al. 2013), and examples of values include human welfare, ownership and property, privacy, trust and autonomy. Stakeholders can be direct or indirect, where direct stakeholders are people that directly interact with a system or its output, and indirect stakeholders are people that are impacted by the system without interacting with it directly. Value tensions arise when supporting one value, such as openness, comes at the expense of another, such as privacy (see, e.g., Denning et al. 2010; Czeskis et al. 2010; Miller et al. 2007).

The approaches mentioned above all offer tools and methods to address values in design. VSD, for example, contains the Value Scenarios method (Nathan et al. 2007), which involves writing one or more scenarios about the use of a system that explicate the relation between that system and its stakeholders' values. Another VSD tool is formed by the Envisioning Cards (Friedman and Hendry 2012), which helps designers to envision the influence of a new technology across time, if it becomes pervasive throughout society, affects the lives of different stakeholders and raises issues that touch human values. A last example of a VSD method is the Value Dams and Flows method (Miller et al. 2007), which involves systematically investigating all possible positive and negative effects of a technology on a stakeholder group.

Existing tools and methods for values in design provide various ways to identify and analyze values. However, in most work on values in design, relatively little attention is paid to the operationalization (defining values in concrete terms) and implementation (specifying design features that correspond to identified values) of values. VSD has been criticized for lacking clarity on how to translate results of value analyses into concrete user requirements (Manders-Huits 2011; Poel 2013). One way to overcome these criticisms is to extend design approaches accounting for values (see, e.g., Harbers et al. 2015). An alternative way—which we adopt in this paper—is to use an existing design methodology that covers the complete design process, and extend it to account for values.

2.2 Situated Cognitive Engineering

The situated Cognitive Engineering (sCE) design methodology is highly suitable for the design of the virtual assistant, as it is specifically geared for the development of human–machine collaboration. sCE combines a human-centered approach such as used in classical cognitive engineering with an emphasis on technology, and it stresses the importance of theory creation based on practical experiences. sCE contains methods from user-centered design, cognitive engineering and requirements analyses. It has been successfully applied in various domains, for instance, to develop a task support system in naval ships (Neerincx and Lindenberg 2008), electronic partners for astronauts (Neerincx 2011), social agents for elderly (Spiekman et al. 2011; Peeters et al. 2016), social robots for children with diabetes (Looije et al. 2016), and a support system for human–robot team performance in disaster response (Mioch et al. 2012).

Figure 1 provides an overview of the sCE methodology, showing its three main components: foundation, specification and evaluation. sCE is always applied to solve a problem or fulfill a particular demand in a certain context. The *foundation* includes an analysis of the operational demands, which involves analyses of the work domain and the support that is needed there. Furthermore, the foundation includes a human factors analysis, in which relevant theories, guidelines and support concepts are investigated. Lastly, it includes an analysis of technological principles in which possible technological solutions for the problem at stake are identified. The *specification* component of sCE consists of a number of concepts. First, one or more objectives of the system to be developed are identified. Second, an objective is achieved by one or more functions of the system to be developed. These functions form the requirements of the system. Third, functions are contextualized in scenarios describing the human–technology interaction. Fourth, for each function a number of expected effects are identified. The specification of effects per function (called claims) can later guide the evaluation of the system. The *evaluation* component of sCE involves the building of a prototype or simulation in order to test a design empirically or analytically, respectively. All three components are connected to each other since the activities in each component inform and influence the activities in the other components. There is no specific order for the activities in the different components, and it is possible to go back and forth between different components. The methodology supports a systematic incremental development of social, cognitive and affective support functions, iteratively (a) enhancing the foundation, (b) refining and extending the design specifications and (c) validating the core functions.

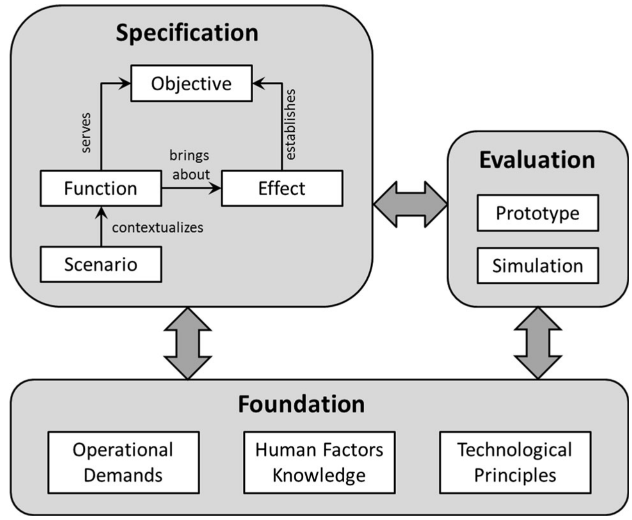

Fig. 1 Overview of the situated Cognitive Engineering methodology

2.3 Embedding values in situated Cognitive Engineering

Values are not explicitly addressed in sCE as described above, but as sCE already contains an arsenal of design methods from different fields, VSD methods and techniques can rather easily be integrated into the methodology. As described below, values are relevant in all three components of the sCE methodology.

To account for values in the *foundation* component, the operational demands analysis should include an analysis of who are the direct and indirect stakeholders, what are their values, and what are possible value tensions. A suitable method, for instance, is to organize a Value Story workshop (Harbers et al. 2015), in which designers and stakeholders systematically analyze how a technology affects stakeholder values. The Envisioning Cards, mentioned earlier, can also play a role in the foundation component, as they offer design activities that focus on values and stakeholders, e.g., ‘Create a list of the system’s direct stakeholders’ (Friedman and Hendry 2012). Besides the VSD methods, literature on specific values at stake can be considered. There is for instance ample literature available on values such as privacy, trust and well-being. Finally, there is an increasing amount of literature that describes the effects of technology on human values. These findings should be considered as part of the study of relevant technological principles.

When values are well-considered in the foundation component of sCE, they will naturally be taken into account in the *specification* component. The results of the activities in the foundation component are reflected in the scenarios, objectives, functions and effects that are specified. More specifically, the Value Scenario method proposed as part of the VSD methodology (Nathan et al. 2007) is an example of a scenario specification that accounts for values. Furthermore, it can be an objective to support a certain value, e.g., security or creativity. This relates to the concept of an ‘explicitly supported value,’ often used in the VSD literature (Friedman et al. 2013). We believe that by paying attention to values in the foundation phase, the number of explicitly supported values will increase. Lastly, to achieve the design objectives in the context at stake, the functions and expected effects that are specified should serve and support values, respectively. A simple example of a function that serves the value of privacy is that users are given the option to control their own privacy settings, and an expected effect of this function could be that privacy is preserved.

Values also have a place in the *evaluation* component of sCE. A specification that takes values into account should lead to one or more prototypes in which values are regarded and to evaluation studies that test the effects of the prototypes on values. Evaluating multiple prototypes and comparing the different results resembles the Value Dams and Flows method in the VSD literature (Miller et al. 2007). Results of evaluation studies can inform activities in the specification and foundation components. For instance, evaluation studies may lead to the identification of new functions, or to the formation of new theories.

3 Design of the virtual assistant

In this section, we describe how we applied sCE with values, as outlined above, to design a virtual assistant for workload harmonization in train traffic control teams. We start with the foundation and specification of the design, followed by the prototype of the virtual assistant.

3.1 Foundation

We conducted this research in the context of railway traffic control in the Netherlands—which is operated by the organization called ProRail.[1] All train traffic in the Netherlands is controlled from thirteen regional and one national control center. Each regional control center is manned by (among others) a team of operators and a team leader controlling train traffic. Under normal circumstances, i.e., most of the time, train traffic is regulated by an automatic train scheduler and the operators merely monitor the situation. This yields relatively low levels of workload for the operators. In case of a disruption, however, they have to respond quickly to avoid (more) delay, which can increase their workload considerably in a matter of seconds. Tasks involve, for instance, manually controlling the signals and switches on the tracks, and communicating with train drivers.

Each operator is responsible for a particular section of the railway network and in principle needs to perform all tasks associated with that section. Thus, in case of a disruption, the operator assigned to that railway section is primarily in charge. This division of work yields situations in which some operators experience a very high workload level, while others have low levels of workload. The protocol dictates operators to warn others in such cases, but in practice, operators do not always ask for help in a timely manner. An explanation for this behavior is that operators underestimate the size of the disruption at hand or that they overestimate their own capacities to deal with it. This can result in highly uneven workload distributions, which is undesirable as high workload levels can yield overload, and low workload levels can lead to boredom and decreased motivation. An uneven distribution of workload can make

[1] http://www.prorail.nl/.

the team as a whole less resilient (Siegel and Schraagen 2014).

3.1.1 Human factors knowledge

For many decades, mental workload and its assessment have received considerable interest from researchers. Despite this long-lasting interest in the topic, there is no clearly defined, universally accepted definition of workload (Cain 2007). An intuitive, general definition of workload is 'the amount of mental work necessary for a person to complete a task over a given period of time' (Longo 2015). Research on workload is often performed in the context of (the design of) human–machine interfaces. It has been widely recognized that both mental overload and underload can negatively affect human performance (e.g., Gawron 2008). Thus, in order to maximize performance of human–machine systems, workload optimization of human operators has to be taken into account. To that end, many instruments for measuring workload have been proposed. These instruments are often divided into the following three categories: physiological measures, performance-based measures and subjective measures (Eggemeier and Wilson 1991; Galy et al. 2012). Sometimes, a combination of multiple measures is used to measure workload or predict performance (e.g., Cohen et al. 2015; De Regt et al. 2016). Our prototype will also combine multiple workload measures.

The first set of instruments, *physiological measures*, is a diverse category. Heart rate, heart rate variability, pupil size, eye blink frequency and duration, electrodermal measures, respiration frequency and EEG are among the variables that have been used as an indirect of measure workload (Hogervorst et al. 2014). Our prototype will contain a physiological state measure that is based on a simplified version of Mehrabian's Pleasure-Arousal-Dominance (PAD) model (Mehrabian 1996). The PAD model quantifies someone's emotional state according to three dimensions: pleasure, arousal and dominance. The measure used in the prototype leaves out the dimension of dominance, which still produces a useful classification of emotional states (Neerincx 2007). We do not claim that this physical state measure precisely indicates someone's workload, but there is a lot of evidence suggesting that the two are correlated. We therefore believe that the physical state measure can be helpful in obtaining more insight in someone's workload.

Performance-based measures of workload collect real-time behavioral data on an operator's task performance, e.g., speed, error rate or number of completed tasks. Our prototype will include an measure of cognitive load that falls into this category. The measure is based on a model of cognitive task load that can be used to assess an operator's cognitive task load over a certain period of time (Neerincx 2003). The model contains three components: time occupied, level of information processing and task-set switches. Time occupied captures the amount of time that the operator is working during a given time period. Level of information processing captures the complexity of the tasks that the operator is working on in the time period. Task-set switches capture the number of times the operator switched from one task to another in the time period. Cognitive task load is calculated by combining these three measures, where its value increases with more time occupied, higher levels of information processing and more task switches.

Subjective measures are based on self-reported workload levels by the human operator, either during break points or after interaction with the system. Two often used subjective measures for workload are the NASA Task Load Index and the SWAT subjective workload assessment technique (Galy et al. 2012). An advantage of a subjective measure is that it is less intrusive than the other two types of measures. However, it places an extra burden on the operators who have to report their experienced workload, particularly if real-time data are to be obtained. Our prototype does not include a subjective workload measure as, as will be discusses later, our main goal is to inform operators about their workload. The prototype would not be useful if it would inform operators about workload levels they had indicated themselves.

3.1.2 Technological principles

There are several considerations to be made for the design of the virtual assistant. First of all, the advantage of using a virtual assistant (rather than a human) is that it can continuously monitor workload of all team members and immediately respond if a disturbance in the team's workload distribution arises, at all times. In earlier work, based on observations of and conversations with train traffic operators, we already created a first prototype of the virtual assistant. This prototype is a monitor that assesses and displays the workload of train traffic operators (Neerincx et al. 2014). The monitor assesses two types of workload. The first type of workload, called *cognitive load*, is based on a cognitive task load model proposed by Neerincx (2003). Cognitive load is determined through a performance-based measure that assesses an operator's interaction with the traffic control system. The second type of workload, *physiological state*, is based on the Pleasure-Arousal-Dominance model (Mehrabian 1996) and assessed through physiological measures such as heart rate and galvanic skin response. The monitor displays these two measures in real time on a graphical user interface. Operators can thus continuously observe their own cognitive load and physiological state.

The measures used in the monitor are suitable for our purposes. To continue this work in line with previous work as much as possible, we therefore decided to use the same cognitive load and physiological state measures for the design of the virtual assistant (for more details on the monitor, see Neerincx et al. 2014).

3.1.3 Operational demands

In a workshop, we demonstrated the prototype of the monitor to a group of domain experts working at ProRail and jointly made an analysis of the monitor's stakeholders and their values. Details about the workshop are provided in Harbers and Neerincx (2014). The result of the workshop was a list of stakeholders of the monitor (e.g., operator, team leader, passengers), their values (e.g., insight, recognition, trust) and effects of the monitor on these values (e.g., the monitor supports a team leader's insight into how the team is functioning). Of all values that were identified we will discuss the values insight, helping others and privacy, as they match our initial ideas about the virtual assistant (Neerincx et al. 2014) and/or were considered very important by the workshop participants (Harbers and Neerincx 2014).

The first value that received a lot of attention in the workshop is *insight*. Domain experts in the workshop deemed insight into one's own workload valuable, because operators may not always be aware of their own workload. They expected that operators that become more aware of their own workload learn to make better estimations of when they are able to cope with a situation themselves, and when they need the assistance of others. This makes train traffic operators more likely to notify others in a timely manner when necessary. The workshop participants also appreciated insight into the two different workload measures: workload based on physiological state and workload based on interaction with the system.

A value that forms one of the core values of the virtual assistant is *helping others*. The workshop participants saw benefits in sharing workload information in order to support operators to help each other. If operators are informed about others' workload, they know when a team member needs help. For instance, if the virtual assistant detects a high and low workload level for operator A and B, respectively, sharing this information allows operator B to take over tasks from operator A in a timely way. The domain experts also mentioned that sharing workload information helps operators to avoid disturbing other operators with less urgent questions at moments when they are very busy. Finally, they expected that sharing information would allow team leaders to better organize the team, e.g., assign the most challenging railway section to an operator who can deal with that best.

A third value that received a lot of attention in the stakeholder value analysis was *privacy*. Though the domain experts acknowledged the potential value of collecting and sharing workload information in traffic control teams, they were concerned that it could lead to *privacy* loss, particularly when operators and team leaders could continuously see each other's workload levels. They made remarks like 'The differences between good and bad operators will become very explicit' and 'Operators may become vulnerable and insecure.' Also, they pointed out that operators may be stressed due to personal factors, rather than because of the work situation, and that this is something that most people will want to keep private.

The activities described above are in line with the sCE with values design approach. First, the design activities to create the prototype were *situated* in the sense that they took place in a real-world context. Second, *values* were explicitly addressed by performing a stakeholder and value analysis with domain experts. sCE advocates an iterative design process, and these activities can be seen as a first iteration in the design of the virtual assistant. This paper reports the next design iteration, in which insights obtained from the first iteration are used to improve and further elaborate the design of the virtual agent.

3.2 Specification

Based on insights obtained from observations of the traffic control operation, conversations with train traffic operators, team leaders and other domain experts, the development of the workload monitor, the value and stakeholder analysis and existing literature, we created a specification of the virtual assistant. The specification follows the sCE (with values) approach (see Fig. 1) and specifies the virtual assistant's objectives, functions, effects and scenarios.

As discussed in the previous section, the values insight, helping others and privacy were considered very important by stakeholders and/or the problem description we started off with. Accordingly, we made these three values the main objectives of the design. More specifically, the three objectives of the virtual assistant are to: (1) provide operators with insight into their own workload, (2) support the operators to help each other when needed and 3) preserve operators' privacy. Below we describe the functions we identified per objective, and the effects associated with them.

- *Objective 1* Provide operators with *insight* into their own workload.

As discussed in Sect. 3.1.1, workload can be measured through different measures, e.g., physiological (Hankins and Wilson 1998), subjective (Reid and Nygren 1988) or task related (Neerincx 2003). The workshop participants said that they appreciated insight into different workload measures, particularly if rare situations are marked. Our first prototype, the monitor, contained workload measures for cognitive load and physiological state (Neerincx et al. 2014). In the literature, however, it has been argued that affective responses to tasks, e.g., level of urgency, also have an impact on someone's workload (Neerincx 2007; Colin et al. 2014). Therefore, in our second prototype, we add a measure for affective load. This yields the following first function to achieve objective 1.

Often used technologies for providing information to operators about real-time measures or situations are observability panels and situational awareness displays (Carroll et al. 2006; De Greef 2012). Though the workshop participants were positive about the provision of different workload measures, they also mentioned that providing multiple measures may be confusing or more difficult to process for the operator. We therefore add a function that combines the different measures into one indicator for total workload and give operators the option to choose which measure(s) they want to observe.

- *Function 1A* The virtual assistant shall measure and display real-time information about the operator's cognitive load, affective load and physiological state, or, if preferred, the operator's total workload.
    - *Effect 1Aa* Operators are more aware of their own workload.
    - *Effect 1Ab* Operators are more aware of different components (cognitive load, affective load and physiological state) comprising their workload.

Normally, one would expect that an operator's cognitive load, affective load and physiological state will show strong correlations with each other (Van Broekhoven et al. 2016). For instance, an increase in someone's cognitive workload will most likely coincide with an increased heart rate. If the different measures show a discrepancy, this may be an indication that something is not going well. For example, if an operator experiences a lot of stress without having much work to do. Rare combinations of workload measures thus may be an indication of a decrease in resilience (Siegel and Schraagen 2014).

- *Function 1B* The virtual assistant shall detect rare combinations of different workload measures and inform the operator about them.
    - *Effect 1B* Operators are more likely to respond to workload-related problems in a timely manner, e.g., by warning others.

- *Objective 2* Support operators to *help each other* when needed.

Multiple studies have shown that sharing information between team members improves team performance (Mesmer-Magnus and DeChurch 2009). Sharing workload information allows team members to take over work and help each other when needed, yielding a more even distribution of workload over the team members, and thus a more resilient team (Siegel and Schraagen 2014). Based on these insights, we identified the following function and effects.

- *Function 2A* The virtual assistant shall share the operator's current workload with (virtual assistants of) other team members and display other team members' current workload—if permitted.[2]
    - *Effect 2Aa* The operator will provide help to and receive help from others more timely.
    - *Effect 2Ab* The operator will less often disturb others and be disturbed by others when that is inconvenient.
    - *Effect 2Ac* Team performance will improve (i.e., less train delays).

Besides sharing workload information between team members, the virtual assistant can provide additional support by analyzing workload information and providing operators advice on when to take over tasks from whom (De Greef 2012). For instance, if a virtual assistant observes that operator A has a high workload, it can advise (the virtual assistant of) another operator to help operator A.

- *Function 2B* The virtual assistant shall provide advice to the operator about when and whom to help.
    - *Effect 2Ba* The operator will provide help to and receive help from others more timely.
    - *Effect 2Bb* Team performance will improve (i.e., less train delays).

- *Objective 3* Preserve operators' *privacy*.

On the one hand, information sharing can promote team performance, but on the other hand, it can violate privacy (Harbers et al. 2014). Literature on privacy shows that the amount of privacy loss due to information sharing depends on a number of factors such as the receiver of the information, the sensitivity of the information and the context in which the information is shared (see e.g., Lederer et al. 2003; Nissenbaum 2009). Lederer et al. (2003) performed a study in which they found that people are more likely to share more sensitive information with people they know than with strangers and that they are more likely to share sensitive information when they are proximately located

[2] The clause "if permitted" was added to preserve privacy (see objective 3).

than when they are remotely located. Nissenbaum (2009) also stresses the importance of the context in which information is used for whether privacy is violated or not. To preserve operators' privacy, we believe that these factors should be taken into account and that the person whose information is being shared (or not) should be in charge. This is specified in the following function.

- *Function 3* The virtual assistant shall allow the operator to control its information sharing policy, taking information type, receiver and context into account.
  - *Effect 3a* Operators will share workload information in urgent situations.
  - *Effect 3b* Operators' privacy is preserved.

To contextualize the objectives, functions and effects specified above, we created two scenarios. In this research project, we formed a picture of train traffic operators through interviews and observations. Operators A and B featuring in the two scenarios are based on our findings and mean to be representative of different types of operators in train traffic control teams. The scenarios aim to illustrate the functioning and envision the value of the virtual assistant in a train traffic control context for different operators.

Scenario 1: *Operator A recently finished his training and has worked in a train traffic control team for a few weeks now. He generally performs well, but still finds it hard to quickly estimate the size of a disruption and the amount of work it takes to solve it. Operator A therefore adapted his information sharing settings on the virtual assistant such that others are always informed about his cognitive load. One day, there is a small disruption that yields relatively little work for operator A. However, he is unsecure about the actual size of the disruption. A's virtual assistant then says: 'According to my measures you seem to be nervous, while your cognitive load is relatively low. Is everything ok?' This way A realizes he is overreacting to the situation and is able to calm himself down.*

Scenario 2: *Operator B has been working as a train traffic controller for years and does not like her data to be shared with everyone all the time. She therefore sets the information sharing settings on her virtual assistant to not sharing information under normal circumstances. One time during a disruption, she is so focused on solving the problem that she forgets to request her team members for help. Her virtual assistant detects a high workload and asks: 'There is a severe disruption going on and you have a high workload, shall I check if someone is available to help you?' Operator B clicks on 'agree.' the virtual assistant requests operator C, who is available, to help operator B. Together they solve the incident.*

3.3 Prototype

We created a prototype of the virtual assistant that accommodates the objectives and functions specified in the previous section. The design involves a graphical user interface for interaction with the operator and an internal agent model that comprises the 'intelligence' of the virtual assistant. The prototype is integrated with a high-fidelity simulator of train traffic control in the Netherlands (Meijer 2012), enabling the collection of realistic workload information in real-time and immediately providing that information to the person playing the simulation game.

3.3.1 User interface

In the previous section, we identified three functions that concern the operator receiving information from the virtual assistant related to his/her own work and workload: display workload information (function 1A), inform operator about rare combinations (function 1B) and provide advice about when and whom to help (function 2B). We decided to combine all the functions in which the virtual assistant provides information in one display, the 'monitor,' which is shown in Fig. 2. This figure shows a graph in the upper left corner and middle of the window that displays real-time information about the operator's cognitive, affective and total load. The operator can choose which of these three measures he/she wants to be displayed. In the upper right corner, real-time information about the operator's physiological state is displayed. In this prototype, physiological state information only consists of the operator's heart rate, but other physiological measures can be added. The workload and the physiological state graph together implement function 1A. The bottom area offers space for text messages of the virtual assistant, and in the right lower corner there are two buttons 'yes' and 'no' that allow the operator to respond to questions of the virtual assistant. This enables the realization of function 1B and 2B.

Function 2A specifies that the virtual assistant shall share the operator's current workload with (virtual assistants of) other team members and display other team members' current workload—if permitted. This function is implemented by another window called 'team status.' In this tab, shown in Fig. 3, the operator can see the cognitive load, affective load, total load and heart rate of the other team members, provided that they allowed this information to be shared (sharing preferences are set in another window). In the figure, there is only one row, where information of one team member can be displayed, but more rows are added if the virtual assistant receives more

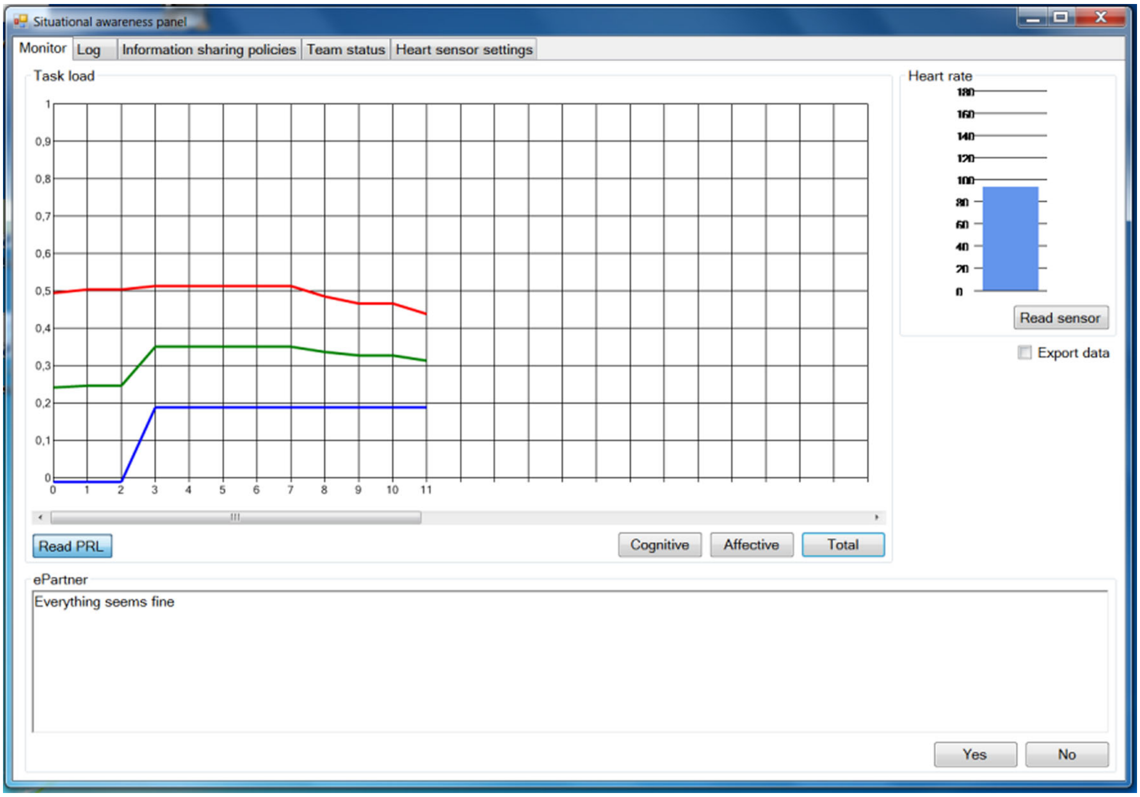

Fig. 2 Tab that displays real-time workload measures and messages from the virtual assistant and that enables the operator to respond to the virtual assistant

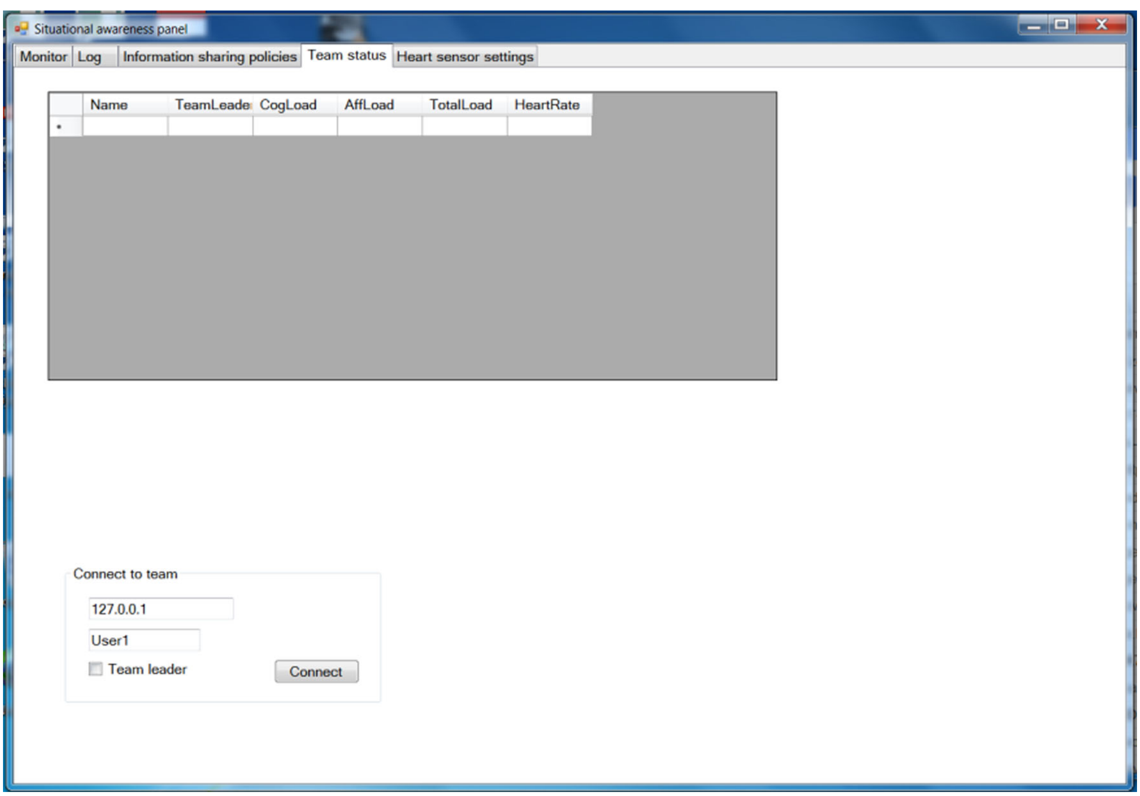

Fig. 3 Tab that displays workload information of other team members

information. This tab thus implements the front end of function 2A.

The final function, function 3, dictates that the virtual assistant shall allow the operator to control its information sharing policy, taking information type, receiver and context into account. This is implemented in the 'information sharing policies' tab, displayed in Fig. 4. This tab enables operators to adjust their information sharing preferences. Operators can choose to have the virtual assistant always share information, ask permission for sharing information or never share information. The option 'ask permission' allows operators to take the *context* into account. Operators can set these preferences specific to *information type* (cognitive load, affective load and emotional state) and *receiver* of the information (team member and team leader).

The graphical user interface has two other tabs besides the 'monitor,' 'team status' and 'information sharing policies' tabs. The 'log' tab displays a log of the operator's workload, the virtual assistant's messages and the operator's interactions with the virtual assistant over time. The 'heart sensor settings' tab allows for a calibration of the heart rate sensor. These two tabs were added for research purposes. In a final product, they would disappear from the graphical user interface.

3.3.2 Internal agent model

The graphical user interface can only function properly with an intelligent agent behind the scenes. There are a number of architectures that can be used to create cognitive models of intelligent agents. Often used architectures in literature on cognitive modeling are for instance the ACT-R (Anderson 2014), Soar (Laird 2012) and Belief Desire Intention (BDI) (Bordini et al. 2009) architecture. Each of these architectures has a different focus. ACT-R is strongly inspired by cognitive neuroscience and tries to mimic thinking in the human brain (Anderson 2014). Soar aims to handle all capabilities of an intelligent agent, from very simple to highly complex tasks, and mostly uses symbolic representations for that (Laird 2012). The BDI architecture uses a representation that is based on the way humans think that they think, i.e., in terms of beliefs, desires and intentions (Bratman 1987).

We decided to use a BDI-based approach for the internal model of the virtual assistant because of its intuitive way to represent human practical reasoning. Someone could reason, for instance, 'I warn my team mates because I *believe* that the incident is severe' (belief) or 'I warn my team mates because I *want* to avoid as much delay as possible' (desire). Beliefs, desires and intentions do not necessarily describe how human thinking actually works, but they do

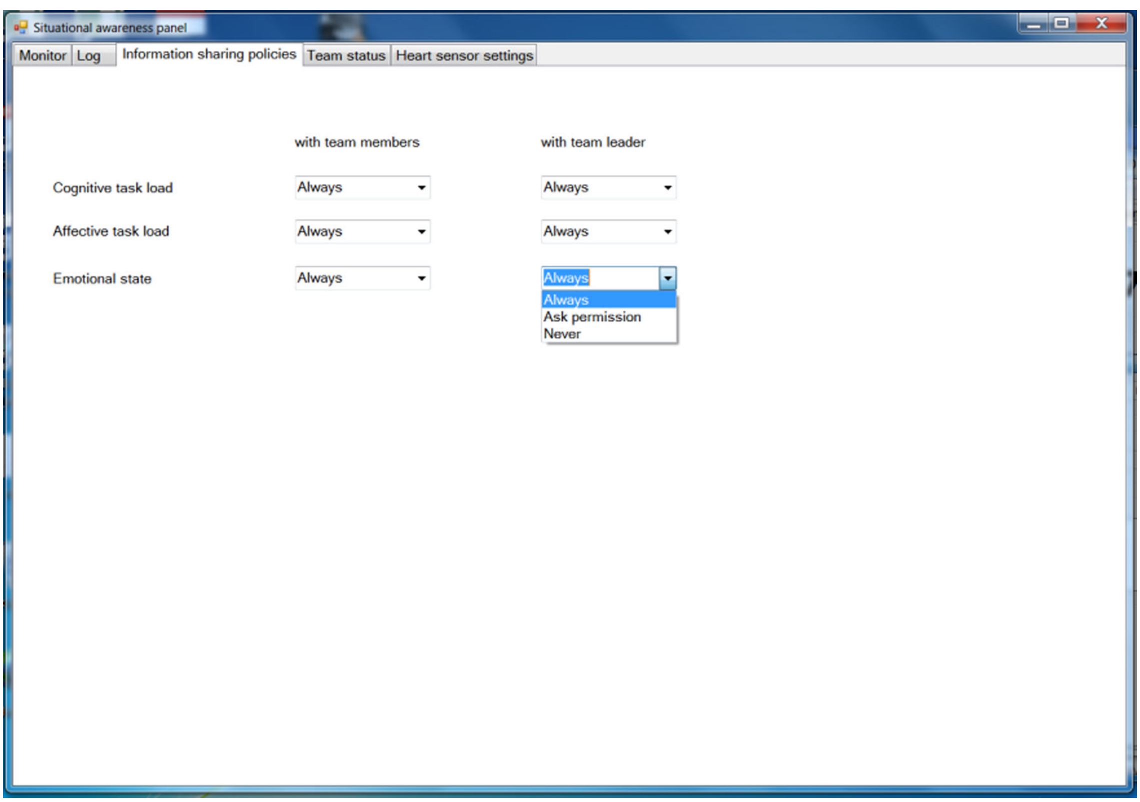

Fig. 4 Tab that enables the operator to set context-specific information sharing policies

capture the way people describe their own though process and how they understand and explain the behavior of others. This idea forms the basis of a family of BDI-based agent programming languages with beliefs, goals (desires) and plans (intentions) as their basic building blocks (Bordini et al. 2009). BDI-based programming languages thus provide an intuitive way to program agent behavior, closely related to human practical reasoning. Moreover, it is easy to understand and explain how an agent got to a certain action when inspecting the code behind it (Harbers et al. 2010).

There is plenty of literature explaining the details different BDI-based agent programming languages (Bordini et al. 2009), but the basic functioning of a BDI agent is as follows. A BDI agent has a set of beliefs (belief base) and a set of goals (goal base) and a set of reasoning rules (rule base) describing which action to pick for which combination of beliefs and goals. When active, the BDI agent performs a so-called sense-reason-act cycle. In one cycle, it *senses* its environment based on which it updates its beliefs, it then *reasons* over its current beliefs and goals to select an action (i.e., it applies its reasoning rules), and finally it *performs* the selected action in its environment. After performing an action, the BDI agent again starts sensing its environment, which may have changed. Once one of the agent's goals is achieved, it is automatically removed from its goal base.

We used the BDI-based agent programming language GOAL (Hindriks 2009) to implement the reasoning capacity of the virtual assistant. We represented the three main objectives (provide insight, support helping others and preserve privacy) of the virtual assistant as *goals* of the BDI agent. We represented information about measures (cognitive load, affective load, physiological state and total load) of the operator and, if available, of other team members as *beliefs* of the agent. Furthermore, the preferred privacy settings of the operator and messages received from other virtual assistants are also represented as beliefs. These beliefs are updated in each sense-reason-act cycle of the agent. Finally, we specified a set of reasoning rules to select actions based on the agent's beliefs and goals, where the actions either involve displaying a message to the user or sending a message to another virtual assistant. We will discuss a selection of the virtual assistant's reasoning rules.

One of the reasoning rules that aims to increase insight is the following:

```
if bel(cl(self)<0.1, al(self)>0.7)
then self.send("Your cognitive load is low, but your affective load is
  high. Is everything going well?").
```

This rule expresses that if the virtual assistant has the belief (bel) that the operator's cognitive load (cl(self)) is lower than 0.1 (on a scale from 0–1) and the operator's affective load (al(self)) is larger than 0.7, it sends a message to the operator (self.send('…')). That means that the text is displayed in the lower part of the monitor tab of the user interface (see Fig. 2). This rule detects an odd combination of workload measures. Normally, a low cognitive load coincides with a high affective load; when that is not the case, it may be an indication that something is going wrong.

The following reasoning rule supports operators in helping each other:

```
if bel(cl(self)=X, sharingPolicy(cl,teamMembers)=always)
thenteamMembers.send(cl(self)=X).
```

This rule is applicable if the virtual assistant has the belief that the operator's cognitive load (bel(cl(self))) is X, or in other words, that the virtual assistant knows what the operator's cognitive load is, and if the operator's information sharing policy (set by the operator through the interface in Fig. 4) allows the virtual assistant to always share cognitive load with team members (sharingPolicy(cl,teamMembers) = always). If this is the case, then the virtual assistant will send a message to all teamMembers (teamMembers.send(…)) with information about the cognitive load of the operator (cl(self) = X)). This information will then appear in the team status tab of other team members (see Fig. 3). This rule thus makes sure that, if allowed by the operator, the operator's cognitive load is shared with all team members in every time step.

The following reasoning rule also supports operators in helping each other, while preserving the operator's privacy:

```
if bel(cl(self)>0.8, cl(op1)<0.3, sharingPolicy(cl,teamMembers)=ask)
then self.send("Your cognitive load is high, and operator 1's
cognitive load is low. Shall I request operator 1 for help?").
```

This rule fires if the virtual assistant has the beliefs that the operator's cognitive load is more than 0.8 (cl(self) $>$ 0.8), the cognitive load of operator 1 is less than 0.3 (cl(op1) $<$ 0.3), and the operator indicated (see Fig. 4) that the virtual assistant should ask for permission to share his/her cognitive load with team members (sharingPolicy(cl,teamMembers) = ask). If so, the message 'Your cognitive load is high, and operator 1's cognitive load is low. Shall I request operator 1 for help?' is displayed in the monitor tab (see Fig. 2). The operator can then choose to accept or decline via the 'yes' and 'no' buttons in the lower right corner. This rule identifies situations in which there are large workload disparities between team members and supports operators in fixing that.

The reasoning rules shown above serve as examples. We currently only implemented a small set of reasoning rules to demonstrate the functionality of the virtual assistant. The thresholds in the reasoning rules were set after some experimentation with different values and seemed appropriate at face value. However, this set can (and will) be refined and extended, based on empirical experiences, to improve the effectiveness of the virtual assistant.

4 Evaluation in focus group

One of the bigger challenges in this project was to get in touch with professionals. Though the research was supported by the organization, middle layer managers tended to be rather protective of their workers. They did not want their workers, like train traffic operators, to be disturbed by experiments, interviews, questionnaires and the like, so that the operators could focus on their actual work. We spent considerable time in convincing managers of the importance of involving direct stakeholders in the design process, throughout the whole trajectory, with moderate success. Ideally we wanted to evaluate this design iteration of the virtual assistant by letting operators interact with it, but at this point, we were only allowed to organize a three member focus group evaluation. In this section, we will describe the results of this evaluation.

We evaluated the specifications and prototype in a focus group consisting of a team leader and two operators from the train traffic control center of Amsterdam. For each of the three objectives (insight, providing help, privacy), we showed them our specifications on paper, and the corresponding tabs of the interface on a laptop screen. We explained what we meant by the specified objectives, functions and effects and clarified how we translated the specifications to an implementation of the prototype. After presenting and explaining the specification and prototype related to one objective, we asked the members of the focus group for observations, remarks and comments. We will discuss their feedback related to the three objectives: providing insight, supporting helping each other and protecting privacy.

The members of the focus group were positive about the objective to provide operators *insight* into their own workload. They confirmed our assumption that there are situations in which operators fail to ask for assistance in a timely manner. The focus group members thought that such inadequate responses can sometimes be explained by shame, but that they mostly occur due to an overestimation of one's own capacities. The members thought that the virtual assistant, by providing operators insight in their own functioning, could help operators to better estimate their own capacities.

The virtual assistant provides insight via the monitor tab, which displays information about an operator's cognitive, affective and total load, physiological state, and rare combinations of these measures (see Fig. 2). The focus group members raised two points of critique when shown the interface. First, they commented that the exact meaning of cognitive and affective load may be hard to grasp. This is a valid point and should be further researched with train traffic operators. If the meaning of both concepts indeed is unclear, even after explanation, it may be better to only display total load. The virtual assistant can still use the two measures in its reasoning process to determine what to do. Second, one of the group members remarked that operators are already working with multiple screens and that they are reluctant to add yet another screen and source of information. Another group member added that operators are particularly unlikely to look at the screen when they have a high workload. An interesting suggestion was to use information about workload for a reflection session after the shift.

The second objective, supporting *helping each other*, was also positively received by the focus group. They repeated that operators sometimes fail to ask for assistance on time, in most cases because they overestimate their own capacities in handling a disruption. The members of the focus group explained that operators in function usually notice more or less what the other operators in their team are working on because they are located in the same space, but that these operators can only provide limited assistance, as they cannot leave their own station. However, at least at the train traffic control center in Amsterdam, there is always an extra (senior) operator available during the early and late day shifts, who is available for providing help. This person is often not present in the same space as the other operators, so the team would highly benefit from a virtual assistant that would notify this extra person.

The group members discussed that 'providing assistance' is not always easy, as there is no protocol for how to do that. It occurs that when a team member starts helping out an operator, the operator in function loses overview of the situation, and no longer knows what strategy to pursue and which actions need to be taken. At the moment, neither our specification nor our prototype addresses this problem, but the virtual assistant has potential to provide support.

The third objective, *preserving privacy*, was considered important by the group members. Though the team leader in the focus group was particularly enthusiastic about the idea that he could monitor the workload of all of his team members (see Fig. 3), he understood that they would have their reservations about that.

The group members responded to the information sharing policies tab of the virtual assistant (Fig. 4) in a mixed way. One of the operators in the focus group was very positive about the possibility for each operator to set his/her own preferences. The other two focus group members, however, preferred to have a team policy for information sharing, rather than letting the operators determine their information sharing policies individually. They thought that strong differences between operators in their information sharing settings could lead to awkward situations. It would be possible to design the virtual assistant such that both options are possible, that is, preferences can be set individually or for the team.

To conclude, the focus group members were positive about all three objectives, but they had some comments on how these objectives were implemented in the prototype of the virtual assistant. Their most important comments and suggestions for improvement can be summarized as follows. First, information provided in real-time should be as simple as possible because operators already process a lot of information during task performance. Second, besides real-time information provision, workload information should be made available such that it can be used during a reflection after a shift. Third, besides notifying operators to assist team members, the virtual assistant should support the process of providing assistance. Fourth, more attention should be paid to the way a team of operators sets its information sharing policies.

5 Discussion and conclusion

In this paper, we proposed to extend the situated Cognitive Engineering (sCE) methodology such that values are explicitly accounted for throughout the design process. Subsequently, sCE with values was applied to design a virtual assistant for workload harmonization in the domain of train traffic control. The design process consisted of a foundation, specification and evaluation component (see Fig. 1). As part of the foundation component, we organized a workshop with stakeholders to systematically identify the values at stake. Of all values that were identified, we selected 'insight,' 'helping others' and 'privacy' as the most relevant values for the design of the virtual assistant. In the specification component of the design process, we specified these three values as the main objectives of the design. The prototype that resulted from the design process was evaluated with (different) stakeholders, yielding positive feedback and a number of suggestions for improvement.

Regarding the *design process*, in our experience, applying sCE with values made it possible to incorporate values and account for them in the design process in a natural way. After introducing values in the foundation component, they were automatically embedded in the other components of the sCE design approach. In both sessions with stakeholders, the workshop (see Sect. 3.1) and the focus group (see Sect. 4), stakeholders were enthusiastic about the goal to account values in the design. Furthermore, the members of the focus group were in favor of the values we selected as objectives of the design, and they were mostly positive about the way we had accounted for them.

sCE does not prescribe a specific order of design activities. That means that the application of sCE with values to other design problems may differ from the process described in this paper. For instance, one may start the design process with a specification or evaluation, rather than with the foundation. Based on the description of how values are integrated in sCE in Sect. 2.3 and our experiences in this project, we believe that values can be inserted in the design at any point in the process, regardless of the order in which different design activities are performed, and that once inserted, they will naturally have a place in other parts of the design process.

The application of sCE with values in this project leads to a value-informed specification and prototype was positively evaluated by domain experts in this project. Though there are some limitations to this study, these results are positive and sCE with values seems a promising approach to account for values in design throughout the complete design process. More insight and experience into how to embed values in sCE need to be obtained by applying the sCE with values approach in other design projects.

As for the *design outcome*, this paper presented an agent framework for knowledge-based adaptive task support that is transparent and more flexible than procedural support systems (e.g., Neerincx and De Greef 1998). More specifically for this application, the feedback of the focus group will allow us to further improve the design of the virtual assistant. In our next iteration, we plan to address the suggestions of the focus group members to make the presentation of information simpler and to improve the control of information sharing policies. These two suggestions involve improvements of existing features of the virtual assistant and are therefore relatively easy to address in a next prototype. Inspired by Siegel and Schraagen (2017), who show that after-action-reflection is a promising direction to support train traffic control teams, we will also investigate this option. In a next iteration, we plan to test the next prototype of the virtual assistant in the high-fidelity simulator by letting operators actually interact with it. We thus hope to create a solution for workload harmonization in train traffic control settings that will also be of value for workload harmonization in similar domains.

Acknowledgements The authors thank the members of the focus group for their valuable input. This research is part of the RAILROAD project and is supported by ProRail and the Netherlands organization for scientific research (NWO) (Under Grant 438-12-306).

References

Anderson JR (2014) Rules of the mind. Psychology Press, Routledge

Bordini RH, Dastani M, El Seghrouchni AEF (2009) Multi-agent programming: languages, tools and applications. Springer, Berlin

Bratman M (1987). Intention, plans, and practical reason. Harvard University Press

Cain B (2007) A review of the mental workload literature. Defence Research And Development Toronto, Toronto

Carroll JM, Rosson MB, Convertino G, Ganoe CH (2006) Awareness and teamwork in computer-supported collaborations. Interact Comput 18(1):21–46

Cohen I, Brinkman WP, Neerincx MA (2015) Modelling environmental and cognitive factors to predict performance in a stressful training scenario on a naval ship simulator. Cogn Technol Work 17(4):503–519

Colin TR, Smets NJ, Mioch T, Neerincx MA (2014). Real time modeling of the cognitive load of an Urban Search And Rescue robot operator. In robot and human interactive communication, 2014 RO-MAN: The 23rd IEEE international symposium on (pp 874–879)

Czeskis A, Dermendjieva I, Yapit H, Borning A, Friedman B, Gill B, Kohno T (2010). Parenting from the pocket: value tensions and technical directions for secure and private parent-teen mobile safety. Proceedings of the sixth symposium on usable privacy and security, ACM, p 15

De Greef T (2012) Virtual assistants for dynamic task allocation and coordination. Delft University of Technology, TU Delft

De Greef T, Mohabir A, Van der Poel I, Neerincx M (2013) sCEthics: embedding ethical values in cognitive engineering. In Proceedings of the 31st European conference on cognitive ergonomics, ACM, p 4

De Regt A, Siegel AW, Schraagen JM (2016) Toward quantifying metrics for rail-system resilience: identification and analysis of performance weak resilience signals. Cogn Technol Work 18(2):319–331

Denning T, Borning A, Friedman B, Gill BT, Kohno T, Maisel WH (2010) Patients, pacemakers, and implantable defibrillators: human values and security for wireless implantable medical devices. Proceedings of the SIGCHI conference on human factors in computing systems, ACM, p 917–926

Eggemeier FT, Wilson GF (1991) Performance-based and subjective assessment of workload in multi-task environments. In: Damos D (ed) Multiple Task Performance, CRC Press, pp 217–278

Flanagan M, Howe D, Nissenbaum H (2008) Embodying values in technology: theory and practice. In: Van den Hoven J, Weckert J (eds) Information Technology and Moral Philosophy, Cambridge University Press, pp 322–353

Flemisch F, Heesen M, Hesse T, Kelsch J, Schieben A, Beller J (2012) Towards a dynamic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations. Cogn Technol Work 14(1):3–18

Friedman B, Hendry D (2012) The envisioning cards: a toolkit for catalyzing humanistic and technical imaginations. Proceedings of the 2012 ACM annual conference on human factors in computing systems, ACM, pp 1145–1148

Friedman B, Kahn Jr PH, Borning A (2013) Value sensitive design and information systems. In: Doorn N, Schuurbiers D, van de Poel I, Gorman ME (eds) Early engagement and new technologies: opening up the laboratory, Springer, Netherlands, pp 55–95

Galy E, Cariou M, Mélan C (2012) What is the relationship between mental workload factors and cognitive load types? Int J Psychophysiol 83(3):269–275

Gawron VJ (2008) Human performance, workload, and situational awareness measures handbook. CRC Press, Boca Raton

Hankins TC, Wilson GF (1998) A comparison of heart rate, eye activity, EEG and subjective measures of pilot mental workload during flight. Aviat Space Environ Med 69(4):360–367

Harbers M, Neerincx MA (2014) Value sensitive design of automated workload distribution support for traffic control teams. Proceedings of the international conference on human computer interaction

Harbers M, Van den Bosch K, Meyer J-J (2010) Design and evaluation of explainable BDI agents. In: Proceedings of International Conference on Web intelligence and intelligent agent technology, vol. 2, IEEE, pp 125–132

Harbers M, Aydogan R, Jonker CM, Neerincx MA (2014) Sharing information in teams: giving up privacy or compromising on team performance? In Proceedings of the 2014 international conference on autonomous agents and multi-agent systems, IFAAMAS, pp 413–420

Harbers M, Detweiler C, Neerincx MA (2015) Embedding stakeholder values in the requirements engineering process. Springer, Berlin, pp 318–332

Hindriks KV (2009) Programming rational agents in GOAL. In: El Fallah Seghrouchni A, Dix J, Dastani M, Bordini RH (eds) - Multi-Agent Programming: Languages and Tools and Applications, Springer, pp 119–157

Hogervorst MA, Brouwer AM, van Erp JB (2014) Combining and comparing EEG, peripheral physiology and eye-related measures for the assessment of mental workload. Front Neurosci 8:322

Laird JE (2012) The Soar cognitive architecture. MIT Press, Cambridge

Lederer S, Mankoff J, Dey AK (2003) Who wants to know what when? Privacy preference determinants in ubiquitous computing. In CHI'03 extended abstracts on human factors in computing systems, ACM, pp 724–725

Levin S, Aronsky D, Hemphill R, Han J, Slagle J, France DJ (2007) Shifting toward balance: measuring the distribution of workload among emergency physician teams. Ann Emerg Med 50(4):419–423

Looije R, Neerincx MA, Hindriks KV (2016) Specifying and testing the design rationale of social robots for behavior change in children. Cognit Syst Res. doi:10.1016/j.cogsys.2016.07.002

Longo L (2015) A defeasible reasoning framework for human mental workload representation and assessment. Behav Inform Technol 34(8):758–786

Manders-Huits N (2011) What values in design? The challenge of incorporating moral values into design. Sci Eng Ethics 17(2):271–287

Mehrabian A (1996) Pleasure-arousal-dominance: a general framework for describing and measuring individual differences in temperament. Curr Psychol 14(4):261–292

Meijer S (2012) Gaming simulations for railways: lessons learned from modeling six games for the Dutch infrastructure management. In Infrastructure design, signaling and security in railway, pp 275–294

Meshkati N, Hancock PA (eds) (2011) Human mental workload. Elsevier, Amsterdam

Mesmer-Magnus JR, DeChurch LA (2009) Information sharing and team performance: a meta-analysis. J Appl Psychol 94(2):535

Miller JK, Friedman B, Jancke G (2007) Value tensions in design: the value sensitive design, development, and appropriation of a corporation's groupware system. Proceedings of the 2007 international ACM conference on supporting group work, ACM, pp 281–290

Mioch T, Smets NJ, Neerincx MA (2012) Predicting performance and situation awareness of robot operators in complex situations by unit task tests. In ACHI 2012, The fifth international conference on advances in computer–human interactions, pp 241–246

Nathan LP, Klasnja PV, Friedman B (2007) Value scenarios: a technique for envisioning systemic effects of new technologies. In CHI'07 extended abstracts on Human factors in computing systems, ACM, pp 2585–2590

Neerincx MA (2003) Cognitive task load design: model, methods and examples. In: Hollnagel E (ed) Proceedings of the International Conference on Handbook of cognitive task design, CRC Press, pp 283–305

Neerincx MA (2007) Modelling cognitive and affective load for the design of human–machine collaboration. Proceedings of the International Conference on Engineering Psychology and Cognitive Ergonomics, Springer, Berlin, pp 568–574

Neerincx MA (2011) Situated cognitive engineering for crew support in space. Pers Ubiquit Comput 15(5):445–456

Neerincx MA, de Greef HP (1998) Cognitive support: extending human knowledge and processing capacities. Hum Comput Interact 13:73–106

Neerincx MA, Lindenberg J (2008) Situated cognitive engineering for complex task environments. In: Schraagen JM, Militello L, Ormerod T, Lipshitz R (eds) Naturalistic decision making and macrocognition, Ashgate Publishing Limited, Aldershot, pp 373–390

Neerincx MA, Harbers M, Lim D, Van der Tas V (2014) Automatic feedback on cognitive load and emotional state of traffic controllers. Proceedings of the international conference on human computer interaction

Nissenbaum H (2009) Privacy in context: technology, policy, and the integrity of social life. Stanford University Press, Palo Alto

Peeters MM, Harbers M, Neerincx MA (2016) Designing a personal music assistant that enhances the social, cognitive, and affective experiences of people with dementia. Comput Hum Behav 63:727–737

Poel I (2013) Translating values into design requirements. In: Michelfelder DP, Mc-Carthy N, Goldberg DE (eds) Philosophy and engineering: reflections on practice, principles and process, philosophy of engineering and technology. Springer, Berlin, pp 253–266

Porter CO, Hollenbeck JR, Ilgen DR, Ellis AP, West BJ, Moon H (2003) Backing up behaviors in teams: the role of personality and legitimacy of need. J Appl Psychol 88(3):391

Reid GB, Nygren TE (1988) The subjective workload assessment technique: a scaling procedure for measuring mental workload. Adv Psychol 52:185–218

Salas E, Cooke NJ, Rosen MA (2008) On teams, teamwork, and team performance: discoveries and developments. Hum Factors: J Hum Factors Ergon Soc 50(3):540–547

Siegel AW, Schraagen JMC (2014) Measuring workload weak resilience signals at a rail control post. IIE Trans Occup Ergon Hum Factors 2(3–4):179–193

Siegel AW, Schraagen JM (2017) Team reflection makes resilience-related knowledge explicit through collaborative sensemaking: observation study at a rail post. Cogn Technol Work 19(1):127–142

Spiekman ME, Haazebroek P, Neerincx MA (2011) Requirements and platforms for social agents that alarm and support elderly living alone. In: Mutlu B, Bartneck C, Ham J, Evers V, Kanda T (eds) Social robotics, Springer, Berlin, pp 226–235

Sycara K, Lewis M (2004) Integrating intelligent agents into human teams. In Team cognition: understanding the factors that drive process and performance, Washington, DC, APA, pp 203–231

Van Broekhoven R, Siegel AW, Schraagen JM, Noordzij ML (2016) Comparison of real-time relative workload measurements in rail signalers. In Proceedings of the 2nd German conference on rail human factors, pp 30–40

Van den Hoven J (2007) ICT and value sensitive design. In The information society: Innovation, legitimacy, ethics and democracy in honor of Professor Jacques Berleur SJ, Springer, pp 67–72