Pers Ubiquit Comput (2011) 15:445–456
DOI 10.1007/s00779-010-0319-3

ORIGINAL ARTICLE

Situated cognitive engineering for crew support in space

Mark A. Neerincx

Received: 1 October 2008 / Accepted: 1 October 2009 / Published online: 13 July 2010

Abstract Space crews are in need for excellent cognitive support to perform nominal and off-nominal actions. This paper presents a coherent cognitive engineering methodology for the design of such support, which may be used to establish adequate usability, context-specific support that is integrated into astronaut's task performance and/or electronic partners who enhance human–machine team's resilience. It comprises (a) usability guidelines, measures and methods, (b) a general process guide that integrates task procedure design into user interface design and a software framework to implement such support and (c) theories, methods and tools to analyse, model and test future human–machine collaborations in space. In empirical studies, the knowledge base and tools for crew support are continuously being extended, refined and maintained.

Keywords Cognitive engineering · Human–machine collaboration · Electronic partner · Crew support · Usability · Space missions

Abbreviations

AI	Artificial Intelligence
CE	Cognitive Engineering
EVA	ExtraVehicular Activities
ISS	International Space Station
MECA	Mission Execution Crew Assistant
SCOPE	Supporting Crew OPErations
SUITE	Situated Usability engineering for Interactive Task Environments
VCP	Virtual Control Panel

1 Introduction

Current and future task environments of astronauts are complex and high demanding, both for activities in space laboratories and extravehicular activities (EVA; maybe even at planetary surfaces). Current technology allows astronauts to operate in such environments, supporting the crew to conduct experiments, perform maintenance and deal with anomalies. However, inadequate usage of technology will result in decreased task performance and may even increase the risks for astronauts' health. So, there is a clear need for a concise and coherent design approach for the space domain that guarantees *usability* of the in-orbit user interfaces ([1]; NASA Appendix H, [2]).

In addition to the requirement of excellent usability, the technology should provide excellent *cognitive support* to perform the nominal and off-nominal actions. Astronauts have to do a diverse set of activities according to predefined procedures, but should also show adequate responses to unforeseen situations or system failures. Context-specific support—integrated into astronaut's task performance—is required in both situations to provide the right information (e.g., procedure or alert) at the right time (e.g., fit with astronaut's agenda or responsibility) and in the right way (e.g., in browser on the screen or via the audio system) [3].

Future manned planetary exploration missions ask for increased human–machine crew autonomy, in which

M. A. Neerincx (✉)
TNO Human Factors, P.O.Box 23,
3769 ZG Soesterberg, The Netherlands
e-mail: mark.neerincx@tno.nl

M. A. Neerincx
Department of Mediamatics, Faculty of EEMCS,
Delft University of Technology, Mekelweg 4,
2628 CD Delft, The Netherlands

electronic partners (*e*Partners) cooperate with the astronauts to accomplish safe, effective and efficient operations. The distributed personal *e*Partners help the team to assess the situation, to determine a suitable course of actions to solve a problem, and to safeguard the astronauts from failures. Overall, human–machine's team resilience will be substantially enhanced [4].

In sum, crew support needs are increasing, starting from usability, via cognitive support to partnership. Technological progress (e.g., in artificial intelligence, AI) provides more and more opportunities to meet the needs for more advanced support. This paper presents a coherent cognitive engineering (CE) methodology that first addresses the basic user interface design issues and subsequently—if appropriate—the more advanced types of assistance. This methodology prescribes an iterative development process that integrates task procedure design into user interface design, provides a software framework to implement the proposed cognitive support and exemplifies methods and tools to analyse, model and test future human–machine collaborations in space. Section 2 discusses the operational needs and support technology in more detail. Section 3 presents the methodology and some best practices. Section 4 contains the conclusions and discussion.

2 Background

2.1 Human–machine collaboration

"Classical" CE methods consist of an iterative process of generation, evaluation and refinement of design specifications [5–7]. Technological progress enables the development of cognitive systems that consist of human and synthetic actors who collaborate for successful attainment of their joint operation objectives (e.g., [8]). To address the opportunities and constraints for such human–machine collaboration, we propose to combine the classical human-centred perspective with a technology-centred perspective so that systematically account can be taken of the adaptive nature of both human and synthetic actors with their reciprocal dependencies. Furthermore, the CE method should entail an explicit transfer and refinement of general state-of-the-art theories and models, which include accepted features of human cognitive and affective processes, into situated support functions for the specific operational contexts [9]. In this way, the situated CE method can coherently address the interaction between human cognition, technology and context.

2.2 Space laboratories

In manned space laboratories, astronauts supervise scientific experiments for various research institutes around the world. These institutes provide specific equipment (i.e. the payload) and corresponding procedures for conducting the experiments and for maintenance of the equipment. The laboratories are a good example of the problems that appear during the design of operation support in complex task environments: the involvement of diverse stakeholders, the implementation of diverse applications (platform systems and so-called payloads), the differences in design approaches and the separation of a task and a user interface design community. In previous manned space missions, procedural support, the mapping of task procedures on the user interfaces, the usability of the individual systems (including the fit to "context of use") and the consistency between interfaces showed serious shortcomings [10]. This resulted in extensive training and preparation efforts and non-optimal task performance of the astronauts and cosmonauts. It should be noted that operations onboard the International Space Station (ISS) are being performed more and more in a paperless environment. The crew laptop is hosting more and more instruction material containing crew procedures with associated reference documentation and interactive payload virtual control panels (VCP) for supervision, monitoring and control.

2.3 Future exploration missions

Different scenarios for manned long-duration missions to the Moon and Mars have been developed. Such scenarios set high operational, human factors and technical demands for a distributed support system, which enhances human–machine teams' capabilities to establish safe and effective operations under nominal and off-nominal conditions. The Mission Execution Crew Assistant (MECA) project aims at such a system by empowering the cognitive capacities of human–machine teams during planetary exploration missions in order to cope autonomously with unexpected, complex and potentially hazardous situations. An elaborate and sound method for requirements analysis has been developed and applied, focusing on a manned mission to Mars. It should be noted that the project outcomes are of relevance for manned space missions where a greater need for autonomy exists (i.e., most outcomes also apply to Moon missions, and a substantial part is relevant for International Space Station missions and ground-based control missions of planetary robots). A central concept is the notion of *electronic partners* (*e*Partners), helping the crew to assess the situation, determine a suitable course of actions to solve a problem and safeguard the astronaut from failures [11]. This concept comprises a collection of distributed and connected personal *e*Partners to support the (distributed) crew members during exploration missions. A personal *e*Partner predicts its crew members momentary support needs by on-line gathering and modelling of human, machine, task and context information. Based on these models, it attunes the user interface to these needs in order to establish optimal human–machine performance. The user interface of the *e*Partner is "natural or intuitive" by expressing and interpreting communicative acts based on a common reference of the human and machine actors.

3 Situated cognitive engineering methodology

Cognitive engineering is an *iterative* process with active involvement of end-users (or representatives) to better understand their support needs and to enhance user acceptance. Furthermore, it is a *collaborative* process in which experts from different disciplines contribute to address the operational, human factors, and technical issues (and possibly relevant statutory or legislative issues). In our methodology, it is also a process in which the support functions are *incrementally* developed, providing increasingly more "intelligent" crew assistance. First, this section describes the usability framework as a minimal guideline for user interface design. Subsequently, we summarize the process and methods for cognitive support design and, finally, additional requirements for the development of so-called electronic partners.

3.1 Usability framework

Neerincx and others [3] developed a usability framework to integrate human factors knowledge into the software development process. The usability framework can be viewed as a customization of general usability engineering approaches [12–14] and scenario-based design techniques [15], addressing the development process requirements of ISO-13407 "human-centred design processes for interactive systems" (ISO: International Standardisation Organisation). Adapted from ISO 9241, Part 11, [16] and the NASA practices (NASA Appendix H, [2]), the framework distinguishes four *usability objectives*. First, *effectiveness* is the degree of success (i.e., accuracy and completeness) with which users achieve their task goals. Second, *efficiency* concerns the amount of resources required for task completion (e.g., time and mental effort). The third and fourth objectives are to establish user's *satisfaction* (user comfort and acceptability) and *learnability* (i.e., the resources expended to acquire and maintain the knowledge and skills for effective and efficient operations).

According to the usability framework, user interfaces can be described and assessed at two levels. At the first level, based on users' goals and information needs, the system's functions and information provision are specified or assessed (i.e. the *task level* of the user interface). At the second level, the control of the functions and the presentation of the information are specified or assessed (i.e. the "look-and-feel" or the *communication level* of the user interface).

The usability framework distinguishes five general design guidelines at *the task level* of the user interface:

1. *User fit*. The user interface design should take account of both the general characteristics of human perception, information transfer, decision-making and control and (the variation of) specific user characteristics with respect to education, knowledge, skills and experience.
2. *Goal conformance*. There should be an appropriate allocation of functions to human and system (hardware and software), addressing human and system capabilities and the particular task requirements (e.g., a robust system should provide the functions to an operator that corresponds to his or her responsibilities, knowledge and skills, in particular for safety critical systems). The functions and function structure of the user interface should map, in a one-to-one relation, on users' goals and corresponding goal sequences. Functions that users do not need should be hidden from these users.
3. *Information needs conformance*. The information that is provided by the user interface should map, in a one-to-one relation, on the information needs that arise from users' goals. Irrelevant information should not be presented to the users.
4. *User's complement*. The user interface should provide cognitive support to extend users' knowledge and capacities when needed. For example, to improve the users' effectiveness, the interface should extend the user's expertise by providing task knowledge and in order to improve efficiency the interface may take over routine actions.
5. *Work context*. The human-computer interaction should fit to the envisioned work context and/or situation, and the context dynamics should be taken into account for the four guidelines described earlier (e.g., the requirements of dual tasks, pilot and night-day working schedules).

The usability framework distinguishes eight guidelines at the *communication level* that are concerned with more detailed user interface issues:

1. *Consistency*: The differences in dialogue should be minimal within a user interface and across user interfaces that are used by the same persons.
2. *Compatibility*: Dialogue styles should correspond to the knowledge, skills and expectations of the users so that the amount of information re-coding is minimal (cf. intuitive).
3. *Usage context*: The user interface should fit with the momentary usage context. Whereas task-level guideline

5 on “work context” centres on the environmental dynamics that may interfere with the task and information structure, the “usage context” guideline concerns the specific input and output characteristics of the user interface (e.g., fit of brightness and colouring schemes to the lighting conditions, of character size to user-display distance and of alarm sounds to background noise).

4. *Structure and pattern*: Imposed dialogue sequences should correspond to users’ strategies so that they can navigate through the interface easily and can execute functions or procedures in an adequate way.
5. *Feedback and mode awareness*: The user interface should provide the user with feedback about the current state, action and result, both for the actual interface (e.g., menu) and the underlying process (e.g., cool water flow) at all levels of descriptions. This information can apply to the current situation but can also concern future situations (such as predictive displays). For critical operations, the user interface can ask for confirmation.
6. *Interaction load*: The user should be able to process the information that is provided by the interface without excessive (physical and mental) effort and repetition.
7. *Integrated support*: The support should be both integrated in the task performance and easily accessible for consultation and preparation independent from the actual performance.
8. *User control and tailoring*: It should be possible to accommodate individual differences among users through adaptation (user initiated or automatic) of the interface. Such accommodations could be “simple” (e.g., colour and contrast settings of displays). However, it should *not* introduce (a) the risk for new user errors or (b) hindrances to share a common representation by a group of users.

3.2 Design of integrated task support

3.2.1 Process guide

Historically, the operations group within a development team specified procedures, whereas the software group focused on the display design for space systems. Display and (electronic) procedure design are still rather separated in the current development practices of the first ISS payloads, resulting in different types of interaction for the displays and procedures. There is a clear need to better synchronize the activities of the two development groups to establish coherence in the user interfaces, correspondence between procedure specifications and interfaces, and an adequate mapping of user tasks (or goals) on the interfaces. To establish this synchronization and to systematically address the 13 guidelines of Sect. 3.1, Neerincx and others [10] developed a task-based, top-down, iterative design process (see Fig. 1), which consists of three phases—analysis, design and implementation—specifying and assessing the procedures and user interfaces at the task, communication and implementation level, respectively. The process results in three types of deliverables: requirements, usage descriptions and the resulting system (in which requirements and usage descriptions are reflected). The analysis phase results in two main deliverables providing information at the task (functional) level: the requirements baseline and the operations manual. The design phase results in two main deliverables providing information at the communication (dialogue) level, the detailed design document and the flight operations products. These deliverables can be considered updates and refinements of the respective deliverables at the task level. Finally, the implementation phase results in the final deliverable, the system. In all three stages, the results of assessments may lead to updates of both requirements and usage information. In addition to the classical focus on user, task and context aspects, technical aspects are being addressed explicitly (such as software architecture). Recently, the process guide was included in the ECSS-E-ST-10-11C “Space Engineering: Human Factors Engineering” standard of the European Cooperation for Space Standardization (ECSS).

Prior to the assessments mentioned earlier, an optional ‘Assessment 0’ could be carried out in order to assess the project risks and establish the current state of technology. In projects of a longer duration, it is advisable to iterate this assessment to ascertain that newly emerging risks and technological developments are identified.

Each activity in the process has its own stakeholders, i.e. a specific set of software developers, user interface developers, user interface testers, procedures developers, flight crew, flight controllers and planners, hardware providers (e.g. payload developers) and principle investigators. In each phase and activity, specific human factors principles and guidelines, specification techniques, assessment techniques and technological issues should be applied. It should be emphasized that the analysis activities provide the foundation for both the procedures and the user interfaces (including displays and interaction support), which is efficient and effective (i.e. establishing consistency between procedures and interfaces). The CE know-how base has been selected on the basis of previous experiences with similar projects and an analysis of “best practices” of payload development. Table 1 gives an overview of the content of the CE method.

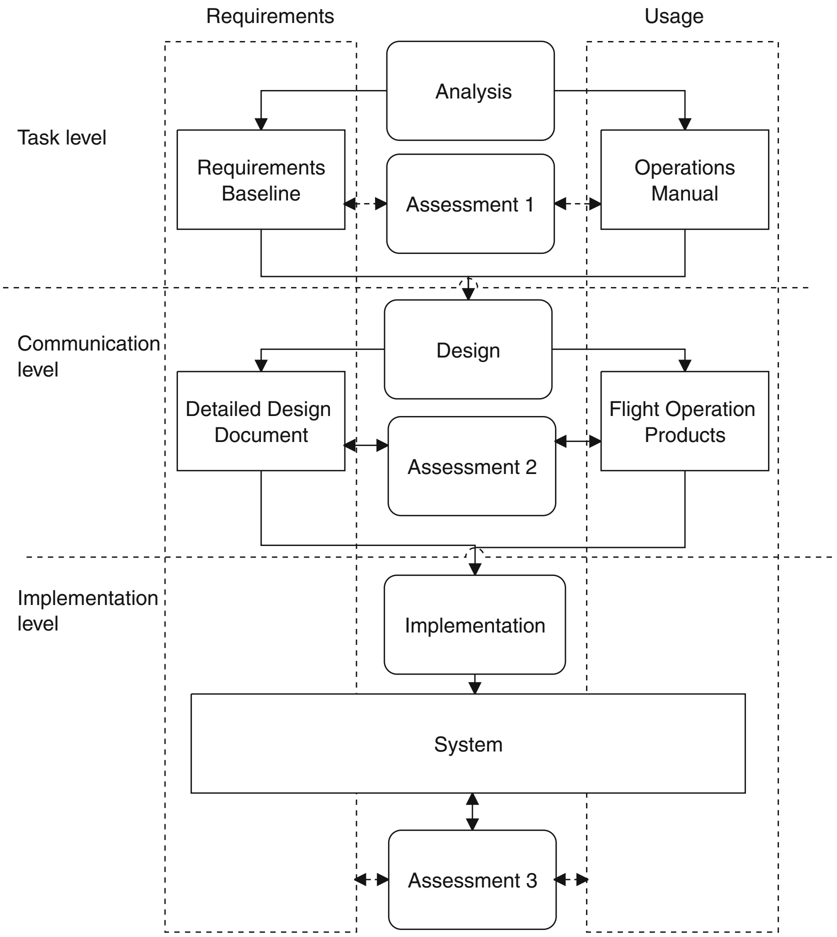


Fig. 1 Phases, activities and deliverables in the proposed CE-process

3.2.2 Cognitive engineering toolkit

For supporting the diverse stakeholders of the overall design process as described earlier, we developed a cognitive engineering toolkit called SUITE (Situated Usability engineering for Interactive Task Environments; [9]). In SUITE, the CE method of Section 3.2 was provided as an electronic handbook that contains context- and user-tailored views on the recommended human factors method, guidelines and best practices (that is, the development of procedures and user interfaces for three different payloads). Furthermore, SUITE provides a task support and dialogue framework, called Supporting Crew OPErations (SCOPE), as both an implementation of these methods and guidelines and an instance of current interaction and (AI) technology. This framework defines a common multimodal interaction with a system, including multimedia information access, virtual control panels, alarm management services and the integrated provision of context-specific task support for nominal and off-nominal situations. In addition to the support of supervision and damage control, it provides support to access and process multimedia information, for instance, for preparation of actions, training on the job or maintenance tasks. In the following paragraphs, we will briefly present some cognitive ingredients of SUITE: the support functions for hypermedia interaction and for task guidance.

Astronauts have to process a lot of information during space operations. For searching and navigating in hypermedia environments via manual and speech commands, we developed the following support functions that address the contextual- and individual-affected limitations on spatial ability and memory [17]:

1. The *Categorizing Landmarks* are cues that are added to the interface to support the users in recognizing their presence in a certain part of the multimedia content (that is, it arranges information into categories that are meaningful for the user's task). This should help the users to perceive the information in meaningful clusters and prevents the user from getting lost. For example, specific categories of information have a dedicated background colour; hyperlinks that refer to this content have the same colour.
2. The basis for the *History Map* is a "sitemap": a representation of the structure of the multimedia

Table 1 Overview of CE activities for the design and evaluation of operational support applied in different stages of the development process

Development stage	Specification techniques	Assessment techniques	Technological design space
Analysis	*Task decomposition*	Task load analysis	*Technology assessment*
	Task allocation	Interaction analysis	*Functional components*
	Task features	*User requirements validation*	
	Data model, flows		
	Scenarios, use cases		
	Action sequences		
	User requirements		
Design	Sketching	*Expert review*	*Hardware/software selection*
	Procedure structure	*User walkthrough*	*Architecture*
	User interface structure	Thinking aloud	
	Story-boarding	Constructive interaction	
	Dialogue definition	Usability test	
	Graphics definition		
	Prototyping		
	User interface requirements		
Implementation		*Consistency check*	*Algorithm design*
		Final test	*Coding*
		User feedback	

	Techniques to acquire the information for specification and assessment (in all stages)
	Focus groups
	Interview
	Questionnaire
	Observation
	Document analysis
	Comparative analysis
	Critical incidents
	Logging

Techniques in italics *should* be done always, whereas the other elements are recommended for more complex and/or critical systems

content. It shows a hierarchical overview or tree to indicate the location of the content currently being viewed and may include a separate presentation of the current “leaf” (“breadcrumb”). History information is annotated in the overview. This memory aid should improve users’ comprehension of the content structure in relation to their task and provide information about the status of their various subgoals.

3. A *Speech Command View* presents the specific commands that a specific user is allowed to use for controlling the current active part of the application. Current and most recent commands can be indicated in the view.

In order to provide integrated task support, we additionally developed the following two support functions:

1. The *Rule Provider* gives the normative procedure for solving (a part of) the current problem, complementing user’s procedural knowledge. Due to training and experience, people develop and retain procedures for efficient task performance. Performance deficiencies may arise when the task is performed rarely so that procedures will not be learned or will be forgotten or when the information does not trigger the corresponding procedure in human memory.
2. The *Diagnosis Guide* is an important support function of SCOPE. It detects system failures, guides the isolation of the root causes of failures and presents the relevant repair procedures in textual, graphical and multimedia formats [18]. The diagnosis is a joint astronaut-SCOPE activity: when needed, SCOPE asks the astronaut to perform additional measurements in order to help resolve uncertainties, ambiguities or conflicts in the current machine status model. SCOPE will ask the user to supply values to input variables it has no sensors for measuring by itself. Each new

question is chosen on the basis of an evaluation function that can incorporate both a cost factor (choose the variable with the lowest cost) and a usefulness factor (choose the variable that will provide the largest amount of new information to the diagnosis engine). After each answer, the diagnosis re-evaluates the possible fault modes of the system on the basis of the additional values (and new samples for the ones that can be measured). As soon as SCOPE has determined the most likely health state(s) of the system with sufficient probability, it presents these states to the user, possibly with suggestions for appropriate repair procedures that can be added to the todo list and executed. As soon as the machine has been repaired, SCOPE will detect and reflect this.

3.2.3 *Iterations*

According to our general cognitive engineering approach, we apply and refine SUITE during "real" development processes. The number of astronauts is relatively small, and, furthermore, it is difficult to involve these busy people in ongoing research. An evaluation with other participants is possible if their task knowledge and the task context in the evaluation reflect the crucial performance factors of the real task domain.

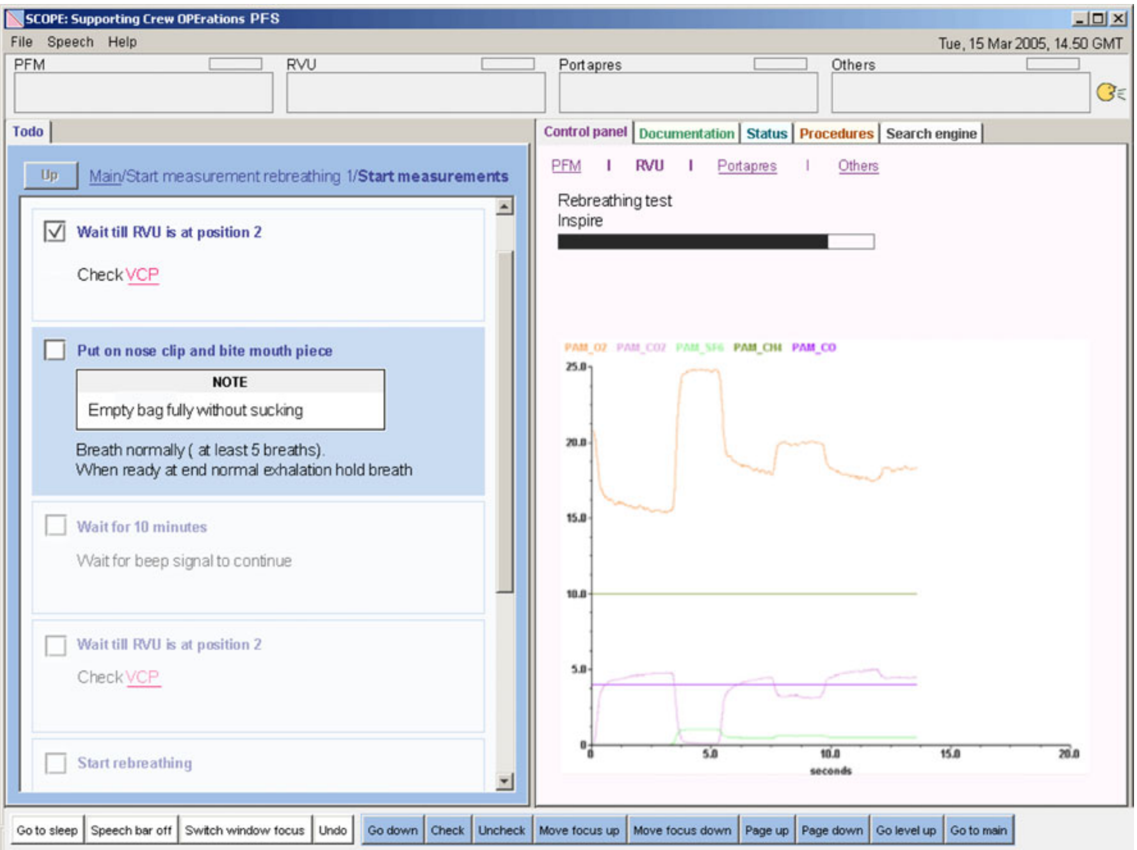
First, we conducted an evaluation of a prototype user interface for "chemistry and physics payloads" with the three types of astronauts' tasks: conduct experiment, keep up maintenance and deal with anomalies (see [3] for the details). Forty-five students in physics and chemistry participated in a controlled evaluation of the concerned prototype elements as a first validation of the example interface design. The integration of procedures into the rest of the user interface and the navigation support proved to substantially increase the efficiency of payload operations. Navigation support resulted initially in faster general task performance (efficiency), but worse effectiveness. So, this kind of support seems to cause initially relatively fast navigation, which possibly leads to less effectiveness due to a speed/accuracy trade-off. However, it also brought about a positive learning effect on effectiveness, whereas no learning effect on this performance measure was present for the condition without navigation support. So, a minimal level of training is required to keep users in the loop of instructed task performance.

Second, we developed a "real" application for the Cardiopres payload (a system for continuous physiological measurement, such as blood pressure and ECG, which will be used in several space missions). The user interface of the prototype was running on a Tablet PC with direct manipulation and speech dialogue. Via a domain analysis and technology assessment, we defined the scope of this prototype (that is, the scenarios and technology to be implemented). The prototype design (that is, the user interface framework) was tested via expert reviews, user walkthroughs with astronauts and a usability test with 10 participants measuring effectiveness, efficiency, satisfaction and learnability. In the evaluations of the SCOPE system for the Cardiopres, the user interface and cooperative task support functions proved to be effective, efficient and easy to learn, and astronauts were very satisfied with the system [10].

Subsequently, the SCOPE framework was applied for the development of an intelligent user interface for the Pulmonary Function System (PFS) payload [19]. Its task support functions were improved to deal with dependencies of actions with each other and the usage context. In general, the PFS prototype showed that the SCOPE framework can be applied for a diverse set of payloads. We concluded that the SUITE toolkit reduces the time and cost of development efforts, whereas it improves the usability of user interfaces that provide integrated task support. Embedded in a cognitive engineering process, user interfaces and the underlying AI methods are systematically and coherently specified, implemented and assessed from early development phases on, which is in itself efficient and prevents the need for late harmonization efforts between user requirements and technological constraints.

3.3 Electronic partners

Figure 2 shows a joint human and machine task performance, in which the machine can execute subtasks automatically and can schedule tasks for the astronaut. In the Mission Execution Crew Assistant (MECA) project, we extended the concept of human–machine collaboration and the corresponding situated CE methodology to establish a theoretically sound and empirically proven requirements baseline for a distributed support system that contains *e*Partners. Currently, the MECA focus is mainly on the analysis phase. However, in order to address the reciprocal adaptive nature of human and synthetic actors, design activities and some form of prototyping have to be included in the requirements analysis. It should be emphasized that the results of such design activities (e.g., the prototype) are "only" tools for the refinement and validation of requirements and not meant as interim product of the final system. The process of requirements specification, refinement and validation is based on a work domain and support analysis and analytical and empirical assessments (cf, the 'analysis' and 'assessment 1' activities to derive a requirements baseline in Fig. 1).



Main/Start measurement rebreathing 1/**Start measurements**

☑ **Wait till RVU is at position 2**

Check VCP

□ **Put on nose clip and bite mouth piece**

NOTE

Empty bag fully without sucking

Breath normally (at least 5 breaths).
When ready at end normal exhalation hold breath

□ **Wait for 10 minutes**

Wait for beep signal to continue

□ **Wait till RVU is at position 2**

Check VCP

□ **Start rebreathing**

Rebreathing test
Inspire

Fig. 2 SCOPE showing a "joint machine-astronaut" procedure (*left*) and control panel (*right*)

3.3.1 Analysis

First, the *mission and domain analysis* consists of a meta-analysis of studies of previous (e.g., [20, 21]) and future operations of manned space missions (e.g., [22–26]). This analysis showed that the performance of astronauts—who have excellent competencies, are well-trained and have a high work motivation—can deteriorate due to diminished motor, perceptual and cognitive capacities and emotional or social-psychological problems. There is a clear need to support the crew for both nominal and off-nominal operations among other things to accommodate team-members creative problem solving processes and counter-balancing initiatives to share or take over specific critical tasks. In addition, a Work Domain Analysis was performed. This analysis is the first phase of the Cognitive Work Analysis, aiming at socio-technical systems that can fully exploit the problem-solving capabilities and adaptability of human resources in unexpected situations, improving efficiency, productivity and safety. The analysis is event-independent, based on the notion that it is impossible to predict all possible system states. Systems are therefore defined in terms of their environmental and cognitive constraints: their physical environment, priorities and functionality [27, 28]. For MECA, the Work Domain Analysis provided an Abstraction Decomposition Space offering substantial insight into the properties or and relationships within the Mars surface mission system. Outcomes were compared with outcomes of the mission analysis and the first version of the Requirements Baseline. The main conclusion was that the Requirements Baseline considered all technical aspects of the Mars surface mission in sufficient detail but that further specifications were required in the areas of general living [29].

Second, a *human factors analysis* was conducted to address the generic support needs of well-trained human operators who act in complex high-demanding task environments, possibly in extreme and hostile situations (such as the defence and safety domains). This study provided key issues that MECA should address for supporting the human–machine collaboration: cognitive task load [30], situation awareness [31], sense-making [32], decision-making [33], diversity of cognitive capacities [34], trust [35], emotion [36], collaboration [37] and crew resource management [38].

Third, via a *technology assessment,* we identified key technologies. It was concluded that MECA will act in a Smart Task Environment with automatic distribution of data, knowledge, software and reference documents. However, MECA should still provide operational support—based on history and current available information and knowledge—when infrastructure failures occur. It will apply state-of-the-art Agent and Web technology, model-based reasoning and health management, human–machine (e.g. robot) collaboration and mixed reality.

In conclusion, the work domain and support analysis identified operational, human factors and technological challenges of manned planetary space missions and, subsequently, derived the general MECA support concept out of it with specific support functions. Core MECA support functions concern health management, diagnosis, prognosis & prediction, collaboration, resource management, planning and sense making. The concept and functions were exemplified in a set of scenarios. Figure 3 shows part of a scenario in which the *e*Partner of an astronaut helps to diagnose a space suit problem, notifies other human and synthetic actors of the problem and asks for specific resource deployments (such as a rover for transportation). The scenarios were annotated with a coherent set of claims on the expected operational benefits of the support functions. The first three types of claims consist of standard usability measures, while the subsequent three types of measures concern additional human experience and knowledge measures:

- *Effectiveness and efficiency* will be improved both for nominal situations and anomalies, because MECA extends astronauts cognitive resources and knowledge (e.g. procedure and planning).
- Astronauts will express high *satisfaction* for the MECA support, because (a) it is based on human–machine partnership principles (e.g., for sharing of knowledge) and (b) the astronauts remain in control.
- Working with MECA will be *easy to learn*, because the support is integrated into the task execution and can be accessed via intuitive multimodal user interfaces (application of visual, auditory and tactile modalities).
- *Situation awareness* will be enhanced by an overview of relevant situation knowledge with the current plan and the provision of context-sensitive notification mechanisms.
- Astronauts show appropriate *trust* levels for MECA, because they share knowledge via situated models that astronaut can easily access and understand.
- MECA accommodates *emotional responses* appropriately in critical situations.

3.3.2 Assessment

We tested the claims via expert and task-analytical reviews and via human-in-the-loop evaluations of a simulation-

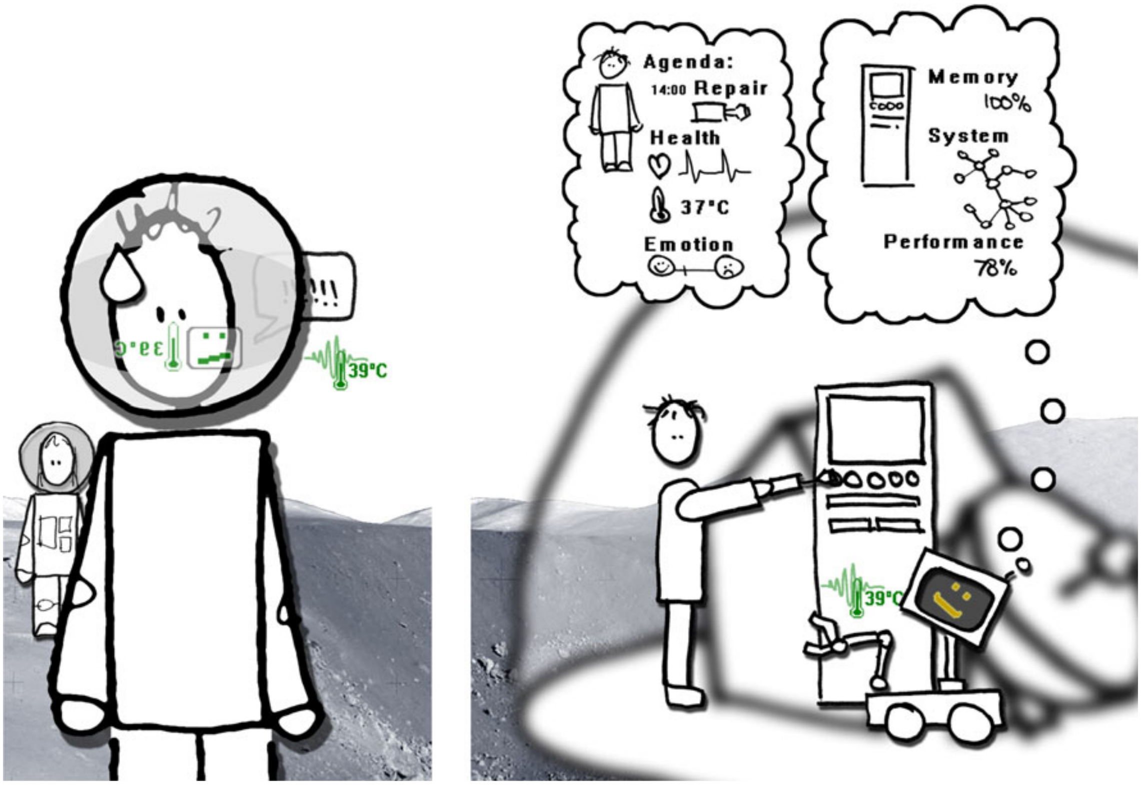


Fig. 3 Part of a MECA scenario in which ePartners help to solve a space suite problem

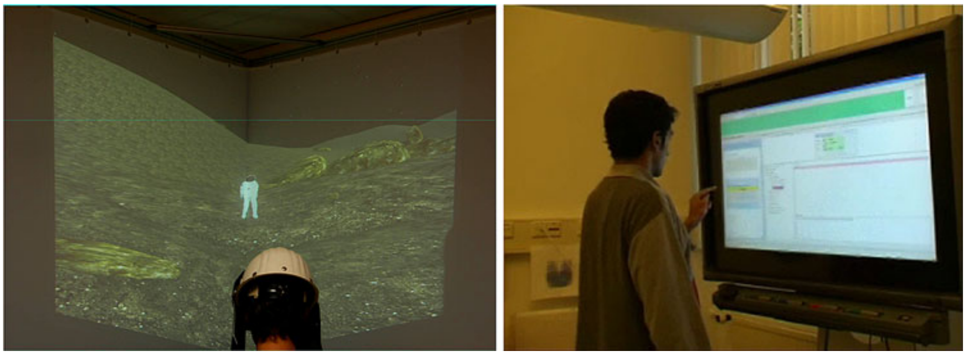


Fig. 4 These pictures show two participants playing different roles in the scenario (*left picture* EVA at planetary surface; *right picture* operations in planetary habitat)

based prototype in a virtual environment (see Fig. 4; for an overview of the review and simulation-based methods, see [39]). In general, the evaluation results confirmed the claims on effectiveness, efficiency, satisfaction, learnability, situation awareness, trust and emotion. Issues for improvement and further research were identified and prioritized (e.g., crew acceptance of mental load and emotion sensing). In general, the situated CE method provided a reviewed set of 167 high-level requirements that explicitly refers to the tested scenarios, claims and core support functions on health management, diagnosis, prognosis and prediction, collaboration, resource management, planning and sense-making. A first version of an ontology for this support was implemented in the prototype, which will be used for further *e*Partner development.

4 Conclusions and discussion

This paper presented a situated cognitive engineering (sCE) methodology for the design of user interfaces, cognitive support and human–machine collaboration, aiming at adequate usability, context-specific support that is integrated into astronaut's task performance and/or electronic partners who enhance human–machine team's resilience. It comprises (a) usability guidelines, measures and methods, (b) a general process guide that integrates task procedure design into user interface design and a software framework to implement such support, and (c) theories, methods and tools to analyse, model and test future heterogeneous human–machine cognitive systems. In empirical studies, the knowledge-base and tools for crew support are continuously being extended, refined and maintained. This knowledge-base is tailored to the specific needs of the application domain. The development process facilitates the contribution of different expertise types (for example, interaction design, software technology, and payload) at the right time (that is, as early as possible), whereas the use scenarios facilitate the communication between the different stakeholders.

The empirical studies show that the CE methodology provides support functions that improve the effectiveness and efficiency of both nominal and off-nominal operations. In general, it should be stressed that the in-orbit environment and working conditions are extreme, requiring an extra need to "situate" the support. For example, due to microgravity, a restricted living area and unusual day–night light patterns, there is sleep deprivation leading to reduced capacities to process information and a corresponding increased need for support.

For more advanced support functions, the situated CE approach focuses on the performance of the mental activities of human actors and the cognitive functions of synthetic actors to achieve the (joint) operational goals. In this way, the notion of collaboration has been extended, incorporating social synthetic actors that can take initiative to act, critique or confirm in joint human–machine activities. This way the envisioned MECA system seems to be able to substantially enhance human–machine teams' capabilities to cope autonomously with unexpected, complex and potentially hazardous situations. We specified a sound—theoretical and empirical founded—set of requirements for such a system and its rational consisting of scenarios and use cases, user experience claims and core support functions. For further development of the MECA knowledge base and its ontologies, we will conduct evaluations of long duration missions (e.g., Mars 500; [40]), for both off-nominal and nominal situations, with crews in relatively isolation and with more astronaut involvement (cf. [41]).

By evaluating the cognitive functions in a systematic way, a "library" of best practices is being built with the sCE methodology. Current research and development projects will provide intelligent user interfaces to train astronauts for the International Space Station, support crew's situation and self-awareness and improve crew's resilience in long-duration missions. The design rationale of the support functions is being founded theoretically and empirically, so that they can be generalized for corresponding work contexts relatively easy (e.g. the submarine domain).

Acknowledgments A large number of persons contributed to the research that is presented in this paper: Anita Cremers, Kim Kranenborg, Jasper Lindenberg, Mark Ruijsendaal, Nanja Smets, Leo Breebaart, André Bos, Antonio Olmedo Soler, Tim Grant, Uwe Brauer and Mikael Wolff. Furthermore, several astronauts, domain and task experts and MSc students participated in parts of the study. SUITE and MECA are developments funded by the European Space Agency (respectively contract C16472/02/NL/JA and contract 19149/05/NL/JA).

References

1. Flensholt J, Neerincx MA, Ruijsendaal M, Wolff M (1999) A usability engineering method for international space station onboard laptop interfaces. Conf. Proc. DAISA'99 Data Systems in Aerospace (SP-447). ESA/ESTEC, Noordwijk, pp 439–446
2. NASA Appendix H (1998) Appendix H–Payload displays of the document DGCS (Display and Graphics Commonality Standards), International Space Station program document (SSP) 530313, Rev Draft, Sep 1998
3. Neerincx MA, Ruijsendaal M, Wolff M (2001) Usability engineering guide for integrated operation support in space station payloads. Int J Cogn Ergon 5(3):187–198
4. Neerincx MA, Lindenberg J, Smets N, Grant T, Bos A, Olmedo Soler A, Brauer U, Wolff M (2006) Cognitive engineering for long duration missions: human–machine collaboration on the moon and mars. SMC–IT 2006: 2nd IEEE International Conference on Space Mission Challenges for Information Technology. IEEE Conference Publishing Services, Los Alamitos, California pp 40–46
5. Hollnagel E, Woods DD (1983) Cognitive systems engineering: new wine in new bottles. Int J Man Machine Stud 18:583–600
6. Norman DA (1986) Cognitive engineering. In: Norman DA, Draper SW (eds) User-centered system design: new perspectives on human–computer interaction. Lawrence Erlbaum Associates, Hillsdale, pp 31–62
7. Rasmussen J (1986) Information processing and human–machine interaction: an approach to cognitive engineering. Amsterdam, Elsevier
8. Hoc J-M (2001) Towards a cognitive approach to human-machine cooperation in dynamic situations. Int J Hum Comput Stud 54(4):509–540
9. Neerincx MA, Lindenberg J (2008) Situated cognitive engineering for complex task environments. In: Schraagen JMC, Militello L, Ormerod T, Lipshitz R (eds) Natural decision making & macrocognition. Ashgate Publishing Limited, Aldershot
10. Neerincx MA, Cremers AHM, Bos A, Ruijsendaal M (2004) A tool kit for the design of crew procedures and user interfaces in space stations (Report TM–04–C026). TNO Human Factors, Soesterberg
11. Neerincx MA, Bos A, Olmedo-Soler A, Brauer U, Breebaart L, Smets N, Lindenberg J, Grant T, Wolff M (2008) The mission execution crew assistant: improving human-machine team resilience for long duration missions. Proceedings of the 59th International Astronautical Congress (IAC2008), 12 pages. Paris, France: IAF. DVD: ISSN 1995-6258
12. Nielsen J (1993) Usability engineering. Morgan Kaufmann, San Francisco
13. Wixon D, Wilson C (1997) The usability engineering framework for product design and evaluation. In: Helander MG, Landauer TK, Prabhu PV (eds) Handbook of human-computer interaction, chapter 27, 2nd edn. Elsevier Science, Amsterdam, pp 653–688
14. Mayhew DJ (1999) The usability engineering lifecycle: a practitioner's handbook for user interface design. Morgan Kaufmann, San Francisco
15. Rosson MB, Carroll JM (2002) Scenario-based design. In: Jacko J, Sears A (eds) The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, Chap. 53. Lawrence Erlbaum Associates, pp 1032–1050
16. ISO 13407. Human-centered design processes for interactive systems. Geneva, Switzerland: International Organisation for Standardisation
17. Neerincx MA, Lindenberg J, Pemberton S (2001) Support concepts for Web navigation: a cognitive engineering approach. Proceedings Tenth World Wide Web Conference. ACM Press, New York, pp 119–128
18. Bos A, Breebaart L, Neerincx MA, Wolff M (2004) SCOPE: An intelligent maintenance system for supporting crew operations. Proceedings of IEEE Autotestcon, pp 497–503. ISBN 0-7803-8450-4
19. Bos A, Breebaart L, Grant T, Neerincx MA, Olmedo Soler A, Brauer U, Wolff M (2006) Supporting complex astronaut tasks: the right advice at the Right Time. SMC–IT 2006: 2nd IEEE International Conference on Space Mission Challenges for Information Technology. Los Alamitos, California: IEEE Conference Publishing Services, pp 389–396
20. Kanas NA, Salnitskiyb VP, Ritsher JB, Gushin VI, Weiss DS, Saylor SA, Kozerenko OP, Marmar CR (2007) Psychosocial interactions during ISS missions. Acta Astronautica 60:329–335
21. Kanas NA, Salnitskiy VP, Boyd JE, Gushin VI, Weiss DS, Saylor SA, Kozerenko OP, Marmar CR (2007) Crewmember and mission control personnel interactions during international space station missions. Aviat Space Environ Med 78(6):601–607
22. Hoffman SJ, Kaplan DI (eds) ((1997)) Human exploration of Mars: the reference mission of the NASA Mars Exploration Study Team. NASA Special Publication 6107. NASA Johnson Space Center, Houston
23. Hoffman SJ (ed) (2001) The mars surface reference mission: a description of human and robotic surface activities. TP-2001-209371, Dec 2001. NASA Johnson Space Center, Houston, TX
24. Engel K, Hirsch N, Junior A, Mahler C, Messina P, Podhajsky S, Welch CS (2004) Lunares: International Lunar Explroation in Preparation for Mars. In: Proc. of 55th Int. Astronautical Congress, Vancouver, Canada, pp 1–11
25. Kminek G (2004) Human Mars Mission Project: Human. surface operations on Mars, ESA/Aurora/GK/EE/004.04, issue 1, revision 1, June 2004. ESA Publications Division, Noordwijk
26. HUMEX (2003) HUMEX: A study on the survivability and adaptation of humans to long-duration exploration missions. ESA Special Publication 1264. ESA Publications Division, Noordwijk
27. Vicente KJ (1999) Cognitive work analysis: toward safe, productive, and healthy computer based work. Lawrence Erlbaum Associates, London
28. Naikar N, Sanderson PM (2001) Evaluating design proposals for complex systems with work domain analysis. Hum Factors 43(4):529–542
29. Baker C, Naikar N, Neerincx MA (2008) Engineering planetary exploration systems: integrating novel technologies and the human element using work domain analysis. Proceedings of the 59th International Astronautical Congress (IAC2008), 15 pages. Paris, France: IAF. DVD: ISSN 1995-6258
30. Neerincx MA (2003) Cognitive task load design: model, methods and examples. In: Hollnagel E (ed) Handbook of cognitive task design. Lawrence Erlbaum Associates, Mahwah, pp 283–305

31. Endsley MR (2000) Theoretical underpinnings of situation awareness. In: Endsley MR, Garland DJ (eds) Situation awareness analysis and measurement. LEA, USA
32. Weick K (1995) Sense-making in organisations. Sage, Thousand Oaks
33. Klein G (1998) Sources of power: how people make decisions. MIT Press, Cambridge
34. Scerbo MW (2001) Stress, workload and boredom in vigilance: a problem and an answer. In: Hancock PA, Desmond PA (eds) Stress, workload and fatigue. Lawrence Erlbaum Associates, New Jersey
35. Adams BD, Bruyn LE, Houde S, Angelopoulos P (2003) Trust in automated systems: literature review. DRDC Report No. CR-2003-096. Defence Research and Development, Toronto
36. Neerincx MA (2007) Modelling cognitive and affective load for the design of human-machine collaboration. In: Harris D (ed) Engineering psychology and cognitive ergonomics, HCII 2007, LNAI 4562. Springer, Berlin, pp 568–574
37. Mohammed S, Dumville BC (2001) Team mental models in a team knowledge framework: expanding theory and measurement across disciplinary boundaries. J Org Behav 22(2):89–106
38. Helmreich RL, Merritt AC, Wilhelm JA (1999) The evolution of crew resource management training in commercial aviation. Int J Aviat Psychol 9(1):19–32
39. Streefkerk JW, van Esch-Bussemakers MP, Neerincx MA, Looije R (2008) Evaluating context-aware mobile user interfaces. In: Lumsden J (ed) Handbook of research on user interface design and evaluation for mobile technology, Chap XLV. IGI Global, Hershey, pp 759–779
40. Rauterberg M, Neerincx MA, Tuyls K, van Loon J (2008) Entertainment computing in the orbit. In: Ciancarini P, Nakatsu R, Rauterberg M, Roccetti M (eds) IFIP International Federation for Information Processing, Vol. 279; New Frontiers for Entertainment Computing. Springer, Boston, pp 59–70
41. Clancey WJ (2007) Observation of work practices in natural settings. In: Ericsson A, Charness N, Feltovich P, Hoffman R (eds) Cambridge handbook on expertise and expert performance. Cambridge University Press, New York, pp 127–145