24th ICCRTS

24th International Command and Control Research and Technology Symposium

MANAGING CYBER RISK TO MISSION

October 29 – 31, 2019
Laurel/Maryland, USA

Title: **Calm Interfaces for Integrated C2 Systems**

Authors:

Names: **Hans-Christian Schmitz, Alessia Cornaggia-Urrigshardt, Fahrettin Gökgöz, Samantha Kent, Kevin Wilkinghoff**

Organization: Fraunhofer FKIE

Address: Fraunhoferstraße 20
53343 Wachtberg-Werthhoven, Germany

Phone: +49 228 9435 386

E-Mail: {hans-christian.schmitz | alessia.cornaggia-urrigshardt | fahrettin.goekgoez | samantha.kent | kevin.wilkinghoff}
@fkie.fraunhofer.de

Calm Interfaces for Integrated C2 Systems

H.-C. Schmitz, A. Cornaggia-Urrigshardt,
F. Gökgöz, S. Kent, K. Wilkinghoff

Abstract

The concept of calm technology, according to Marc Weiser and John Seely Brown [26] as well as, more recently, Amber Case [3], demands that technology must be robust and unobtrusive. It should require the least amount of human attention possible and make use of the periphery of attention. It should support natural human behaviour and be integrated into established working processes. Technology must enable users to accomplish their essential tasks, which go beyond system interaction, as easily as possible. We investigated Command & Control Information Systems (C2IS) that are integrated into battle tanks and provide the crew with a picture of the current operational situation. The systems enable data exchange within the platoon and the company. However, it seems as if integrated C2IS in use do not reach their full potential as relevant information is not always entered in time and, therefore, the operational picture is not always complete and up-to-date. One reason for this is that system interaction via the existing GUI, a touchscreen and a keyboard, requires more cognitive effort than can actually be provided, particularly in stressful situations. Following the principles of calm technology, we developed the concept of a distributed, multimodal interface, allowing for both input and output in diverse modalities and via various channels. This also includes other systems in use, such as the tank's periscope. Interaction via a calm interface does not absorb the attention of the operators but instead allows them to interact with the system simultaneously with other tasks. We implemented a prototype and evaluated it with military subject matter experts (SME). Our results indicate the prototype's usefulness and ease of use.

1 Introduction

In order to establish the effective usability of information systems, user interfaces are required with which users can operate the systems and control their functionalities with the lowest cognitive effort possible. Users must not be bound by the systems but they must always be able to perceive information from outside the systems and fulfil tasks aside from system interaction.

In conversations with military subject matter experts (SME), we were told that Command & Control Information Systems (C2IS) that are integrated into battle tanks are mostly used in pauses or phases of low intensity. The reason seems to be that the interaction with the systems consumes operators and often cannot be performed in parallel with other tasks that might be considered more urgent. As a consequence, new information is not always entered in time, and only information that is deemed absolutely necessary is entered into the systems at all. Thus, the operational pictures are not always complete and up-to-date, and the control of task completion within the platoon and the company is not supported in an optimal way. In order to bring integrated C2IS to their full potential, and ensure complete and up-to-date operational pictures, user interfaces are required that enable system usage also in phases of higher intensity. Better usability of integrated C2IS has to be ensured as a precondition of their effective usefulness.

These considerations, together with the intuition that speech is an adequate modality for a usable interface, led to the task of investigating opportunities and challenges of voice control for integrated C2IS and to design a respective concept. Soon it became apparent, that it is not sufficient only to consider speech as an input mode and to consider it only as an input mode. Firstly, the existing GUI should be preserved. Secondly, operators need appropriate system feedback. Therefore, feedback mechanisms have to be provided and adequate modalities for different types of feedback have to be found. Speech can be among these modalities but need not be the only one. Thirdly, relevance and criticality of information have to be assessed so that it can be specified which information has to be provided to whom in a given context. We developed a concept of a distributed, multimodal user interface for an integrated C2IS that fulfils these demands.

In this paper, we will reflect on the concept of calm, effectively usable interfaces and we will present and discuss our prototype for multimodal interaction with an integrated C2IS. The outline of the paper is as follows: in Section 2, we will elaborate on the purpose and usage of C2IS in battle tanks. Then, in Section 3, we will explain the principles of calm technology and briefly sketch state-of-the-art methods of Automatic Speech Recognition (ASR), in particular Small-Footprint Keyword-Spotting (KWS). In Section 4, we will describe the concept of a calm interface for an integrated C2IS, a prototypical implementation, and its evaluation with tank commanders. Finally, in Section 5, we will sum up and give an outlook on open issues and future work.

2 Integrated C2 Information Systems in Battle Tanks

An integrated C2IS provides a battle tank crew with a digital map and allows for better orientation and navigation. It gives a situational overview to the extent that the respective information is available, and it ensures a connection to leadership within the platoon and the company. The system must adhere to the requirements of usefulness and usability. Firstly, the operational picture needs to be correct and up-to-date and the system should work without any technical issues. Secondly, the operation of the system should be simple and only require a small cognitive load. The systems that we have so far had the opportunity to interact with, do not sufficiently fulfil these requirements. System update rates are too slow, meaning that the usefulness of having a situational overview is restricted. Furthermore, these systems can be rather complex and the amount of effort that is required to operate them is high. Their usability has therefore not been optimized.

A battle tank operates in adverse conditions, as there is noise pollution and constant movement and shaking, among other hardships. The crew, consisting of a commander, a gunner, a loader and a driver, can be under considerable pressure, especially in battle situations. Let us assume that the tank is equipped with an intercom system that uses active noise cancellation. The crew communicate with one another through this system. Communication with the platoon and the company takes place in two separate radio circuits, which are also used for data exchange. The battle tank is equipped with an integrated C2IS, which shows Battle Space Objects (BSO) represented by symbols on a map. The crew are continually contributing to the operational picture, because they provide new BSO reports and update the information for existing BSOs. The crew observe their surroundings through a periscope and determine the positions of recognized BSOs with a laser. For each BSO, they create a representation in form of a BSO report, which specifies the BSO's type (e.g., "battle tank"), hostility status (e.g., "friendly" or "hostile"), position, and further attributes such as movement and direction.

We have made the simplifying assumption that only one person can control the C2IS, which is either the commander or the gunner. We are aware that this may be controversial, as the commander and the gunner often work together, however, it allows us to simplify the procedure. Adjustments to include multiple operators can be made at a later stage (we discuss this point in more detail in Section 4).

Let us now turn to four paradigmatic use cases:

1. Creating a new BSO report – " Battle tank, hostile, moving west".

   The interaction with the current GUI is as follows: the operator looks through the periscope, recognizes a battle tank and determines its position with the laser. As a result, a dummy symbol representing the still underspecified BSO appears on the map. The operator turns her attention to the map and uses a touchscreen to select the dummy symbol. She then selects the BSO type (in this case: "hostile battle tank") from a predefined symbol menu. She picks up the keyboard and provides additional information ("moving west"). Once complete, she puts the keyboard back in its place, closes the BSO report and sends it to the relevant parties, again via touchscreen.

   During the interaction with the GUI, the operator has to interrupt the observation of the surroundings and turn her attention to the display. The obvious disadvantage is that the surroundings are no longer being observed. The advantage, however, is that she receives immediate system feedback.

   A Voice User Interface (VUI), or a multimodal interface using voice recognition, would change the way an operator interacts with the system as follows: the operator looks through the periscope, recognizes a battle tank and determines its position with the laser (as before). She then starts speech recognition, e.g. by using Push-to-talk (PTT), and states "Battle tank, hostile, moving west". The system recognizes and transcribes the input and provides the symbol for the BSO, in this case a battle tank, as feedback. The symbol is shown in the periscope so that the operator does not have to turn her attention to the display. Additionally, it is again placed in the correct position on the map. The operator can also provide speech input to forward the BSO report and to close the interaction (instead of using the touchscreen).

   It is crucial for an operator to receive system feedback to ensure that speech input has been correctly understood. Such feedback can be provided via various channels, among them devices that are already present in the tank, such as the periscope.

2. Modifying an existing BSO – “Battle tank, destroyed”.

   The interaction with the GUI approach is as follows: the operator looks at the map and selects a symbol representing an existing BSO (namely the battle tank that has just been destroyed). She changes the tank’s operational status to “destroyed”, using the keyboard and/or the touchscreen. She concludes her input and forwards the updated BSO representation.

   If the operator had the option of using a VUI, the interaction would be as follows: she looks at the map and selects an existing object using the touchscreen (as with the GUI) or she interacts via the VUI and names the BSO. She does so by either directly stating the BSO’s name, if it has been named at creation, or by using another type of referential expression. Examples are “last” for the previously created/modified BSO or “the battle tank moving west”, which should only be possible if the object in question is the only object identifiable by this definite description. The operator then updates/modifies the BSO using voice input and, as explained in the previous use case, the system will update the BSO and provide feedback. The operator can also use speech input to forward the BSO report and end the interaction.

3. Sounding a standard alarm, e.g. a mine alarm: an alarm will be given by radio and not through the C2IS. However, a VUI would also allow the alarm to be sounded using the C2IS. The advantage is that if the operator sounds the alarm, her voice could automatically be recorded as system input and a new BSO (“mine”/“mine field”) would be created. The map then shows the respective symbol directly in front of the tank’s own position on the map. Note that in the case of an alarm, the system will override the PTT function that is needed to use the voice input in the other use cases. The system will automatically recognize the alarm.

4. Using the VUI to control the display: currently, the only way to control the graphical interface is by using the touchscreen. It would be possible to add a voice command function for a more hands-free functionality. Examples of features are: “refresh” to refresh the current view, “zoom in” und “zoom out” to zoom in and zoom out, “center” to center the display for a specific location, saying the name of a map to select and view it, and saying the names of menu items to navigate the controls.

The four use cases illustrate that a VUI is not a replacement for an existing GUI but should rather be seen as an extension to supplement the current system and make it easier to navigate. A challenge is to provide the necessary feedback without overtaxing the user. To this end, criticality of information has to be determined. Note that a tank is a collaborative working environment with different roles and different information requirements. Both information and its modality of presentation are to be tailored to specific roles so that each crew member, be it the driver or the gunner, is supported optimally. (Cf. Section 4.3.)

3 Calm Technology and Automatic Speech Recognition

A multimodal interface as outlined in the previous section must be calm and unobtrusive. To achieve this, such an interface relies, to a large extent, on speech interaction. In the present section, we will describe two related concepts/ methodologies, namely Calm Technology and Automatic Speech Recognition (ASR).

3.1 Calm Technology

In Allen und Hussain's hyperwar scenario, the captain of a battle ship successfully war/ensure ds off an attack by commanding his ship via a highly sophisticated interface, which displays all possible types of information, including information about the ship's surroundings and any weapons:

> "The captain moved quickly from the bridge into the CIC [Combat Information Center] and, along with the others in the center, donned the augmented reality headgear and attendant gauntlets to assimilate and react to the totality and complexity of the battle he was about to lead. His first thought was the status of his weapons. He had only seconds as some elements of the swarm were supersonic, maybe hypersonic. Because of the elevated threat level, the captain had been given a high level of authority and autonomy to engage any potential attackers. He quickly cycled to the 'weapons status' views in his headset, and all were green, being continuously fed targeting information from the ship's fire-control complex now locked onto and tracking and analyzing the incoming attacking swarm. He had to act and shifted to the 'ASB [Anti Swarm Battery] status view.' With a sweep of his hand in virtual reality, he initiated the ASB." [1]

The extract shows that the user interface is seen as a crucial technical element that is needed to succeed in battle. However, Allen and Hussain's targeted description cannot be considered realistic without reservations: the augmented reality headgear – though it is only augmented reality and not full virtual reality – might fully consume the captain's attention so that he can no longer communicate and cooperate with the crew members except via the system. He takes over additional tasks that are usually fulfilled by others, such as the control of the weapons status, and thus, faces an additional workload. Finally, his actions are essentially interlinked with the system – it is unclear how he would act in case of a system breakdown.

The concept of an interface that completely opens the information space but simultaneously might fully consume the operator, seems to prevail. It is in stark contrast to a concept called Calm Technology, which was first developed by Mark Weiser and John Seely Brown in 1995 [26].[1] Recently, this concept is undergoing a renaissance. It can be considered to be a blueprint for the development of effective and user-friendly technology. The following principles, which are all relevant to the design of user interfaces, have been postulated ([3], p. 16ff):

[1] Calm Technology grew out of Weiser's idea of Ubiquitous Computing [25]. It has been further developed in the Disappearing Computer initiative [20].

1. "Technology should require the smallest possible amount of attention." – Systems should require as little attention as possible, so that users are capable of performing other tasks while still engaging with the system if that should be required. The system should not fully absorb all of a user's attention if this is not necessary. In order to do so, systems need to present information in such a way that the users receive only the relevant information at the time that it is actually required. Furthermore, information should be present in an appropriate way, where the user can easily process the information. This could be via a visual or acoustic signal but could also be achieved through other signals, e.g., haptic signals such as the vibration of a mobile phone.
2. "Technology should inform and create calm." – Even when a system is working perfectly, users should be informed that this is the case. The system should provide reliable and unobtrusive feedback to affirm that everything is working accordingly and create a relaxing and supportive working environment.
3. "Technology should make use of the periphery." – Information that does not require our full attention but still needs to be available, should be placed in the periphery of our attention. For example, driving a car has evolved into a multisensory experience, with road signs and traffic lights requiring our full attention but engine lights only turning on when relevant. The core idea is that the user can recall the relevant information when necessary.
4. "Technology should amplify the best of technology and the best of humanity." – Human users are very good at adapting themselves to technical systems to compensate for any potential flaws a system may have. However, a system should not force users to adapt to it, rather, systems should take natural human interactions and limitations into account.
5. "Technology can communicate, but doesn't need to speak." – The output of a system should not be too complex. Often, reduced forms of communication will suffice. Complex system outputs, such as voice, should be avoided if the only goal is to make the systems seem more human-like and natural, even if the system does not require that type of output.
6. "Technology should work even when it fails." – It should be possible for operators to complete their task even if technology partially or fully fails. If a system's voice recognition software stops working, it should still be possible to operate the system by using a different medium, such as a keyboard or mouse. In a battle tank, if the electronic map stops working, the crew must be able to navigate using a regular map and if data cannot be transferred using the C2IS, users should still be able to communicate through radio signal. The advantages of legacy systems should be preserved as far as possible.
7. "The right amount of technology is the minimum needed to solve the problem." – Ideally, technology should be self-explanatory and a part of everyday life and should not require any special attention. This can only be achieved if technology is stripped back and only contains the necessary features and all superfluous functions are minimized.
8. "Technology should respect social norms." – Technology can only fade into the background and become a part of everyday life if it adheres to existing norms. This includes both

> social and cultural norms as well as the norms that define the usual standard of practice. Openness and acceptance are more likely to occur when users feel that the technical solutions support and enhance current practices. As soon as this is achieved, technology can also contribute to changing current norms and standards.

An intelligent user interface should provide the right information at the right time to the right person. It should not provide any other information. Which information is “right” depends on the current context. Therefore, the system should be provided with a context representation. Elements of such representations can be static, e.g., if they are related to predefined roles of operators. Other elements may be dynamic. They can be related to the area of responsibility, existing tasks and processes as well as a person’s physical and mental state. Respective data are to be collected and distilled so that they can be accessed to determine the relevance of information.

An intelligent user interface must reliably recognize user input and aggregate input from different sources. This includes sources that can be taken for granted, such as keyboard, touchscreen and mouse, as well as others, like speech, hand written text, gestures, mimic, and more. The interface must also provide information in the right (combination of) modalities. To this end, it must select the right modality dependent on the given context representation and generate the respective output.

3.2 Automatic Speech Recognition and Keyword Spotting

Central elements of a multimodal user interface are an Automatic Speech Recognition (ASR) system and a dialogue manager that controls the interaction. In this section, we will present the fundamental methods underlying ASR.

ASR can be divided into two subareas. The first is large-vocabulary continuous speech recognition (LVCSR) and the second is small-footprint keyword spotting, in short: keyword spotting (KWS). LVCSR aims at completely recognizing and transcribing spoken language with the aid of complex systems, whereas KWS recognizes a certain number of predefined keywords with limited resources as accurately and quickly as possible. An example that easily explains the difference between both subareas is Amazon’s Alexa. Alexa is started locally by a single keyword and, thus, uses KWS for that purpose. Only after recognizing the keyword, a complex LVCSR system is used remotely for further speech recognition. This way, users can interact with their devices anytime without overloading external servers.

Both subareas, LVCSR and KWS, are on opposite ends of a complexity scale. Hence, LVCSR systems can also be utilized for limited vocabulary and, potentially, KWS systems are able to recognize whole phrases or small sequences of words. Designing an ASR system highly depends on the concrete application, underlying data, and user requirements, including hardware requirements.

To model human language and be able to completely transcribe speech, LVCSR systems require massive amounts of labeled data and computing capacity when being trained. Because of their high complexity, trained LVCSR systems are also relatively slow when being run with standard hardware, which results in high latency. Since interacting with a C2IS through speech needs to work as quickly as possible, and usually requires only a limited number of speech commands, a KWS system is much more suitable for our application than a complex LVCSR system.

The goal of KWS is to find keywords or keyphrases with low latency in a continuous audio stream. Usually, only limited computing resources are available in KWS applications, e.g., in smart phones. Thus, the number of keywords should also be relatively small. Although the whole audio stream needs to be searched for keywords, actual occurrences are relatively rare. In conclusion, most parts of the audio stream do not contain them and can be ignored. The fundamental problems of KWS are (i) detecting an alleged keyword in material that is to be ignored (false alarm) and (ii) ignoring material that does contain a keyword (false rejection). When designing a KWS system for a specific application, these two types of errors need to be balanced in a suitable way.

There are three core paradigms of KWS:

- The classical approach has been developed analogously to the traditional LVCSR systems that are based on Hidden Markov Models (HMMs) [14]. First, audio features such as Mel-frequency cepstral coefficients (MFCCs) [6] or features based on perceptual linear prediction (PLP) [10] are extracted. Based on these features, a model for each keyword is trained. Moreover, a so-called filler model is trained for non-keywords and segments without speech. All trained models are then combined into a single HMM, the keyword/filler Hidden Markov Model, allowing to search for a most likely sequence of keywords, non-keywords and segments without speech. Based on the application, this search can be computationally expensive, especially if the number of keywords to be detected is not very small [15, 16, 27].
- Another KWS approach is Query-by-Example (QbyE) KWS [11, 5, 24]. In contrast to keyword/filler HMMs, which need to be trained in a supervised manner, QbyE KWS is based on unsupervised learning. For QbyE KWS, the first step is to create a database with templates derived from audio samples of keywords provided by the user. When running the KWS system, encountered audio segments are compared to these templates in order to find the most likely keyword. The biggest advantage over the previously described keyword/filler HMM is that training another model for each keyword is not necessary. Thus, QbyE KWS takes less effort and resources. Furthermore, additional keywords can easily be added afterwards. This approach is also very flexible, however, the resulting error rates are usually higher than those obtained with systems based on supervised learning.
- The third way to detect keywords in audio data is based on Deep Neural Networks (DNNs) [4, 23]. These discriminative approaches result in posterior probabilities for all keywords to be found. As with keyword/filler HMMs, all keywords must be known in advance and a neural network must be trained with many spoken samples of these keywords. After being trained, neural networks require far less computational resources leading to lower latency. Moreover, they tend to significantly outperform keyword/filler HMMs in terms of error rate. Hence, using neural networks instead of keyword/filler HMMs is favourable in any way. Even better error rates than those of feed-forward DNNs can be obtained with

more sophisticated approaches such as Convolutional Neural Networks (CNNs) [17, 22] and Recurrent Neural Networks (RNNs) [21, 9].

When designing an ASR system for the application in battle tanks, certain challenges need to be taken into consideration. The first concerns environmental issues like noise that interfere with the speech signal. The second is intra-speaker variation: individual speakers articulate differently under different circumstances. Their voice and articulation depend on both environmental conditions and inner states. Even a personalized system has to cope with such variation. The third is inter-speaker variation: speakers differ in their regional, gender-specific and social codes. Inter-speaker variation affects the robustness of non-personalized systems. Finally, an ASR system needs to cope with slips of the tongue and diverse repairs.

4 A Calm Interface for an Integrated C2 Information System

Let us now discuss the concept for the multimodal control of an C2IS that is integrated into a battle tank. The concept first and foremost has to fulfil the use cases 1 to 3 named in Section 2, that is, the creation and update of BSO reports as well as alarm calls. We consider a solution to the fourth use case, the control of the GUI, as straightforward but of relatively minor importance.

4.1 Concept

The concept is based on the presumption that the battle tank is already equipped with a C2IS and that a GUI is available. The given interface is to be extended with (i) a dialogue manager, for processing input in different modalities, (ii) an ASR component and (iii) components for providing feedback.

Interaction with the C2IS is task-oriented, and the course of an interaction is clearly structured. Therefore, the dialogue manager can be designed with a relatively simple finite-state machine [13]. The dialogue manager receives input from the user interface components, connects to the local database, calls a symbol renderer to place symbols on the map, forwards information to other parties, and produces feedback. It implements a dialogue model, such as the one depicted in Figure 1.

The range in functionality is defined by the existing system and the attributes of the BSOs that need to be specified. BSO reports naturally provide frames for input: it must be possible to name different types of BSOs, such as “battle tank”, “infantry fighting vehicle”, and “truck”. Each BSO has a hostility status, namely “hostile”, “neutral” or “friendly”. Additional attributes, such as the direction of movement, can be specified. These attributes and their values thus automatically provide the input vocabulary. Since the vocabulary is limited, recognition can be accomplished using a keyword spotter. A more complex, and thus less robust, ASR component is not needed.

In our use cases, user input can be considered to be unambiguous. That is, ‘battle tank” is always a BSO type and never the value of another attribute, “friendly” is always a hostility status and never a direction of movement. Therefore, the keyword spotter can directly assign

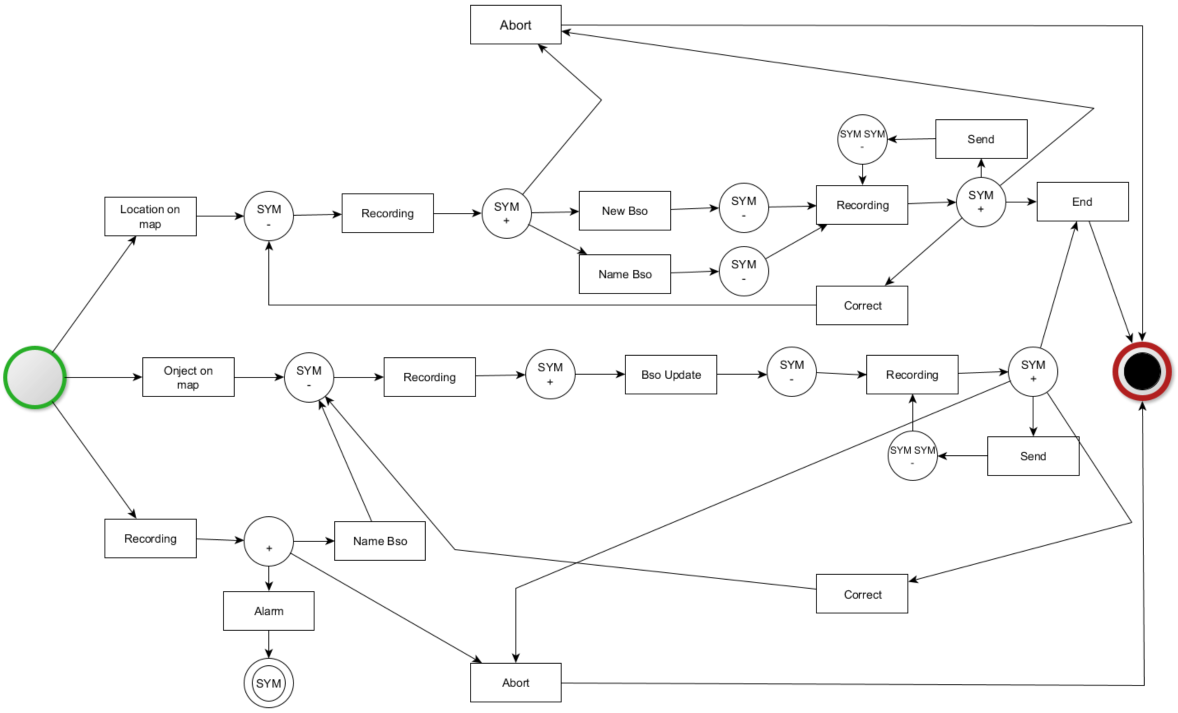

Figure 1: Dialogue model for an integrated C2IS

the spoken input to the corresponding input fields, and the dialogue manager need not provide an extra speech understanding component.

An important function is the possibility to name BSOs, so that they can be recalled at a later date. A respective slot for a name is provided by a BSO report frame. Since a keyword spotter must robustly recognize every name, only names which are predefined in the keyword spotter's vocabulary can be chosen. Therefore, by definition, names always consist of a base word, e.g. "Brady"[2] and a number: "Brady One", "Brady Two", etc. This particular naming convention is, of course, arbitrary, and one could also choose another base word or another mode of numbering ("Amber Alpha", ...). We assume that names are only used locally within a specific tank, and that they are not forwarded. Thereby, we assure that conflicts of identical names given by different operators for different BSOs cannot arise. It should be investigated, however, whether a global name space would be desirable, so that BSOs which have been recognized before an operation and which are, thus, already present on the C2IS map, can be named in advance. To enable both the global naming of existing BSOs and the local naming of newly recognized BSOs, we would have to distinguish between a global and a local namespace. Furthermore, since one cannot be certain that all names, in particular all global names, are always correctly remembered, it must be possible to ask for names.

System feedback can be both acoustic, via sound signals or text-to-speech (TTS) synthesis, and visual, via the GUI or other devices. Visual feedback was deemed most appropriate by the

[2] In NATO exercises, red forces come from Bradyland.

military SMEs we talked to, due to noise pollution and the general tendency to overload the acoustic channels. We therefore decided to provide visual feedback that consists only of the most critical information. In the case of a new BSO report, we display the BSO symbol which contains the information that must be entered into the system correctly. We accept that minor mistakes in specifying the BSO attributes may go unnoticed. However, we decided to refrain from providing too much information so that operators are not overloaded, which is one of the core concepts of calm technology.

4.2 Prototype

We prototypically implemented the concept described in the previous section for German use cases. (The examples in this section will be English, but in reality the prototype is in German.)

Our prototype is based on the InSAne (Intelligent Situational Awareness [2, 18]) C2IS demonstrator. InSAne realises a micro-service architecture which enables the rapid design, integration and evaluation of new services. It has been developed at Fraunhofer FKIE as an experimental test environment. It also serves as a reference implementation of the MIP4 Information Exchange Specification (MIP4-IES, [12]). The MIP4-IES defines structures of BSO reports. In our prototype, we transform user input to fit into a subset of these structures.

Our speech recognition system is based on KWS, more concretely a QbyE approach. This means that similarities between predefined keywords and speech input are computed and utilized. For each keyword, one or more samples are recorded and their specific features are stored in a database. Human factor cepstral coefficients – energy normalized statistics (HFCC-ENS) [24] are used for this purpose. (HFCC-ENS are MFCC-based features that are adapted to human auditory perception.) The templates resulting from this procedure are used for the online recognition of keywords. Hence, also speech input is transformed into a sequence of features. For each keyword, a similarity matrix between the stored features and the user input is computed. By using dynamic time warping, this similarity matrix is converted into a cost matrix, which can be used to find start and end points of potential matches, as well as a corresponding score. For our application, a slightly adapted version of the original algorithm is used, that runs faster but delivers less precise start times of potential matches. This is not a disadvantage, as the required output is just a textualisation of speech and accurate points in time are not needed for this purpose. Since all features of the database are based on speech data recorded under similar conditions (same speaker, similar noise conditions, same microphone), the magnitudes of the scores can be related to each other. A list of matches is created and the amount of matches per keyword can be limited accordingly. In our application, we used three matches for each keyword. If the user input is short, this value is reduced. When analysing the matches of a single user, there will be a large number of overlaps in time depending on the total length of the recording, the amount of keywords and the number of matches. Therefore, the keyword related to the highest score in a given time frame is returned as a match and other, simultaneously occurring matches are discarded. In addition, only parts of the signal containing speech are considered. To do so, every recording is inspected for pauses of speech so that false matches can be rejected. Lastly, the resulting sequence of keywords is transformed into a predefined XML structure for further processing by InSAne.

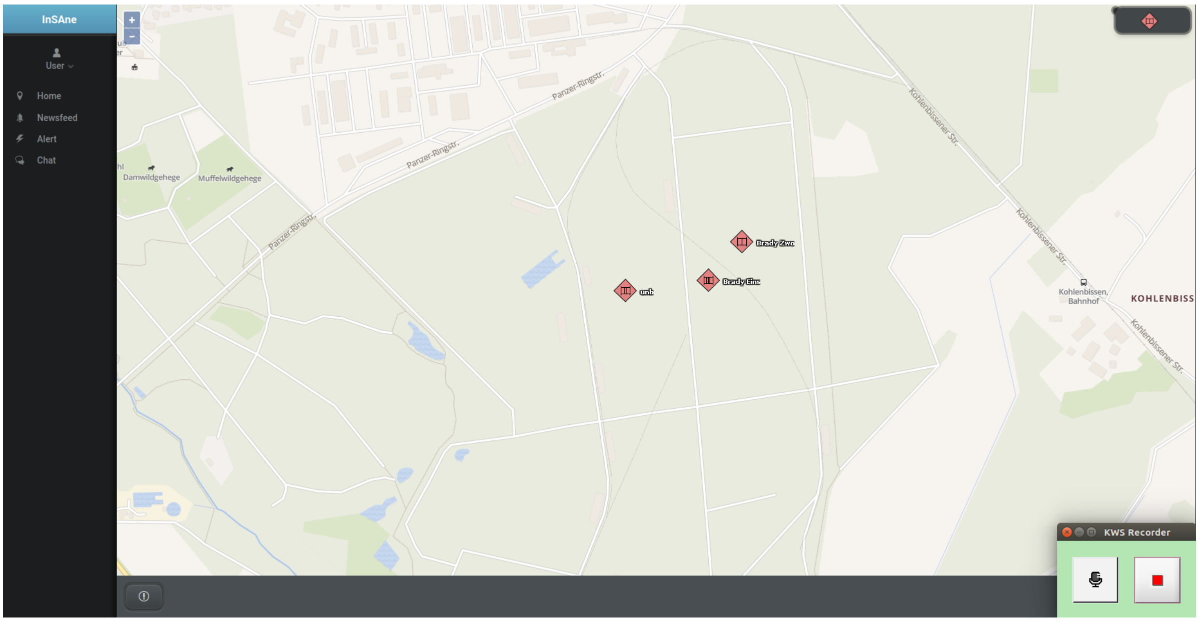

Figure 2: Screenshot: InSAne during the second example dialogue

The following example dialogues illustrate the functioning of the prototype. They also serve the explanation of the dialogue model depicted in Figure 1. To this end, markers in round or square brackets point to the respective nodes in the diagram.[3]

Let us start with the creation of a new BSO report:

- We select a location on the map with the mouse pointer: [Location on map]. In the demonstrator, the mouse replaces the laser for determining the position of a BSO. A dummy symbol appears in a window in the upper right corner of the map: (SYM -). This window simulates the display within the periscope.
- The keyword spotter is started. The operator says "Battle tank, hostile, moving west, name Brady Two" – [Recording] (SYM +) – and the speech input is recognized by the system: [New Bso] (SYM -).[4] To show that the input has been recognized, the dummy symbol in the periscope window is replaced by the symbol for a hostile battle tank, and the same symbol is placed on the map. Figure 2 shows a screenshot of the demonstrator after this dialogue step.[5] Three BSO symbols are placed on the map. Two BSOs have been named "Brady One" and "Brady Two", respectively. The leftmost BSO has not been named so far.
- The interaction is closed with "over": [Recording] (SYM +) [End]. The tank symbol disappears from the periscope window.

[3] Round brackets refer to circles, square brackets to boxes.

[4] The "+" next to "SYM" means that a recording, which is still to be processed, is available, while a "-" means that no such recording is available.

[5] The screenshot was made during a demonstration in German. Therefore, the tank is named "Brady Eins and "Brady Zwo", not "Brady One" and "Brady Two".

Now that the battle tank Brady Two is in the system, we can continuously update its attributes. Let us consider the case of a position update:

- Again, a location is selected on the map – as in the previous use case, the mouse serves as a simulator for the tank's laser system: [Location on map] (Sym -).
- The operator utters the name "Brady Two": [Recording] (SYM +). The name is recognized by the system, the respective BSO is selected and its symbol is displayed in the periscope window. Moreover, the respective symbol on the map is moved to the newly specified location: [Name Bso] (SYM -).
- The interaction is closed with "over": [Recording] (SYM +) [End]. The Brady Two symbol disappears from the periscope window.

Next, Brady Two will be destroyed:

- The operator starts the keyword spotter with the PTT mechanism and utters the name "Brady Two" to select the BSO: [Recording] (+) [Name Bso]. The system recognizes the BSO and displays its symbol in the periscope window: (SYM -).
- The operator continues the speech input: "destroyed." The C2IS updates the BSO report accordingly and changes the symbol both in the periscope window and on the map: [Recording] (SYM +) [Bso Update] (SYM -).
- The interaction is closed with "over": [Recording] (SYM +) [End]. The updated Brady Two symbol disappears from the periscope window.

All attributes, including names, can be updated at any time. That is, BSOs that have not been named at creation can be named later, or names can be changed for whatever reason. Moreover, false input can be corrected, either implicitly, by modifying an existing BSO report, or explicitly, by starting a correction procedure with utterance of "correct". If an interaction breaks down completely, the dialogue can always be aborted by saying "abort" so that the input given during this interaction will be discarded.

Finally, let us look at an alarm call:

- According to a standard procedure, the operator sounds a mine alarm: [Recording] (+). The alarm is recognized by the system, a respective BSO is created and forwarded via all radio circuits, and a symbol for the mine or mine field is positioned right in front of the tank's own position: [Alarm](SYM). The interaction is closed automatically.

The mine alarm is an example of a standard warning procedure that is rehearsed by soldiers and must be correctly understood by the system. Further warnings are, for example, artillery warnings, air raid warnings, and missile warnings. Such warnings, and other standardized requests, can be implemented in a similar way.

4.3 Evaluation

We demonstrated the prototype to military SMEs twice. The first group consisted of four tank commanders, the second of seven officers from different military branches. Both demonstrations were based on a script consisting of ten example dialogues. Below is the feedback we received.

The SMEs confirmed that the scenarios and use cases were plausible and should guide further development. However, not all details in the use cases are realistic. As mentioned before, roles have been simplified. Usually, reconnaissance is a collaborative process, with different crew members participating in the determination of positions and entering information into the system. Dialogue management has to be adapted for collaborative system interaction, so that, e.g., a gunner can start input and the commander can continue or correct, if necessary.

Moreover, we started the demonstration with an empty operational picture. In reality, this would not be the case. At the beginning of an operation, the operational picture should already contain all potential or recognized BSOs. Thus, the tank crew does not predominantly detect "new" objects but rather confirms or updates preexisting information. Therefore, mechanisms for linking information with BSOs that are already in the system – such as the naming/referencing mechanism provided by our prototype – should be further developed.

The possibility to update the position of a BSO by determining the new position with the laser and just uttering the BSO's name was considered very useful.

It would be desirable if as much information as possible was fed into the system without being explicitly entered by an operator. This could be achieved by including automatic blue force tracking and linking the determination of positions with a robust friend-or-foe identification system.

So far, we only confirm input on BSOs entered into the system. According to the SMEs, however, also the forwarding of information should be confirmed. Such a confirmation could be given by an extra symbol within the periscope, possibly based on the WhatsApp symbology, with a grey tick for "sent", two grey ticks for "received" and two blue ticks for "read". However, firstly, we must refrain from overloading the periscope. Secondly, we delete all symbols from the periscope when an interaction is closed, although it might well be that the forwarded information has neither been received nor read. We must therefore elaborate on how long information should be provided via the periscope.

Not all information is relevant for all crew members and not every modality is optimal for every role. Confirmations and further feedback should be tailored according to the roles of the different crew members, both regarding content and modality. Confirmations of the successful forwarding of information might be relevant for the commander but not for the gunner or the driver. The SMEs assumed that the gunner is served best with acoustic information while for the driver visual signals will be more appropriate. Role-specific needs are thus to be further examined.

In general, it was assumed that using a dialogue model can be trained but that it might still well be that in very stressful situations, operators are not able to fully adhere to the model. The model must leave enough leeway.

In addition to the qualitative evaluation, we also conducted a quantitative evaluation using a "Perceived Usefulness and Ease of Use" (PUEU) questionnaire. It is based on the Technology Acceptance Model (TAM) that claims that the perceived usefulness and perceived ease of use are the most significant factors for predicting user acceptance and effective usage [8]. Our questionnaire was derived from Davis' PUEU-questionnaire [7]. It was completed by our first group of SMEs, consisting of four tank commanders. We measured the degree of acceptance towards six statements regarding usefulness and usability. The statements were rated on a scale from 1 (“do not agree at all”) to 7 (“fully agree”). (Originally, the statements were presented in German.)

These are the results:

1. “The voice controls would be useful to me.”

   Average evaluation: 6,25 (3 * 7 + 1 * 5), clear agreement

2. “I could fulfil my tasks better with the voice control.”

   Average evaluation: 6,25 (3 * 7 + 1 * 5), clear agreement

3. “The usage of the voice control would be simple.”

   Average evaluation: 6,125 (3 * 7 + 1 * 4), clear agreement

4. “The usage of the voice control would be frustrating.”

   Average evaluation: 1,5 (3 * 1 + 1 * 3), no agreement

5. “The usage of the voice control would be cumbersome.”

   Average evaluation: 1,75 (2 * 1 + 1* 2 + 1 * 3), no agreement

6. “I would use the voice control.”

   Average evaluation: 6,25 (3 * 7 + 1 * 5), clear agreement

The results are very positive and provide support for the design concept, although the test subjects cannot be considered representative as their number is too small. For the evaluation of a more advanced prototype, we will have to involve more subject matter experts (SMEs).

5 Conclusions and Outlook

We developed the concept of a calm and distributed user interface for an integrated C2IS. We implemented a prototype making use of ASR and evaluated it with military SMEs. We were able to confirm the essential usefulness and usability of such an interface.

The following points need to be taken into consideration for further development:

- the extension of the dialogue model and the basic vocabulary for additional use cases,
- the adoption of the model for collaborative usage,
- the comparison of different ASR methods under realistic conditions, outside the laboratory, ideally “in the wild”,
- the extension of feedback mechanisms, together with experiments in the use of text-to-speech synthesis (TTS) and role-specific information tailoring,
- further development of the naming and referencing mechanism and the elicitation of requirements for local and global name spaces, and

- the analysis of whether a multimodal user interface, such as the one presented, can be applied in other contexts as well, e.g. in other types of vehicles, command posts, or for unmounted soldiers.

References

[1] Allen, J., Hussain, A. (2018): *On Hyperwar*, Fortuna's Corner, `https://fortunascorner.com/2017/07/10/on-hyper-war-by-gen-ret-john-allenusmc-amir-hussain/` (last visit on April 30, 2019).

[2] Bau, N., Endres, S., Gerz, M., Gökgöz, F. (2018): *A Cloud-Based Architecture for an Interoperable, Resilient, and Scalable C2 Information System*, IEEE Explore (ICMCIS 2018).

[3] Case, A. (2016): *Calm Technology. Principles and Patterns for Non-Intrusive Design*, O'Reilly: Sewastopol et al.

[4] Chen, G., Parada, C., and Heigold, G. (2014): *Small-footprint keyword spotting using deep neural networks*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4087–4091.

[5] Chen, G., Parada, C., and Sainath, T. N. (2015): *Query-by-example keyword spotting using long short-term memory networks*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5236–5240.

[6] Davis, S. B. and Mermelstein, P. (1990): *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*, Readings in speech recognition, 65–74.

[7] Davis, F. (1989): *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*, MIS Quartlery, 319–340.

[8] Davis, F., Bagozzi, P., Warshaw, P. (1989): *User acceptance of computer technology – a comparison of two theoretical models*, Management Science 35 (8), 982–1003.

[9] He, Y., Prabhavalkar, R., Rao, K., Li, W., Bakhtin, A., and McGraw, I. (2017): *Streaming small-footprint keyword spotting using sequence-to-sequence models*, Automatic Speech Recognition and Understanding Workshop (ASRU), 474–481.

[10] Hermansky, H. (1990): *Perceptual linear predictive (plp) analysis of speech*, The Journal of the Acoustical Society of America, 87(4), 1738–1752.

[11] Levin, K., Henry, K., Jansen, A., and Livescu, K. (2013): *Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings*, Automatic Speech Recognition and Understanding Workshop (ASRU), 410–415.

[12] Multilateral Interoperability Programme (MIP): *MIP Website*, `https://www.mip-interop.org` (last visit on April 30, 2019).

[13] Pearl, C. (2017): *Designing Voice User Interfaces: Principles of Conversational Experience*, O'Reilly: Sewastopol et al.

[14] Rabiner, L. R. (1989): *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE, 77(2), 257–286.

[15] Rohlicek, J. R., Russell, W., Roukos, S., and Gish, H. (1989): *Continuous hidden markov modeling for speaker-independent word spotting*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 627–630.

[16] Rose, R. C. and Paul, D. B. (1990): *A hidden markov model based keyword recognition system*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 129–132.

[17] Sainath, T. N. and Parada, C. (2015): *Convolutional neural networks for small-footprint keyword spotting*, Sixteenth Annual Conference of the International Speech Communication Association.

[18] Schmitz, H.-C., Bau, N., Endres, S., Gerz, M., Gökgöz, F., Käthner, S. Mück, D. (2018): *A Newsfeed for C2 Situational Awareness*, Proceedings of the 23rd International Command and Control Research and Technology Symposium (ICCRTS 2018).

[19] Schmitz, H.-C., Cornaggia-Urrigshardt, A., Gökgöz, F., Kent, S., Wilkinghoff, K. (2019): *KI im Einsatz*, FKIE-Bericht, Wachtberg.

[20] Streitz, N., Kameas, A., Mavrommati, I. (eds.) (2007): *The Disappearing Computer. Interaction Design, Smart Infrastructures and Applications for Smart Environments*, Springer: Heidelberg.

[21] Sun, M., Raju, A., Tucker, G., Panchapagesan, S., Fu, G., Mandal, A., Matsoukas, S., Strom, N., and Vitaladevuni, S. (2016): *Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting*, Spoken Language Technology Workshop (SLT), 474–480.

[22] Tang, R. and Lin, J. (2018): *Deep residual learning for small-footprint keyword spotting*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5484–5488.

[23] Tucker, G., Wu, M., Sun, M., Panchapagesan, S., Fu, G., and Vitaladevuni, S. (2016): *Model compression applied to small-footprint keyword spotting*, INTERSPEECH, 1878–1882.

[24] Von Zeddelmann, D., Kurth, F., and Müller, M. (2010): *Perceptual audio features for unsupervised key-phrase detection*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 257–260.

[25] Weiser, M. (1991): *The Computer for the 21st Century*, Scientific American, September, 94–104.

[26] Weiser, M., Brown, J. S. (1995): *Designing Calm Technology*, XEROX Prac, `https://calmtech.com/papers/designing-calm-technology.html` (last visit on April 30, 2019).

[27] Wilpon, J., Miller, L., and Modi, P. (1991): *Improvements and applications for key word recognition using hidden markov modeling techniques*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 309–312.