2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)

2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS) | 978-1-6654-1261-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACSOS52086.2021.00040

A Framework for Self-Explaining Systems in the Context of Intensive Care

Börge Kordts
Institute of Telematics
University of Lübeck
Lübeck, Germany
kordts@itm.uni-luebeck.de

Jan Patrick Kopetz
Institute for Multimedia and Interactive Systems
University of Lübeck
Lübeck, Germany
kopetz@imis.uni-luebeck.de

Andreas Schrader
Institute of Telematics
University of Lübeck
Lübeck, Germany
schrader@itm.uni-luebeck.de

***Abstract*—Ventilated intensive care patients represent a sizable group in the intensive care unit that requires special attention. Although intensive care units are staffed with more nurses per patient than regular wards, the situation is often precarious. A situation that has become more acute during the COVID-19 pandemic. Weaning from mechanical ventilation as well as the limited communication abilities pose substantial stress to the patients. The incapability to impart even basic needs may negatively impact the healing process and can lead to delirium and other complications. To support the communication and information of weaning patients as well as to foster patient autonomy, we are developing a smart environment that is tailored to the intensive care context. While the provision and connection of smart objects and applications for this purpose can be time-consuming, self-organization and self-explainability may present helpful tools to reduce the effort.**

In this paper, we present a framework for self-explaining and semi-automatically interconnected ensembles of smart objects and ambient applications (that are integrated into smart spaces) used to realize the assistive environment. Based on a description language for these components, ensembles can be dynamically connected and tailored to the needs and abilities of the patients. Our framework has been developed and evaluated iteratively and has been tested successfully in our laboratory.

***Index Terms*— smart object guidance, self-explainability, mechanical ventilation, human-centered design, augmentative and alternative communication**

I. INTRODUCTION

Mechanical ventilation forms a cornerstone of modern intensive care therapy. Due to the severity of diseases, the accelerated use of technology and comprehensive interventions, intensive care is characterized by a high degree of complexity and forms a highly specialized setting for the treatment of the most seriously and critically ill patients. While the intensive care unit (ICU) setting has a higher nurse-patient ratio than general wards, staffing is still precarious: the recommended 1:2 ratio is often not feasible in Germany, particularly during night shifts [1]. This situation has recently become even more acute due to the COVID-19 pandemic, but has also become more prominent in public awareness.

A critical phase is the weaning from mechanical ventilation, where the body has to re-adapt to breathing on its own. This process may last from minutes to days or even weeks, a duration that is hard to predict.

Weaning can be a lengthy and stressful process that puts a lot of strain on patients, both physically and psychologically. Additional stress can be caused by the limited ability to communicate. This limitation can also lead to patients being treated worse because they are unable to impart their needs adequately [2]. An empirical study indicates that the needs and symptoms of ventilated patients are inadequately recognized by the staff [3]. If these needs and symptoms are not recognized in a timely manner or are insufficiently recognized, the risk of complications such as delirium may be increased and the weaning process may be more difficult and prolonged. On the other hand, positive communicative activities with ventilated patients have been linked to improved patient-related outcomes [4], which emphasizes the need for *augmentative and alternative communication* (AAC).

In AAC, a distinction can be made between high-tech and low-tech solutions. While low-tech devices are considered inappropriate since they do not take the holistic impact of impairments of patients in intensive care units into account [5], high-tech approaches may provide a complex and potentially expensive solution tailored to very specific needs. Without training concepts and taking the staff's processes into account, this may imply a steep learning curve, both for the healthcare provider's staff as well as the patients.

The situation in the intensive care unit is contrasted by the rapid evolvement of modern information and communication technology. Particularly, smart spaces have evolved from promising research projects to products assisting users in their everyday lives. Integrating smart objects in patient rooms may provide a convenient way to increase patient autonomy but may also assist caregiver's tasks. Decoupling and dynamically composing input and output of smart devices further allows for a needs-based provision of interaction techniques tailored to the abilities and preferences of the patients. Due to the heterogeneous user group of intensive care patients and their specific impairments and abilities, a single input device likely cannot address every situation, so that decoupling interaction from device and application logic and an individual provision of device ensembles comes to the fore.

In this paper, we present a framework that forms the basis of an ambient interaction space that is developed within the

DOI 10.1109/ACSOS52086.2021.00040

scope of the research project ACTIVATE[1] and that supports communication and fosters re-orientation of intensive care patients. By providing basic information about time, location, etc., we intend to reduce the disorientation caused by the medical condition. Additionally, the framework offers the possibility to control ambient devices (such as smart lighting) as soon as the patient's situation allows it.

The central component of the ACTIVATE system is the patient application, an application that provides a user interface for the intensive care patients informing them about the place and time, the nursing staff and their situation. The application further allows patients to impart their needs and currently provides a menu for the control of smart lights (as an example). In doing so, we hope to overcome the information deficit and to improve the patients' communication skills. Thereby, we aim to reduce the patients' stress caused by the overall situation (illness, medication, weaning, etc.), to achieve a better treatment, to make nurses more confident, and reduce their load.

In principle, our framework supports the integration of any (wireless) interaction device, controllable smart object, and/or ambient application. Note, that we refer to applications that are used within smart environments as ambient applications and that we refer to groups of connected devices and applications as ensembles. Ensembles can be compiled by the staff based on the patient's medical condition and their preferences. While providing and connecting these ensembles can be complex and time-consuming, clinics do not employ technical experts that could undertake this task.

We consider a semi-automatic interconnection of the components a useful tool to reduce the effort and integrated this approach in our system. In doing so, the staff only needs to select devices and applications that shall be connected to form an ensemble and the connections are self-organized by the system following the plug-and-play paradigm.

Since we were not able to identify any appropriate device for the use in an intensive care bed, we developed and integrated BIRDY (Ball-shaped Interactive Rehabilitation Device), a novel device for in-bed interaction and the special constraints of typical weaning patients (cf. [6]). We further identified possible device candidates that could potentially be used by intensive care patients with more comprehensive cognitive and physical skills in later phases when providing minor hardware changes or when used as disposable devices. Most promising candidates were the Litho controller[2], the Kai controller[3], and the Leap Motion[4]. These devices were also integrated in the ACTIVATE system using our framework.

While dynamically connecting smart objects and ambient applications allows for solutions tailored to the needs and respective tasks, it obfuscates the interaction and requires patients to learn its control patterns. An automatic generation and provision of instructions may be used to guide users throughout the usage of the interactive space provided by our framework. This self-explainability is achieved by merging and processing structured self-descriptions of respective components to generate manuals and tutorials on-the-fly, directly describing input actions and the respective expected effect. Since existing description languages of smart objects are insufficient to address self-explainability of dynamically interconnected ensembles, we previously designed the Smart Object Description Language (SODL) to be able to describe capabilities of smart objects and ambient applications as well as interaction possibilities [7].

Our contribution presented in this paper consists of (1) an extension of our previous work on self-explainability in smart environments to be applicable in the intensive care context, (2) the provision of self-descriptions using SODL as well as (3) the integration of all relevant components (interaction devices, patient application, and output devices). We use the framework to address the following research questions: Can self-explainability improve the descriptiveness of dynamically connected ensembles when used in an intensive care context? Does the semi-automatic connection present a useful process and can it reduce the effort required to provide ensembles to patients?

II. RELATED WORK

The extensive insights and findings of various other approaches including low- and high-tech solutions to overcome the communication barrier in intensive care have been summarized in the literature [5].

Low-tech approaches are considered inappropriate for the ICU due to the lack of consideration for the holistic impact of intensive care patients' impairments [5]. More advanced solutions may be adapted to the situation and context and may, for instance, cover specialized content and speech synthesis. Typical approaches are adapted from digital AAC communication boards (*talkers*) and often rely on either touch input or gaze control. Yet, these devices are not specifically designed for use in the intensive care context and thus are not focusing on in-bed usage, meeting sanitary regulations, coping with typical limitations and impairments of intensive care patients, or addressing the weaning phase in particular. Notably, a prototype of a communication system for intensive care patients including a specialized device to control the system has been developed [8]. However, the research is still in an early-stage and only few information has been presented up to now.

Regarding ubiquitous environments, an organic computing middleware that introduces self-* features has been proposed to cope with increasingly complex computer systems and setups [9]. The middleware is based on a Monitor-Analyse-Plan-Execute (MAPE) cycle to provide self-configuration, self-optimization, self-healing, and self-protection features. While the presented architecture is similar to our approach in that it uses a feedback cycle to react to changing situations, the middleware is not focused on self-explainability or the provision of usage instructions.

---

[1] https://projekt-activate.de/en/

[2] https://www.litho.cc/

[3] https://kai.vicara.co/

[4] https://www.ultraleap.com/product/leap-motion-controller/

TABLE I
REQUIREMENTS FOR THE FRAMEWORK REALIZING THE AMBIENT INTERACTION SPACE

ID	Requirement	Note
NFR-1	Interactions and their effect must be immediately clear	Weaning patients often have only short responsive phases
NFR-2	Patients should be instructed by the system whenever feasible	Reduce the effort, mismatch responsiveness and staff presence
...	...	...
NFR-4	Interaction alternatives should be provided and individually selected	Special vulnerable group of users with limitations and disabilities
NFR-5	Tutorials that guide the users should be implemented	Suitability for learning, mismatch responsiveness and staff presence
...	...	...
NFR-7	System components and ambient devices should be self-descriptive	Particularly relevant when an interaction option becomes unusable
NFR-8	The system must be error-tolerant	Reduce the effort, ensure continuous operation
NFR-9	The system should function as automatically as possible	Reduce the effort, ensure continuous operation
NFR-10	Components should be easily replaceable	Reduce the effort, ensure continuous operation
...	...	...
FR-8	Unintended input should be filtered if possible	When unconscious or unaware of the system's features, particularly during early weaning
...	...	...
FR-14/16	The system architecture should allow the integration of potentially usable interaction & output devices as well as smart room components	Control the system, manage a feelgood atmosphere
...	...	...

Self-explainability capabilities have been discussed as a requirement for intelligent environments with a focus on intelligent assistance processes [10]. Arguably, explanations or intrinsic behavioural intuitiveness of assistance processes can improve their acceptance, particularly for intelligent systems that perform barely foreseeable or unexpected actions.

Recently, a conceptual framework based on different layers of explanations has been proposed to provide self-explainability capabilities [11]. To explain actions and their cause, cause-effect relationship models are transitively combined to cause-effect chains. Based on these concepts, the authors further present a case study where explanations are added to a robot controller.

While this approach is focused on describing the general behaviour of digital systems online and not only addresses the users but also other persons (indirectly) involved, interactions and further steps required to control the respective systems play only a subordinate role. In contrast, our approach is more focused on explaining the interaction perspective, which is particular relevant when decoupling input and output components. Consequently, explaining physical interactions and the corresponding system behaviour becomes more relevant. Furthermore, the authors do not provide a concrete description format for such actions.

III. METHOD – HUMAN-CENTERED DESIGN

A successful deployment of the ACTIVATE system depends largely on the acceptance of the stakeholders, especially patients and nursing staff. Acceptance depends, among other things, on whether the expected benefit exceeds the additional effort required when using the system and whether the needs and peculiarities of the users are adequately taken into account. Consequently, we regard the consideration of the needs of potential users already during development as well as a focus on usability as crucial factors. Hence, our development follows the Human-Centered Design process (HCD) as specified in DIN EN ISO 9241-210.

In ACTIVATE, we reviewed the literature and conducted qualitative user research (particularly, interviews and focus groups with the target user groups) to generate data-driven user personas and corresponding problem scenarios specifying the context of use [12]. We discussed and refined these results and examined their implications in further workshops. Based on these insights, we finally specified requirements for the system. Following the principles of the HCD process we designed the overall system (including the framework) iteratively and constantly discussed and evaluated our designs with interdisciplinary experts, including nursing scientists and practitioners.

IV. REQUIREMENTS AND DESIGN CHOICES

In our previous work, we identified key criteria, barriers and enablers and requirements for our system supporting patient communication in intensive care [13]. Here, we focus on the non-functional and functional requirements that are particularly relevant for the framework realizing the ambient interaction space that supports patient communication, information and early autonomous control of ambient devices (see Table I).

Note that limited (time) resources of the staff, as well as the reduced cognitive skills and short consciousness/responsiveness phases of intensive care patients, are key aspects that particularly impact acceptance and usability of the system. Towards this end, self-* characteristics may present useful features. In this paper, we focus on adaptivity and self-explainability to address the requirements linked to descriptiveness and a dynamic integration of the respective components. While self-organization plays a subordinate role for our use case focusing on the patient application, it may greatly reduce the effort for the staff in more complex scenarios. Regarding the degree of autonomy, we argue that an intermediate autonomy realized by a semi-automatic connection of respective components is preferable over a full autonomy due to the domain-specific restrictions (cf. Section VI).

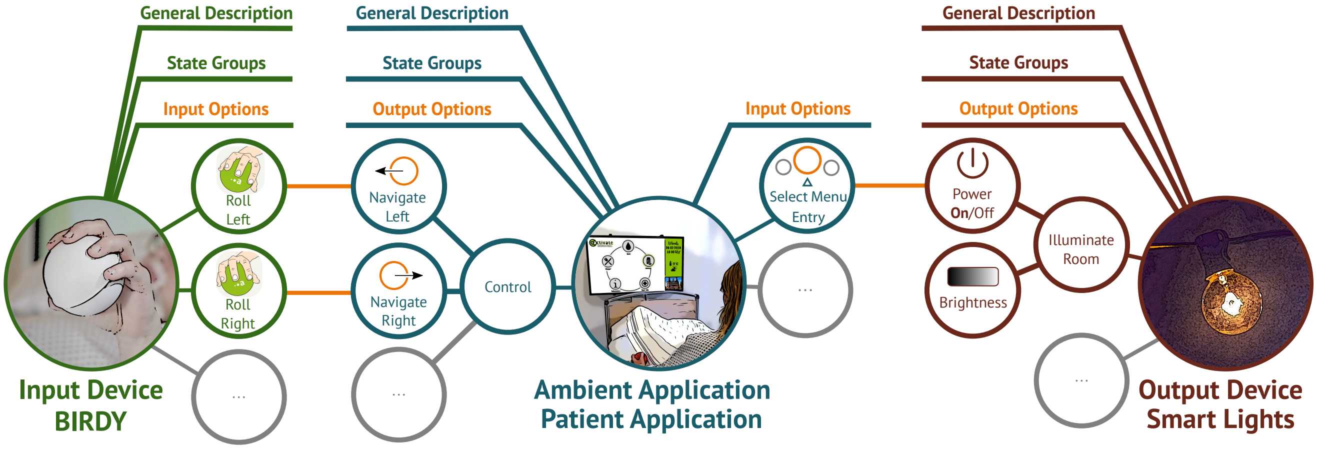


Fig. 1. Descriptions for an example ensemble using BIRDY, the patient application, and smart lights, along with suitable connections that define the controls. Descriptions cover a general section, state groups, as well as interaction primitives and output options. Note that various other interaction devices and smart room components can be used, too.

V. A DESCRIPTION LANGUAGE FOR (VIRTUAL) SMART OBJECTS

We previously developed a human and machine-readable description language for smart objects and ambient applications, called the Smart Object Description Language (SODL), that is used by our framework to form connections and generate instructions. Refer to [7] and [14] for more details on the language, as well as the framework in general.

Note that we focus on the interaction. Thus, SODL covers only the control of the applications (e.g., navigation) and possible actions used to control further devices (e.g., pressing a button in the application to switch smart lights). As a result, we describe only the interfaces of ambient applications in terms of input, output, and basic system behavior, like selecting menu elements.

All components currently used within the ACTIVATE system and that participate in an ensemble via our framework are described using SODL. This includes BIRDY, the Leap Motion, the Litho controller, the Kai controller, the patient application, and smart lights. A complete version of the descriptions is available online[5]. The self-descriptions of these components are explained in the remainder of this section.

The patient application is an ambient application and is described as a virtual device that is categorized as an app. It consists of an input and an output component. On one hand, the menu component's actions cover the navigation to the left and the right as well as a selection of the current element. A corresponding state group is defined with states for these three actions. The states are controlled using the connected interaction device (e.g., BIRDY). On the other hand, the patient application can provide menus for controlling hardware in the environment, currently supporting different types of smart lights. The menu entries are described as virtual input in the self-description. For this purpose, the description contains state groups for each virtual input.

The smart lights are described as a stationary smart object that provides state groups for the power state, discrete colors, and brightness levels, but also continuous ranges for brightness, hue, and saturation. Several goals (e.g., *change the saturation*), required tasks, state transitions, and concrete states are described in detail.

BIRDY is described as a portable input device that currently provides three gestures but can be extended to cover more gestures when realized. The self-description contains state groups for these gestures, the gesture's name, and the physical actions required to perform the gestures (e.g., to trigger the *roll left gesture*, rotate the device to left by 60 degrees, and afterward, rotate it back to the original position).

The other interaction devices are also described as portable input devices that cover the respective gestures of the devices. Their self-descriptions are structured correspondingly.

Figure 1 illustrates self-descriptions and indicates connections for an ensemble consisting of BIRDY, the patient application, and a smart light. Each description consists of three sections, a general section, state groups reflecting the behavior, and a task analysis that is used to describe the interaction in greater detail.

VI. A FRAMEWORK FOR SELF-EXPLAINING ASSISTIVE ENVIRONMENTS

Based on the self-descriptions written using SODL and reflecting the described requirements, we extended our previous work on a framework for self-explaining ambient applications and devices, called Ambient Reflection [15]. While the framework is capable of addressing many of the requirements, some modifications and extensions are needed to meet the given requirements.

Ambient Reflection is able to collect the self-descriptions of all components in the same network segment, to dynamically

---

[5] http://www.ambient.uni-luebeck.de/projects/activate/sodl/sodl.html

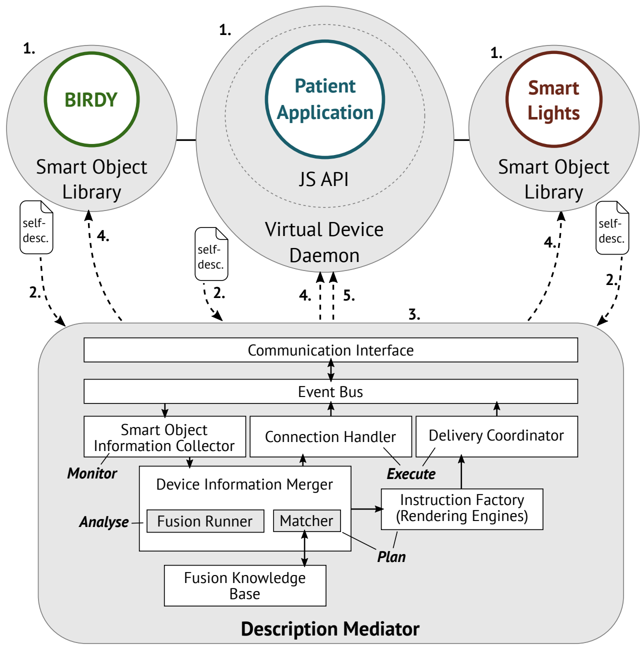


Fig. 2. System overview of the framework for exemplary connections between BIRDY, the patient application and the smart lights. After a discovery (1), the description mediator collects self-descriptions (2) and performs the matching as well as the rendering of instructions (3). Finally, the connection information is sent to the components (4) and the rendered instructions are delivered to components capable of presenting it (in this case the patient application) (5).

connect these components, and to ultimately provide merged overall descriptions for these dynamically connected ensembles. It is able to automatically create, restore and sever device connections as well as to replace (parts of) ensembles to keep the users able to solve their tasks.

While the underlying concept of Ambient Reflection is protocol-agnostic, the Java-based reference implementation makes use of either of two protocols: Devices Profile for Web Services (DPWS) and Universal Plug and Play (UPnP).

The framework consists basically of two components. First, devices and applications that are directly integrated using the Smart Object Library or use an API to bridge non-compliant or legacy objects or applications into the framework by connecting to the Virtual Device Daemon (VDD). Second, the description mediator used to mediate the devices and applications into ensembles based on their reflected functionality as well as to generate manuals and to inform the respective components about both. Figure 2 illustrates these components which are further described in the remainder of this section.

A. *Description Mediator*

The description mediator is the central component of the framework and is used for the mediation of smart objects and ambient applications. It handles the discovery, collects self-descriptions (based on SODL) of the respective components, identifies possible connections, and finally informs the devices and applications about the connections. After being informed, they are then capable of communicating directly with each other. The description mediator further generates and disseminates instructions to devices capable of rendering them.

Generally, the mediation process can be run automatically considering different extensible rules and quality criteria [16]. It is based on the self-descriptions of the respective input and output devices and identification of the interaction primitives as well as the output operations. The mediator searches for tuples of interaction primitives and output operations and matches them by using a goodness function that takes the rules and criteria into account. Due to the computational complexity, a probabilistic method is typically used for this process.

The description mediator also features a REST interface that can be used to configure certain options and to request information about collected devices, their descriptions, the mediated connections, etc.

Patients must not be able to control the system of other patients. In other words: components that are part of an ensemble must not be connected to another ensemble. Thus, certain restrictions are required when mediating the components. Hence, we extended the description mediator, in particular the matching process, to respect predefined assignments of interaction primitives to output operations and predefined ensembles that are allowed to connect for this use case. We extended the REST interface to configure the description mediator and further introduced a database to store these configurations persistently.

Consequently, the mediation can be controlled by providing a configuration (containing either a tuple of an interaction primitive and an output operation or pre-selected devices) allowing for a semi-automatic connection of device and application ensembles, for instance, by scanning QR codes and sending the configuration to the description mediator. We prototypically realized this option, where nurses use a smartphone to scan QR codes of the devices involved in an ensemble. An additional system component is informed about the scanned components and identifies known devices and connects them in a preset way (e.g., the *Roll Left Gesture* of BIRDY is always connected to the *Navigate Left* operation in the patient application). Devices that are not further known or configured are also taken into account and are automatically connected by the mediator respecting the given ensemble. This means, that an interaction device of a patient will still only be connected to the patient application of the same patient but a mapping of the input techniques to the control of the patient application is open to the mediator and based on the present rules.

B. *Smart Object Library and Virtual Device API*

To integrate smart objects into the framework, the Smart Object Library is used. The programming interface provides a factory pattern to create, initialize, and run smart objects in terms of Ambient Reflection and is also realized using Java. It is responsible for network communication and can process invocations for interactions or for non-interaction-related messages (miscellaneous messages) using an event bus. Besides, the library also handles the exchange of messages relating to the connection configurations, and the self-descriptions.

We used the Smart Object Library to integrate BIRDY as well as the Leap Motion into Ambient Reflection (further devices are integrated using the Virtual Device API, see below). Since most of the code for BIRDY is written in C/C++, we relied on SWIG[6] to provide a Java interface that in turn is used to provide the smart object to the framework.

To make sure miscellaneous messages are processed correctly, developers can provide a schema for their messages which can be used to validate incoming messages and finally parse the content. We defined message schemas for the messages exchanged between our input devices and the patient application. This includes battery states, calibration information, lock/unlock states, activities, configurations, and the connection state to the devices (e.g., check whether BIRDY is still paired and alive).

Notably, the patient application is implemented using JavaScript and particularly, Vue.js[7]. It is served in a web browser and thus, cannot easily connect directly to Ambient Reflection. Instead, we developed a bridge between these two worlds by introducing a component, called the Virtual Device Daemon (VDD), that runs in the background and connects to the framework. To communicate with the web application, the daemon routes messages through web sockets.

For a simple integration, we developed a set of different libraries (currently supporting JavaScript, Python and C#) that handle the connection to the daemon. While the patient application makes use of the JavaScript API, we relied on the libraries for Python, and C# to integrate some of the alternative interaction devices, including the Kai controller and the Litho controller. Notably, the API is easily extensible and other programming languages can be supported effortlessly.

VII. PROOF OF CONCEPT AND EVALUATION

Our framework forms the basis of the overall system realized within the research project ACTIVATE. The framework, its libraries, and APIs have been used and self-descriptions for the devices have been created by the team to integrate and describe the mentioned interaction devices, the smart lights, and the patient application.

The system was iteratively developed and formatively evaluated. We extensively and repeatedly discussed design choices with our team as well as other experts and peers and adapted the system accordingly and finally agreed on a consensus of its realization. In total, more than 20 experts in the fields of nursing and computer science, hard- and software engineering, as well as human-computer interaction and psychology were involved during the human-centered design process.

Finally, we conducted a semi-structured focus group interview with four participants (two ICU nursing practitioners and two HCI experts). During the interview, they created and released ensembles using the semi-automatic connection process based on predefined evaluation tasks. The latency and ease of use were positively mentioned by the participants. They also stated that the complex issue of connecting ensembles is appropriately simplified by our process. The concept of self-explainability and generated tutorials was considered useful. While interactive tutorials were favored over static instructions, combining tutorials and oral instructions by the staff were preferred overall. The participants noted that tutorials should be designed to carefully and gradually increase task difficulty. Several minor improvements were proposed (regarding wording, icon design, and color coding for QR codes as well as warning modals) and will be integrated in the future.

The dynamic interconnection realized by our framework allows for an integration of any (wireless) interaction device, controllable smart object, and/or ambient application. Hence, NFR-4, as well as FR-14 and FR-16 are met by our system. The self-explainability provided by the self-descriptions of respective components and the generation of instructions (and tutorials in the future) directly address the requirements NFR-1, NFR-2, NFR-5, and NFR-7. However, further work is required to integrate instructions directly in the patient application. The ability of Ambient Reflection to re-create device connections, to cope with lost devices and to re-organize the ensembles, as well as the persistent storage of configurations support NFR-8. The semi-automatic and dynamic interconnection addresses NFR-9 and NFR-10. FR-8 depends on the interaction technique and device but can further be improved by the interplay of several components. For instance, a lock screen and an unlock gesture can be used to reduce unintended input. We address this requirement by providing a message channel to exchange miscellaneous information between the components. In summary, our system addresses the described requirements and respects the nursing processes discussed with the domain experts.

The system is capable of dynamically connecting the proposed components and generates manuals that can be used to instruct users about the usage. As a proof-of-concept, we used an HTML rendering engine [14] to generate these manuals and render them in a web browser (see Figure 3). While these instructions already provide a benefit for users when (first) confronted with the system, the self-descriptions of the overall system could be used in interactive tutorials directly integrated in the patient application. In doing so, the tutorials can respect the potentially limited cognitive and physical skills of weaning patients and may ensure high usability.

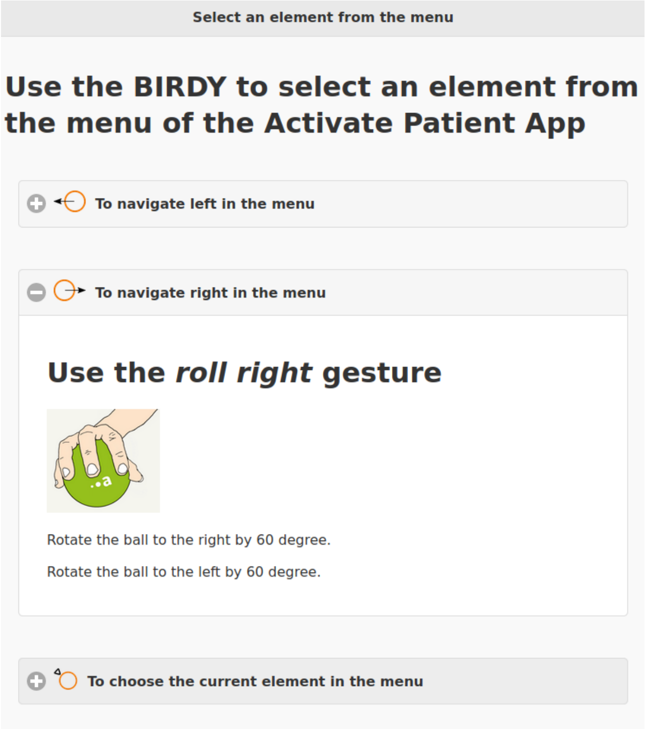
VIII. CONCLUSION

In this paper, we introduced a framework for dynamically connected, self-explaining systems that forms the basis of the research project ACTIVATE. Based on previous work on a description language for smart objects and a framework for self-explaining device ensembles, we developed a system that is applicable in the intensive care domain. The framework is used to realize a self-explaining smart interaction space consisting of smart objects as well as an ambient application,

---

[6] http://swig.org

[7] https://vuejs.org



Select an element from the menu

Use the BIRDY to select an element from the menu of the Activate Patient App

To navigate left in the menu

To navigate right in the menu

Use the *roll right* gesture

Rotate the ball to the right by 60 degree.

Rotate the ball to the left by 60 degree.

To choose the current element in the menu

Fig. 3. A screenshot of instructions generated by the framework that describe how to control the patient application using BIRDY. The instructions are rendered inside a web browser.

namely the patient application used to present a graphical user interface to the patients. We also integrated a specialized interaction device that is designed for in-bed usage in the intensive care context, namely BIRDY, in our system.

Our system is capable of organizing ensembles of smart objects and applications by dynamically connecting them to control each other. A semi-automatic connection process allows for an easy provision of device and application ensembles tailored to the needs, abilities, and preferences of the critically ill intensive care patients. It further allows for the generation of usage instruction for these ensembles, which include the interaction techniques, controls of applications, and output devices. While these insights indicate positive outcomes regarding the research questions, further work is required to ultimately assess the impact of our framework and the proposed processes.

The next steps are implementing and testing interactive tutorials in the patient application. They are supposed to include graphical representations to animate gestures for BIRDY, illustrate interaction techniques of the other identified input devices, and depict the menu navigation (system response).

We are currently preparing the evaluation of the framework, the overall ACTIVATE system, and particularly the generated instructions in a clinical setting and also ultimately in practice in a comprehensive field study. In doing so, we plan to evaluate the effects of supported communication on weaning patients and the nursing staff.

REFERENCES

[1] Michael Isfort, "Personalsituation in der Intensivpflege - Permanent 'am Limit'," *BibliomedPflege*, no. 04-2017, pp. 22–27, Nov. 2017.

[2] C. Handberg and A. K. Voss, "Implementing augmentative and alternative communication in critical care settings: Perspectives of healthcare professionals," *Journal of Clinical Nursing*, vol. 27, no. 1-2, pp. 102–114, Jan. 2018.

[3] A. W. Schindler, N. Schindler, F. Enz, A. Lueck, T. Olderog, and D. A. Vagts, "ICU personnel have inaccurate perceptions of their patients' experiences," *Acta Anaesthesiologica Scandinavica*, vol. 57, no. 8, pp. 1032–1040, Sep. 2013.

[4] M. Nilsen, S. M. Sereika, L. A. Hoffman, A. Barnato, H. Donovan, and M. B. Happ, "Nurse and Patient Interaction Behaviors Effects on Nursing Care Quality for Mechanically Ventilated, Older Adults in the ICU," *Research in gerontological nursing*, vol. 7, no. 3, pp. 113–125, May 2014.

[5] H. Carruthers, F. Astin, and W. Munro, "Which alternative communication methods are effective for voiceless patients in Intensive Care Units? A systematic review," *Intensive & Critical Care Nursing*, vol. 42, pp. 88–96, Oct. 2017.

[6] B. Kordts, J. P. Kopetz, A. Henkel, A. Schrader, and N. Jochems, "Requirements and Interaction Patterns for a Novel Interaction Device for Patients in Intensive Care," *i-com*, vol. 18, no. 1, pp. 67–78, 2019.

[7] D. Burmeister, F. Burmann, and A. Schrader, "The smart object description language: Modeling interaction capabilities for self-reflection," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 503–508.

[8] M. Goldberg, L. Hochberg, D. Carpenter, J. Isenberger, S. Heard, and J. Walz, "Testing a Novel Manual Communication System for Mechanically Ventilated ICU Patients," *UMass Center for Clinical and Translational Science Research Retreat*, May 2017.

[9] M. Roth, J. Schmitt, R. Kiefhaber, F. Kluge, and T. Ungerer, "Organic Computing Middleware for Ubiquitous Environments," *Organic Computing — A Paradigm Shift for Complex Systems*, pp. 339–351, 2011.

[10] S. Autexier and R. Drechsler, "Towards self-explaining intelligent environments," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, 2018, pp. 1–6.

[11] G. Fey and R. Drechsler, "Self-explaining Digital Systems: Technical View, Implementation Aspects, and Completeness," in *Advanced Boolean Techniques: Selected Papers from the 13th International Workshop on Boolean Problems*, R. Drechsler and M. Soeken, Eds. Cham: Springer International Publishing, 2020, pp. 1–20.

[12] A. Henkel, B. Hussels, J. P. Kopetz, S. Krotsetis, N. Jochems, and K. Balzer, "Nutzer- und Aufgabenanalyse für ein sozio-technisches System zur Unterstützung der Kommunikation und Reorientierung beatmeter Patientinnen und Patienten in Intensivstationen: Ergebnisse und methodische Herausforderungen," in *Zukunft Der Pflege : Tagungsband der 1. Clusterkonferenz 2018 - Innovative Technologien für die Pflege*. BIS-Verl. der Carl von Ossietzky Universität Oldenburg, 2018, pp. 201–206.

[13] B. Kordts, J. P. Kopetz, K. Balzer, and N. Jochems, "Requirements for a System Supporting Patient Communication in Intensive Care in Germany," in *Zukunft der Pflege: Tagungsband der 1. Clusterkonferenz 2018 - Innovative Technologien für die Pflege*. BIS-Verl. der Carl von Ossietzky Universität Oldenburg, 2018, pp. 131–136.

[14] B. Kordts, B. Gerlach, and A. Schrader, "Towards Self-Explaining Ambient Applications," in *Proceedings of the 14th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA 2021)*. Corfu, Greece: ACM, 2021.

[15] D. Burmeister, "Selbstreflexive Geräteverbünde in smarten Umgebungen," Ph.D. dissertation, Universität zu Lübeck, 2018.

[16] D. Burmeister, B. Gerlach, and A. Schrader, "Formal definition of the smart object matching problem," *Procedia computer science*, vol. 130, pp. 302–309, 2018.