

Coping with the Complexity of Software for a Humanoid Robot Pepper

Peter Forbrig^[0000-0003-3427-0909] and Alexandru Bunea^[0000-0001-8315-3405]

University of Rostock, Department of Computer Science, Albert-Einstein-Str. 22,
18055 Rostock, Germany

{Peter.Forbrig | Alexandru-Nicolae.Bunea}@uni-rostock.de

Abstract. Robots are used in several domains in our society. Humanoid robots have especially many features and are used in shopping malls and as coaches for training activities. We use a humanoid robot Pepper as motivator for perform specific exercises that have to be performed by post-stroke patients. The number of people living with stroke-related disability is increasing. Because the number of physiotherapists and occupational therapists is not sufficient, humanoid robots might help to bridge the gap between demands and available resources.

However, the necessary software is quite complex. This report discusses a case study that resulted in an architecture that allows to externalize some services from the robot. In this way a kind of “thin robot” is used.

Keywords: Thin Social Humanoid Robot · Software Architecture · Rehabilitation of Stroke Patients · Arm Basis Training · Neglect Therapy · Mirror Therapy · Arm Ability Training

1 Introduction

Humanoid robots became popular during the last decade. They are used as advisors in shopping malls, as therapists for people with dementia, as coaches in training sessions or teachers for children.

In our project E-BRAiN (Evidence-based Robot-Assistance in Neurorehabilitation; www.ebrain-science.de) we study whether a humanoid robot can support patients after stroke in daily repetitive training tasks that are intended to recover disabled body functions. After an introduction of the exercises by a human therapist, Pepper is intended to replace the therapist in after the first session.

Stroke is a wide spread disease [8] that can lead to activity limitations. Therapies for stroke survivors with severe arm paresis exist. (i.e. the Arm Basis Training (ABT) ([8], [9], [12]) or the Mirror Therapy (MT) [20]).

We started implementing the instruction for training tasks and the feedback according to the tutorials of Softbank Robotics [17]. After a while we recognised that the dialogs became quite complex and it was difficult to maintain the software. Wagner and Deissenboeck mention in [21]: “Consequently, the determination of an appropriate abstraction level for the multitude of artefacts that form a software system is an

integral part of software engineering.” We recognised that we should model the navigation structure on a more abstract level as a state machine. An interpreter was implemented for this purpose. Some details of this approach will be discussed in this report. However, the exercises for post-stroke patients will be presented at the beginning. Afterwards, we focus on modelling the navigation structure and on providing a software architecture. Related work is discussed in the fourth section. The report closes with a summary and an outlook.

2 Training Tasks

The clinical members of our E-BRAiN team selected the Mirror Therapy (MT), the Arm Basis Training (ABT), the Arm Ability Training (AAT) and the Neurovisual Therapy [19] as candidates to be implemented digitally. These training therapies require daily repetitive training schedules. These therapies will be shortly described.

The Mirror Therapy is a treatment option for stroke survivors with severe arm paresis. A mirror is placed on a table between the arms. In this setting, the illusion of “normal” movements of an even severely affected paretic limb can be provided by the mirror image of a moving non-affected limb [20].



Fig. 1. A Situation during a mirror therapy.

A patient receives instructions to exercise his non-affected arm, to look into the mirror and to imagine that his affected arm is moving (see **Fig. 1**). This imagination is able to stimulate the brain areas around the stroke-affected areas and promotes the recovery of motor control of these areas and hence can result in a recovery of the ability to move the affected arm.

The Arm Basis Training has been designed for patients with severe arm paresis but ability to move the arm. Within a systematic training structure all segments of the arm and hand are repetitively trained. Fig. 2 gives an impression of the therapy.



Fig. 2. ABT training exercises with a therapist (from [10]).

The Arm-Ability Training has been especially designed to promote manual dexterity recovery for stroke patients who have a mild to moderate arm paresis. Platz and Lotze report in [11] about its design, clinical effectiveness, and the neurobiology of the actions. **Fig. 3** gives an example of one training task where a patient has to follow a labyrinth as fast as possible. Touching the labyrinth wall results in a one second penalty and the patient is supposed to hold his impaired training arm up in the during the exercise. The patient's performance will later be evaluated.

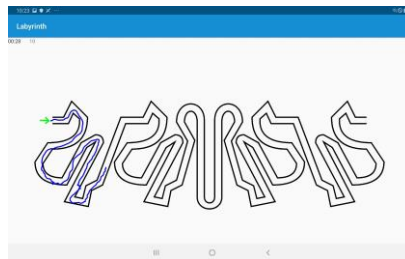


Fig. 3. 1 of the 8 exercises of the AAT: following a maze on a tablet with a pen.

Neglect is a visual problem after stroke. Left hemiplegic patients have evident difficulties orienting their eyes towards their left field of vision. As a result, affected patients might e.g. leave the entire left-hand side of their meals untouched, because their field of vision is lacking the information of their left eye.

The neglect therapy provides exercises to cope with this problem. **Error! Reference source not found..** Provides an example for this.



Fig. 4. Exercise for finding the number of persons

During the classical therapy a professional therapist observes and supports the performance of exercises. In our project Pepper will do this job. One precondition was

the digitalization of training task. The sensors of the robot were otherwise not able to identify mistakes. Additionally, dialogs for instructions, feedback and motivational sentences had to be implemented.

3 Using the Humanoid Robot Pepper as Therapist

The humanoid Robot Pepper [17] can talk, provide data on its tablet, move the head and its arms. **Fig. 5** provides an impression of the usage of Pepper. It shows a situation where the robot explains a training task of the mirror therapy and shows two pictures. The first picture demonstrates the start situation and the second picture the end position of the arm. A video of the whole movement was shown beforehand.

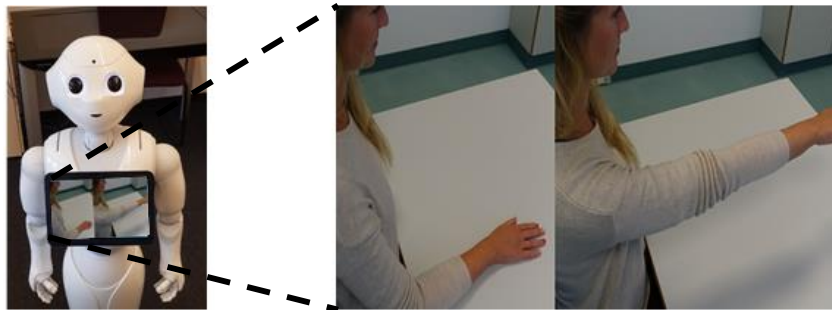


Fig. 5. Pepper with pictures of a training instruction on its tablet.

All data from the robot's tablet can be displayed on other tablets as well. For the mirror therapy there is no feedback about the performance because the exercises are performed with the healthy arm and one can assume that all patients can perform them correctly. More important are the sentences of the robot after the half of the prescribed repetition of a exercise: "Have a look into the mirror and imagine that your handicapped arm is moving". The goal here is to have an intense stimulation on the brain.

However, for the arm basis training we will analyse the movement of a patient and a helper. For this purpose, 3D cameras are used. **Fig. 6** gives an impression of one analyzed perspective



Fig. 6. Analyzed picture of one perspective of an ABT training task.

4 Modelling the Dialogs of the Robot

4.1 Implementing the Dialog on the Robot

According to the tutorial of Softbank Robotics applications can be installed on the robot. One can program them in such a way that the robot shows text, pictures and videos on its tablet. Additionally, the robot can talk and listen to persons. The mirror therapy was completely implemented in this way. This app worked quite well with therapists and students as test persons. For the arm-ability training we recognized that using the tablet of the robot is very difficult. Therefore, it was decided to use additional tablets that can be put on a table. The connection with the robot was established via direct wireless communication. Unfortunately, for longer interactions we realized problems because some messages were lost sometimes. This resulted in arbitrary deadlocks on the robot. We looked for other communication solutions and identified the MQTT technology [15] as a good option. Messages are not sent directly between sender and receiver but via a MQTT server. At this server clients can register for specific channels and topics. Messages are not sent only once by the server but again and again until the receiver got them. No message gets lost in this way. With this change in our software architecture we reached a stable application.

4.2 Abstracting the Dialog Specification and Using External Components

Having this technology available we thought about breaking down our monolithic application on the robot into separate components. From our experience in HCI and supported by [16] we decided to abstract the dialogs of the robot by final state machines. An own interpreter was implemented in Python that allows the execution of state-machine specifications. Our implementation allows hierarchical states like `s1/s11/s111` and integrates MQTT messages. Therefore, messages can be sent in states and during transitions. Currently, four different types of methods are implemented in our class `DialogEngine`. Below is an example for each of the four cases. The variable “`de`” has an object of the dialog engine as its value.

1. `de.add_message_for_state('state/s1', {'text': 'hello', 'image': 'ebrain1.jpeg'})`
2. `de.add_event_transition(source='state/s1', target='state1/s3', guard=is_button_pressed)`
3. `de.add_timed_transition(source='state1/s11', target='state1/s12', timeout_in_s=4)`
4. `de.add_interaction_for_state('state2/s14', video='video.mp4')`

The first method results in the construction of substate `s1` of state. While entering the state a MQTT message with the corresponding name value pairs is sent. The receivers have to interpret the corresponding content. For this example, pepper could say hello and present the picture on his tablet like in **Fig. 5**. The second method results in a state transition of the dialog. However, a condition has to be fulfilled. In this case the function is called `is_button_pressed`. Transitions allow very flexible and extendable

specifications. New ideas can be introduced easily. A timed transition is executed after a certain interval of time without any side effect. The fourth method results in a MQTT message for the presentation of a specific video. Other options are “say”, “text” or “image”. This list of options is extendable if new requirements are identified.

Based on the MQTT technology we were able to develop a software architecture where robot control, apps for exercises, management of therapies and tool support for programmers are separated (see Fig. 1).

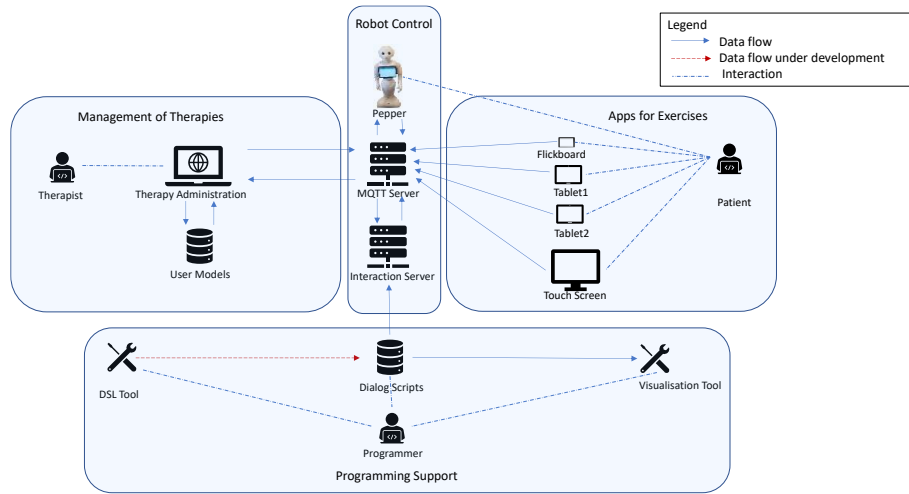


Fig. 7. Software architecture of our E-BRAiN system.

Fig. 7 gives an impression of the architecture we use. The communication between different software components is performed by message exchange via a MQTT server [15]. A flickboard [2] is used for therapy sessions to represent “yes” if a hand is put on it, since some patients might have problems to pronounce a “yes” in such a way that Pepper can understand it. The two tablets are necessary for some exercises of the arm-ability training and the touch screen is necessary for neglect patients.

On the interaction server the interpreter of the finite-state machine is installed. Dialog scripts control the interaction of the robot and a patient. We have been working on a domain-domain specific language that allows the generation of the python scripts (see e.g. [4]). A visualisation tool for hierarchical state diagrams allows the identification of errors like missing transitions or wrong guards. User models, diagnosed data, training data and the therapy sessions are managed by a web-based administration tool. A session of a robot with a patient can be started from the web tool. It delivers the necessary context information like starting time, number of repetition of exercises, previous training results, etc. With the provided architecture an integration of further tools and services is easily possible. Only MQTT clients have to be implemented for new tools and messages have to be specified. The existing clients have to be extended

by an interpretation of the new messages if they are expected to be receivers. Some more details can be found in [5].

With this architecture most software is external to the robot. It becomes a kind of “thin” robot with data presenting facilities only. (In analogy to thin and fat clients.) New developed services can be integrated easily. However, the capacity of the robot is not used. It could be built cheaper.

5 Related Work

The discussed approach has similarities to SOA transformation of businesses [7], where the authors mention “Determine what legacy/existing components can satisfy the service needs, determine what needs to be done with those components and migrate and reuse them as services. In doing this there is a need to determine the cost and effort of such a migration so as to make informed decisions about the feasibility of mining and reusing”. For a humanoid robot Pepper external services can be integrated to extend the performance of an application.

Wu et al. [22] provide an architecture to improve patient safety by coordinating medical devices’ behaviour to avoid preventable safety hazards. The goal is different but the solution to introduce a coordinator is similar.

From an abstract point of view our provided solution corresponds to the application of the mediator pattern of Gamma et al. [6].

There exist already several attempts to use robots for rehabilitation. However, most applications are implemented on the robot only and do not integrate management aspects. Choe et al. [1] studied already in the year 2013 a combination of physical and speech training for stroke patients with a robot. The robot was called humanoid but it looked like a machine with a tv on top. Nevertheless, positive results were reached.

In [18] Schrum et al. report about an application of using a Pepper robot for exercises with dementia patients. The robot demonstrates a gesture and encourages the patient to copy the move. Pepper is additionally used to motivate patients by saying sentences like: “Do not slow down”. Unfortunately, it is not explained how situations for encouragement are detected and how it is decided what to say.

Pulido et al. [14] provide an architecture for a mirror game with the Nao robot is provided by Pulido et al. [14]. It uses the robot for providing instructions only. For precepting interaction information a Kinect sensor is used. An external decision support system selects the next action. The Nao is used similar to our Pepper as a kind of thin robot. However, communication seems to be performed directly.

Raux and Eskenazi [16] provide already 2009 the idea of using state machines for spoken dialogs.

Polak and Levy-Tzedek [13] developed a gamified system for post-stroke long-term rehabilitation. They use the humanoid robot Pepper. In one game a patient and a robot are placed on different sides of a shelf. The patient has to put jars on the shelves according to the order displayed on the robot’s tablet. Game instructions and feedback are provided by Pepper. The authors mention the following: “After each trial, the robot either gives the patient feedback on the timing (e.g., “try to do it faster next

time”) or on the success on the task (e.g., “you succeeded!”, “you were not right this time, but I'm sure you will make it next time!”).” The approach shares similarities with our ideas. However, the therapeutic approach is different and the software is running on the robot only.

6 Summary and Outlook

In this report, we discussed ideas of a software architecture for humanoid robots like Pepper that allows the communication with external services. We used the MQTT technology that does not need much resources. It allows to cope with the complexity of our E-BRAiN system. However, other technologies would be possible as well. In all cases humanoid robots can be used as a kind of “thin robot”, where most of the software is running external. In the discussed case studies, the robot is used as speaker and data displayer only. However, the movement of arm and head while speaking is part of the operating system of Pepper.

Our application is the supervision of training tasks for stroke survivors with moderate to severe arm paresis. The humanoid robot acts as therapeutic assistant. It provides instructions and gives feedback to training tasks. Currently, we are working on rules for feedback to the patients that are based on clinical personal data, individual details regarding the prescribed training and the emotional states. The robot will refer to various information sources and especially to individualized goals of a patient in order to provide an appropriate motivation.

Additionally, further identified characteristics of human therapists might be used to modify the humanoid robot’s behavior. Further apps on tablets and touch screen for other training tasks will be developed and evaluated.

We hope that we can start evaluations with patients in the near future and hope that they will show relevant clinical benefits by training with the E-BRAiN system. More details about the digitalisation of training tasks can be found in [3] and [4]. Videos are available at <https://www.swt.informatik.uni-rostock.de/webebrain/>.

Acknowledgements

This joint research project “E-BRAiN - Evidenz-based Robot Assistance in Neurorehabilitation” is supported by the European Social Fund (ESF), reference: ESF/14-BM-A55-0001/19-A02, and the Ministry of Education, Science and Culture of Mecklenburg-Vorpommern, Germany.

References

1. Choe, Y.-k.; Jung, H.-T.; Baird, J. & Grupen, R. A.: Multidisciplinary stroke rehabilitation delivered by a humanoid robot: Interaction between speech and physical therapies, *Aphasiology*, Routledge, Vol. 27, 252-270 (2013).
2. Flickboard: <https://uk.pi-supply.com/products/flick-hat-3d-tracking-gesture-hat-raspberry-pi>, last visited July 31st 2012.

3. Forbrig, P.; Bunde, A. & Platz, T.: Assistance App for a Humanoid Robot and Digitalization of Training Tasks for Post-stroke Patients, Proc. of Human Centred Intelligent Systems, Zimmermann, A.; Howlett, R. J. & Jain, L. C. (Eds.) Springer Singapore, 2021, 41-51 (2020).
4. Forbrig, P., Bunde, A., Pedersen, A., and Platz, T.: Digitalisation of Training Tasks and Specification of the Behaviour of a Social Humanoid Robot as Coach, HCSE conference, Eindhoven, The Netherlands, December, Springer LNCS Vol. 12481, p. 45-57 (2020).
5. Forbrig, P., Bunde, A., and Bader, S. (2021), Engineering the Interaction of a Humanoid Robot Pepper with Post-Stroke Patients During Training Tasks}, ACM, New York, NY, USA, <https://doi.org/10.1145/3459926.3464756>,
6. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) Design patterns: Elements of reusable object-oriented software, Addison-Wesley - Reading, Mass.
7. O'Brien, L., Brebner, P., and Gray, J. (2008) Business Transformation to SOA: Aspects of the Migration and Performance and QoS Issues, ACM, New York, NY, USA}, <https://doi.org/10.1145/1370916.1370925>},
8. Platz, T. Impairment-oriented training (IOT) – scientific concept and evidence-based treatment strategies, *Reestor Neurol Neurosci.* Vol. 22, no 3-5, 301-315 (2004).
9. Platz, T., van Kaick S, Mehrholz J, Leidner O, Eickhof C, Pohl M.: Best conventional therapy versus modular Impairment-oriented training (IOT) for arm paresis after stroke: a single blind, multi-centre randomized controlled trial. *Neurorehabilitation and Neural Repair* 23:706-16 (2009).
10. Platz, T., Elsner B., and Jan Mehrholz. J. (2015). Arm basis training and arm ability training: two impairment-oriented exercise training techniques for improving arm function after stroke. *Cochrane Database of Systematic Reviews* 17, 6, 666. DOI: <https://doi.org/10.1002/14651858.CD011854>.
11. Platz, T., & Lotze, M. (2018). Arm Ability Training (AAT) Promotes Dexterity Recovery After a Stroke—a Review of Its Design, Clinical Effectiveness, and the Neurobiology of the Actions. *Frontiers in neurology*, 9, 1082. doi:10.3389/fneur.2018.01082
12. Platz, T.: Impairment-Oriented Training - Official Homepage. Retrieved February 24, 2021 from <http://www.iottraining.eu/abt.html> (2019).
13. Polak, R. F., and Tzedek, S. L.: Social Robot for Rehabilitation: Expert Clinicians and Post-Stroke Patients' Evaluation Following a Long-Term Intervention. In Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (Cambridge, United Kingdom) (HRI '20). Association for Computing Machinery, New York, NY, USA, 151–160, <https://doi.org/10.1145/3319502.3374797> (2020).
14. Pulido, J. C., Suarez-Mejias, C., Gonzalez, J. C., Ruiz, A. D., Ferri, P. F., Sahuquillo, M.E.M., Ruiz, C. E. De Vargas, Infante-Cossio, P., Calderon, C.L.P. and Fernandez F.: A Socially Assistive Robotic Platform for Upper-Limb Rehabilitation: A Longitudinal Study With Pediatric Patients. *IEEE Robotics Automation Magazine* 26, 2 (2019), 24–39. <https://doi.org/10.1109/MRA.2019.2905231> (2019).
15. Pulver, T. (2019). Hands-On Internet of Things with MQTT: Build connected IoT devices with Arduino and MQ Telemetry Transport (MQTT), Packt Publishing Ltd., Birmingham
16. Raux, A., and Eskenazi, M.: A Finite-State Turn-Taking Model for Spoken Dialog Systems. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Boulder, Colorado) (NAACL '09). Association for Computational Linguistics, USA, 629–637 (2009).
17. Softbank Rob.: <https://www.softbankrobotics.com/corp/robots>, last visited July 20, 2021.
18. Schrum, M., Park, C. H., and Howard A.: Humanoid Therapy Robot for Encouraging Exercise in Dementia Patients. In Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction – HRI 19, Daegu, Republic of Korea, IEEE Press, 564–565 (2019).

19. Trauzettel-Klosinski, S.: Rehabilitation for Visual Disorders, *Journal of Neuro-Ophthalmology*: March 2010 - Volume 30 - Issue 1 - p 73-84, <http://doi:10.1097/WNO.0b013e3181ce7e8f>
20. Thieme H, Morkisch N, Mehrholz J, Pohl M, Behrens J, Borgetto B, Dohle C. Mirror therapy for improving motor function after stroke. *Cochrane Database of Systematic Reviews* 2018, Issue 7. Art. No.: CD008449. DOI: 10.1002/14651858.CD008449.pub3 (2018).
21. Wagner, S., and Deissenboeck, F. (2008). Abstractness, Specificity, and Complexity in Software Design, ROA'08, May 11, 2008, Leipzig, Germany.
22. Wu, P-L., Kang, W., Al-Nayeem, A., Sha, L., Berlin, R. B., and Goldman, Julian M. (2013), A Low Complexity Coordination Architecture for Networked Supervisory Medical Systems, ACM, New York, NY, USA, <https://doi.org/10.1145/2502524.2502537>.