# Reusable Motivational Instruction Patterns for Socially Assistive Robots

Sebastian Schneider*, Michael Goerlich* and Franz Kummert*
*Applied Informatics
CITEC, Bielefeld University
Email: [sebschne,mgoerlic,franz]@techfak.uni-bielefeld.de

***Abstract*—Robots are increasingly tested in socially assistive scenarios. Future applications range from dieting, coaching, tutoring to autism therapy. In order to have a successful interaction with an artificial system, these systems need to have an interactional motivation model of how to assist users and encourage them to keep on with the task. In previous work, we have investigated how to build such a model for a specific scenario (e.g. indoor cycling). In this paper we want to show how to advance this model to be generalizable for other sport scenarios like rowing or bodyweight training. Therefore, we describe our framework for coordinating interaction scenarios with socially assistive robots.**

## I. INTRODUCTION

Research in Socially Assistive Robotics (*SAR*) aims at designing scenarios, where robots instruct people during rehabilitation tasks, diet coaching or cognitive tasks [3, 6, 7]. Those scenarios and systems are often build from scratch and underlying interactional instruction patterns are hand-crafted for each scenario. This leads to recurring implementation of interaction structures for each scenario that are hard to compare across different systems or use cases. To the authors knowledge, few publications in SAR research exists which describe the architecture of their proposed SAR systems with a focus on how motivational feedback is generated and defined. Mead et. al. [9] describe in their paper an architecture for rehabilitation task practice in SAR. They describe a system offering different server and controllers managing the interaction between the robot and the human. However, it is not presented how the conversational feedback for motivating the user is designed during the rehabilitation tasks. In [5], Jayawarden et. al. propose a three layered architecture for rapid prototyping of SAR system and easy to use behavior description for subject matter experts (SME). However, the focus of their implementation was not on general motivational patterns robots can use for providing assistance. Therefore, the interaction and feedback provided by the robot is customized by the SME during an iterative end-user design process.

Because SARs usually provide hands-off support through verbal interaction and physical presence, the main challenge is to establish a common concept of the user's motivation and the compliance to interact with such systems. To evaluate and compare the effectiveness of the SARs (i.e. the encouraging support), needs to be modeled in interaction patterns that are reusable across different domains and applications. Reusability of common concepts and frameworks, which capture motivational support, in this domain can help researchers to measure the progress in this scientific field and to improve on previous established patterns. In previous work, we have targeted the application scenario of an robotic indoor cycling coach. During our investigations we have developed a motivational interaction model, which we have evaluated in an extended long-term study [14]. Currently, we are working on the generalizability of this model for different sport domains. We have advanced our previous implementation to ease to process of designing sport scenarios with robot assistance. Therefore, we pursue an integrated framework approach which targets four main advantages:

- Help non-expert programmers to implement robotic-assistance scenarios using a domain-specific language,
- use the same instruction patterns for each scenario,
- provide an easy to use configuration setup for the system to make decisions, and
- make components reusable.

We hope that the generalizability and reusability of our approach will help to build a toolbox which eases the process to explore new scenarios that require social assistance. The paper is organized as follows, first we will give a brief introduction of motivation as a key component for building SAR robots. Afterwards, we will explain our prior research efforts in this domain. In Section IV, we explain our current framework for designing SAR robotic scenarios and end our explanation in Section VII with an introduction of our current target scenarios. At last, we give a discussion and conclusion.

## II. MOTIVATION: A KEY COMPONENT

In order to develop a common concept of motivational support for SARs, it is indispensable to identify the key components of motivation from the viewpoint of different disciplines. Motivational psychology discriminates two types of motivation: extrinsic and intrinsic motivation [4]. Extrinsic motivation itself can be divided into instrumental motivation, external self conception and goal internalization. Instrumental motivation influences the behavior of people based on a prospective external reward. External self conception is based on the perception of the ideal from one's personal role and the expectation of one's social surrounding. Goal internalization means that people make the corporate/institutional goals as their own. In contrast, intrinsic motivation is divided into intrinsic process motivation, which means that someone is doing a task because of enjoying to do the task, and internal self conception, referring to behavioral change based on personal values and standards. Research has shown that intrinsic motivation is more effective for long-term interventions. Thus, many assistive systems make use of the theory of flow [1] for their task assistance and adapt the task difficulty to match the user's individual optimal challenge [2, 8]. Hence, motivation is often defined as a force which drives human behavior. This definition focuses on the internal states of an individual person. However, in socially assisted scenarios one main goal is also to collaboratively achieve a target. Therefore, also a sociological and linguistic perspective is important, which analyzes the different multi-modal cues during interactional processes. This means that some form of communication needs to be established which helps express one's desires and intentions. Therefore, future systems ideally also need to deal with wrong communication, need to have repair mechanisms and have a concept of when to trigger which kind of supportive feedback in a multi-modal manner in order to achieve a goal-oriented interaction [11].

## III. PREVIOUS WORK

In our previous work we have investigated the instructional structures and motivational strategies that human trainers incorporate into everyday workout (i.e. indoor cycling) in real world Human-Human Interaction. During field investigations, we recorded and observed the interaction between a coach and an athlete during indoor cycling session. The goal of this investigation was to identify some common interactional patterns or concepts of feedback and acknowledgment that coaches use to motivate and engage their athletes [13]. A qualitative analysis revealed a complex multimodal structure of motivation-relevant processes that are fine-grained and sequentially . This model had to be reduced to an interactive action-based motivation model due to the limitations of current robotic systems (see Fig. 1).

It captures the aspects of *preparation*, *instruction*, *acknowledgment*, *repair* and *feedback* (i.e. continuer-, encouraging-, positive-, end-oriented- feedback) in a systematic way for a single exercise instructions/movements. It has been implemented for a robotic-assisted indoor cycling scenario [14]. The states of the model where modeled as state charts. The transition between states where triggered based on assigned targets for each instruction and the resulting decision of specific decision servers (i.e. cycling with a specific cadence, power or posture). The implementation of this motivation model describes the instructional and motivational structure of **static movement patterns** (e.g. cycling with a target cadence) and **cyclic** repeating **movement patterns** (e.g. doing push ups, standing up - sitting down) in robotic assisted indoor cycling and has yet been tested only in this domain.

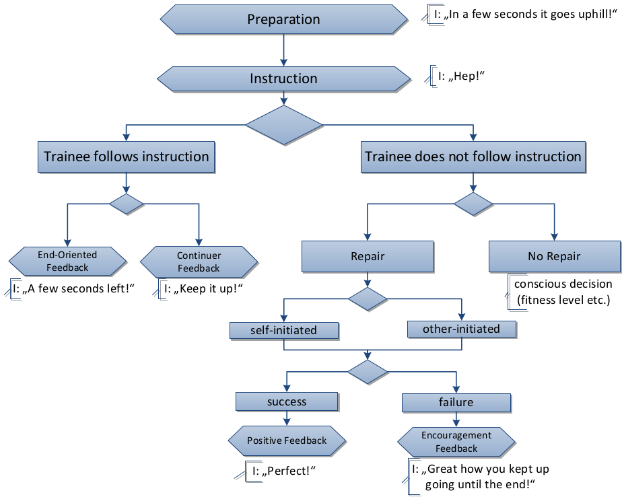
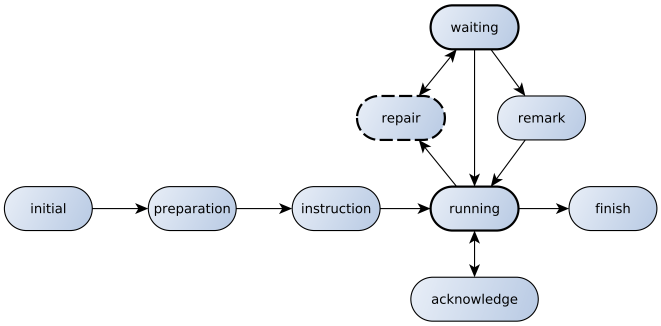
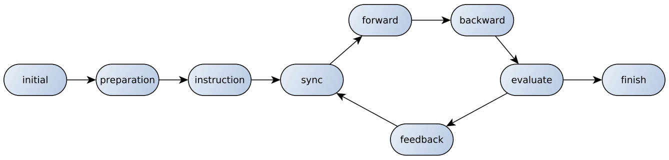


Fig. 1: Interactive action-based motivation model [14].



(a) Static Movement Instructions.



(b) Cyclic Movement Instructions.

Fig. 2: Generic instruction patterns for robot assisted sport scenarios.

## IV. TOWARDS A REUSABLE MOTIVATIONAL INSTRUCTION MODEL FOR SOCIALLY ASSISTIVE ROBOTS

In our current work, we want to make these instruction models applicable for different sport domains as well as intuitive and reuseable for non-expert users. Our proposed scenario design for socially assistance is depicted in Figure 3. In the following, we explain and motivate the different concepts (e.g. scenario coordination and decision server) of our design.

### A. Scenario Coordination

The scenario coordination is implemented using a domain specific language (DSL) which is automatically transformed

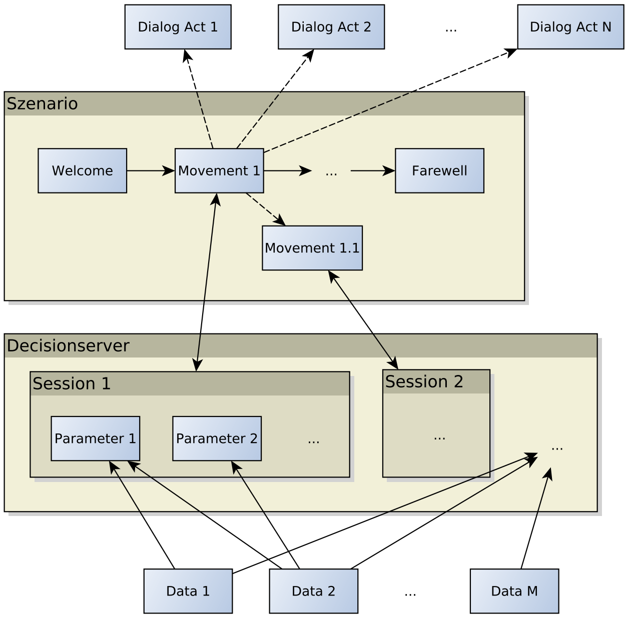


Fig. 3: Proposal for socially assistive scenario description.

to valid State-Chart XML[1] code (SCXML)[15]. State charts are commonly used to describe and coordinate the behavior of programs using events. Also the depicted movement patterns (see Fig. 2) are specified using the DSL. This specification includes the communication between different components in a distributed system and therefore simplifies the coordination (for details regarding the middleware see Section V of [10]). As tool, we use the Meta Programming System developed by JetBrains [2].

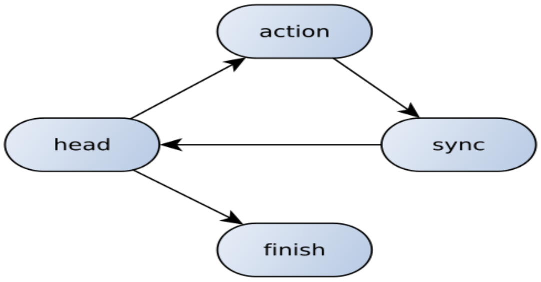
Each scenario is a state machine in which a number of different movements can be embedded and configured. Those movements represent the different exercises that a social robot can enquire a user to do. The movements are configured using the XML format. This configuration includes the actual dialog acts the robot produces during the different states of a movement as well as different targets (e.g. joint angle configuration of the user, speed or number of repetitions of exercises). Dialog acts can be any other state machine specified in our DSL. They can be simple text-to-speech acts, but also more complex dialog acts offered by a dialog system or even movements itself.

### *B. Hierarchical Instruction Patterns*

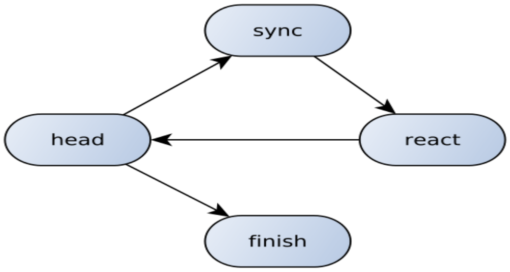
Previously, each state was assigned one verbal instruction from the robot. However, for teaching or learning scenarios it is important that states can trigger interaction movements also. Therefore, we have introduced an hierarchical concept in the current design of our interaction models. This means that each state of a **static** or **cyclic movement** can be a movement itself (see Fig. 3, *Movement 1* initiates *Movement 1.1*). This approach allows to trigger an instructing movement in which the robot helps the person to reach a specific pose required for an exercise or to trigger a correcting exercise if the execution of the user was not adequate.

### *C. Forward-Backward Cycle for Cyclic Instructions*

Cyclic instructions are important for exercises where the user changes his/her position. Consider doing push ups. The user goes down towards the ground and up again which results in a complete push up cycle. For scenarios where the robot and the user are doing an exercise simultaneously together (see Figure 6) the interaction models needs to allow a synchronous execution. This requires the model to have states for going to different positions during a cyclic movement and to synchronize the action of the robot and the user at some point. This synchronization is achieved by a waiting task. During the wait task the decider verifies if the user has reached the desired position. If the user does not comply, the system will run into a time out and continues with the next execution of the cycle. However, there exist also exercises where the order of different states of the exercises are important one synchronization point is not sufficient. We have introduced the concept of **act-/react-actions** for this issue (see Figure 4). Those actions take place during the forward or backward states of the **cyclic movement**.



(a) The act-action.



(b) The react-action.

Fig. 4: Two possible actions for the *forwad-backward* states

*1) Act-Instructions:* During **react-actions**, the user is in charge of the tempo of the exercise execution and the robot follows the user‘s lead.

*2) React-Instructions:* In **act-actions** the robot is initating the exercises and waits for the user to follow.

---

[1] http://www.w3.org/TR/scxml/

[2] https://www.jetbrains.com/mps/

## V. DYNAMIC DECISION COMPONENT

While the scenario coordination configures the interaction movements and executes them, the decision component receives the necessary information to make decisions during movement runtime (see Figure 3). Those decisions include recommendations to give a reparation, instruction, to praise the user or to terminate a movement. In the following, we describe the modular architecture of our decision system that is tailored to give designers of SAR scenarios the opportunity to build on a common framework for motivational support.

In general, decisions in assistive scenarios are based on some kind of data (e.g. strings, numbers, classification results). The way how decisions are made is inherently different between scenarios and implementation.

To give a main guidance for configuring decision systems and to add flexibility in the decision configuration for different scenario, we have implemented a **data-processing system**. This approach eases the process of deciding for different data types.

For each data type specific algorithms can be running which process the data and fulfill a special task. These algorithms are configurable and can be intertwined to solve more complex problems. The system defines components as well as input- and output-slots which can be connected.

### A. Configurable Data-Processing Pipeline

There is a variety of components that can be used to configure a data-processing pipeline available. In the following we explain the current different categories that are used in our system:

*1) datasource:* Data sources create initial data, which are at the moment SimpleRSBDataSource. This source receives data from our used middelware Robotic Service Bus (RSB)[16]. The data source is configured using a scope, where the data events are expected, and an expected data type. Additional, there is the ManualDataSource which purpose is for *Unit-tests*.

*2) transformations:* Transformations transform, as expected, data into another data format. Since we are currently using a lot of skeleton information from the Kinect, which are represented as XML-strings in our system, we have a transformation component that deserializes the skeleton joints in 3D vector objects. Furthermore, we have a component that calculates the joint angle from three 3D vector objects. Hence, it is possible to compute each joint angle by configuration. Additionally, there is a transformation component which deserializes JSON data types. At last, we have a descriptive statistic components from the *Apache Commons math* library[3]. This components allows to compute a running mean or median from incoming numerical values.

*3) deciders:* Deciders transform in-slots to decision results. Currently, there is a decider for floating point numbers, which verifies that an incoming value is in a specific range and a decider for classification results which calculates the entropy and only passes on a decision if the entropy falls below a threshold. For example, those thresholds are certain joint angle configuration the user has to reach or a specified cadence he has to cycle. Since joint angle configurations are mostly the same for many people, we do not have to adapt the threshold. Regarding sports like indoor-cycling, we have run a fitness test with participants to determine their individual thresholds. In the future, those thresholds can also be adapted due to training adaption effects.

Furthermore, there are deciders that filter decisions of other components. This decider can be configured to pass on a negative decision only when it had been raised during a specific time period. At last, there exists the *one of many positive* deciders which checks whether one of many decisions are positive.

### B. Local and Global Decisions

Each interaction session has a set of pairs of static and dynamic decision pipelines. One of these pairs reflects one exercise target of a movement (e.g. cadence of user during the cycling scenario). The static part is identical for each session and the dynamic part is distinct for one session. Furthermore, the static part is shared across sessions and usually does time dependent/consuming computations (e.g. average filter). The dynamic part always consists of at least one decider, which provides local decisions based on the results of the static part and the targets of the current movement.

Local decisions are represented as a *decision reason* which consists of the name of the parameter, the local decision, a timestamp and a boolean variable *good*, which indicates if a decision is negative or positive and reflects whether a goal is violated or not.

During one session all local decisions are collected into a *decision bag* (see Fig. 5). The *decision bag* is verified by the decider which then gives a guidance for a specific supportive behavior of the assistive system. Current implemented deciders are:

**Simple Decider:** The *simple decider* evaluates the *decision bag* for errors. Encountered errors are attached to the *decision reason*. If any errors are found, a *repair* advice will be send and the guidance is set to *failed*. If there is no error, an *acknowledge* is send.

**Hierarchic Reaching:** The *hierarchic reaching* allows to decide on multiple concurrent parameters. If one or many parameters are violated the *hierarchic reaching* can decide which parameter has priority.

**Hierarchic Monitoring:** The *hierarchic monitoring* decider is also an hierarchical decider. Instead of evaluating whether a target has been reached, it observers the specified parameters for a longer range of time.

At last, we have implemented components that *evaluate* the decisions. Those, are separated into *evaluation* strategies and *finishing* strategies. Those classes are decoupled from the decider because sometimes it is necessary to evaluate the state of the interaction due to different scenarios or contexts. The same goes for the finishing component, which can trigger the termination of a session, which in turn can trigger a

---

[3] https://commons.apache.org/proper/commons-math/

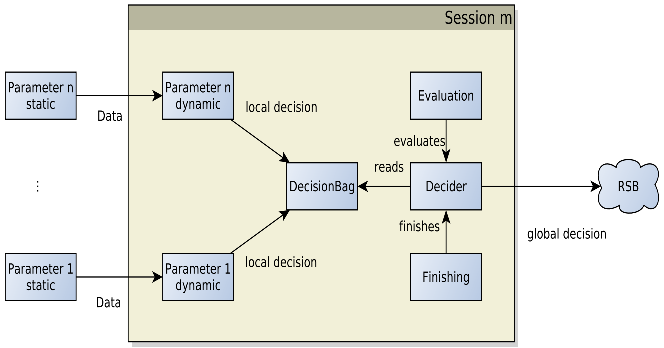


Fig. 5: Overview of the decision system.

supportive behavior. Currently we distinguish five different types of guidance: continue, repair, acknowledge, finished and failed. The details of the different guidance are:

**continue:** No reason for a change in the current situation.
**repair:** The known reasons make a reparation necessary.
**acknowledge:** The reasons favor a praise.
**finished:** The last known state was accurate.
**failed:** The last known state required a reparation.

The evaluation and finishing strategies are usually triggered after a certain amount of time has exceeded or after a threshold of specified events has been reached.

## VI. USAGE

So far we have described the different concepts that we see as building blocks for designing socially assistive robot scenarios. When a user wants to build a new scenario s/he can define an interaction flow using our provided IDE for developing RSB systems [10]. The user has to define at which points in the interaction what kind of movements should be triggered. If the user needs to define new movements, because there is no suitable movement configuration available, s/he can configure a new movement in XML format (for the limitation of the paper we will not include a detailed configuration) and define what the system should do depending on different measures. Those measures can be skeleton data, performance data from indoor bike, classification results or also new data provided from the user which are specific for a certain kind of scenario (e.g scores on a cognitive task [7, 12]). Depending on the type and goals of the intented scenario the user has to define what her/his parameters are. However, if certain parameter configuration already exist they can be easily included into a new configuration.

## VII. TARGET SCENARIOS

In the previous section, we have introduced the different concepts and implementations that we have used to create a scenario coordination for SAR. We hypothesise that the described motivational concepts are universal across different scenarios or applications of SAR and that the set of functionalities is sufficient to many purpose.

To evaluate the generalizability of our proposed scenario coordination and motivational movement patterns, we have implemented three different robot-assisted sport scenarios. You can see examples of Human-Robot Interaction scenarios in Figure 6. In the following we briefly describe our current scenarios:

**Indoor Cycling:** During the indoor cycling scenario the robot is instructing the user to cycle at different speed or resistance and in different positions like standing, sitting or doing push ups on the bike. Each movement is finished after a specific time which is based on the length of the different songs that are played during the indoor cycling session. We have evaluated this scenario during a extended long-term study [14]

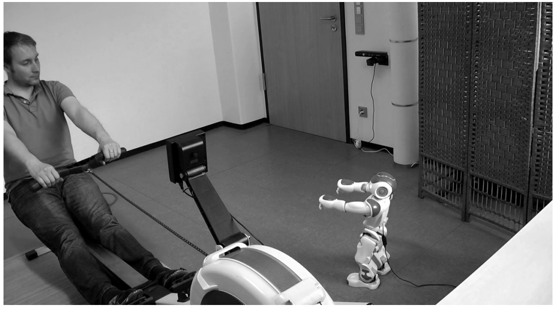
**Rowing:** In the rowing scenario the robot acts as a teacher explaining the user the different typical positions of a rowing stroke. It uses the concept of hierarchic reaching and repairs wrong stroke execution based on the following hierarchy: legs, back, arms. If one of the parameters is violated the system starts a movement which explains the correct execution of an exercise. We will compare this scenario against an interactive video which also explains the execution of a rowing stroke. As measure we will use the retention accuracy of how good participants remembered the steps of a rowing stroke after one week.

**Body Weight Training:** This scenarios aims at exploiting the embodiment of the robot. The robot and trainee do different exercises together (e.g. push ups, squats, lunges, etc.). We will implement different scenarios to test whether the user prefers robot initiated movements (see Fig. 4a) or self-initated movements (see Fig. 4b) and evaluate how different feedback strategies influence the assistive capabilities of the system.

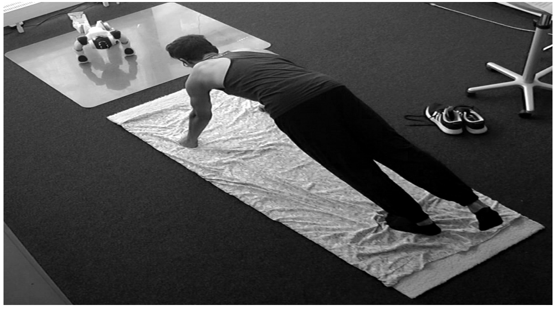
For all scenarios we use the same robot (i.e. Nao) in order to exclude effects due to the embodiment or appearance of the robot. Furthermore, we use the same decision system and scenario coordination as well as similar perceptive systems (skeleton tracking, heartrate, depth image of the user). We only needed to configure the explicit instructions and decision criteria which are unique for each interaction scenario.
Hence, we have acquired a state of the system where it is possible to reuse the same motivational model in all applications and use the same framework and implementations to create unique scenarios without worrying about implementational details. However, we need to evaluate whether the motivational model derived from indoor cycling scenarios is indeed applicable for other sport domains. Therefore, we will extensively test our target scenarios and evaluate the assistance in each scenario.

The implementation of these different scenarios results in a variety of different decision tasks and movement configuration. This different configurations are also reusable across scenarios and usable in new scenarios. We will work on building a set of configurations for movement and decision tasks that can be easily used when implementing a new SAR scenario.

(a) Nao as spinning instructor.



(b) Nao as rowing instructor



(c) Nao as bodyweight instructor.

Fig. 6: Different target scenarios using our proposed scenario coordination and movement patterns

At last, the concept of acknowledgement and reparation allows to easily compare different configurations for one scenario. The number of needed repairs can be used as a measurement to assess the effectiveness of the current configuration or classification system.

## VIII. CONCLUSION

In this paper we have presented our proposed framework for designing and coordinating scenarios for socially assistive robot based on motivational instruction patterns. We have introduced the key concepts and components that will help to guide the design of scenarios across different application domains. Furthermore, we have presented three different sport scenarios where we already use our proposed framework. We hope that in the future, our approach can be used to better evaluate different scenarios using different robots which are based on the same underlying models.

In upcoming implementations, we also target to develop a domain specific language model for the configuration of movement and decision tasks. We hope this will enable non expert programmers to develop and configure instructions for new scenarios or enhancing and reusing existing implementations.

From a motivational perspective, we currently focused on motivation from a multi-modal instructional point of view. In the future, we will further work on the relation between the instructional model and the psychological model of motivation. Since every person needs different types of motivation strategies, it might also help to include a further layer in the current model. This layer can describe what kind of motivational instruction, in relation to extrinsic motivation, is appropriate for which kind of user.

## ACKNOWLEDGMENTS

This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

## REFERENCES

[1] M. Csikszentmihalyi. *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. Jossey-Bass, 25th anniversary edition, April 2000. ISBN 0787951404. URL http://www.worldcat.org/isbn/0787951404.

[2] Juan Fasola and Maja J Matarić. Socially assistive robot exercise coach: motivating older adults to engage in physical exercise. In *Experimental Robotics*, pages 463–479. Springer, 2013.

[3] David Feil-seifer and Maja J Matari. Defining socially assistive robotics. In *in Proc. IEEE International Conference on Rehabilitation Robotics (ICORR05*, pages 465–468, 2005.

[4] R. W. Scholl J. E. Barbuto. Motivation sources inventory: development and validation of new scales to measure an integrative taxonomy of motivation. In *Psychological Reports*, volume Vol. 82 (3), page 10111022, 1998.

[5] Chandimal Jayawardena, I-Han Kuo, Elizabeth Broadbent, and Bruce A MacDonald. Socially assistive robot healthbot: Design, implementation, and field trials. 2014.

[6] Cory D. Kidd and Cynthia Breazeal. Robots at home: Understanding long-term human-robot interaction. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22-26, 2008, Acropolis Convention Center, Nice, France*, pages 3230–3235, 2008. doi: 10.1109/IROS.2008.4651113. URL http://dx.doi.org/10.1109/IROS.2008.4651113.

[7] Daniel Leyzberg, Samuel Spaulding, and Brian Scassellati. Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, pages 423–430, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2658-2. doi: 10.1145/2559636.2559671. URL http://doi.acm.org/10.1145/2559636.2559671.

[8] Manuel Lopes, Benjamin Clement, Didier Roy, and Pierre-Yves Oudeyer. Multi-armed bandits for intelligent tutoring systems. *arXiv preprint arXiv:1310.3174*, 2013.

[9] Ross Mead, Eric Wade, Pierre Johnson, Aaron St. Clair, Shuya Chen, and Maja J. Mataric. An architecture for rehabilitation task practice in socially assistive human-robot interaction. In Carlo Alberto Avizzano and Emanuele Ruffaldi, editors, *RO-MAN*, pages 404–409. IEEE, 2010. ISBN 978-142447991-7. URL http://dblp.uni-trier.de/db/conf/ro-man/ro-man2010.html#MeadWJCCM10.

[10] Arne Nordmann, Sebastian Wrede, and Jochen Steil. Modeling of Movement Control Architectures based on Motion Primitives using Domain Specific Languages. In *International Conference on Automation and Robotics*, 2015.

[11] E. Schegloff. When "others" initiate repair. In *Applied Linguistics*, volume 21(2), page 205243, 2000.

[12] S. Schneider, I. Berger, N. Riether, S. Wrede, and B. Wrede. Effects of different robot interaction strategies during cognitive tasks. In *ICSR*, volume 7621 of *Lecture Notes in Computer Science*, pages 496–505. Springer, 2012.

[13] Luise Süssenbach. Interaction and motivation in fitness communication, 2011.

[14] Luise Süssenbach, Nina Riether, Sebastian Schneider, Ingmar Berger, Franz Kummert, Ingo Lütkebohle, and Karola Pitsch. A robot as fitness companion: towards an interactive action-based motivation model. 2014.

[15] W3C. State chart xml (scml): State machine notation for control abstraction, 2014.

[16] Johannes Wienke and Sebastian Wrede. A middleware for collaborative research in experimental robotics. In *SII2011*. IEEE, 2011.