Fable: Socially Interactive Modular Robot

Magnússon, Arnþór; Pacheco, Moises; Moghadam, Mikael; Lund, Henrik Hautop ; Christensen, David Johan

Published in:
Proceedings of 18th International Symposium on Artificial Life and Robotics

Publication date:
2013

Fable: Socially Interactive Modular Robot

Arnþór Magnússon, Moises Pacheco, Mikael Moghadam, Henrik Hautop Lund, and David Johan Christensen

Technical University of Denmark, Center for Playware, DTU Electrical Engineering, Elektrovej, building 325, DK-2800 Kgs. Lyngby, Denmark
{arnma, mpac, mikm, hhl, djchr}@elektro.dtu.dk

Abstract: Modular robots have a significant potential as user-reconfigurable robotic playware, but often lack sufficient sensing for social interaction. We address this issue with the Fable modular robotic system by exploring the use of smart sensor modules that has a better ability to sense the behavior of the user. In this paper we describe the development of a smart sensor module which includes a 3D depth camera, and a server-side software architecture featuring user tracking, posture detection and a near-real-time facial recognition. Further, we describe how the Fable system with the smart sensor module has been tested in educational and playful contexts and present experiments to document its functional performance.

Keywords: human-robot interaction, modular robots, playware, vision

1 INTRODUCTION

Fable is a modular robotic platform under development, aimed at enabling non-expert users to design and develop socially interactive robotic creatures from various types of modules. A user can create their own Fable robots by assembling its modules into some configuration and programming it with the desired behavior based on a simple programming language. We explore the use of smart sensors modules, which are needed to achieve social interaction between a human and a robot. The smart sensor enables the robot to sense the user's behavior and respond accordantly. Smart sensors modules could for example be cameras or microphones combined with appropriate processing and mechanical encapsulation able to recognize the face or voice of the user. Mixing smart sensors and modularity introduces a set of challenges due to high bandwidth and processing beyond what a low power microcontroller, as typically used in modular robots, can handle.

This paper presents the design and tests of a smart sensor module for the Fable platform. The module contains a 3D camera, Asus Xtion Pro Live (similar to Kinect by Microsoft) and is equipped with connectors for compatibility with other Fable modules. The smart sensor module and corresponding software architecture provides the user with a simple programming interface for user tracking, posture detection and facial recognition. Our working hypothesis is that smart sensor modules will enable non-expert users to construct socially interactive and playful Fable robots and that this will be more motivating for the user than a system with no or only primitive sensors.

In the rest of this paper we first describe related work in Sec. 2. The Fable system is described in Sec. 3 and the software architecture for smart sensor modules is described in Sec. 4. Sec. 5 describes tests performed to study the system performance and the use of the system in an educational and an playful context.

2 RELATED WORK

Modular robots are comprised of independent robotic modules that can be attached and detached from one another, being able to construct various robot morphologies depending on the scenario [1]. Some modular robotic systems are able to self-reconfigure [2, 3] which allows the robots to shift their own shape by rearranging the connectivity of their parts to adapt to circumstances [4] [5], form needed tools [6] or furniture [7].

In this paper we explore user-reconfigurable robots, which allow non-expert users to physically construct their own robots from different types of modules (such as passive, actuators and sensors). This type of robots are particular suited for applications such as rapid robot prototyping [8], play [9, 10], rehabilitation [11, 12], composing music [13], and education (e.g. Mobot, Barobo, Inc. or LEGO Mindstorms). The modularity and open-endedness of these systems motivates the user to be creative, reflect, and iterate on their creations whereby they learn, train, or simply enjoy themselves.

In this paper we are considering modular robots for playful social interaction. Social interaction is being widely utilized for non-modular robots [14]. The interactive and animated iCat [15] is a type of user interface for controlling various media in a more natural way. Probo [16] [17] is a huggable robot designed to improve the living conditions of children in hospitals as a tele-interface for entertainment, communication and medical assistance. MeBot [18] is a telerobot that allows people in different places to feel present and to allow for social expressions, not only from video and audio, but also expressive gestures and body pose. Playware technology [19] has also been developed as a mediator for playful social interaction for over long distances.

3 FABLE: MODULAR ROBOTIC PLATFORM

3.1 Concept

The vision of the Fable project is to build a novel modular robotic platform, which enables non-expert users to assemble innovative robotic solutions from user-friendly building blocks. A robot is assembled by connecting two or more modules together where each module provides functionality to perform a specific task, such as sensing users, environment or actions, moving or manipulating its surroundings. With an easy-accessible software tool-kit provided, the user can create custom functionality and eventually we plan to make it easy to share their creations online with other users to try out, for inspiration, or as a starting point for their own creations. The objective is to develop this platform to enable users to realize their own innovative ideas.

3.2 Hardware

The Fable is a heterogeneous chain-based modular robotic system which consist of various modules, such as different types of joint, branching and termination modules [20]. Joint modules are actuated robotic modules used to enable locomotion and interaction with the environment. Branching modules connects several modules together in tree-like configurations. Termination modules may add structure, a visual expression, additional sensors, or actuators (e.g. grippers or wheels).

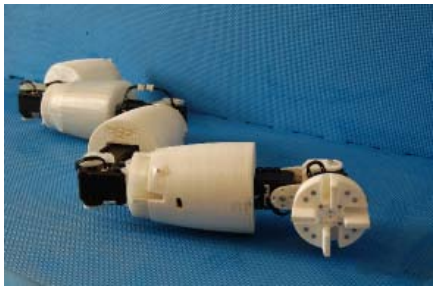
Every module, depending on its type has one or more mechanical magnetic connectors. The connectors are designed to allow rapid and solid attachment and detachment between modules. Further, the connectors are scalable to allow modules of different sizes to be combined and designed to allow neighbor-to-neighbor communication. Flanges mechanically lock the connection against twisting and bending and only allow disconnection by pulling on the axis perpendicular to the connecting surfaces.

Each module will be equipped with an onboard battery and an electronic board with an Atmel Atmega2561 microcontroller. Each board has four IR channels for communicating with neighboring modules and in addition, have the possibility to connect a ZigBee chip for wireless communication between modules or a PC. All modules have an onboard accelerometer and a gyroscope. The actuator modules have in addition a connector to power and can control several daisy chained AX-12A Dynamixel servos. These custom made electronic boards are in the process of being integrated into the system. Meanwhile we have used a compatible embedded system, CM-510, for centralized control of the modules.

The Fable system is still being extended with new module types to allow a wide range of different robots to be created. Fig. 1 shows examples of humanoid, snake and walking robots constructed with the current Fable system.

(a) Humanoid.

(b) Four legged robot.



(c) Snake.

Fig. 1: Example configurations with the Fable system.

3.3 Sensor Module Design

The Fable system is designed to include primitive and smart sensors:

Primitive Sensors sense simple signals, often from the environment, that can be processed by the small 8-bit microcontroller, also controlling the module, to extract meaningful low-level features (e.g. brightness, distance and temperature).

Smart Sensors sense simple or complex signals and process them to extract higher-level information about the user, the robot or its environment. Possible example includes, emotion categorization based on audio signals, gesture recognition based on accelerations, and posture detection based on 3D depth signals.

The design of smart sensor modules should preferably be fully embedded to make the robot system self-contained and independent of external devices. This is however challenged by the limited space, processing power and battery capacity which can be embedded inside a module. Another option is to offload the sensor outputs to a server through a wireless transport to have maximum processing power, but that also introduce challenges, such as the capability to transfer large amount of data wireless, potential delays in the system, and a higher hardware complexity. At this stage of the development we have chosen to use a server-based solution with a software architecture that can be ported to an embedded device in the future. The smart sensor module presented in this paper is therefore connected using

a tether to a server for processing and feature extraction. This high-level information is then offered as services to the modules wirelessly and can be accessed by the user application (as explained in Sec. 4).

4 FABLE SOFTWARE ARCHITECTURE

The section describes the software architecture of the Fable system with special focus on the server-side architecture to process smart sensor signals and provide them as services. The architecture is split into three main components: the user computer, the embedded device(s), and the server (see the diagram in Fig. 2.).

4.1 User Computer Architecture

To enable non-expert users to program their own Fable robots, we provide the LOGO programming language for user application development. LOGO is an educational programming language coming from Seymour Papert's constructionism tradition [21] and gives a more natural English-like syntax than C (e.g. no braces). We use a restricted version of LOGO, called PicoLOGO [22], which has a reduced instruction set and limited data types. An application is created by the user, which is then compiled into a byte-code representation and uploaded through a serial connection or a USB flash drive. This architecture makes it simple for the user to program the robot and easy for us, the developers, to add new Fable specific primitives to the PicoLOGO instruction set.

Listings 1.1 depicts an example LOGO program where the 3D depth camera is utilized to detect the angle of the left elbow joint of a user and maps it to a specific motor of the robot. Note that the symbol ';' starts a comment line and that the program will start on the function `onstart`.

Listing 1.1: Example PicoLOGO application.

```
constants [
  [userId 1]
  [left_elbow 0]
  [motorId 1]
]

to mirror_elbow
  ; Define a variable
  let [angle 0]

  ; Assigns the variable with elbow angle
  make "angle get_joint_angle userId left_elbow

  ; Set the motor position to angle
  set_motor_pos motorId :angle

  wait 1
end
```

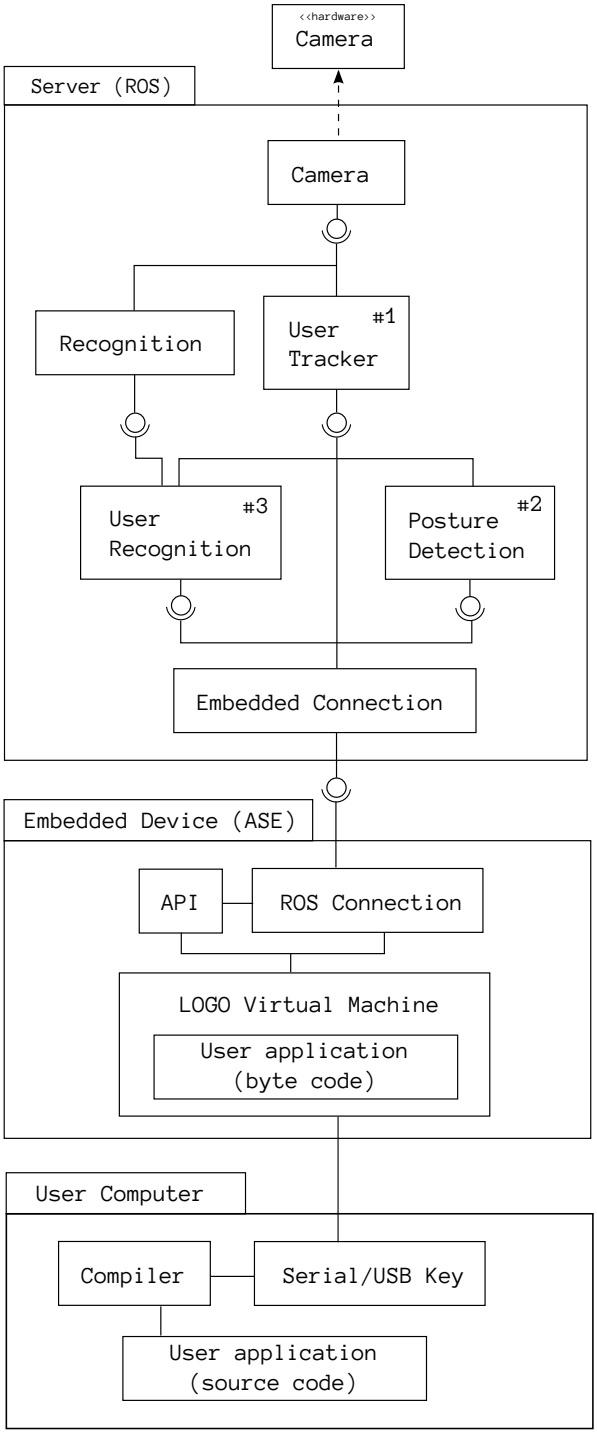


Fig. 2: Diagram of software architecture.

```
to onstart
  ; Requests for left elbow data
  use_user_joint left_elbow

  ; Infinite loop
  forever [mirror_elbow]
end
```

4.2 Embedded Architecture

Every module is controlled by an embedded device containing a small microcontroller. The microcontroller runs the Assemble-and-Animate framework (ASE) [23]. ASE is a flexible and extendible control framework targeted for modular robots, it provides high level of abstraction, libraries of algorithms, components and an asynchronous event-driven system. A virtual machine (VM) able to execute PicoLOGO code compiled into byte-code [22] is included as a process in ASE. Hence, the user application is executed by the VM, which may call functions that use smart sensor services from the server-side. The VM abstracts many of the system resources away, such as low level details of sensors, motors and distributed processing. This enables users to more quickly develop applications and upload them to the robot.

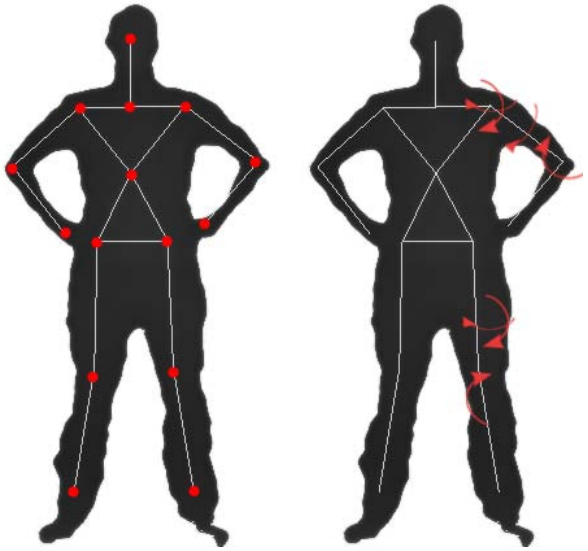
4.3 Server Architecture

The server architecture is designed to process signals from smart sensors and offer them as services to the user application. We utilize the Robot Operating System (ROS) [24] on the server, offering graph node structure, to create loosely coupled components which allows for task abstraction and code re-use. ROS provides a messaging framework for communication between components and also offers access to large sets of software libraries such as OpenCV and many community created components. The components role can vary from receiving sensor inputs, extract features or offering services to the user application running on the modules. The following subsections will describe some of these components and example applications.

User tracker component The robot must be able to sense users and their movements in order to enable interaction based on body language.

As a basis for tracking and detecting users we utilize OpenNI[1] user tracker module and Asus Xtion Live Pro (similar to Kinect from Microsoft). It allows for user detection and tracking by providing coordinates for all major limbs of the body as well the angles of the joints. The component is placed on the server, as seen in Fig. 2, component #1.

[1] OpenNI (Open Natural Interaction) is a framework created for sensor devices published under LGPL license. http://openni.org



(a) Red dots showing joint positions available for each user.

(b) Showing different angles of arms and legs that could be calculated from the joint positions.

Fig. 3: Image showing joint positions and angles available for each user.

These features form the basis for other components for further processing but are also offered as a service for the user application layer. An example application we have developed with this component is a Fable robot configured as a humanoid torso with the 3D camera module on top. The robot then mimics or mirrors the users shoulders and elbow movements with its two degree of freedom joint modules (the application is an extension of Listings 1.1).

Posture detection component An efficient way for humans to communicate is verbally, but in many cases it is supported by body language, such as body posture, gestures, facial expression and eye movements. Body language can provide clues to attitude or state of mind of a person, or simply to give commands or descriptions.

To enable such a communication between a robot and the user, we implemented a posture detection system where the robot can detect pre-defined postures stored in a database. A posture is a set of labeled angles, where each angle represents a joint on the user, gathered from previously described user tracking component. The user's posture is continuously being compared to the database, detection is considered when the Euclidean distance between two sets are below a certain threshold, and then an event is triggered. This component #2 can be seen in Fig. 2.

This functionality allows for a simple interaction between users and the Fable system. As an example we have developed

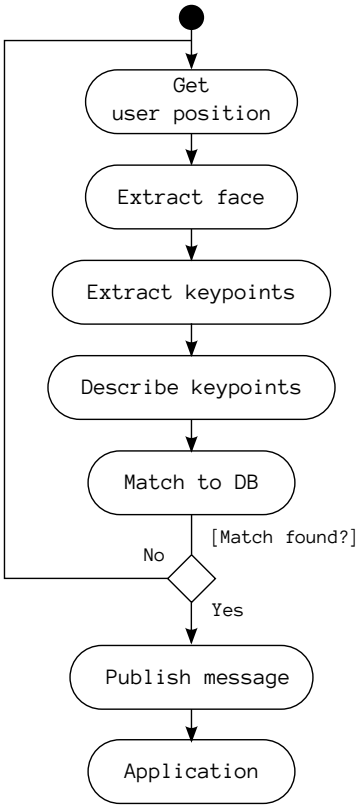


Fig. 4: Facial recognition process.

another mimic application, again with the modules configured into a humanoid torso and the 3D camera module on top. In this game the robot does a certain posture with its arms and then the user mimics that posture, when the robot detects the correct posture it makes a new one, and so on.

Facial recognition component A facial recognition service (see component #3 in Fig. 2) is provided in order to make the robot's behavior dependent on which person it is interacting with. For example a user developing an interactive robot application might want the robot to greet and speak the name of the user's family members whenever it sees them.

The facial recognition is performed in several steps, depicted in Fig. 4. First, the users face is extracted from an image with help from the user tracker component. Keypoints are extracted from these images with the FAST [25] algorithm, which are then given a description with the SIFT [26] algorithm. The descriptors are compared to the database with Fast library approximate nearest neighbor (FLANN) [27] matcher and on a match an event is sent. OpenCV's implementations of the algorithms were used.

5 TESTS

5.1 Playful User Test

The Fable system is designed to enable the construction of playful interactive robots. As a proof-of-concept, a pilot user test was performed with a 6 year old girl who interacted with a simple upper humanoid torso. The robot was pre-programmed with a LOGO application and pre-assembled from Fable modules. The robot was equipped with a smart sensor module, the 3D depth camera, directly connected to a server which was running the software architecture described in Sec. 4. An embedded controller, the CM-510, was running the LOGO application to control the robot's actuators based on the services provided by the server. Specifically, the LOGO application would use the user tracking service to mirror the girl's movement of hers shoulders and elbows. Two special postures, identified with the posture detection service, would start and end application.

We observed the girl's reactions during the pilot test (see Fig. 5). She was given the basic instructions about how a specific posture would trigger the mimicking. As soon as the robot started moving she reacted with positive excitement. Soon she realized the connection between the movement of her hands and the robots. As the play progressed she tried to go beyond the limits of the movements the robot can perform, for example, try to make the robot clap its hands and dance, but still appearing to be enjoying herself. In the end (after around 5 min.) she started to complain about her tired arms, as she had been moving them the whole time. The system was not glitch-free; there were occurrences where an unintended posture was detected, which resulted in ending the mimicking. Overall, this pilot-test confirmed that it was possible with the system to construct robotic application that would trigger play dynamics and interaction between a child and the robot. Further testing is needed to explore this in greater detail.

5.2 Educational User Test

One of the objectives of the Fable system is that it can be used as an educational tool. By tinkering with constructing robots, the user will be motivated to learn something about programming, mathematics, robotics, sensors, actuators, etc. in order to build better robots.

To test the Fable system's potential as an educational platform we have performed a user test with seven high school students. The students had little or no programming experience. They participated in a 2-hour programming exercise where they should create an application for a robot configured as a humanoid torso using the LOGO programming language. The robot should do a posture with its arms and the user is supposed to perform the same posture, then the robot does another posture and so on. This would require the participants to create functions, loops and use pre-defined functions that involve the robot moving its arms

(a) (b)

(c) (d)

Fig. 5: The Fable robot mimicking a 6 year old girl.

in certain positions and detecting users and their postures. A manual was supplied with the problem broken down into smaller steps and a presentation was given explaining the tools to program, compile, and upload the code to the robot.

We observed that the students quickly realized how to use the tools, while the actual programming, understanding functions and loops had a small learning curve. Most grasped the concepts quickly while others needed some extra help. By studying the manual they learned how to create functions for the different postures that the robot was supposed to perform. Some parts required extra effort, e.g. the part of detecting a user's posture. We observed that the students seemed highly motivated to understand the programming in order to make the robot do as requested. Clearly the students learned something about robotics and programming from this exercise. By letting the students define their own projects and create robots that makes sense to them, we anticipate that it will motivate the students to work concentrated for long time periods. In this process we hope that the students will learn useful skills, but the extent to which the Fable system allows for such open-ended creation and learning is a topic which we will explore more in future work.

5.3 Posture Detection

A test was performed in order to determine the detection rate of the posture detection algorithm. 9 test subjects, all male ranging from ages 25-35, were asked to perform a set of nine postures twice. No feedback was given from the system if a posture was detected or not. The camera was placed about 3 meters from the subject at 1.5 meters height.

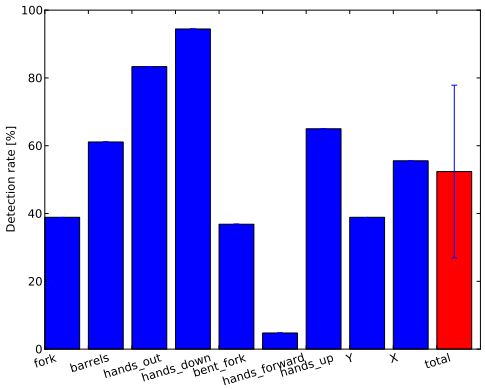


Fig. 6: Showing individual detection rates by postures, while the rightmost (red bar) shows the average detection rate.

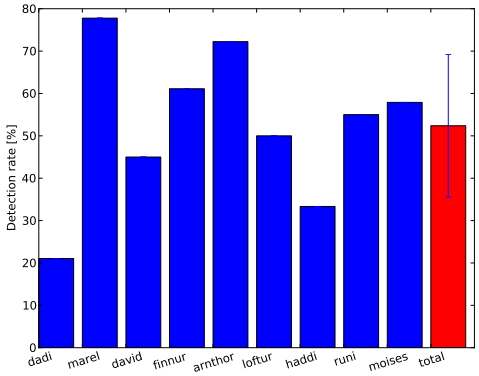


Fig. 7: Showing individual detection rates by subjects, while the rightmost (red bar) shows the average detection rate.

Fig. 6 shows the detection rates for each posture performed, while Fig. 7 depicts the detection rate per user. The accuracy (true positives / total detection) of performed postures was 95 %, where we observed false positives being recorded while the users were moving to the final posture. Note that one posture (`hands_forward`) had a very low detection rate. It turned out that since the hands were directed straight to the camera, it made the camera unable to correctly detect the positions of the hands. If the camera would have been placed lower this would not have affected the test. In summary the posture detection system is generally fairly capable to detect between the nine different postures, but it varies highly depending on the particular user. Further work is needed to make the posture detection component more robust.

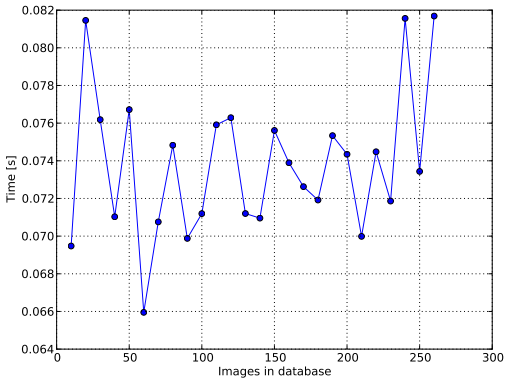


Fig. 8: Average time to match a face to database with varying database size.

5.4 Facial recognition

The facial recognition systems detection rate is highly dependent on the number of images in the database and specially the variation of image types, such as different illumination, face angles and distances. As the system is supposed to work in near real-time the speed of detections is a crucial factor and is therefore tested here in terms of the number of images in the database.

The test was performed on a database containing faces in different scales, distances and orientation of 5 subjects, all male ranging from the ages 25 - 35. Ten random images were incrementally added to the database at a time until 260 images had been reached, each time comparing 50 random images to the current database while recording the average time.

Fig. 8 depicts the average time to match a face to the database with varying database size. As one can see the average time is almost constant at 0.072 s or approximately 14 faces/sec, which is sufficient for most purposes. The corresponding detection rate is 75.8 % and the accuracy is 99.3 % (true positives / (true + false positives)). We anticipate that these results are sufficient for practical applications utilizing facial recognition but this has yet to be tested.

6 CONCLUSION

This paper presented a smart sensor module with corresponding server-side software architecture for the Fable modular robotic system. The sensor module is based on a 3D depth camera providing user tracking, posture detection and a near-real-time facial recognition. We tested the functional performance of these services and found them sufficient for our purpose although there is still room for improvements. Further, a pilot user test was described that demonstrated a playful and interactive robot build from the system. In addition, we described a user test with high-school students who used the system to program a robot and thereby learn about programming and robotics. In conclusion we anticipate that the Fable system equipped with smart sensor modules is a step towards a new type of user-reconfigurable robotic playware that motivate its users to be creative and learn while creating socially interactive robotic applications.

In future work a more integrated/embedded version of the smart sensor module will be developed (not tethered to the server). Further, the developments of new smart sensor modules are being explored and more thoroughly tested. In addition, the Fable system is being improved to be more user-friendly and include more module types.

7 ACKNOWLEDGEMENTS

This work was performed as part of the “Modular Playware Technology” project funded by the Danish National Advanced Technology Foundation. Many thanks to Brian Silverman who is the developer of the PicoLOGO virtual machine and compiler used to program the Fable robot.

8 REFERENCES

[1] T. Fukuda and S. Nakagawa. Dynamically reconfigurable robotic system. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA'88)*, pages 1581–1586, 1988.

[2] M. Yim, W-M Shen, B Salemi, Daniela Rus, M. Moll, H Lipson, and E Klavins. Modular self-reconfigurable robot systems: Challenges and opportunities for the future. *IEEE Robotics & Automation Magazine*, 14(1):43–52, March 2007.

[3] K. Stoy, D. Brandt, and D. J. Christensen. *Self-Reconfigurable Robots: An Introduction*. Intelligent Robotics and Autonomous Agents series. The MIT Press, 2010.

[4] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji. M-tran: self-reconfigurable modular robotic system. *Mechatronics, IEEE/ASME Transactions on*, 7(4):431 –441, dec. 2002.

[5] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund. Design of the atron lattice-based self-reconfigurable robot. *Auton. Robots*, 21(2):165–183, September 2006.

[6] K. W. Gilpin, A. N. Knaian, and D. L. Rus. Robot pebbles: One centimeter modules for programmable matter through self-disassembly. *IEEE*, 2010.

[7] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert. Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture. *IEEE International Conference on Robotics & Automation (ICRA)*, pages 4259–4264, 2009.

[8] A. Lyder, R. F. M. Garcia, and K. Stoy. Mechanical design of odin, an extendable heterogeneous deformable modular robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2008)*, pages 883–888, Nice, France, September 2008.

[9] H. S. Raffle, A. Parkes, and H. Ishii. Topobo: a constructive assembly system with kinetic memory. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 647–654, New York, NY, USA, 2004. ACM.

[10] E. Schweikardt and M. D. Gross. roblocks: a robotic construction kit for mathematics and science education. In *Proceedings of the 8th international conference on Multimodal interfaces*, ICMI '06, pages 72–75, New York, NY, USA, 2006. ACM.

[11] P. Marti, L. Giusti, and H. H. Lund. The role of modular robotics in mediating nonverbal social exchanges. *IEEE Transactions on Robotics*, 25(3):602–613, June 2009.

[12] C. B. Nielsen and H. H. Lund. Adapting playware to rehabilitation practices. In *Serious Games - Theory, Technology & Practice: Proceedings GameDays 2011*, 2011.

[13] H. H. Lund, N. K. Bærendsen, J. Nielsen, C. Jessen, and K. Falkenberg. Robomusic with modular playware. In *Proceedings of International Symposium on Artificial Life and Robotics (AROB'10)*, 2010.

[14] C. L. Breazeal. *Designing sociable robots*. The MIT Press, 2004.

[15] A. J. N. van Breemen. Bringing robots to life: Applying principles of animation to robots. In *Proceedings of the CHI2004 Workshop on Shaping Human-Robot Interaction - Understanding the Social Aspects of Intelligent Robotic Products*, Vienna, 2008.

[16] K. Goris, J. Saldien, I. Vanderniepen, and D. Lefeber. The huggable robot probo, a multi-disciplinary research platform. *Communications in Computer and Information Science*, 33 CCIS:29–41, 2009.

[17] J. Saldien, K. Goris, S. Yilmazyildiz, W. Verhelst, and D. Lefeber. On the design of the huggable robot probo. *Journal of Physical Agents*, 2(2), 2008.

[18] S. Sigurdur O. Adalgeirsson and C. Breazeal. Mebot: A robotic platform for socially embodied presence. In *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction (HRI '10)*, pages 15–22, 2010.

[19] H. H. Lund and T. Thorsteinsson. Social playware for mediating tele-play interaction over distance. *Artificial Life and Robotics*, 16:435–440, 2012.

[20] M. Pacheco, M. Moghadam, A. Magnusson, B. Silverman, H. H. Lund, and D. J. Christensen. Fable: Design of a modular robotic playware platform. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2013. (to appear).

[21] S. Papert. *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc., New York, NY, USA, 1980.

[22] Playful invention company. http://www.picocricket.com/picopeople.html. Developers of PicoLOGO.

[23] D. J. Christensen, U. P. Schultz, and M. Moghadam. The assemble and animate control framework for modular reconfigurable robots. In *Proceedings of the IROS Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems*, 2011.

[24] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, 2009.

[25] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 430–443. Springer Berlin / Heidelberg, 2006.

[26] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[27] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.