The Fourth IEEE RAS/EMBS International Conference
on Biomedical Robotics and Biomechatronics
Roma, Italy. June 24-27, 2012

Design, implementation and field tests of a socially assistive robot for the elderly: HealthBot Version 2

C. Jayawardena[1], I. Kuo[1], C. Datta[1], R. Q. Stafford[2], E. Broadbent[2], and B. A. MacDonald[1]

Abstract— **This paper presents the second version of a mobile service robot (HealthBot) designed for older people. The lessons learned from studies of the first version of the robot at a retirement village, and design decisions for the second version, are discussed. Technical requirements of field trials, a focus on cognitive human-robot interactions, the importance of working together in a multidisciplinary team, and the necessity for rapid iterative development suggested a new software framework. The features of new framework are discussed and implementation details are presented. Details of field trials and user acceptance results are presented. Results are promising for older-user acceptance of the robot.**

I. Introduction

Developing socially assistive robots is an emerging and important goal in robotics research. It is an interdisciplinary research area, which requires collaboration between a wide range of disciplines, including robotics, health sciences, psychology, gerontology, and human-computer interaction. This trend is a result of the increasing abilities of mobile service robots and the increasing needs of people for various kinds of help. In particular the older population of the world is growing dramatically [1]. On the other hand, there is an increasing shortfall in numbers of health professionals and caregivers [2] [3].

There have been many attempts to find assistive robotic solutions to these socio-economic issues. Mobility aids [4], manipulation aids [5], therapeutic aids [6], surgical robots [7], physical and mental rehabilitation robots are some examples [8], [9]. Among these solutions, ‘socially assistive robots’ belong to a distinct category.

Socially assistive robots are different from social robots and entertainment robots, which provide simple human-robot interaction. In contrast, socially assitive robots are expected to provide a broad range of services to support daily activities of users. However, designing such robots poses new challenges, as individualized requirements to cater for the special needs of each user need to be considered [10].

In this paper, we present the design, implementation, field tests and results of a socially assistive robot (the HealthBot) for older people. The first version of the robot and the results of field trials were presented in [11] and [12]. The main objective of the first version was to evaluate the feasibility of deploying socially assistive robots in a rest-home environment, and the results of the field trials were positive. Based on that experience, the second version of the HealthBot robot was developed with more features and a new software framework, which are presented in this paper.

A. Lessons learned from the past research

Several valuable lessons related to practical robot deployment were learned from the first field trial:

- The importance of developing techniques that can satisfy individual needs.
- The requirement of effective tools for faster robot behavior authoring and prototyping.
- The need for field customization.
- More involvement of Subject Matter Experts (SME) in the development process.
- The need for multi-modal interactions.
- The need for accommodating changing requirements.

Section II presents the key design considerations and section III presents the software development approach adopted. Details of the robot are presented in section IV. In section V, details of the software arechitecture are presented. The study setup for the field trials is detailed in section VI. Section VII presents the results of field studies.

II. Design considerations

The robot used for this study is a joint development of the University of Auckland/Auckland UniServices in New Zealand, with ETRI and Yujin Robot Co. Ltd., in South Korea. Robot hardware was provided by Yujin Robot and the software architecture and the study was designed and developed by the University of Auckland/Auckland Uniservices.

A. Feature richness

Our previous studies ([11], [12]) showed that socially assistive robots need to be equipped with a variety of services. In order to provide the features, the robot software framework should support interfacing with measuring devices, communication with external web services, database access, communication with sensor nodes, and invoking third-party software applications, in addition to the basic robot sensing, actuation and navigation services.

B. Customizability and rapid prototyping

In the HealthBots project, there is a focus on field trials with real participants, and, to improve the software continuously, an iterative approach was used throughout software development. The results of trials are used for continuous improvement and to deduce new findings. Therefore, the ability to develop robot applications rapidly is very important.

[1]C. Jayawardena, I. Kuo, C. Datta, and B. A. MacDonald are with the Department of Electrical and Computer Engineering, University of Auckland, New Zealand. c.jayawardena at auckland.ac.nz

[2]E. Broadbent and R. Q. Stafford are with the Department of Psychological Medicine, University of Auckland, New Zealand.

Another key consideration in the software architecture design was rapid customizability. This enabled the inclusion of real-time feedback from the SMEs, pilot groups, end users, and other stake-holders, while reducing the introduction of new bugs and minimizing additional software testing. The software architecture was flexible enough to accommodate new findings, suggestions, and new requirements, even during testing and deployment phases. The ability to customize the robot behavior (robot actions, visual display, and speech) without code modifications, was seriously considered in the design of the software framework (see section V). Customization is required to cater individual needs, incorporate suggestions of SMEs and to satisfy new requirements.

C. Involvement of SMEs

SMEs are professionals with expert knowledge in a particular domain. In health care robotics, SMEs include doctors, nurses, caregivers, health psychologists, and health care researchers, and their involvement is important in application development.

In traditional software development approaches, SMEs were mainly involved in requirement gathering and validation phases, but largely excluded in the development phase. In agile approaches, SMEs are heavily involved in the software design and the development phase, but still the programming is done by programmers. In our approach in HealthBot development, we tried to extend the involvement of SMEs to the extent of doing actual application development. Therefore, significant emphasis was placed on developing tools for robot application (robot behavior) development by non-programmers.

D. End-user programmability

Providing tools for end-users to customize the robot behavior without requiring changes to the code was another consideration. Although these tools will be completed in future versions of the robot, the software architecture was designed to support such front-end tools.

III. SOFTWARE DEVELOPMENT

In the previous study, traditional software development methodologies did not fit well for this domain. Therefore, the iterative approach illustrated in Fig. 1 was adopted. The development began with a set of rough requirements received from the SMEs. These were not complete; SMEs are not aware of all the requirements. That was not considered to be a problem, but an inherent feature of the problem domain. Since these kinds of robots are not yet widely available, the detailed requirements cannot be elicited until the product is tested with real users.

Based on the SME requirements, an initial prototype was developed, then iterated with SMEs and users, and field evaluations. It is very important to emphasize that the development was a collaboration effort between software engineers, robotic researchers, SMEs and users; rather than a purely technical endeavor. Once all stake holders were satisfied, at the field site, the development was frozen and the final application was used to collect study results.

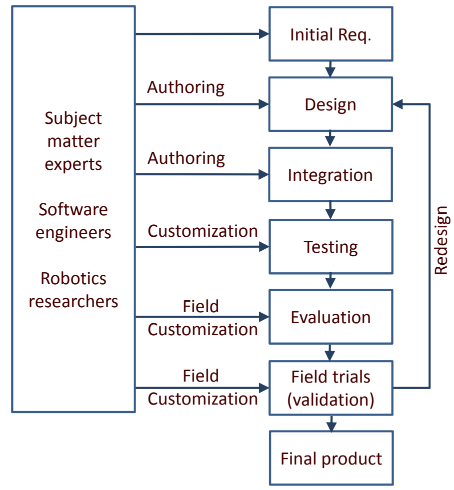


Fig. 1. Development Cycle

IV. ROBOT CAPABILITIES AND FEATURES

The HealthBots robot (Fig. 2(b)) is a differential drive mobile robot, powered by a 24v Li-Polymer battery. It consists of a rotatable touch screen, microphones, ultrasonic sensors, bumper sensors, and a laser range finder. The front-end of the application was developed using Flash/ActionScript 3.0 and the back-end was developed using C++. The robot software communicated with several web-services for information retrieval and update, and was integrated with third-party applications for providing added functionality.

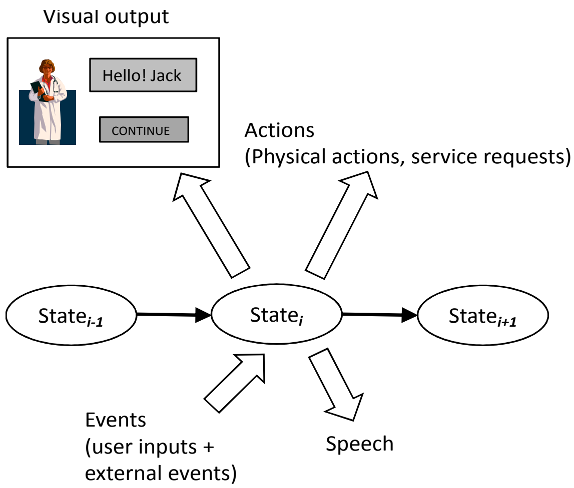
The robot's synthetic speech was generated through diphone concatenation type synthesis implemented with Festival speech synthesis system [13] and used a New Zealand accented diphone voice developed at the University of Auckland [14].

User responses were received via the touch screen and the robot responded to participants with synthesized speech, visual output on the screen and with movements.

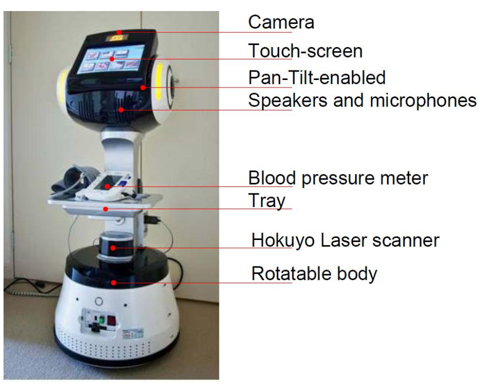
For map building and navigation, the robot uses the StarGazer robot localization system [15], with passive landmarks installed on the ceiling of the robot work-space at approximately one meter separation. A map of the area was built using the built-in map building module of the robot. The robot could then autonomously navigate to designated places, such as the charging station, and participants' apartments, and avoid obstacles using the pre-built map and landmarks.

A. Service modules

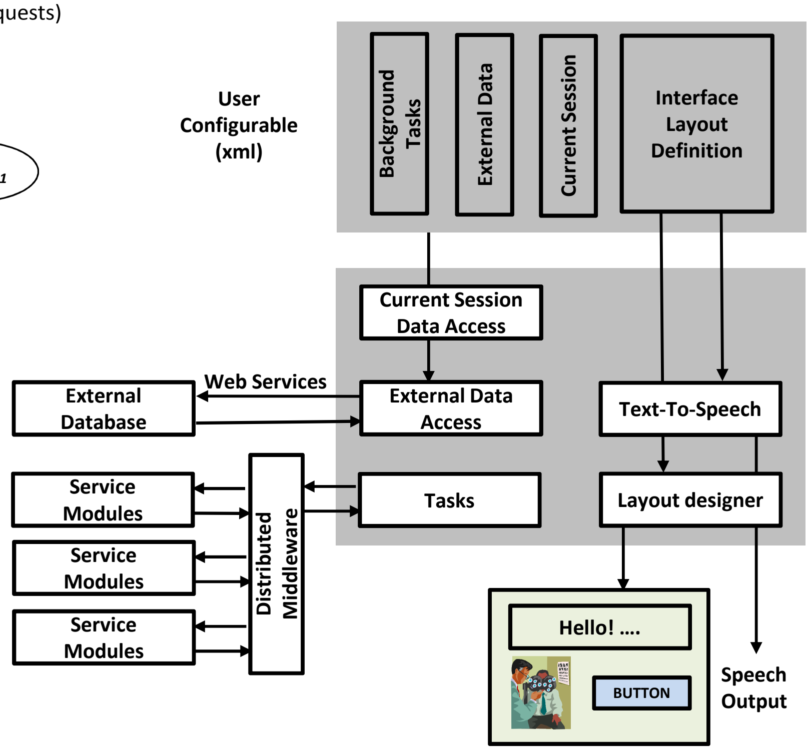
Seven service application modules were either developed or integrated; vital signs measurement, medication reminding, brain fitness games, falls detection, entertainment, brain fitness, and telephone calling. The vital signs module measures blood pressure, arterial stiffness, pulse rate, blood oxygen saturation, and blood glucose levels. The blood glucose monitor is connected to the robot using Bluetooth and the other devices were connected by USB cable. The medication management module reminds users of their medication schedules and consisted of a sophisticated dialog



(a) State behavior.



(b) HealthBots robot.



(c) Overview of the software architecture.

Fig. 2. HealthBot robot and architecture.

system connected to a back-end web service. A third-party software, Brain Fitness, provided games designed to be an enjoyable way for older people to practice their cognition and memory. The falls detection module was able to receive fall events from a sensor network installed in the environment and respond to those events. The entertainment module provided music videos, pictures, and quotes. The robot's calling module; developed using the Skype API, enabled participants to make telephone calls to friends and family.

V. Software Architecture

Distinct features of the architecture, depicted in Fig. 2(c), are as follows:

A. *Combining robot actions, user inputs and the graphical user interface (GUI)*

Usually, in robotic applications, the GUI is not included in the robot behaviour design. Instead, the focus is on robot behaviour such as path planning, navigation, and other actions. However, in service robot applications, the GUI is a dominant part of the robot behavior, since the user experience highly depends on the audio visual output and the interaction with the robot. Therefore, to have effective human-robot interaction, robot actions, user inputs, as well as changes in GUI, should be synchronized. The HealthBot framework supports coherent description of complete robot behaviour.

B. *Separation of the robot behaviour*

The complete robot behavior includes the following items:

- Components of the GUI (text, buttons, images, movies)
- Speech
- Events that the robot can receive and corresponding actions, such as network events and actions
- Background actions. i.e. things that the robot can do transparent to the user
- Messages sent and received

In the HealthBot architecture, the robot behaviour is isolated from the execution engine. The 'execution engine' is the core software and the robot behaviour is the complete robot application as explained above. Therefore, to develop a new application or to customize an existing application, changes to the core software are not necessary. This has the following benefits:

- Since modifications to the execution engine are not necessary to develop a complete new robotic application, the execution engine can be well tested and highly reliable

- The same execution engine can be re-used on multiple robots to deliver completely different applications, without much software development effort.
- Since the behaviour description language (which is based on XML) is very simple compared to a programming language, it can be edited by someone without any programming knowledge (for example, by an SME)
- New behaviors can be defined rapidly, since programming is not involved
- Changes to the robot behavior can be introduced at any time (even after the deployment) just by editing the behavior description file

The complete robot behaviour was modeled as a finite state machine and it was described using an XML notation, as explained in Sections V-C and V-D.

The execution engine was developed using C++ and ActionScript 3.0. The execution was designed to,

- render the screen layout,
- generate text-to-speech,
- access external web-services to get information,
- control robot movements,
- send and receive messages with back-end systems and various sub-systems,
- save data to databases using web-services, and
- invoke third-party applications

pertaining to the current state.

C. Behaviour description as a finite state machine

A state includes visual output, expected events, actions, and speech as illustrated in Fig. 2(a).

D. Behavior description language

The XML state language can be considered a domain-specific language for developing service robot applications, as the current version supports developing fairly complex applications. Fig. 2(a) shows a very simplified view of a state definition:

- *state*: Each state has a unique identifier
- *timeout*: The duration of a state timer. The timer completion event can be used for state transitions when necessary.
- *backgroundactions*: Actions transparent to the user. e.g. accessing a web service, sending a message.
- *screen*: All screen components (currently text boxes, buttons, images, video clips, on-screen keyboards, and on-screen numeric keypad). Buttons can have one or more associated events and events can have one or more associated actions.
- *expectedevents*: Events expected (or processed) by the current state. Timeout events, falls events, message received events, face detected events, are some examples.

E. Robot state transition visualization tool

To aid the fluent and efficient editing of the robot behavior XML, we developed a state transition visualization tool. The aim was to reduce the complexity involved while authoring the states in the FSM and creating new behaviors by showing the transitions between various states. The tool works by defining an XSLT transformation of the Healthbot XML to a more intuitive W3C SCXML schema description. The SCXML generated is further visualized in a graph using the dot language.

VI. Study setup

The studies were conducted at Selwyn Village retirement centre in Point Chevalier, Auckland, New Zealand, which covers 26 acres and has around 650 residents. To live in the village, residents must be aged 65 years or older. The average age of residents in the village is 88, and provided ranges from independent units to dependent and hospital care.

Fig. 3. HealthBot robot in use at the retirement village.

Three parallel studies were conducted:

1) Study 1: in public spaces (in independent apartment building common areas and rest home common rooms)
2) Study 2: in private spaces (independent living apartments and rest home rooms)
3) Study 3: monitoring studies with falls monitoring, wandering and activity monitoring in the rest home

This paper presents the results and observations pertaining to the robot when it used in the above three studies. Psychological results and other technical results are published elsewhere [16].

The robot spent approximately two weeks in an independent living building and approximately two weeks in a rest home. At scheduled times the robot visited apartments or rooms (study 2). The remainder of the day was spent in a public place (study 1). When the robot was in the public place, anyone could approach the robot and interact with it. For study 3, a ZigBee sensor network and other systems has been implemented to receive falls events from wearable accelerometer devices. Once a fall event is received, it was relayed to the robot and the robot reacted by going to the falls location and by starting a remote monitoring session.

VII. Results

In total 67 people interacted with the robot. There were 42 participants in study 1 (public spaces), 25 in study 2 (private spaces), and five in study 3 (falls monitoring). Questionnaires were given before and after people used the robot. While one limitation may be that only people who agreed to take part in the study completed the questionnaires, not all those people

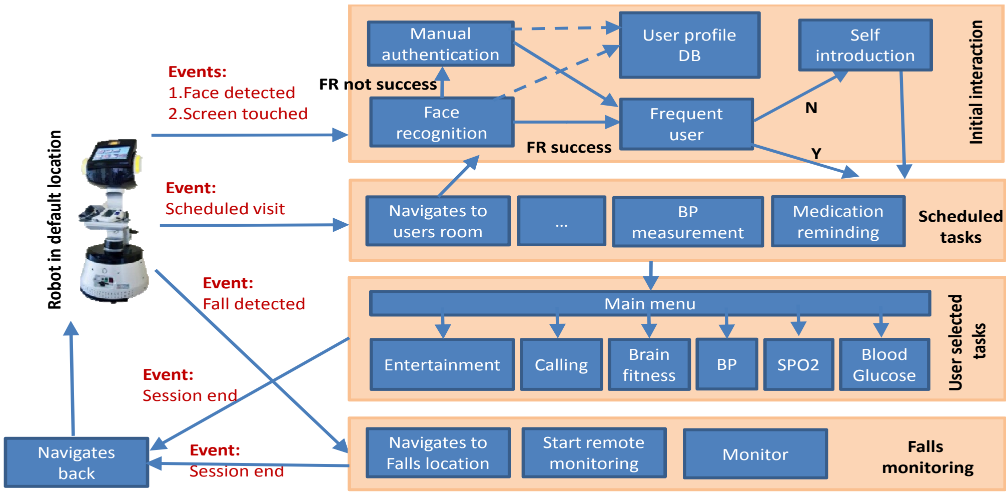


Fig. 4. Scenarios for private, pubic and falls studies.

chose to use the robot. Participants were asked to rate the robot using the following questions:

- **Q1**: How much did you enjoy interacting with that robot?
- **Q2**: On a scale of 0 to 100, how well would you rate your interaction with the robot overall?
- **Q3**: On a scale of 0 to 100, how much would you like to interact with the robot again?

In addition to these other quantitative and qualitative results related to user acceptance were collected. These are reported elsewhere. In addition, several observations were made and lessons were learned. We see these as very useful in future designs for older care robots:

a) Increased involvement of SMEs: The customizability enabled SMEs to more closely participate in the development process. Some could use the XML-based language for robot behavior authoring and some could use it with software engineers. On the other hand, software engineers could rapidly apply the changes requested, using the same framework.

b) Increased stakeholder participation: Changes suggested by the stakeholders could be implemented within a very short time.

c) Field testing: Software issues could not all be resolved in the laboratory and it is important to plan for a substantial period of field testing.

d) Software integration: In this type of research, usually a complex distributed software is built integrating several research software modules. Therefore, a considerable time and effort should be allocated for software integration and integration testing.

e) Practical software component deployment and management: In the field, engineers spent more time than expected manually starting and managing software and hardware components on the robot. More automated component deployment and management should be provided. Reliable operation is critical if the study is to be successful and software should be included to monitor robot operation and notify any issues. Remote monitoring should be provided, so engineers can check robot operation easily.

f) Project planning: There are many technical, non-technical, and non-functional requirements that must be managed, and a detailed project plan is essential.

g) Confidentiality: Researchers should be aware that data collected must be kept confidential and no reports should be published that include participants' identifying information unless the participant has given his or her written consent for this information to be released.

Lessons learned about the operation of the study are elaborated in [17].

VIII. Conclusion

This paper presented a mobile service robot designed for older people. The software development methodology, software framework, implementation details, field studies, and results were presented. Unique features of software framework, which supported rapid prototyping and customizability, were presented.

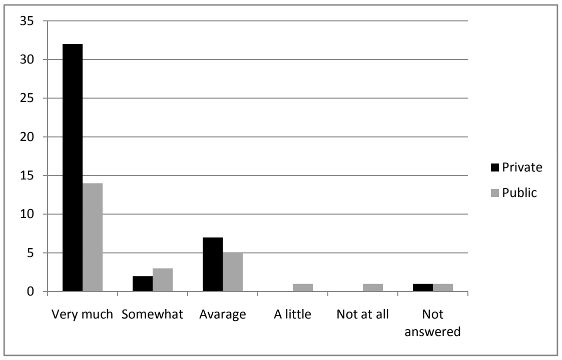
The robot was extensively tested by conducting 3 studies in a retirement village.

In this research, the robot interacted with users via synthesized speech and a touch screen, and provided various services and collected data. This is an ongoing research project and the collected data will be used for the design of the next trial version of the robot.

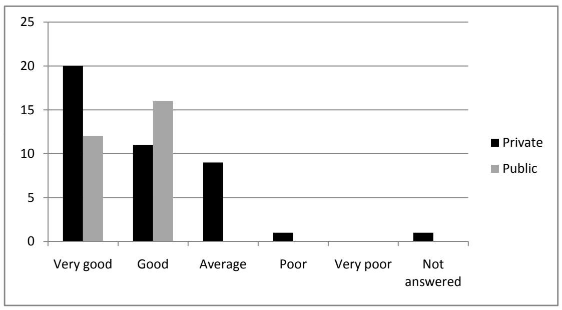
The success of the robot from the users' point of view is shown in Fig. 5. The participants gave the robot high overall ratings.

From the experience of this study, several future improvements were identified. The key ones are given below:

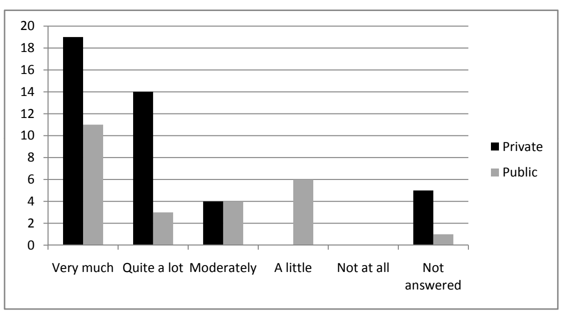
- *Graphical tools*: Although the XML-based language can be used for authoring and customizing robot behaviours, it has to be done very carefully as XML editing is quite error prone. Therefore, a graphical tool, which can use



(a) How much did you enjoy interacting with that robot?



(b) How well would you rate your interaction with the robot overall?



(c) How much would you like to interact with the robot again?

Fig. 5. Overall user satisfaction.

the current XML representation as a meta-representation is required.

- *Support for multiple robot middleware*: The current version of the software runs only on one proprietary middleware called ROCOS. It should be extended to support other mainstream robot middleware such as ROS, OpenRTM, Opros etc.
- *Practical software issues are more significant than in the laboratory environment*: Stakeholder participation, field testing, integration, and component management are critical to a successful trial.

ACKNOWLEDGMENT

This work was supported by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and Korea Evaluation Institute of Industrial Technology (KEIT). [KI001836, Development of Mediated Interface Technology for HRI]. We acknowledge the support of the New Zealand Ministry for Science and Innovation. We thank ETRI for their contributions and help with the work.

REFERENCES

[1] United Nations, *World Population Prospects: The 2006 revision.* New York: United Nations, 2006.

[2] M. K. Murray, "The Nursing Shortage: Past, Present, and Future," *Journal of Nursing Administration*, vol. 32, p. 79, 2002.

[3] N. Super, "Who will be there to care? The growing gap between caregiver supply and demand," in *National Health Policy Forum*, George Washington University, Washington DC, 2002.

[4] S. P. Levine, D. A. Bell, L. A. Jaros, R. C. Simpson, Y. Koren, J. Borenstein, "The NavChair Assistive Wheelchair Navigation System," *IEEE Transactions on Rehabilitation Engineering*, vol.7, no.4, pp.443 – 451, Dec 1999.

[5] X. Zhang, Y. Liu, F. Zhang, J. Ren, Y. Sun, Q. Yang, H. Huang, "On Design and Implementation of Neural-Machine Interface for Artificial Legs," *IEEE Transactions on Industrial Informatics*, vol. -, no. -, pp. -, 0.

[6] N. Neckel, W. Wisman, J. Hidler, "Limb Alignment and Kinematics Inside a Lokomat Robotic Orthosis," in *Proceedings of the 28th Annual International Conference of the IEEEEngineering in Medicine and Biology Society*, pp.2698 – 2701, Aug-Sep. 2006.

[7] A. H. Zahraee, J. K. Paik, J. Szewczyk, G. Morel, "Toward the Development of a Hand-Held Surgical Robot for Laparoscopy," *IEEE/ASME Transactions on Mechatronics*, vol.15, no.6, pp.853 – 861, Dec. 2010.

[8] H. I. Krebs, J. J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B. T. Volpe, and N. Hogan, "Rehabilitation robotics: Performance-based progressive robot-assisted therapy," *Autonomous Robots*, vol. 15, pp. 7-20, 2003.

[9] M. J. Mataric, J. Eriksson, D. J. Feil-Seifer, and C. J. Winstein, "Socially assistive robotics for post-stroke rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 4, p. 5, 2007.

[10] A. Tapus, M. J. Mataric, B. Scasselati, "Socially assistive robotics [Grand Challenges of Robotics]," *IEEE Robotics & Automation Magazine*, vol.14, no.1, pp.35 – 42, March 2007.

[11] C. Jayawardena, I. H. Kuo, U. Unger, A. Igic, R. Wong, C. I. Watson, R. Q. Stafford, E. Broadbent, P. Tiwari, J. Warren, J. Sohn, B. A. MacDonald, "Deployment of a service robot to help older people," in Procedings of *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.5990-5995, Oct. 2010.

[12] R. Q. Stafford, E. Broadbent, C. Jayawardena, U. Unger, I. H. Kuo, A. Igic, R. Wong, N. Kerse, C. Watson, B. A. MacDonald, "Improved robot attitudes and emotions at a retirement home after meeting a robot," *2010 IEEE RO-MAN*, pp.82 – 87, Sept. 2010.

[13] A. W. Black, P. Taylor, and R. Caley. (June, 1999). The festival speech synthesis system. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival

[14] C. I. Watson, J. Teutenberg, L. Thompson, S. Roehling, A. Igic, "How to build a New Zealand voice," presented at the New Zealand Linguistic Society Conference, 2009.

[15] *Localization System StarGazer for Intelligent Robots*, Hagisonic Co. Ltd., Daejeon, Korea, 2009.

[16] R. Q. Stafford, B. A. MacDonald, C .Jayawardena, D. M. Wegner, E. Broadbent, "Does the robot have a mind of its own? Predictors of healthcare robot use in a retirement village," *Journal of Human Robot Interaction*, [Accepted].

[17] E. Broadbent, C. Jayawardena, N. Kerse, R. Q. Stafford, B. A. MacDonald, 'Human-robot interaction research to improve quality of life in elder care - An approach and issues," Presented at the Workshop on Human-Robot Interaction in Elder Care, AAAI 2011, San Francisco, August 7-11.